Collaborative Pushing and Grasping of Tightly Stacked Objects via Deep Reinforcement Learning

Yuxiang Yang, Zhihao Ni, Mingyu Gao, Jing Zhang, *Member, IEEE*, and Dacheng Tao, *Fellow, IEEE*

***Abstract*—Directly grasping the tightly stacked objects may cause collisions and result in failures, degenerating the functionality of robotic arms. Inspired by the observation that first pushing objects to a state of mutual separation and then grasping them individually can effectively increase the success rate, we devise a novel deep Q-learning framework to achieve collaborative pushing and grasping. Specifically, an efficient non-maximum suppression policy (PolicyNMS) is proposed to dynamically evaluate pushing and grasping actions by enforcing a suppression constraint on unreasonable actions. Moreover, a novel data-driven pushing reward network called PR-Net is designed to effectively assess the degree of separation or aggregation between objects. To benchmark the proposed method, we establish a dataset containing common household items dataset (CHID) in both simulation and real scenarios. Although trained using simulation data only, experiment results validate that our method generalizes well to real scenarios and achieves a 97% grasp success rate at a fast speed for object separation in the real-world environment.**

***Index Terms*—Convolutional neural network, deep Q-learning (DQN), reward function, robotic grasping, robotic pushing.**

I. Introduction

GRASPING is one of the most fundamental problems in the area of robotics [1], [2], which has important applications in many scenarios, such as sorting robot, service robot and human-robot interaction. It has attracted increasing attention in recent years, however, remaining challenging for a robot arm to grasp tightly stacked objects automatically.

Manuscript received April 24, 2021; revised June 14, 2021; accepted July 6, 2021. This work was supported by the National Natural Science Foundation of China (61873077, 61806062), Zhejiang Provincial Major Research and Development Project of China (2020C01110), and Zhejiang Provincial Key Laboratory of Equipment Electronics. Recommended by Associate Editor Hui Yu. *(Corresponding author: Jing Zhang.)*

Citation: Y. X. Yang, Z. H. Ni, M. Y. Gao, J. Zhang, and D. C. Tao, "Collaborative pushing and grasping of tightly stacked objects via deep reinforcement learning," *IEEE/CAA J. Autom. Sinica*, vol. 9, no. 1, pp. 135–145, Jan. 2022.

Y. X. Yang, Z. H. Ni, and M. Y. Gao are with the School of Electronics and Information, Hangzhou Dianzi University, Hangzhou, and also with Zhejiang Provincial Key Laboratory of Equipment Electronics, Hangzhou 310018, China (e-mail: yyx@hdu.edu.cn; nzh@hdu.edu.cn; mackgao@hdu.edu.cn).

J. Zhang is with the School of Computer Science, Faculty of Engineering, University of Sydney, Darlington, NSW 2006, Australia (e-mail: jing.zhang1@sydney.edu.au).

D. C. Tao is with JD Explore Academy, JD.com, Beijing 101111, China (e-mail: dacheng.tao@gmail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JAS.2021.1004255

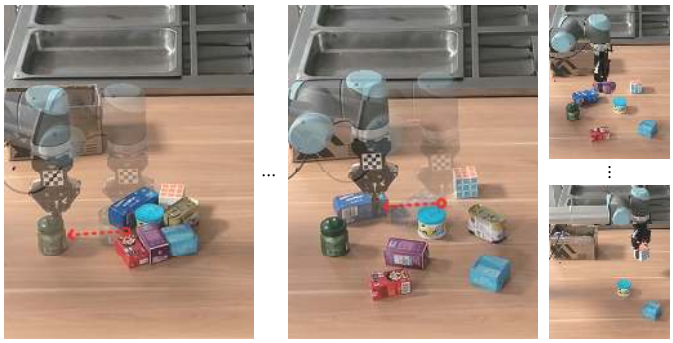
Traditional grasping methods are usually applied in a controlled environment with the known object model [3], which have limited the adaptability for different objects and scenarios. Recently, researchers apply deep learning and reinforcement learning into robotic tasks to improve the grasping success rate in various scenarios with different targets. For example, in [4]–[6], deep neural networks were used to predict the grasp point, angle, and jaw width from the input image. In [7]–[9], deep learning and reinforcement learning were combined for robotic grasping, which mapped the RGB-D image to specific action strategy and performed unsupervised learning to use the reward function. Although these methods can achieve grasping at a reasonable success rate, they struggle in handling tightly stacked objects since it is hard to find a suitable grasp point on an object and grasp it without causing collisions [10]. Therefore, how to design effective strategies to grasp tightly stacked objects remains challenging.

In practice, first pushing tightly stacked objects to a state of mutual separation can facilitate the subsequent grasping phase and significantly increase the success rate [11]. Therefore, how to model both tasks into a unified multi-task framework to enable collaborative pushing and grasping is a promising direction to solve the problem. Recently, some collaborative pushing and grasping methods [12]–[16] based on deep reinforcement learning have been proposed. Zeng *et al*. [12] proposed a deep Q-learning framework to tackle this task. However, its reward function only accounts for whether there should be a push action without evaluating the consequence of the push action, which affects the effectiveness of the pushing strategy. The pushing reward functions in [13]–[15] were defined using the image difference before and after the pushing action, while the validity of the pushing action was still not evaluated. Yang *et al*. [16] evaluated the pushing effect using the maximum Q value of local area around the push point before and after the pushing action. Since the evaluation only accounted for the consequence of pushing at a local area, it may result in predicting ineffective actions that achieve no gains from a global perspective, e.g., separating a small group of objects while some of them may be closer to the remaining objects. Indeed, how to design a pushing reward function to comprehensively evaluate the consequence of the pushing action remains under-explored.

Besides, these methods [12]–[16] mainly used toy blocks as representative objects in the experiments, which have simple

colors and shapes and lack of diversity and generality. Using simple objects during training may lead to a poor generalization capability when transferring to new scenarios, e.g., from the simulation environment to the real environment and from specific objects to unknown objects. Therefore, it is also very important to construct an object dataset containing objects in various shapes and colors to improve the generalization capability of the trained model.

To address these issues, we propose a novel collaborative pushing and grasping method based on deep Q-learning with an efficient non-maximum suppression policy (PolicyNMS), which can help to suppress unreasonable actions. Moreover, a novel pushing reward function based on convolutional neural networks called PR-Net is devised, which can comprehensively assess the degree of aggregation or separation between objects for each candidate pushing action from a global perspective, therefore helping the model to predict more effective pushing actions. Furthermore, we establish a dataset named CHID (common household items dataset) containing common household items in various colors and shapes and construct training scenarios from easy to difficult following the curriculum learning idea, which are beneficial to enhance the generalization capability of the collaborative pushing and grasping model. Experiments show that our method can efficiently accomplish the grasping task of tightly stacked objects via collaborative pushing and grasping and generalize well from simulation to real application and from specific objects to unknown objects as illustrated in Fig. 1. The proposed method has a wide range of applications like industrial parts sorting and household clutter sorting. The contributions of this study can be summarized as:



(a) Pushing (b) Grasping

Fig. 1. Illustration of the proposed method for collaborative pushing and grasping tightly stacked objects.

1) A novel model-free deep Q-learning method is proposed for grasping tightly stacked objects via collaborative pushing and grasping, where an efficient PolicyNMS is devised to suppress unreasonable actions.

