2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Learning a Pre-Grasp Manipulation Policy to Effectively Retrieve a Target in Dense Clutter

Marios Kiatos[1], Leonidas Koutras[1], Iason Sarantopoulos[2] and Zoe Doulgeri[1]

***Abstract*— Robotic grasping of a target object in cluttered environments poses considerable challenges, often due to limited collision-free grasp affordances caused by the close proximity of other objects. To overcome this limitation, non-prehensile actions like pushing can be strategically employed to manipulate the environment and improve the chances of successful grasps. In this paper, we introduce a novel pre-grasp manipulation policy designed to efficiently retrieve a target object from dense clutter by leveraging pushing actions and considering the gripper's kinematic capabilities to strategically position the target object within the gripper's closing region for a secure grasp. Unlike conventional approaches, our policy incorporates sequential pushing, allowing the robot to make decisions while within the camera's field of view without retracting to a home position, leading to significantly reduced execution time per action. Our policy, trained in simulation, seamlessly transfers to real-world scenarios. Extensive experimental evaluation demonstrates superior performance, faster completion times, and robust generalization to unseen objects compared to existing baselines.**

Fig. 1. The proposed policy rearranges clutter around the blue bottle before grasping it.

I. Introduction

Effective robotic grasping in realistic unstructured environments remains a significant challenge, particularly in settings commonly inhabited by humans such as homes and workplaces. These environments are often characterized by high levels of clutter, which can prohibit the robotic fingers from reaching and grasping target objects. Dealing with the clutter means that the robot should rearrange the obstacles around a target object, in order to create enough space for the robotic fingers to perform a prehensile grasp. Consequently, the development of pre-grasp manipulation policies that enable the creation of grasping affordances holds immense value, as it can significantly enhance grasp success rates in complex and cluttered environments. By enabling robots to effectively navigate and manipulate clutter, they can perform tasks more autonomously and seamlessly integrate into human-centric environments, thereby unlocking a wide range of practical applications in areas such as household assistance, manufacturing, and healthcare.

This work has received funding from the European Union's Horizon Framework Programme for Research and Innovation under grant agreement no 101120823, project MANiBOT.

Marios Kiatos was funded by the Excellence Scholarship he received from the Research Committee AUTH – Special Account for Research Funds AUTH.

[1]M. Kiatos, L. Koutras, and Z. Doulgeri are with the Department of Electrical and Computer Engineering, Aristotle University of Thessaloniki, Thessaloniki, 54124 Greece (email: mkiatos@ece.auth.gr; kleonidas@ece.auth.gr; doulgeri@ece.auth.gr).

[2]Iason Sarantopoulos is with Microsoft Research Cambridge, UK (email: iason.sarantopoulos@microsoft.com). This work was conducted while the author was affiliated with Aristotle University of Thessaloniki.

The majority of research efforts [1], [2] have been focused on finding synergies between pushing and grasping policies particularly by combining pushing actions with top-down precision grasps. While effective for certain tasks such as pick-and-place operations common in industrial settings, these strategies may be insufficient for more complex tasks. For instance, tasks like grasping a bottle and pouring water require stable side grasps rather than top-down grasps. Thus, some approaches [3], [4] have employed side grasps by leveraging pushing actions to roll objects into the closing region of the gripper, thereby enhancing contact area and stability. However, these methods result in substantial execution times, as the robot frequently returns to a home position for new image capture to determine the next action. While some approaches [5] attempted to close this loop to reduce execution time, they typically require complete knowledge of objects and the environment to plan the next optimal action for grasping the target object.

In this work, we propose and experimentally evaluate a pre-grasp manipulation policy for rearranging the clutter around a target object (Fig. 1) to facilitate its side grasping. The proposed policy employs sequential pushing actions, allowing the robot to make decisions within the field of view of the camera without moving to a home position. In summary, this paper introduces the following contributions:

- A novel sequential pushing policy that positions the target object into the gripper's closing region by exploiting the kinematics of the gripper.

- Reduction of the task completion time through the integration of the robot into the state representation, eliminating the need for repositioning the robot to a home position.
- Experimental evaluation that demonstrates superior performance of the proposed policy over baselines and generalization to unseen objects.

II. Related Work

Robotic grasping has evolved a lot over the past two decades, with Bohg *et al.* [6] dividing the various methods of robotic grasping to analytic and data-driven. Analytic methods require full knowledge of object geometry and physics for grasp calculations [7] and tend to not transfer well to the real world due to the inability to model physical parameters related to the interaction between a manipulator and an object [8]. This led to the rising popularity of data-driven approaches [9], [10] that leverage large datasets to predict successful grasps, but the majority of them consider the robotic grasping of a single isolated object in an uncluttered environment. While certain methods [11]–[13] have shown promising grasp success rates in cluttered scenes, they often yield low-quality or non-existent collision-free grasps within densely cluttered environments.

