Learning to Group and Grasp Multiple Objects

Takahiro Yonemaru, Weiwei Wan*, Tatsuki Nishimura, Kensuke Harada

***Abstract*—Simultaneously grasping and delivering multiple objects can significantly enhance robotic work efficiency and has been a key research focus for decades. The primary challenge lies in determining how to push objects, group them, and execute simultaneous grasping for respective groups while considering object distribution and the hardware constraints of the robot. Traditional rule-based methods struggle to flexibly adapt to diverse scenarios. To address this challenge, this paper proposes an imitation learning-based approach. We collect a series of expert demonstrations through teleoperation and train a diffusion policy network, enabling the robot to dynamically generate action sequences for pushing, grouping, and grasping, thereby facilitating efficient multi-object grasping and delivery. We conducted experiments to evaluate the method under different training dataset sizes, varying object quantities, and real-world object scenarios. The results demonstrate that the proposed approach can effectively and adaptively generate multi-object grouping and grasping strategies. With the support of more training data, imitation learning is expected to be an effective approach for solving the multi-object grasping problem.**

***Index Terms*—Multi-Object Grasping, Imitation Learning**

I. Introduction

THIS study focuses on enabling robots to perform simultaneous multi-object grasping while considering pushing-based grouping. Previous studies on simultaneous multi-object grasping have identified several challenges. For example, Sakamoto et al. [1] proposed a rule-based method to push objects closer together before grasping them simultaneously. However, the method cannot handle complex pushing actions, and the combinations of objects that can be grasped together are limited. Agboh et al. [2] studied identifying near objects arranged on a plane and grasping multiple objects by pushing them together as the gripper closes. While effective under specific conditions, the method assumes that objects are initially spaced within the gripper's maximum opening width. Or else, it degenerates into single-object grasping. These limitations highlight the need for more advanced strategies that can group objects through flexible pushing actions and select suitable object combinations for simultaneous grasping. Designing such strategies through hand-crafted rules, however, remains a challenging task.

In this context, we propose addressing these issues by using imitation learning and enabling robots to learn how humans decide which objects to group and grasp. By acquiring grasping strategies through imitation learning, the proposed method enables robots to automatically choose suitable object combinations and perform the necessary pushing actions to group them. The method can overcome the limitations of existing methods and improve the efficiency and versatility of robotic picking tasks. Fig. 1 illustrates the components and workflow of the proposed framework. In the data collection phase shown in Fig. 1(a), a teleoperation system is developed for human operators to control the robot using natural hand movements. The setup enables collecting data like robot joint angles, gripper opening and closing states, and images from the global and local cameras mounted in the environment and on the robot hand. In the learning phase shown in Fig. 1(b), the framework employs Diffusion Policy [3] to learn appropriate robot actions several steps ahead, based on the robot's state from previous steps. Finally, in the inference phase shown in Fig. 1(c), the framework uses the robot's gripper poses, opening states, and camera images observed from a few steps before to predict new gripper poses and states for subsequent steps, thereby generating pushing and simultaneous grasping motions for multiple objects.

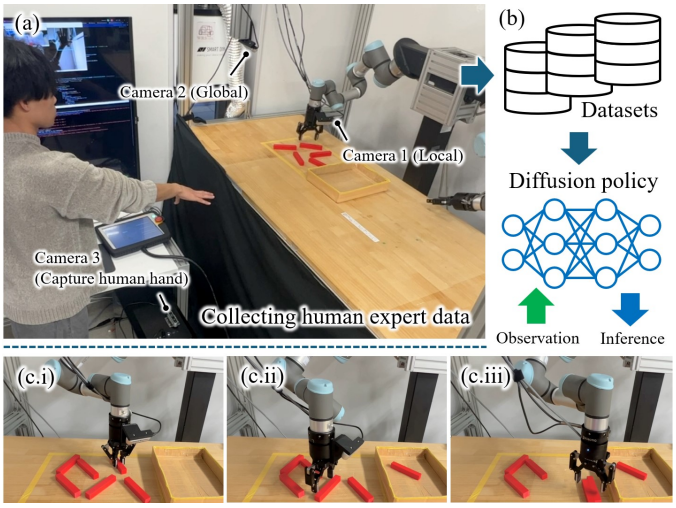

Fig. 1: (a) Collecting human expert data using teleoperation. (b) The collected dataset is used to train a diffusion policy network for inferring actions. (c) The policy network uses past observations to generate pushing or grasping actions.

Three groups of experiments were conducted to validate the proposed framework. The first group focused on comparing and analyzing the influence of different training data volumes and object quantities on learning and inference. The second group conducted fine-tuning of a model pre-trained on a dataset of simple objects to more practical scenarios and evaluated the generality of the proposed method to real-world objects. The third group evaluates the generalization ability of the learned policy across different object geometries. The experimental results validated the effectiveness of the proposed method and validated the adaptivity of fine-tuned models. Additionally, the results indicated that training a well-performing policy network requires a large amount of high-

All authors are from the Graduate School of Engineering Science, the University of Osaka, Japan. This work was supported by JSPS KAKENHI Grant Number JP22K12205. *Contact: wan.weiwei.es@osaka-u.ac.jp

quality data, with a positive correlation between data volume and learning performance. Developing efficient methods for collecting such data could potentially improve both learning and inference performance.

To the best of our knowledge, this is the first work to apply Diffusion Policy to choosing among pushing, grouping, and multi-object grasping. The proposed method is simple to implement and data-driven. We provide extensive experiments and in-depth analysis, and discuss both the strengths and limitations of the method. With larger-scale training data, imitation learning is expected to become an even more effective solution for multi-object grasping. We hope our findings and insights can inspire further research in this direction.

II. Related Work

Multi-object simultaneous grasping has been attracting significant attention for decades. As seminal work, Aiyama et al. [4] studied using two manipulators to cooperatively grasp and lift boxes. Harada and Kaneko [5][6] studied the neighboring equilibrium of multiple cylindrical objects and their enveloping grasps. Yoshikawa et al. [7] proposed the optimal power grasp condition for multiple objects from the viewpoint of reducing finger joint torques. Yamada et al. [8][9][10] analyzed the grasp stability of multiple objects by considering object surface curvatures and contact spring models.

More recent studies by Chen et al. [11] inserted the Barrett hand into a pile of objects and estimated the number of grasped items using a deep learning model. Shenoy et al. [12] focused on efficiently transferring the grasped objects to a separate container. Takahashi et al. [13] studied grasping a given amount of granular foods. Post-grasping methods were explored for fine adjustments [14]. Li et al. [15] developed a framework for planning and learning multi-object grasping (lifting). Jiang et al. [16] developed a vacuum gripper equipped with multiple suction cups and the related grasp planning methods to simultaneously pick multiple objects from a tray. Nguyen et al. [17] proposed the Wiring-Claw Gripper, which took advantage of the soft interaction between wires in a claw and objects for simultaneous multi-object grasping. Yao et al. [18] used a dexterous hand with progressive planning to exploit redundancy for multi-object grasping.

