

Self-supervised 6-DoF Robot Grasping by Demonstration via Augmented Reality Teleoperation System

Xiwen Dengxiong

Rochester Institute of Technology

Rochester, NY, USA
sd6384@rit.edu

Xueting Wang

Rochester Institute of Technology

Rochester, NY, USA
xw4860@rit.edu

Shi Bai

Figure AI

Sunnyvale, CA, USA
baishi.bona@gmail.com

Yunbo Zhang

Rochester Institute of Technology

Rochester, NY, USA
ywzeie@rit.edu

Abstract—Most existing 6-DoF robot grasping solutions depend on strong supervision on grasp pose to ensure satisfactory performance, which could be laborious and impractical when the robot works in some restricted area. To this end, we propose a self-supervised 6-DoF grasp pose detection framework via an Augmented Reality (AR) teleoperation system that can efficiently learn human demonstrations and provide 6-DoF grasp poses without grasp pose annotations. Specifically, the system collects the human demonstration from the AR environment and contrastively learns the grasping strategy from the demonstration. For the real-world experiment, the proposed system leads to satisfactory grasping abilities and learning to grasp unknown objects within three demonstrations.

I. INTRODUCTION

Unknown object grasp pose detection aims to generate 6-DoF grasp poses for unknown objects. It is a significant task in real-world applications because robots may need to grasp unknown objects under extreme environments, such as nuclear plants [1], underwater environments [2], and outer space [3].

However, detecting grasp pose for unknown objects is challenging because 6-DoF grasp pose needs to be learned from accurate grasp pose annotations. Although the problem can be solved by manually providing accurate grasp poses, physically annotating the robot grasp pose in an extreme environment is impractical. Moreover, limited grasp pose annotations may impact the performance of the grasp pose detection method [4], [5]. Therefore, a solution that can efficiently learn to grasp unknown objects in remote environments is required.

In this paper, we propose a self-supervised 6-DoF robot grasp pose detection framework via an Augmented Reality (AR) teleoperation system that can efficiently learn to grasp unknown objects through human demonstrations in the AR environment. Specifically, the framework first generates the initial grasp pose to detect the location of the object based on the image segmentation method. Although the initial grasp pose cannot perfectly grasp the object, it still can roughly illustrate the location of the detected object for further adjustment. Then, the AR teleoperation system collects a few human demonstrations through AR software, which enables humans to provide accurate demonstrations remotely. Human demonstration contains waypoints from the start pose to the grasp pose, and the RGB-D image at the start pose. After that, the framework utilizes a point clouds-based contrastive

learning method to learn the hidden morphology representation and evaluate the similarity between the detection object and the object demonstrated by humans. Finally, the framework generates 6-DoF adjustment to improve the initial grasp pose to the 6-DoF grasp pose for unknown objects.

As shown in Fig 1 section c, the framework first detects the initial grasp pose for four objects, which cannot grasp the unknown objects, e.g., plates, small screws, bowls, and small screwdrivers. In Fig 1 section b, The AR interface is used to provide waypoints remotely. After learning several demonstrations, the framework can generate an adjusted 6-DoF grasp pose for unknown objects (Fig 1 section c). Compared with the existing supervised robot grasping works [6], [4], our system can generate the 6-DoF grasp pose and grasp the object without grasp pose supervision because the contrastive learning model learns the hidden morphological representation and can be transferred to adjust the 6-DoF grasp pose based on demonstration. Different from robot imitation work [7], [8], our framework can efficiently use human demonstrations as prompts to generate the 6-DoF grasp pose and learn the waypoints from demonstrations. Additionally, the system is designed as a distributed system to allow remote control by the user and analyze the RGB-D information in different platforms. Our contributions are summarized as follows:

- We propose a novel self-supervised demonstration learning framework to contrastively learn the grasp pose from human demonstration and generate 6-DoF grasp poses.
- We propose a 6-DoF grasp pose detection method that efficiently learns the 6-DoF adjustment solution within several demonstrations (3 times in our experiment).

II. RELATED WORK

a) Vision Based Robot Grasping: The grasp pose detection problem can be formulated as detecting the grasp pose and moving the end effector to the detected grasp pose. Earlier methods consider learning the correlations between multiple representations from detection and the final grasp pose [9]. The input representations include point cloud representations [10], [11], rectangle representations [12], and grasp quality maps representations [13], [14], [15]. However, these methods may ignore the key poses on the target object

