Robotic Bin Picking for Entangled Objects

XINYI ZHANG

SEPTEMBER, 2023

Robotic Bin Picking for Entangled Objects

A dissertation submitted to
THE GRADUATE SCHOOL OF ENGINEERING SCIENCE
OSAKA UNIVERSITY
in partial fulfillment of the requirements for the degree of
DOCTOR OF PHILOSOPHY IN ENGINEERING

BY

XINYI ZHANG

SEPTEMBER, 2023

Abstract

Robotic Bin picking is a valuable task in manufacturing, aiming to automate the assembly process by utilizing robots to pick necessary objects from disorganized bins. This task eliminates the need for human workers to arrange the objects or the usage of a large amount of part feeders. Previous studies have addressed various challenges related to bin picking, such as processing visual information in heavily occluded scenes and planning grasps under rich physical interaction for dense clutter. However, when objects with complex shapes or deformable properties are randomly placed in a bin, they tend to get entangled, making it difficult for the robot to pick up individual items. This poses challenges in perception, as the robot must be capable of distinguishing between isolated objects and potentially tangled ones in a cluttered environment. Manipulation is also difficult in planning effective and general disentangling motions due to the complexity of estimating entanglement and executing real-world actions.

This dissertation introduces methods to develop unified, dexterous, and robust bin picking systems for entangled objects. The target objects include both rigid (e.g., U-bolts, S-hooks) and deformable objects (wire harnesses). My research enables the robot to flexibly perform appropriate actions based on the current observation: (1) picking objects while avoiding entanglement and (2) performing disentangling manipulation when the bin does not contain any isolated objects. The goal is to equip the robot with these two capabilities for handling cluttered, complex-properties objects that are prone to entanglement, all without relying on their models. I discuss how to design effective and dexterous motion primitives for separating entangled objects. I also investigate how to infer the implicit and explicit representations for mapping these actions to visual or haptic perception. By leveraging both analytic and

data-driven approaches, I study how to efficiently learn from real-world and simulated environments under a set of criteria.

In this dissertation, I first provide a review of the progress in robot bin picking over the decades and analyze the remaining challenges. Then, to address the problem of robotic bin picking for entangled objects, I propose methods for visually abstracting entanglement, planning skillful disentangling motions, automatically selecting picking strategies, and utilizing multiple sensory modalities. Finally, I conclude by discussing the directions for future work in this research topic.

Contents

List of Tables

List of Figures

Chapter 1

Introduction, Background and Motivation

1.1 Background of Robotic Bin Picking

Robotic bin picking is a vital task in manufacturing industry that enables a robot to pick objects individually from an unstructured bin. If we try to automate an assembly process without using bin picking, we need to prepare a large amount of parts feeders according to the number of assembly parts. In this thesis, I describe the task of robotic bin picking as the process of picking objects individually from a bin and dropping them to another bin regardless of aligning their poses.

A high-performance bin picking system relies on several subproblems such as accurate scene processing, robust grasp and motion planner, and practical system integration. Researches for robotic bin picking can be divided into **analytic** and **learning-based**. **Analytic** approach recognizes the scene and plans grasps and motion based on rigid criteria without verifying the criteria using existing experiences. **Learning-based** approaches accomplish this task in an empirical way by utilizing data. For industrial bin picking, a classical solution is to match the cluttered scene with the known object model or shape to confirm the locations or poses of each object, and plan grasps considering some constraints such as collision. On the other hand, the categorization can also be described as *model-based* and *model-free* based

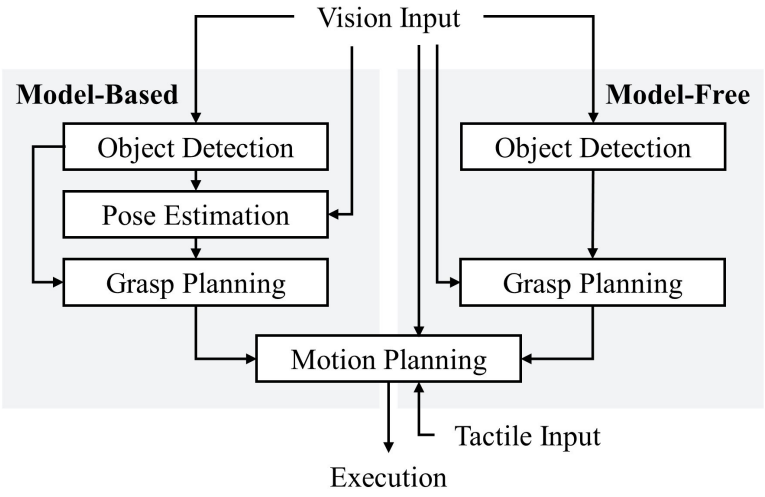


Figure 1.1: **Rough workflow of bin picking categorized as model-based and model-free approaches.** The subproblems in bin picking systems are also presented.

on whether they employ prior object knowledge or not. I roughly illustrate these two categories and the necessary subproblems in Fig. 1.1. These modules have been independently or collaboratively studies in decades using analytic or learning-based approaches. The ultimate challenge in bin picking is always handling clutter scenes while the ultimate goal is always efficiency.

1.1.1 Brief History of Analytic Bin Picking

The development of robotic bin picking is almost equivalent to the evolution of computer vision, especially in at the beginning. In the 1970s, researchers put a lot of effort to make a computer system to see objects even for occluded objects. Studies have addressed the problem of recognizing objects from the scenes containing multiple industrial parts [4, 5]. Ikeuchi and Horn [6] proposed an initial but classic bin-picking workflow, including a vision system for object localization, grasp configuration and planning and hand-eye calibration. They also provide a set of promising discussion on developing bin picking systems [7, 8, 9]. In the 1980s, computer system can now see an object at the top of a bin of mixed parts and direct a mechanical arm to pick it up. After classic algorithms for 3D object detection were born [10, 11], bin picking

method concentrated on 3D object localization for objects with simple shapes and surfaces by matching a 3D model of an object to the input scenes [12, 13, 14].

Up to the year 1990, robotic bin picking started to become mature and gradually produced some standards that can deal with more complex scenarios. During this time, bin picking methodologies have gradually diverged into three branches, image segmentation, pose estimation and picking method. In the 1990s and early 2000s, Range images were widely used so that a set of vision algorithms utilizing the height information [15, 16] .3D object detection algorithms using segmentation and pose estimation mainly focused on the level of shapes by simple features such as edges, surfaces and corners [17, 18]. With the development of shape matching algorithms that are invariant of scales or template shapes [19, 20, 21, 15], object detection in bin picking started to use high-level features and the systems became more and more robust and integrated [22, 23, 16, 24, 25]. In the 2010s, bin picking methods have been rapidly developed in various field from vision to motion [26, 27, 28, 29, 30, 31]. Studies on 6D pose estimation became more popular. Voting-based pose estimation using global features were proven to be robust and efficient for bin picking performance [32, 33]. In the practical bin picking systems, modern 6D pose estimation methods are widely used [34, 35, 36]. Motion planning algorithms were also rapidly developed [37].

For the researches in picking method, grasp planning for bin picking is always popular in these decades. Some early works utlized the known object models and knowledge and estimated the location or pose of the target by the aforementioned object detection, or fitting simple shape primitives to plan grasps [24, 38, 39]. On the contrary, another approach assumes that the object model is unknown so that it can plan grasps to novel objects. Domae et al. [40] considered the relationship between the input depth image and the gripper model and proposed a method of finding collision-free grasps. For the motion planning in bin picking, classic path planning algorithms coupled with expert experiences are used in practical systems. Recently, several approaches propose more efficient and fast methods for planning trajectories for industrial bin picking [38, 41].

1.1.2 Development of Learning-Based Bin Picking

Up to the year 2010, classic bin picking solutions are widely applied in the industrial production and contribute to factory automation. Many studies also have addressed the challenging task of grasping from the clutter using daily objects. However, there still exists some difficulties that must be overcome. Conventional model-based approaches heavily rely on the results of recognition that leads to cumulative errors. There are also limited to be generalized to more complex environments considering interactions between objects. These methods also has difficulties in handling complex-shaped or deformable objects. Also, picking up novel objects is very essential but challenging tasks for bin picking.

To tackle these challenges, learning-based methods help bin picking in almost every such as pose estimation [42, 43, 44], grasp planning [45, 46, 47], hand-eye calibration [48] or task-oriented planning [49, 50]. Table 1.1 lists several bin picking system, especially those with real world experiments. We can glance the evolution for bin picking in these decades from analytic approaches to learning. Most works have addressed the problem of vision in decades. Especially, vision and grasp planning tend to be addressed as a whole issue using deep learning since the objective of bin picking is to produce grasps with the highest accuracy and stability. Model-based methods rely on accurate pose estimation to improve the grasp quality while deep learning learns metrics of grasp planning that can be empirically verified [51, 46]. Partial information of object such as shape primitives is also utilized for improve the system flexibility [39, 52]. On the other hand, the development of depth/range image acquisition contributes to industrial bin picking in last century. Depth information is useful in either 3D modelling or grasp planning for a cluttered environments. Most of the studies or even practical bin picking system in the assembly line utilize depth images as the main perception input. As the computer vision developed especially With the advent of deep learning, using RGB image can also produce great performance in object detection Especially when the target objects are daily objects, RGB images provide more essential information other than height such as colors, textures or labels for object recognition. Despite that, point cloud is useful for accurate pose estimation, which is still important in many practical industries.

Table 1.1: Approaches of Robotic Bin Picking

	System Focus				Prior Knowledge			Hand			Visual Input	Learning Involved?	Simulation Involved?
	Perception	Grasp	Motion	Specified Task	Full	Partial	Model-Free	Parallel-Jaw	Suction	Designed			
Ikeuchi et al., 1983 [6]	✓	✓				✓		✓			Photometric Stereo		
Al and Sood et al., 1990 [14]	✓						✓		✓		Range Image		
Zuo et al., 2004 [23]	✓				✓				✓		Stereo Pair Images		
Choi et al. 2012, [33]	✓				✓						Point Cloud		
Buchholz et al., 2013 [38]	✓	✓			✓			✓			Depth Image		
Harada et al., 2013 [39]	✓	✓				✓		✓			Point Cloud		
Domae et al., 2014 [40]	✓	✓					✓	✓			Depth Image		
Harada et al., 2016 [44]	✓	✓				✓		✓			Point Cloud	✓	
Mahler et al., 2017 [53]	✓						✓	✓	✓		Depth Image	✓	✓
Zeng et al., 2017 [54]	✓						✓	✓	✓		RGB-D Image	✓	✓
Levine et al., 2018 [48]	✓	✓					✓	✓			RGB-D Image	✓	
Matsumura et al., 2018 [46]	✓	✓					✓	✓	✓		Depth Image	✓	✓
Fujita et al. [55]	✓					✓		✓	✓	✓	RGB-D Image	✓	✓
Ishige et al., 2020[56]		✓	✓	✓			✓			✓	Vision-Less	✓	
Tachikake and Watanabe, 2020 [57]	✓	✓					✓	✓			RGB Image	✓	✓
Song et al., 2020 [58]	✓	✓					✓	✓			RGB-D Image	✓	
Ichnowski et al., 2020 [41]		✓					✓	✓			Depth Image		
Tong et al., 2021 [59]		✓	✓	✓		✓				✓	RGB Image	✓	
Moosmann et al., 2022 [60]	✓		✓	✓	✓			✓			Point Cloud	✓	✓
Zhang et al., 2021 [61]		✓		✓			✓	✓			Depth Image		
Zhang et al., 2023 [3]	✓		✓	✓			✓	✓			Depth Image	✓	
Zhang et al., 2023 [62]	✓		✓	✓			✓	✓			Depth Image	✓	✓
Zhang et al., 2023 [63]			✓	✓			✓	✓			Depth Image	✓	

Learning method and various simulators are frequently used in bin picking researches. Several approaches rely on the simulator to collect a a large scale of data [49, 45, 57] while some studies learn from real world [3, 44, 48]. With the development of simulators that can handle complex physical phenomenon or deformable objects, it is promising of collecting large-scale datasets for complex scenarios or efficient online training. A large amount of bin picking systems has focused on some specific tasks or grasping specific challenging objects varying from manufacturing to food industry, logistic and even daily life. For instance, when the objects near the bin wall cannot be picked by top-down approaching, studies proposed non-prehensile manipulation to increase grasp access [64] while industries solve it using extra machines for vibrating or shaking. The primary robot hand used in bin picking is typically a two-finger parallel gripper or a suction cup, due to their low cost, ease of control, and ability to handle a wide range of objects. Recently, more studies develop their own robot hand to address the challenges of complex objects [59, 65] or accuracy problem [66, 67]. Generalizing specially designed robot hands to practical production is also very promising.

1.2 Research Focus in Robotic Bin Picking

In the last section, I introduced the background of robotic bin picking and reviewed the studies of both analytic approaches and learning. Here, I provided a perspective for surveying the details of bin picking systems over the last two decades as shown in Table 1.1. Note that some studies lying in the related field of grasping from the clutter are also mentioned. Bin picking methods in earlier years focus on analytic methods applied in the production automation, mainly by detecting the geometric feature or localization and pose of the object and then pick them. During this area, grasp planning problem are rarely referred due to the number of the clutter and the shape of the objects is quite simple. After that, several warehouse automation approaches become popular by the increasing amount of learning algorithms. These approaches assume object models are unknowns and directly predict grasp poses or even if motor commands. A variety of gripper is also developed in order to handle different levels of specific tasks. Finally, hybrid approaches including analytical planning with deep

learning also can become a new trend.

After thoroughly reviewing the researches under the topic of robotic bin picking and grasping from clutter, I categorize the research focus in robotic bin picking as three groups: **accurate state estimation**, **robust grasp planning** and **challenging objects**.