2) A novel pushing reward function called PR-Net is devised to predict the global reward for each candidate pushing action by comprehensively assessing the degree of aggregation or separation between objects.

3) A common household item dataset with curriculum training scenarios from easy to difficult is established to train and evaluate the model. Experimental results demonstrate the generalization capability of our model.

The remainder of the paper is organized as follows. Section II reviews related work. In Section III, we present the details of the proposed method, including the PolicyNMS, the reward functions, and the proposed CHID dataset. The experimental results and analysis are presented in Section IV. Finally, we conclude the paper in Section V.

II. Related Work

A. Grasping Methods

Grasping is one of the most fundamental and interesting problems in robotics research. Recently, data-driven robotic grasping methods have achieved a lot of progress. 6D pose estimation methods [17], [18] were proposed to complete precise positioning of objects and achieved grasping. Kehl *et al*. [17] extended the popular object detection network SSD (single shot multibox detector) [19] for 6D pose estimation and achieved good results from a single RGB image. Wang *et al*. [18] proposed a DenseFusion network to extract RGB and depth features separately and fuse them to estimate precise 6D pose. But these methods all need the 3D model information of the target objects, which is difficult to acquire in many practical applications.

Differently from them, deep neural networks [4]–[6] were used to directly predict the grasp point, angle, and jaw width from the image, which can be well generalized to unknown objects and accomplish the grasping task. Mahler *et al*. [4] proposed a grasp quality convolutional neural network that predicts grasps location from synthetic point cloud data. Kumra *et al*. [6] proposed a generative residual convolutional neural network that uses n-channel input data to generate images that can be used to infer grasp rectangles for each pixel. Although good grasping performance has been achieved, these methods [4]–[6] based on supervised learning are limited to single grasp strategy and unable to achieve the coordination of different strategies throughout the task.

Reinforcement learning using long-term future views can help the agent to learn a more robust and comprehensive policy. In [8], [9], [20]–[22], deep reinforcement learning methods were proposed to model the grasping task and use reward functions to guide the grasp strategy for accomplishing the task, achieving good generalization performance. However, for densely stacked objects, grasping them directly will cause collisions between objects as well as collisions between the gripper and objects, resulting in failures. Differently from these methods, we propose a collaborative pushing and grasping method based on reinforcement learning, which first pushes the tightly stacked objects to separate them from each other and then grasps each object sequentially. In this way, our method can significantly improve the success rate.

B. Pushing Methods

Pushing is another fundamental task in robotics research [23]. Pushing to separate the tightly stacked objects can help improve the success rate of grasping. Actually, separating objects in close proximity is the prerequisite for many other

subsequent operations [24], such as object classification, object arrangement, and object stacking. Deep learning methods are widely applied in robotic pushing problems [25], [26]. Katz *et al*. [25] presented an interactive segmentation algorithm to push cluttered objects. Similarly, Eitel *et al*. [26] also applied an object segmentation algorithm and then generated a series of push action sets based on the segmentation results. However, the segmentation results may be incorrect, especially for unknown objects, which greatly affect the robustness of these segmentation-based methods.

Reinforcement learning based pushing also attracts increasing attention in recent years [27], [28]. Andrychowicz *et al*. [27] proposed the hindsight experience replay method to train the policy for those robotic tasks like pushing from the sparse and binary reward. But their environments were pretty simple, where only one object was needed to be pushed in the workspace. Kiatos *et al*. [28] designed a pushing method to separate a target object from the cluttered environment based on reinforcement learning. However, it is designed to separate single specific target rather than all generic objects in the complex environment. Differently from these methods, we focus on obtaining the suitable push sequence to separate all the objects in dense clutter, which is essential to improve the success rate of subsequent grasping.

C. Multi-task Learning

Recently, researchers focused on multi-task learning of collaborative pushing and grasping [12]–[16] based on deep reinforcement learning. In [12], a deep Q-learning framework was proposed to address this task. However, the pushing reward function in [12] only assessed whether the object was pushed, rather than evaluating the effectiveness of the push action. Hence, this method [12] may result in pushing the whole objects in a certain direction. The pushing reward functions in [13]–[15] were defined using the scene image difference before and after the pushing action, while the effectiveness of the pushing action was still not assessed. The reward function in [16] evaluated the pushing consequence by comparing the Q value around the push point before and after the pushing action. However, only evaluating the pushing effectiveness in a local area may result in non-optimal pushing action from a global perspective, e.g., separating a small group of objects while some of them may be closer to the remaining objects. Besides, these methods mainly used toy blocks during training and testing, which have simple colors and shapes and lack of diversity and generality. Using simple training objects may lead to a poor generalization capability when transferring to new scenarios, e.g., from the simulation environment to the real environment and from specific objects to unknown objects.

By contrast, we establish a CHID dataset containing common household items in various shapes and colors, which can be used to improve the generalization capability of the trained model. In addition, following the multi-task learning idea, we design a novel collaborative pushing and grasping method based on deep Q-learning, where an efficient non-maximum suppression policy is designed to suppress unreasonable actions. Furthermore, we propose a new data-driven pushing reward network that can comprehensively assess the degree of separation and aggregation between objects from a global view rather than the local neighborhood based assessment in previous method [16].

III. The Proposed Method

Pushing and grasping objects using a robotic arm can be expressed as a Markov decision process (MDP) [29], [30]. MDP is commonly represented by a quaternion (S, A, P, R), where S denotes the state space, A denotes the action space, P denotes the transition probability, and R denotes the reward function. The value-based reinforcement learning (RL) method can effectively deal with the MDP problem. Among them, deep Q-learning (DQN) methods [31]–[33] aim to obtain an end-to-end mapping function $Q(S, A; \theta)$ from state space S to action space A by learning the network parameters θ, which have demonstrated good performance and great potential in the field of robotics. In this paper, a novel collaborative pushing and grasping framework based on DQN is proposed for automatically pushing and grasping tightly stacked objects. As shown in Fig. 2, the proposed framework consists of a pushing network (Action-PNet) and a grasping network (Action-GNet), which follows the idea of first separating the cluttered objects by pushing and then grasping them one-by-one.

A. Collaborative Pushing and Grasping Network

The action space includes four components: action type $\varnothing = \{push, grasp\}$, locations (x, y, z), rotation angle Θ, and push length L. During pushing, we set $\Delta\Theta = 22.5°$ to indicate the interval of pushing directions in a range of 360°, i.e., a total of 16 pushing directions. During grasping, we set $\Delta\Theta = 11.25°$ in a range of 180° to indicate the interval of grasping directions, i.e., a total of 16 grasping directions.

At time t, the state s_t is obtained from the RGB-D images. Specifically, we map color and depth images to the robotic arm coordinate system and obtain the color-state-map and the depth-state-map. As shown in Fig. 2, our Action-PNet and Action-GNet are built upon the 121-layer DenseNet [34] pre-trained on ImageNet [35] to extract features from the color-state-map and depth-state-map. After feature concatenation, two identical blocks with batch normalization (BN) [36], rectified linear unit (ReLU) [37], and 1×1 convolution are used in Action-PNet and Action-GNet for further feature embedding. Then, the bilinear interpolation layer is used to obtain the pixel-wise state-action prediction value $Q(s_t, a; \theta)$. Note that the pushing process switches to the grasping process according to the separation degree of objects, i.e., the pushing state-action prediction value will decrease to a low level when the objects are already separated from each other.