The above approaches could achieve stable grasping of multiple or unknown objects. However, they often assumed that the objects were initially positioned in proximity and were less applicable to scattered scenes. To tackle such limitations, several studies have explored using pushing actions to rearrange objects before simultaneous grasping. For example, Sakamoto et al. [1] proposed a method that can determine whether to grasp a single object, grasp two objects simultaneously, or push one object closer for simultaneous grasping based on the distance and friction between objects. Agboh et al. [2][19] proposed the μ-MOG framework to simultaneously grasp multiple rigid convex polygonal objects scattered on a plane. The method took advantage of a gripper's opening jaw to align objects before grasping. Shrey et al. [20] leveraged linear pushing actions along lines perpendicular to the gripper fingers to bring object centroids closer, allowing a robot to grasp multiple objects that initially lay beyond the gripper's opening width. Ye et al. [21] used weighted graphs to model object distributions and assess collision-free clusters, thereby enabling efficient and reliable multi-object picking. Kishore et al. [22] proposed a method for efficiently delivering cups, bowls, and utensils on tables by combining pushing and stacking actions. The method especially employed inward-pushing to reduce positional uncertainty.

The pushing-based methods have addressed the problem of scattered objects to some extent. However, they are predominantly rule-based, making it challenging to generate complex pushing trajectories for intricate grouping and grasping. Previously, reinforcement learning methods have attracted much attention in robotic pushing and grasping as they can adapt to various object arrangements. For instance, Zeng et al. [23] achieved efficient grasping by enabling the complementary relationship between pushing and grasping actions to be learned through self-supervised deep reinforcement learning. Chen et al. [24] proposed a grasping strategy that combines rule-based methods with reinforcement learning to integrate pushing and grasping. Wang et al. [25] attained high success rates and robustness by leveraging self-supervised deep reinforcement learning to learn an integrated model of pushing and grasping. These studies demonstrate that reinforcement learning has the advantage of adapting to unknown environments. However, a key challenge in reinforcement learning lies in the design of rewards. This issue becomes particularly pronounced in the context of this study as we use pushing to achieve grouping, followed by grasping. Designing rewards that facilitate rapid convergence for the pushing-grouping-grasping routine is notably difficult. A learning method conceptually similar to reinforcement learning but not dependent on reward design is imitation learning. Recent studies have showcased the potential of imitation learning in robotic manipulation applications. For example, ACT [26] has achieved high success rates in delicate tasks such as opening and closing Ziploc bags and inserting batteries into controllers. Diffusion policy [3] has enabled high success rates in tasks requiring diverse actions and significant flexibility, such as spreading sauce on pizza dough or relocating T-shape blocks and mugs placed at random positions on a plane to target positions and orientations. Given its advantages, this research adopts imitation learning as the primary methodology. By learning flexible pushing actions and grasping decisions from human demonstrations, our approach enables dynamic selection of object combinations, rearrangement through complex pushing trajectories, and adaptive grasping sequences that were previously difficult to attain with rule-based or reinforcement learning methods.

III. Data Collection Using Visual Teleoperation

Similar to AnyTeleop [27], we develop a teleoperation system for a robot by detecting the hand skeleton pose using an RGB-D camera. We use WiLoR [28] for detecting the hand skeleton pose. The output of WiLoR is a 3D skeleton described in the Σ_{wl} coordinate system shown in Fig. 2. It is then integrated with depth values of the same RGB-D camera to derive the 3D skeleton in the coordinate system of the depth

camera, Σ_{cam}. After that, we transform the 3D data into actions for the robot, thereby enabling teleoperation. The coordinates and transformations used in the process are as follows.

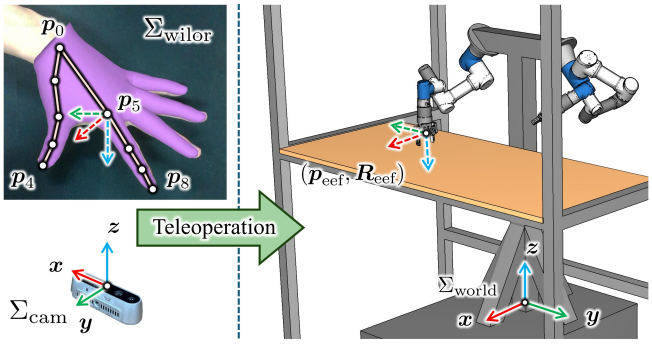

Fig. 2: Mapping relationship between hand poses detected by WiLoR and the end-effector of the target robot.

Jaw bottom of the robot gripper: The jaw bottom center of the robot's end-effector is defined as a position corresponding to the MCP joint of the index finger (p_5 in Fig. 2) and is transformed to the world coordinate system using

$$^{\Sigma_{\text{wd}}}\boldsymbol{p}_{\text{eef}} = {}^{\Sigma_{\text{wd}}}\mathbf{R}_{\Sigma_{\text{cam}}} \cdot \left({}^{\Sigma_{\text{cam}}}\mathbf{R}_{\Sigma_{\text{wl}}} \cdot {}^{\Sigma_{\text{wl}}}\boldsymbol{p}_5 + {}^{\Sigma_{\text{cam}}}\mathbf{p}_{\text{wl}}\right) + {}^{\Sigma_{\text{wd}}}\mathbf{p}_{\text{cam}}$$

Rotation of the robot gripper: The rotation of the end-effector, $^{\Sigma_{\text{wd}}}\boldsymbol{R}_{\text{eef}}$, is constrained such that its approaching direction is always vertically downward (the last column or $\boldsymbol{z}$ axis of $^{\Sigma_{\text{wd}}}\boldsymbol{R}_{\text{eef}}$ is parallel with and inversely aligns with the $\boldsymbol{z}$ axis of Σ_{world} in Fig. 2(a)). The rotation around the approaching direction is defined using the position of the thumb tip ($^{\Sigma_{\text{wl}}}\boldsymbol{p}_4$) and the position of the index finger tip ($^{\Sigma_{\text{wl}}}\boldsymbol{p}_8$) following equation (1).

