Leveraging Dexterous Picking Skills for Complex Multi-Object Scenes

Anagha Rajendra Dangle, Mihir Pradeep Deshmukh, Denny Boby, Berk Calli

***Abstract*— This work focuses on the problem of robotic picking in challenging multi-object scenarios. These scenarios include difficult-to-pick objects (e.g., too small, too flat objects) and challenging conditions (e.g., objects obstructed by other objects and/or the environment). To solve these challenges, we leverage four dexterous picking skills inspired by human manipulation techniques and propose methods based on deep neural networks that predict when and how to apply the skills based on the shape of the objects, their relative locations to each other, and the environmental factors. We utilize a compliant, under-actuated hand to reliably apply the identified skills in an open-loop manner. The capabilities of the proposed system are evaluated through a series of real-world experiments, comprising 45 trials with 150+ grasps, to assess its reliability and robustness, particularly in cluttered settings. The videos of all experiments are provided at https://dexterouspicking.wpi.edu/. This research helps bridge the gap between human and robotic grasping, showcasing promising results in various practical scenarios.**

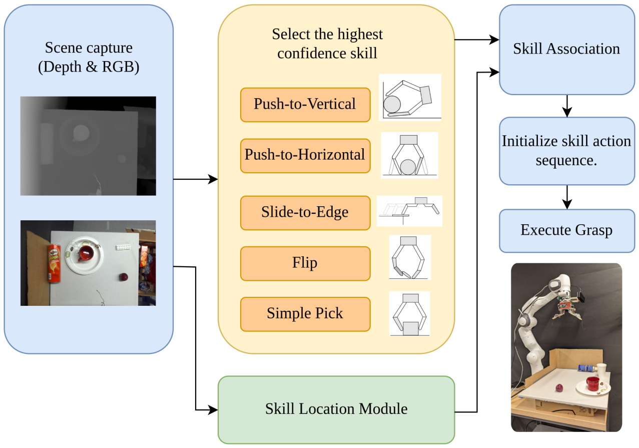


Fig. 1. Overview of our dexterous picking pipeline.

I. Introduction

Humans possess an extraordinary ability to grasp objects with remarkable precision and adaptability, which is particularly evident in how we approach each object uniquely. The intricate skill of grasping an object involves the combined effort of sensory perception, motor control, and cognitive decision-making that enables us to tailor our grasp to the specific characteristics of the target object and the manipulation scene. The dexterous nature of human picking has been a source of inspiration aiming to replicate the versatility in grasping techniques [1], [2]. However, despite extensive efforts, there have been significant challenges in translating the nuances of human grasping into robotic systems. A significant issue lies in dynamically adjusting grasp strategies for a wide range of objects, especially within cluttered or unstructured environments, where the unique characteristics of each object and the conditions in the manipulation scene demand a flexible and *context-aware* approach to grasping.

Building on the foundation of human dexterity [3], we propose a set of algorithms and a manipulation pipeline that utilizes four dexterous manipulation skills inspired by frequently employed human picking skills. These skills include sliding objects to the edge, pushing them to a vertical surface, leveraging a horizontal surface to facilitate their picking, and flipping them. Our algorithms are capable of choosing when and how to utilize these skills for picking objects in challenging multi-object scenarios, including complex objects and varying conditions (Fig. 1). Leveraging these algorithms, we have developed an end-to-end pipeline aimed at efficiently clearing objects from a tabletop.

In the robotics literature, it is demonstrated that these skills can be highly effective in picking difficult objects and handling complex scenarios. [1], [4] underscores the potential and significance of these skills. For instance, sliding flat, medium-sized objects to the edge of the table frees up grasping surfaces. Furthermore, for flat and small objects, flipping skill helps stabilize the object with one finger while the other pivots and lifts it towards a grasp. Another distinctive use case is leveraging horizontal surfaces to pick medium-sized round objects that might otherwise move during the grasp motions. Similarly, pushing the object to a vertical surface helps stabilize it and is especially useful when the object is flushed to that surface.

Nevertheless, given an image of the manipulation scene, the system needs to be able to decide when and how to adopt these skills. Choosing the right skill for a target object surely depends on its shape, but it is not the only factor. The context that the object is in, i.e., the relative positions of the other objects, and the configuration of the environment (e.g., the existence and locations of table edges and vertical walls), both play essential roles. We would like to illustrate this with two examples: when a cylindrical object is lying on a table, well-separated from other items, it can be manipulated using simple picking methods. If the object is adjacent to a wall, the push-to-vertical skill becomes particularly effective, allowing the object to be cornered and securely grasped. Similarly, for picking a flat plate from a tabletop, the system would benefit from sliding the plate to the edge of the table.

All authors are from the Robotics Engineering Department of Worcester Polytechnic Institute.