Moreover, efficient prior constraints are devised to reduce the complexity of action space and accelerate the training process. As shown in Fig. 2, we present the dynamic action mask $M(s_t, \varnothing)$ to optimize the action strategy $\pi_\theta^*(s_t)$

$$\pi_\theta^*(s_t) = \arg\max_{a \in A} (M(s_t, \varnothing) \times Q(s_t, a; \theta)) \tag{1}$$

where, $M(s_t, \varnothing)$ is obtained by the object contours for pushing

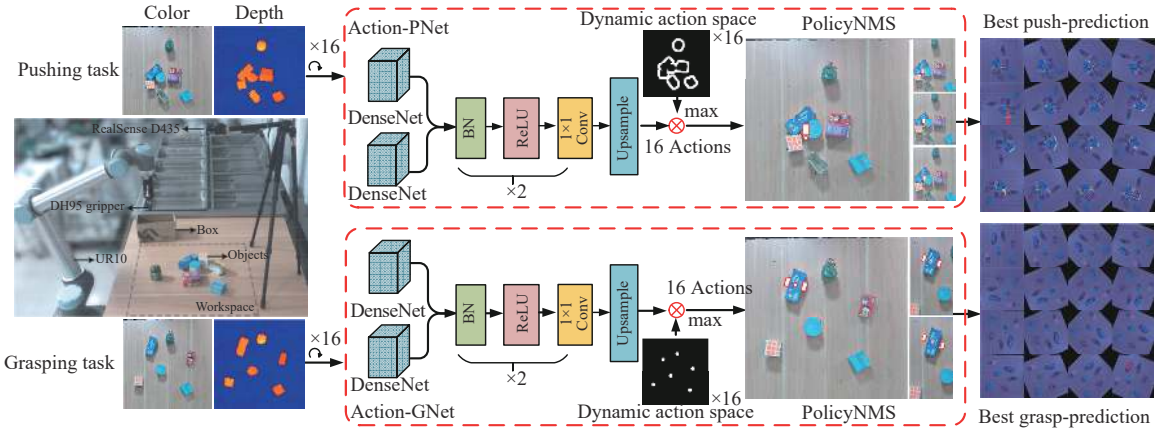
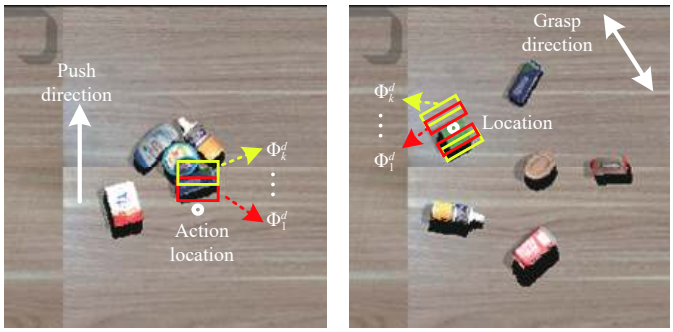


Fig. 2. Illustration of the proposed collaborative pushing and grasping method based on deep reinforcement learning.

actions, and $M(s_t, \varnothing)$ is obtained by the centers of object contours for grasping actions.

B. Non-maximum Suppression Policy (PolicyNMS)

Non-maximum suppression (NMS) algorithms [38], [39] are widely applied to deal with highly redundant candidate boxes in object detection tasks. Inspired by these NMS algorithms, we propose an efficient PolicyNMS to suppress unreasonable actions. Specifically, we construct redundant boxes on each candidate action and calculate the confidences (i.e., the object percentage) of redundant boxes to evaluate the reasonableness of an action as shown in Fig. 3.



(a) Pushing (b) Grasping

Fig. 3. Illustration of our PolicyNMS.

PolicyNMS aims to use a constraint $\pi_{\text{NMS}}(s_t)$ to suppress unreasonable actions and help to obtain the final action as

$$\pi^*_{\text{final}}(s_t) = \arg\max_{a \in A} \left(\pi^*_\theta(s_t) \times \pi_{\text{NMS}}(s_t) \right). \tag{2}$$

According to (1), we can obtain the action locations (x, y, z) and the corresponding state-action predictions in 16 action directions, respectively. As shown in Fig. 3, different shifts at the original action location are implemented along each action direction to obtain the boxes Φ^d_k, where $k \in [1, K]$ denotes different shifts in each direction, $d \in [0, 15]$ denotes 16 directions. For pushing and grasping, boxes of different shapes are designed as shown in Figs. 3(a) and (b). The probability P^d_k is defined as the percentage of objects in the box Φ^d_k. Then, the probabilities P^d_k corresponding to different shifts in the same direction are averaged to get the action probability P^d

$$P^d = \frac{1}{K} \sum_{k=1}^{K} P^d_k. \tag{3}$$

During pushing, a larger P^d represents a higher possibility of successfully pushing the object. For grasping, a smaller P^d means a larger grasping space for the gripper and a lower possibility of collision. Therefore, for 16 action directions we can obtain the constraint on unreasonable action $\pi_{\text{NMS}}(s_t)$

$$\pi_{\text{NMS}}(s_t) = \begin{cases} P^d, & \text{for pushing} \\ 1 - P^d, & \text{for grasping} \end{cases} \tag{4}$$

where $\pi_{\text{NMS}}(s_t)$ is a 16-dimensional vector.

By using such a constraint for unreasonable action suppression, our method can significantly improve the convergence speed and predict more reasonable actions.

C. Rewards for Pushing and Grasping

To better evaluate the quality of the action strategies, novel rewards for pushing and grasping are designed in this paper. As shown in Fig. 4, a convolutional neural network based pushing reward is designed to evaluate the separation or aggregation trend after pushing, called PR-Net.

Firstly, two sequential depth-state-maps $I_{\text{push\_before}}$, $I_{\text{push\_after}}$ of size $224 \times 224 \times 3$ are fed into two branches, respectively. In each branch, the VGG-16 (visual geometry group 16-layer) network is used as the backbone and outputs the $7 \times 7 \times 512$ feature maps. Then, the feature maps from both branches are concatenated to obtain the $7 \times 7 \times 1024$ fused feature maps I_{fusion}, which are fed into a convolution layer with a kernel of size 1×1, followed by a BN layer and an ReLU layer, i.e.,

$$I_{\text{cov1}} = \sigma\left(BN\left(\omega_{(512,(1,1))}(I_{\text{fusion}})\right)\right). \tag{5}$$

where $I_{\text{cov1}} \in \mathbb{R}^{512 \times 7 \times 7}$ denotes the output feature maps, and $\omega_{(512,(1,1))}$ denotes the learnable parameters, $BN(\cdot)$ denotes the batch normalization layer, and $\sigma(\cdot)$ denotes the ReLU activation layer.