1) Accurate State Estimation: State estimation is challenging due to such cluttered and unstructured environments. Vision recognition suffers from heavy occlusion, sensory noise and uncertainties during the bin picking process. Studies have addressed on this challenges by improving the visual recognition results in a iterative way or using end-to-end methods to skip over the certain visual processing. Suzuki et al. [68] proposed an online self-supervised method to adjusts grasp poses via feedback to decrease the error and achieve more towards ground truth. Doumanoglou et al. [42] refined the 6D object poses and predict next-best-view in heavy clutters. Harada et al. [44] also proposed a method for better seeing the occlusions in the container. The accuracy problems on manipulation or execution address on hand-eye calibration, closed-loop manner or feedbacks. Levine et al. [48] directly overcomed the difficulties in hand-eye calibration. By learning hand-eye calibration to servo the robot to reach the objects under self-supervision, they aim at helping the circumvent the accumulation of errors and leading to accurate results. They method significantly improve the success rates in grasp from the clutter thanks to their robust closed-loop control planner which gives the system the ability to interact with the environments. These approaches can reduce the accumulated errors from the hierarchical system and precisely lead to the final objectives for bin picking. Although analytic 6D pose estimation provides robust performance in practical bin picking, learning can further process the scenes with heavy occlusions and refine the poses to improve the recognition accuracy. Dong et al. [69] estimated 6D pose from the point cloud in the industrial setup. Zeng et al. [54] proposed a self-supervised method to estimate the 6D pose of objects from RGB-D images for warehouse automation. All these approaches significantly improve the recognition accuracy compared with analytic methods.

In order to catch up with the development trend of deep learning especially in bin

picking, several standard datasets and benchmarks need to be mentioned. It will be very helpful if there exists several open-source datasets for the systems to reach more reliable and generalized performance. Unlike other object or image datasets, they all share one goal such as instance/semantic segmentation, 6D pose estimation or image classification. Unfortuately, every bin picking datasets are designed for the special problems. They can not gives a benchmark for the future researches. Periyasamy et al. [70] proposed a synthetic dataset for understanding dynamic scenes where the objects poses can be tracked. Yang et al. [71] proposed a multi-view dataset for the reflective objects in bin picking. Kleeberger et al. [72] provided large-scale industrial parts with 6D pose. Instead of directly using the existing datasets, general methods of large-scale synthetic datasets will be useful.

2) Robust Grasp Planning: The essential techniques in bin picking is to grasp from the clutter. Two-finger parallel jaw grippers or suction cups are usually used in this tasks. Grasp detection faces the difficulties from rich contact and collision, fast motion planning algorithm and the environmental uncertainties. Classic bin picking methods samples force closure grasps with the object model and select the one without causing collision. Studies focus on improving grasp quality metrics considering more and more aspects. The underlying idea is to treat grasp perception analogously to object detection in computer vision. Some approaches focus on using implicit features instead of geometry primitives [73]. Some approaches focus on considering collisions between fingers and objects [40, 38].

Development of deep learning in grasp synthesis fields inspires the bin picking works [74, 75, 51, 48, 76]. Different grasp quality metrics can be learned based on a large-scale of dataset for parallel grippers [77, 45, 46], suction cups [78] and more generally, both [53]. On the other hand, in the case of heavily cluttered objects or various constraints with the environments, approaching in a top-down orientation may not be the optimal solution. High dimensional grasp configurations enable robots to pick up object precisely and flexibly [58].

3) Challenging Objects: First, some objects are very difficult for a 3D camera to capture, such as reflective or transparent parts. Dyrstad et al. [50] proposed a method to pick up reflective steel parts by training the network using synthetic

data. Tachikake and Watanabe [57] used RGB images as input and generate solutions for semi-transparent objects. Others requires specified manipulation primitives other than directly lifting to solve. As Sajjan et al. [79] proposed a method that can recognize the transparent objects from 3D data, we believe that approaches for grasping transparent objects in bin picking will soon be tackled. Second, if adapting the top-down grasps with parallel jaw grippers as the picking method, some objects can be very hard to find a grasp pose. To address this challenge, Le et al. [80] proposed a learning-based method to grasp planar objects in bin picking. Tong et al. [81] proposed a planner for picking up thin objects without suction gripper. They only use learning-based method to detect the object locations and plan a manipulation policy analytically to grasp the objects. Similarly, Tong et al. [59] proposed a method to grasp go-stones due to its specially shape and challenges in the clutter and He et al. [65] also proposed scooping manipulation to grasp thin objects. Although these approaches only use learning for locating the objects, it is also meaningful that the object detection stage can be easily done to serve for a better manipulation policy. Tachikake and Watanabe [57] solved the problem existing in picking up geometric variation objects and objects with a biased center of gravity. Finally, objects involved with rich physical interactions, e.g., tangled-prone objects are challenging since the robot has high possibilities to grasp entangled objects following a top-down approaching-grasping-lifting strategy. I will explain more details about the challenges in picking entangled-prone objects in next section.

1.3 Robotic Bin Picking for Entangled Objects

It is difficult yet important to automate the bin picking process for objects that tend to get entangled when randomly placed in a bin, e.g., U-bolts, S-hooks or deformable wire harnesses. The challenges come from high occlusion in the clutter, elusive entanglement phenomena, and the need for skilled manipulation planning. For picking simple shaped objects, the robot usually lifts the target in the vertical direction after a successful grasp. Simply adapting the existing bin picking strategies for picking

entangled-prone objects shows unsatisfied performance. For this reason, the manufacturing industry still relies on human workers to grasp and separate entangled-prone objects. Therefore, developing intelligent systems to automate this process is highly demanded. Here, I elaborate the challenges of realizing such systems by three aspects: **perception**, **manipulation planning** and **entanglement representation**.

1) Perception: This task poses challenges in perception since the robot must be able to distinguish isolated and potentially tangled objects in a cluttered environment. Some studies uses the prior knowledge of the objects to better understand the scene first. Studies address the segmentation problem on deformable linear objects [82, 83]. Leao et al. [52] proposed a method to pick up soft tubes by fitting shape primitives to clutter. Moosmann et al. [84] proposed to estimate the 6D pose of the target and then leverage reinforcement learning to plan separation motion. However,it is challenges for state estimation without object models. Specifically, for deformalbe objects such as wire harnesses, model-based bin picking polices may not suitable since it is difficult to construct the 3D models. Instead of understanding the scene first, we can directly obtain the task-relevant visual features. To avoid estimating the state of each object, Matsumura et al. [49] evaluated the scores of a set of grasps based on if it will lead the robot picking up entangled objects.

Multiple sensory is also used in difficult tasks for bin picking. Robotic bin picking relies on computer vision processing for object recognition and grasp planning. However, the integration of other sensory inputs can enhance the robustness for handling heavy occlusion and the intricate objects properties. Haptic feedback is widely adopted in industrial robots for failure detection. Moreira et al. [85] utilized a force sensor to assess the success of picking operations. Hegemann et al. [86] proposed a failure detection algorithm for grasping based on both visual and haptic inputs. Studies also use tactile information to guide the bin picking process instead of vision. Ishige et al. [56] proposed a vision-less system that relied solely on tactile feedback to pick small bulked objects. Multiple modalities of vision and haptic signals in a closed-loop manner shows great promise in the field of smart manufacturing.

2) Manipulation Planning: Manipulation is challenging for planning effective and general separation motions due to the complexity of entanglement estimation

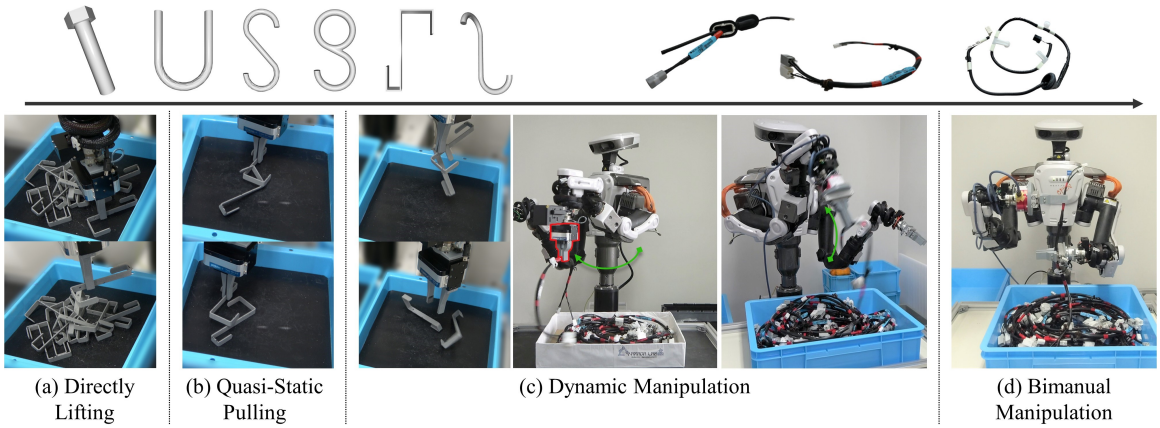


Figure 1.2: **Manipulation planning for entangled objects.** Skillful and specific manipulation is required under different entangled and cluttered environments for various objects. I consider quasi-static, dynamic or bimanual manipulation for disentangling the objects. Action affordance is also used to encode the action with perception.

and real-world uncertainties. Some approaches focus on searching and grasping untangled objects but are insufficient for cases where all the objects are entangled in the bin. There usually exists scenes where no untangled or isolated objects exist. The robot must be flexible enough to understand the current situation and plan the corresponding actions. Thus, the challenges would be, different objects and different entanglement patterns require different manipulation strategies. For examples, Leao et al. [52] planned a escaping trajectory to drag the object from the clutter. Moosmann et al. [84] proposed a multi-step escaping trajectory.

While considerable progress has been made with rigid tangled-prone objects [49, 61, 62], deformable objects with complex structures, such as wire harnesses, remain relatively unexplored. For deformable object manipualtion, Grannen et al. [87] proposed pin-and-pull to untangle the cable knots. Recently, dynamic manipulation is also demonstrated effective for cable manipulation [88, 89]. On the other hand, manipulating multiple deformable objects poses unique challenges. The solutions must simultaneously consider the heavy occlusion and rich contact formed by a large number of objects and also the entanglement issues. Viswanath et al. [90] proposed to disentangle multi-cable knots by task-relevant keypoint prediction and knot graph representation. Huang et al. [91] presented a method for untangling multiple cables

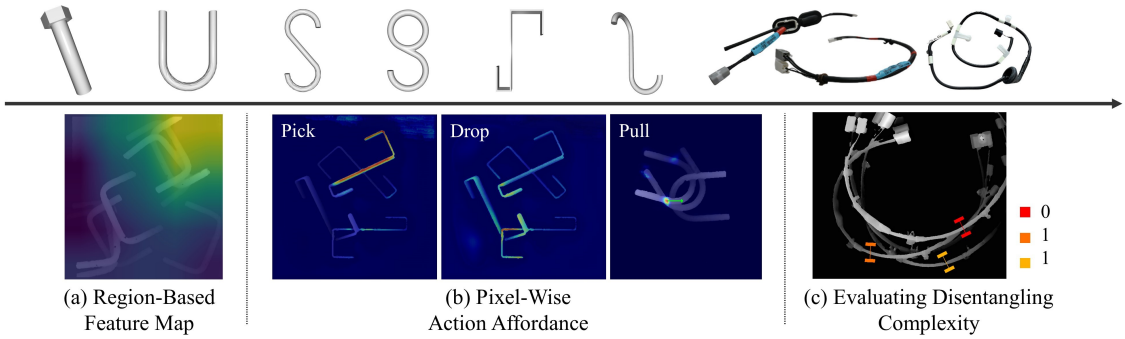


Figure 1.3: **Scene understanding and entanglement representation**. To better understand entanglement in an unstructured environment, I focus on extracting suitable features from visual inputs. As the degree of entanglement increase, the entanglement representation evolves from region-based feature maps, pixel-wise spatial action maps, to complexities of disentangling actions.

by tracing the topological representation. Other studies focus on scenes with higher degrees of entanglement in food industry. Ray et al. [92] proposed a method using a designed two-finger gripper to untangle herbs from a pile. Takahashi et al. [93] developed a learning-based separation strategy for grasping a specified mass of small food pieces.

3) Entanglement Representation: The entanglement representation can be done by physical trials where the metric is binary based on the robot will successfully grasp the objects or not. This method requires a large amount of experiments to verify either in real world [93] or simulation [49]. However, executing picking attempts in real world is time-consuming and simulated training always suffers from sim-to-real gap. Building a simulator with the real dynamics effects is also extremely challenging for complex-shaped objects or even deformable objects. Another method comes from the knot theory. Lui et al. [94] represented a knot as a graph and annotated each intersections. Sundaresan et al. [95] extended this representation to non-planar knots. Huang et al. [91] proposed a topological representation space to estimate the state for multiple cables. However, the full state is requires for applying knot states. In the real-world experiments, it is difficult to access the true state visually. Some studies proposed to use task-relevant visual features in deformable object manipulation [96, 97]. Thus, there seem lack a general representation of entanglement for bin picking

using cluttered and entangled objects.

In this thesis, I formulate the problem of picking entangled objects as a vision, motion/sequence planning problem. The proposed bin picking systems do not require any prior knowledge to flexibly pick isolated objects and separate entangled objects. Picking entangled objects requires specific strategies in both state estimation and manipulation. First, to avoid grasping the entangled objects, visually distinguishing if an object is entangled or not is important. It is challenging since the clutter scenes often contain heavy occlusions or self-occlusions of the objects themselves. Locating the objects using object models is difficult while it becomes more challenging for approaches without using models. They're also cases where the vision information might not be sufficient, requiring coupling with other sensory data, such as haptic feedback. Fig. 1.3 shows our effort of understanding the entanglement using visual features. After processing the entanglement scenes, it is natural that a reasonable picking sequence is planned: first select the isolated objects to grasp, then disentangle the rest objects. Therefore, disentangling motions are required to effectively separation the entangled objects as Fig. 1.2 shows. These motion should be well designed and planned, precisely predicted and robustly executed without relying on object models. The strategies of avoiding entanglement and disentangling should be flexibly combined in designing bin picking policies with excellent efficiency, robustness and generalization. Building on a hybrid approach combined both analytic and deep learning, my research can handle entangled objects with different difficulties and realize high efficiency in real-world environments.

