ShapeSync - Autonomous Shape Insertion Robot

Kartik Agrawal[1], Simson D'Souza[2], Rohit Satishkumar[3] and Shivang Vijay[4]

Abstract— **This report presents an autonomous shape insertion system developed using a Franka Emika 7-DOF robotic arm equipped with a gripper and an Intel RealSense depth camera mounted on the end-effector. The objective is to enable efficient pick-and-place operations where the robot autonomously detects, grasps, and inserts geometric objects into corresponding slots within a stencil. The system is composed of integrated perception and control pipelines that ensure accurate object recognition, pose estimation, and precise manipulation.**

The perception subsystem uses a YOLOv8-based object detection model trained on a custom dataset of 2D images representing various geometric shapes. To achieve precise grasping, the system employs an HSV-based masking technique to accurately determine the knob center of each object, improving the estimation of 3D poses. These coordinates are then transformed into the robot's base frame using intrinsic parameters for reliable grasp planning. Orientation is estimated using an ArUco marker placed at the block's bottom surface, enabling correction of rotational misalignment during insertion.

The control system focuses on ensuring precise alignment during insertion by correcting the orientation of the robot's end-effector. The perception pipeline estimates the object's orientation using an ArUco marker, and the orientation error with respect to the desired placement is computed. A PID controller is then used to correct this error, aligning the object appropriately to fit into the stencil slot.

Experimental evaluations demonstrate the system's ability to reliably insert objects into stencil slots with high precision. This autonomous shape insertion system is applicable in various fields, such as automated assembly, quality control, and precision manufacturing, where accurate object placement is critical.

I. INTRODUCTION

A. Background and Motivation

While object manipulation tasks such as shape insertion are relatively straightforward for humans, they pose significant challenges for robots due to the complexity of accurately identifying, selecting, and placing objects based on their shape and size. In applications such as automated assembly or precision manufacturing, the ability to autonomously select the right shape and insert it into a predefined slot with high accuracy is critical. These tasks require seamless integration of perception and control systems, robust decision-making capabilities, and real-time response to dynamic changes in the environment.

The task addressed in this project involves autonomously picking up different geometric objects and inserting them into their corresponding slots within a stencil. To achieve this, the system utilizes a Franka Emika 7-DOF robotic arm with a gripper and an Intel RealSense depth camera mounted on the end-effector. The camera captures RGB-D data to support object detection, 3D pose estimation, and orientation analysis. Shape classification is performed using a YOLOv8-based deep learning model, while the object's orientation is determined using ArUco markers placed on the bottom surface of each block. The control subsystem complements this perception pipeline by computing the orientation error between the held object and the target slot, and uses a PID controller to correct the end-effector's orientation, enabling accurate alignment and insertion into the stencil.

This project aims to tackle the challenges of shape detection, object selection, and precise insertion while ensuring that the robot adapts to various object configurations and can recover from potential failures during the task. The integration of perception techniques enhances the robot's efficiency and allows it to operate in real-time, adjusting to changing conditions within its workspace. This project explores how the Franka Emika robotic arm, in combination with vision-based perception, can autonomously perform shape insertion tasks, providing valuable insights for future developments in robotic automation, manufacturing, and assembly systems.

B. Contributions

The majority of the work in this project was done collaboratively, with a significant focus on the perception subsystem, which included object detection and classification, pose estimation, and orientation correction using a PID controller. While all components were developed through team collaboration, each member took primary responsibility for specific tasks. Simson led the development of the YOLO-based object detection pipeline, pose estimation, contributed to orientation correction, and integrated the perception subsystem. Kartik led pose estimation, orientation correction using a PID controller, and the integration of the perception and control subsystems. Rohit also led pose estimation, focusing on tuning and validation.

II. RELEVANT WORK

In this section, we discuss the recent work performed in the shape-insertion task, as well as its variants, using

This work was supported by Dr. Oliver Kroemer and the teaching assistants for the Robot Autonomy Course at Carnegie Mellon University

[1] Kartik Agrawal is a Masters Student in Robotic Systems Development within the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213,USA

[2] Simson D'Souza is a Masters Student in Robotic Systems Development within the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213,USA

[3] Rohit Satishkumar is a Masters Student in Robotic Systems Development within the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213,USA

[4] Shivang Vijay is a Masters Student in Robotic Systems Development within the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213,USA

manipulators.