This work was supported in part by Amazon Robotics Greater Boston Tech Initiative (GBTI) and the National Science Foundation under grant FRR-2338703.

However, if an obstacle lies between the plate and the table's edge or if another object is resting on the plate, the system must first remove these obstructions before attempting to slide the plate. Also, while sliding the plate to the edge, it is essential to identify how to apply these skills, i.e. where to establish contact with the object so that it allows for successful execution of the skill.

The algorithms proposed in this paper provide precisely these kinds of decisions. Summarizing our contributions: 1) We have developed a Skill Detection Module that leverages deep learning to predict the most suitable picking skills for each object within a multi-object scene, along with confidence levels for each prediction. As such, given an RGB-D image of the scene, our algorithm decides which object should be picked first (based on the confidence value) along with the skill that should be applied. The algorithm is capable of producing the appropriate skills, taking into account both the object's shape and its context (including the positions of nearby objects and the overall environmental setup). 2) We developed a Skill Location Module based on an attention gate-based neural network that is capable of identifying the application location of the chosen skill. Both models were trained on datasets comprising manually labeled images from both simulated environments and real-world settings, with approximately a 70:30. 3) We present an end-to-end pipeline that applies these neural networks to the task of clearing objects from a tabletop. The effectiveness of this system was thoroughly evaluated through 45 real-world tests, encompassing more than 150 instances of object grasping. As supplementary data [5], we have provided detailed information, including accounts of each experiment and their corresponding video recordings.

For executing these skills, we utilize an open-source compliant underactuated hand: OpenHand Model-O [6]. Compliant hands not only simplify robotic grasping by adapting to the shape of the object but also facilitate contact-rich operations, enabling dexterous manipulation as outlined above. These hands allow us to implement the selected skills in an open-loop manner, similar to [7]. While our neural network models can also be deployed with rigid grippers, this would necessitate additional development effort and precise force sensing.

In our approach to skill execution, we employ predefined motion parameters for the robot arm, specific to each skill. For most skills, the orientations of the gripper are fixed and the distances to environmental features like table edges and walls are assumed to be known in advance. The robot determines the locations to apply these skills based on the skill location model.

II. Related work

In this discussion, we focus on dexterous picking strategies that utilize various human-inspired hand motions. Various studies in the literature have concentrated on motion planning and execution of dexterous manipulation skills. For instance, [4] presents methods for specific skills like slide-to-edge, push-to-vertical, and push-to-horizontal. Another study [7] implements the flip skill. Notably, these works focus on single-object scenarios and lack mechanisms for automatically identifying suitable skills for different objects. In contrast, our approach targets the automatic identification of skills and their application in multi-object environments.

The method in [8] explores human-like grasp strategies learned from minimal examples. This paper is also designed for single-object scenarios, and purely relying on RGB data reduces its capability to utilize contextual information in multi-object scenes. Similarly, [1] employs anthropomorphic soft hands for a more human-like grasping approach. Again, focusing on single object scenes, this work mainly demonstrates the efficacy of the various dexterous skills.

Considering multi-object scenes, the work in [9] proposes a multi-affordance approach. Rather than dexterous hand-oriented skills, this method utilizes a suction cup and a parallel-jaw gripper. The system is capable of choosing among multiple affordances to pick the objects. Different from that work, our method focuses on hand-oriented dexterous manipulation skills.

We would also like to highlight the potential of the method in [10] leveraging synthetic data to train deep networks for primitive shape recognition. While, that work does not explicitly focus on dexterity, it could be useful to assign skills based on the object shape. Nevertheless, incorporating contextual information would still require training additional models.

III. Methodology

In this work, we aim to enable robots to utilize dexterous picking skills in complex multi-object scenarios via automated skill selection and application algorithms. We focus on five picking strategies as explained in Section III-A. The dataset that is used to train our algorithms is explained in Section III-B. Our skill selection algorithm is explained in Section III-C. Our network that identifies the application location of a chosen skill is presented in Section III-D. Finally, we present our entire pipeline utilizing these two networks for clearing a tabletop in Section III-E.

A. Primitive skills and gripper

In this work, we utilize five picking strategies as follows:

- Slide-to-Edge: Previously proposed by [4], this skill is utilized for grasping medium-sized flat objects by sliding the objects to the edge of the table and exposing the lower side for easier grasping. Example objects: plates, books.
- Push-to-Vertical: Introduced by [4], this skill corners the objects by pushing them against vertical surfaces such as shelf walls or bin walls. This skill is especially useful when the objects are flushed against the vertical surface and/or could otherwise move during the grasping motion (e.g., a cylindrical object). Example objects: bottle, banana.
- Flip: This skill, inspired by the insights from [7] and [11], is used for picking up small flat objects. It supports one side of the object with one of the fingers while

another finger sweeps the surface, establishes contact with the object, and lifts it to a grasp position. Example objects: coins, keys.
- Push-to-Horizontal: As detailed in [4], in this skill, the hand approaches the object from the top, gets close to the table surface so that the fingers of the hand sweep the surface while grasping the object. This strategy makes the picking system more robust to the object position uncertainties and is especially useful when grasping round objects that may move during the grasping motions. For example, if a non-dexterous picking strategy is used while grasping a ball, one of the fingers could touch the ball before the other, which may make it move and fail the grasp. Sweeping the horizontal surface allows to guide the fingers towards the object, cage it before the grasp, and make the grasp more robust to measurement noise. Example objects: ball, plum.
- Simple-Pick: Some objects (and conditions) might not require dexterous picking skills, they can simply be grasped without needing dexterity. Here, we choose a very simple picking strategy, taking advantage of the compliant nature of the underactuated hand: the gripper approaches the object from the top and orients its fingers to grasp the object along its principle axis. Example objects: cup, small box.

In the manipulation literature, it is shown that the execution of these skills is greatly simplified when adaptive/compliant grippers are used [1], [6]. Gripper compliance allows the robot to safely make contact with the object and the environment without requiring force sensing. For instance, in the push-to-horizontal skill, the gripper is initially positioned in a predefined pose with its fingers partially closed. It is then pushed towards the horizontal surface, mechanically adapting its shape during this motion while its fingers sweep the surface to grasp the object. In this paper, we utilize an Openhand Model-O three-finger underactuated hand. As such, the execution of the skills is largely open-loop, following very similar procedures to the cited papers above (we do not claim any novelty in the execution of these skills). Nevertheless, the robot should still know the locations of the objects to apply these skills as well as identify when to use which skill. We first present our dataset to train the algorithms for these purposes.

B. Dataset

To train our Skill Detection and Skill Location models, we compiled a dataset of depth images (top-down view) from 570 diverse multi-object scenes, with over 1,500 object instances. These instances are manually labeled with the appropriate skill ID and skill location with all the labeling informed by the context of the scene. For example, when a cylindrical object is flushed toward a vertical wall, it is labeled as push-to-vertical, whereas, when the same object is away from the wall, it is labeled as simple-pick. To ensure our model develops an understanding of skill applicability, we intentionally included a larger number of instances depicting straightforward contexts for each skill. This approach allows the model to assign higher confidence scores to simpler scenarios during skill classification. For example, we provided slightly more examples of flat objects being at the edge of the table, than the examples when it is in the middle of clutter. This training strategy enables the model to more confidently identify the "slide-to-edge" skill when an object is near the table's edge. The dataset is available at [5].

For every object within our dataset, we assigned labels indicating the precise location for skill application. Specifically, for the slide-to-edge skill, it is the exact point where the gripper should initiate contact to slide the object. Similarly, for other skills such as push-to-vertical, push-to-horizontal, simple-pick, and flip, it is the point where the gripper should approach to execute the skill successfully. These detailed positional labels are utilized for the training of our Skill Location Model to learn where to effectively apply each skill.

We developed separate skill detection and location models for depth and RGB images, with the reasoning that depth data offers greater consistency across simulated and real-world environments. This allows us to train the models using both simulations and real-world samples, saving us data collection effort. In our approach, simulation images contribute to 70% of our dataset. To bridge the gap between the simulated and real-world depth images, we applied techniques like reducing contrast and adding Gaussian noise around the edges of objects in simulated images. These measures aim to reduce the domain gap between simulated and real-world scenarios, improving the models' ability to adapt and generalize across different settings.

However, depth data has its limitations, particularly in capturing very small or flat objects due to their minimal depth profiles. Relying solely on depth-based models would result in overlooking these objects with no significant depth signature. To address this, we also compiled a smaller RGB dataset focused exclusively on these left-out small and/or flat objects. This dataset, comprising 80 images, is specifically labeled by concentrating on the slide-to-edge and flip skills (since these are the only two skills suitable for such objects). In contrast, the depth image dataset includes other skills (except flip) and a large variety of objects.

C. Training the Skill Detection Models

Using the above-mentioned dataset, we trained two Skill Detection Models: one depth-based and one RGB-based. These models can identify the appropriate skill for each object in the scene and determine its success confidence based on contextual information. Both models utilize the following architecture.