1.4 Organization of the Thesis

This dissertation is organized as follows.

Chapter 2 addresses the problem of picking rigid objects by avoiding the entangled regions from visual inputs. The core technique is the entanglement map, which is a topology-based feature map to measure the entanglement possibilities of each region from the input image. The entanglement map is then used to select probable regions containing graspable objects.

Chapter 3 aims to solve the difficult case that the last chapter leaves when the bin contains no isolated objects. A learned policy enables the robot to dexterously pick or disentangle the objects based on the current observation. Two disentangling strategies: quasi-static motion pulling and dynamic motion dropping, are used to effectively separate the objects. To efficiently collect data for training, I leverage the self-supervised learning paradigm using an algorithmic supervisor in a physics simulator.

Chapter 4 tackles the problem of picking entangled deformable objects. The target objects are wire harnesses - essential connecting components in manufacturing industry, but long, flexible and tend to get entangled when randomly placed in a bin. A learning framework is trained using real-world data to infer the suitable disentangling actions based on the level of entanglement.

Chapter 5 keeps increasing the success rates of picking entangled wire harnesses the same as the last chapter. A dual-arm robot is deployed to grasp, extract and disentangle wire harnesses from heavy clutter using dynamic manipulation. This closed-loop system with multi-sensory inputs can significantly improve the accuracy, robustness and generalization of picking wire harnesses.

Finally, in Chapter 6, the achievements of the proposed methods presented in this dissertation and future work ideas are discussed.

Chapter 2

Topological Visual Representation for Entangled Objects

2.1 Introduction

This chapter addresses the problem of picking rigid, tangled-prone objects while avoiding any entanglement in bin picking. The core technique is the entanglement map, which is a feature map used to measure the possibilities of entanglement obtained from the input image. Topological knowledge can be used to generate the entanglement map from a single depth image. The grasp positions are detected by select probable regions containing isolated objects from the entanglement map, taking into consideration the collision between the robot hand and the objects.

Several studies focus on visually recognizing the entangled objects using prior knowledge [52, 98] or without any object model[49]. Matsumura et al. [49] first tackle the problem of picking only one object from a stacked pile without causing entanglement [49]. A Convolutional Neural Network (CNN) is proposed to predict whether if the robot can pick up a single object among several pre-computed grasp candidates. They also use a physics simulator to collect training data by simulating bin-picking processes. However, we found some limitations in this research as follows. On one hand, data-driven method requires a large amount of training data and time-consuming training procedure. On the other hand, since CNN only makes predictions

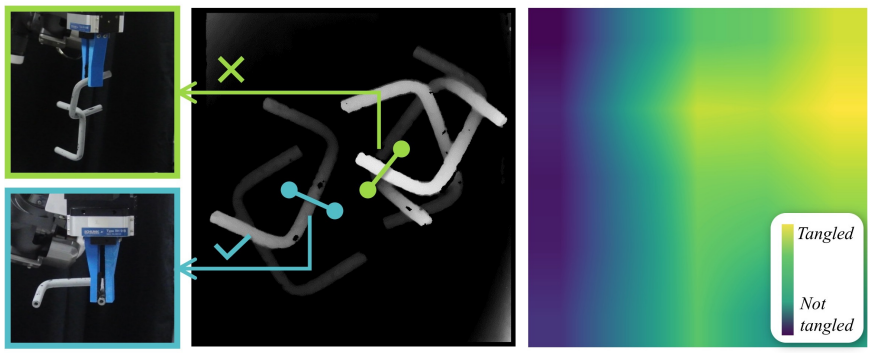


Figure 2.1: **A failed grasp and a successful grasp for picking up a single object.** Grasp marked as green shows the failure example of picking up multiple objects. Grasp marked as blue is successfully generated avoiding the region containing tangled objects. We also present the entanglement map using the proposed method, demonstrating our method can recognize the location of the entangled or isolated objects.

on cropped regions from the image, it cannot observe all the entangled objects from the whole input scene. Especially when the objects are heavily entangled, CNN may predict that all pre-computed grasp candidates share high possibilities of potential entanglement.

Motivated by the previous work, this research proposes an analytic approach to solve the entanglement issue in bin-picking using topological knowledge. Topological representation has been studied for decades and it is widely used for analyzing the relationship of characters. Ho and Komura firstly present the definition of topology coordinates to analyze the whole-body behaviors between two humanoid robots [99], [1], [2]. They calculate the topological relationship between two robot characters applying for different scenes. The most significant representation of entanglement would be Gaussian Link Integral (GLI) developed from knot theory [100]. It describes a mathematical relationship between two tangled strands as Fig. 2.2 shows. Moreover, topological representation plays an important role in robotic manipulation for deformable objects, such as tubes or ropes [101], [102], [103], [91]. This research is the first one to use topological knowledge in robotic bin-picking. The topological solution provides a more comprehensive measurements for dense clutters than the previous learning-based method.

In particular, this research introduces topology coordinates to obtain a series of metrics which can describe entanglement situation from a single depth image. Besides, the input depth image is scanned in a sliding window manner to generate a feature map called entanglement map, which indicates the possibilities of containing entangled objects for each region. As Fig. 2.1 shows, the entanglement map is able to discriminate which regions may contain tangled objects and which regions may not from a depth map. The regions marked as blue has high possibilities of containing graspable objects. Once the entanglement map is obtained, we can select non-tangle regions and detect collision-free grasp candidates using Graspability measure [40] respectively on selected regions. The output is a set of ranked grasp configurations of avoiding all entanglement and collisions.

The main contributions are as follows.

- A topology-based approach that can detect optimal grasps avoiding entanglement, which is a challenging problem in robotic bin-picking. We fix the problems existing in previous work. Besides, this method only requires simple parameter tuning instead of time-consuming training and data collection.
- A feature map that provides a complete observation and intuitive measurement of entanglement so that the bin-picking performance is improved dealing with complex-shaped parts.

Code, video illustrations can be found on our project website: https://github.com/xinyiz0931/bin-picking-robot.

2.2 Topology Coordinates

The original theory of topology coordinates is proposed by Ho and Komura [1, 2]. The topology coordinate has three attributes: **writhe**, **density** and **center**. Writhe explains how much the two curves are twisting around each other. For instance, entangled objects get a higher score than separated objects. Writhe between two objects is calculated by Gaussian Link Integral (GLI) using Equation 2.1. If we have

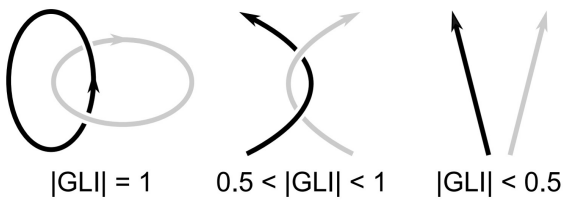


Figure 2.2: GLI identifies whether two strands are tangled or not [1, 2].

two curves γ_1 and γ_2 which are point sets in Cartesian Coordinate, writhe can be calculated by GLI as follows. The second attribute is density, which describes how much the twisted area is concentrated on two curves. The third attribute is center, which is a location that explains the center location of the twisted area.

$$GLI(\gamma_1, \gamma_2) = \frac{1}{4\pi} \int_{\gamma_1} \int_{\gamma_2} \frac{d\gamma_1 \times d\gamma_2 \cdot (\gamma_1 - \gamma_2)}{||\gamma_1 - \gamma_2||^3} \tag{2.1}$$

2.2.1 Definition

Given a single depth image of a cluttered scene, the topology coordinate can be constructed to measure the entanglement (Fig. 2.3). **Writhe** is a scalar attribute that indicates how much the objects are tangled together. A depth map containing tangled objects has higher writhe than the one with objects just overlapped together. **Density** is also a scalar attribute that indicates the distribution of the entanglement is evenly or intensively on the depth map. **Center** indicates the center position of entanglement on the depth map.

Original topology coordinates [1] can only be applied for two characters and assume the exact position of characters are known. Different from the definition and calculation, we construct topology coordinates only using a single depth image containing multiple objects so that the position of each object remains unknown. Instead of computing the relationship between two objects, we extract the line segments of edge from the depth map and calculate the topology coordinates using the relationship between each pair of line segments. Edge contains all the information we need to describe the shape and position of objects. Even though the line segments of

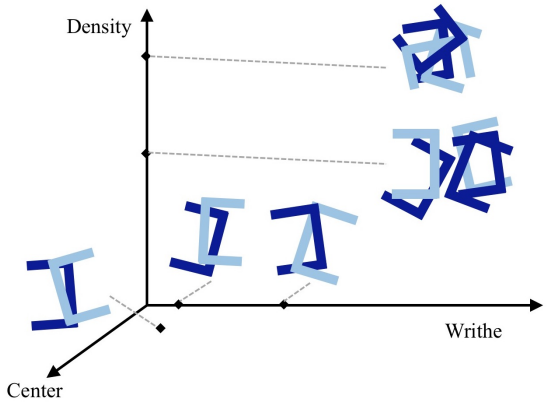


Figure 2.3: Topology coordinate illustrated by rigid industrial parts. We revise this definition originated in [1, 2].

edges may not indicate the complete contours of all parts, the topological relationship calculated by line segments can still reflect how and where the entanglement occurs.

2.2.2 Calculation

We use a depth map I to compute topology coordinates $G = (w, \boldsymbol{c}, d)$. In order to calculate these three attributes, we need to generate a matrix called writhe matrix T firstly. Taken I as input, we detect the edges and transfer them to a collection of 3-dimensional vectors $L = (l_1, l_2, ..., l_n)$. Writhe matrix T is a $n \times n$ matrix that stores GLI of each segment pair in L. Particularly, instead of using Eq.(1), GLI between two 3-dimensional line segments is computed using the algorithm proposed by Klenin and Langowski [104]. For instance, $T_{i,j}$ in the writhe matrix between i-th segment l_i and j-th segment l_j can be calculated by

$$T_{i,j} = GLI(l_i, l_j) \tag{2.2}$$

It can be seen that the writhe matrix T is an upper-triangle-like matrix where half of the elements in T are zero. We can compute the writhe w, density d and center $\boldsymbol{c}$ using writhe matrix T. First, writhe w is the sum of all values in T divided by the

number of line segments as follows.

$$w = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n} T_{i,j} \tag{2.3}$$

Then, density d is calculated by the ratio of the pairs that have higher values in writhe matrix T. We extract the non-zero elements from T and compute d using the number of elements higher than some threshold divided by the total number of non-zero elements. Here, we define the threshold as the mean of extracted non-zero elements. Finally, center $\boldsymbol{c}$ is simply obtained by the center of mass for matrix T, which is a segment pair that contributes the most to the entanglement. Moreover, we introduce a mask called center mask which has the same size as the input depth image (Fig. 2.5). A center mask is a binary matrix with an area consisting of both center segments.

2.2.3 Explanations

Firstly, writhe is a quantitative measure that denotes how much the entanglement is on the depth image, while density and center denote the position information for the entanglement. Therefore, we present the visualization of the writhe matrix and the center mask to elaborate the density and center more intuitively. Fig. 2.5 presents which region is the entanglement center from the input depth image while Fig. 2.4 shows four different clutters with various writhe and density values by presenting the corresponding input depth images, detected edge segments, and visualized writhe matrices. This visualized matrix derives from T since it only remains the larger elements and is resized to a certain size. We can observe which pairs of edge segments share the larger writhe value and what is the distribution of the segment pairs from the matrix. If the edge segments are tangled with those near them, the brighter values are concentrated around the axis of the matrix. On the contrary, if the edge segments share rather larger writhe values with those all over the image, the distribution in the writhe matrix is rather even. Therefore, the more concentrated around the axis in the writhe matrix, the higher the corresponding density is. The details are elaborated as

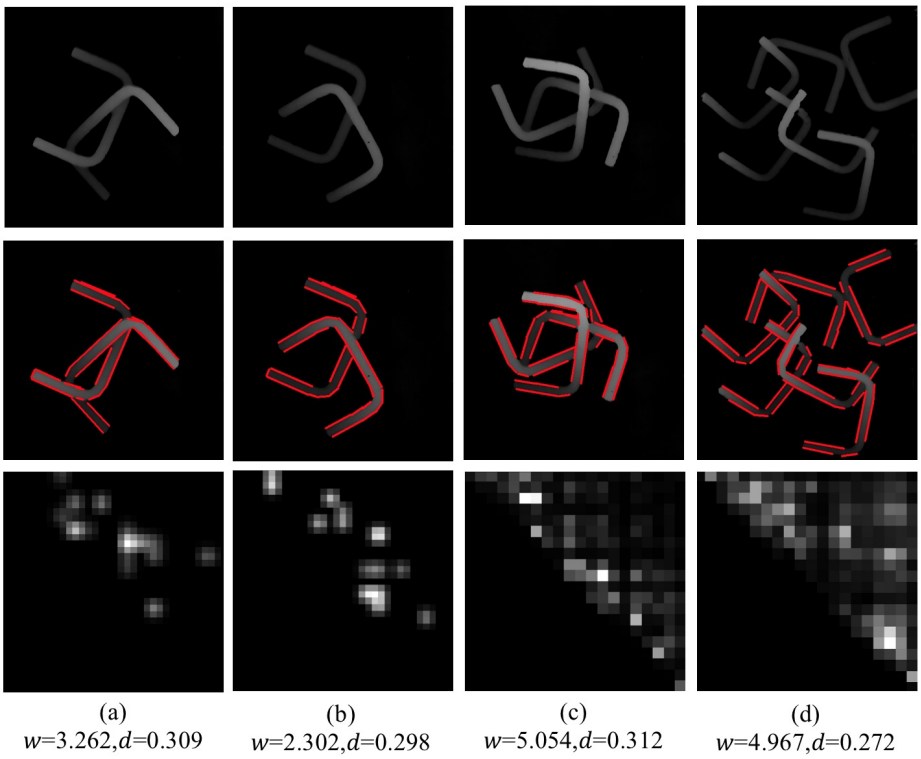


Figure 2.4: **Input depth images, detected edge segments, and visualized writhe matrices for 4 different clutter patterns.** (a-b) Scenes with different writhe and similar density. Overlapped objects in (b) has lower writhe. Writhe value w and density d is also written. (c-d) Scenes with different density and similar writhe. Visualized writhe matrix show that for the sparse clutter such as (b), density would be lower. Writhe value w and density d is also written.

follows.