The shape insertion task using robotic arms and its variants such as peg insertion and deformable object insertion is a well-studied topic in the literature. For example, Kim et al.[1] performed peg insertion into a shallow hole, using the dexterous manipulation action of their robotic fingers. Kang et al.[2] used a compliant control method with contact localization for insertion of complex objects with concavities. Spector et al. [3] formulate shape insertion as a regression problem with visual and force inputs to perform visuo-tactile shape insertion for complex assemblies.

Recently, many works have been published that incorporate deep learning and reinforcement learning techniques for robotic shape insertion. Furthermore, there has been a rise in the works that integrate both vision and tactile modalities for precise shape insertion. For example, Yan et al. [4] propose the usage of a transformer-based deep learning architecture to estimate the error distance from the correct insertion position using tactile information. The use of reinforcement learning for complex shape insertion tasks has also been on the rise, as seen in [5], [6], and [7]. In the new age of AI, the use of learning based techniques have been instrumental in performing complex tasks with good precision.

In this report, we present our work on shape insertion of blocks onto stencils, which can be seen as a preliminary task to various real-life insertion tasks in various industries. We incorporate computer vision techniques for identifying the block to be inserted, compute the relative transforms between the block and the end effector for accurate grasping. Our methodology is detailed out in the next section.

III. METHODOLOGY

The system consists of two primary subsystems: the perception subsystem and the control subsystem. The goal is to enable an autonomous pick-and-place operation, where the robot selects a geometric object and inserts it into the corresponding slot in a stencil based on its shape and orientation.

The perception subsystem is responsible for detecting the shape of the object using a YOLO-based object detection model, estimating the 3D position of the object's knob center, and determining its orientation using ArUco markers. The control subsystem receives this information, computes the orientation error between the held object and the desired slot orientation, and uses a PID controller to correct the end-effector's orientation for accurate insertion.

A. Hardware Setup

The robotic setup includes a designated pickup area where objects are scattered, as illustrated in Figure 1, and a placing area containing the stencil. The robot must accurately position each picked object into its corresponding slot based on its shape and size. An Intel RealSense depth camera is mounted on the robot's end-effector, providing the necessary data for the perception system to identify the object's shape and the knob center. Additionally, a second RealSense depth camera is placed near the stencil, facing upward, to detect the ArUco marker placed on the bottom of each block for orientation estimation.

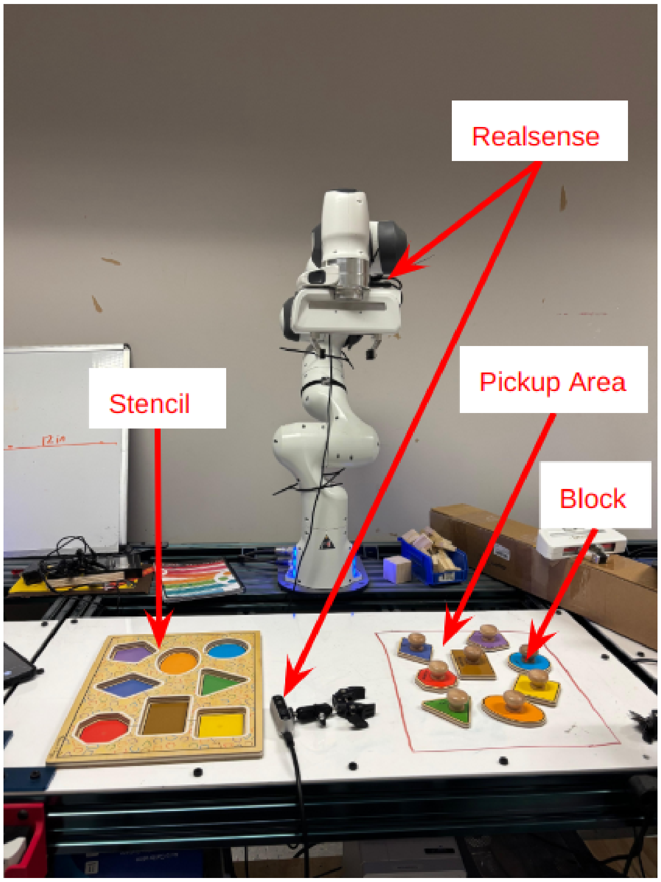

Fig. 1. Hardware Setup

B. Perception Pipeline

The perception pipeline is responsible for accurately detecting geometric-shaped blocks and estimating their poses by determining the center of the knob. This ensures precise grasping by the Franka robotic arm. Additionally, the system must detect the pose of the placement area within the stencil to facilitate correct object insertion. The Intel RealSense camera is used for real-time object detection, pose estimation and orientation correction.

