CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

ORCA – Online Research @ Cardiff

This is an Open Access document downloaded from ORCA, Cardiff University's institutional repository:https://orca.cardiff.ac.uk/id/eprint/181994/

This is the author's version of a work that was submitted to / accepted for publication.

Citation for final published version:

Gong, Meiyuan and Ji, Ze 2025. Reinforcement learning-based pushing-grasping tasks with grasp success prediction. Presented at: 30th International Conference on Automation and Computing (ICAC), Loughborough, United Kingdom, 27-29 August 2025. 2025 30th International Conference on Automation and Computing (ICAC). IEEE, 10.1109/icac65379.2025.11196276

Publishers page: https://doi.org/10.1109/icac65379.2025.11196276

Please note:
Changes made as a result of publishing processes such as copy-editing, formatting and page numbers may not be reflected in this version. For the definitive version of this publication, please refer to the published source. You are advised to consult the publisher's version if you wish to cite this paper.

This version is being made available in accordance with publisher policies. See http://orca.cf.ac.uk/policies.html for usage policies. Copyright and moral rights for publications made available in ORCA are retained by the copyright holders.

Reinforcement Learning-based Pushing-Grasping Tasks with Grasp Success Prediction

Meiyuan Gong
School of Engineering
Cardiff University
Cardiff, United Kingdom
gongm3@cardiff.ac.uk

Ze Ji
School of Engineering
Cardiff University
Cardiff, United Kingdom
jiz1@cardiff.ac.uk

***Abstract*—Robotic grasping in cluttered environments presents significant challenges due to occlusions and physical interactions that hinder successful object retrieval. Traditional grasping strategies often fail in densely cluttered scenes, where direct grasp attempts lead to frequent collisions or inaccessible target objects. To address this limitation, we propose an RL pushing-grasping framework that incorporates a grasp success prediction (GSP) model to guide pre-grasp manipulation. An autoencoder compresses RGB-D data and object masks into a latent representation for a predictor that estimates grasp success. When this estimate falls below a threshold, an SAC-trained agent first chooses to push to improve the object's pose and then executes the grasp. The proposed method is evaluated in randomised cluttered environments within the PyBullet simulation platform. Experimental results demonstrate that our approach significantly improves grasping performance, achieving an 81% compared to 56% for a heuristic-based baseline, showing that it selects effective movements for rearranging the scene. These results show that predicting grasp success is important for deciding how to prepare the scene, helping the robot retrieve objects more efficiently and reliably in cluttered environments.**

***Index Terms*—Robotic grasping, manipulation, reinforcement learning, grasp success prediction, cluttered environments**

I. Introduction

Grasping is a basic problem in robot manipulation. This research field has improved greatly in recent years, especially by studying how pushing and grasping actions can work together when the environment becomes more complicated [1]. The purpose is to use pushing to achieve a pre-grasping condition in the cluttered environment and learn the synergy between the two actions [2] [3] [4]. By leveraging insights derived from human actions, the agent is encouraged to utilise action primitives to isolate the target object from a densely cluttered environment, thereby facilitating grasping [5]. To achieve this goal, the agent integrates visual information from both the current scene and the target object to infer the necessary actions for a given scenario [2], rather than relying solely on basic target-driven grasping. Learning how to push effectively is essential for improving task efficiency, as pushing actions can help isolate or expose the target object within the clutter [6]. Therefore, to improve the efficiency of our approach, we train the pushing network using a reward signal only when an increase in the predicted grasping success rate is observed. While most existing methods rely on Q-maps from the pushing network or execute pushing actions at the highest point in the workspace , our approach encourages the agent to perform more purposeful actions that facilitate future grasping.

In our approach, we first introduce a grasp success predictor to estimate the probability of a successful grasp given the current scene. This predictor assesses grasp feasibility, helping the agent to make informed decisions before attempting to grasp. If the predicted success rate is below a predefined threshold, we employ an RL model [7] to determine optimal pushing actions that enhance the graspability of the target object. The RL model learns effective strategies to manage clutter by pushing obstructing objects aside and isolating the target object from its surroundings. The agent iteratively executes these pushing actions until the grasp success prediction (GSP) surpasses the threshold, after which it performs the grasp to achieve the desired goal.

Our framework, shown in Fig. 1, is capable of allowing the agent to actively interact with the environment rather than passively attempting to grasp under unfavourable conditions. Compared to existing methods that rely solely on static Q-maps or heuristic-based pushing strategies [2] [8], our approach dynamically adapts to the scene, ensuring that each action meaningfully contributes to a successful grasp.