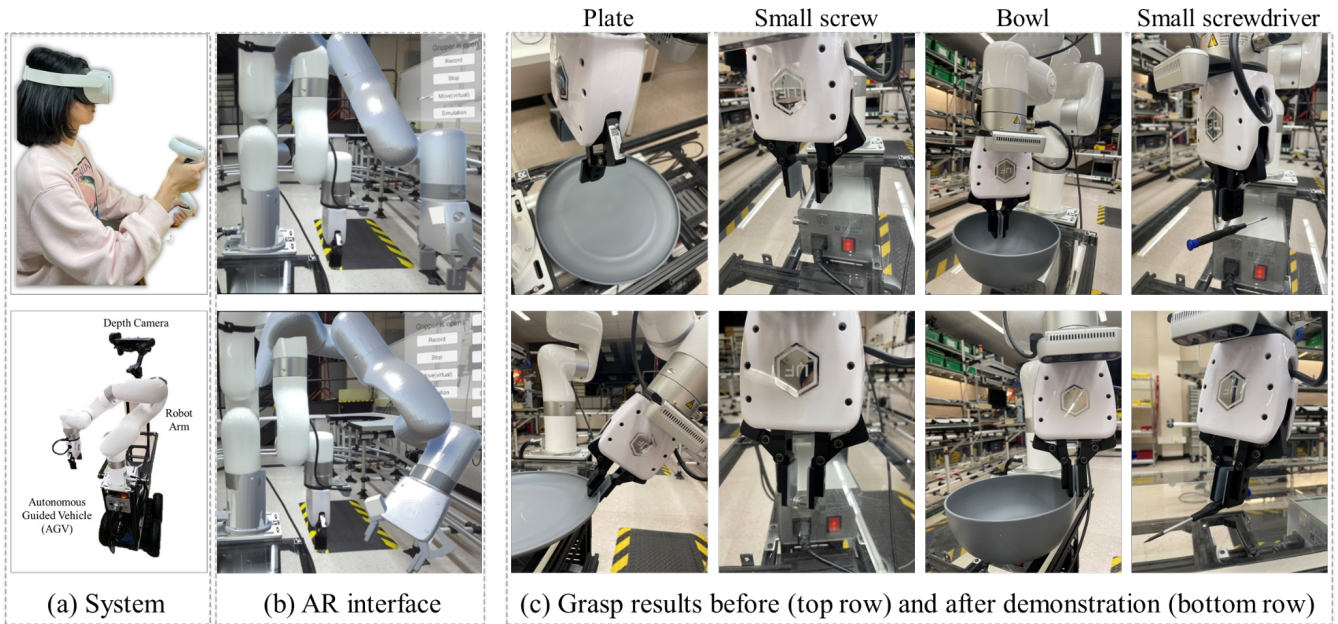


Fig. 1. 6-DoF grasp pose for unknown objects. Red gripper shows the grasp pose. The first row illustrates the generated grasp poses without human demonstration. The second row shows the final grasp pose after learning demonstrations.

and fail, e.g., flat plates. Some approaches consider RGB-D inputs[16], [10] to regress the 6-DoF grasp pose, and other works [17], [4] propose to use the cloud points to generate redundant grasp poses for the objects. Specifically, Fang *et al.* [17], [6] Generate abundant grasp poses based on the large-scale dataset with analytic labels on grasp pose, and Wang *et al.*[4] eliminate the unfeasible grasp poses by proposing graspness score to evaluate the generated grasp pose. Although the grasp pose generating module shows high accuracy in grasping seen and similar objects based on the large-scale dataset, the performance is limited when only a few grasp pose annotations are given. In contrast, our framework proposes to acquire the grasp strategy without grasping annotations and use the morphology representations to learn the human demonstrations selectively. This allows our framework to enhance the grasping ability for unseen objects after learning a few human demonstrations.

b) Human Demonstration Learning via Teleoperation System: The robot teleoperation system aims to enable human operators to manipulate the robot in different working spaces [18], [19], [20], [21], which provides an efficient solution to transfer human instructions to the robot system. Compared with teleoperation solutions in recent works [22], [23], [24], [25], [26], [27], [28], [29], AR-based teleoperation system [30], [31], [32], [33], [34], [35], [36] is notably beneficial because it provides fully immersive or context-rich interactions with the remote environments and enables easy and accurate human demonstrations collecting feature. Specifically, Pan *et al.* [30] proposes to use the RGB-D image and pose teaching device to enhance the interaction between human and robot interaction. Li *et al.* [37] collect human demonstrations and learn robot manipulation. Although these frameworks have better interaction experience, the robot

system can barely learn useful grasp strategies through limited human demonstration. Some imitation learning methods [7], [38] propose to learn the human demonstration through reinforcement learning. These methods need to learn the reward policy of grasping when demonstrations are given. In our works, we propose to learn the point clouds of the objects in a contrastive way. Demonstrations are used as prompts to enable the 6-DoF grasp pose adjustment. This allows the system to efficiently generate 6-DoF grasp poses by comparing the morphology representations.

III. SYSTEM OVERVIEW

The proposed AR teleoperating system contains an immersive AR environment for users to provide demonstrations [39], a grasp pose control module, and a demonstration learning module. It is designed as a distributed system that includes a user demonstration client and a remote robot control server. Fig 2 illustrates the proposed system and the key components in the system. We realize the following features based on the proposed system.

- *Initial grasp pose control* The AR teleoperating system can find the target objects and generate a 4-DoF initial grasp pose based on the segmentation masks and the depth map from the RGB-D camera.
- *Collecting user demonstration in augmented reality* Users can provide grasping instructions by demonstrating the grasping procedure in the AR environment.
- *Demonstration learning* The system records user demonstrations and the corresponding RGB-D camera input as prompts to generate 6-DoF grasp pose and waypoints based on user demonstrations.