Writhe. Fig. 2.4(a) is the situation where two objects are twisting together while Fig. 2.4(b) refers to two simply overlapped objects. They have similar density d but differ from writhe w. The writhe of twisted objects is larger than overlapped ones. If the robot wants to pick objects from these scenes, Fig. 2.4(b) with lower writhe has s higher possibility of a successful picking.

Density. From Fig. 2.4(c-d) we can tell by the human observation that Fig. 2.4(d) would be a better choice for robot simply by taking a look. Visualized writhe matrix and density value can also explain the scene numerically. The visualized matrix in Fig. 2.4(d) has an even distribution of brighter pixels since every line

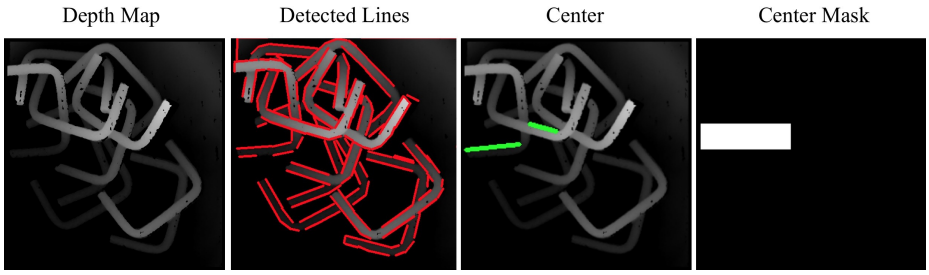


Figure 2.5: Illustration of the center in a depth map.

segment tends to tangle with more segments. Because as the number of segment pairs with larger GLI increases, the number of bright pixels in the visualized matrix also increases. Thus, these brighter pixels distribute more evenly, in other words, the density becomes smaller. On the contrary, every object in Fig. 2.4(c) is twisted with the other objects, thus, the entanglement is distributed intensely on the depth map. For the visualized matrix, the pixels with larger writhe are concentrated around the axis. Therefore, when writhe values are similar, density can also contribute to entanglement analysis.

Center. The center is computed by the center of mass of the writhe matrix, which is a pair of line segments that contributes the most to the entanglement. We present how the center affects the entanglement by presenting a mask that has the same size as the input depth image (Fig. 2.5). The center mask indicates the position information of the entanglement but not as much as writhe and density do.

To summarize, by focusing on the metrics of the entanglement regardless of the number of objects, situations with lower writhe and lower density is preferred. Therefore, the topology coordinate proposed in this section can be used to determine where a non-tangle grasp should be located.

2.3 Grasping Avoiding Entanglement

This section elaborates grasp detection method for picking up only one object by measuring the entanglement metrics using the proposed topology coordinates.

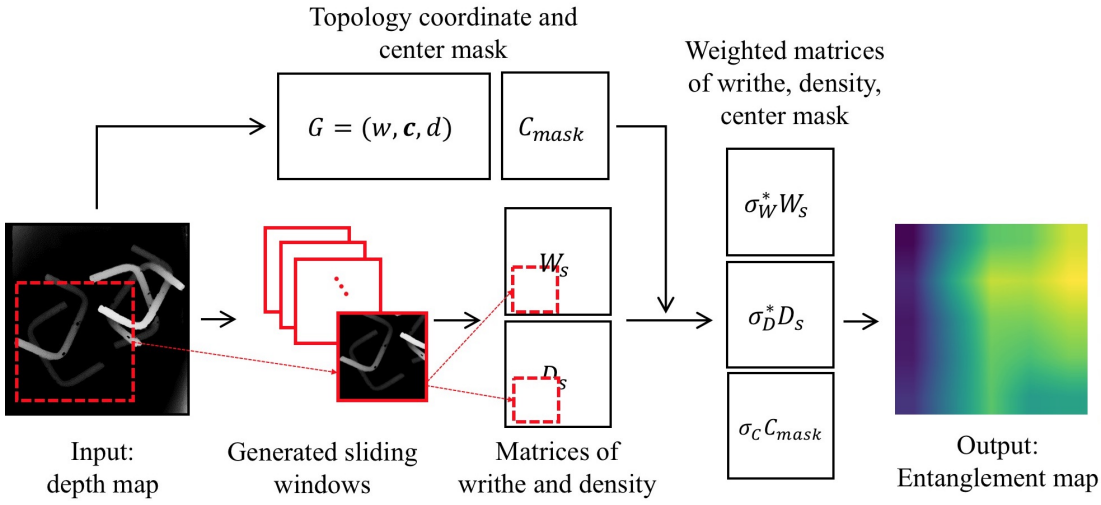


Figure 2.6: Entanglement map generation.

2.3.1 Entanglement Representation: Entanglement map

We explain how to compute an entanglement map by a given depth map I. First, we compute the line segments for edges on input depth map I, and generate topology coordinate $G = (w, \boldsymbol{c}, d)$ along with center mask C_{mask} for I . Then, we use a pre-defined sliding window function for I to obtain expanding information. For each window, we calculate its own writhe and density. This sliding window function returns two matrices W_s, D_s, which respectively store writhe and density of each region. The combination of two matrices refers to the rough entanglement information on each regions from I. However, we would like to precisely evaluate the entanglement situation upon the whole image. We use calculated topology coordinate G to evaluate the weights for these matrices and center mask. The initial weights are manually defined as $\sigma_W = 0.8$, $\sigma_D = 0.15$, $\sigma_D = 0.05$ respetively for W_s, D_s and C_{mask} since writhe affect more on predicting potential tangled regions. If $\overline{d}$, average of D_s, is larger than d of the coordinate G, it means that density may affect the result of entanglement map generation. Therefore, the weights are modified as,

$$\sigma_D^* = (\overline{d}/d)\sigma_D; \quad \sigma_W^* = 1 - \sigma_D^* - \sigma_C \tag{2.4}$$

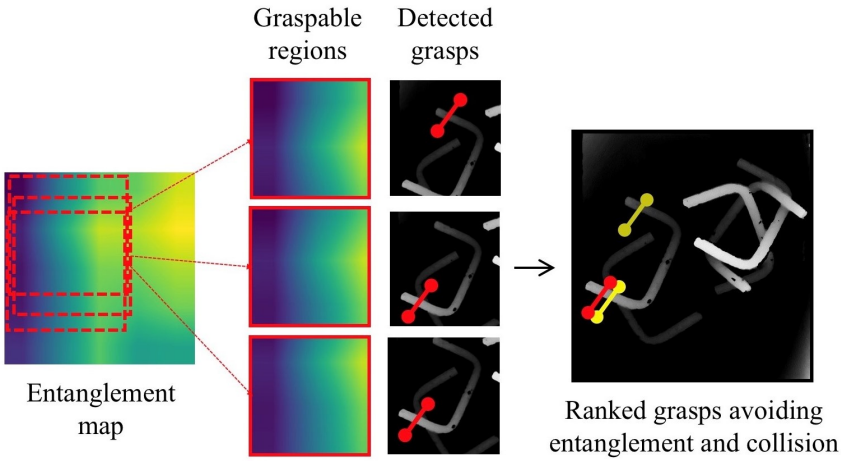


Figure 2.7: Generating optimal grasps avoiding any entanglement and collision..

The center mask is independent of the sliding window algorithm so that the weight σ_C remains the same. Finally, entanglement map E is obtained by the addition of weighted metrics as Eq.(2.5) shows following a bi-linear interpolation.

$$E = \sigma_W{}^* W_s + \sigma_D^* D_s + \sigma_c C_{mask} \tag{2.5}$$

2.3.2 Grasp Detection

We select a parallel jaw gripper and use a single depth map as input to compute grasp hypotheses. The overview of the proposed grasp detection method is illustrated in Fig. 2.8. First, a depth map is captured and used to construct the topology coordinate. Then, we use the writhe in topology coordinates to determine if the objects in the bin are tangled. If not, grasp is detected only considering the collision. If the entanglement exists, we calculate the entanglement map which evaluates where the potential tangled parts are in the box. We crop several regions with high possibilities of containing graspable objects from the entanglement map. Finally, grasp is detected and ranked in each selected region using graspability measure [40]. Graspability is an index for detecting a grasping point by convoluting a template of contact areas and collision areas for a robot hand. To put it more precisely, it is based on the idea that the object should be in the trajectory of hand closing, and there should be no object

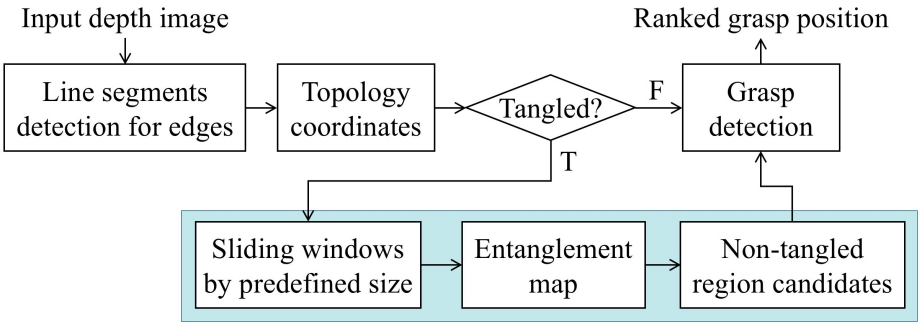


Figure 2.8: Proposed grasp detection method avoiding the entangled objects.

in the position to lower the robot hand. We use a parallel jaw gripper and rotate the gripper template along for 4 orientations. For the detected grasp candidates in each region, we rank them simply by pixel values in the entanglement map of the corresponding positions combined with the graspability score. Finally, the best grasp position is selected as the top of ranked grasp candidates.

2.4 Experiments and Results

2.4.1 Experiment Setup

We perform several real-world robot experiments to evaluate our method in bin-picking. We use NEXTAGE from Kawada Robotics and set a fixed 3D camera YCAM3D-II one meter straight above from the bin. We use Choreonoid and graspPlugin to simulate and execute the movement of the robot. The execution time was recorded on a PC running Ubuntu 16.04 with a 2.7 GHz Intel Core i5-6400 CPU. Our experiment system is set as Fig. 2.9 shows. As Fig. 2.10 shows, two types of industrial parts with complex shapes are selected. We prepare three patterns of clutter state by only C-shaped objects, only S-shaped objects, and mixed objects. In particular, three picking trials are performed for each clutter. Each trial contains 20 times of picking to record the success rate of only picking one object.

As Fig. 2.10 shows, two types of industrial parts with complex shapes are selected. We prepare three patterns of clutter state by only C-shaped objects, only S-shaped

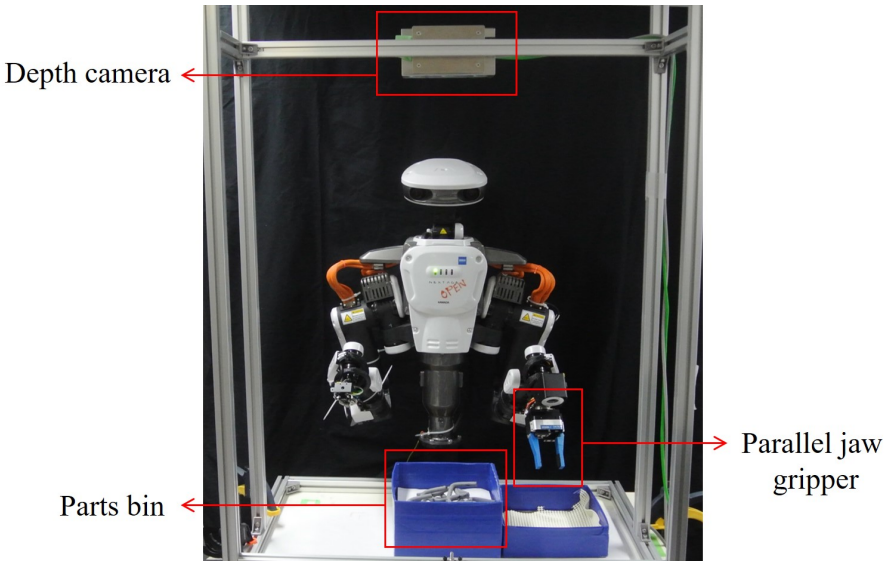


Figure 2.9: Experiment setup.

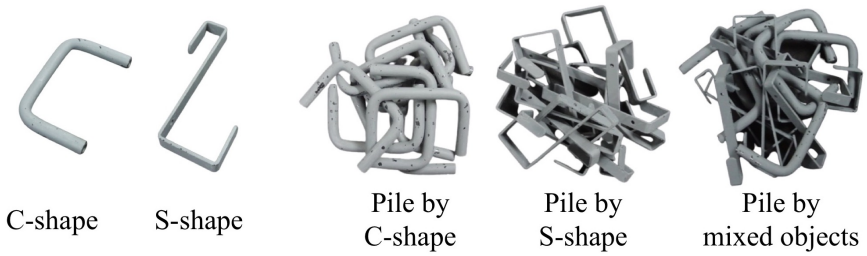


Figure 2.10: Types of objects and pile patterns used in the experiment.

objects, and mixed objects. In particular, three picking trials are performed for each clutter. Each trial contains 20 times of picking to record the success rate of only picking one object.

The purpose of the experiment is to compare our method with two baselines.

- **Graspability**: A general grasp detection algorithm [40] only using a hand template. It outputs several grasp candidates ranked by graspability measure.

- **CNN**: our previous work [49] which is the first approach to predict potential tangled objects in bin-picking. It takes the same grasp candidates as **Graspabilty**, but ranks them with a prediction network.