$$^{\Sigma_{\text{wd}}}\boldsymbol{R}_{\text{eef}} = \begin{bmatrix} \hat{\boldsymbol{d}}_x & \hat{\boldsymbol{d}}_y & 0 \\ \hat{\boldsymbol{d}}_y & -\hat{\boldsymbol{d}}_x & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \hat{\boldsymbol{d}} = \frac{^{\Sigma_{\text{wl}}}\boldsymbol{p}_8 - {}^{\Sigma_{\text{wl}}}\boldsymbol{p}_4}{\left\| {}^{\Sigma_{\text{wl}}}\boldsymbol{p}_8 - {}^{\Sigma_{\text{wl}}}\boldsymbol{p}_4 \right\|}$$

Gripping state: The robot gripper's open/close state, g, is defined based on the distance between the thumb and the index finger following

$$g = \begin{cases} 0 & \text{if } d \geq \mu_{\text{up}} \\ 1 & \text{if } d < \mu_{\text{bot}} \end{cases}, \quad d = \frac{\left\| {}^{\Sigma_{\text{wl}}}\boldsymbol{p}_8 - {}^{\Sigma_{\text{wl}}}\boldsymbol{p}_4 \right\|}{\max \left\| {}^{\Sigma_{\text{wl}}}\boldsymbol{p}_8 - {}^{\Sigma_{\text{wl}}}\boldsymbol{p}_4 \right\|}$$

Here, $g = 0$ denotes the gripper in an open state, while $g = 1$ indicates the gripper in a closed state. The threshold values μ_{up} and μ_{up} are chosen to be 0.8 and 0.1 for practical purpose. The distance between the thumb and the index finger is normalized by dividing its maximum distance.

Based on these mapping relationships, we define the observation $\boldsymbol{o}_t$ at a timestamp t as $\boldsymbol{o}_t = \{\boldsymbol{i}_t^{\text{gcam}}, \boldsymbol{i}_t^{\text{lcam}}, \boldsymbol{s}_t\}$, where $\boldsymbol{i}_t^{\text{gcam}}, \boldsymbol{i}_t^{\text{lcam}}$ are video frames captured by the global and local cameras. $\boldsymbol{s}_t$ is the state of the robot and is defined as

$$\boldsymbol{s}_t = \{p_x, p_y, p_z, q_w, q_x, q_y, q_z, g\}\,.$$

Here, (p_x, p_y, p_z) are the $^{\Sigma_{\text{wd}}}\boldsymbol{p}_{\text{eef}}$ position of the robot at time t. (q_w, q_x, q_y, q_z) are the $^{\Sigma_{\text{wd}}}\mathbf{R}_{\text{eef}}$ rotation of the robot at time t. g is the gripper open/close state at time t.

Based on this observation and the definition of the robot's state, the robot's action at timestamp t is defined as:

$$\boldsymbol{a}_t = \{p_x, p_y, p_z, q_w, q_x, q_y, q_z, g\}\,. \tag{1}$$

It has the same mathematical representation as $\boldsymbol{s}_t$, except that the constituent values denote the position, rotation, and gripper state that the robot is going to act on at the next time step.We clarified the above discussion at the beginning of the experimental section.

During data collection, a human operator controls the robot via the teleoperation system to demonstrate how to group and grasp multiple objects. The collected demonstrations were used to train a Diffusion Policy to learn human strategies. Further details are provided in the following section [Note 1].

IV. Network Selection and Learning

Based on the past T_o time steps of observations at time t $\boldsymbol{O}_t = \{\boldsymbol{o}_{t-T_o+1}, \boldsymbol{o}_{t-T_o+2}, \ldots, \boldsymbol{o}_t\}$, our diffusion policy is expect to infer actions for T_p time steps, namely $\boldsymbol{A}_t = \{\boldsymbol{a}_{t-T_o+1}, \boldsymbol{a}_{t-T_o+2}, \ldots, \boldsymbol{a}_t, \boldsymbol{a}_{t+1}, \ldots, \boldsymbol{a}_{t+T_p-T_o}\}$. The inferred $\boldsymbol{A}_t$ includes the predicted past actions $\{\boldsymbol{a}_{t-T_o+1}, \boldsymbol{a}_{t-T_o+2}, \ldots, \boldsymbol{a}_{t-1}\}$ and the predicted future actions $\{\boldsymbol{a}_t, \boldsymbol{a}_{t+1}, \ldots, \boldsymbol{a}_{t+T_p-T_o}\}$. The predicted past actions correspond to the observations of the past T_o time steps. The predicted future actions are the learned policies for the next $T_p - T_o$ time steps. The robot will execute T_a steps of actions $\{\boldsymbol{a}_t, \boldsymbol{a}_{t+1}, \ldots, \boldsymbol{a}_{t+T_a-1}\}$ for collecting new observations and conduct new predictions. T_a is selected to be a value smaller than $T_p - T_o$. Fig. 3 illustrates the details of the time step division for observation, prediction, and execution.

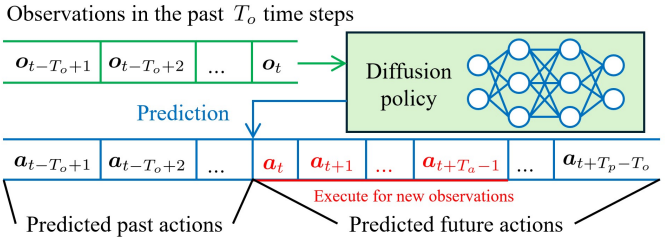

Fig. 3: The diffusion policy takes as input the observations from the past T_o time steps. It infers likely actions for the past T_o time steps and predicts actions for the future $T_p - T_o$ time steps. The first T_a predicted future actions are executed for iterative observation-prediction looping.

The internal structure of the used method is illustrated in Fig. 4. There are in total T_o observations. Each observation consists of two image frames respectively captured from the global and local cameras. They are processed separately using two ResNet-18 for feature extraction. Each image frame is compressed into a N-dimensional feature vector after feature

[Note 1] A follow-up interview with the human operator revealed the overall heuristics used during teleoperated data collection. To avoid overloading the gripper, the number of grouped objects was limited to three, based on object size. Objects were grouped if they were spatially close and roughly aligned to avoid significant rotational adjustments. When choosing the first object, the operator tended to select one surrounded by nearby candidates to facilitate multi-object grouping. These human strategies may offer useful inspiration for readers interested in understanding the underlying rules.

extraction, and the total observations form two $N \times T_o$-dimensional vectors. Additionally, the observation includes the robot's state, which is an 8-dimensional vector for each observation as discussed in the previous section. In total, the robot states form a $8 \times T_o$-dimensional vector. The two $N \times T_o$-dimensional vectors together with the $8 \times T_o$-dimensional robot state vector are concatenated to form a $(2N + 8) \times T_o$-dimensional vector which serves as the observation embedding of a U-Net core used in the diffusion policy network.