Pushing actions are commonly employed to address challenges posed by dense clutter, particularly in singulating target objects from surrounding obstacles [14]–[16]. Recent research efforts [2], [17] have delved into exploring synergies between pushing and grasping actions, aiming to enhance the grasping success of target objects. While these approaches showcase promising results, they primarily demonstrate their efficacy on simple toy blocks. To address this limitation, recent advancements [18]–[20] have been made towards improving learning efficiency and achieving high grasp success rates on objects with more complex geometries. However, a common drawback among these methods is their reliance on top-down precision grasps and the necessity of moving the robot to a home position for capturing new images of the scene to determine the next action.

To facilitate stable side grasps in dense clutter, Dogar et al. [3] utilize pushing as a preceding realignment step, but their method relies on comprehensive knowledge of object properties and poses for successful grasping. Benjjani et al. [5] proposed a physics-based planner for side grasping a target object in cluttered and confined workspaces, yet they also require precise poses and geometries of each object in the scene. To mitigate the need for full knowledge of objects and the environment, data-driven approaches have been proposed. Sidiropoulos and Doulgeri [21] trained a deep neural network to predict pushing action trajectories for positioning the target object within the gripper's closing region. However, their approach relies on expert human demonstrations for training data and lacks closed-loop planning, as the entire pushing trajectory is inferred only from the initial state. This one-step planning approach results in low success rates, primarily because it does not allow for necessary corrections and adjustments during the execution of the task. In our previous work [4], we proposed a push-grasping policy for side grasps but the policy's objective of grasping every object in the scene and the need to move the robot arm outside of the workspace for capturing new images result in high execution times for target-oriented tasks. In contrast to these works, our proposed policy generates actions in a closed-loop planning fashion without the need to move the robot arm outside the field of view of the camera between decisions, leveraging a novel sequential pushing strategy.

III. Problem Formulation

We address the challenge of grasping a target object that is surrounded by other objects on a flat surface. To facilitate its grasping, we utilize consecutive linear pushing actions to create free space around the target object without moving the robot to a home position. The robot rearranges the clutter through non-prehensile manipulation to position the target object within the closing region of the gripper. The closing region of the gripper refers to the workspace between the fingers, where if an object is present, closing the fingers will result in a stable grasp with high probability (Fig. 1). This two-step approach decomposes the grasping process into two problems: first, rearranging the clutter to position the target within the closing region, and second, closing the fingers to perform a prehensile grasp, ensuring a secure hold on the object.

We formulate the problem of rearranging the clutter to position the target object in the closing region of the gripper as an episodic Markov Decision Process (MDP) with time horizon T. The agent (e.g. robot) chooses and executes an action a_t according to a policy $\pi(\boldsymbol{s}_t)$, then transitions to a new state $\boldsymbol{s}_{t+1}$ and receives an immediate reward $r(\boldsymbol{s}_t, a_t)$. The objective is to properly rearrange the clutter to position the target object inside the robot hand's closing region to achieve a stable power grasp.

A. Environment

We assume that there is a planar squared surface on which random objects with arbitrary geometry are resting in a tightly packed configuration. A tightly packed configuration implies the existence of contact points between neighboring objects. The inertial frame is placed on the center of the surface, with its $\boldsymbol{z}$-axis, normal to the surface and opposite to the gravity vector.

B. State Representations

Each state $\boldsymbol{s}_t$ is represented as a depth d_t and segmentation heightmap m_t of the scene captured by an RGB-D image from a fixed-mounted camera. Initially, the RGB image is passed to a semantic segmentation module to discern the target object and obstacles. To incorporate the robot into the segmentation, we render its 3D model within the scene [22] , given its joint positions and forward kinematics. Subsequently, we fuse these segmented images, assigning pixel values: 1 for obstacle pixels, 2 for target pixels, 3 for robot pixels, and 0 for the support surface (Fig. 3). This fusion process enables a comprehensive representation