Table 2.1: Success rates of picking one single object

		Graspability [40]	CNN [49]	Ours
Success rate	C-shaped object	11/20	14/20	15/20
	S-shaped object	6/20	8/20	10/20
	Mixed objects	8/20	10/20	14/20
	Total	25/60	32/60	39/60
Time cost (s)		2.1	2.7	7.8

2.4.2 Bin-picking Performance

First, we evaluate the success rates and time costs for bin-picking experiments (Table 2.1). The number after slash denotes to the total picking times of one trial, and the one before the slash is the number of times when the robot picks up only one object. As a baseline, Graspability struggles in success rates since it can not discriminate whether the target is entangled with others or not. Our method and CNN both reach relatively higher success rates for picking a single object. Particularly, our method improves the performance of picking from S-shaped objects by a success rate of 50%.

The reason why CNN struggles with an S-shaped object is that it uses quarters of depth map to make predictions. Even if the cropped image contains the complete shape of target objects, it still lacks information of entangling with others. Our method directly evaluates entanglement for a complete depth map to solve the problem bothering CNN. For the mixed objects, our method reaches a high success rate of 70% since our model-free method only focuses on the information of edges in the depth map. The superior performance of our method indicates that our hand-engineered approach can analyze the relationship between these objects directly and efficiently. CNN may require more evaluation for generalization while our method can be utilized without training.

2.4.3 Qualitative Analysis

From the examples presented in Fig. 2.12, we validate how our method selects graspable objects qualitatively. For the same depth map as input, we use baselines and

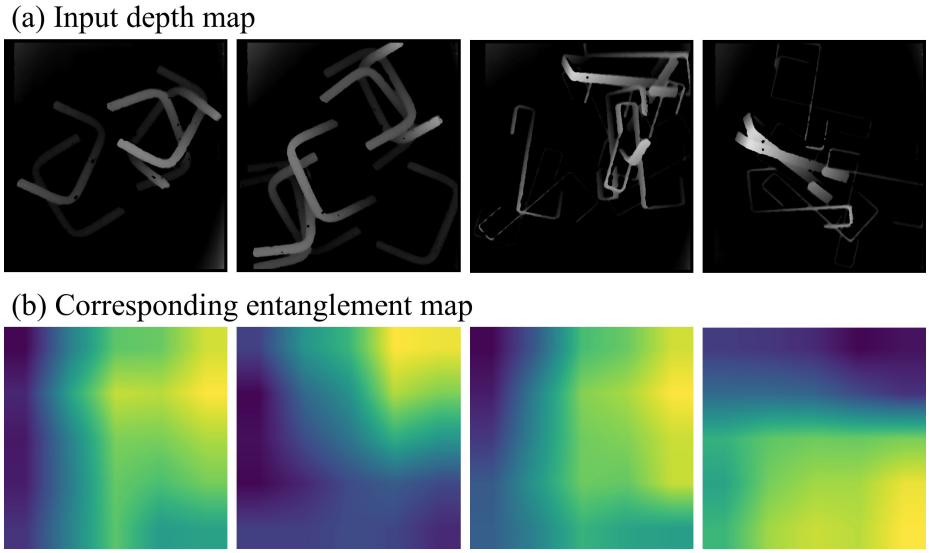


Figure 2.11: **Input depth map and corresponding entanglement map.** Yellows stands for entangled region. Blue area is the region where has low possibility of entanglement.

our method to detect optimal grasp positions, and the top-ranked grasp is marked using green dots. It is observed that our method can directly find the objects that are not tangled with others in the bin, while Graspability and CNN always focus on objects at the top of the clutter. Especially in the fifth column when all five grasp candidates are classified as tangled, both existing approaches predict poorly while our method successfully finds the graspable objects without any entanglement in the bin. Graspable objects selected by the proposed method are similar to the human observations. The reason is that our method uses edge and topological knowledge to explain the entanglement relationship more intuitively, which guarantees a complete observation of all potential entanglement in the bin. Some more entanglement maps are presented in Fig. 2.11. In our perspective, the entanglement map is the visualization that indicates possibilities of entanglement in every region for the whole depth map. We can observe those areas where objects are heavily tangled with each other are marked as yellow, while blue areas refer to non-tangled regions. We prefer to generate grasps on blue areas.

In addition, the average time costs of Graspability, CNN, and our method are

2.1s, 2.7s, 7.8s. The time cost of our method depends on how many line segments are detected from the depth map. For our experiment setting of 10 objects placed in the bin, the time cost per trial is limited to 8s.

2.4.4 Discussion and Limitation

Let us consider other kinds of industrial parts like Fig. 2.13 shows. This type of object provides too much edge information for topology coordinates, which may cause some misunderstandings. In this case, a learning-based approach would be necessary. However, since our method prefers objects with the shape of pure edges such as rigid linear objects with a smooth edge, it is possible to develop our method on manipulation of deformable linear objects such as tubes or ropes in future work.

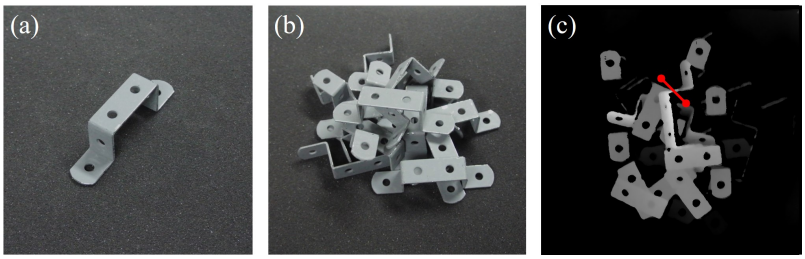


Figure 2.13: Our method can not apply to this type of object.

Common failures that result from our method are caused by the situation where the selected region does not contain any graspable positions. Even if the entanglement map can be generated correctly, grasps can not be detected due to the collisions in the selected region. In the future, it may be possible to add more collision information during generating the entanglement map to improve the performance.

2.5 Summary

This paper presents a topology-based solution for a robot to only pick only one object in robotic bin-picking. We present a topological feature map called entanglement map to describe the entanglement situation of cluttered objects in a bin. A grasp synthesis method is proposed to search for the optimal grasp without picking up entangled

objects from the whole input image. Our method reach fine success rates on real-world experiments. Our method is dependable upon its generalization capability even if for mixed objects in bin-picking. Particularly, we do not need large training data to make predictions since the proposed method can obtain the topological relationship of entangled objects even for complex-shaped objects.

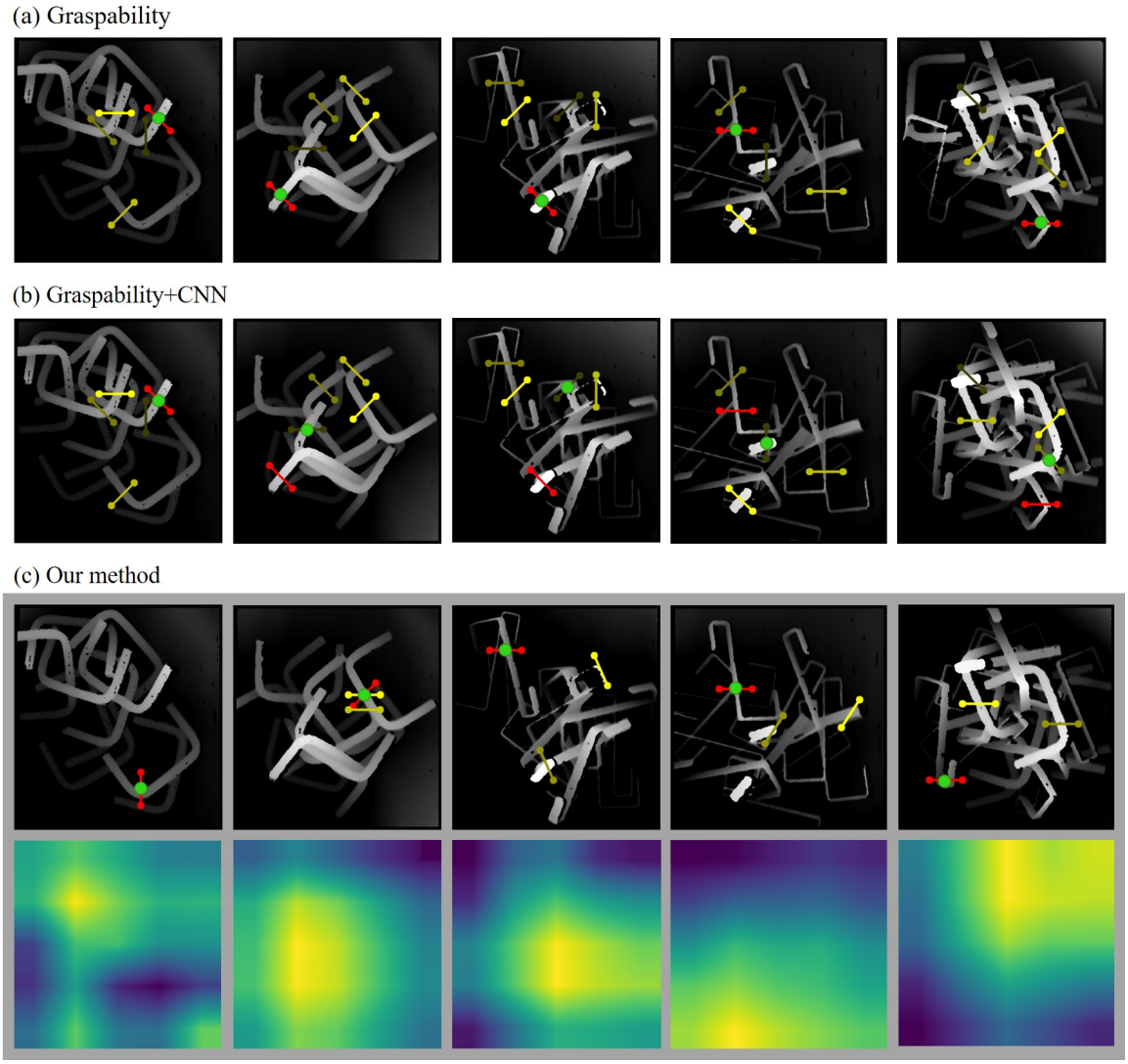


Figure 2.12: **Experiment results using the same depth maps.** Best grasps are emphasized using green dot. (a) Results of **Graspability**. Grasps marked as red denote to the best grasp. Yellow refers to grasps with the ranked order of 2nd, 3rd, 4th, and 5th. (b) Results of **CNN**. The grasp candidates are the same as (a) while **CNN** predicts the best grasps marked as green. (c) Result of proposed method. Both depth maps and entanglement maps are presented. Red denotes to the best grasp. Yellow refers to grasps with the ranked order of 2nd and 3rd.

Chapter 3

Learning Action Affordance for Entangled Rigid Objects

3.1 Introduction

The previous chapter (Chapter 2) introduces a bin picking method just by avoiding grasping the entangled objects. By visually constructing a feature map representing the entangled areas, the robot can find and grasp the untangled objects. However, we have to consider the cases where there is no isolated objects to grasp and the robot must separate the entangled objects. Synergies between visual representation and action selection are required to solve this problem. This framework poses challenges in perception since the robot must be able to distinguish the isolated and potentially tangled objects in a cluttered environment. Our prior works [49, 61] uses partial visual observation or simple geometrical features such as edges, making it challenging to be adopted in dense clutter. Model-based paradigm relies on the full knowledge of the objects and may suffer from cumulative perception errors due to heavy occlusion or self-occlusion of an individual complex-shaped object. Manipulation is also challenging for planning effective and general separation motions due to the complexity of entanglement estimation and real-world executions. Studies have proposed tilting the gripper to discard the entangled objects [52] or dragging the entangled object out of the clutter [84]. However, these object-specific strategies require prior knowledge

of objects and may be insufficient for objects with different geometries. Additionally, the aforementioned learning-based approaches rely on simulated supervision [84] or verifying the entanglement by simulated execution [49]. They do not provide any general criteria for entanglement in cluttered environments.

Specific strategies for separating an individual object in cluttered environments require visually reasoning the actions. Zeng et al. [105] proposed to learn the synergies between pushing and grasping to create enough space for grasping in the clutter. Danielczuk et al. [106] proposed to learn pushing policies to singulate the target object for future grasping in bin picking. Although pushing is useful for singulating daily objects or simple-shaped objects, some industrial objects face another challenge where they tend to get entangled. Studies address this problem by utilizing specifically crafted singulation strategies [84]. Action affordance can be used for encoding the action with perception. For instance, grasp affordance can be learned by predicting a pixel-wise heatmap mapped with the observation where each pixel indicates the possibilities of the grasp success [76]. Other studies also leverage action affordance for various tasks such as perceiving the 3D spatial structure of visual input for pick-and-place task [107], predicting the keypoints associated with visual input for cable untangling [96] or inferring both position and direction for manipulating articulated objects [108]. The direction of the applied action can be encoded by rotating the input image for the inference [105, 108].

To address these challenges, a novel bin-picking system is developed by leveraging self-supervised learning to flexibly and efficiently pick or separate various complex-shaped objects:

- PickNet learns to map the visual observations of the unstructured bin to affordance maps that indicate the pixel-wise possibilities of potential actions: to **pick** isolated objects or separate entangled objects. Our policy then selects the corresponding action with the highest action possibility. The network is trained with the idea that the untangled objects tend to present a complete contour in clutter, making it more interpretable than black-box classifiers or using insufficient object features.