The U-Net core takes a noisy action sequence (represented as a $8 \times T_p$-dimensional vector) as input and predicts the noise in the representation. The output of the U-Net is another $8 \times T_p$-dimensional vector where each value corresponds to the estimated noise in the input sequence. This predicted noise is subsequently subtracted from the input sequence within the denoising box shown in Fig. 4, which is mathematically

$$\boldsymbol{A}_t^{k-1} = \alpha(\boldsymbol{A}_t^k - \gamma\boldsymbol{\varepsilon}_{\boldsymbol{\theta}}(\boldsymbol{A}_t^k, \boldsymbol{O}_t, k) + \mathcal{N}(0, \sigma^2\boldsymbol{I})). \quad (2)$$

Here, k indicates the kth denoising step. $\boldsymbol{A}_t^k$ represents the noisy action sequence at the kth denoising step. $\boldsymbol{\varepsilon}_{\boldsymbol{\theta}}()$ represents the noise prediction network (namely the U-Net core). θ indicates the parameters of the network. For each denoising iteration, the iteration step index k is encoded using a P-dimensional sinusoidal positional embedding and is also incorporated into the U-Net core to provide temporal context. The positional embedding helps guide the network in estimating how much noise remains to be removed. α, γ, σ are parameters for controlling the learning stability [Note 2]. The noisy action sequence is initialized by sampling a Gaussian noise distribution. After undergoing K iterative denoising steps, the diffusion policy network will output the denoised $\boldsymbol{A}_t^0$ as predicted action sequences.

To train the denoising network, we draw a sequence of $\boldsymbol{O}_t$ and a sequence of expert action $\boldsymbol{A}_t^0$ from the collected data set, select a random diffusion step k and add Gaussian noises ε^k to the expert actions. The parameters θ of the U-Net core are learned by minimizing

$$\mathcal{L} = ||\boldsymbol{\varepsilon}^k - \boldsymbol{\varepsilon}_{\boldsymbol{\theta}}(\boldsymbol{A}_t^0 + \boldsymbol{\varepsilon}^k, \boldsymbol{O}_t, k)||^2. \quad (3)$$

V. Experiments and Analysis

Our experiments were conducted using a Universal Robots UR3 manipulator with a Robotiq 2F-85 gripper, as shown in Fig. 1(a). The target objects for grouping and grasping were placed within a yellow frame measuring 300 mm × 400 mm and are expected to be delivered to an adjacent tray. Two Intel RealSense D435 cameras are used as the global and local cameras. Only the RGB data was used in the experiments. The RGB image sizes are compressed to 240 × 320 pixels to avoid too much GPU memory consumption. A PC with Intel Core Ultra 7 285K, DDR5-5600 192GB memory, and NVIDIA RTX A6000 GPU was used to train the diffusion policy. When collecting human expert data, another Intel RealSense D435 camera is used to track human hand poses and gripping states.

For evaluation, we used two categories of elongated cuboids and conducted additional experiments on several unseen object geometries. The first category consists of uniformly red 3D-printed cuboids with dimensions of 120 mm (length), 20 mm (width), and 20 mm (height). These objects are easily distinguishable and serve as a controlled setting for verifying the algorithm's effectiveness. The second category includes real-world commercial products with similar elongated shapes (i.e., chocolate snack bars and fructose snack bars). The dimensions of these products will be detailed in later subsections. The model was initially trained on the first category and subsequently fine-tuned using the second to evaluate its transferability and performance on real-world data. Finally, to assess generalization, we tested the learned policy on several unseen object geometries beyond elongated cuboids.

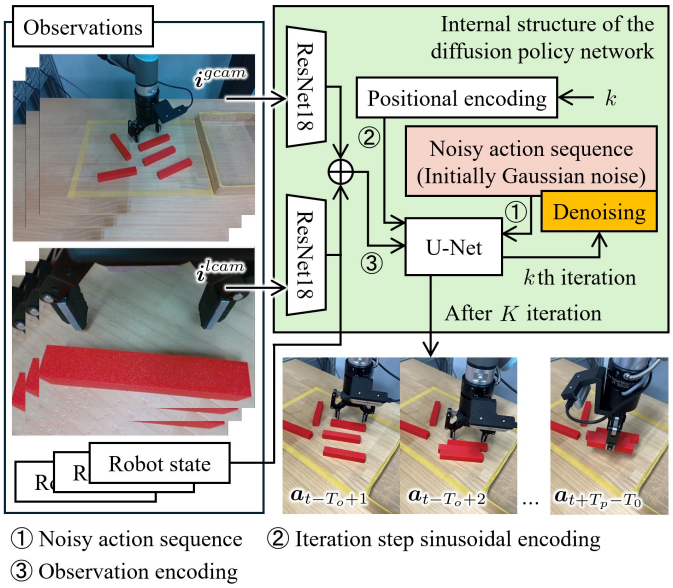

Fig. 4: Internal mechanism of the diffusion policy network. It has a U-Net core that accepts a noisy action sequence and predicts the noise within it. Through K iterations, the U-Net core progressively removes the noise to acquire the action sequence most suitable for the observations.

In our particular diffusion policy implementation, we set both the ResNet-18 output dimension N and the positional encoding dimension P to be 128, set T_o, T_a, and T_p to be 2, 8, and 16, respectively. That is, the learned policy takes the consecutive observations $\boldsymbol{o}_{t-1}$ and $\boldsymbol{o}_t$ as input and predicts future actions $\boldsymbol{a}_t$, $\boldsymbol{a}_{t+1}$, ..., $\boldsymbol{a}_{t+7}$ for the next 8 steps. By performing inference every 8 steps and executing the predicted actions in a closed-loop control manner, the system continuously adjusts its behavior. The hidden layers of the U-Net are set to [512, 1024, 2024]. The optimization method used is AdamW, with a learning rate of 10^{-4} and a weight decay of 10^{-6}. The batch size is set to 64. For the diffusion process, the number of denoising steps is set to $K = 100$.

Note that in our experiments, we did not include a comparison with rule-based methods. This is because our goal is not to outperform rule-based approaches, but to provide a more streamlined and accessible solution. In fact, rule-based methods can potentially achieve comparable performance with sufficient engineering effort. However, reaching such performance

[Note 2] The default values from the LeRobot source code was adopted. Check the following webpage for details. https://github.com/huggingface/lerobot

typically requires substantial domain expertise and extensive manual tuning [Note 3]. Furthermore, implementing a rule-based system involves multiple interdependent components, such as visual calibration, object detection, and motion planning. The overall system performance is highly sensitive to the quality of each module, making it difficult to construct a fair and reliable baseline without significant engineering investment. The challenge is analogous to the difficulty of designing effective rewards in reinforcement learning, which also requires repeated trial-and-error and task-specific adjustments.