1) Object Detection: The primary objective of object detection is to accurately identify geometric shapes. Initially, color masking was tested, but due to variations in lighting conditions, the results were inconsistent, making it difficult to estimate shapes precisely. To address this issue, a custom dataset was created using RoboFlow. A total of 80 images of geometric blocks and stencils were captured from different camera perspectives, considering varying lighting conditions and object positions. Each image was manually annotated, and data augmentation was performed, increasing the dataset to 192 images. The dataset was split into training (88%), validation (8%), and testing (4%) sets. The YOLOv8 model was trained on this dataset and achieved a mean average precision (mAP) of 93.5%, precision of 77.6%, and recall of 95.8% as shown in Figure 2.

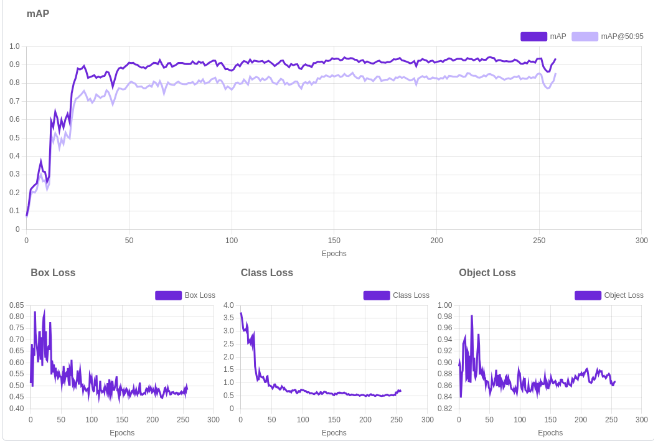

Fig. 2. Trained Object Detection Model Performance

The dataset included the following class labels: Square, Rectangle, Diamond, Circle, Oval, Pentagon, Octagon, and Triangle. Since the detection model was trained on these specific objects, it successfully detects both the geometric blocks and the corresponding stencil slots with high accuracy. The detected objects are enclosed within bounding boxes, as illustrated in Figure 3.

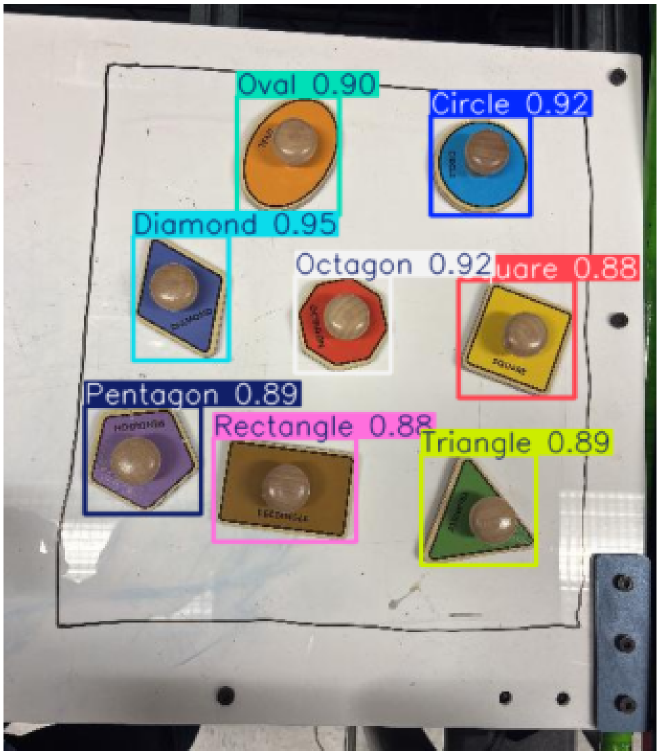

Fig. 3. Geometric Objects Detection

2) Pose Estimation: To ensure precise grasping, the center of the geometric object must be accurately determined. The Franka arm first moves to a predefined position directly above the scattered objects to obtain a bird's-eye view. Relying solely on the bounding box center can lead to inaccuracies, especially when objects are positioned far from the image center. Hence, once the objects are detected, a color-masking technique is applied to isolate the knob region within the bounding box. This is achieved by filtering HSV color values to identify the knob's brown color. To improve precision, OpenCV is used to draw a blue circle around the detected knob based on the color mask. The center of this circle is then computed as the object's true center. By carefully tuning the HSV values, the system accurately determines the knob's position.