Then, we obtain $I_{\text{cov2}} \in \mathbb{R}^{512 \times 5 \times 5}$ by feeding the output feature maps into another convolution layer with a kernel of size $3 \times 3 \times 512$, followed by a BN layer and an ReLU layer, i.e.,

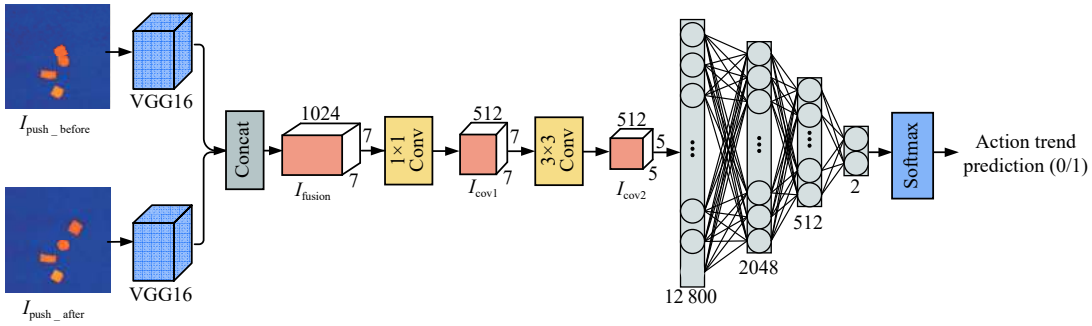


Fig. 4. The proposed PR-Net architecture.

$$I_{\text{cov2}} = \sigma(BN(\omega_{(512,(3,3))}(I_{\text{cov1}}))). \tag{6}$$

The feature maps I_{cov2} are flattened and fed into three fully connected layers. We use dropout to avoid overfitting and an ReLU layer as the activation function after the first two layers. The last fully connected layer is fed into a softmax layer to predict a probability vector for a binary classification task, i.e., whether or not the objects are separated further after a pushing action.

We use the cross-entropy loss to train the PR-Net

$$L_p = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} p_{ij} \ln\left(f^j_{\theta_{\text{PR-Net}}}\left(I^i_{\text{before}}, I^i_{\text{after}}\right)\right) \tag{7}$$

where I^i_{before} and I^i_{after} represent the ith input image pair of the PR-Net, $\theta_{\text{PR-Net}}$ represents the learnable parameters of the PR-Net, $f^j_{\theta_{\text{PR-Net}}}(\cdot)$ represents the mapping function of the PR-Net, p_{ij} represents the one-hot encoding vector of the ground truth label of the ith sample, C represents the number of classes, N represents the total number of samples in the training set, $L_p(\cdot)$ represents the loss function of the PR-Net.

Finally, a pushing reward r_p can be derived from the output of the PR-Net, which is defined as

$$r_p = \begin{cases} -0.5, & \text{if } output = 0 \\ 0.5, & \text{if } output = 1 \end{cases} \tag{8}$$

where $output = 0$ means that the push action aggregates the objects, and $output = 1$ means that the push action separates the objects.

PR-Net can efficiently predict the global reward for each candidate pushing action by assessing the degree of aggregation or separation from the full view of scene.

An efficient grasping reward function r_g is also designed

$$\begin{cases} r_g = G - \lambda\Delta\Theta \\ \Delta\Theta = |E_\Theta - O_\Theta| \in [0°, 90°] \end{cases} \tag{9}$$

where G denotes the grasp result, i.e., 0 for a failed grasp and 1.5 for a successful one. $\Delta\Theta$ denotes the angle constraint indicating the absolute difference between the rotation angle E_Θ of gripper and the angle O_Θ of the object, λ is a hyper-parameter, which is set to 0.02. The angle constraint $\Delta\Theta$ can help to obtain a more precise grasp policy, which will be discussed in Section IV.

D. The Common Household Item Dataset (CHID)

Differently from [12]–[16], [40], we use common household items as the targets in our pushing and grasping task. To this end, we establish a common household item dataset (CHID), which contains many different household items in various shapes, colors, textures, and sizes, i.e., a better collection of various generic objects in the household scenario. Specifically, we select the household object meshes from Freiburg spatial relations dataset [41] and 3D Warehouse Web[1]. We also set the physical properties for these objects, so that the dataset can simulate physical collision, friction, and other phenomena in the real world. The simulation items in the training set and testing set are shown in Figs. 5(a) and (b), each includes 15 kinds of objects. Note that the objects in the testing set are disjoint with those in the training set. The real-world testing items for testing are presented in Fig. 5(c), which are also disjoint with the simulation items in the training set.

(a) (b) (c)

Fig. 5. Some items from the CHID dataset. (a) Simulation items in the training set; (b) Simulation items in the testing set; (c) Real-world items for testing.

Then, we randomly select $n \in [3,6]$ objects in the training set to build training scenarios and randomly select $n \in [3,8]$ objects in the testing set to build testing scenarios. To learn an effective pushing strategy to separate objects, we set two difficulty levels during training by following the idea of curriculum learning. As shown in Fig. 6(a), there are clear gaps between objects in the easy scenarios, which are used for the initial stage of training. As shown in Fig. 6(b), objects in the difficult scenarios are packed tightly, which are used for the later stage of training. Training from easy to difficult is beneficial for speeding up the convergence and learning a robust pushing strategy. As shown in Fig. 6(c), the real-world testing scenarios are very different from the simulation ones, which are used to evaluate the generalization capability of the proposed method.

Finally, we establish the training and testing sets for our PR-Net as shown in Fig. 7, including 31 628 training pairs and

[1] https://3dwarehouse.sketchup.com

Fig. 6. Training and testing scenarios. (a) Easy scenarios for the initial stage of training; (b) Difficult scenarios for later stage of training; (c) The real-world testing scenarios.

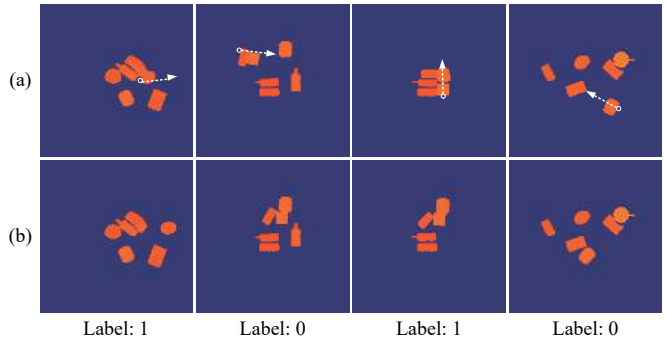


Fig. 7. Some training and testing pairs for our PR-Net: (a) Depthstate-maps before a pushing action; (b) Depth-state-maps after the action. The bottom labels denote aggregation (0) or separation (1). The white arrows indicate the pushing directions.

7907 testing pairs. Each pair contains the depth-state-map before a pushing action and the depth-state-map after the pushing action, as well as the ground truth label, i.e., 0 means aggregation while 1 means separation.

IV. Experimental Results

In the experiments, we evaluated the proposed method in both the simulation and real-world environments. First, we compared our method with the Non-RL pushing method and directly grasping method to verify the performance of the proposed method. Then, we performed the ablation studies to validate the effectiveness of the proposed PolicyNMS and PR-Net. Finally, we demonstrated the generalization capability of the proposed method from the simulation testing scenarios to real-world scenarios.