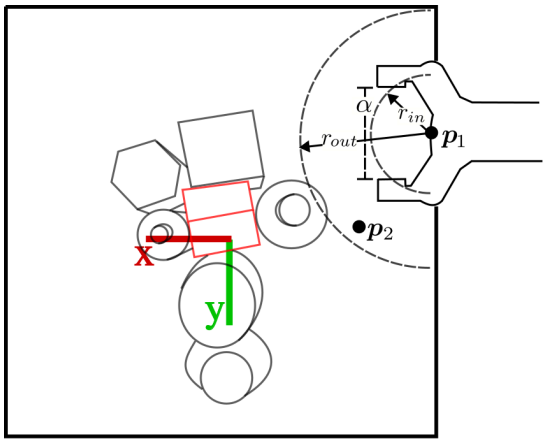


Fig. 2. Parametrization of the pushing primitive. The target object is reached through consecutive linear pushing actions executed parallel to the support surface from a predetermined height z. Each pushing action involves moving the hand to a final position $\boldsymbol{p}_2$ and adjusting the hand aperture α accordingly. The world frame is placed on the center of the workspace.

of the scene, essential for effective decision-making in the rearrangement task.

Then, each image is projected onto a 3-D point cloud and orthographically back-projected upward in the gravity direction to construct the depth and segmentation heightmaps. The segmentation heightmap m_t is normalized to a range of 0-1, while the depth heightmap d_t is normalized by subtracting the mean and dividing by the standard deviation. This scalar height-from-bottom information enables the deep models to learn features that are rich in geometric shape. Similar to [4], the edges of the heightmaps are defined with respect to the workspace limits. In our experiments, this area covers 0.5×0.5 m tabletop surface. The heightmaps have image resolution 100×100 and hence each pixel represents a 5×5 mm vertical column of 3-D space in the robot's workspace.

C. Actions

The pushing primitive involves a straight motion of the robot hand along a line segment $[\boldsymbol{p}_1, \boldsymbol{p}_2] \in \mathbb{R}^3$, positioned above and parallel to the support surface. Previous approaches [2], [4] assume that the robot's initial position is outside of the camera's field of view in a predetermined home position and learn both points for determining the line segment of the pushing action. To the contrary, we specify the initial position $\boldsymbol{p}_1$ directly from the state representation, which corresponds to the robot's last position from the preceding action. Consequently, our approach learns the final position $\boldsymbol{p}_2$ for defining the line segment of the pushing action, eliminating the need to move the robot to a home position between actions. Additionally, the agent can specify the opening of the robot hand before the push begins. Thus, we parameterize the pushing primitive as $\boldsymbol{a} = \{\boldsymbol{p}, \alpha\}$, where:

- $\boldsymbol{p} \in \mathbb{R}^2$ denotes the final position at which the push ends, with the distance z from the table kept fixed during execution (noting that $\boldsymbol{p}_2 = [\boldsymbol{p}^T, z]^T$). It's noteworthy that the height z is not a learnable parameter and is empirically set to $z = 0.1$ m. This value is determined as the minimum distance from the table that the robot hand can move freely without collision with the support surface during the execution of the motion primitive.
- α represents the aperture of the hand during the push, maintaining a constant finger distance in an opposable finger configuration throughout the primitive's execution. In our approach, the aperture options are binary, meaning that the robot hand can either be fully open or fully closed. When the hand is fully open, the fingers are spread apart to their maximum extent, when the hand is fully closed, the fingers are brought together.

To avoid large pushing actions which may result to imprecise pushes, we constrain the action space within a half-ring region surrounding the gripper's grasp position Fig. 2, defined by an outer radius of r_{out} and an inner radius of r_{in}. This means that the final position $\boldsymbol{p}_2$ is defined as:

$$\boldsymbol{p}_2 = \boldsymbol{p}_1 + \begin{bmatrix} r\cos\theta \\ r\sin\theta \\ 0 \end{bmatrix}, \theta \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right], r \in [r_{in}, r_{out}] \quad (1)$$

This constrain also prevents unnecessary backward movements which means that it decreases execution time during deployment.

D. Rewards and Terminal States

We define a sparse reward function as follows: If the target object is within the closing region of the gripper, a reward of $+2$ is assigned. If the gripper moves closer to the target object, a reward of $+0.5$ is assigned. Otherwise, the reward is 0. To check if the target is inside the closing region, we compute if the intersection-over-union (IoU) between the target mask and the closing region area is greater than 0.95.

The episode is terminated under the following conditions: if the target object or an obstacle enters the closing region, if an object falls from the support surface, and if the robot is unable to reach the final position $\boldsymbol{p}_2$. The latter condition implies that the agent learns implicitly to avoid positions that are either unreachable due to singularities or robot joint limits, or could potentially cause a collision between the body of the arm and the support surface.