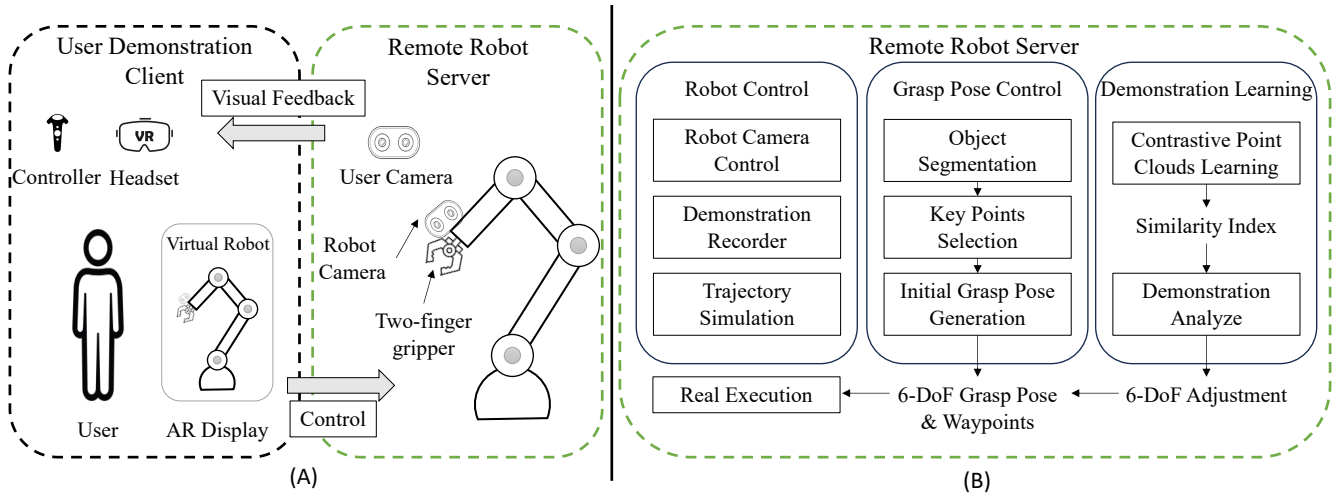


Fig. 2. System Overview. Section A illustrates the system design. The user camera is used to render the remote environment in the AR display. Users may use the AR software to control the remote robot. The robot camera can collect RGB-D images for robot grasping. Section B shows the architecture of the remote robot server, including Robot Control, grasp pose control, and demonstration learning.

A. System Design

In the AR teleoperation system, the remote robot server directly links with two cameras and a robotic arm. The user client connects the VR headset and controllers. As shown in Fig 2, the robot server is designed for collecting visual data, finding the grasp pose, and providing waypoints for grasping. Both cameras are RGB-D cameras, and the robot camera is located on the end-effector of the robotic arm. The visual data from the user camera can be rendered in the AR environment, while visual data from the robot camera are used for generating grasp pose. Users can use the AR software to control the virtual robot to visualize the demonstration and manipulate the real robot to grasp objects. The user demonstration contains object point clouds and the waypoints that move the end effector from the start pose to the grasp pose. If the computing resources of the robot computer are limited and cannot process the complicated segmentation and 6-DoF grasp pose adjustment workload, the grasp pose control module and demonstration learning module can be executed in another powerful remote server. Overall, users can directly use the controller to provide demonstrations in the user client and control the remote robot server.

IV. USER DEMONSTRATION CLIENT

To visualize the user demonstration feature in the immersive AR environment [40], we implement demonstration software based on Unity 3D. Inspired by [41], we integrate the Oculus VR headset and controller in Unity to render the visual data from the user camera and control components in the AR environment. The user camera is physically connected to the robot computer and transfers the visual data to the user computer through the internet to realize remote teleoperation tasks.

In the demonstration software, we import a virtual robot with a virtual two-finger gripper. The virtual robot is cal-

ibrated with the real robot in the world coordinate. There is another virtual gripper in the scene indicating the pose of the end-effector recorded by the software. During the demonstration process, the user needs to use the controller to drag the virtual gripper. The virtual gripper can follow the user controller and move in the AR environment. Here, we provide two gripper following solutions. Users can click the button on the controller to decide pure position following or position and orientation following. Besides the user can also control the gripper status by pressing the button on the controller. There are several virtual buttons in the panel of the AR environment that offer significant features for controlling the virtual robot and the real robot. Specifically, the real robot can be driven by initialize, search, and execute buttons. The back button can cancel the recorded virtual trajectory. Move and simulate all buttons only manipulate the virtual robot based on the recorded trajectory. We connect the software and the robot control system through the ROS-TCP connector [42] library and enable the information transfer feature.

V. REMOTE ROBOT SERVER

The remote robot server is proposed to generate grasp poses and control the robot based on the RGB-D data from the robot camera. It contains three modules (Fig 2 (B)): (1) Robot control module, which enables teleoperation feature on the real robot and the visualization of the robot movement in the AR environment. (2) Grasp pose control module, which generates the grasp pose without training on grasp pose annotation. (3) Demonstration learning module, which learns the correlation between demonstrations and generates the adjustment on trajectory and final grasp pose. We integrate the robot control module with ROS and realize camera control, record demonstrations, and trajectory simulation features.

