2025 IEEE/SICE International Symposium on System Integration (SII)
January 21-24, 2025, Munich, Germany

Teleoperation Experience Like VR Games: Generating Object-Grasping Motions Based on Predictive Learning

Ryuya Shuto[12], Pin-Chu Yang[12], Naoki Hashimoto[12], Mohammed Al-Sada[23] and Tetsuya Ogata[14]

Abstract— **Teleoperation is popular due to its several advantages, including the ability to control a robot from a distance and the capacity for the operator to manage the robot safely. However, teleoperation also presents challenges, including operational complexity and the requirement for a certain level of proficiency from the operator. For instance, when attempting to grasp an object via teleoperation, issues such as communication delays, inadequate feedback from the robot to the operator, and the complexity of the grasping trajectory can arise. To address this issue, we propose an intuitive teleoperation method that facilitates data collection using VR devices and a technique for generating object-grasping motions through predictive learning with the collected data. First, we collect the motion data while the robot is teleoperated using a VR device. The collected motion data is used to create a predictive model through predictive learning, which in turn is used to generate object-grasping motions. This approach allows us to collect motion data suitable for machine learning while performing intuitive teleoperation. It also enables the generation of object-grasping motions with simple operations, making robot teleoperation experience similar to a VR game. We evaluated our approach's ability to generate object-grasping motion with predictive model. The results show that our approach can generate object-grasping motions with a certain level of success. In light of our results, we discussed the factors that pose challenges to predictive learning and explored the future prospects of this approach.**

I. Introduction

Recently, industrial robots and humanoid robots have been increasingly utilized in various applications and environments. Real-time teleoperation is often employed to control these robots [1] using a variety of methods, such as VR devices, motion capture systems, joysticks, and mice [2]. In particular, VR devices are considered ideal for teleoperation due to their advantages over traditional interfaces, as they offer higher fidelity and flexibility for control and feedback [3]. Teleoperation offers several advantages as they are generally intuitive to learn and safe, which is especially important when deploying the robot at hazardous environments [4]. However, teleoperation also presents challenges, requiring a learning and familiarization phase that is interface-dependent, and largely depends on the operator's proficiency to control the robot and fulfill the task requirements [3]. For example, when attempting to grasp an object via teleoperation, issues such as communication delays, insufficient feedback from the robot to the operator, and the complexity of the grasping trajectory may arise. To address these challenges, gesture-based teleoperation [5] and intuitive teleoperation for dual-armed robots using one-handed gestures [6] have been explored. These studies propose robot control methods that use gestures to switch between human teleoperation and autonomous control, where the robot automatically recognizes and grasps objects.

In recent years, VR games have become increasingly popular, offering users the ability to interact with virtual environments in an intuitive manner. For instance, players can grasp virtual objects by moving their hand near the object and pressing a button on the VR controller, which prompts the object to automatically move toward the hand and simulate a grasping action. This approach allows users to effortlessly grasp even complex objects, enhancing the interactivity and intuitiveness of VR games. By eliminating the need to focus on precise grasping mechanics, this method reduces both the cognitive and physical effort required, contributing to a seamless and engaging user experience.

Inspired by this intuitive grasping mechanisms in VR games, we propose a teleoperation method that leverages predictive learning to replicate similar grasping experiences for robots. The objective of this research is to develop a teleoperation system utilizing a VR device, along with an assistive teleoperation system which enables a robot to grasp an object through a simple operation, bringing the robot's hand close to the target object and pressing a button on VR device controller. To achieve this, we first designed a teleoperation system capable of collecting motion data via a VR device. This collected data is then utilized for predictive learning, allowing the robot to grasp objects through a straightforward button press.

Our evaluation centered on two primary objectives: (1) collecting object-grasping motion data through teleoperation using the proposed method, and (2) verifying the accuracy of the predictive model trained with the collected motion data. The results demonstrate that our approach effectively collects robot motion data suitable for machine learning applications. Moreover, we achieved a certain level of success in generating object-grasping motions like user experience from VR games.

We discussed the results and highlighted potential use

[1]R. Shuto, P. -C. Yang, N.Hashimoto and T. Ogata are with Waseda University, Tokyo, Japan
[2]R. Shuto, P. -C. Yang, N.Hashimoto and M. Al-Sada are also with Cutieroid Project: www.cutieroid.com
[3]M. Al-Sada is also with Qatar University, Doha, Qatar
[4]T. Ogata is also with WISE at Waseda University, Tokyo, Japan
imijiku@fuji.waseda.jp
yang.pinchu@aoni.waseda.jp
purple.columbine@ruri.waseda.jp
mohammed.alsada@qu.edu.qa
ogata@waseda.jp

This project is jointly supported by JST Moonshot R&D Grant Number JPMJMS2031-2-1 and Qatar Japan Research Collaboration fund (QJRC)-grant number M-QJRC-2023-325.

cases that could benefit from the generation of object-grasping motions. Finally, we conclude our work and outline future research directions.

This paper contributes by intuitive robot motion data collection through teleoperation using a VR device and simplification of grasping objects in teleoperation using predictive learning.

II. Related Works

Our approach extends three strands of related research: intuitive teleoperation, collection of robot motion data, and generating humanoid robot motions based on a procedural animation IK Rig method. We discuss each of these as follows:

Teleoperation is widely regarded as an effective control method for robots and has been the focus of extensive research [7][8][9]. However, teleoperation—particularly without force feedback—poses challenges for individuals who are not familiar with robot control [10]. As a result, numerous studies have explored more intuitive and user-friendly teleoperation techniques to make control easier [11][12][13][14].