The main contributions are summarised below:

- We introduce a grasping success rate predictor that estimates the likelihood of a successful grasp based on the current scene observation and object mask, enabling more informed decision-making.
- We design an RL-based pushing network that is trained using feedback from the predictor, allowing the agent to strategically manipulate objects and improve grasp efficiency.
- We propose a unified framework integrating predictive modelling with reinforcement learning, enabling the agent to adaptively and dynamically improve grasp conditions through targeted scene rearrangements.

II. Related Work

In this section, we review relevant literature on grasping strategies and pre-manipulation techniques.

A. Grasping

Grasping is a fundamental capability in robotic manipulation and serves as the foundation for executing tasks in complex environments. Many recent approaches leverage RL to model grasping frameworks, where predefined reward functions encourage the agent to maximise grasp success and generalise across diverse scenarios [9]. These methods enable robotic systems to autonomously learn grasping strategies by interacting with objects in both simulated and real-world environments, improving their adaptability to unseen configurations [10].

However, in dense, cluttered environments, direct grasping is often challenging due to occlusions and collisions between the target object, surrounding objects, and the robotic gripper [11]. These unintended interactions frequently lead to grasp failures and reduce retrieval success rates. To mitigate this issue, some works incorporate scene modification techniques to rearrange clutter before attempting a grasp [12]. These approaches improve grasp feasibility by creating more accessible grasp configurations, reducing occlusion, and ensuring more stable object interactions [13], [14].

B. Pushing

Pre-manipulation has been increasingly explored in robotic manipulation to improve target object retrieval in cluttered environments [15] [16]. Non-prehensile actions, particularly pushing, have been extensively studied in robotic manipulation as a means of reorganising cluttered environments to facilitate grasping. Pushing has proven to be an effective strategy for reducing occlusions, rearranging objects, and improving the accessibility of a target object [5] [12]. Recent research has explored both heuristic-based and learning-based approaches to optimise pushing actions [17].

Previous works have applied learning-based methods to rearrange the environment, aiming to isolate the target object and improve graspability [18]. Some approaches rely on predefined rules, such as pushing the highest point in the workspace, while others incorporate deep RL to enable adaptive pushing strategies. However, many existing methods treat pushing as an independent action without explicitly evaluating its impact on subsequent grasp success [15].

C. Synergy Between Grasping and Pushing

Integration of pushing and grasping has been extensively explored in recent years, particularly through RL. Pre-grasp manipulation strategies, such as pushing, can significantly improve grasp success rates by modifying the spatial arrangement of objects, thereby increasing the accessibility of the target object [19].

Among the most relevant works, Zeng et al. proposed a framework that jointly learns pushing and grasping policies using a deep Q-network (DQN). The method employs two Q-maps of equal size to encourage useful pushing actions until a grasp is successful [2]. Similarly, in [5], learning optimal push policies by trial and error. However, their reward structure prioritises pushing actions as long as any object is moved, without explicitly considering the efficiency or necessity of the push in relation to the final grasp.

Other studies have introduced dual Q-maps that separately represent pushing and grasping actions, with rewards only assigned when the grasping Q-value exceeds a predefined threshold [20] [21]. Similarly, Kalashnikov et al. proposed QT-Opt, a closed-loop RL framework for bin-picking tasks, but its application is limited by the high computational cost of training [22]. Liu et al. introduced GE-Grasping, which focuses on reducing occlusions near the target object through learned pushing action [9]. While their approach efficiently integrates pushing and grasping, its reliance on predefined action primitives may limit adaptability to more dynamic and unstructured environments [9].

Our method explicitly evaluates the effectiveness of each pushing action based on real-time GSP. Instead of treating pushing as a separate optimisation task, our system learns to execute only the most beneficial pushing actions, directly maximising grasp feasibility. By integrating GSP into the RL process, our approach reduces unnecessary pushing and optimises the synergy between non-prehensile and prehensile actions for more efficient target retrieval.

III. Method

This section provides an overview of the proposed method (see Fig. 1), followed by a detailed explanation of our prediction module and the RL-based pushing and grasping network.