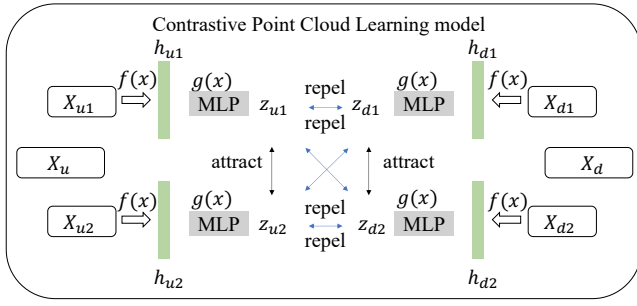


Fig. 3. The contrastive point cloud learning model first augments the input and extracts features through projector $g(f(\cdot))$.

A. Grasp Pose Control Module

The grasp pose control module aims to generate the initial grasp pose without human demonstrations or grasp pose annotations. Given the RGB-D image from the robot camera, the module first segment the RGB image and select the key points from the detected masks. The module then generates 4-DoF grasp poses in the world coordinate with transformation and rotation on the yaw axis.

a) *Object Segmentation*: The object segmentation process aims to segment objects from the RGB frame. In the initial grasp pose generating module, we utilize the FastSAM [43], a lightweight segmentation model that only needs an RGB frame as input. The FastSAM model can generate potential object masks and detection bounding boxes for objects.

b) *Key Points Selection*: The key points selection aims to select the most suitable grasp point based on the segmentation masks and the bounding boxes from the RGB frame. We select five key points for each object and use the pixel position to calculate the confidence of the key points. Five key points contain one point at the center of mass of the segmentation mask, one point located on the mask and has the closest distance from the bounding box center, and the other three points are located on the outline of the segmentation mask.

c) *Initial Grasp Pose Generation*: The initial grasp pose output consists of translation at three axes and the yaw rotation. To calculate the yaw rotation, we use the center of mass point as the center point and calculate the bounding rectangle. The shorter edge of the bounding rectangle is selected to calculate the yaw rotation. After that, the pixel positions of five key points are utilized to calculate the camera position. The pixel position needs to be transformed to the grasp position. We first align the RGB frame and the depth map and get the camera intrinsic to transform pixel position to camera position. Then, we calibrate the camera and the robotic arm gripper and get the eye-in-hand calibration matrix. Finally, we transform the pixel position to the 4-DoF grasp pose.

B. Demonstration Learning Module

The purpose of the demonstration learning module is to define which object should learn the demonstrations

from humans. Since human demonstrations are input in the demonstration learning module as prompts, the system needs to recognize objects that have similar morphology representation with the demonstrated objects and learn the corresponding human demonstration. According to the previous 6-DoF grasp pose generating works [44], [45], [15], [46], [9], the 6-DoF grasp pose is mainly influenced by the point clouds of target object. In the demonstration learning module, we use a self-supervised method to contrastively learn the hidden representation of the object point cloud. The Demonstration learning module contains (1) a contrastive point cloud learning model, and (2) a demonstration analyzing section. The output of the module is the 6-DoF grasp pose adjustment.

a) *Contrastive point cloud learning*: The contrastive point cloud learning model aims to separate the hidden point cloud representation in a self-supervised manner. The output of the model is the similarity index. Inspired by the SimCLR [47] that separates the hidden representation of RGB image, we propose to contrastively learn the hidden representation of point cloud information by maximizing the agreement between augmented point clouds in the latent space. Since we do not have the grasp labels for the training point clouds, each point cloud sample needs to be augmented into two point clouds to enable comparison between the same objects and different objects in the latent space.

Specifically in Fig 3, during the training process, the point cloud sample x_u is defined as the original sample. Firstly, the original sample augments into two samples through jittering, resizing, or flipping, e.g., x_u to x_{u1}, x_{u2} . Three augmented methods are randomly selected to increase the robustness of the model. Secondly, the augmented samples are extracted by encoder $f(\cdot)$ to get the representation vectors h_{u1}, h_{u2} of the augmented samples. In this paper, we adopt the ResNet [48] as encoder and h is the output after the average pooling layer. After that, the representation vector h_{u1}, h_{u2} are projected by MLP $g(\cdot)$ to map the representation to the latent space. The projected representation z_{u1} and z_{u2} is a positive pair, and the task of contrastive loss is to identify x_{u2} in a set of positive pair $\{x_{uk}\}_{k \neq i}$ for x_{u1} . Assuming we have N original sample in a batch, we get $2N$ augmented samples. Given a positive pair in the batch, the other $2N - 1$ samples are negative samples. The contrastive NT-Xtent loss for x_{u1}, x_{u2} can be formulated as follows.

$$\mathcal{L}_{i,j} = -\log \frac{\exp(\text{sim}(z_{u1}, z_{u2})/\tau)}{\sum_{k=1}^{2N} \mathbf{1}_{[k \neq i]} \exp(\text{sim}(z_{u1}, z_{uk})/\tau)}, \quad (1)$$

where $\text{sim}(z_{u1}, z_{u2})$ is the cosine similarity and τ is the temperature parameter. $\mathbf{1}_{[k \neq i]}$ is an indicator function equal to 1 only if $k \neq i$. In this paper, we use the ShapeNet [49] point cloud dataset to pretrain the model and update the non-linear projector $g(\cdot)$.

b) *Demonstration Analyzing Section*: Before the demonstration starts, the system will record the RGB frame and the depth map of the experiment surroundings. For the detected objects, the system first transforms the detected

masks into point clouds based on the camera coordinate and normalizes the point clouds. Then, the system can generate the initial grasp pose for detected objects. When the demonstration starts, the AR software can record a set of waypoints from the start pose to the grasp pose. To avoid the display error in the AR software and get the accurate demonstrated grasp pose, we fine-tune the position of the grasp pose. Each demonstrated sample contains the point clouds and waypoints.