We adopted Mask R-CNN [12] implemented via Detectron2 [13] for our architecture choice. Mask R-CNN is a convolutional neural network designed for instance segmentation, enabling the identification and delineation of objects at a pixel level within images. It comprises several key components: a backbone network for feature extraction, a Region Proposal Network (RPN) for generating regions of

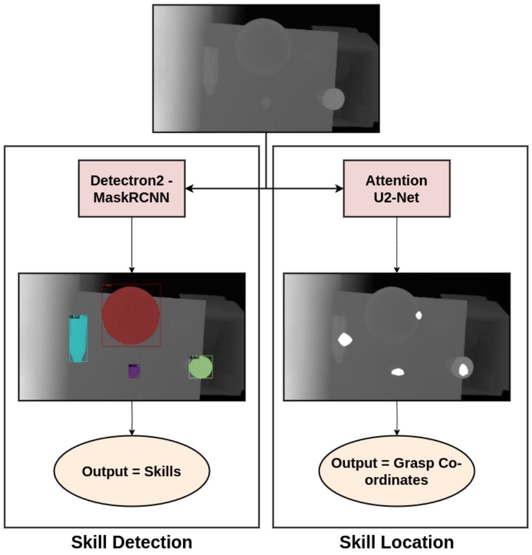


Fig. 2. Visualization of Skill Detection and Skill Location heatmap. Red Mask: Slide-to-Edge, Blue Mask: Push-to-Vertical, Purple Mask: Push-to-Horizontal, Green Mask: Simple-Pick. The top image shows scene captured by the eye-in-hand depth camera. The left side of the image corresponds to the wall in the scene and the middle part is the robot workspace.

interest, and ROI Alignment for precise mapping. It also supports multi-task learning with diverse loss functions and can adapt to custom tasks through transfer learning.

The architecture is well-tailored for our task as it adeptly captures global context, by extracting the relevant features from the entire image. By leveraging these features, the RPN identifies regions of interest, allowing us to segment objects corresponding to different manipulation skills. Our implementation involves decomposing an input depth image into segmentations aligned with the five skills and an extra class designated for the “no object” category.

For the optimization of our skill detection model, we used the Stochastic Gradient Descent (SGD) method, setting the learning rate at 0.00025, momentum at 0.9, and implementing a weight decay of 0.0001. To enhance the training efficiency, the model’s ResNet50 backbone was initialized using pre-trained weights from the ImageNet dataset.

D. Training the Skill Location Models

For training the Skill Location Models, we utilized the above-mentioned manually-labeled dataset, using the U2Net [14] architecture combined with an attention-gate to generate a heatmap corresponding to the optimal grasp regions. The proposed approach fuses the previous encoder and the decoder features maps via the attention gate, which helps enhance the target regions while suppressing the irrelevant features. This improved feature map and the previous decoder output are then passed on to the next decoder for aggregating features. The introduction of the attention gate helps generate a more focused and rich representation, enhancing our model’s generalizability to unseen objects.

In the implementation of the Attention U2Net model, we opted for the Adam optimizer, setting $\beta 1 = 0.9$ and $\beta 2 = 0.999$, along with a learning rate set to 0.001. The training process utilizes a composite loss function, which encompasses the complement of the Structural Similarity Index (SSIM), Intersection over Union (IOU), and Binary Cross Entropy(BCE) loss. The overall loss function is given by:

$$\begin{aligned} L = \mathrm{BCE}_{\text{pred,target}} + (1 - \mathrm{SSIM}_{\text{pred,target}}) \\ + \mathrm{IoU}_{\text{pred,target}} \end{aligned} \tag{1}$$

where:

- BCE: The Binary Cross Entropy loss, which measures the pixel-wise error in binary classification between the predicted and target heatmaps.
- SSIM: The Structural Similarity Index, which assesses the similarity between the predicted and target heatmaps. We use its complement, $1-\mathrm{SSIM}$, to penalize dissimilarity and enhance structural consistency.
- IoU: The Intersection over Union loss, which measures the overlap between the predicted and target regions, penalizing any mismatch.

E. End-to-end pipeline for Table Clearing

We utilize the above-mentioned models for a table-clearing task, where the robot systematically addresses the complexity of scenes with multiple objects. In this task, the robot is expected to resolve the challenges of the multi-object scenes by assigning the appropriate skills to the object/situations, prioritizing the objects with the highest likelihood of picking success, and continuing a cyclic picking process until every object is successfully picked. The process begins with capturing a depth image and sending it to the depth-based Skill Detection model, where every detectable object is segmented and assigned an appropriate skill and confidence value.

The system focuses on the object with the highest skill confidence and runs the corresponding Skill Location model, which returns a heatmap. Choosing the highest value on the heatmap as the skill application location, the algorithm triggers the execution of the chosen skill. The subsequent phase involves motion planning for the open-loop execution of the skill (adjusted with the identified skill application location). In our implementation, we utilized Relaxed IK [15], which generates robust and smooth IK solutions. Notably, slide-to-edge and push-to-vertical skills impose particular constraints on the end effector’s orientation, which must be maintained during the Cartesian motion.