A. Overview

The goal for our system is to improve grasp success in cluttered environments by integrating the GSP model and an RL-based pushing strategy. The main idea is to enable the agent to evaluate the graspability of an action before attempting the grasp. Rather than directly grasping, the agent first processes an RGB-D image and obtains the mask of the target object. A pre-trained autoencoder then encodes this information into a compact latent representation. Fig. 2 shows how autoencoder converts raw RGB-D data into the latent representation. The grasp prediction model utilises this representation to estimate the likelihood of a grasp succeeding. If this estimated probability exceeds a predefined threshold, the agent then proceeds with the grasp.

However, when the grasp success probability is below the pre-defined threshold, the agent applies an RL-based pushing strategy to improve object accessibility. The trained pushing network determines the optimal pushing action, modifying the scene to create a more favourable grasp condition. After each push, the system re-evaluates the grasp success probability, repeating the process until the threshold is met. By integrating predictive modelling with RL, our system enables the agent to interact with the environment actively rather than relying on direct, potentially ineffective grasp attempts.

B. Prediction module

An overhead RGB-D camera first gathers both colour and depth data for the scene. The target object's mask is also

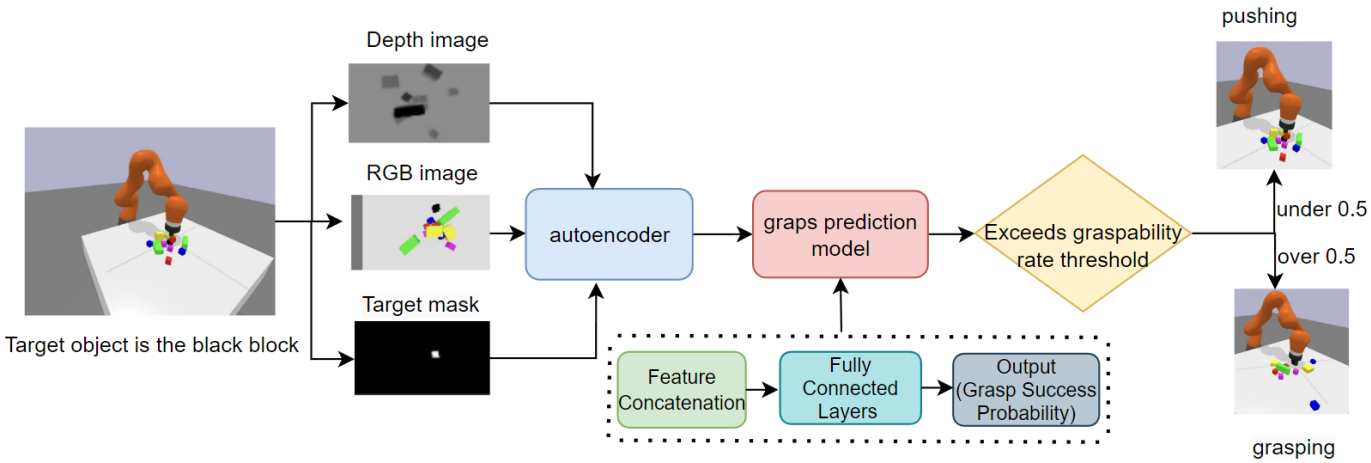

Fig. 1: **System Overview**: Our system is implemented in a **PyBullet** simulation environment using a KUKA robotic arm equipped with an RGB-D camera. The input includes the RGB-D image, which is processed through an autoencoder to generate a latent space representation of the cluttered scene. The grasp prediction model evaluates the graspability of the target object based on the encoded latent features to decide the corresponding actions. The process iterates until the grasp success probability meets the threshold.

provided, allowing a pre-trained autoencoder to encode the images. However, if the object is heavily blocked by clutter, only a partial or no mask can be obtained. By combining the RGB-D image and the mask, the autoencoder learns a compact latent representation that captures key spatial features.

This latent space serves as a high-level encoding of the environment, enabling the agent to understand the spatial relationships between objects better and make more informed decisions. The GSP model utilises this learned representation to estimate the likelihood of a successful grasp. Given an input scene, the model outputs a grasp feasibility probability. If the predicted grasp success rate is low, the agent opts to perform pushing to improve the scene configuration before attempting a grasp.