E. Reinforcement Learning using Fully-Convolutional Neural Networks

Inspired by [1], we model the Q-function as a fully-convolutional neural network (FCN). As shown in Fig. 3 the network takes as input the normalized depth and segmentation heightmaps and outputs two dense pixel-wise maps of Q values with the same image size and resolution as that of the inputs. Each map corresponds to a distinct aperture configuration, where each pixel $q_i \in Q$ denotes the anticipated future reward of executing a pushing action terminating at pixel $\boldsymbol{p}$, associated with pixel i and aperture $j \in \{0, 1\}$. Employing a variant of Q-learning, we discretize the action space. This enables the robot to navigate within a 100×100 grid derived from the state representation, accommodating

two distinct aperture configurations (open/close). Thus, the action space encompasses $100 \times 100 \times 2$ possibilities.

The FCN is trained to minimize the temporal difference error δ_t of $Q(s_t, a_t)$ to a fixed target value y_t:

$$\delta_t = |Q(s_t, a_t|\theta_t) - y_t| \tag{2}$$

$$y_t = r(s_t, a_t, s_{t+1}) + \gamma \max_{a'} Q(s_{t+1}, a'|\theta_t^-)) \tag{3}$$

via Huber loss:

$$L = \begin{cases} \frac{1}{2}\delta_t, \text{if } |\delta_t| \leq 0.1 \\ |\delta_t| - \frac{1}{2}, \text{otherwise} \end{cases} \tag{4}$$

where θ_t, θ_t^- are the parameters of the FCN and the corresponding target network at time t respectively. The target network is updated with a polyak averaging of 0.999. We pass gradients only through a single pixel i and thus all other pixels backpropagate zero loss. We train the network using the Adam optimizer with learning rate 10^{-4} and with a discount factor of $\gamma = 0.5$. The model is trained in PyTorch with an Nvidia GeForce RTX 4070. Furthermore, we use prioritized experience replay [23] to balance the data that update the network. To enforce the action space constraint of Eq. (1) we filter the output by masking the output maps and keep only the Q-values corresponding to the half-ring area.

The constraint specified in Eq. (1) guides the exploration process to deter unnecessary backward motions, thereby enhancing the efficiency of the training process. This constraint restricts the action space to a half-ring region around the gripper's grasp position, discouraging the agent from performing actions that lead to redundant movements. By focusing exploration within this constrained space, the agent can more effectively discover optimal manipulation strategies while minimizing unnecessary actions, leading to accelerated learning and improved policy performance.

During training, the placement of the target object within the robot hand's closing region poses a challenge due to the inherent randomness in exploration. Occasionally, episodes arise where the robot hand unintentionally encloses an obstacle instead of the target object. To tackle this variability and harness failed episodes for improved learning, we adopt the hindsight experience replay (HER) strategy. Drawing inspiration from [24], this approach involves replaying episodes where an obstacle is positioned in the closing region of the gripper and treating it as the target object. Leveraging insights from unsuccessful attempts, HER enhances learning efficiency, facilitates data augmentation, and mitigates challenges associated with sparse rewards in the learning process.

IV. Experiments and Results

We conducted experiments both in simulation and in a real environment to evaluate the performance of the proposed policy. The goals of the experiments are four-fold: 1) to investigate if the proposed architecture improves the derived policy compared to the existing methods, 2) to test if the design choices contribute to the increased quality of the final policy, 3) to test the effectiveness of the proposed policy on different environments and 4) to demonstrate the robust transfer in a real world robotic system without a significant drop in performance.

A. Comparison to Baselines

The simulation environment comprises a UR5e robotic arm equipped with an RG2 parallel gripper, operating within a tabletop workspace measuring 0.5×0.5 m. Additionally, a simulated 640×480 RGB-D camera is positioned overhead, providing a top view of the workspace. Each grasp is initiated after successfully placing the target object within the gripper's closing region. After closing the fingers, the hand is raised upwards by 20 cm. A grasp is considered successful if the target object was fully lifted and pertained within the hand for 5 seconds. We use the Bullet physics engine [25] to advance the simulation. All the dynamic parameters of the simulated environment are kept to their default values. We trained the proposed policy for 45K episodes. Each episode is initialized with objects arranged in a densely packed configuration, typically comprising 5 to 8 objects sampled from the "Seen" set as described in [4].