Our approach does not utilize a 6D object pose to handle the orientation of the end effector. Instead, we predefine end-effector orientation values for each skill except simple-pick. For instance, consider the “push-to-vertical” skill, which consists of four execution steps: approaching the object, pushing the arm towards the object, closing the gripper, and placing the object in the bin. For each step, we have predefined the roll, pitch, and yaw values of the end effector. Whereas for the Simple-pick, we utilize the yaw given by the skill detection model, which is calculated using the orientation of the segmentation mask. The arm maintains these orientation values while adjusting to the variable position values provided by the skill location model during skill execution.

In the execution of the slide-to-edge and push-to-vertical skills, we consider two motion parameters. The direction of movement for slide-to-edge is determined to be the opposite of the normal vector of the table edge. Similarly, the movement direction for push-to-vertical is opposite of the normal vector of the wall. To calculate the distances that need to be covered in the specified directions, assume that the positions of walls and table edges are predefined and known. This allows us to calculate the motion parameters accurately by measuring the distance from these fixed environmental features to the locations identified for skill application, ensuring the effective execution of each skill.

This process continues until all the objects segmented by the depth image are cleared off of the table. Nevertheless, there can be objects that are too small and/or too flat to be detected by the depth image (e.g. a coin). To clear these objects, we run the RGB-based models and follow the same iterative process, which involves executing the skills sequentially, starting with the one identified with the highest confidence, until all objects have been successfully picked from the table.

IV. Experiments

Our experimental setup uses a Franka Emika Panda 7-DOF robot arm integrated with an OpenHand Model-O three-fingered gripper affixed to its end effector. The camera is attached to the robot's end effector, capturing images from a consistent height of 0.81 m above the tabletop. We have delimited our workspace dimensions to 41x41 cm^2, considering the workspace of the robot and the motions that are required to apply skills. The computational setup is an Intel i7-8400 CPU with an NVIDIA GeForce GTX 2080 GPU, running Ubuntu 20.04 operating system and ROS Noetic.

A. Evaluation of the Models

We employ the accuracy score metric to assess the performance of our skill detection model, while the Mean Absolute Error (MAE) and Intersection over Union (IOU) are utilized to evaluate the skill location model as presented in Table I. To obtain the skill detection accuracy, we utilized a training set of 570 images and a separate test set of 84 images. We calculated the accuracy for the skill location model by an Intersection over Union (IOU) threshold, which was set to 0.5. We also present the average IOU. Both of our models provide over 92% accuracy.

TABLE I
Performance Evaluation of Skill Detection and Skill Location Models

Model	Metric	Value
Skill Detection	Accuracy	92.85
	mAP	76.076
Skill Location	MAE	0.027
	Accuracy	92.85
	Average IOU	0.66

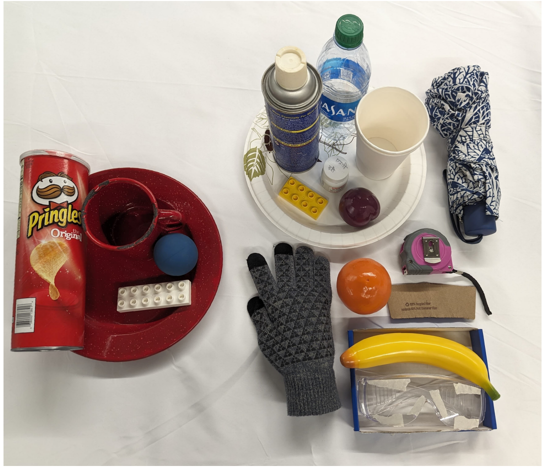


Fig. 3. Objects used in our experiments. The five objects on the left are "known" objects utilized in the model training process. The other ten objects on the right are "unknown" objects not included in the dataset.

B. Real-world experiments

We implemented the pipeline in Section III-E for clearing a tabletop with vertical walls. The experiments were conducted using both known and unknown objects, representing a range of shapes, sizes, stiffness, and materials as presented in Fig. 3. The five known objects are chips can, mug, racket ball, metal plate, and lego piece, which were used in our dataset for training skill detection and skill location models. All the other 15 objects in Fig. 3 are unknown to the system a priori. We conducted experiments for three different scenarios of increasing difficulty:

- Experiments with only known objects.
- Experiments with both known and unknown objects.
- Experiments with only unknown objects.

While designing these experiments, we included many challenging scenarios, e.g. where certain skills were obstructed by the placement of objects, instances where objects were positioned directly against a wall, and cases involving overlapping objects, with some resting atop others. These experiments comprised 45 multi-object manipulation scenes with over 150 grasping trials, with a minimum of 10 tests conducted for each object. The complete set of results for every single experiment is provided as a table in the supplementary document. The videos of all the experiments are provided in [5].