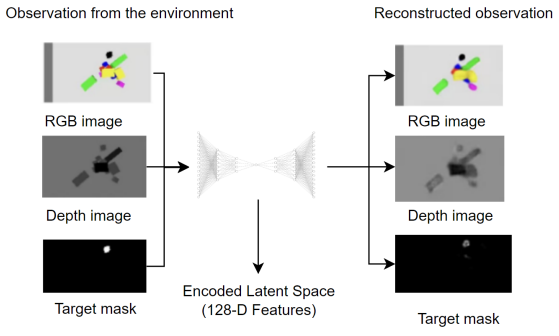

Fig. 2: Autoencoder to process observation data

C. RL-based pushing network

To improve grasp success in cluttered environments, we use an RL model to strategically push objects before grasping. Instead of following fixed pushing actions, the agent learns optimal pushes based on the scene's latent representation.

We define this problem using RL notations here. The agent interacts with the environment through a sequence of states s_t, actions a_t, and rewards r_t. At each timestep t, the agent observes the current observation s_t, selects an action a_t, and receives a reward r_{t+1} along with transitioning to the next state s_{t+1}. The goal of the agent is to maximise the expected cumulative reward $G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$, where γ ($0 \leq \gamma \leq 1$) is the discount factor, representing the importance of future rewards.

1) Observation Space: The observation space contains the RGBD images s_t provide essential information that guides the agent's decisions when selecting pushing actions. Formally, it is represented as $s_t = \{l_t, p_t\}$, where l_t denotes the latent space representation of scene features encoded by autoencoder, integrating RGB-D image data and the mask of the target object, and p_t is the grasp success probability, the predicted likelihood of successful grasping, provided by the grasp prediction model, offering immediate feedback about grasp feasibility.

2) Action Space: At each step, the agent chooses an initial contact point $\mathbf{p}_t = (x_t, y_t, z_t)$ inside a $10\,\text{cm} \times 10\,\text{cm} \times 5\,\text{cm}$ box centred on the target object:

$$a_t = \{(x_t, y_t, z_t) \mid |x_t| \leq 5\,\text{cm},\ |y_t| \leq 5\,\text{cm},\ |z_t| \leq 2.5\,\text{cm}\}.$$

After selecting $\mathbf{p}_t$, the gripper height z_t is fixed and the gripper moves a constant $5\,\text{cm}$ in a predetermined horizontal direction within the xy–plane.

The agent learns to strategically push objects to maximise the grasp success probability. This ensures that objects are rearranged in a way that makes grasping easier and more successful. Using RL, the model continuously adapts its pushing actions based on immediate feedback, allowing the agent to dynamically adjust the scene and improve grasp performance in cluttered environments.

D. Model Training

1) Training the Grasp Success Prediction Model: Data Collection: To train the grasp success predictor, we first collect a dataset of **30,000 RGB-D images** with corresponding target object masks from diverse cluttered environments. Each grasp attempt is labelled based on its outcome: We define labels for grasp outcomes using the following binary representation:

$$y = \begin{cases} 1, & \text{if grasp succeeds} \\ 0, & \text{if grasp fails} \end{cases} \tag{1}$$

We train a supervised learning model using this labelled dataset to predict grasp success probability. The training pipeline consists of:

- **Feature Extraction (Autoencoder-based)**: RGB-D images and object masks are encoded into a 128-dimensional latent representation.
- **Prediction Model Architecture**: A multi-layer neural network maps the latent features to a grasp success probability. ReLU activation functions are used in intermediate layers, and a sigmoid activation layer is used in the final layer to produce a probability.
- **Loss Function and Optimisation**: The model is trained using Focal loss and optimiser is AdamW with an adaptive learning rate.
- **Evaluation Metrics**: Accuracy for overall correctness, F1-score for the precision–recall balance, and AUC-ROC, which shows how often the model ranks a true success above a failure without fixing a decision threshold.

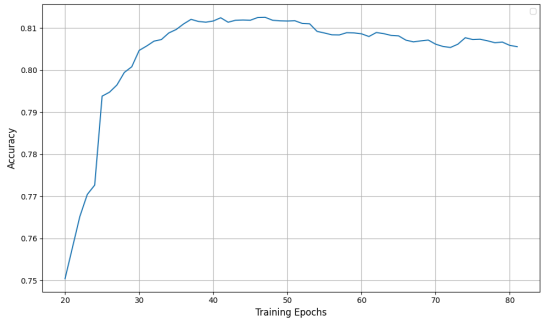

Fig. 3: Accuracy during training for grasping success rate prediction model.

2) Training the RL-Based Pushing Model: The pushing policy is trained using SAC from stablebaseline3 [23], an RL algorithm designed for continuous action spaces. SAC is chosen due to its sample efficiency, stability, and ability to balance exploration and exploitation through entropy regularisation.