It's worth noting that direct comparisons with approaches focusing on top-down grasps may not be equitable, as such methods might not suffice for scenarios requiring subsequent tasks after grasping, potentially necessitating additional motions or adjustments for effective task completion. Thus, our approach will be compared with methods that specifically perform side grasps. These include: 1) **PPG** [4] which is a target-agnostic policy that plans side power grasps of objects in dense clutter using a single push-grasping action. 2) **Target-PPG** which is a variant of PPG designed for performing side grasps. Its operation is as follows: first, PPG generates Q-maps for target-agnostic tasks and executes actions predicted from PPG until enough free space is created around the target object. Subsequently, it performs a push-grasping action, starting from the free space and executing a linear pushing action until the target is within the closing region of the hand, where it grasps the target object. To execute the policies of PPG and Target-PPG, we utilize a floating Barrett Hand, as done in [4]. Notice that in [4], the evaluation of the PPG policy was based on the ratio of successful grasps to the total attempted grasps. In contrast, our evaluation approach focuses on target-oriented performance, where the success rate is determined by the proportion of successful episodes out of the total number of episodes. In each episode, the baseline policies attempt multiple grasp attempts, and the episode is considered successful only if the target object is grasped stably.

To assess the policies, we conduct N episodes, each starting with objects initialized in a densely packed configuration comprising 5-8 objects (Fig. 5a). A successful episode involves grasping the target object within 10 timesteps. Failure occurs if the maximum timestep is reached, the target falls off the surface, or the robot grasps an obstacle. Note that in this work, we measure the total number of motions executed by the robot. We utilize two object sets similar to

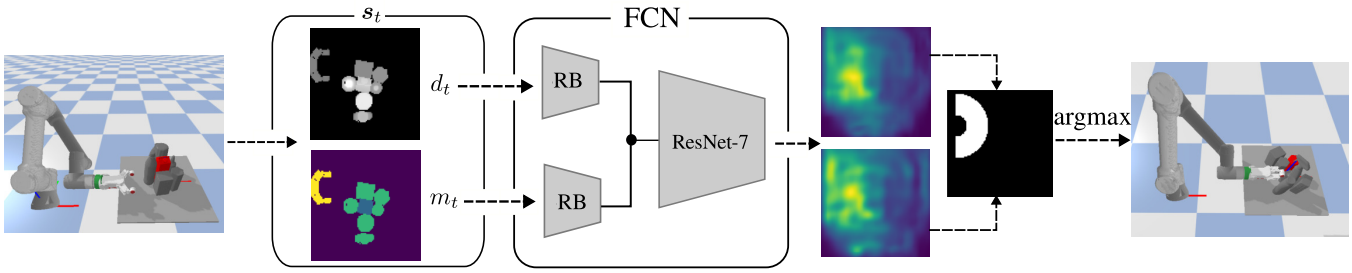


Fig. 3. Illustration of the proposed architecture. The FCN takes as input the depth heightmap d_t and the scene segmentation m_t and outputs 2 pixel-wise Q maps with the same size and resolution. These maps correspond to different apertures: the upper one for the closed aperture and the lower one for the open aperture. Each image is fed to a residual block RB(c) for feature extraction and subsequently the output features are concatenated and fed to a seven-layer fully convolutional residual network. Specifically, the ResNet-7 consists of the following layers: C(3,64)-MP-RB(128)-MP-RB(256)-RB(512)-RB(256)-RB(128)-UP-RB(64)-UP-C(1,1), where C(k, c) denotes convolutional layer with $k \times k$ filters and c channels, RB(c) denotes a residual block with two convolutional layers using 3×3 filters and c channels, MP denotes a 3×3 max pooling layer with stride = 2, and UP denotes a bilinear 2× upsampling layer. We filter the output by masking out the output maps and keeping only the pixels that correspond to the half-ring area.

TABLE I

PERFORMANCE EVALUATION ON SEEN AND UNSEEN OBJECTS (%)

Metric:	**Grasp success (%)**		**Mean Motions**		**Std Motions**	
Object Set:	Seen	Unseen	Seen	Unseen	Seen	Unseen
PPG	73.0	74.0	8.24	7.16	2.88	2.8
Target-PPG	77.0	80.0	3.96	4.52	1.68	2.18
Ours	**93.0**	89.0	2.51	**2.46**	**0.7**	0.9

[4]: the "Seen" set, used for training the proposed policy, and the "Unseen" set containing objects not encountered during training. Initially, we perform $N = 100$ evaluation episodes with objects sampled only from the "Seen" object set. Subsequently, we conduct another $N = 100$ episodes sampling objects only from the "Unseen" object set. These object sets were created by the YCB object set [26] and the KIT object set [27].