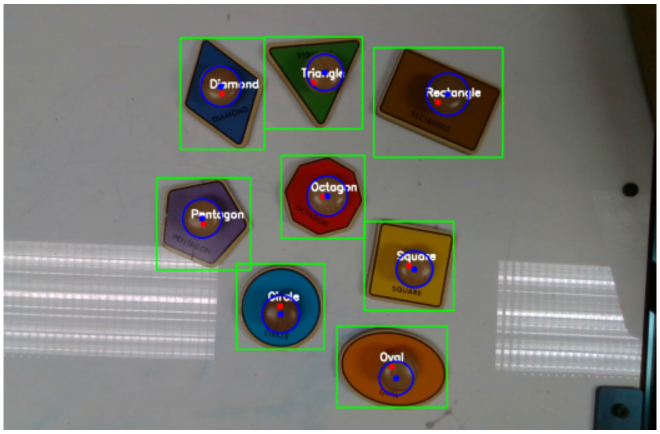

Fig. 4. Geometric Objects Detection with knob center

Figure 4. illustrates the RealSense camera's captured image from the predefined position, where blue circles and dots indicate the correctly computed object centers, red dots represent the bounding box centers, showing visible discrepancies and the blue dots (masking-based method) demonstrate better accuracy in detecting the knob's center. Once the image coordinates of the detected object centers are obtained, they must be converted into 3D coordinates (x, y, z). This is done using the camera's intrinsic parameters to transform the image coordinates into the camera frame and subsequently to the base frame. The transformation provides both the position and orientation (rotation and translation) required for the Franka arm to pick up the object. A similar approach is applied to detect the placement locations within the stencil, ensuring accurate positioning of each geometric object in its respective slot.

3) Orientation Estimation and Correction: After the object is picked up, the Franka arm moves it to a fixed orientation correction station. At this location, a second Intel RealSense depth camera is mounted on the table, facing upward toward the bottom of the block. Each object has an ArUco marker placed on its underside, which is used to estimate the current orientation of the block.

Once the ArUco marker is detected, the system computes the block's orientation relative to the camera frame. Since the orientation of the stencil slot is known beforehand, this detected orientation is compared to the target orientation. The difference between them, known as the orientation error, is calculated and used to correct the block's pose.

With the corrected orientation and the previously estimated position, the robot is able to place the object accurately into its designated slot in the stencil. This process ensures precise alignment during insertion and reduces the likelihood of failure due to misorientation.

Figure 5 illustrates the overall workflow of the perception subsystem, including both pose estimation and orientation correction stages. It highlights how image-based detection, knob center extraction, and ArUco marker-based orientation alignment are integrated to enable accurate shape pickup and placement.

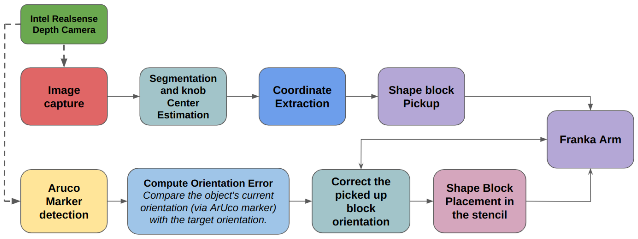
Intel Realsense Depth Camera
Image capture
Segmentation and knob Center Estimation
Coordinate Extraction
Shape block Pickup
Franka Arm
Aruco Marker detection
Compute Orientation Error
Compare the object's current orientation (via ArUco marker) with the target orientation.
Correct the picked up block orientation
Shape Block Placement in the stencil

Fig. 5. Perception Subsystem Workflow

C. Controls Pipeline

The control pipeline is responsible for aligning the object held by the robot's end-effector with the corresponding stencil slot for successful insertion. The perception subsystem provides the 6D pose of the object, including its orientation, using an ArUco marker placed on the bottom surface of the block. The desired orientation of the stencil slot is predefined, and the control subsystem computes the orientation error between the detected and desired poses.