When controlling a robot, collecting its motion data is crucial. In particular, the motion data collected from a robot controlled through intuitive teleoperation is more precise and closely resembles human motion. These data can be leveraged for motion analysis and applied to machine learning, enabling improvements in robotic performance and adaptability [11][15][16].

Stanton et al [11] used a full-body motion capture suit to teleoperate a humanoid robot. They collected the motion capture data and utilized it to train a neural network capable of performing complex tasks. However, the study did not involve grasping objects. Zhang et al [15] employed a VR headset and hand-tracking hardware to teleoperate a dual-armed robot, allowing it to perform complex tasks through imitation learning, including object-grasping. However, since the robot used was dual-armed rather than humanoid, its applicability to Human-Robot Interaction (HRI) remains uncertain.

Yang et al [17] proposed a method for generating natural motions that can adapt to various situations for humanoid robots, using an IK rig. This method is based on the IK rig, a typical procedural animation technique commonly used in the animation and game industries. From a 3D model of the robot, two armatures were created: a humanoid rig for procedural animation and a robot rig for controlling the robot. The humanoid rig, which uses procedural animation, was converted to IK rig animation and applied to the robot rig to control the humanoid robot. However, this method [17] is limited to pre-prepared animations and is challenging to use for real-time robot teleoperation.

In our paper, we developed a real-time teleoperation system that facilitates motion data collection using VR devices, building upon the methods introduced in previous research [17]. We then used the system to collect data on the object-grasping behavior of humanoid robots, applicable to HRI, and leveraged the data to enable generating object-grasping motion through predictive learning. By combining this motion generation with a teleoperation system, a VR game-like teleoperation experience is realized, in which the robot's hand is brought close to an object, and the motion of grabbing the object is generated by pressing a foot pedal.

III. Proposed Method

In this section, we detail our proposed method, which comprises three main components: (A) a teleoperation system utilizing a VR device for intuitive control, (B) a method for collecting object-grasping motion data, and (C) an object-grasping motion generation process based on predictive learning. Fig. 1 shows overview of our proposed method.

A. Teleoperation System Implementation

Fig. 2 shows an overview of the teleoperation system using a VR device. We control the robot's VRM [18] model, one commonly used file format for humanoid 3D models, using a VR device and the data is transmitted to Unity3D [19]. We used the VRM model due to its two key characteristics: ease of control with VR devices and its compatibility with existing methods [17] for controlling humanoid rigs. We then convert the motion into an IK rig animation for a robot rig and operate the robot in a similar manner to previous work [17].

1) Software: Virtual Motion Capture [20] and SteamVR [21] were software used to control VRM models with VR devices. Specifically, VRM models in Virtual Motion Capture were manipulated by VR devices through SteamVR. Easy Virtual Motion Capture For Unity [22], a Unity3D asset, was then used to transmit this information to Unity3D.

2) Hardware: We used Meta Quest 3 [23] to teleoperate a robot and collect robot motion data. AHF-HATSUKI Mk.I [24] was used as the teleoperated robot, and HatsuHand Mk.I [25] was used as the teleoperated hands (Fig. 3). Tundra Tracker [26] was utilized to collect data beyond robot motion, such as the position of the object to be grasped.

B. Object-grasping motion data collection method

In our motion data collection method, we record the movement of each joint in the VRM model within Unity3D, capturing its transform data to serve as the target motion for the robot. The collected data consists of time-series information, with both position and rotation being recorded per frames. In this study, the collected time-series data was stored as 30 frames of data per second. Specifically, the position data $\boldsymbol{p}$ is represented as a 3-dimensional vector in Unity3D's left-handed coordinate system, while the rotation data $\boldsymbol{q}$ is stored as a Quaternion, allowing for efficient representation and manipulation of orientation in 3D space.

We collected grasping a box object motions using this method. Table. I presents the parameters in the motion data, which consist of the position and rotation information for 20 joints, including the head, neck, shoulders, arms, elbows, wrists, and five fingers—these being the moving parts of the robot. Unity3D represents an object's position and rotation in