A. Implementation Details

We built a simulation environment in Gazebo, including a UR10 robotic arm equipped with a robotiq85 gripper and a Kinect RGB-D camera fixed on the table. We trained our RL method using stochastic gradient descent (SGD) with a fixed learning rate of 0.0001, momentum of 0.95, and weight decay of 2E–5 on an Ubuntu 16.04 server with two NVIDIA GTX 1080Ti GPUs. We applied DQN [33] with a prioritized experience replay [42] to train our Action-PNet and Action-GNet for 20 000 steps and 8000 steps, respectively. It took about 15 s for each step. And we updated the parameters of the target network in every 200 steps. ε-greedy [31] was used as the action selecting policy, where ε was initialized as 0.4 and then annealed to 0.1 during training. The future discount factor γ was set as 0.5. For PR-Net, we used the VGG-16 network pretrained on ImageNet [35] as the backbone and trained it for 60 epochs using SGD with a fixed learning rate of 0.0001, momentum of 0.95, and weight decay of 2E–5. The batch size was set to 32. Horizontal and vertical flipping was used for data augmentation during training.

B. Dataset and Evaluation Metrics

We adopted the established dataset CHID described in Section III-D as the benchmark. Specifically, we used the training set including the easy scenarios and difficult scenarios to train the proposed model and tested it on the simulation testing scenarios as well as the real-world scenarios. For each number of objects ($n \in [3,8]$), we conducted 25 tests, respectively. The performances of different methods were evaluated in terms of the following metrics: 1) success rate of separation (pushing times $\leq 2\times n$), where n represents the number of objects to be separated. If the pushing times in one test exceed $2\times n$, this test is regarded as failure; 2) pushing efficiency metric, i.e., the mean and standard variance of pushing times in 25 tests at different settings ($n \in [3,8]$). The smaller this mean value is, the more effective the current method is. Besides, a smaller variance indicates a more robust pushing strategy; and 3) success rate of grasping, which is defined as the average ratio between the number of objects and the total grasping times in 25 tests at different settings.

C. Pushing and Grasping Results in Simulation Scenarios

First, we compared our RL-based pushing method with the supervised learning method (named as Non-RL pushing), which has the same structure with the Action-PNet but is trained in a supervised manner, where the binary classification labels are predicted by PR-Net. For each number of objects ($n \in [3,8]$), we conducted 25 scenarios, i.e., randomly selected the corresponding number of objects from the testing set and tightly stacked them together for each scenario. The success rate of separation and pushing efficiency metrics are reported in Table I. Compared with the Non-RL pushing method, the performance of our method is much better. As the number of objects increases, the difficulty of the pushing task becomes higher, and the advantage of our method becomes more and more obvious. Besides, the less pushing times demonstrate that the proposed RL based pushing method using long-term future rewards separates objects more effectively while the smaller variance shows its robustness. Although only $n \in [3,6]$ objects were used during training, the proposed method still obtained high performance for pushing more tightly stacked objects (e.g., $n \in [7,8]$) during testing, which demonstrates the generalization capability of our method. In addition, we replaced the proposed PR-Net reward with the local reward

TABLE I
COMPARISON WITH OTHER PUSHING METHODS

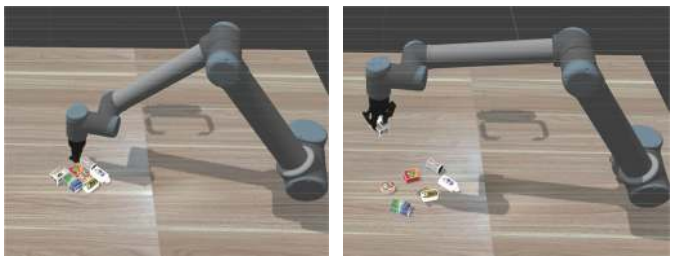
Number of objects (n)	3	4	5	6	7	8
	Success rate of separation (pushing times $\leq 2\times n$) (%)					
Non-RL pushing	100	96	96	96	84	60
Local-reward RL [16]	100	92	88	84	84	76
The Proposed method	100	100	100	100	100	100
	Mean and standard variance of pushing times					
Non-RL pushing	3.20±0.71	4.80±2.10	6.88±2.19	7.84±2.13	10.67±3.18	14.40±3.22
Local-reward RL [16]	3.88±1.01	5.80±1.71	7.84±1.89	9.32±2.78	11.40±3.00	14.32±3.73
The Proposed method	2.80±0.71	3.59±0.85	5.72±1.34	6.68±1.28	8.21±1.31	10.90±1.53

TABLE II
COMPARISON WITH THE GRASPING-ONLY METHOD

Number of objects (n)	3	4	5	6	7	8
	Success rate of grasping (%)					
Grasping-only method	32.75	32.26	34.63	30.26	28.59	25.81
The Proposed method	98.68	97.09	100	97.40	98.31	97.56

function of a recently proposed RL-based collaborative pushing and grasping method [16] and constructed comparative experiments. As shown in Table I, the proposed method can obtain significant advantages over local reward RL pushing [16]. It is because that only evaluating the pushing effectiveness in a local area may result in non-optimal pushing action from a global perspective, e.g., separating a small group of objects while some of them may be closer to the remaining objects, which well demonstrates the superior performance of our designed PR-Net reward.

Then, we conducted experiments to compare grasping-only method and the proposed collaborative pushing and grasping method for grasping tightly stacked objects. The grasping-only method has the same structure with our Action-GNet and was used for directly grasping objects without pushing. As shown in Table II, the success rate of grasping of the grasping-only method is very low. It is because directly grasping the tightly stacked objects will cause collisions and result in failures. By contrast, the proposed method can achieve a much higher success rate, which demonstrates the superiority of our collaborative pushing and grasping framework over the grasping-only one. The simulation testing environment is presented in Fig. 8.



(a) Pushing (b) Grasping

Fig. 8. Illustration of the simulation testing environment.

D. Ablation Study

Ablation studies of the components of the proposed method were performed to validate their effectiveness. First, experiments were conducted to verify the performance of the pushing reward network PR-Net and the PolicyNMS in the pushing task. Specifically, we conducted experiments for the following three models.

Model 1: the push reward without PR-Net, i.e., only getting the final reward when the separation is done, and Action-PNet without PolicyNMS.

Model 2: the push reward using PR-Net and Action-PNet without PolicyNMS.

Model 3: the push reward using PR-Net and Action-PNet using PolicyNMS. The training results of these three models for the setting of 6 objects are plotted in Fig. 9. It can be seen that the proposed PR-Net can help the method achieve better pushing efficiency by adequately evaluating the rationality of pushing actions and the proposed PolicyNMS contributes significantly to the faster learning speed by suppressing unreasonable pushing actions.

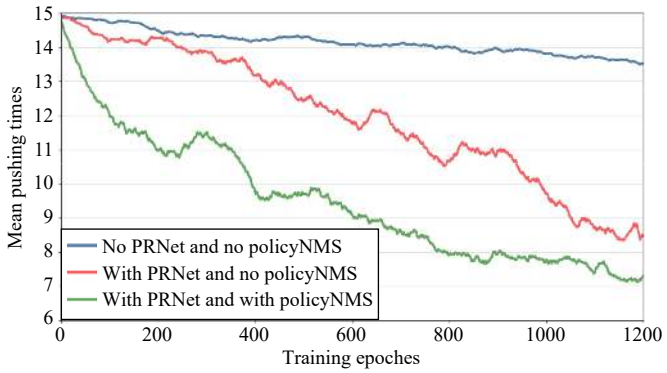


Fig. 9. Ablation study of PR-Net and PolicyNMS for pushing.