A. Experiment 1: Randomly placed 3D printed objects

In the first group of experiments, we focused on the influence of the number of randomly placed objects and the number of training data. During data collection, we conducted 200 demonstrations, where the number of objects was randomly set between 3 and 6, and the target objects were randomly placed within the yellow frame. The collected training data were divided into two datasets of sizes 100 and 200, respectively, and a separate diffusion policy network was trained for each dataset to analyze the impact of data quantity on performance. Each diffusion policy network underwent 300000 training steps. The evaluation is conducted on tasks where 3 to 6 randomly placed printed objects are grouped and delivered using learned policies. The initial position of the UR3 end effector is set above the bin. 20 trials were conducted for each number of objects. In each trial, the robot's performance was assessed until it either successfully completed the task (delivered all objects) or failed. In case of failure, the causes could be: (i) Pushing failure (P): Objects may slip from the gripper during pushing and fail to be grouped. (ii) Grasping failure (G): The robot may fail to grasp or drop objects during delivery. (iii) Stagnation (-): The policy may repeat the same action and cause stagnation. If the robot remains stationary or repeats for more than 5 seconds, the trial is considered a failure. Stagnation during grouping is classified as a pushing failure. Otherwise, it is classified as a grasping failure.

Table I shows the detailed results for all trials and their statistics. Due to the limited space on the article page, this table contains numerous abbreviations. The specific meanings are shown in the notes at the bottom. Especially in the statistical section, we computed the following values: (i) Completion (Compl.): The rate at which all objects were successfully delivered over 20 trials. (ii) Delivery rate (Delr.): The average number of successfully delivered objects divided by the initial number of objects across 20 trials. (iii) Grasp success rate: The number of successful grasps over all grasps across 20 trials. (iv) # Objects per grasp: The average number of objects delivered per successful grasp across 20 trials. Taken together, this value and the previous one are conceptually similar to the Overall Success Rate (OSR) metric proposed by Chen et al. [29]. The maximum values of these statistics across both trained networks and all numbers of objects are highlighted in lime, while the minimum values are highlighted in pink.

[Note 3] As a probabilistic generative model, our method enables online responses to dynamic changes such as human intervention or sudden state shifts, and is more robust than rule-based methods that lack extensive tuning.

We can see from the results that as the number of randomly placed graspable objects increases, the task completion rates decrease. Failures frequently occur during pushing and grouping. The frequency of the failures increases with the number of initially randomly placed objects. The majority of pushing failures are caused by stagnation. This may be attributed to the increasing variety in the input images as the number of objects grows, making it more difficult for the neural network to determine the optimal pushing strategy based on limited information. We can also see from the results that the sequence of grouping and grasping is quite random. The network chose different policies according to object distributions. Furthermore, the results indicate that an increase in training data has a positive impact on all metrics. Notably, when the number of objects reaches 6, the robot grasps an average of 1.68 objects per attempt. This demonstrates that the policy network can effectively support multi-object grasping and thus help a robot improve picking efficiency.

Table II shows the detailed time costs when applying the policy trained on the dataset trained using 200 demonstrations to scenes containing 6 objects. We can see from the results that while multi-object picking involves additional grouping operations that increase the average time per pick, the per-object efficiency improves significantly. In particular, the efficiency (time required per object) is reduced by approximately 3 seconds compared to single-object picking.

Fig. 5 presents an exemplary result. The robot first pushed, grouped, and grasped three objects (a∼c), then grouped and grasped two objects (d, e), and finally grasped a single object without pushing (f). The robot selected appropriate policies based on the object arrangement to successfully complete the task. For other detailed robot behaviors and grasping motions, please refer to the supplementary video.

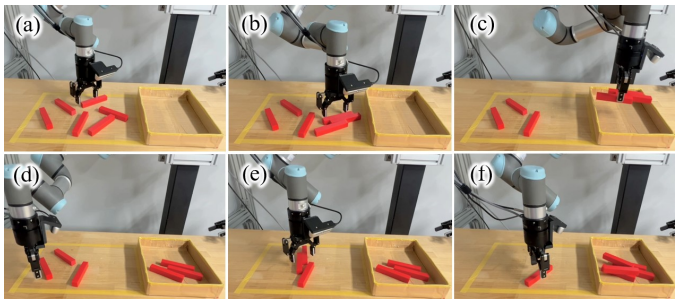

Fig. 5: An exemplary result. (a∼c) First grouping and grasping. (d, e) Second grouping and grasping. (c) Third grasping.

B. Experiment 2: Fine-tuning to snack bars

We conducted fine-tuning on the pre-trained model using the dataset (200 demonstrations) from the previous experiments and evaluated the adaptivity of the proposed method to commercial products. The fine-tuning dataset consisted of 50 demonstration samples using six objects. The demonstrations were carried out in the same way as the printed objects. The commercial products are 6 snack bars, including three Meiji chocolate bars (size: 125 mm × 38 mm × 15 mm, available in two different appearances) and three HiChu fructose bars