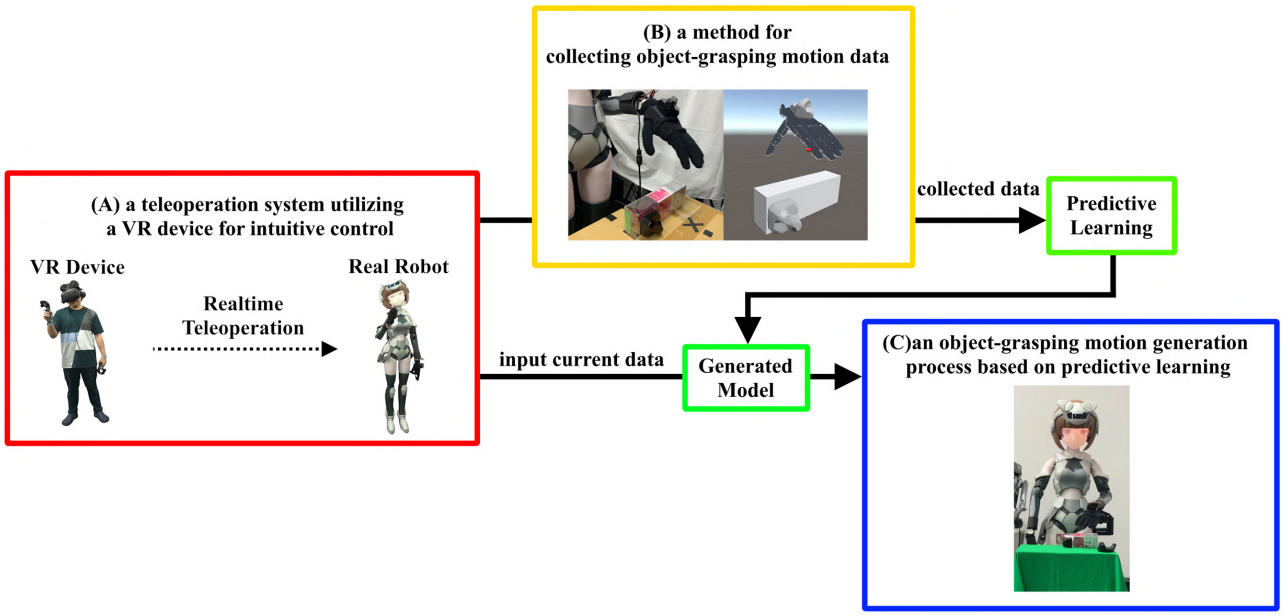

Fig. 1. Overview of our proposed method

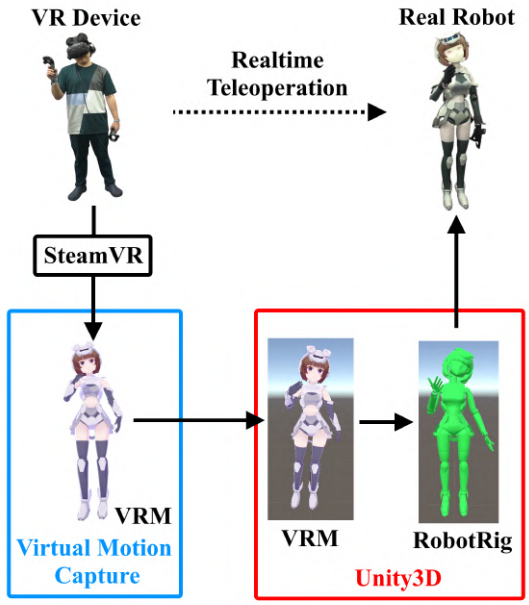

Fig. 2. Overview of our proposed teleoperation system using a VR device

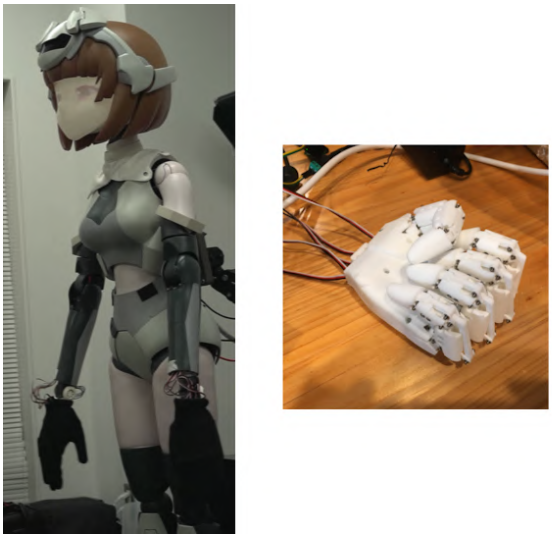

Fig. 3. AHF-HATSUKI Mk.I, Teleoperated robot and HatsuHand Mk.I, teleoperated hands

two ways: Global Position and Global Rotation, relative to the scene's origin, and Local Position and Local Rotation, relative to the object's parent. For the VRM model, we mainly collected global position and rotation data, while for the fingers, we specifically recorded local rotation data due to the importance of the angles relative to the wrist. To simplify the generation of object-grasping motions, we also included relative position and rotation data between the left hand and the box using Tundra Tracker. In total, the motion data is 147 dimensions. Additionally, we attached colliders to left hand and the box in Unity3D to define the initial position of the grasp and the moment of collision, marking the start of the grasping (Fig. 4, Fig. 5 and Fig. 6). The object-grasping motion sequence begins with the collision between the left hand and the box and ends after lifting the box and holding it in the air for one second.

C. Predictive learning

Predictive learning is a method that enables the real-time prediction of actions suitable for real-world scenarios based on past learning experiences[27]. This approach allows for the performance of flexible actions in untrained environments or on untrained targets, facilitating efficient

TABLE I
COLLECTED DATA

Data	Description	Dimension
VRM Data except Fingers	Global Position and Rotation of Head, Neck, Shoulder, Arm, Elbow and Wrist on each side	70
VRM Fingers Data	Global Position and Local Rotation of Five fingers on each side	70
Tracker Data	Relative Position and Rotation between Left Hand's Tracker Data and Box's Tracker Data	7

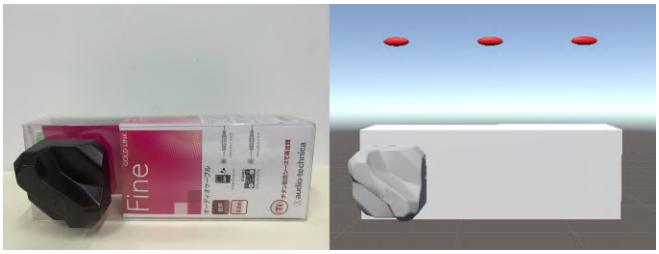

Fig. 4. The used Box with Tundra Tracker along side its virtual counterpart in Unity3D. The small red spheres in Unity3D are used to detect collisions with left hand during movement. Their placements represent the upper left, middle, and upper right of the used Box.

learning and generation of actions with minimal teaching costs. For example, tasks that require flexibility to adapt to the environment, such as the Put-In-Box task[28], the door-opening task[29], and the repeatable folding task[30], have been successfully achieved through predictive learning. Based on this foundation, we adopted this method to generate object-grasping motions in this research.