To correct this misalignment, a PID controller is implemented to adjust the end-effector's orientation based on the calculated error. This correction is critical, as even minor deviations can prevent successful insertion due to the tight tolerances between the blocks and the stencil slots. While the controller improves robustness, the Franka Emika arm's limited millimeter-level precision still results in occasional failures. To address this, we also considered mechanical tolerancing improvements such as filing the block edges to compensate for small alignment errors.

We tuned the PID controller to achieve more precise orientation correction and improve placement accuracy in the stencil.

IV. Evaluations

To evaluate our system, we conducted a series of pick-and-place experiments with various geometric shapes. The stencil location was fixed within the workspace, while the blocks to be picked were randomly placed in the pickup area, with arbitrary orientations, as long as they remained visible within the Intel RealSense camera's field of view.

The object detection performance was highly accurate, thanks to the YOLOv8 model trained on our custom dataset. It significantly outperformed traditional color masking methods, especially under varying lighting conditions. However, detecting the knob center remained challenging at times due to illumination inconsistencies. By carefully tuning the HSV values, the system was able to accurately identify the knob's position, as illustrated in Figure 6.

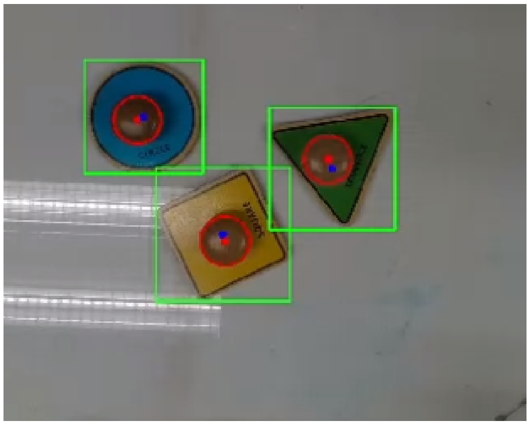

Fig. 6. Geometric Objects and Knob center detection

The Franka arm was generally able to grasp the object at the knob with good accuracy. However, there were occasional failures where the robot missed the object despite correct knob center estimation. This typically occurred when the object was placed near the Franka arm's virtual walls, minor deviations in the arm's trajectory near these constraints could lead to missed pickups, as demonstrated in the full system demonstration video.

Following the pickup, the next step involved orientation detection and correction. This was one of the most challenging aspects early in development, as misalignments frequently caused insertion failures. Instead of relying on fixed orientations, we aimed to make the robot capable of autonomously correcting orientation errors. The implementation of the PID controller significantly improved this process. After tuning the controller parameters, we achieved much more precise orientation correction and robust placement, particularly for circular shapes, as shown in Figure 7.

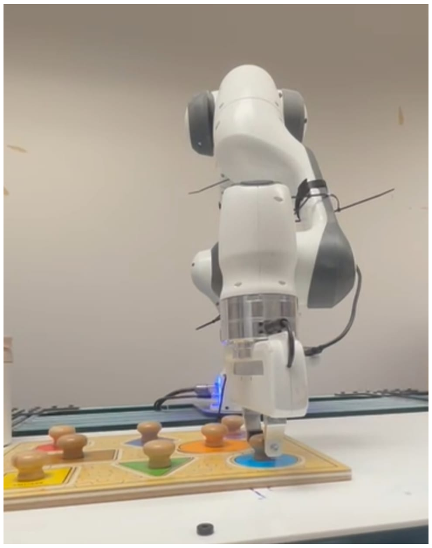

Fig. 7. Circle Block placement onto the stencil

For more complex shapes, however, insertion remained unreliable even when pickup and orientation correction were successful. This was primarily due to minimal clearance between the object and stencil slot, which demanded extremely precise positioning. The Franka arm, while capable, does not consistently achieve millimeter-level repeatability required for such tight fits, making accurate insertion particularly difficult in these cases.

V. Challenges

We encountered several challenges throughout the implementation of our system. One of the major issues was inconsistent lighting conditions, which significantly affected the performance of the perception subsystem. In particular, lighting variations caused the HSV-based knob center detection to occasionally fail, resulting in inaccurate center estimation despite accurate object detection by the YOLO model.