Then, to evaluate the effectiveness of the proposed grasping angle reward function defined in (9) and PolicyNMS in the

grasping task, we conducted experiments for the following three models.

Model 1: the grasping reward without the angle constraint defined in (9) and the Action-GNet without PolicyNMS.

Model 2: the grasping reward with the angle constraint and the Action-GNet without PolicyNMS.

Model 3: the grasp reward with the angle constraint and the Action-GNet with PolicyNMS. The training results of these three models for the setting of 6 objects are plotted in Fig. 10. It can be seen that PolicyNMS and the grasping angle reward bring a higher success rate of grasping. Specifically, the grasping angle constraint reward benefits the final performance while contributing less to the learning speed. By contrast, PolicyNMS has a larger impact on the learning speed, which helps the agent learn much faster by suppressing unreasonable grasping actions to avoid failure grasping and collisions.

E. Evaluation Results in Real-world Scenarios

We evaluated the proposed method in real-world scenarios. The testing suit consists of a UR10 robotic arm with a DH-95 gripper, and a Realsense RGB-D camera fixed on the desktop as shown in Fig. 2. We used the network trained in the simulation training scenarios directly to the real-world testing scenarios. Specifically, we randomly selected $n \in [3,8]$ real-world household objects and tightly stacked them together. In all the real-world tests, our method successfully separated all the objects under the push times limitation, i.e., $\leq 2\times n$. An visual demo of pushing and grasping in the real-world environment is presented in Fig. 11. As shown in the first column of Table III, our method can achieve a robust and efficient pushing performance in the real-world tests, which are comparable to the results in the simulation tests as shown

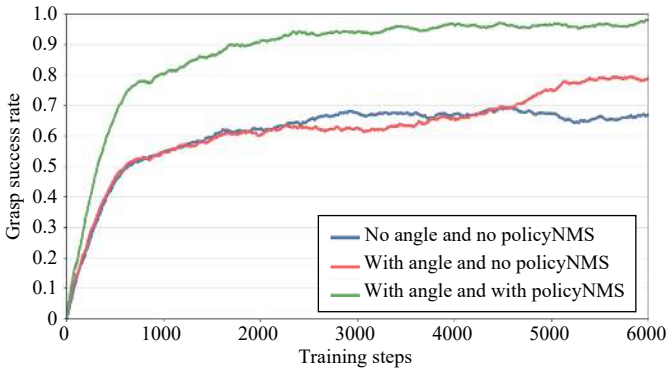


Fig. 10. Ablation study of the proposed grasping angle reward and PolicyNMS for grasping.

TABLE III
RESULTS OF OUR METHOD IN REAL-WORLD ENVIRONMENT

Number of objects	Mean and standard variance of push times	Grasp success rate (%)
3	2.40±0.96	97.40
4	3.68±1.28	97.09
5	5.80±1.68	96.90
6	6.67±1.76	96.77
7	8.26±1.83	97.22
8	10.47±1.81	96.15

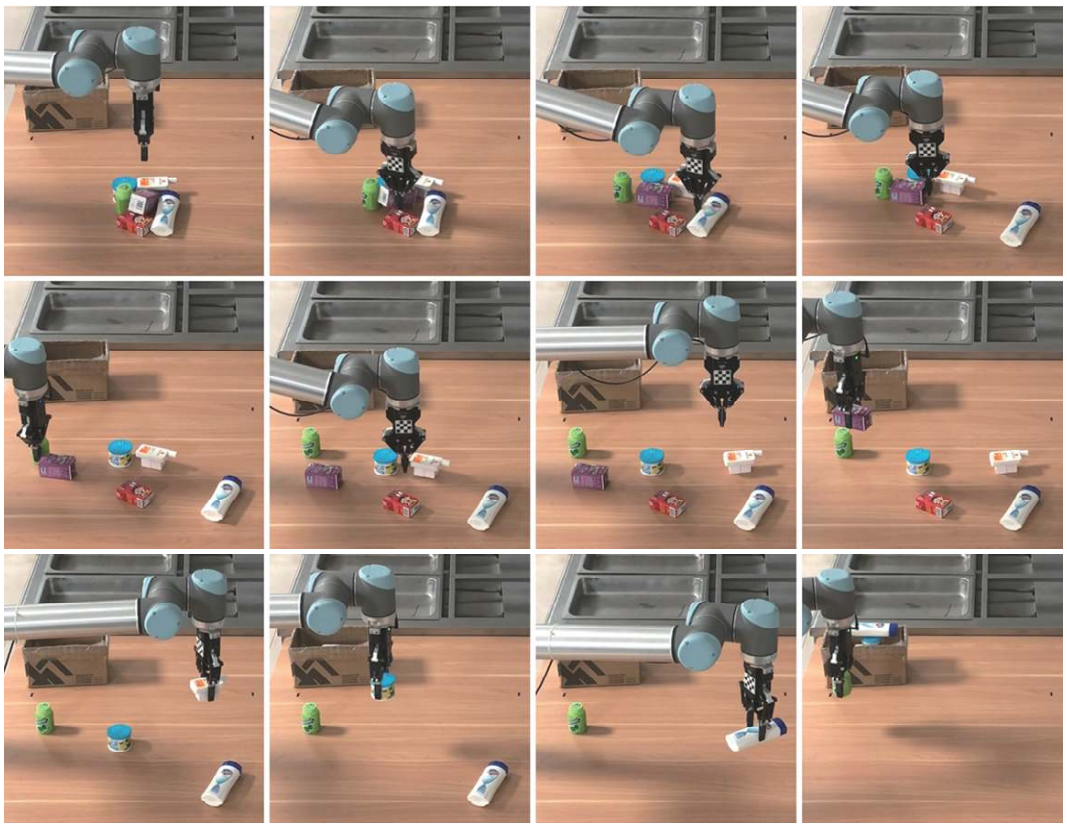


Fig. 11. Testing for random objects stacked tightly in real-world environment.