As shown in Table I the proposed policy not only achieves higher success rates with fewer actions compared to PPG and Mask-PPG but also exhibits robust performance across both "Seen" and "Unseen" object sets. By eliminating the need for the robot to repeatedly return to a home position between actions, the proposed policy leverages the kinematics of the robot to accomplish the task and also minimizes unnecessary movements. Specifically, the PPG policy for our target-oriented task shows a grasp success rate of 73% and 74% on "Seen" and "Unseen" object sets, with higher number of motions. PPG tends to prioritize easily graspable objects, sometimes leading to inadvertently pushing the target object out of the workspace while attempting to grasp neighboring objects. Target-PPG improves its task success rate to 77% and 80% on "Seen" and "Unseen" object sets, with a smaller number motions per episode. In comparison, our approach surpasses Target-PPG terms of task success rate and requires fewer motions on average.

B. Contribution of Design Choices

In order to investigate the impact of different design choices on the final policy, we conducted three distinct experiments using the "Seen" set of objects. Firstly, we trained an agent without employing the HER strategy (No-HER). As shown in Fig. 4, incorporating the HER strategy resulted in higher success rates and faster convergence to the optimal policy. Subsequently, we trained an agent with the ability to move back (Full-Ring), meaning that the post-processing step filtering the action space encompassed the entire ring. As it is observed from Table II, this agent required more actions to complete the task, often performing unnecessary actions, leading to slower convergence to the optimal policy. Finally, we trained an agent without guided exploration (Random-Exp), utilizing only random exploration. This experiment yielded poor performance, underscoring the importance of guided exploration during training. These experiments collectively highlight the significance of various design choices in shaping the learning process and influencing the effectiveness of the trained policy.

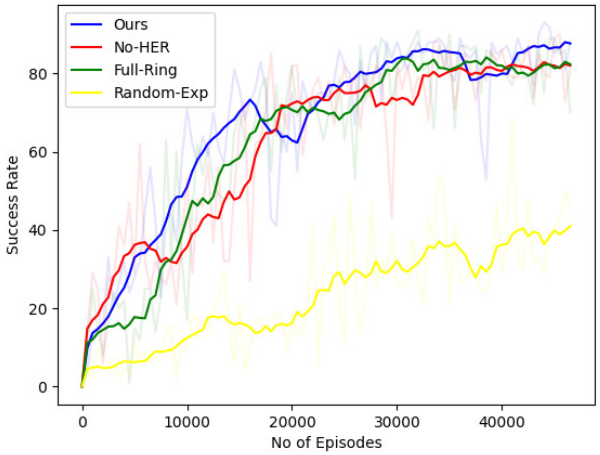


Fig. 4. Training performance for the ablation studies.

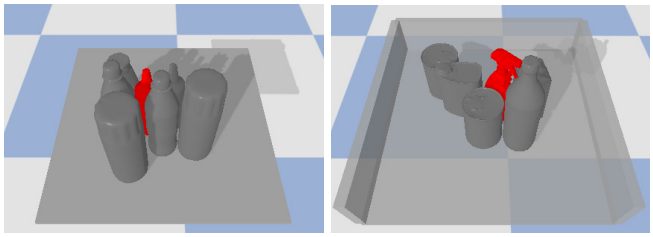
C. Performance Evaluation on a confined workspace

To assess the adaptability of the proposed policy to constrained environments resembling shelves, where collisions between the robot and the surroundings are more likely, we conducted an experiment in a confined space (Fig. 5b). This

TABLE II

PERFORMANCE EVALUATION OF THE ABLATION STUDIES POLICIES ON THE "SEEN" SET

Policy	Success rate	Mean motions	Std motions
Random-Exp	68.0	3.38	1.50
No-HER	89.0	3.04	1.4
Full-Ring	85.0	3.20	1.58



(a) (b)

Fig. 5. The simulation environment. (a) Open workspace (b) Confined workspace

setup emulates scenarios where the robot operates within limited spaces, such as shelves or narrow tabletops. We fine-tuned the policy by training it on additional 45K episodes and subsequently evaluated its performance using 100 episodes from both the "Seen" and "Unseen" object sets in the enclosed workspace.

Training the model from scratch without the pretrained weights resulted in a success rate of 54%. After fine-tuning in the new environment, the proposed policy demonstrated remarkable improvement, achieving an 83% and 84% success rate in the "Seen" and "Unseen" sets, respectively. Furthermore, the mean and standard deviation of motions during these successful grasps were found to be 2.61 and 1.17 for the "Seen" set of experiments and 2.45 and 0.76 for the "Unseen", respectively, indicating not only high success rates but also minimal variability in action execution, highlighting its effectiveness in navigating the constrained environment. These results verify that the agent learns to avoid positions that result in collisions between the body of the arm and the environment.