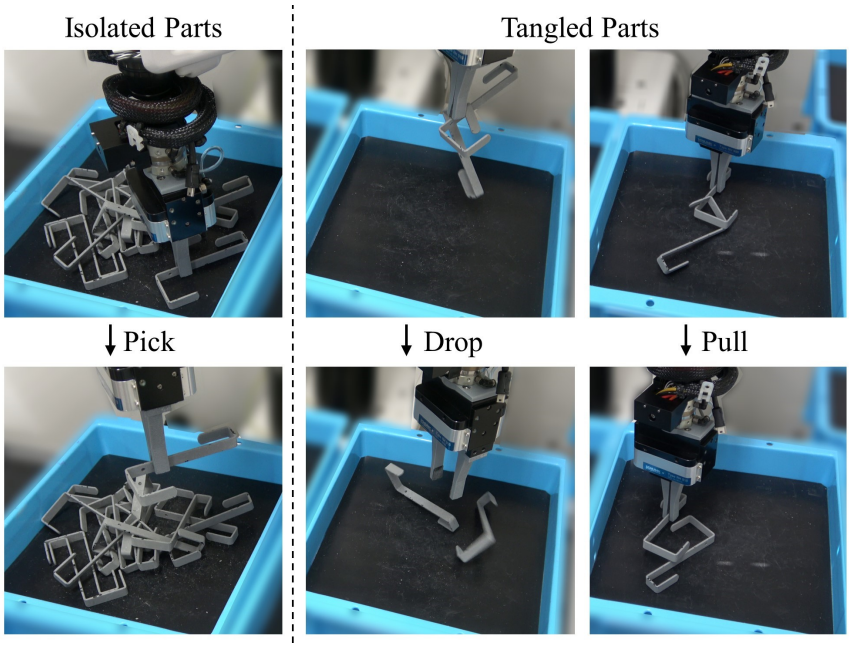


Figure 3.1: **Our policy learns to flexibly pick or separate tangled-prone objects for bin picking.** The robot can search the untangled objects in the bin and pick them up. If all objects are entangled, the robot can drop them into another bin to separate them dynamically. It can also perform pulling actions to disentangle the objects.

- Two efficient separation motion primitives are proposed to cope with different entanglement levels. The first motion is to **drop** the tangled objects into a buffer bin after grasping. Dropping can dynamically untangle the objects by utilizing the interactions with the environments instead of directly performing motions in dense clutter. It acts as an initial separation strategy to reduce the degree of entanglement and is suitable for a wide range of objects. The second motion is to **pull** the target object out of the entanglement. The robot can simultaneously pull and transport the objects when the degree of the entanglement is rather lower, increasing the action efficiency compared with dropping. PullNet is used to infer the position and direction for pulling from visual observations.

- PickNet and PullNet are trained using synthetic data collected in a self-supervised manner. An algorithmic supervisor is used to estimate the entanglement state and increase the efficiency of the collection process.

Experimental results suggest the effectiveness of our method using both simulated and real-world experiments with an average success rate of 90%. Our method can also be adopted on unseen objects and shows impressive results. Fig. 3.1 shows the proposed actions in our system. The contributions of this work are five-fold.

- A bin-picking system for tangled-prone objects that enlarges the accuracy, efficiency, flexibility and generalization.
- PickNet for distinguishing untangled or tangled objects in clutter and inferring the appropriate actions for them.
- Two novel and efficient motion primitives for separating entangled objects: dropping and pulling.
- PullNet inferring the pulling actions without object models.
- An algorithm for simulated self-supervised data collection.

Code, videos illustrations and supplemental material can be found at https://xinyiz0931.github.io/tangle.

3.2 Problem Statement

Let o denote the depth image of the clutter, (q, θ) denote a grasp with 4 degrees of freedom, where $q \in \mathbb{R}^3, \theta \in \mathbb{R}$ is the position and orientation of the gripper about the vertical axis to the workspace. The grasp pixel $p \in \mathbb{R}^2$ is inferred by our policy from the depth image o and then transformed to a 3-D location q for execution. We then leverage the method in [40] to compute the collision-free grasp orientation θ by convoluting the depth image o with the gripper model. We parameterize the action a with three motion behaviors:

- Picking: $a_{\text{pick}} = (q, \theta)$. The robot executes a grasp centered at q oriented θ, lifts in a vertically upward direction, and transports the objects to the goal bin

- Dropping: $a_{\text{drop}} = (q, \theta)$. The robot grasps at q with an orientation of θ and drop the objects into the buffer bin.

- Pulling: $a_{\text{pull}} = (q, \theta, u)$. The robot executes a grasp at (q, θ), pulls along $u \in \mathbb{R}^3$, and transports it to the goal bin. Our policy produces a 2-D pulling direction $v \in \mathbb{R}^2$ from the depth image o. For the physical execution, we transform v to a 3-D vector $u = (u_x, u_y, u_z)$ where the gripper pulls in the x-y plane along (u_x, u_y) while slightly lift along the z axis about $u_z = 0.2$ [cm]. The pulling action ends before the gripper collides with the bin walls. The robot also performs a wiggling motion during pulling to reduce the effects of friction with the bin plane.

The goal is to learn a policy π_Φ that maps the input depth image o to the action $a \in \{a_{\text{pick}}, a_{\text{drop}}, a_{\text{pull}}\}$ where the trained networks PickNet and PullNet are parameterized as Φ: $a \leftarrow \pi_\Phi(o)$.

3.3 Learning to Pick Entangled Objects

3.3.1 Method Overview

To efficiently pick up tangled-prone objects, the robot prioritizes grasping isolated objects in the clutter. If the bin contains no such objects, we leverage a buffer bin to reduce the degree of entanglement and help to perform the disentangling motions. The overview of our system is shown in Fig. 3.2. We first use a neural network PickNet to detect the untangled objects in the main bin. If such objects exist, the robot grasps them and transports them to the goal bin. Otherwise, the robot drops the entangled objects in a buffer bin to separate them. Then, the robot uses PickNet again to examine the buffer bin. If the objects are not successfully untangled, we use a neural network PullNet to perform a pulling action and transport the untangled objects to the goal bin. The buffer bin helps to create an environment with few collisions for pulling, and effectively disentangle the objects by the dropping motion. This process proceeds in iterations.

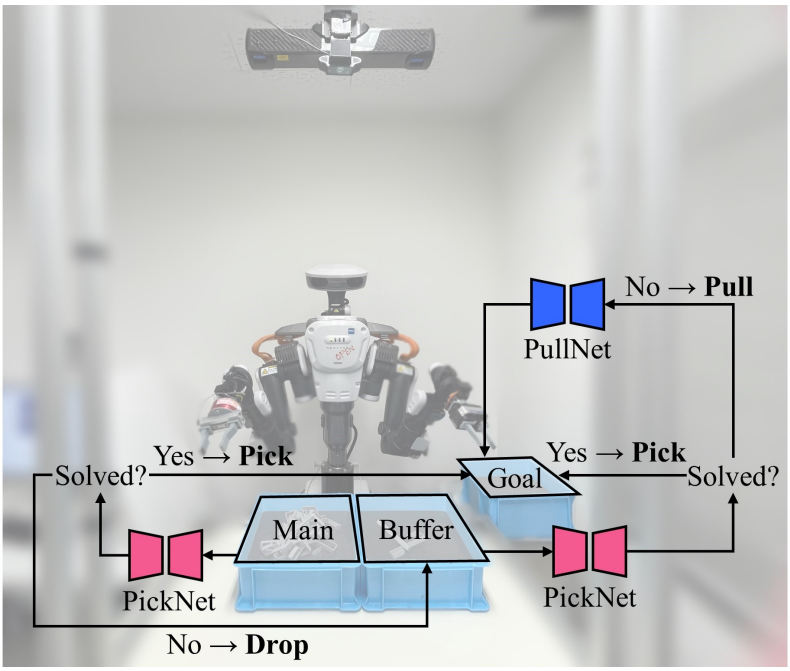


Figure 3.2: **Overview of the proposed policy.** The robot first uses PickNet to search untangled objects in the main bin and transport them to the goal bin. If such objects do not exist, the robot grasps the entangled objects, drops them in a buffer bin and uses PickNet to check if the separation succeeds. The robot then transports the isolated objects to the goal bin or separates the entanglement by pulling, which is inferred by PullNet.

3.3.2 PickNet: Learning to Pick or Separate

PickNet f_{pick} is trained to (1) classify if the bin contains untangled objects for picking or if the robot should perform separation motions (dropping for the main bin and pulling for the buffer bin) and (2) predict the pixel-wise grasp affordance for picking and dropping actions. Given a depth image $o \in \mathbb{R}^{512\times512\times3}$ with triplicated depth values across three channels, the output is two heatmaps $f_{\text{pick}}(o) \in \mathbb{R}^{512\times512\times2}$: PickMap and SepMap. PickMap predicts the pixel-wise possibilities of picking untangled objects while SepMap calculates the possibilities of containing entangled objects. To infer the actions in the main bin, we select the heatmap with the highest values between PickMap and SepMap to perform either picking or dropping action. In this case, the grasp position p is selected at the highest pixel on the corresponding heatmap. For the buffer bin, if the maximum pixel on the PickMap is higher than

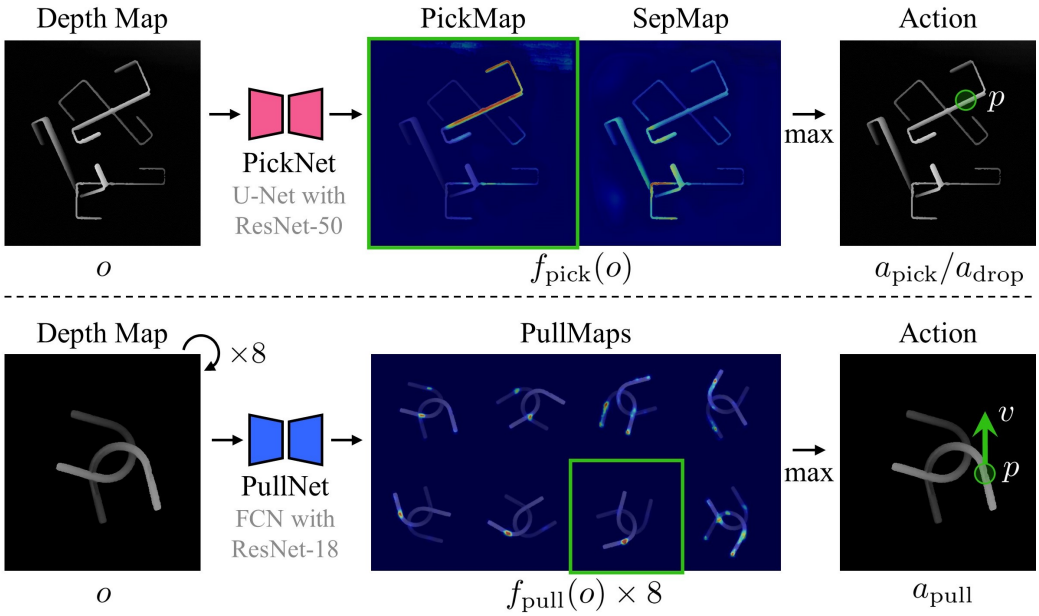


Figure 3.3: **Inference details of PickNet and PullNet.** Given a depth image as input, PickNet predicts two affordance maps representing the pixel-wise possibilities of picking and separating. We rotate the depth image by eight orientations denoting eight pulling directions and feed it to PullNet. The pulling action is determined by the affordance map that yields the highest score.

that on the SepMap, the robot picks the objects to the goal bin. Otherwise, the robot performs the pulling motion, as inferred by our proposed PullNet. Fig. 3.3 illustrates the inference process of PickNet. We use a ResNet50 [109] with U-Net [110] skip connections for PickNet pre-trained on ImageNet [111]. We use an MSE loss during training.

3.3.3 PullNet: Learning to Pull for Separation

We learn PullNet f_{pull} to infer the pulling action including position p and direction v. PullNet takes a depth image $o \in \mathbb{R}^{512\times512\times3}$ as input and generates a heatmap called PullMap $f_{\text{pull}}(o) \in \mathbb{R}^{512\times512}$ as output. Each pixel located in the PullMap represents the success possibility of pulling to the right of the image. We encode the pulling directions by rotating the input depth image for $\pi i/4, (i = 0, 1, \cdots, 7)$ [rad]. PullNet can reason about pulling to the right for each rotated image. Then,

the pulling direction v and position p are selected at the highest pixel value among eight PullMaps. Fig. 3.3 shows how the inference process using PullNet. We use a ResNet18 [109] as the encoder followed by a bi-linear upsampling layer in PullNet pre-trained on ImageNet [111]. We use a binary cross-entropy loss for training. The pulling position is encoded as a 2D Gaussian.

3.4 Self-Supervised Data Generation

We develop a physics simulator using the NVIDIA PhysX library to collect synthetic data for PickNet and PullNet. We randomly drop 3D object models in a bin and use a simulated parallel gripper to execute consecutive pickings repeatedly. The picking process is executed under physical constraints. Instead of randomly exploring actions in the simulator, an algorithmic supervisor containing a set of representations for entangled objects is used to efficiently control the collection process and adjust the dataset.

3.4.1 Algorithmic Supervisor

To distinguish if the object is entangled, we leverage the method for skeletonization and crossing annotation in [94] using the object model and poses. Takes the full state of objects in the bin as input, our algorithm can (1) classify if the objects are tangled or not, (2) plan effective pulling actions for disentangling objects and (3) determine the order for picking demonstrations.

1) Tangle Recognition: To distinguish if the object is entangled, we leverage the method for skeletonization and crossing annotation in [94] using the object model and poses. Fig. 3.4 shows the detail of this function `RecogTangle()`. We first skeletonize each object into an undirected graph of nodes and edges. We project all objects onto the bin plane to obtain a collection of undirected graphs. We then calculate the crossings where the objects intersect with others and add them as nodes to the corresponding graph. We annotate the crossings that each object forms with others with $+1$ or -1. If the edge intersects above the edge of other objects, $+1$ is

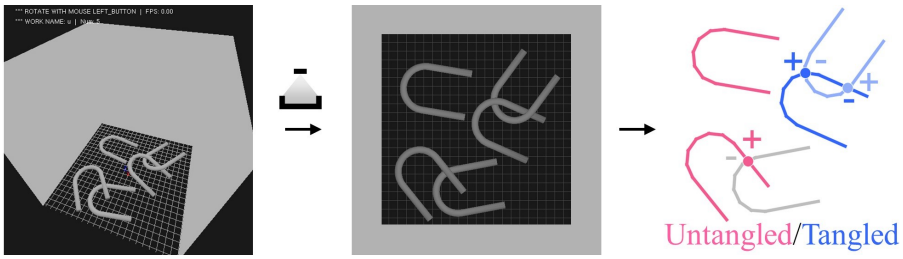


Figure 3.4: **Process of distinguishing untangled/tangled objects in our algorithm.** Given the full state of all objects as input, our algorithm skeletonizes the objects and obtains a graph collection by projecting along the vertical angle to the bin plane. For each object, we annotate the under-crossings it formed with others as -1 and otherwise as $+1$. The untangled objects (pink) are determined when the annotations of the crossings are $+1$ or without any crossings. The tangled objects (blue) have both $+1$ and -1 annotations.

annotated for the corresponding object. Otherwise, -1 is annotated. From the graph collection using vertical projection, untangled objects have only $+1$ or no annotation while tangled objects have annotations of both $+1$ and -1.