The agent receives a reward based on the improvement in the predicted grasp success probability after executing a pushing action. If the action increases the grasp success probability beyond 0.5 and results in a meaningful improvement ($\Delta P_{success} > \delta$), the agent is positively rewarded. Conversely, if the action fails to enhance the likelihood of grasp success, a penalty is applied. The reward function is formulated as:

$$R(s,a) = \begin{cases} 1 + \Delta P_{success}, & \text{if } P_{success}(s') > 0.5 \\ -1, & \text{otherwise} \end{cases} \tag{2}$$

where $\Delta P_{success} = P_{success}(s') - P_{success}(s)$ represents the change in the predicted grasp success probability after executing action a. This formulation encourages the agent to perform pushing actions that significantly increase grasp feasibility. The base reward of 1 ensures that the agent is always encouraged to interact with the environment, while the additional term $\Delta P_{success}$ further rewards actions that lead to greater improvements in grasp success probability.

Fig. 4 illustrates the training process of the RL-based pushing policy, showing the reward progression over training episodes.

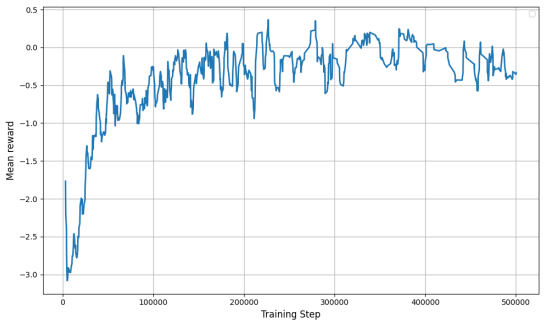

Fig. 4: Training reward curve of the RL-based pushing policy.

IV. EXPERIMENT AND RESULTS

To evaluate our RL-based pushing-grasping strategy, we conducted experiments in simulated environments. The objective was to compare its success rate and efficiency against baseline methods and assess its stability across different cluttered scenarios, ensuring consistent performance.

A. Experiment Setup and Metrics

Specifically, our environment is adapted from the PyBullet Multi-Goal RL framework [24], originally designed for multi-goal robotic manipulation. We modified this environment to better suit our pushing-grasping task by incorporating a GSP model, redefining the reward structure, and customising object placements to generate diverse cluttered scenarios.

The environment contains 11 blocks that vary randomly in size, colour, and pose. At the start of each episode, the blocks are dropped from a fixed height, creating diverse clutter that challenges the agent's adaptability.

To assess the performance of our RL-based pushing-grasping strategy, we compare it against baseline methods under identical simulation settings. The evaluation metrics used in our experiments are:

- Grasp Success Rate (GSR): The percentage of successful grasps overall attempts.
- Task Completion Rate (TCR): Percentage of completed the task.

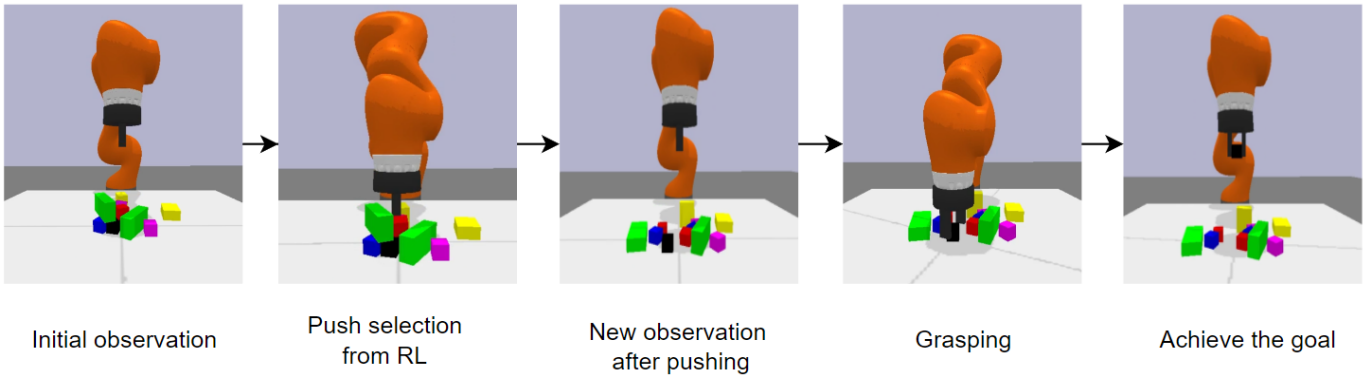

Fig. 5: **Experiment**:The sequential process of our pushing-grasping strategy. The agent first observes the scene, selects a pushing action based on RL, updates its observation after pushing, and finally executes the grasping action to achieve the goal.