To facilitate learning, we standardized the data collected through teleoperation with the VR device by calculating the mean and standard deviation for each item separately. For items with standard deviations close to zero, the value was replaced with the overall standard deviation to avoid division by zero. When creating the dataset, we adjusted the number of frames in all data to match the longest motion by repeating the last frame of each motion. This ensured consistency in the number of frames across all motions.

The model developed through predictive learning is used to generate object-grasping motions by inferring from the input of the robot's posture data and the object's position data. Fig. 7 shows the flow of motion prediction using the generated model. Specifically, the input data is a tensor with a batch size of 10, 346 frames, and a data size of 147(Table. I), while the output data is a tensor with a batch size of 10, 200 frames, and a data size of 147(Table. I). The number of frames in the output data was set to 200 because the object-grasping motion took approximately 7 seconds during data collection. Since the data used for predictive learning is time-series data, we employed the LSTM layer, which is one of the layers suitable for time-series data analysis [31], in

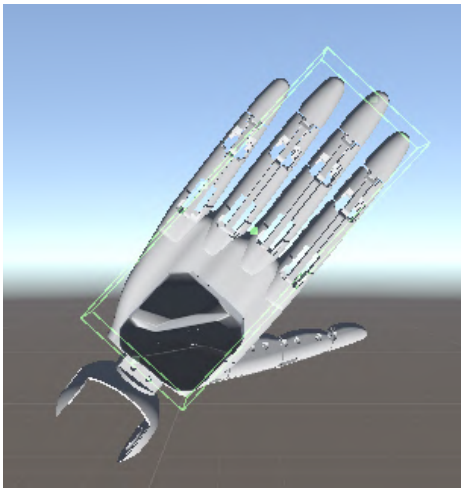

Fig. 5. Left Hand with Tracker in Unity3D. Green frame detects collisions with box.

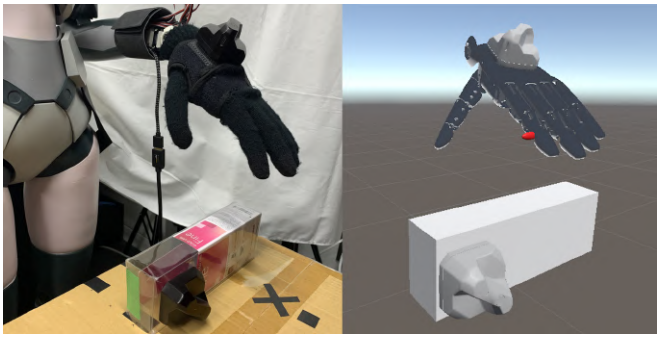

Fig. 6. Real and virtual robot snapshot of the hand while collecting object-grasping motion data in reality and Unity3D. We defined the start of motion data collection as the moment when the red ball collided with the left hand collider in Unity3D, as depicted in the figure on the right.

the predictive learning model. We used PyTorch 1.13.1 for predictive learning and fixed the seed for random number generation in PyTorch and Python to 23456. Table. II shows the parameters used for predictive learning.

The loss function was the average of the mean squared error (MSE) and the mean absolute error (MAE). Adam was used as the optimization algorithm. During predictive learning, since some motions involve initial alignment, which may result in a lack of information for generating object-grasping motions, we considered the first 20 steps as a run-up phase, and the backpropagation of errors in the motion data is not performed during this period. To enhance the model's generalizability, noise was added to the input data with a certain probability. In each batch, noise was assigned with a 0.5 probability, and a uniform random value in the range [-0.5, 0.5] was added with a 0.2 probability to all items.

When generating motions in practice, inference is performed within Unity3D. This inference is triggered by simple actions, such as pressing a button on the keyboard or VR controller, based on the settings within Unity3D.

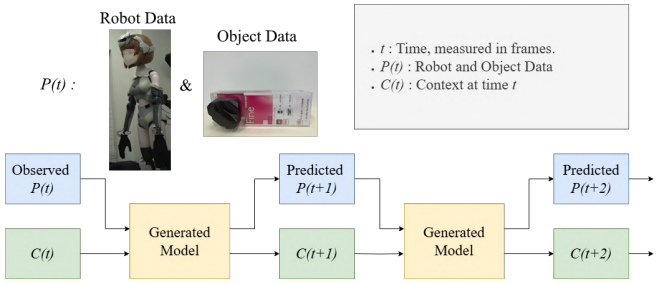

Fig. 7. The flow of motion prediction using the generated model. By initially inputting the Observed data, the Context is updated to output the Predicted data for the next time step. This Predicted data is then used as input to generate the data for the subsequent time step, and this process is repeated iteratively.

TABLE II

PARAMETER FOR PREDICTIVE LEARNING

Symbol	Description	Value
α	Learning Rate	0.001
	Number of epoch	30000
	Execution probability of Dropout layer	0.2
	Input data size of LSTM layer	147
	The number of hidden layers	2
	Dropout ratio	0.2
	Input data size of output layer	512
	Output data size of output layer	147
β_1	β_1 of Adam	0.9
β_2	β_2 of Adam	0.999

IV. EXPERIMENT AND RESULT

A. Experiment setup

In this section, we explains the details of experimental environment and setup for predictive learning. Experimental environment includes the following: positioning of the box and robot, type and number of collected object-grasping motion data. Setup for predictive learning includes its epochs and loss curves.

