Autonomous Robotic Manipulation with a Clutter-Aware Pushing Policy

Sanraj Lachhiramka, Pradeep J, Archanaa A Chandaragi, Arjun Achar, and Shikha Tripathi
Dept. of ECE, PES University, Bangalore, India - 560085

***Abstract*— This work addresses the challenge of grasping a target object in cluttered environments, even when it is partially visible or fully occluded. The proposed approach enables the manipulator to learn a sequence of strategic non-prehensile (pushing) actions to rearrange the scene and make the target object graspable. Our pipeline integrates a deep reinforcement learning (DRL) agent for pushing with the GR-ConvNet model for grasp prediction. When the object is considered ungraspable, a Soft Actor-Critic (SAC) model guides optimal pushing actions. A key contribution is a novel pixel-wise clutter map, which is integrated directly into the agent's state representation to provide an explicit, quantitative measure of environmental clutter. This clutter-aware state representation guides the decision-making process, leading to more efficient policies. Experimental results demonstrate that incorporating the clutter map significantly improves performance, reducing the number of actions required to complete the task by approximately 25%. The system generalizes well to diverse objects and transfers directly from simulation to hardware without requiring additional training for real-world deployment.**

***Index Terms*— Robotic manipulation, Cluttered environments, Deep reinforcement learning (DRL), Soft Actor-Critic (SAC), Strategic pushing, GR-ConvNet, Grasp planning, Continuous action space, Clutter map, Occluded object grasping.**

I. INTRODUCTION

Grasping a specific target object in a cluttered scene is a significant challenge for autonomous robots, often hindered by occlusion and restricted access. In such cases, non-prehensile actions like pushing are essential to rearrange the environment and improve the target's visibility and graspability. This paper introduces a system that learns to strategically declutter a scene to grasp a designated target.

Our approach separates the complex, strategic task of clutter reduction from the well-defined sub-task of grasp detection. We employ Deep Reinforcement Learning (DRL) exclusively for learning a pushing policy, avoiding the sample-inefficiency of using DRL for grasping. For grasp detection, we use GR-ConvNet model [5] . This partitioning makes the overall approach more practical for real-world application.

Our key contributions are: 1) **A Continuous Pushing Policy**: We utilize SAC in a continuous 5D action space (push point, angle, length, etc.), allowing finely tuned and adaptive motions, unlike prior discrete and fixed-length approaches. 2) **Clutter-Aware State Representation**: We propose a novel pixel-wise clutter map integrated into the DRL state space, providing a direct quantitative measure of clutter. 3) **End-to-End Sim-to-Real Transfer**: Our policy, trained in simulation with domain randomization, transfers directly to hardware without fine-tuning, made possible by normalized actions that generalize across workspaces.

The video demonstrating the implementation is available at: https://youtu.be/l8wNi_xMgeA

The code is available at: https://github.com/previous-team/armor

II. RELATED WORKS

Prior research in robotic manipulation has explored various strategies for coordinating prehensile (grasping) and non-prehensile (pushing) actions in cluttered environments. These approaches can be broadly categorized by their choice of action space and the degree to which they decouple the two sub-tasks. The table I summarizes key related works.

III. METHODOLOGY

A. System Overview and Problem Formulation

The task is formulated as a Markov Decision Process (MDP). At each timestep, the agent observes the state, selects an action, transitions to a new state and receives a reward. The pipeline as illustrated in Figure 1 operates in a loop: it first perceives the environment to determine the target's graspability using GR-ConvNet. If the target is not graspable, the SAC-based push policy is invoked to execute a decluttering action. This loop continues until a grasp is validated, at which point a grasping action is executed.

B. Learning the Pushing Policy

The policy is learned using the Soft Actor-Critic (SAC) algorithm, an off-policy actor-critic method designed for continuous control tasks. SAC's key feature is its entropy regularization framework, which encourages exploration by augmenting the standard reward objective with a policy entropy term. This helps the agent avoid premature convergence to suboptimal policies and learn more robust behaviors. The learning process is defined by the state the agent observes, the actions it can take, and the rewards it receives.