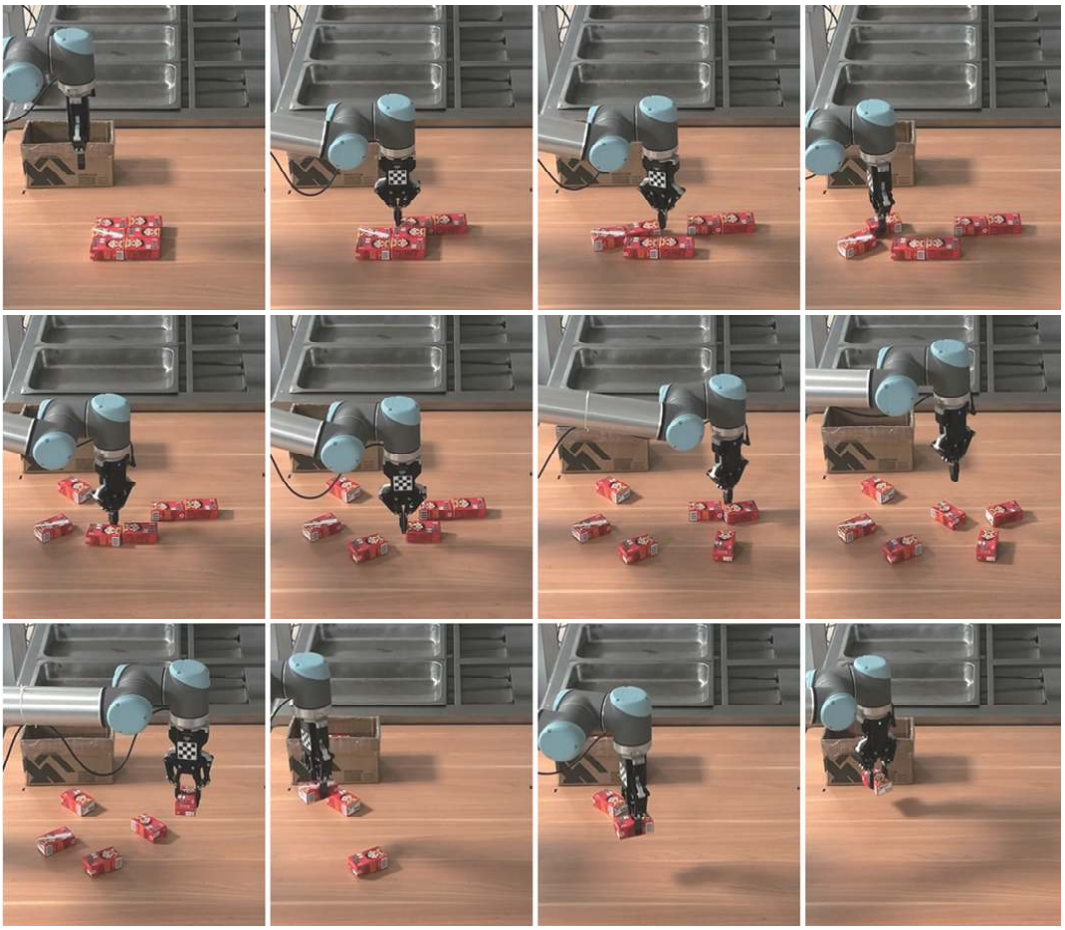


Fig. 12. Testing for identical objects stacked tightly in real-world environment.

in Table I. Besides, the high success rates of grasping are also comparable to the results in Table II. The results validate the good generalization capability of the proposed method from simulation environment to real-world environment as well as from specific objects to unknown objects. Furthermore, we also prepared much more difficult testing scenarios, where $n \in [3, 8]$ identical objects are tightly stacked together as shown in Fig. 12. These tests further demonstrate the excellent generalization capability and adaptability of our method, which is important for practical applications. A video demo of the testings in the real-world environment is also provided[2].

V. Conclusions

In the paper, we propose a novel deep Q-learning method for collaboratively pushing and grasping tightly stacked objects. Specifically, a novel efficient non-maximum suppression policy is designed, which can help accelerate the learning speed by suppressing unreasonable actions to avoid bad consequences. For the pushing task, an end-to-end data-driven pushing reward network is designed to assess the state of aggregation or separation after different pushing actions from a global perspective. For the grasping task, an efficient grasping reward function with angel constraint is defined to help optimize the angle of grasping actions. They contribute to developing an efficient and robust pushing strategy as well as the high success rates of pushing and grasping. Moreover, we establish the common household item dataset containing various objects in different colors, shapes, textures, and sizes, forming lots of easy to difficult training scenarios. Experimental results demonstrate the superiority of the proposed method over the non-RL pushing method and directly grasping method for this challenging task, as well as its fast learning speed, good generalization capability and robustness. One of the limitation of the method is that there is no constraint of the pushing distance, which may push some objects out of the boundary. In the future work, we can explore an effective constraint to deal with this limitation.

[2] https://github.com/nizhihao/Collaborative-Pushing-Grasping

References

[1] A. Rakshit, A. Konar, and A. K. Nagar, "A hybrid brain-computer interface for closed-loop position control of a robot arm," *IEEE/CAA J. Autom. Sinica*, vol. 7, no. 5, pp. 1344–1360, Sep. 2020.

[2] J. Zhang and D. C. Tao, "Empowering things with intelligence: A survey of the progress, challenges, and opportunities in artificial intelligence of things," *IEEE Int. Things J.*, vol. 8, no. 10, pp. 7789–7817, May 2021.

[3] A. Bicchi and V. Kumar, "Robotic grasping and contact: A review," in *Proc. IEEE Int. Conf. Robotics and Autom.*, San Francisco, CA, USA, 2000, pp. 348–353.

[4] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Y. 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," 2017. [Online]. Available: https://arxiv.org/abs/1703.09312.

[5] D. Morrison, P. Corke, and J. Leitner, "Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach," 2018. [Online]. Available: https://arxiv.org/abs/1804.05172.

[6] S. Kumra, S. Joshi, and F. "Sahin, Antipodal robotic grasping using generative residual convolutional neural network," 2021. [Online]. Available: arXiv: https://arxiv.org/abs/1909.04810.

[7] I. Popov, N. Heess, T. Lillicrap, R. Hafner, G. Barth-Maron, M. Vecerik, T. Lampe, Y. Tassa, T. Erez, and M. Riedmiller, "Data-efficient deep reinforcement learning for dexterous manipulation," 2017. [Online]. Available: http://export.arxiv.org/abs/1704.03073.

[8] D. Quillen, E. Jang, O. Nachum, C. Finn, J. Ibarz, and S. Levine, "Deep reinforcement learning for vision-based robotic grasping: A simulated comparative evaluation of off-policy methods," in *Proc. IEEE Int. Conf. Robotics and Autom. (ICRA)*, Brisbane, QLD, Australia, 2018, pp. 6284–6291.

[9] M. Breyer, F. Furrer, T. Novkovic, R. Siegwart, and J. Nieto, "Comparing task simplifications to learn closed-loop object picking using deep reinforcement learning," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1549–1556, Apr. 2019.

[10] U. Viereck, A. ten Pas, K. Saenko, and R. Platt, "Learning a visuomotor controller for real world robotic grasping using simulated depth images," in *Proc. 1st Conf. Robot Learning*, Mountain View, United States, 2017, pp. 291–300.

[11] M. R. Dogar and S. S. Srinivasa, "A planning framework for nonprehensile manipulation under clutter and uncertainty," *Auton. Robot.*, vol. 33, no. 3, pp. 217–236, Oct. 2012.

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

[13] A. Hundt, B. Killeen, N. Greene, H. T. Wu, H. Kwon, C. Paxton, and G. D. Hager, ""Good robot!": Efficient reinforcement learning for multi-step visual tasks with SIM to real transfer," *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 6724–6731, Oct. 2020.

[14] B. Tang, M. Corsaro, G. Konidaris, S. Nikolaidis, and S. Tellex, "Learning collaborative pushing and grasping policies in dense clutter," in *Proc. IEEE Int. Conf. Robotics and Autom. (ICRA)*, Xi'an, China, 2021.

[15] G. Peng, J. H. Liao, and S. B. Guan, "A pushing-grasping collaborative method based on deep q-network algorithm in dual perspectives," 2021. [Online]. Available: https://arxiv.org/abs/2101.00829v1.

[16] Z. P. Yang and H. L. Shang, "Robotic pushing and grasping knowledge learning via attention deep q-learning network," in *Proc. Int. Conf. Knowledge Science, Engineering and Management*, Hangzhou, China, 2020, pp. 223–234.