The results of the experiments can be found in Table II. Here, we have included an additional row for exclusively presenting the experiments with overlapping (occluded) objects within the known+unknown and unknown object experiments. It is observed that the system is successful in picking known objects even in challenging scenarios, and it still presents high success rates when unknown objects are introduced. In cases where not all objects were cleared from the table, the rate of object picking remained satisfactory. Table III further breaks down these results by the number of

TABLE II

EXPERIMENT-SPECIFIC GRASPING RESULTS

Experiment Type	Successful Trials	Average % of table cleared
Known	5 / 5	100%
Known + Unknown	16 / 20	90.83%
Unknown	15 / 20	93.9%
Occluded	8 / 10	91.6%

TABLE III

OVERALL TABLE CLEARING SUCCESS RATES

Number of objects	Successful Trials / Total # of Trials	Average Grasped Objects Count
2	4 / 5	1.6
3	13 / 15	2.86
4	13 / 17	3.76
5	4 / 5	4.8
6	1 / 2	5.5
7	1 / 1	7

TABLE IV

OVERALL GRASPING SUCCESS RATES AND SKILL-SPECIFIC BREAKDOWN

Skills	Success Rate	Algorithmic failures	Mechanical Failures
Push-to-Vertical	92%	-	Obj. slipped during grasping
Push-to-Horizontal	85.29%	Wrong skill detected	Obj. slipped during grasping.
Slide-to-Edge	66.66%	Optimal skill location not found.	Unachievable robot pose
Simple-Pick	92.5%	-	Environmental interference.
Flip	71.88%	Optimal skill location not found.	Object too small

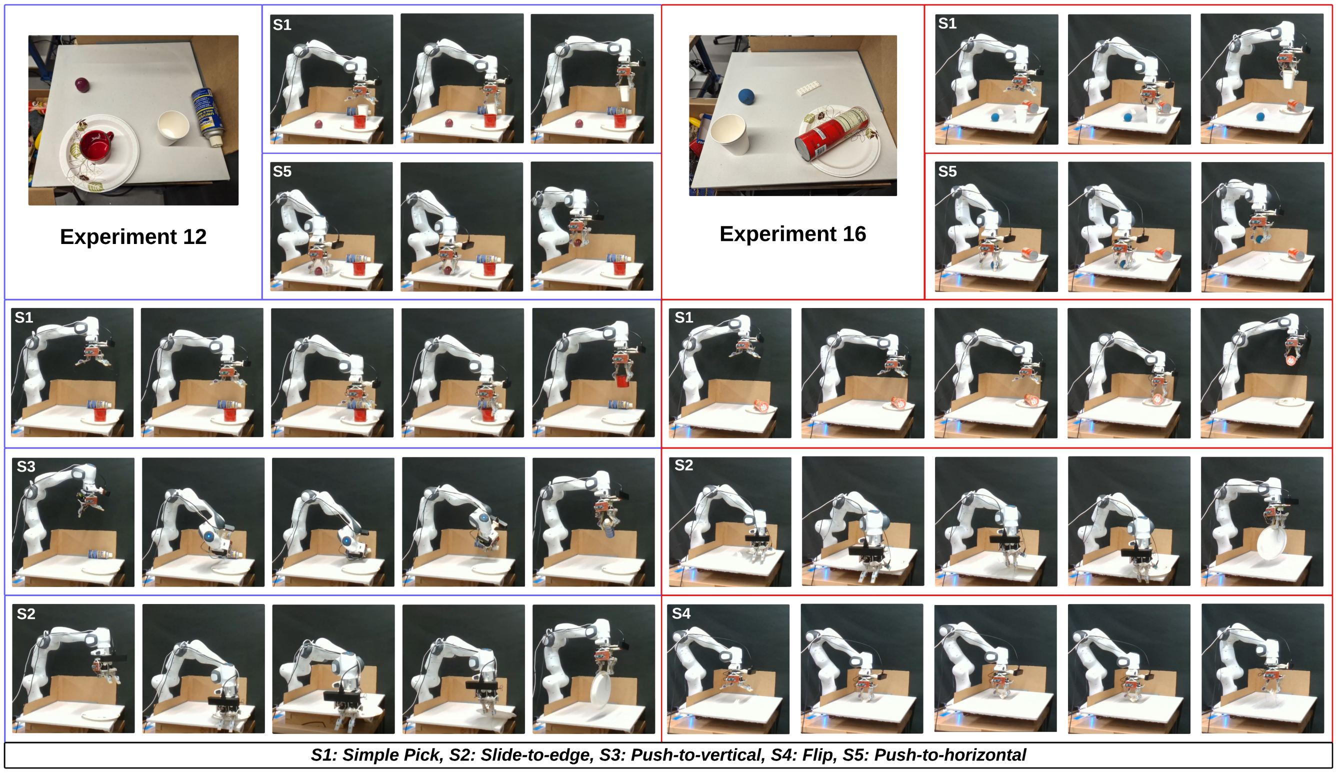
objects present in each scene. The findings consistently show that the system effectively picks a high average number of objects, even as the scenes become more cluttered.