1) Experimental environment: Fig. 8 shows the experimental environment. In this experiment, we collected data on robot motions for grasping a box-shaped object with its left hand. Specifically, we gathered 5 data samples for grasping the box at 5 different positions from 3 directions, resulting in a total of 75 object-grasping motion data points. The box positions used for data collection and model accuracy validation are the 9 positions shown in Fig. 9. These positions are within the robot's workspace, allowing it to grasp the box with its left hand comfortably. For data collection, we selected positions 1, 3, 5, 7, and 9. We believed that collecting object-grasping data at the maximum, minimum, and intermediate positions in the front/rear and left/right directions would facilitate the generation of object-grasping motions for untrained positions. In positions on the left side of the robot — positions 3, 6, and 9 - part of the box was not touching the desk. We anticipated that predictive learning would enable the generation of more varied types of object-grasping motions with data in these positions.

The 3 directions from which the object-grasping motion is initiated are above, above right, and above left (Fig. 4).

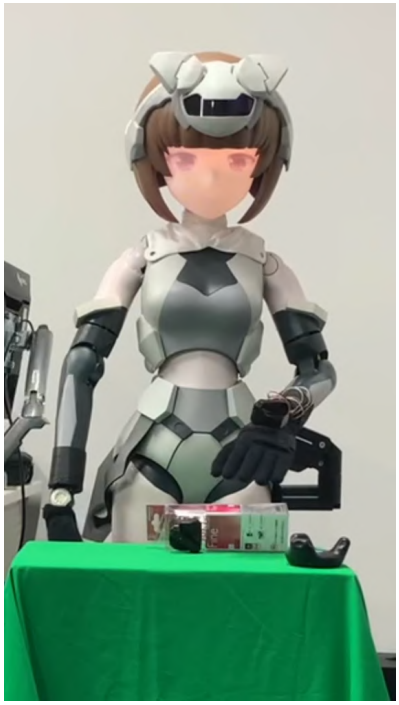

Fig. 8. The experimental environment consists of the teleoperated robot, hands and the object to be grasped.

By collecting object-grasping motion data from multiple directions, we aim to generate more natural object-grasping motions.

2) Predictive learning: We divided 75 data to 53 learning data and 22 test data. Predictive learning was run for 60,000 epochs. We used the model trained for 30,000 epochs to validate its performance because training for more than 37,000 epochs resulted in loss explosion.

B. Results

The main objectives of our evaluation are: 1) to collect object-grasping motion data through teleoperation using the proposed method, and 2) to verify the accuracy of the model by applying predictive learning to the collected motion data. We collected data on grasping a box-shaped object from three different directions. Subsequently, we generated object-grasping motions using a model trained through predictive learning with LSTM, and measured the success rate of these generated motions. We were able to collect object-grasping motion data with our propose method. Fig. 10 shows the movement of the left arm in one of the data. We evaluated the success rate of the generated object-grasping motions after the hand was brought close to the box using the predictive model. A motion was considered successful if the robot could hold the box in the air for one second. In cases where the motion generation terminated during the grasping and lifting process, we considered the motion successful if the robot could hold the box in the air for one second after continuing the grasping motion and lifting the robot's left arm. The delay between starting the motion generation and the actual execution of the motion was less than 1/60

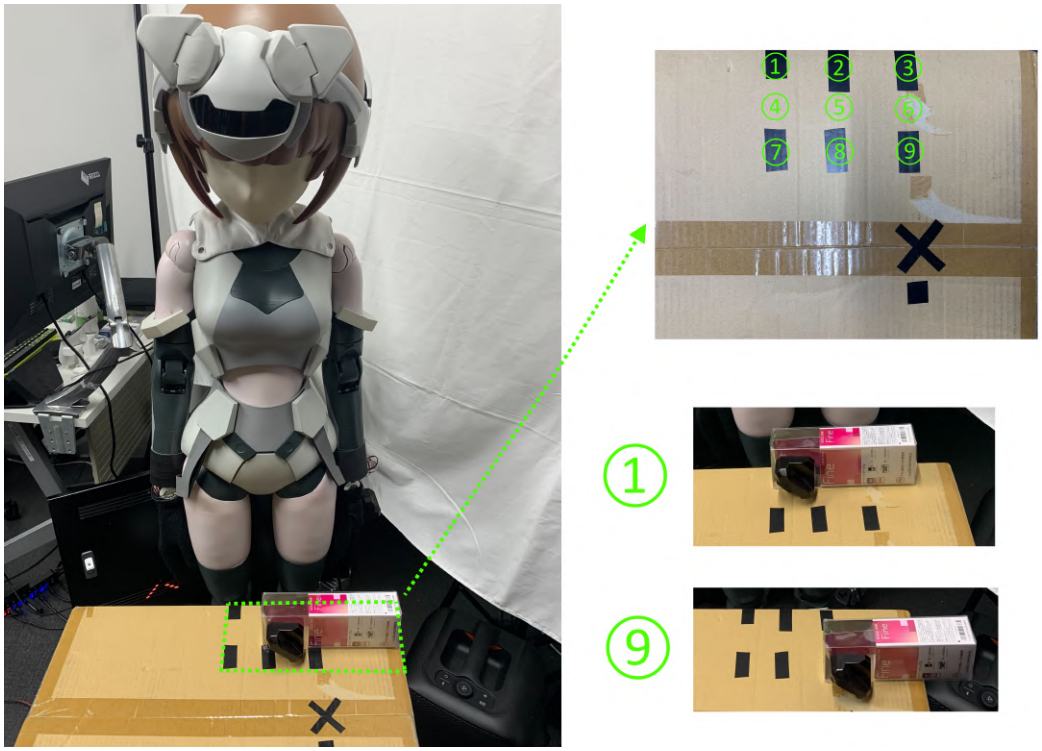

Fig. 9. This picture shows the robot, the box to be manipulated as well as the nine box positions (black labeled on the card board box). We used positions 1, 3, 5, 7, and 9 for data collection. We present images with the box placed positions 1 and 9 as examples. Part of the box in positions 3,6 and 9 was not touching the desk.