[17] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab, "SSD-6D: Making RGB-based 3D detection and 6D pose estimation great again," in *Proc. IEEE Int. Conf. Computer Vision*, Venice, Italy, 2017, pp. 1530–1538.

[18] C. Wang, D. F. Xu, Y. K. Zhu, R. Martín-Martín, C. W. Lu, F. F. Li, and S. Savarese, "DenseFusion: 6D object pose estimation by iterative dense fusion," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Long Beach, CA, USA, 2019, pp. 3338–3347.

[19] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. European Conf. Computer Vision*, Amsterdam, The Netherlands, 2016, pp. 21–37.

[20] S. R. Song, A. Zeng, J. Lee, and T. Funkhouser, "Grasping in the wild: Learning 6DoF closed-loop grasping from low-cost demonstrations," *IEEE Robot. Autom. Lett.*, vol. 5, no. 3, pp. 4978–4985, Jul. 2020.

[21] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, and S. Levine, "Scalable deep reinforcement learning for vision-based robotic manipulation," in *Proc. 2nd Conf. Robot Learning*, Zürich, Switzerland, 2018, pp. 651–673.

[22] A. Ghadirzadeh, A. Maki, D. Kragic, and M. Björkman, "Deep predictive policy training using reinforcement learning," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, Vancouver, BC, Canada, 2017, pp. 2351–2358.

[23] C. Finn and S. Levine, "Deep visual foresight for planning robot motion," in *Proc. IEEE Int. Conf. Robotics and Autom. (ICRA)*, Singapore, 2017, pp. 2786–2793.

[24] M. Gupta, J. Müller, and G. S. Sukhatme, "Using manipulation primitives for object sorting in cluttered environments," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 2, pp. 608–614, Apr. 2015.

[25] D. Katz, A. Venkatraman, M. Kazemi, J. A. Bagnell, and A. Stentz, "Perceiving, learning, and exploiting object affordances for autonomous pile manipulation," *Auton. Robot.*, vol. 37, no. 4, pp. 369–382, Dec. 2014.

[26] A. Eitel, N. Hauff, and W. Burgard, "Learning to singulate objects using a push proposal network," in *Robotics Research*, N. M. Amato, G. Hager, S. Thomas, and M. Torres-Torriti, Eds. Puerto Varas, Chile: Springer, 2020, pp. 405–419.

[27] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, P. Abbeel, and W. Zaremba, "Hindsight experience replay," 2018. [Online]. Available: https://arxiv.org/pdf/1707.01495.pdf.

[28] M. Kiatos and S. Malassiotis, "Robust object grasping in clutter via singulation," in *Proc. Int. Conf. Robotics and Autom. (ICRA)*, Montreal, QC, Canada, 2019, pp. 1596–1600.

[29] D. P. Bertsekas, "Feature-based aggregation and deep reinforcement learning: A survey and some new implementations," *IEEE/CAA J. Autom. Sinica*, vol. 6, no. 1, pp. 1–31, Jan. 2019.

[30] L. Jiang, H. Y. Huang, and Z. H. Ding, "Path planning for intelligent robots based on deep Q-learning with experience replay and heuristic knowledge," *IEEE/CAA J. Autom. Sinica*, vol. 7, no. 4, pp. 1179–1189, Jul. 2020.

[31] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," 2013. [Online]. Available: https://arxiv.org/abs/1312.5602v1.

[32] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.

[33] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Proc. 13th AAAI Conf. Artificial Intelligence*, Phoenix, Arizona, 2016, 2094–2100.

[34] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Honolulu, HI, USA, 2017, pp. 2261–2269.

[35] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and F. F. Li, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Miami, FL, USA, 2009, pp. 248–255.

[36] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Machine Learning*, Lille, France, 2015, pp. 448–456.

[37] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Machine Learning*, Haifa, Israel, 2010, pp. 807–814.

[38] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, "Soft-NMS-improving object detection with one line of code," in *Proc. IEEE Int. Conf. Computer Vision*, Venice, Italy, 2017, pp. 5562–5570.

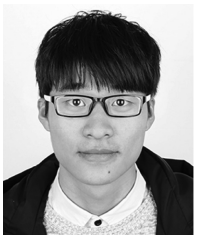
[39] J. Zhang, Z. Chen, and D. C. Tao, "Towards high performance human keypoint detection," *Int. J. Comput. Vis.*, vol. 129, no. 9, pp. 2639–2662, Sep. 2021.

[40] J. H. Zhang, W. Zhang, R. Song, L. Ma, and Y. B. Li, "Grasp for stacking via deep reinforcement learning," in *Proc. IEEE Int. Conf. Robotics and Autom. (ICRA)*, Paris, France, 2020, pp. 2543–2549.

[41] O. Mees, N. Abdo, M. Mazuran, and W. Burgard, "Metric learning for generalizing spatial relations to new objects," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, Vancouver, BC, Canada, 2017, pp. 3175–3182.

[42] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," 2016. [Online]. Available: https://arxiv.org/abs/1511.05952.

Yuxiang Yang received the B.S. and Ph.D. degrees in control science and engineering from University of Science and Technology of China in 2008 and 2013, respectively. He joined Hangzhou Dianzi University, in 2013, where he is currently an Associate Professor at the School of Electronic and Information. His research interests include computer vision and deep learning. In particular, his current research focuses on industrial automation and mobile robot. He has published more than 30 papers on journals such as *IEEE Transactions on Industrial Electronics* (*IEEE T-IE*), *KBS*, and conferences such as ICME, AIM, IECON.



Zhihao Ni received the B.S. degree in electronic information engineering from Hangzhou Dianzi University, in 2019. Now, he is an M.S. student at the School of Electronic and Information, Hangzhou Dianzi University. His research interests include deep reinforcement learning and robotics. He participated in the development of multiple industrial automation projects. His current focus is on deep reinforcement learning, robotics autonomous grasping.

Mingyu Gao received the M.S. degree in power electronics from Zhejiang University, in 1993, and received the Ph.D. degree in information and communication engineering from Wuhan University of Technology, in 2013. He joined Hangzhou Dianzi University, in 2001, where he is currently a Professor at the School of Electronic and Information, and the Inaugural Director of the Zhejiang Provincial Key Laboratory of Equipment Electronics. His research interests include industrial electronics and vehicle electronics.

Jing Zhang (Member, IEEE) is a research fellow at the School of Computer Science of the University of Sydney. His research interests include computer vision and deep learning. He has published more than 30 papers on prestigious conferences such as CVPR, ICCV, NeurIPS, and journals such as *IJCV, IEEE T-IP*. He serves as a Reviewer for many journals and conferences. He is a senior program committee member of the AAAI Conference on Artificial Intelligence and the International Joint Conference on Artificial Intelligence.

Dacheng Tao (Fellow, IEEE) is currently the Inaugural President of JD Explore Academy and a Senior Vice President at JD.com. He mainly applies statistics and mathematics to artificial intelligence and data science, and his research is detailed in one monograph and over 200 publications in prestigious journals and proceedings at leading conferences. He is a Fellow of the Australian Academy of Science, AAAS, and ACM. He received the 2015 Australian Scopus-Eureka Prize, the 2018 IEEE ICDM Research Contributions Award, and the 2021 IEEE Computer Society McCluskey Technical Achievement Award.