Table IV offers a detailed analysis of the picking success associated with each specific skill, including common failures observed during the experiments. Note that we allowed for one retry if a skill fails for the first time and the object is still on the table. We did not do anything special for this retry mechanism; since the pipeline works cyclically, it captures a new image of the scene and runs the process again. The success rates reported in Table IV also reflect those retries. An important observation here is the lower success rate of the slide-to-edge skill. The failures are partially due to the failure of the skill location model, which, we believe, can be fixed by adding more training data for this skill. We have also observed that in some cases, even though the sliding action was successful, the robot was unable to complete the pick due to the execution of the grasping motion. Specifically for slide-to-edge, 10 out of 26 picks failed in the first attempt. However, in 6 of the 10 cases, the system was successful in the second attempt. All the experimental details can be found in our supplementary document.

The flip skill also exhibits a lower success rate, primarily due to occasional inaccuracies in predicting the precise location for skill application. Given that objects targeted for flipping are typically small, even minor deviations in the identified skill location can lead to unsuccessful attempts, necessitating retries.

Discussing Specific Experiments: We discuss the steps involved in executing 2 specific experiments in detail as shown in Fig. 4. In Experiment 12, the manipulation scenario involved a variety of objects necessitating different robotic skills, and the system/trained model automatically determined and applied a successful skill sequence for clearing these items. The notable aspects of this experiment were the positioning of the mug on a plate, the cylinder being flushed to the wall, and the white cup blocking a potential application of a push-to-vertical skill. The robot strategically started with a simple-pick skill for the white cup, followed by a push-to-horizontal skill for the unobstructed plastic plum to declutter the scene. Next, the robot used a simple-pick to clear out the red mug. This decision was important as it freed up space to apply the push-to-vertical skill to the bottle. Consequently, in the next step, the system chose push-to-vertical to pick the bottle, and finally, it chose slide-to-edge to pick the plate. These automatic skill decisions demonstrate the efficacy of prioritization: the system chose to declutter the table top by prioritizing easier cases, unblocked the skill application locations by removing the obstructing objects, and successfully handled overlapping objects, highlighting the precision of the Skill Location Module. In all the executions of this experiment, depth-based models are used.

A similar skill selection and execution capability is demonstrated in experiment 16. Here, the system again chose to declutter the scene by prioritizing objects that are not blocked by others, i.e. the white cup and blue ball with simple-pick and push-to-horizontal respectively. The robot then picked up the cylindrical chips box with simple-pick which freed up the plate. This decision demonstrates a key capability to select appropriate skills based on the context: when a similar object was flushed to the wall in Experiment 12, the system chose push-to-vertical skill. However, in Experiment 16, simple-pick was a more appropriate choice since this skill can easily be applied without any obstructions. This and similar variations in skill determination based on the object's location demonstrate the system's nuanced understanding of context and its ability to dynamically adapt its strategy to maximize efficiency and success rate. Experiment 16 also demonstrates the utility of the RGB-based models. After all the objects that can be identified by the depth



Experiment 12
Experiment 16
S1
S5
S1
S5
S1
S1
S3
S2
S2
S4
S1: Simple Pick, S2: Slide-to-edge, S3: Push-to-vertical, S4: Flip, S5: Push-to-horizontal

Fig. 4. Individual steps for performing grasps. Left column: Experiment 12, Right column: Experiment 16. Grasping sequence from left to right

models were picked, the white Lego piece remained on the table. This object was not detected by the depth-based models due to its small size and flatness. Switching to the RGB model, the system was able to choose the flip skill and pick the Lego piece.

C. Limitations:

We observed the following limitations of our system:

- Having separate models for RGB and depth images occasionally creates problems. Since the depth model is run first, and it is not able to pick very small and/or flat objects, if these objects are on the way of the slide-to-edge actions, the small objects are swept off the table. Similarly, if these objects are on top of other objects, they go undetected and get dropped off the table. These scenarios underscore the limitations of using separate models. In our future work, we aim to develop a unified model that could effectively replace the distinct RGB and depth modules, resulting in a more efficient framework.
- Currently, our models are only trained for a specific table configuration; the locations of the walls and the table edges are the same in all training data and experiments. We believe that the models can generalize to other environment configurations if more diverse training data is collected and utilized. Similarly, our system is only trained for a top-down camera viewpoint. While this constraint might also be relaxed by more training data, other challenges might arise for side viewpoints, such as object occlusions.
- Some of our skill executions are not implemented very effectively. Even though the right skills and seemingly right locations are chosen on the objects, the motions of the slide-to-edge and flip skills sometimes require retries. Currently, the manipulator operates under open-loop control, without utilizing visual feedback during motion execution. Although the camera is mounted on the gripper, the images captured along the arm's trajectory are not used to adapt the gripper's position in real-time. Implementing closed-loop control strategies that leverage real-time visual feedback from the gripper-mounted camera could significantly enhance the precision and efficiency of our skill executions.