second. We evaluated the success rate across 9 positions (Fig. 9). Positions 2, 4, 6, and 8 were untrained positions. We generated motions two times from three directions: above, above right, and above left of the box. In total, we generated 6 motions for each position. Table. III shows the number of successes and the success rate for generating object-grasping motions with the predictive model at each position.

TABLE III
EVALUATION OF OBJECT-GRASPING MOTION GENERATION

Position	Number of successes/fails	Success rate
1	5/1	83.3%
2	3/3	50%
3	2/4	33.3%
4	5/1	83.3%
5	4/2	66.7%
6	0/6	Failed
7	6/0	100%
8	0/6	Failed
9	0/6	Failed
Total	25/29	46.3%
Trained	17/13	56.7%
Untrained	8/16	33.3%

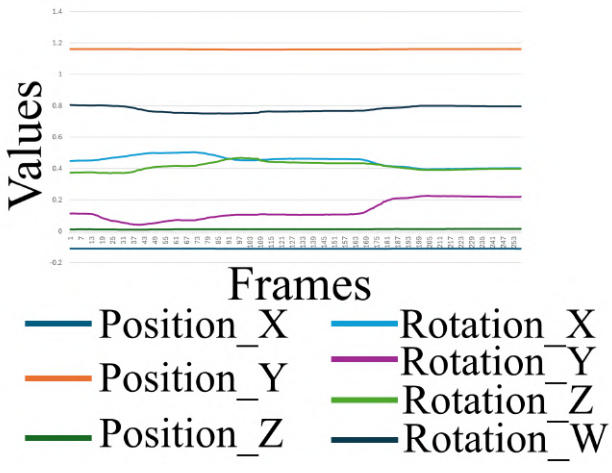

Fig. 10. The movement of the left arm joint in the VRM model, included in an object-grasping motion data

Table. IV shows the distance from the robot and the success rate of object-grasping motion generation at each position. The distance from the robot refers to the distance between the Spine joint of the VRM model and the box object in Unity3D.

Fig. 11 shows plots of Table. IV. The correlation coefficient for this graph was -0.84.

Although the results were less favorable in some positions, object grasps were still achieved in untrained positions, indicating potential for generalization.

V. DISCUSSION

First, we demonstrate that our proposed method is capable of collecting robot motion data suitable for machine learning applications. Next, we discuss the performance of the models generated through predictive learning. According to Table. III, the success rate is lower for positions on the left side of

TABLE IV

DISTANCE FROM ROBOT AND SUCCESS RATE OF OBJECT-GRASPING MOTION GENERATION IN EACH POSITION

Position	Distance from robot [cm]	Success rate
1	22.79485	83.3%
2	26.81186	50%
3	31.58762	33.3%
4	25.21516	83.3%
5	28.30249	66.7%
6	32.95531	Failed
7	27.45277	100%
8	30.54475	Failed
9	35.17581	Failed

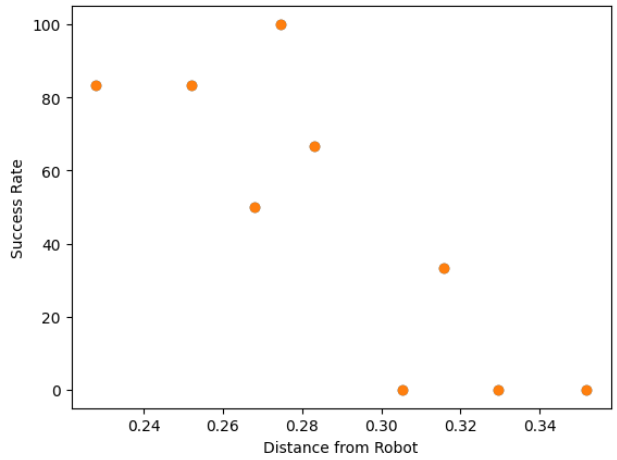

Fig. 11. Distance from robot and success rate of object-grasping motion generation

the robot, specifically positions 3, 6, and 9. We hypothesize that the issue may arise because a portion of the box does not make contact with the desk, thereby complicating predictive learning. As shown in Fig. 11, the further a position is from the robot, the more difficult it becomes to generate object-grasping motions. In fact, the success rate of object-grasping motion generation decreases sequentially across position columns 1, 4, 7; columns 2, 5, 8; and columns 3, 6, 9. Specifically, the success rate for generating object-grasping motions at positions 6 and 9, where the box is farther from the robot and a portion is not in contact with the desk, failed. In contrast, the success rate at learned positions, excluding positions 3, 6, and 9, was 83.3%, whereas at unlearned positions, it was 44.4%. While the success rate was not particularly high in certain positions, the results highlight valuable insights. For instance, the system achieved a success rate of 33.3% in untrained positions, suggesting a degree of generalization beyond trained data. Additionally, the 56.7% success rate in trained positions indicates that the proposed data collection method effectively supports the framework's objectives. These findings suggest steady progress toward enabling predictive learning and generating grasping-motions.