TABLE I: Results of randomly placed 3D printed objects

(s) Data	ID	Number of Objects: 3					Number of Objects: 4					Number of Objects: 5					Number of Objects: 6				
		#P	#G	Seq.	C	R	#P	#G	Seq.	C	R	#P	#G	Seq.	C	R	#P	#G	Seq.	C	R
100	1	2	2	(1,1)	N	P	2	2	(1,2)	N	P	1	1	(2)	N	P	0	0	(0)	N	P
	2	2	2	(1,2)	Y	-	3	3	(1,1,2)	Y	-	4	4	(1,1,1,2)	Y	-	0	0	(0)	N	P
	3	2	2	(2,1)	Y	-	0	0	(0)	N	P	0	0	(0)	N	P	0	0	(0)	N	P
	4	0	0	(0)	N	P	0	0	(0)	N	P	3	3	(1,1,1)	N	P	0	0	(0)	N	P
	5	2	2	(2,1)	Y	-	1	0	(0)	N	P	1	1	(1)	N	P	1	1	(1)	N	P
	6	0	0	(0)	N	P	2	2	(2,2)	Y	-	1	1	(1)	N	P	0	0	(0)	N	P
	7	2	2	(1,2)	Y	-	2	2	(2,2)	Y	-	0	0	(0)	N	P	2	2	(1,2)	N	P
	8	0	0	(0)	N	P	1	1	(1)	N	P	0	0	(0)	N	P	2	2	(1,1)	N	P
	9	1	1	(1)	N	P	2	2	(2,2)	Y	-	2	2	(3,2)	Y	-	1	1	(1)	N	P
	10	2	2	(1,2)	Y	-	3	3	(2,1,1)	Y	-	2	2	(1,1)	N	P	1	1	(1)	N	P
	11	1	1	(1)	N	P	0	0	(0)	N	P	1	1	(1)	N	P	0	0	(0)	N	P
	12	1	1	(1)	N	P	3	3	(1,2,1)	Y	-	0	0	(0)	N	P	3	3	(3,2,1)	Y	-
	13	1	1	(3)	Y	-	1	1	(1)	N	P	0	0	(0)	N	P	3	3	(1,3,2)	Y	-
	14	1	1	(3)	Y	-	0	0	(0)	N	P	2	1	(3)	N	G	1	1	(3)	N	P
	15	3	3	(1,1,1)	Y	-	3	3	(2,1,1)	Y	-	0	0	(0)	N	P	2	2	(1,2)	N	P
	16	2	2	(2,1)	Y	-	3	3	(1,2,1)	Y	-	3	3	(1,3,1)	Y	-	4	4	(2,2,1,1)	Y	-
	17	2	2	(1,2)	Y	-	2	2	(2,2)	Y	-	0	0	(0)	N	P	0	0	(0)	N	P
	18	2	2	(2,1)	Y	-	0	0	(0)	N	P	0	0	(0)	N	P	3	3	(2,2,1)	N	P
	19	1	1	(1)	N	P	4	4	(1,1,1,1)	Y	-	2	1	(1)	N	G	1	1	(1)	N	P
	20	3	3	(1,1,1)	Y	-	1	1	(1)	N	P	0	0	(0)	N	P	5	5	(1,1,1,1,2)	Y	-
	St	Compl.: 60%; Delr.: 70% Grasp success rate: 100% # Objects per grasp: 1.4					Compl.: 50%; Delr.: 58% Grasp success rate: 100% # Objects per grasp: 1.44					Compl.: 15%; Delr.: 29% Grasp success rate: 90% # Objects per grasp: 1.45					Compl.: 10%; Delr.: 57% Grasp success rate: 100% # Objects per grasp: 1.52				
200	1	2	2	(2,1)	Y	-	1	1	(2)	N	P	3	3	(1,3,1)	Y	-	3	3	(3,2,1)	Y	-
	2	1	0	(0)	N	G	2	2	(3,1)	Y	-	4	4	(1,1,2,1)	Y	-	4	4	(1,2,2,1)	Y	-
	3	2	2	(2,1)	Y	-	3	3	(2,1,1)	Y	-	3	3	(2,2,1)	Y	-	3	3	(2,3,1)	Y	-
	4	1	1	(3)	Y	-	3	3	(2,1,1)	Y	-	3	3	(2,2,1)	Y	-	3	2	(1,3)	N	
	5	2	2	(2,1)	Y	-	2	2	(2,2)	Y	-	2	2	(1,1)	N	P	3	3	(3,2,1)	Y	-
	6	0	0	(0)	N	P	3	3	(1,2,1)	Y	-	4	4	(1,1,1,2)	Y	-	4	4	(1,2,1,2)	Y	-
	7	3	3	(1,1,1)	Y	-	1	1	(3)	N	P	2	1	(1)	N		5	5	(2,1,1,1,1)	Y	-
	8	2	2	(1,2)	Y	-	4	4	(1,1,1,1)	Y	-	2	2	(1,1)	N	P	1	1	(3)	N	P
	9	3	3	(1,1,1)	Y	-	2	2	(2,2)	Y	-	2	2	(3,2)	Y	-	3	3	(3,2,1)	Y	-
	10	1	1	(3)	Y	-	3	3	(1,2,1)	Y	-	2	1	(1)	N		4	4	(2,2,1,1)	Y	-
	11	3	3	(1,1,1)	Y	-	0	0	(0)	N	P	4	4	(2,1,1,1)	Y	-	0	0	(0)	N	P
	12	1	1	(3)	Y	-	3	3	(2,1,1)	Y	-	0	0	(0)	N	P	3	2	(2,1)	N	
	13	1	0	(0)	N	G	2	2	(2,2)	Y	-	0	0	(0)	N	P	1	1	(2)	N	P
	14	2	2	(1,1)	N	P	2	1	(2,)	N	G	3	3	(2,2,1)	Y	-	0	0	(0)	N	P
	15	3	3	(1,1,1)	Y	-	3	3	(2,1,1)	Y	-	4	4	(1,1,2,1)	Y	-	4	4	(2,2,1,1)	Y	-
	16	2	2	(2,1)	Y	-	2	2	(2,1)	N	P	1	1	(2)	N	P	4	4	(1,2,2,1)	Y	-
	17	3	3	(1,1,1)	Y	-	4	4	(1,1,1,1)	Y	-	2	1	(2)	N		1	1	(1)	N	P
	18	2	2	(2,1)	Y	-	2	2	(1,3)	Y	-	4	4	(1,1,1,2)	Y	-	3	3	(2,1,3)	Y	-
	19	1	1	(3)	Y	-	2	2	(1,1)	N	P	0	0	(0)	N	P	1	0	(0)	N	
	20	2	2	(1,2)	Y	-	2	2	(3,1)	Y	-	4	4	(1,1,1,2)	Y	-	0	0	(0)	N	P
	St	Compl.: 80%; Delr.: 83% Grasp success rate: 94.4% # Objects per grasp: 1.43					Compl.: 70%; Delr.: 85% Grasp success rate: 97.8% # Objects per grasp: 1.51					Compl.: 55%; Delr.: 65% Grasp success rate: 100% # Objects per grasp: 1.42					Compl.: 55%; Delr.: 66% Grasp success rate: 100% # Objects per grasp: 1.68				

Note Meanings of abbreviations: #P – Number of grouping actions; #G – Number of grasping actions; Seq. – The number of objects grasped by each grasping action, recorded in sequential order from the start of the task as a list; C – Task completion (Y means Yes, all objects are cleared and delivered. N means No, some objects remained due to failures); R – Reason for task failure (P means a pushing failure. G means a grasping failure); St – Statistical summary of the results above the row (Maximum values of the statistics are highlighted in lime, while minimum values are highlighted in pink); Compl. – Completion rate; Delr. – Delivery rate.

(size: 130 mm $\times$ 28 mm $\times$ 15 mm, available in three different appearances). The object placement positions were randomly determined, and the fine-tuning process was conducted over 50000 training steps.

Similar to the previous experiments, we carried out 20 trials for evaluation. Table III shows the detailed results for these trials. We can see from the results that the task completion rate was 0%, suggesting that additional learning using a small amount of data is insufficient for transferring policy knowledge effectively. An important reason for failing to clear all objects was that, during pushing, objects deviated in unexpected directions and became detached from the end effector. Additionally, the robot encountered lots of stagnation and repeated similar policies. Grasping failures also occurred frequently. The stagnation and grasping failures were attributed to differences in the shape and surface texture of the snack bars compared to the objects used in pretraining. The fine-tuning struggles to adapt to such differences.