D. Transfer to the real world

Our real-world setup comprises a bimanual robot equipped with an RG2 parallel gripper on one arm and a wrist-mounted Intel RealSense D455 camera on the other arm. The camera captures RGB-D images, providing perception data for the robotic system. We performed 30 experiments with an object set which consists of 8 objects similar to the ones used in simulation experiments as shown in figure Fig. 6. Each scene contains $5-8$ objects arranged in a random, tightly packed configuration, with the target object being predefined before the episode begins and varying from one episode to another.

The conducted experiments consist of the following steps. At first, the object detection algorithm runs on the acquired RGB image, which in our case is a pretrained Mask-RCNN [28] fine tuned on the object set and generates a segmentation

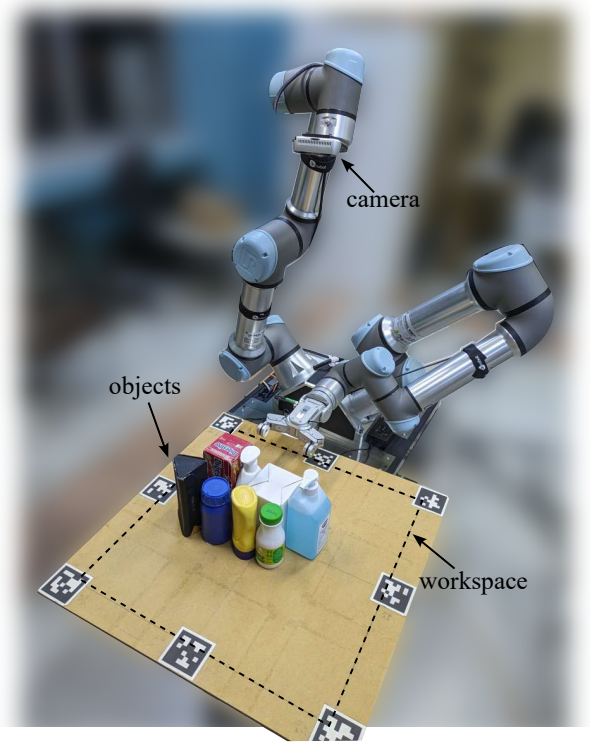


Fig. 6. Experimental setup for the real world experiments.

image that contains the mask of each object of the scene. Given the robot's joint positions and forward kinematics, we render its 3D model within the scene [22], to incorporate the robot into the segmentation image. Then, the depth and segmentation heightmaps are generated and are fed to FCN network to predict the final position of the push and the aperture of the hand. If the target object is inside the hands closing region, a grasp is executed by closing the gripper's fingers. To execute a pushing action, we plan smooth trajectories in the Cartesian space in order to move the robot arm to the final position $\boldsymbol{p}$ with the predicted aperture α. A grasp is executed by applying constant torques to the finger joints for 5 seconds. Finally, the robot hand was raised upwards by 10 cm. We consider a grasp to be successful if a target object was fully lifted and pertained to the hand.

The transferred policy achieves success rate of 83.3%, with an average of 3.1 motions per episode in the real world. This performance highlights the robustness and adaptability of the learned policy, showcasing its seamless transition from simulation to the real world without necessitating fine-tuning or extensive retraining efforts. However, this performance drop is accounted to the inherent noise present in real-world data captured by the camera and the disparities in physics between the simulated and real environments. These discrepancies, such as variations in friction coefficients and object dynamics, can impact the policy's decision-making process and lead to suboptimal outcomes compared to the simulated environment. Most failure cases occur when the

target object is toppled during the pushing motion, resulting in an unsuccessful outcome. Additional failures happen when an obstacle is within the gripper's closing region or when the target object is not stable grasped. In all these cases, the episode is considered as failed.

V. Discussion and Future Work

In summary, our novel pre-grasp manipulation policy tackles the challenge of robotic grasping in cluttered environments by strategically employing pushing actions and leveraging gripper kinematics. Unlike conventional approaches, our policy incorporates sequential pushing, reducing execution time per action. Extensive evaluation demonstrates superior performance, faster completion times, and robust generalization to unseen objects.

It is clear that the height z from the table at which the push starts restricts the number of objects that can be grasped, as shorter objects cannot be manipulated. Therefore, generating 6-DoF actions could be a promising approach, as it would allow the policy to move beyond parallel table pushing actions. This would enable the handling of objects with varying heights and orientations, thereby enhancing the policy's versatility and applicability in three-dimensional environments.