During the experiment, some objects may be demonstrated multiple times. If the demonstrated samples are recording point clouds for the same object, they belong to the same category. Assuming the software records waypoints from the human demonstration (start pose to grasp pose), the system needs to output the same number of key points from the start pose to the grasp pose for target objects. There are two scenarios. If the system can generate the initial grasp pose for the demonstrated object, the system can adjust both position and orientation based on the demonstration. If the system cannot generate the initial grasp pose, we only learn the orientation adjustment. To ensure the number of key points in the initial grasp trajectory is equal to the number of waypoints in the human demonstration, we separate the initial trajectory evenly into M points. For each segmentation, the system can prepare point clouds for segmentation masks. The prepared point clouds first compare all recorded demonstration point clouds and get the similarity between all demonstrated categories. If there are multiple demonstrated samples for one category, we use the demonstration information with maximum similarity as the similarity index i_s for the category. For each demonstration, we first analyze the recorded waypoints. After finding the center point of the point clouds, we generate four pseudo grasp poses around the outline of the object based on the demonstrated grasp pose. The pseudo grasp poses have the same distance from the center point of the object. The distance from the center point to the pseudo grasp pose can be resized based on the scaling factor between the demonstrated object and the detected object. There are two thresholds indicating learning rotation t_r and learning translation t_l . If $i_s > t_r$, the system utilizes the generated pseudo grasp pose as the final grasp pose. If $i_s > t_l$, the system will use the demonstrated grasp pose provided by human.

VI. SYSTEM EVALUATION

A. Experiment Details

We use an xArm 6 robotic arm with its two-finger gripper to conduct the experiment. In the system, the robot is directly linked with an RTX3060 6GB laptop, and we install a Realsense d435i at the end effector of the robot arm as the robot camera. The frequency of the robot camera is 7 Hz. The ZED 2 camera is connected to the robot computer and sends the vision data back to the user computer with a frequency of 30Hz. For the normal grasping experiment, we only run the grasp pose control module on the robot computer. For the demonstration learning experiment, we run both grasps pose control and demonstration learning module



Fig. 4. Example objects in hardware, household, food, and toys.

in the same computer with a maximum batch size equal to 4. In the proposed demonstration learning module, the temperature parameter is 0.1.

B. Evaluation Metric

Similar to previous literature [6], [50], we adopt two success rates to evaluate the performance. The attempt-centric success rate is defined as the fraction of the number of successful grasp attempts over the number of total grasp attempts. This metric measures the ability of the grasp solution in a single attempt. The object-centric success rate is the ratio of the number of successfully grasped objects over the total number of objects and each object allows two attempts. The attempt-centric rate is more strict than the object-centric success rate.

C. Normal Grasping

In the normal grasping experiment, we set a workspace limit on a plane for safety. The detection pose is set to 0.45 meters above the plane. During the experiment, we randomly place the objects on the plane at a random orientation and take one picture. After that, the initial grasp pose generating module will generate initial grasp poses for all detected objects in the scene. The segmentation model only uses the RGB input to generate key points for objects, and the grasp pose control module can select key points and predict the grasp pose for the scene according to the segmentation results and the depth map. The initial grasp pose has the vertical direction and adjusted rotation on the yaw. The pre-grasp pose is 0.15 meters backward in the approaching direction. Additionally, we collect over 200 objects from 78 categories. Similar to objects in any grasp [6] and YCB dataset [51]. The size of the objects ranges from $1.5 \times 1.5 \times 1.5 \text{cm}^3$ to $19.5 \times 14.5 \times 25 \text{cm}^3$. Fig 4 shows part of unknown grasping objects. These objects can be roughly divided into hardware, fruit, food, toy, and household.

For each detection trial, the grasp pose control module segments the objects and provides an initial grasp pose in the scene within 80ms. After learning the human demonstration, the mean inference speed for each trial is less than 1s.

TABLE I

SUCCESS RATES OF DIFFERENT METHODS ON REAL NORMAL GRASP. "DEX." REPRESENTS DEXNET 4.0. "ANY." REPRESENTS ANY GRASP. "DEMO." MEANS THE RESULT AFTER LEARNING DEMONSTRATIONS.

Object	Attempt-Centric Success Rate (%)				Object-Centric Success Rate (%)			
	Dex	Ours	Demo	Any	Dex	Ours	Demo	Any
Hardware	59.4	78.0	89.3	81.2	97.2	97.7	100.0	100.0
Ragdoll	87.6	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Toy	72.8	82.0	84.3	92.9	93.7	100.0	100.0	100.0
Food&Snack	52.0	85.8	86.3	98.3	93.9	95.4	95.4	100.0
Household	63.5	59.5	65.0	84.5	98.9	95.0	100.0	100.0
All	72.2	78.2	81.9	93.3	97.4	98.2	98.7	100.0

The average robot trajectory planning time is 2.9s. The mean picks per hour (MPPH) mainly depends on the robot's moving speed and the gripper's executing speed. In the normal grasp experiment, the average pick speed is over 300 MPPH. The experiment results in Table I show the superior grasping ability in some objects, especially for ragdolls and foods. Specifically, we achieved 100% attempt-centric success rate on ragdolls. The attempt-centric successful rates on hardware (78%), toy (82%), and food (85.8%) objects are also better than the supervised grasp solution Dexnet 4.0 [52]. The performance on the household is not superior to the AnyGrasp [6] because the segmentation model does not use the grasp pose dataset and the RGB-D images provided in the Graspnet [17]. Before demonstration information is applied, our self-supervised grasp pose-generating method lacks key information in grasping unseen objects.