The primary reasons for the failure of generating object-grasping motions are as follows: the left hand was positioned behind the box, the box fell off the desk during the grasp attempt, and the hand mistakenly attempted to grasp the portion tracker attached, resulting in improper lifting. In this experiment, only the relative position data between the left hand and the box was collected to simplify the learning process. Consequently, it is believed that the box was unable to accurately predict how far to reach and how to move the hand when it was positioned far away. HatsuHand Mk.I, the hand used in this study, has one degree of freedom (DOF) for each finger, totaling 5 DOFs. However, it was sometimes challenging to grasp objects during teleoperation with this hand. We hypothesized that adding a DOF to the thumb base would make the grasping motion easier. In addition, we used LSTM as the predictive learning model due to its suitability for time-series data analysis in this study. However, it would also be worthwhile to explore training with other models.

In this experiment, the extent to which object grasping via teleoperation was made easier by the proposed method was not evaluated. As part of a user study, it would be possible to assess this aspect by comparing the time taken to lift an object using the proposed grasping motion generation with and without the use of the method, among users who are not familiar with robot teleoperation through a VR device.

Use cases for generating object-grasping motions include applications such as robots performing handshakes or selling products. In these scenarios, the robot must be capable of recognizing objects and autonomously grasping them at the appropriate moment. For instance, in a handshake scenario, the robot would need to recognize the other person's hand and perform the handshake with natural movements. Similarly, in a product-selling scenario, the robot would autonomously grasp an item and hand it over to the customer. In such tasks, accurate and swift object grasping can significantly enhance the robot's usability and efficiency.

VI. CONCLUSION AND FUTURE WORK

In this study, we propose a method for robot teleoperation that utilizes a VR device to collect motion data and generates object-grasping motions through predictive learning using the collected data. First, motion data is collected while the robot is teleoperated using a VR device. The main advantages of this approach are that the VR device facilitates more intuitive robot teleoperation and enables the collection of natural motion data suitable for machine learning. The collected motion data is used to create a predictive model through predictive learning, which in turn is used to generate object-grasping motions. The primary benefit of this method is its applicability to robot teleoperation with a user experience similar to VR games, allowing objects to be grasped with simple movements. The success rate of the grasping motion generation was evaluated, demonstrating that the proposed method can produce a certain degree of effective object-grasping motion.

Our future work is focused on two main directions. First, as an extension of this research, the method can be applied to various types of objects. In addition to grasping objects, it could potentially be used to grasp tools and operate them with simple actions, similar to interactions

in VR games. Furthermore, introducing alternative methods for recognizing the object's position, such as camera-based recognition instead of using trackers, could improve adaptability for generating object-grasping motions for new objects. Secondly, this research can also be applied to HRI (human-robot interaction) scenarios such as handshakes and product sales, which were mentioned as use cases earlier. By utilizing object-grasping motion generation, robots can perform these tasks and provide customers with a new and innovative experience.

ACKNOWLEDGEMENTS

This project is jointly supported by JST Moonshot R&D Grant Number JPMJMS2031-2-1 and Qatar Japan Research Collaboration fund (QJRC)- grant number M-QJRC-2023-325.

REFERENCES

[1] S. Mick, M. Lapeyre, P. Rouanet, C. Halgand, J. Benois-Pineau, F. Paclet, D. Cattaert, P.-Y. Oudeyer, and A. de Rugy, "Reachy, a 3d-printed human-like robotic arm as a testbed for human-robot control strategies," *Frontiers in Neurorobotics*, vol. 13, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:201116898

[2] P. M. Kebria, H. Abdi, M. M. Dalvand, A. Khosravi, and S. Nahavandi, "Control methods for internet-based teleoperation systems: A review," *IEEE Transactions on Human-Machine Systems*, vol. 49, no. 1, pp. 32–46, 2019.

[3] Y. Zhu, B. Jiang, Q. Chen, T. Aoyama, and Y. Hasegawa, "A shared control framework for enhanced grasping performance in teleoperation," *IEEE Access*, vol. 11, pp. 69 204–69 215, 2023.

[4] I. Tsitsimpelis, C. J. Taylor, B. Lennox, and M. J. Joyce, "A review of ground-based robotic systems for the characterization of nuclear environments," *Progress in Nuclear Energy*, vol. 111, pp. 109–124, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0149197018302750

[5] Z. Xue, X. Chen, Y. He, H. Cao, and S. Tian, "Gesture- and vision-based automatic grasping and flexible placement in teleoperation," *The International Journal of Advanced Manufacturing Technology*, vol. 122, no. 1, pp. 117–132, Sept. 2022. [Online]. Available: https://doi.org/10.1007/s00170-021-08585-z

[6] M. Laghi, M. Maimeri, M. Marchand, C. Leparoux, M. Catalano, A. Ajoudani, and A. Bicchi, "Shared-autonomy control for intuitive bimanual tele-manipulation," in *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*, 2018, pp. 1–9.

[7] T. Sheridan, "Teleoperation, telerobotics and telepresence: A progress report," *Control Engineering Practice*, vol. 3, no. 2, pp. 205–214, 1995. [Online]. Available: https://www.sciencedirect.com/science/article/pii/096706619400078U

[8] S. Lichiardopol, *A survey on teleoperation*, ser. DCT rapporten. Technische Universiteit Eindhoven, 2007, dCT 2007.155.