Another critical challenge was the variance between the desired and actual end-effector pose of the Franka arm. Due to the tight tolerances required for stencil insertion, even small errors in position or orientation could lead to failure. The Franka arm does not consistently provide millimeter-level precision, which is essential for this application, especially when inserting complex shapes with minimal clearance.

Additionally, some stencil blocks, such as rectangles and ovals, had knob colors that were visually similar to the blocks themselves. This reduced the effectiveness of color masking, further complicating the accurate localization of knob centers.

These sources of error ranging from perception inaccuracies to hardware limitations—accumulated and occasionally resulted in failed pick-and-place operations, particularly during insertion.

VI. Future Work

This project was a successful demonstration of autonomous shape insertion, with a robust perception subsystem capable of accurately detecting, classifying, estimating pose, and correcting orientation. It was a valuable learning experience for the team and highlighted several areas for further improvement and development. Our future work will focus on the following:

- **Filing Shapes for Better Insertion:** To mitigate the small translation and orientation deviations introduced by the manipulator's limited precision, we plan to file the edges of the blocks. This will introduce slight tolerance and increase the likelihood of successful insertions.
- **Motion Planning for Collision-Free Placement:** Currently, the robot follows a simple trajectory that occasionally leads to collisions with objects after placement. Implementing a motion planning algorithm will help generate smooth, collision-free paths for both pickup and placement, improving overall reliability.
- **Improved Control Algorithms for Precision:** Given the millimeter-level precision required by this task, we aim to explore more advanced control strategies beyond PID to minimize pose deviation and enhance accuracy during insertion.
- **Shape Selection Based on Fit:** A potential extension is to introduce multiple similar-shaped blocks with varying dimensions. The robot would then be required to select the best-fitting shape for a given stencil slot using perception and reasoning.

VII. Conclusion

ShapeSync is an autonomous robotic system designed to detect, grasp, and insert geometric shapes into a stencil with high accuracy. Leveraging a YOLOv8-based object detection pipeline, HSV-based knob center estimation, and ArUco marker-based orientation correction, the system effectively combines computer vision and robot control. A PID controller is used to align the end-effector's orientation for precise insertion.

The perception subsystem proved to be robust and reliable, enabling accurate detection and pose estimation under varying conditions. While the control system performed well for simpler shapes, insertion failures occurred in cases demanding millimeter-level precision, highlighting the limitations of the manipulator.

Overall, the project demonstrates the feasibility of integrating vision-based perception with real-time control for shape-based object manipulation. It provides a strong foundation for future improvements in motion planning, advanced control, and adaptive shape selection.

VIII. Acknowledgement

We sincerely thank the Robotics Institute at Carnegie Mellon University for providing the infrastructure, tools, and collaborative environment that made this project possible. We are especially grateful to Professor Oliver Kroemer for his insightful guidance and encouragement throughout the course of the 16-662 Robot Autonomy class. His feedback played a vital role in shaping the direction of our work.

We would also like to extend our thanks to the teaching assistants, Madhusha Goonesekera and Shrudhi Ramesh Shanthi, for their consistent support and constructive suggestions during development. Their involvement was instrumental in helping us overcome technical challenges and refine both our perception and control pipelines.

References

[1] Kim, Chung Hee, and Jungwon Seo. "Shallow-depth insertion: Peg in shallow hole through robotic in-hand manipulation." IEEE Robotics and Automation Letters 4.2 (2019): 383-390.

[2] Kang, Sung C., et al. "A compliant motion control for insertion of complex shaped objects using contact." Proceedings of International Conference on Robotics and Automation. Vol. 1. IEEE, 1997.

[3] Spector, Oren, and Dotan Di Castro. "Insertionnet-a scalable solution for insertion." IEEE Robotics and Automation Letters 6.3 (2021): 5509-5516.

[4] Yan, Gang, et al. "Exploratory Motion Guided Tactile Learning for Shape-Consistent Robotic Insertion." 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2024.

[5] Liu, Zihao, et al. ”Tactile Active Inference Reinforcement Learning for Efficient Robotic Manipulation Skill Acquisition.” 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2024.

[6] Hoang, Tai, et al. ”Geometry-aware RL for Manipulation of Varying Shapes and Deformable Objects.” arXiv preprint arXiv:2502.07005 (2025).

[7] Li, Mingen, and Changhyun Choi. ”Learning for Deformable Linear Object Insertion Leveraging Flexibility Estimation from Visual Cues.” 2024 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2024.