D. Grasping After Demonstration

In this section, we choose the objects with low attempt-centric success rates in normal grasp experiments and demonstrate the grasping procedure through the AR-teleoperation system. To avoid the display error in the AR teleoperation system, we fine-tune the position of the last waypoints in the demonstration. The adjustment is based on the center point of the object. The number of waypoints in a single demonstration is less than 10 during the demonstration. The rotation threshold $t_r = 0.7$ and the translation threshold $t_l = 0.9$. During the experiments, we first use the RGB-D input from the robot camera to segment objects in the scene and extract the point cloud feature. After that, the point cloud features are compared with the demonstrated point cloud features and generate the grasp pose. The contrastive point cloud learning model is pre-trained in the Shapenet [49] dataset. For each demonstrated object, we provide three demonstrations. Table II illustrates the attempt-centric successful rate for grasping results after learning the demonstration. The overall attempt-centric success rate increases from 78.3% to 81.2% after learning three demonstrations. In normal grasp experiment, it is difficult for the self-supervised model to grasp flat objects like plates and some small objects like screws and small screwdrivers. The framework learns the 6-DoF grasp pose from the demonstrated point clouds and waypoints. From the results, we also find that the grasping

TABLE II

ATTEMPT-CENTRIC SUCCESSFUL RATE FOR 6-DOF GRASP AFTER LEARNING DIFFERENT TIMES OF HUMAN DEMONSTRATION. "NO DEMO" REPRESENTS NO HUMAN DEMONSTRATION.

Objects	Attempt-Centric Success Rate(%)			
	No Demo	1 Demo	2 Demos	3 Demos
Plates	0.0	70.0	90.0	90.0
Bowls	20.0	70.0	90.0	90.0
Mustard source	50.0	40.0	50.0	60.0
Small screws	20.0	90.0	90.0	90.0
Small screwdrivers	50.0	90.0	90.0	90.0
All	78.2	80.6	81.2	81.9

TABLE III

ABLATION STUDY ON CONTRASTIVE POINT CLOUD LEARNING MODEL

Method	RGB	Point Clouds	Attempt-Centric Success Rate (%)
No Demo			78.2
Case1		✓	81.9
Case2	✓		79.6
Case3	✓	✓	80.6

ability is enhanced when more demonstrations are provided. The maximum robot planning time is 0.5 seconds. The latency of teleoperation in robot manipulation is lower than 0.5 seconds.

E. Ablation Study

In this section, we report the ablation study about the demonstration learning. Table III illustrates the performance improvement after using point clouds contrastive learning (Case1), RGB image contrastive learning (Case2), and the RGB-D contrastive learning model (Case3). For comparison, we report the grasp performance without learning demonstrations. The RGB images come from the ImageNet [53] dataset with the same category label. During training, we normalize the point clouds and use the same data augmentation methods. We find that solely point clouds learning model (Case1) provides the most useful information and assists the system in generating 6-DoF grasp pose. Although RGB image contains morphological information, the point clouds may easily adapt the information to the adjustment of 6-DoF grasp pose.

VII. CONCLUSIONS

This paper proposes a self-supervised 6-DoF robot grasping framework that efficiently learns human demonstration via an AR teleoperation system. In our framework, hidden morphology representation captured from point clouds can be efficiently learned by the contrastive learning model, which allows the robot to learn from human demonstrations. We apply the framework to the AR system to facilitate the visualization of robot grasping, demonstration recording, and demonstration learning in robot manipulation. In future work, we are interested in extending the grasping ability of our system in a more dynamic environment.