1) State Space (S_t):

The agent's ability to make informed decisions depends critically on its perception of the environment. The complete state representation at time step t for our model is defined as: $S_t = \{G, D, N, C, \rho\}$, where:

- G, D: Normalized grayscale and depth images $G, D \in \mathbb{R}^{H \times W}$ provide the agent with the primary visual and spatial context of the workspace. Normalization to the range ensures training stability.
- N: Pixel count of the target object is a scalar value representing the number of pixels corresponding to the target object. This is obtained by applying a color mask in the HSV space. This simple metric robustly quantifies the target's visibility without relying on complex segmentation, which helps prevent overfitting to specific object shapes or colors.

TABLE I: Comparison of Related Approaches for Robotic Manipulation in Cluttered Environments

Approach	Action Space (Push/Grasp)	Key Idea	Limitations
Zeng et al.	Discrete / Discrete	Two separate Q-networks for pushing and grasping policies.	Relies on fixed motion primitives, limiting precision.
Ren et al.	Discrete / Discrete	A Fast Learning Grasping (FLG) framework uses Q-learning.	Discrete actions are less effective in dense clutter.
Chen et al.	RL (Push) / Rule-based	Decouples pushing (RL) from grasping (morphological processing).	Pushing policy is target-agnostic.
Yang et al.	Discrete / Discrete	Bayesian policy for target search and a classifier for actions.	Constrained by predefined strategies.
Proposed work	Continuous / Supervised	SAC learns a 5D push action, guided by a novel pixel-wise clutter map.	Relies on a separate, pre-trained model for grasp generation.

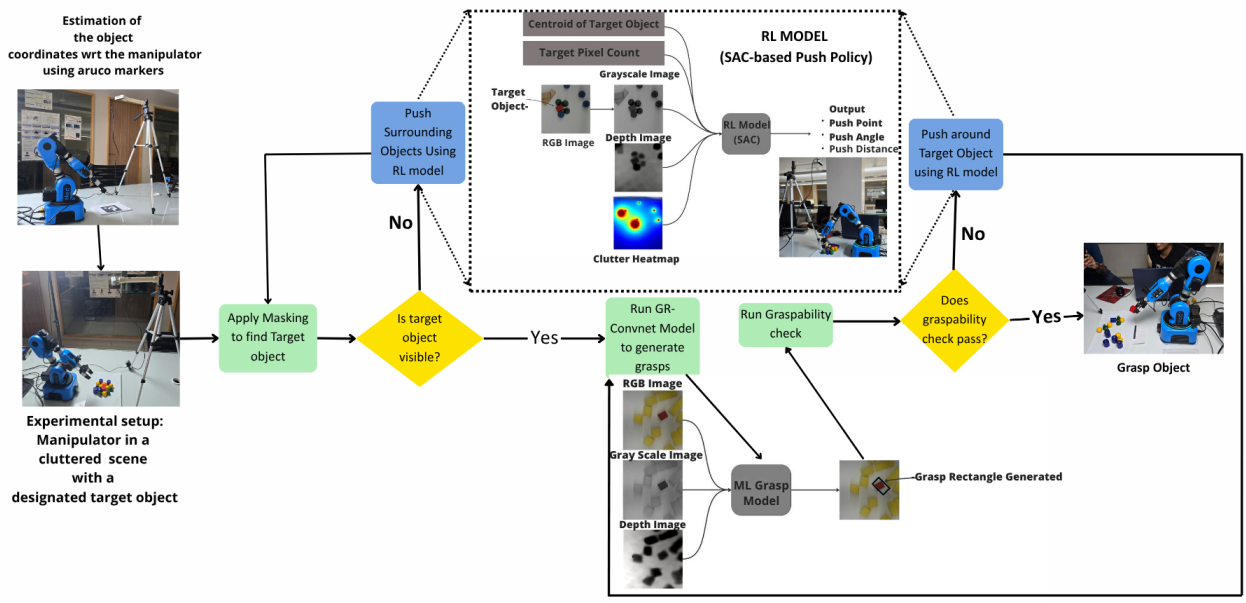

Fig. 1: Proposed pipeline combining GR-ConvNet with a continuous action space based SAC algorithm to aid decluttering and grasping of a designated target object in cluttered and occluded environments. The system first detects the target object using color masking and estimates visibility. If occluded, an RL-based push policy is used to reveal the object. Once visible, GR-ConvNet predicts potential grasp configurations, which are then verified using a graspability check before execution.