After computing the annotated graphs, we define four conditions that lead to different output action a: (a) If there exists an empty annotation list, the gripper lift the corresponding object; (b) Otherwise, ff there exists an annotation list where all elements equal $+1$, the gripper lift the corresponding object; (c) Otherwise, it means that the bin only contains the entangled objects, if the bin contains less than three objects, we plan pulling actions to disentangling them; (d) Otherwise, if the bin contains more than three entangled objects, the gripper lift the one with the least number of -1. Finally, we detect the grasp using the depth image and the mask of the target object using the modified algorithm from [40].

2) Pulling Planning: To plan to pull, we project objects from multiple angles to find feasible pulling directions (see Fig. 3.5). If the graph collection of a projection angle contains untangled objects, it is possible to separate the entanglement by pulling this object along the corresponding projection angle. Thus, the feasible pulling direction u is equivalent to the projection angle when the collection of projected graphs contains untangled objects. From a set of feasible pulling directions and pulling objects, our algorithm selects the object with the least number of -1 annotations in the graph collection using vertical projection. The entanglement level

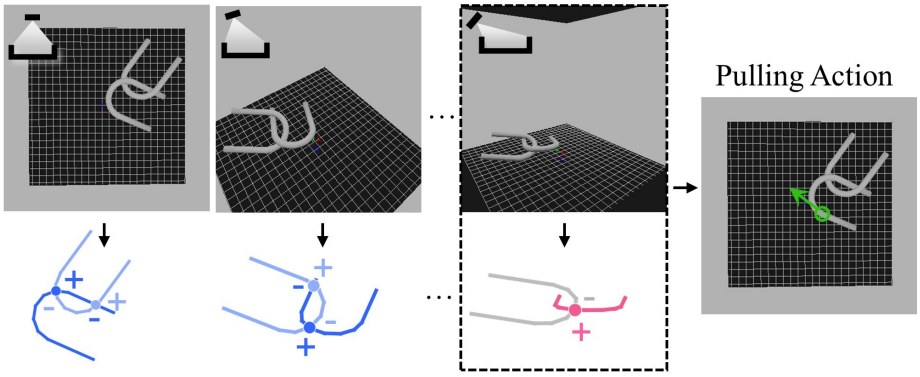


Figure 3.5: **Process of calculating feasible directions and corresponding objects for pulling in our algorithm.** By projecting and labeling the crossing from multiple angles, the feasible pulling direction is determined as the vector along the projection angle where the corresponding graph collection contains untangled (pink) objects.

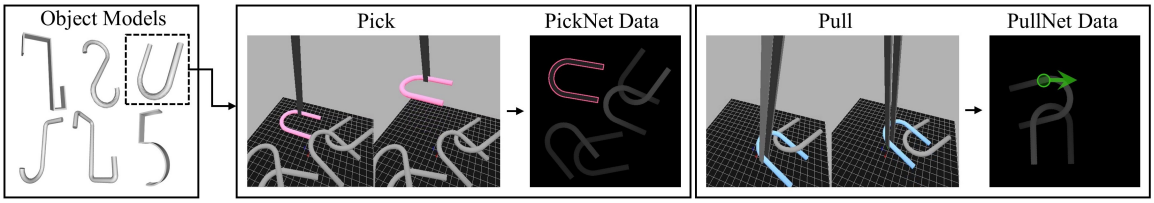


Figure 3.6: Demonstrations in simulation and data examples.

of this object is expected to be lower than others. The grasp q, θ for pulling is selected by considering the non-collision grasps as in [40]. Specifically, we uniformly sample 48 projection angles as candidates in SO(3) space and define that the sampled angles about the vertical axis should be in the range from $\pi/4$ [rad] to $\pi/2$ [rad] to reduce the search cost.

3) Sequential Demonstration: Our algorithm first randomly drops the objects in the bin and selects untangled objects to grasp (Fig. 3.6). If the bin contains no untangled objects but more than three entangled objects, the gripper picks the object with the least number of -1 annotations. If the bin only has less than three tangled objects, pulling motions is planned and performed. Note that our algorithm resets and drops the objects when the bin is empty or the gripper takes no objects out of the bin five times consecutively.

Algorithm 1: Algorithmic Supervisor

```
1  while True do
2      Drop objects in the bin;
3      N_fail ← 0;
4      while bin contains objects do
5          a ← RecogTangle ();
6          Execute a;
7          if only one object is out of the bin then
8              if a_pull is executed then
9                  Record for PickNet (masked SepMap) and PullNet;
10             else
11                 Record for PickNet (masked PickMap);
12         else if more than one object is out of the bin then
13             Record for PickNet (masked SepMap);
14         else
15             N_fail ← N_fail + 1;
16         if N_fail > 5 then
17             Continue;
18     end
19 end
```

Algorithm 1 shows the detailed process of data collection during simulated demontrations. First, objects are randomly dropped in to the bin (line 2). Line 5-6 denote the function `RecogTangle` of this algorithm, which returns the picking or pulling actions a for the execution. After detecting the grasping object and executing the corresponding action (line 5-6), one attempt is terminated when the gripper is out of the bin. Then, we count the number of objects in the bin before and after the attempt. If only one object is taken out of the bin, we record the data including the depth image, mask and corresponding action (line 7-13). Otherwise, the count of failure attempts adds one and the simulator tries again to find the grasp and action (line 14-15). If the number of failed attempts exceeds five (line 16-17), the bin is reloaded by randomly dropping the objects (line 2) and resetting the number of failed attempts (line 3).

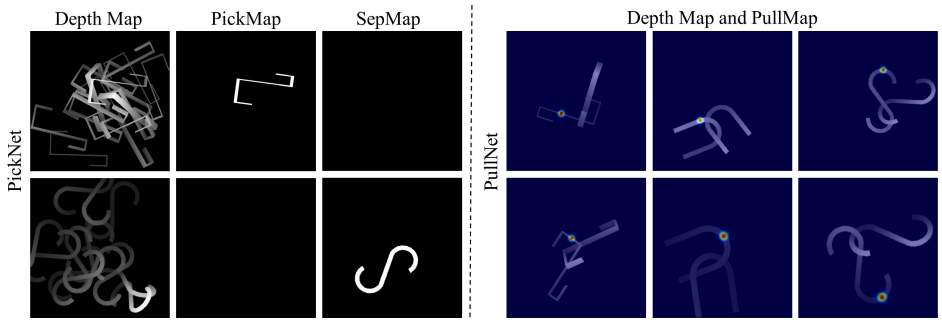


Figure 3.7: Ground truth labels for PickNet and PullNet.

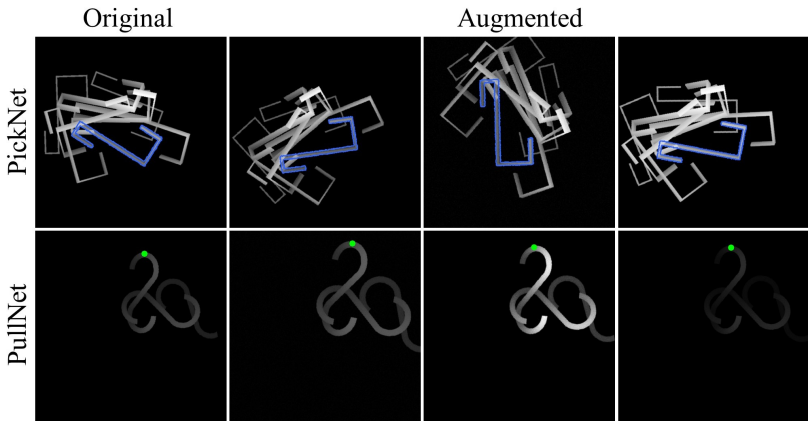


Figure 3.8: Augmented data for PickNet and PullNet.

3.4.2 Training Details

We use six models of tangled-prone objects including four planar objects and two non-planar objects. Each sample for PickNet contains a depth image and two masks PickMap and SepMap. After each attempt, if only one object is lifted without pulling, PickMap is masked with the target shape while SepMap is set to all zeros. Otherwise, if multiple objects are lifted, SepMap is masked with the target shape and PickMap is set to all zeros. On the other hand, if the gripper pulls and lifts only one object, the depth image and pulling action are recorded for training PullNet. Each sample for PullNet contains a depth image and a Gaussian 2D encoding of the pulling point the same size as the depth image. The depth image is rotated so that the pulling direction points to the right in the image. The ground truth labels are shown in Fig. 3.7.

Table 3.1: PickNet/PullNet Data Augmentations

Augmentation Parameters	Amount	
	PickNet	PullNet
Additive Gaussian Noise	(0.0, 0.01*255)	(0.0, 0.01*255)
Gamma Contrast	(0.5, 2.0)	(0.5, 2.0)
Elastic Transformation	(1,1)	(1,1)
Scale	(0.9,1.1)	(0.9,1.1)
Shear	(-10,10)	(-10,10)
Rotate	(-180,180)	-

Here are the detils of two networks. PickNet we use a ResNet50 [109] pre-trained on Imagenet with U-Net [110] skip connections. For the input, we triplicate depth values across three channels to match with the default input size of the pretrained backbone ResNet50. We use the mean square error (MSE) as loss function. We train PickNet with a batch size of 2 using the stochastic gradient descent (SGD) optimizer with a learning rate of 0.001 and a weight decay of 0.0001 on a Nvidia GeForce RTX 3080 GPU. We finally select the weights from the 8-th epoch since it achieve the best performance. We also present the training loss curve. textitTraining Details: For the network architecture of PullNet, we use a ResNet18 [109] as the encoder, followed by a bi-linear upsampling layer pre-trained on ImageNet [111]. We use the binary cross entropy Loss (BCE) as loss function. We train PullNet with a batch size of 2 using the Adam optimizer with a learning rate of 0.001 on a Nvidia GeForce RTX 3080 GPU. We finally select the weights from the 11-th epoch since it achieve the best performance.

We augmented the datasets by image-based transformations as Table 3.1 shows. Since we encodes the direction of pulling by rotating the image so that the pulling direction points to the right, we didn't apply rotations on PullNet dataset. We provide some examples of the data augmentation in Fig. 3.8. Finally, we augmented the PickNet dataset 2X to 85,921 samples and the PullNet data 4X to 22,208.

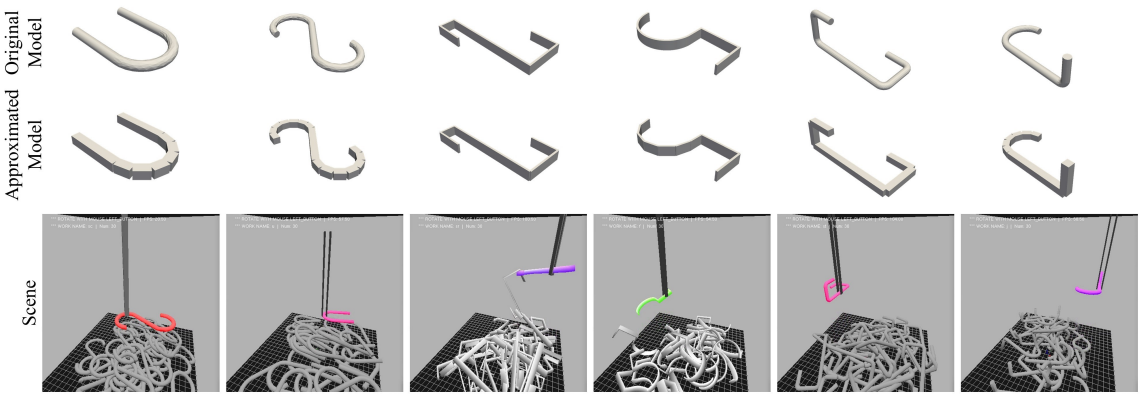


Figure 3.9: Overview of objects and picking scenes in simulation.

3.4.3 Physics Simulator Details

We use NVIDIA's PhysX physics engine to collect synthetic data. We approximate the objects as a set of rigid-body cuboids to (1) balance the trade-off between the simulation accuracy and calculation time and (2) decrease the effects of unreal physical phenomenon when calculating collisions in clutter. We model the parallel jaw gripper as two parallel cuboids and the bin as five rigid-body planes. We manually adjust the size and physical parameters of the fitted rigid bodies to achieve the similar interaction behaviors as that of the real-world. Table 3.2 shows these parameters. We also present the origin model, the approximated model, the clutter scene and the moment of grasping of each object used in the data generation process in Fig 3.9. Since we only use depth maps as dataset, we do not consider the visual appearance of the objects such as textures. Moreover, the shapes of the objects are designed and selected based on the real-world experiments in previous works of bin picking.

Meanwhile, we explain how the grasp is executed in the simulator under physical constraints. The policy includes three parameters: (a) v_g^{close}: Velocity of moving the fingers for closing. (b) v_g^{lift}: Velocity of the fingers for lifting. (c) d_g: Distance between two fingers.