REFERENCES

- [1] T. Moore, "Robots for nuclear power plants," *IAEA Bulletin*, vol. 27, no. 3, pp. 31–38, 1985.
- [2] T. Wang, J. Gao, and O. Xie, "Sliding mode disturbance observer and q learning-based bilateral control for underwater teleoperation systems," *Applied Soft Computing*, vol. 130, p. 109684, 2022.
- [3] Z. Wang, H.-K. Lam, B. Xiao, Z. Chen, B. Liang, and T. Zhang, "Event-triggered prescribed-time fuzzy control for space teleoperation systems subject to multiple constraints and uncertainties," *IEEE Transactions on Fuzzy Systems*, vol. 29, no. 9, pp. 2785–2797, 2020.
- [4] C. Wang, H.-S. Fang, M. Gou, H. Fang, J. Gao, and C. Lu, "Grasp-net discovery in clutters for fast and accurate grasp detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15964–15973.
- [5] X. Dengxiong and Y. Kong, "Ancestor search: Generalized open set recognition via hyperbolic side information learning," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, January 2023, pp. 4003–4012.
- [6] H.-S. Fang, C. Wang, H. Fang, M. Gou, J. Liu, H. Yan, W. Liu, Y. Xie, and C. Lu, "Anygrasp: Robust and efficient grasp perception in spatial and temporal domains," *IEEE Transactions on Robotics*, 2023.
- [7] P. Sermanet, K. Xu, and S. Levine, "Unsupervised perceptual rewards for imitation learning," *arXiv preprint arXiv:1612.06699*, 2016.
- [8] T. Asfour, P. Azad, F. Gyarfas, and R. Dillmann, "Imitation learning of dual-arm manipulation tasks in humanoid robots," *International journal of humanoid robotics*, vol. 5, no. 02, pp. 183–202, 2008.
- [9] B. Wen, W. Lian, K. Bekris, and S. Schaal, "Catgrasp: Learning category-level task-relevant grasping in clutter from simulation," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 6401–6408.
- [10] P. Ni, W. Zhang, X. Zhu, and Q. Cao, "Pointnet++ grasping: Learning an end-to-end spatial grasp generation algorithm from sparse point clouds," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 3619–3625.
- [11] Y. Li, T. Kong, R. Chu, Y. Li, P. Wang, and L. Li, "Simultaneous semantic and collision learning for 6-dof grasp pose estimation," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 3571–3578.
- [12] Y. Jiang, S. Moseson, and A. Saxena, "Efficient grasping from rgb-d images: Learning using a new rectangle representation," in *2011 IEEE International conference on robotics and automation*. IEEE, 2011, pp. 3304–3311.
- [13] C. Choi, W. Schwarting, J. DelPreto, and D. Rus, "Learning object grasping for soft robot hands," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2370–2377, 2018.
- [14] C.-H. Wang and P.-C. Lin, "Q-pointnet: Intelligent stacked-objects grasping using a rgb-d sensor and a dexterous hand," in *2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE, 2020, pp. 601–606.
- [15] C. Wu, J. Chen, Q. Cao, J. Zhang, Y. Tai, L. Sun, and K. Jia, "Grasp proposal networks: An end-to-end solution for visual learning of robotic grasps," *Advances in Neural Information Processing Systems*, vol. 33, pp. 13 174–13 184, 2020.
- [16] M. Gou, H.-S. Fang, Z. Zhu, S. Xu, C. Wang, and C. Lu, "Rgb matters: Learning 7-dof grasp poses on monocular rgb-d images," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 13 459–13 466.
- [17] H.-S. Fang, C. Wang, M. Gou, and C. Lu, "Graspnet-1billion: A large-scale benchmark for general object grasping," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 444–11 453.
- [18] J. Guo, C. Liu, and P. Poignet, "A scaled bilateral teleoperation system for robotic-assisted surgery with time delay," *Journal of Intelligent & Robotic Systems*, vol. 95, pp. 165–192, 2019.
- [19] J. Liang, B. Wen, K. Bekris, and A. Boularias, "Learning sensorimotor primitives of sequential manipulation tasks from visual demonstrations," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 8591–8597.
- [20] B. Wen, W. Lian, K. Bekris, and S. Schaal, "You only demonstrate once: Category-level manipulation from single visual demonstration," *Proceedings of Robotics: Science and Systems*, 2022.
- [21] J. Liang, A. Boularias, A. Dollar, K. Bekris *et al.*, "Vision-driven compliant manipulation for reliable; high-precision assembly tasks," in *Proceedings of Robotics: Science and Systems*, 2021.
- [22] S. Baklouti, G. Gallot, J. Viaud, and K. Subrin, "On the improvement of ros-based control for teleoperated yaskawa robots," *Applied Sciences*, vol. 11, no. 16, p. 7190, 2021.
- [23] S. N. Young and J. M. Peschel, "Review of human-machine interfaces for small unmanned systems with robotic manipulators," *IEEE Transactions on Human-Machine Systems*, vol. 50, no. 2, pp. 131–143, 2020.
- [24] P. Gliesche, T. Krick, M. Pflingsthorst, S. Drolshagen, C. Kowalski, and A. Hein, "Kinesthetic device vs. keyboard/mouse: a comparison in home care telemanipulation," *Frontiers in Robotics and AI*, vol. 7, p. 561015, 2020.
- [25] J. P. Clark, G. Lentini, F. Barontini, M. G. Catalano, M. Bianchi, and M. K. O'Malley, "On the role of wearable haptics for force feedback in teleimpedance control for dual-arm robotic teleoperation," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 5187–5193.
- [26] M. Aggravi, D. A. Estima, A. Krupa, S. Misra, and C. Pacchierotti, "Haptic teleoperation of flexible needles combining 3d ultrasound guidance and needle tip force feedback," *IEEE Robotics and automation letters*, vol. 6, no. 3, pp. 4859–4866, 2021.
- [27] C. Mizera, T. Delrieu, V. Weistroffer, C. Andriot, A. Decatoire, and J.-P. Gazeau, "Evaluation of hand-tracking systems in teleoperation and virtual dexterous manipulation," *IEEE Sensors Journal*, vol. 20, no. 3, pp. 1642–1655, 2019.
- [28] A. Handa, K. Van Wyk, W. Yang, J. Liang, Y.-W. Chao, Q. Wan, S. Birchfield, N. Ratliff, and D. Fox, "Dexpilot: Vision-based teleoperation of dexterous robotic hand-arm system," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 9164–9170.
- [29] A. Sivakumar, K. Shaw, and D. Pathak, "Robotic telekinesis: Learning a robotic hand imitator by watching humans on youtube," *arXiv preprint arXiv:2202.10448*, 2022.
- [30] Y. Pan, C. Chen, D. Li, Z. Zhao, and J. Hong, "Augmented reality-based robot teleoperation system using rgb-d imaging and attitude teaching device," *Robotics and Computer-Integrated Manufacturing*, vol. 71, p. 102167, 2021.
- [31] Y. Qin, W. Yang, B. Huang, K. Van Wyk, H. Su, X. Wang, Y.-W. Chao, and D. Fox, "Anyteleop: A general vision-based dexterous robot arm-hand teleoperation system," in *Robotics: Science and Systems*, 2023.
- [32] S. P. Arunachalam, I. Güzey, S. Chintala, and L. Pinto, "Holo-dex: Teaching dexterity with immersive mixed reality," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5962–5969.
- [33] C. Wang and A. Belardinelli, "Investigating explainable human-robot interaction with augmented reality," in *5th International Workshop on Virtual, Augmented, and Mixed Reality for HRI*.
- [34] F. Xiang, Y. Qin, K. Mo, Y. Xia, H. Zhu, F. Liu, M. Liu, H. Jiang, Y. Yuan, H. Wang *et al.*, "Sapien: A simulated part-based interactive environment," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 097–11 107.
- [35] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2012, pp. 5026–5033.
- [36] C. Gan, J. Schwartz, S. Alter, D. Mrowca, M. Schrimpf, J. Traer, J. De Freitas, J. Kubilius, A. Bhandwajdar, N. Haber *et al.*, "Threedworld: A platform for interactive multi-modal physical simulation," *arXiv preprint arXiv:2007.04954*, 2020.
- [37] C. Li, P. Zheng, S. Li, Y. Pang, and C. K. Lee, "Ar-assisted digital twin-enabled robot collaborative manufacturing system with human-in-the-loop," *Robotics and Computer-Integrated Manufacturing*, vol. 76, p. 102321, 2022.
- [38] S. Calinon and A. Billard, "Incremental learning of gestures by imitation in a humanoid robot," in *Proceedings of the ACM/IEEE international conference on Human-robot interaction*, 2007, pp. 255–262.
- [39] S. Qiu, H. Liu, Z. Zhang, Y. Zhu, and S.-C. Zhu, "Human-robot interaction in a shared augmented reality workspace," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 11 413–11 418.
- [40] L. Cao, H. Zhang, C. Peng, and J. T. Hansberger, "Real-time multi-modal interaction in virtual reality—a case study with a large virtual interface," *Multimedia Tools and Applications*, pp. 1–22, 2023.
- [41] H. Zhang, L. Cao, G. Howell, D. Schwartz, and C. Peng, "An educational virtual reality game for learning historical events," *Virtual Reality*, vol. 27, no. 4, pp. 2895–2909, 2023.