- C: Centroid $C = (x_c, y_c)$ of the visible portion of the target object provides positional information. If the object is not visible (N=0), a default value of (-1, -1) is used.
- ρ: Pixel-wise clutter map is the key component of our state representation. It encodes the spatial density of objects in the scene. The clutter score at a given point is influenced by all objects in the scene, based on their size (s_i) and distance (d_i) from that point. The score is directly proportional to an object's size s_i, since larger objects contribute more to clutter and inversely proportional to its distance d_i, reflecting the greater influence of nearby objects. The total clutter score at a pixel is given by:

$$\text{Total Clutter Score at a pixel} = \sum_{i=1}^{N} \frac{s_i}{d_i} \tag{1}$$

This formulation provides a information about the clutter intensity, allowing the agent to distinguish between

densely packed and sparse regions and to reason about where pushing actions would be most effective.

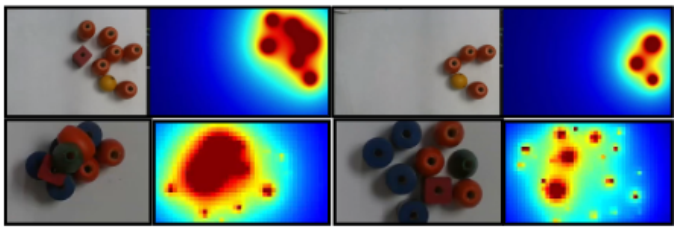

Fig. 2: RGB images (left) and respective pixel-wise clutter heatmaps (right) for various object arrangements. The clutter maps highlight densely packed regions, aiding the RL model in identifying suitable push locations to declutter the scene.

2) Action Space (A_t):

To enable precise and adaptive manipulation, we define a 5-dimensional continuous action space A_t=x,y,z,θ,l. Each component is normalized to facilitate stable learning and generalization:

- (x,y,z): Normalized 3D coordinates where the end-effector initiates its push motion.
- θ: Normalized yaw angle of the push, mapped to a physical range of [-180°, 180°].
- l: Normalized length of the push motion.

3) Reward:

The agent is trained using a dense reward function designed to guide it toward the desired outcome. The reward structure includes:

- Graspability Reward: A positive reward is given when the target becomes graspable.
- Visibility Reward: A positive reward is given for increasing the target's pixel count (N) and a negative reward is given for decreasing it.
- Clutter Reward: A positive reward is provided for reducing the mean clutter score (calculated from (ρ), either globally (if the target is not visible) or in a local region around the target's centroid (if it is visible).
- Collision Reward: A negative reward penalizes any detected collisions.

C. Grasp Generation and Validation

Once the SAC agent has rearranged the scene to a state where the target is potentially graspable, the GR-ConvNet model is used to generate antipodal grasp candidates from the RGB-D input. However, a predicted grasp may be infeasible due to surrounding objects that could cause a collision during the gripper's approach. To ensure physical feasibility, we implement a rule-based grasp validation check. For a given grasp candidate, a 3D grasp rectangle corresponding to the real-world dimensions of the gripper is computed and oriented according to the predicted grasp angle. Points are sampled along the edges of this rectangle, and their depth values are compared to the depth at the grasp center. A grasp is considered valid only if the depth values along the approach edges are greater than the depth at the center, ensuring a clear path for the gripper fingers. This validation step filters out infeasible predictions and significantly increases the reliability of executed grasps.

IV. EXPERIMENTAL VALIDATION

A. Experimental Setup

The proposed pipeline was trained and validated in a simulation environment built with Gazebo 11 and ROS Noetic, featuring a 6-DOF Niryo Ned2 robotic arm. To promote the learning of a generalizable policy, the training environment utilized domain randomization. At the start of each episode, the workspace was populated with a random number of objects (between 10 and 50) of varying shapes, sizes, and colors, creating diverse and challenging clutter configurations.