References

[1] A. Zeng, S. Song, S. Welker, J. Lee, A. Rodriguez, and T. Funkhouser, "Learning synergies between pushing and grasping with self-supervised deep reinforcement learning," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 4238–4245.

[2] Y. Yang, H. Liang, and C. Choi, "A Deep Learning Approach to Grasping the Invisible," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2232–2239, apr 2020.

[3] M. R. Dogar and S. S. Srinivasa, "Push-grasping with dexterous hands: Mechanics and a method," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 2123–2130.

[4] M. Kiatos, I. Sarantopoulos, L. Koutras, S. Malassiotis, and Z. Doulgeri, "Learning push-grasping in dense clutter," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 8783–8790, 2022.

[5] W. Bejjani, W. C. Agboh, M. R. Dogar, and M. Leonetti, "Occlusion-aware search for object retrieval in clutter," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 4678–4685.

[6] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven grasp synthesis—a survey," *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 289–309, 2013.

[7] B. Siciliano and O. Khatib, *Springer handbook of robotics*. Springer, 2016.

[8] A. Sahbani, S. El-Khoury, and P. Bidaud, "An overview of 3d object grasp synthesis algorithms," *Robotics and Autonomous Systems*, vol. 60, no. 3, pp. 326–336, 2012.

[9] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 705–724, 2015.

[10] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg, "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," in *Robotics: Science and Systems (RSS)*, 2017.

[11] M. Gualtieri, A. Ten Pas, K. Saenko, and R. Platt, "High precision grasp pose detection in dense clutter," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 598–605.

[12] A. Mousavian, C. Eppner, and D. Fox, "6-dof graspnet: Variational grasp generation for object manipulation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 2901–2910.

[13] A. Murali, A. Mousavian, C. Eppner, C. Paxton, and D. Fox, "6-dof grasping for target-driven object manipulation in clutter," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 6232–6238.

[14] M. Kiatos and S. Malassiotis, "Robust object grasping in clutter via singulation," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, May 2019, pp. 1596–1600.

[15] I. Sarantopoulos, M. Kiatos, Z. Doulgeri, and S. Malassiotis, "Split deep q-learning for robust object singulation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 6225–6231.

[16] I. Sarantopoulos, M. Kiatos, Z. Doulgeri, and S. Malassiotis, "Total singulation with modular reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 4117–4124, 2021.

[17] B. Huang, S. D. Han, J. Yu, and A. Boularias, "Visual foresight trees for object retrieval from clutter with nonprehensile rearrangement," *IEEE Robotics and Automation Letters*, vol. 7, no. 1, pp. 231–238, 2021.

[18] K. Xu, H. Yu, Q. Lai, Y. Wang, and R. Xiong, "Efficient learning of goal-oriented push-grasping synergy in clutter," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 6337–6344, 2021.

[19] E. Li, H. Feng, S. Zhang, and Y. Fu, "Learning target-oriented push-grasping synergy in clutter with action space decoupling," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11 966–11 973, 2022.

[20] Z. Liu, Z. Wang, S. Huang, J. Zhou, and J. Lu, "Ge-grasp: Efficient target-oriented grasping in dense clutter," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 1388–1395.

[21] A. Sidiropoulos and Z. Doulgeri, "From rgb images to dynamic movement primitives for planar tasks," in *2023 IEEE-RAS 22nd International Conference on Humanoid Robots (Humanoids)*, 2023, pp. 1–8.

[22] S. Sampaziotis, S. Antonakoudis, M. Kiatos, F. Dimeas, and Z. Dougleri, "A lightweight method for detecting dynamic target occlusions by the robot body," in *International Conference on Robotics in Alpe-Adria Danube Region*. Springer, 2023, pp. 3–11.

[23] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," *arXiv preprint arXiv:1511.05952*, 2015.

[24] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, O. Pieter Abbeel, and W. Zaremba, "Hindsight experience replay," *Advances in neural information processing systems*, vol. 30, 2017.

[25] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," 2016.

[26] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar, "The ycb object and model set: Towards common benchmarks for manipulation research," in *2015 International Conference on Advanced Robotics (ICAR)*, 2015, pp. 510–517.

[27] A. Kasper, Z. Xue, and R. Dillmann, "The kit object models database: An object model database for object recognition, localization and manipulation in service robotics," *The International Journal of Robotics Research*, vol. 31, no. 8, pp. 927–934, 2012.

[28] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.