- Average Push Count (APC): The mean number of pushing actions.
- Push Success Rate (PSR): The percentage of pushing actions that led to an increase in grasp success probability.

B. Baseline Method

To evaluate the effectiveness of our proposed method and assess the impact of the GSP model, we design a baseline approach that does not utilise grasp success probability predictions. Instead of making decisions based on estimated grasp feasibility, the baseline method follows a **fixed heuristic** for selecting pushing actions. However, aside from the absence of the prediction model, the baseline method is trained using the same SAC-based RL framework as our proposed method.

The primary objective of the baseline is to modify the spatial structure of the environment in a way that improves the likelihood of successfully grasping the target object. The baseline method follows a height-based heuristic to determine where to push. Specifically, the agent identifies the height distribution in the scene, operating under the assumption that taller structures are more likely to occlude or obstruct the target object when the target is not visible. The agent then executes a pushing action in that region to clear potential obstructions.

The baseline agent is allowed to perform a maximum of five pushing actions before attempting a grasp. Both methods share the same RL setup, network architecture, and training strategy using SAC. The only difference is that the baseline agent does not receive feedback from a grasp success predictor.

To ensure a fair comparison, the agent in both approaches receives rewards under the same conditions. The agent in the baseline is rewarded only when a grasp is successful or when the latent space representation of the scene changes, thereby reinforcing the goal of actively modifying the environment's layout to facilitate grasping.

By comparing this baseline with our method, we aim to demonstrate the advantages of prediction-driven pushing strategies, particularly the role of GSP in guiding adaptive and efficient object manipulation. The evaluation is conducted based on grasp success rate and task completion rate, allowing for a direct comparison between static heuristic-based pushing and learning-based adaptive pushing.

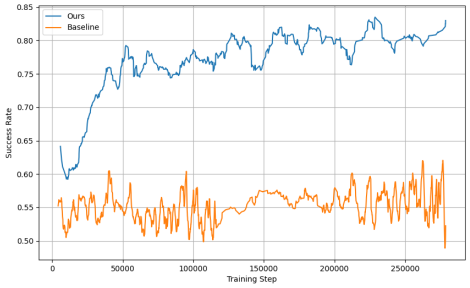

Fig. 6: Success rate comparison over training episodes.

C. Results and Analysis

To evaluate the effectiveness of our approach, we conducted experiments in **100 randomly generated cluttered environments** and compared our method with the baseline, which follows the same SAC-based RL framework but does not incorporate GSP. The results are summarised in Table I.

TABLE I: Performance comparison between our method and the baseline

Metric	Ours	Baseline
Grasp Success Rate (GSR)	88%	56%
Task Completion Rate (TCR)	81%	44%
Average Push Count (APC)	1.9	5
Push Success Rate (PSR)	86%	58%

The results demonstrate that our method significantly outperforms the baseline in all evaluation metrics. Our approach achieves an 88% grasp success rate, a notable improvement over the baseline 56%, highlighting the advantage of integrating GSP. Similarly, the task completion rate improves from 44% in the baseline to 81%, showing that the agent is more effective in retrieving the target object. Fig. 6 demonstrates

how our method successfully completes the task with just a single push.

A key observation is the substantial increase in push success rate. Since the baseline method follows a predefined heuristic-based pushing strategy rather than learning an adaptive policy, it often performs ineffective pushes that do not contribute to grasping success. In contrast, our approach evaluates the effectiveness of each push based on GSP, ensuring that pushing actions are executed only when they improve the target object's accessibility.

These findings suggest that incorporating a GSP model allows the agent to make more informed pushing decisions, effectively reducing unnecessary scene modifications and enhancing grasp efficiency in cluttered environments.

V. Conclusion

In this work, we presented an RL-based pushing-grasping framework that integrates GSP to enhance object retrieval in cluttered environments. By leveraging an autoencoder to encode RGB-D observations and target object masks into a latent space, our method enables real-time estimation of grasp success probability. When direct grasping is deemed infeasible, an SAC-based RL agent executes non-prehensile pushing actions to rearrange the environment before attempting a grasp.