The hardware setup consisted of a physical Niryo NED2 manipulator and an overhead Intel RealSense D415 RGB-D camera. The transformation between the camera and robot coordinate frames was established using a fixed ArUco marker, enabling accurate localization of objects in the robot's workspace. The simulation-trained model was deployed directly to hardware without any fine-tuning.

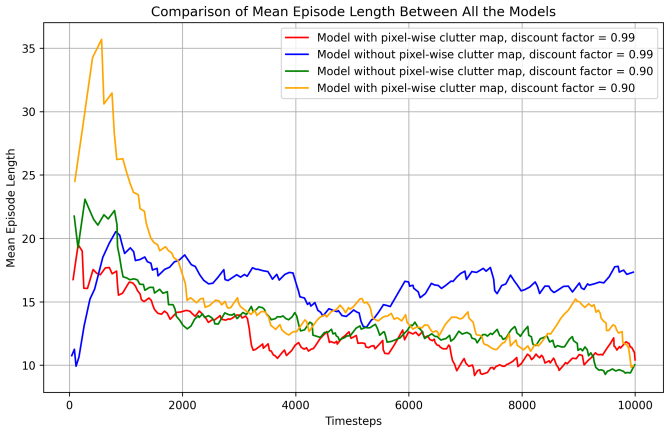

Fig. 3: Mean Episode Length Comparison Between All the Trained Models

B. Experimental Results

To quantify the benefit of our proposed clutter-aware state representation, we conducted a study comparing the performance of the SAC agent trained with the full state space (S_t=G,D,N,C,ρ) against a baseline agent trained without the pixel-wise clutter map (S_t=G,D,N,C). Both models were trained for 10,000 timesteps under identical conditions, using a discount factor of γ=0.90 and γ=0.99. The models were then evaluated over 50 episodes in a consistent test environment.

The results, summarized in Table II, shows a clear and significant performance improvement when the clutter map is included. The clutter-aware agent trained with γ=0.90 required approximately 25% fewer actions to complete the task (average episode length of 15.4 vs. 20.5) and achieved an average episode reward that was nearly an order of magnitude higher (17.2 vs. 1.7). The negative average reward for the baseline model indicates that it frequently became stuck in states from which it could not recover or took actions that reduced target visibility.

The learning efficiency is further illustrated in Fig. 3, which plots the mean episode length during training. The model with the clutter map not only achieves a lower final episode length