V. Conclusion

In our research, we have tried to demonstrate a system that can introduce human-like dexterity. The dexterous skills can successfully solve complex multi-object manipulation problems. The approach can be further generalized to add more skills to the existing system. While the current system only trained for a specific environment configuration, a larger and more diverse dataset, encompassing different scenes, object configurations, and camera perspectives, could generalize the system to handle a broader range of real-world scenarios. An interesting finding is that even though we had no overlapping objects in our dataset, the system was successful in handling such cases.

While the system has an inherent retry capability, we believe that, by monitoring and analyzing the outcome of each grasp attempt (e.g. by utilizing force sensing), it would be possible to create a better error-handling mechanism. For example, the system can ascertain whether the selected skill was executed successfully, and refine its decision-making process over time. It is also worth looking into closing the loop for the motion execution of the dexterous skills.

References

[1] C. D. Santina, V. Arapi, G. Averta, F. Damiani, G. Fiore, A. Settimi, M. G. Catalano, D. Bacciu, A. Bicchi, and M. Bianchi, "Learning from humans how to grasp: A data-driven architecture for autonomous grasping with anthropomorphic soft hands," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1533–1540, 2019.

[2] Y. Li, P. Wang, R. Li, M. Tao, Z. Liu, and H. Qiao, "A survey of multifingered robotic manipulation: Biological results, structural evolvements, and learning methods," *Frontiers in Neurorobotics*, vol. 16, 2022.

[3] H. Xiong, Q. Li, Y.-C. Chen, H. Bharadhwaj, S. Sinha, and A. Garg, "Learning by watching: Physical imitation of manipulation skills from human videos," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 7827–7834.

[4] C. Eppner, R. Deimel, J. Álvarez Ruiz, M. Maertens, and O. Brock, "Exploitation of environmental constraints in human and robotic grasping," *The International Journal of Robotics Research*, vol. 34, no. 7, pp. 1021–1038, 2015.

[5] Supplementary Material, 2024. [Online]. Available: https://dexterouspicking.wpi.edu/

[6] L. U. Odhner, L. P. Jentoft, M. R. Claffee, N. Corson, Y. Tenzer, R. R. Ma, M. Buehler, R. Kohout, R. D. Howe, and A. M. Dollar, "A compliant, underactuated hand for robust manipulation," *The International Journal of Robotics Research*, vol. 33, no. 5, pp. 736–752, 2014.

[7] L. Odhner, R. R. Ma, and A. M. Dollar, "Open-loop precision grasping with underactuated hands inspired by a human manipulation strategy," *IEEE Transactions on Automation Science and Engineering*, vol. 10, pp. 625–633, 2013.

[8] L. Collodi, D. Bacciu, M. Bianchi, and G. Averta, "Learning with few examples the semantic description of novel human-inspired grasp strategies from rgb data," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2573–2580, 2022.

[9] A. Zeng, S. Song, K.-T. Yu, E. Donlon, F. R. Hogan, M. Bauza, D. Ma, O. Taylor, M. Liu, E. Romo, N. Fazeli, F. Alet, N. C. Dafle, R. Holladay, I. Morena, P. Qu Nair, D. Green, I. Taylor, W. Liu, T. Funkhouser, and A. Rodriguez, "Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 3750–3757.

[10] Y. Lin, C. Tang, F.-J. Chu, and P. A. Vela, "Using synthetic data and deep networks to recognize primitive shapes for object grasping," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 10 494–10 501.

[11] L. Odhner, R. R. Ma, and A. M. Dollar, "Precision grasping and manipulation of small objects from flat surfaces using underactuated fingers," *2012 IEEE International Conference on Robotics and Automation*, pp. 2830–2835, 2012.

[12] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2980–2988.

[13] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," Available: https://github.com/facebookresearch/detectron2, 2019.

[14] X. Qin, Z. Zhang, C. Huang, M. Dehghan, O. R. Zaïane, and M. Jägersand, "U^2-net: Going deeper with nested u-structure for salient object detection," *CoRR*, vol. abs/2005.09007, 2020.

[15] D. Rakita, B. Mutlu, and M. Gleicher, "RelaxedIK: Real-time Synthesis of Accurate and Feasible Robot Arm Motion," in *Proceedings of Robotics: Science and Systems*, Pittsburgh, Pennsylvania, June 2018.