- [42] U. Technologies, “Ros tcp connector,” 2022, accessed: september 1st, 2023. [Online]. Available: <https://github.com/Unity-Technologies/ROS-TCP-Connector>
- [43] X. Zhao, W. Ding, Y. An, Y. Du, T. Yu, M. Li, M. Tang, and J. Wang, “Fast segment anything,” 2023.
- [44] R. Newbury, M. Gu, L. Chumbley, A. Mousavian, C. Eppner, J. Leitner, J. Bohg, A. Morales, T. Asfour, D. Kragic *et al.*, “Deep learning approaches to grasp synthesis: A review,” *IEEE Transactions on Robotics*, 2023.
- [45] M. Breyer, J. J. Chung, L. Ott, R. Siegwart, and J. Nieto, “Volumetric grasping network: Real-time 6 dof grasp detection in clutter,” in *Conference on Robot Learning*. PMLR, 2021, pp. 1602–1611.
- [46] B. Zhao, H. Zhang, X. Lan, H. Wang, Z. Tian, and N. Zheng, “Regnet: Region-based grasp network for end-to-end grasp detection in point clouds,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 13 474–13 480.
- [47] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.
- [48] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [49] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su *et al.*, “Shapenet: An information-rich 3d model repository,” *arXiv preprint arXiv:1512.03012*, 2015.
- [50] Y. Qin, R. Chen, H. Zhu, M. Song, J. Xu, and H. Su, “S4g: Amodal single-view single-shot se (3) grasp detection in cluttered scenes,” in *Conference on robot learning*. PMLR, 2020, pp. 53–65.
- [51] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar, “Benchmarking in manipulation research: The ycb object and model set and benchmarking protocols,” *arXiv preprint arXiv:1502.03143*, 2015.
- [52] J. Mahler, M. Matl, V. Satish, M. Danielczuk, B. DeRose, S. McKinley, and K. Goldberg, “Learning ambidextrous robot grasping policies,” *Science Robotics*, vol. 4, no. 26, p. eaau4984, 2019.
- [53] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.