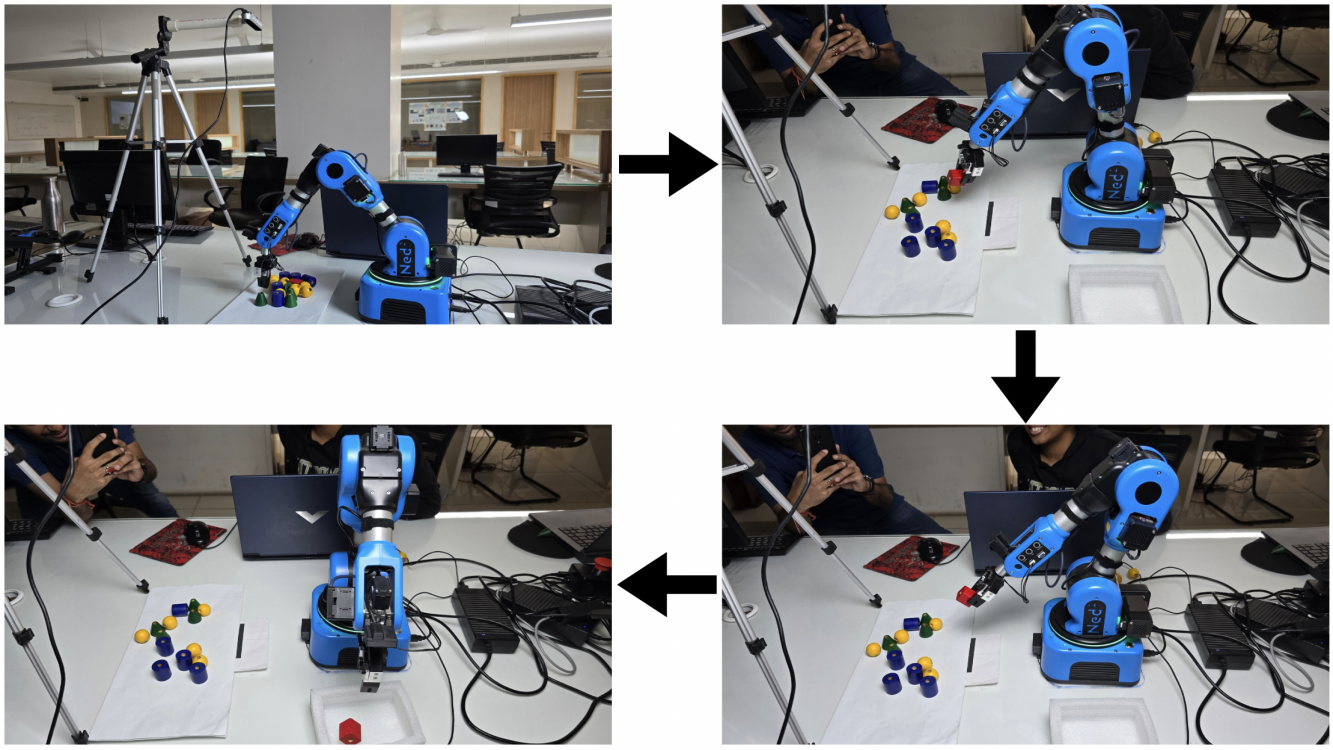

Fig. 4: Grasping and picking the target object in cluttered environment

TABLE II: Evaluation Results of Different Models

State Space	γ	Avg. Ep. Length	Avg. Reward
Without Clutter Map	0.99	35.5	-7.4
	0.90	20.5	1.7
With Clutter Map	0.99	28.8	15.1
	0.90	**15.4**	**17.2**

but also converges to an effective policy much more rapidly than the baseline model. This confirms that providing the agent with an explicit representation of clutter significantly accelerates the learning process.

C. Hardware Deployment

The simulation-trained policy transferred successfully to the physical hardware. This zero-shot transfer is attributed to the normalized action space, extensive world randomization while training and a state representation based on fundamental physical properties (depth, visibility, clutter) consistent between simulation and reality. On the hardware platform, the system demonstrated robust performance, successfully grasping various objects from cluttered arrangements, as shown in Fig. 4.

On the robot, the system grasped diverse objects (e.g., toothpaste tubes, pens) from cluttered scenes. Consistent with simulation, the clutter-map model showed more focused pushes around the target, while the baseline executed redundant actions, delaying completion. Figure 3 shows the robot successfully grasping a target object in a cluttered real-world scene.

V. CONCLUSIONS

We introduced a pipeline for autonomous robotic grasping in cluttered environments that leverages deep reinforcement learning for strategic pushing and using GR-ConvNet model for grasping. Our primary finding is that explicitly encoding environmental clutter as a feature in the state space significantly improves both the final performance and the learning efficiency of the DRL agent. The clutter-aware model learned to complete its task with fewer actions and achieved substantially higher rewards compared to a baseline without this information. The successful zero-shot sim-to-real transfer underscores the robustness of the approach. Future work could explore more advanced target segmentation methods to enhance generalization further.

REFERENCES

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

[2] Y. Chen, Z. Ju and C. Yang, "Combining Reinforcement Learning and Rule-based Method to Manipulate Objects in Clutter," 2020 International Joint Conference on Neural Networks (IJCNN), pp. 1-6, 2020.

[3] D. Ren, X. Ren, X. Wang, S. T. Digumarti and G. Shi, "Fast-Learning Grasping and Pre-Grasping via Clutter Quantization and Q-map Masking," 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3611-3618, 2021.

[4] 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," 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4238-4245, 2018.

[5] S. Kumra, S. Joshi and F. Sahin, "Antipodal Robotic Grasping using Generative Residual Convolutional Neural Network," *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Las Vegas, NV, USA, 2020, pp. 9626-9633, doi: 10.1109/IROS45743.2020.9340777.