[9] K. Darvish, L. Penco, J. Ramos, R. Cisneros, J. Pratt, E. Yoshida, S. Ivaldi, and D. Pucci, "Teleoperation of humanoid robots: A survey," *IEEE Transactions on Robotics*, vol. 39, no. 3, pp. 1706–1727, 2023.

[10] D. J. Rea and S. H. Seo, "Still not solved: A call for renewed focus on user-centered teleoperation interfaces," *Frontiers in Robotics and AI*, vol. 9, 2022. [Online]. Available: https://www.frontiersin.org/journals/robotics-and-ai/articles/10.3389/frobt.2022.704225

[11] C. Stanton, A. Bogdanovych, and E. Ratanasena, "Teleoperation of a humanoid robot using full-body motion capture, example movements, and machine learning," in *Conference: Proceedings of Australasian Conference on Robotics and Automation (ACRA 2012)*, 12 2012.

[12] L. Zhao, Y. Liu, K. Wang, P. Liang, and R. Li, "An intuitive human robot interface for tele-operation," in *2016 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, 2016, pp. 454–459.

[13] C. Wang, X. Chen, Z. Yu, Y. Dong, R. Zhang, and Q. Huang, "Intuitive and versatile full-body teleoperation of a humanoid robot," in *2021 IEEE International Conference on Advanced Robotics and Its Social Impacts (ARSO)*, 2021, pp. 176–181.

[14] P. Wu, Y. Shentu, Z. Yi, X. Lin, and P. Abbeel, "Gello: A general, low-cost, and intuitive teleoperation framework for robot manipulators," 2024. [Online]. Available: https://arxiv.org/abs/2309.13037

[15] 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," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 5628–5635.

[16] M. Seo, S. Han, K. Sim, S. H. Bang, C. Gonzalez, L. Sentis, and Y. Zhu, "Deep imitation learning for humanoid loco-manipulation through human teleoperation," in *2023 IEEE-RAS 22nd International Conference on Humanoid Robots (Humanoids)*, 2023, pp. 1–8.

[17] P.-C. Yang, S. Funabashi, M. Al-Sada, and T. Ogata, "Generating humanoid robot motions based on a procedural animation ik rig method," in *2022 IEEE/SICE International Symposium on System Integration (SII)*, 2022, pp. 491–498.

[18] VRM Consortium, "What is vrm ? what can vrm do ? — vrm," https://vrm.dev/en/vrm/vrm_about/, 2024, [Accessed: 2024-08-30]. [Online]. Available: https://vrm.dev/en/vrm/vrm_about/

[19] Unity Technologies, "Unity real-time development platform 3d, 2d vr & ar engine," https://unity.com/, 2024, [Accessed: 2024-08-30]. [Online]. Available: https://unity.com/

[20] sh_akira, "Virtualmotioncapture — バーチャルモーションキャプチャー," https://vmc.info/, 2024, [Accessed: 2024-08-30]. [Online]. Available: https://vmc.info/

[21] Steam, "Steamvr on steam," https://store.steampowered.com/app/250820/SteamVR/, 2024, [Accessed: 2024-08-30]. [Online]. Available: https://store.steampowered.com/app/250820/SteamVR/

[22] VaNiiShop, "Evmc4u - easyvirtualmotioncaptureforunity - vaniishop - booth," https://booth.pm/en/items/1801535, 2024, [Accessed: 2024-08-30]. [Online]. Available: https://booth.pm/en/items/1801535

[23] Meta, "Meta quest 3: New mixed reality vr headset - shop now — meta store — meta store," https://www.meta.com/quest/quest-3/, 2024, [Accessed: 2024-08-27]. [Online]. Available: https://www.meta.com/quest/quest-3/

[24] Cutieroid Project =Season 2= - Vtuber Hatsuki, "Cutieroid Project," https://www.cutieroid.com/AboutHatsuki, 2024, [Accessed: 2024-08-30]. [Online]. Available: https://www.cutieroid.com/AboutHatsuki

[25] HatsuMuv, "Hatsuhand/en — 株式会社hatsumuv," https://www.hatsumuv.com/en/service/hatsuhand, 2024, [Accessed: 2024-08-30]. [Online]. Available: https://www.hatsumuv.com/en/service/hatsuhand

[26] Tundra Labs, "Tundra tracker (single) – tundra labs," https://tundra-labs.com/en-jp/products/additional-tracker, 2024, [Accessed: 2024-09-10]. [Online]. Available: https://tundra-labs.com/en-jp/products/additional-tracker

[27] K. Suzuki, H. Ito, T. Yamada, K. Kase, and T. Ogata, "Deep predictive learning: Motion learning concept inspired by cognitive robotics," 2024. [Online]. Available: https://arxiv.org/abs/2306.14714

[28] K. Kase, K. Suzuki, P.-C. Yang, H. Mori, and T. Ogata, "Put-in-box task generated from multiple discrete tasks by ahumanoid robot using deep learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 6447–6452.

[29] H. Ito, K. Yamamoto, H. Mori, and T. Ogata, "Efficient multitask learning with an embodied predictive model for door opening and entry with whole-body control," *Science Robotics*, vol. 7, no. 65, p. eaax8177, 2022. [Online]. Available: https://www.science.org/doi/abs/10.1126/scirobotics.aax8177

[30] 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 Robotics and Automation Letters*, vol. 2, no. 2, pp. 397–403, 2017.

[31] B. Lim and S. Zohren, "Time-series forecasting with deep learning: a survey," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 379, no. 2194, p. 20200209, Feb. 2021. [Online]. Available: http://dx.doi.org/10.1098/rsta.2020.0209