Extensive evaluations in randomly generated cluttered environments demonstrated that our approach significantly outperforms a heuristic-based baseline. Moreover, our method ensures that pushing actions are executed in a targeted and efficient manner to facilitate grasping. These findings highlight the importance of integrating GSP into RL-based manipulation strategies to achieve adaptive and efficient robotic interaction in unstructured environments.

Our long-term goal is to further enhance the adaptability of our framework by incorporating more sophisticated scene understanding and action planning, enabling robots to operate effectively in highly dynamic and real-world environments.

References

[1] 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*, vol. 39, no. 5, pp. 3994–4015, 2023.

[2] 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.

[3] 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.

[4] Y. Wang, K. Mokhtar, C. Heemskerk, and H. Kasaei, "Self-supervised learning for joint pushing and grasping policies in highly cluttered environments," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 13 840–13 847.

[5] 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)*. IEEE, 2020, pp. 6225–6231.

[6] B. Tang, M. Corsaro, G. Konidaris, S. Nikolaidis, and S. Tellex, "Learning collaborative pushing and grasping policies in dense clutter," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 6177–6184.

[7] S. Ravichandiran, *Hands-on reinforcement learning with Python: master reinforcement and deep reinforcement learning using OpenAI gym and tensorFlow*. Packt Publishing Ltd, 2018.

[8] 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, 2020.

[9] 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.

[10] M. Kang, H. Kee, J. Kim, and S. Oh, "Grasp planning for occluded objects in a confined space with lateral view using monte carlo tree search," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 10 921–10 926.

[11] 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.

[12] B. Serhan, H. Pandya, A. Kucukyilmaz, and G. Neumann, "Push-to-see: learning non-prehensile manipulation to enhance instance segmentation via deep q-learning," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 1513–1519.

[13] Z. Xu, B. Qi, S. Agrawal, and S. Song, "Adagrasp: Learning an adaptive gripper-aware grasping policy," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 4620–4626.

[14] W. Wei, P. Wang, S. Wang, Y. Luo, W. Li, D. Li, Y. Huang, and H. Duan, "Learning human-like functional grasping for multi-finger hands from few demonstrations," *IEEE Transactions on Robotics*, 2024.

[15] L. Wu, Y. Chen, Z. Li, and Z. Liu, "Efficient push-grasping for multiple target objects in clutter environments," *Frontiers in Neurorobotics*, vol. 17, p. 1188468, 2023.

[16] H. Zhang, H. Liang, L. Cong, J. Lyu, L. Zeng, P. Feng, and J. Zhang, "Reinforcement learning based pushing and grasping objects from ungraspable poses," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 3860–3866.

[17] M. Kiatos, I. Sarantopoulos, S. Malassiotis, and Z. Doulgeri, "Improved reinforcement learning pushing policies via heuristic rules," *arXiv preprint arXiv:2111.11156*, 2021.

[18] Y. Geng, B. An, H. Geng, Y. Chen, Y. Yang, and H. Dong, "Rlafford: End-to-end affordance learning for robotic manipulation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5880–5886.

[19] B. Huang, S. D. Han, A. Boularias, and J. Yu, "Dipn: Deep interaction prediction network with application to clutter removal," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 4694–4701.

[20] H. Kasaei and M. Kasaei, "Harnessing the synergy between pushing, grasping, and throwing to enhance object manipulation in cluttered scenarios," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 13 855–13 861.

[21] X. Cao, T. Lu, L. Zheng, Y. Cai, and S. Wang, "Plot: Human-like push-grasping synergy learning in clutter with one-shot target recognition," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 16, no. 4, pp. 1391–1404, 2024.

[22] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke *et al.*, "Scalable deep reinforcement learning for vision-based robotic manipulation," in *Conference on robot learning*. PMLR, 2018, pp. 651–673.

[23] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, and R. Traoré, "Stable-baselines3: Reliable reinforcement learning implementations," Journal of Machine Learning Research (JMLR), 22(268):110, 2021. [Online]. Available: https://github.com/DLR-RM/stable-baselines3

[24] X. Yang, Z. Ji, J. Wu, and Y.-K. Lai, "An open-source multi-goal reinforcement learning environment for robotic manipulation with pybullet," in *Towards Autonomous Robotic Systems (TAROS)*. Springer, 2021, pp. 14–24.