On the other hand, the delivery rate and # objects per grasp are respectively 23% and 1.47. They indicate that a certain level of multi-object grasping was achieved. The network can selectively group and grasp a few objects, as shown by the sequence column of the table, the example in Fig. 6, and also the supplementary video. Overall, although the fine-

TABLE II: Time costs of 6 objects trained on the 200 dataset

ID	t_1(s)	t_2(s)	t_3(s)	T	ID	t_1(s)	t_2(s)	t_3(s)	T
1	(14)	(21)	(26)	61	(11)	-	-	-	-
2	(14,14)	(23,21)	0	72	(12)	(16)	(28)	-	-
3	(14)	(23)	(24)	61	13	-	(16)	-	-
4	(12)	-	(24)	-	14	-	-	-	-
5	(12)	(33)	(28)	73	15	(13,14)	(29,24)	-	80
6	(13,13)	(26,26)	-	78	16	(17,14)	(24,19)	-	74
7	(18,15,12,12)	(20)	-	77	17	(12)	-	-	-
8	-	-	-	-	18	(21)	(27)	(31)	79
9	(14)	(26)	(26)	66	19	-	-	-	-
10	(15,16)	(19,23)	-	73	20	-	-	-	-
St	Avg. time for 1 object: 14.3 s				Efficiency for 1 object: 14.3 s				
	Avg. time for 2 objects: 23.8 s				Efficiency for 2 objects: 11.9 s				
	Avg. time for 3 objects: 26.0 s				Efficiency for 3 objects: 8.7 s				

Note t_1 – Time required to pick and deliver a single object; t_3 – Time required to pick and deliver two objects simultaneously; t_3 – Time required to pick and deliver three objects simultaneously; T – Total time required to clear all objects; Efficiency is computed by dividing the average time by the number of simultaneously picked objects.

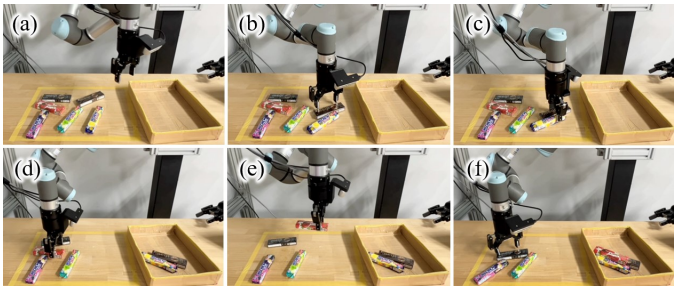

Fig. 6: Snack bar trial with a fine-tuned model. (a∼c) First grouping and grasping. (d, e) Second grasping. (f) Stagnation.

tuned network did not help completely clear all objects, it still demonstrates a certain level of effectiveness. Learning from expert data requires a large volume of high-quality data. Developing efficient methods for collecting such data could potentially help improve learning and inference performance.

TABLE III: Results of 6 randomly placed snack bars using a model additionally fine-tuned with 50 times of demonstrations

ID	#P	#G	Seq.	C	R	ID	#P	#G	Seq.	C	R
1	1	1	(2)	N	P	11	3	2	(2,1,1)	N	G
2	1	1	(2)	N	P	12	0	0	(0)	N	P
3	1	0	(2)	N	G	13	2	2	(1,1)	N	P
4	1	0	(1)	N	G	14	1	1	(2)	N	P
5	3	3	(2,1,1)	N	P	15	1	0	(1)	N	G
6	2	1	(1,1)	N	G	16	0	0	(0)	N	P
7	0	0	(0)	N	P	17	2	2	(2,1)	N	P
8	2	1	(2,2)	N	G	18	2	1	(2,1)	N	G
9	0	0	(0)	N	P	19	0	0	(0)	N	P
10	4	3	(1,2,1,1)	N	G	20	1	1	(1)	N	P
St	Completion: 0%; Delivery rate: 23%; # Objects per grasp: 1.47										

Note See note of Table I for meanings of abbreviations.

C. Experiment 3: Unseen shapes

We also performed experiments to evaluate the generalization ability of the learned policy across different object geometries. Specifically, we tested the trained policy on three distinct object types: L-shaped objects, cubes, and hexagons. The results are presented in Table IV. The L-shaped objects consist of two connected arms, each measuring 60 mm in length. The cross-sectional edge length is 20 mm. The cube and hexagonal objects both have edge lengths of 20 mm. These dimensions were carefully selected to test the robustness of the policy under moderate changes in object shape and structure.

The results show that the hexagon shape had the lowest success rate, likely due to its significant geometric deviation from the training objects. The success rate for the cube was also relatively low, as the robot often misidentified a cube positioned outside the gripper jaws as part of an elongated cuboid, leading to grasp failures. In contrast, the L-shaped objects achieved a high success rate. However, some trials involving L-shapes resulted in notably long grasping durations. These delays primarily occurred when one arm of the L-shape was inside the gripper jaws, causing the robot to hesitate between adjusting toward the other arm or attempting a direct grasp by closing the fingers. This ambiguity led to repeated stalling and prolonged execution. Although many trials were successful, no instances of simultaneous multi-object grasping were observed. Fig. 7 shows some exemplary results of the trials. Full videos can be found in the supplementary file.

TABLE IV: Success and failures on three unseen objects

ID	L-shape		Cube		Hexagon	
	#S (Time (s))	#F	#S (Time (s))	#F	#S (Time (s))	#F
1	5 (78,18,12,90,45)	0	3 (42,35,41)	0	1 (21)	5
2	1 (9)	0	1 (42)	2	2 (13,14)	8
3	0	0	2 (23,23)	4	0	3
4	4 (95,18,25,16)	0	1 (16)	3	0	0
5	4 (12,18,18,40)	1	0	0	1	6

Note #S – Number of successful grasps; #F – Number of failed grasps. All successful grasps involved only a single object. No instances of simultaneous multi-object grasping were observed. The grasping time for each object is shown in the parentheses.

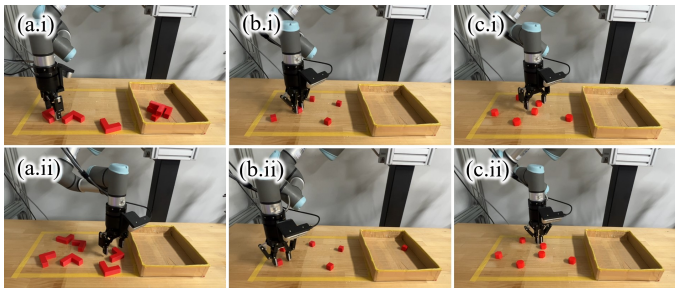

Fig. 7: Exemplary results of unseen shapes. (a.i∼ii) L-shapes. (b.i∼ii) Cubes. (c.i∼ii) Hexagons. (c.ii) is a grasping failure.

VI. Conclusions and Future Work

In this study, we proposed a diffusion policy-based imitation learning framework for grouping and grasping multiple objects. Experiments show that the robot can flexibly perform pushing, grouping, and grasping across various object configurations, and the pre-trained policy can be fine-tuned for new objects. While effective, the method faces limitations

when handling large numbers of objects, scaling to larger workspaces, or adapting to diverse shapes and textures. Future work will focus on exploring the impact of expert strategies on data efficiency, improving data collection, and integrating analytical techniques to enhance robustness and generalization.

REFERENCES

[1] T. Sakamoto, W. Wan, T. Nishi, and K. Harada, "Efficient picking by considering simultaneous two-object grasping," in *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2021, pp. 8295–8300.

[2] W. C. Agboh, J. Ichnowski, K. Goldberg, and M. R. Dogar, "Multi-object grasping in the plane," in *Int. Symp. Robot. Res. (ISRR)*. Springer, 2022, pp. 222–238.

[3] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," *Int. J. Robot. Res.*, p. 02783649241273668, 2023.

[4] Y. Aiyama, M. Minami, and T. Arai, "Manipulation of multiple objects by two manipulators," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 1998, pp. 2904–2909.

[5] K. Harada and M. Kaneko, "Neighborhood equilibrium grasp for multiple objects," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2000, pp. 2159–2164.

[6] ——, "Enveloping grasp for multiple objects," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 1998, pp. 2409–2415.

[7] T. Yoshikawa, T. Watanabe, and M. Daito, "Optimization of power grasps for multiple objects," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2001, pp. 1786–1791.

[8] T. Yamada, S. Yamanaka, M. Yamada, Y. Funahashi, and H. Yamamoto, "Grasp stability analysis of multiple planar objects," in *IEEE Int. Conf. Robot. Biomimetics*, 2009, pp. 1032–1038.

[9] T. Yamada, M. Yamada, and H. Yamamoto, "Stability analysis of multiple objects grasped by multi-fingered hands with revolute joints in 2d," in *IEEE Int. Conf. Mechatronics Autom.*, 2012, pp. 1785–1792.

[10] T. Yamada and H. Yamamoto, "Static grasp stability analysis of multiple spatial objects," *J. Control Sci. Eng.*, vol. 3, pp. 118–139, 2015.

[11] T. Chen, A. Shenoy, A. Kolinko, S. Shah, and Y. Sun, "Multi-object grasping – estimating the number of objects in a robotic grasp," in *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2021, pp. 4995–5001.

[12] A. Shenoy, T. Chen, and Y. Sun, "Multi-object grasping-efficient robotic picking and transferring policy for batch picking," in *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2022, pp. 2741–2747.

[13] K. Takahashi, W. Ko, A. Ummadisingu, and S.-i. Maeda, "Uncertainty-aware self-supervised target-mass grasping of granular foods," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2021, pp. 2620–2626.

[14] K. Takahashi, N. Fukaya, and A. Ummadisingu, "Target-mass grasping of entangled food using pre-grasping & post-grasping," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 1222–1229, 2021.

[15] Y. Li, B. Liu, Y. Geng, P. Li, Y. Yang, Y. Zhu, T. Liu, and S. Huang, "Grasp multiple objects with one hand," *IEEE Robot. Autom. Lett.*, vol. 9, no. 5, pp. 4027–4034, 2024.

[16] P. Jiang, J. Oaki, Y. Ishihara, and J. Ooga, "Multiple-object grasping using a multiple-suction-cup vacuum gripper in cluttered scenes," *Robotics*, vol. 13, no. 6, 2024.

[17] V. P. Nguyen and W. T. Chow, "Wiring-claw gripper for soft-stable picking up multiple objects," *IEEE Robot. Autom. Lett.*, vol. 8, no. 7, pp. 3972–3979, 2023.

[18] K. Yao and A. Billard, "Exploiting kinematic redundancy for robotic grasping of multiple objects," *IEEE Trans. Robot.*, vol. 39, no. 3, pp. 1982–2002, 2023.

[19] W. C. Agboh, S. Sharma, K. Srinivas, M. Parulekar, G. Datta, T. Qiu, J. Ichnowski, E. Solowjow, M. Dogar, and K. Goldberg, "Learning to efficiently plan robust frictional multi-object grasps," in *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2023, pp. 10 660–10 667.

[20] S. Aeron, E. LLontop, A. Adler, W. C. Agboh, M. Dogar, and K. Goldberg, "Push-mog: Efficient pushing to consolidate polygonal objects for multi-object grasping," in *IEEE Int. Conf. Autom. Sci. Eng. (CASE)*, 2023, pp. 1–6.

[21] Z. Ye and Y. Sun, "Only pick once – multi-object picking algorithms for picking exact number of objects efficiently," *arXiv:2307.02662*, 2023.

[22] K. Srinivas, S. Ganti, R. Parikh, A. Ahmad, W. Agboh, M. Dogar, and K. Goldberg, "The busboy problem: Efficient tableware decluttering using consolidation and multi-object grasps," in *IEEE Int. Conf. Autom. Sci. Eng. (CASE)*, 2023, pp. 1–6.

[23] 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 *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2018, pp. 4238–4245.

[24] Y. Chen, Z. Ju, and C. Yang, "Combining reinforcement learning and rule-based method to manipulate objects in clutter," in *Int. Joint Conf. Neural Netw. (IJCNN)*, 2020, pp. 1–6.

[25] Y. Wang, K. Mokhtar, C. Heemskerk, and H. Kasaei, "Self-supervised learning for joint pushing and grasping policies in highly cluttered environments," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2024, pp. 13 840–13 847.

[26] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," *arXiv:2304.13705*, 2023.

[27] Y. Qin, W. Yang, B. Huang, K. Van Wyk, H. Su, X. Wang, Y.-W. Chao, and D. Fox, "Anyteleop: A general vision-based dexterous robot arm-hand teleoperation system," in *Robot.: Sci. Syst.*, 2023.

[28] R. A. Potamias, J. Zhang, J. Deng, and S. Zafeiriou, "Wilor: End-to-end 3d hand localization and reconstruction in-the-wild," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2025.

[29] T. Chen, R. Frumento, G. Pagnanelli, G. Cei, V. Keth, S. Gafarov, J. Gong, Z. Ye, M. Baracca, S. D'Avella, M. Bianchi, and Y. Sun, "Benchmarking multi-object grasping," *arXiv:2503.20820*, 2025.