First, the gripper approaches the target object using a 3D position and an orientation angle calculated by our grasp detection algorithm. Then, to let the gripper contact with the object, we set a closing speed v_g^{close} acted as grasping force. If

Table 3.2: Physics Simulator Parameters

Parameters	Value
Bin Static Friction	0.40
Bin Dynamic Friction	0.35
Bin coefficient of restitution	0.05
Bin Size	(22.5,22.5,22.5) cm
Object Static Friction	0.30
Object Dynamic Friction	0.25
Object coefficient of restitution	0.40
Object Density	1 g/cm^3

$v_g^{\text{close}} \rightarrow 0, d_g > 0$, the target is grasped. Next, the gripper lifts with the grasped object by a fixed lifting speed v_g^{lift} and an adjusted closing speed v_g^{close}. v_g^{close} is calculated based on the force where two fingers act on the object. During lifting, if $d_g = 0$, which means the object is slipped from the gripper, v_g^{lift} remains the same while $v_g^{\text{close}} = 0$. Finally, the grasping process is terminated if the gripper is outside the bin.

We also controls the pulling process similar as the grasping or lifting process. All actions are perform in the simulated physical environment.

3.5 Experiments and Results

3.5.1 Experimental Setup

We use a NEXTAGE robot from Kawada Industries Inc. It operates over a workspace captured as a top-down depth image by a Photoneo PhoXi 3D scanner M. A parallel jaw gripper is attached at the tip of the left arm. In physical experiments, we use a PC with an Intel Core i7-CPU and 16GB memory with an Nvidia GeForce 1080 GPU. We use three seen objects and three unseen objects including a non-planar object for testing. When physically implementing the pulling action, we add a wiggling motion during pulling by rotating the wrist joint eight times by 0.1 [rad] with a velocity of 0.35 [rad/s]. The robot stops pulling before the gripper collides with the bin. When transforming the detected 2D pulling vector v to the execution 3D vector u, the robot

slightly raises its arm during pulling.

We compare the performance using two baselines and two versions of our policy. **FGE** is a model-free grasp detection algorithm using a depth image [40]. **EMap** takes a depth image as input and produces a map evaluating where contains entangled objects using edge information [61]. **PickNet** is used in simulation to evaluate the ability to seek untangled objects. **PD** uses PickNet to detect graspable objects in the objects bin and dropping bin. The tangled objects are transported to the dropping bin for separation. If the dropping action is performed three times continuously without solving the entanglement, the robot picks up the objects to the goal bin. **PDP** denotes our complete workflow with both PickNet and PullNet and all three motion primitives.

We define four metrics to evaluate the performance of bin picking. Firstly, let "# Goal attempts" denote the times the robot transports one or multiple objects into the goal bin (# a_{pick} + # a_{pull}), "# Success attempts" denote the times the robot transports only one object into the goal bin. "# Total attempts" means the total times of executing all actions (# a_{pick} + # a_{drop} + # a_{pull}). **Success rate** ($\frac{\text{\# Success attempts}}{\text{\# Goal attempts}}$) evaluates the ability to grasp and transport the untangled objects. **Completion** ($\frac{\text{\# Success attempts}}{\text{\# Objects in main bin}}$) evaluates the ability to accomplish the task of emptying the bin by picking up objects individually. **Action efficiency** ($\frac{\text{\# Success attempts}}{\text{\# Total attempts}}$) evaluates how efficient our policy is of utilizing picking, pulling and dropping actions to complete the task. **Mean Picks Per Hour (MPPH)** evaluates the computation and execution speed of the system.

3.5.2 Simulated Experiments

In the simulation, we conduct a bin-picking task to evaluate the ability to seek untangled objects using FGE, EMap and PickNet. Our simulator locates the grasping target at the output grasp position from the methods and automatically lifts it without grippers, excluding the irregular simulated physics phenomena in grasping or dynamical actions. The bin contains 30 objects and is replenished with the same objects after each attempt. We run 50 picking attempts for each object and 300 for

Table 3.3: Results of Simulated Experiments

	Seen				Unseen			
				Avg.				Avg.
	Success Rate (%)							
FGE	60.0	50.0	44.0	51.3	30.0	60.0	46.0	48.7
EMap	58.0	52.0	54.0	54.7	26.0	52.0	42.0	40.0
PickNet	92.0	86.0	82.0	**86.7**	60.0	88.0	57.0	**68.3**

each method. We evaluate the performance using the success rate only. Note that the success rate in the simulated experiments is equivalent to action efficiency in all methods.

Table 3.3 shows the results of the simulated experiments. PickNet outperforms both baseline methods in success rates. FGE struggles with success rates as it can not discriminate whether the target is entangled. EMap also becomes inefficient in dense clutter. Our policy significantly improves the success rates since the learned affordance map can seek untangled objects for such heavy occlusion. We also observe that unseen objects have success rates lower than objects in the training data for PickNet. However, even if the bin contains no isolated objects, not performing separation motions leads to lower success rates. It demonstrates the necessity of separation strategies to handle unsolvable cases using only PickNet.

3.5.3 Real-World Experiments

Different from the simulated experiments where the robot is required to pick from a bin containing 30 objects every time, the goal of the real-world task is to empty the bin filled with 20 objects. We run three tests for each object using each method.

1) Comparison with Baselines: The results of bin picking in the real world are shown in Table 3.4. PD and PDP outperform baselines FGE and EMap in all metrics. Our policies can perform the task with a success rate of around 90%, almost as high as the success rates of picking simple-shaped objects. Compared with FGE, our policies can detect potentially entangled objects. The affordance map learned by PickNet

can also indicate the state of the entanglement more explicitly than EMap. The proposed separation strategies are useful to improve the performance when picking such entangled objects. The results of completion suggest that our policy outperforms the baselines and doubles their success rates in emptying the bin. Action efficiency suggests our policy PDP performs the best among all methods including PD. PD requires extra actions to separate the entangled objects from the buffer bin while PDP can disentangle and transport by only one action using PullNet. Finally, we compare the speed of each system using the metric MPPH. Our policy can achieve more than 200 mean picks per hour. Specifically, PDP with both networks achieves the highest MPPH than PD since PD requires more actions to complete the task.

2) Does Dropping Help? We investigate the efficiency of dropping as a separation strategy. As Table 3.4 shows, PD uses dropping action as the only separation strategy and achieves a similar success rate and task completion as PDP, which uses two separate actions for both seen and unseen objects. Dropping actions can (1) effectively disentangle the grasped objects and (2) reduce the degree of entanglement from a heavily cluttered environment to a light entangled environment for skillful disentangling manipulation. Both PD and PDP benefit from this action. However, the action efficiency of PD is significantly lower than PDP. The dropping action is an intermediate action that does not contribute to the final object placing stage, requiring the robot to grasp again for placing in the goal bin. Unlike dropping, pulling can separate and place the objects simultaneously. Table 3.5 shows the number of dropping and pulling actions in real-world experiments. PD costs more dropping actions than PDP. If the objects are still entangled after the first time of dropping in the buffer bin, PD repeatedly performs dropping until the objects are untangled. We also observe that dropping cannot solve some difficult entanglement cases, unlike the well-planned skillful motion such as pulling. For these reasons, the robot transports multiple objects to the goal bin, which leads to lower action efficiency.

3) Does Pulling Help? We can observe that PDP, equipped with the two proposed separation strategies dropping and pulling, achieves the best performance especially in action efficiency and MPPH. Unlike dropping actions, pulling requires motion planning from visual observation, which is more interpretable for skillful tasks such as

Table 3.4: Results of Real-World Experiments

	Seen				Unseen			
				Avg.				Avg.
Success Rate (%)								
FGE	67.5	60.0	67.6	65.0	67.4	68.4	42.3	59.0
EMap	71.4	61.5	73.0	70.6	74.9	68.9	42.3	62.0
PD	95.2	85.3	86.9	89.2	89.2	89.2	83.3	87.2
PDP	95.1	85.2	91.7	**90.7**	88.2	93.3	84.5	**88.7**
Completion (%)								
FGE	54.0	30.0	38.3	40.8	48.3	45.0	41.7	44.4
EMap	50.0	45.0	45.0	47.0	58.3	50.0	41.7	50.0
PD	96.7	78.3	88.3	87.8	86.7	93.3	76.7	**85.6**
PDP	96.7	78.3	93.3	**89.4**	86.7	93.3	73.3	84.4
Action Efficiency (%)								
FGE	67.5	60.0	67.6	65.0	67.4	68.4	42.3	59.0
EMap	71.4	67.5	73.0	70.6	74.9	68.9	42.3	62.0
PD	73.4	64.2	73.6	70.4	70.1	63.8	61.8	65.2
PDP	84.0	72.2	75.8	**77.3**	73.2	72.0	68.2	**71.1**
Mean Picks Per Hour (MPPH)								
FGE	171							
EMap	150							
PD	203							
PDP	**220**							

disentangling objects. We observe that dropping is still insufficient for some entanglement patterns while the success rates are relatively higher using pulling as Table 3.5 shows. Pulling contributes to the efficiency of completing the task. Moreover, PDP follows a hierarchical strategy that first drops the objects to create a relatively simple entanglement state and then plans pull actions to further disentangle them. Since the dropping actions decrease the degrees of entanglement in the buffer bin, pulling can effectively disentangle the target and transport it to the goal bin using only one attempt. Pulling and dropping in our policy PDP can be orchestrated together to

Table 3.5: Distribution of Actions in Real-World Experiments

	Seen			Unseen		
						
Dropping Rate (%)						
PD	19.0	25.7	24.7	21.6	26.1	28.4
PDP	10.1	12.3	16.2	15.5	17.9	15.4
Pulling Rate (%)						
PDP	1.45	4.62	2.70	1.41	2.56	7.69
Successful Pulling Rate (%)						
PDP	1.45	3.08	2.70	1.41	2.56	6.15

achieve the best performance for picking tangled-prone objects.

4) Generalization to Unseen Objects: Finally, we evaluate the performance of our policies using unseen objects. Table 3.4 demonstrates that our policies can be generalized to novel objects. Both PD and PDP can recognize untangled objects even if the geometries are not unknown or the self-occlusion for an individual non-planar object (the last column in Table 3.4). Thanks to efficiently collecting a large-scale of synthetic data for training, our networks are capable of handling unknown object geometries and various entanglement scenarios. However, all metrics for unseen objects are slightly lower than seen objects in both PD and PDP. We can assume by the performance of the model-free method FGE that the unseen objects are more challenging. Due to unknown geometries and heavy occlusion, especially for the non-planar object, PickNet might recognize some isolated objects as the entangled objects and performs redundant separation actions. Additionally, the visualized results using PickNet and PullNet are presented in Fig. 3.10 in Fig. 3.11. We also visualize the results from FGE and EMap using the same observation as our policy. It demonstrates that our policy can accurately extract the geometrical information for untangled objects and infer the potentially tangled regions compared with the baselines.

3.5.4 Failure Modes and Limitations

We divide the unsuccessful picking attempts as two types as follows:

(A) **The robot transports nothing to the goal bin.** The situation happens when the grasp poses are not correctly computed. PickNet produce a pixel location for our grasp detection algorithm to compute a 4-DoF grasp. Grasp failure occurs when each grasp orientation around the grasp location collided with the neighbor objects or the visual noise causes miscalculation in transforming 2D pixel locations to 3D locations, leading the gripper collides with the target, the neighbor objects or the bin walls.

(B) **The robot transports multiple objects into the goal bin.** Sometimes due to the sensory noise, the correct locations of each object can not be presented from the depth map, e.g., parts of the objects are missing. Also, PickNet or PullNet sometimes make wrong predictions under some elusive entanglement situation or heavy occlusion. This may comes from the reality differ since the collision modelling of entanglement contact in the simulation still has difference with the real world. The physical execution of pulling sometimes cannot disentangle the objects due to insufficient pulling distance within the bin collisions.

Table 3.6 present a total number of unsuccessful picking attempts through all seen and unseen objects for our policy PD and PDP. The frequency is calculated by the number of unsuccessful picking attempts divided by the total number of attempts. Failure (A) occurs evenly in both policies. Our policy PDP with the entire workflow can significantly decrease the frequency of failure (B), showing the capabilities of disentangling objects.

In the future, this work can be extended to address the following problems: (1) Grasp Failure: The average grasp failure of our policy (PD and PDP) is 4.8%. Grasp fails when there is no collision-free orientation around the predicted grasp point, making the gripper collide with the objects or the environment. We also observe some common failure modes of bin picking, such as objects against walls, providing no space for grasping. (2) Challenging Entanglement Patterns: The proposed PickNet

Table 3.6: Frequency of Unsuccessful Picking Attempts

Method	Explanation	Frequency
PD	(A) Grasps nothing	4.7% (22/462)
	(B) Transport multiple objects	8.8% (41/462)
PDP	(A) Grasps nothing	5.0% (21/422)
	(B) Transport multiple objects	5.9% (25/422)

and PullNet have limitations. First, when the target object forms an endless chain with others, the robot cannot entirely lift the whole chain to drop in the buffer bin. It is also difficult to visually predict how many objects will be grasped based on a top-down depth map. On the other hand, the proposed separation strategies (dropping and pulling) cannot handle several entanglement cases where the objects are tightly wedged together, requiring multi-step actions to solve. (3) Unsuitable Object Shapes: Some object shapes are unsuitable for our policy, such as a tree-like shape since our dataset only includes linear shapes. In the future, we will extend our policy to add objects with various shapes to the training data. It will be interesting to collect data only using a minimal amount of objects based on the entanglement representation of their geometries to efficiently increase the generalization of our policy.

3.6 Summary

This chapter proposes a bin-picking system for efficiently picking tangled-prone objects. We learn a hierarchy bin picking policy from self-supervised simulated data that engages the robot to perform picking or separation actions dexterously based on the observation. Experimental results show the effectiveness of the proposed separation strategies. Our policy outperforms baseline methods in completing the challenging task of emptying the bin with tangled-prone objects with higher success rates and efficiency. We further demonstrate the generalization of our policy using novel objects. In the future, we will expand our policies by leveraging various sensing or more skillful motion primitives for more complex-shaped or deformable objects.

Figure 3.10: **Qualitative results using PickNet and the corresponding physical executions.** Using the same depth map as input, we also present the detected grasps using FGE, the grasps and the entanglement map using EMap (red regions show high possibilities of containing entangled objects). PickNet outputs PickMap and SepMap with their maximum pixel value as the affordance of picking or dropping.

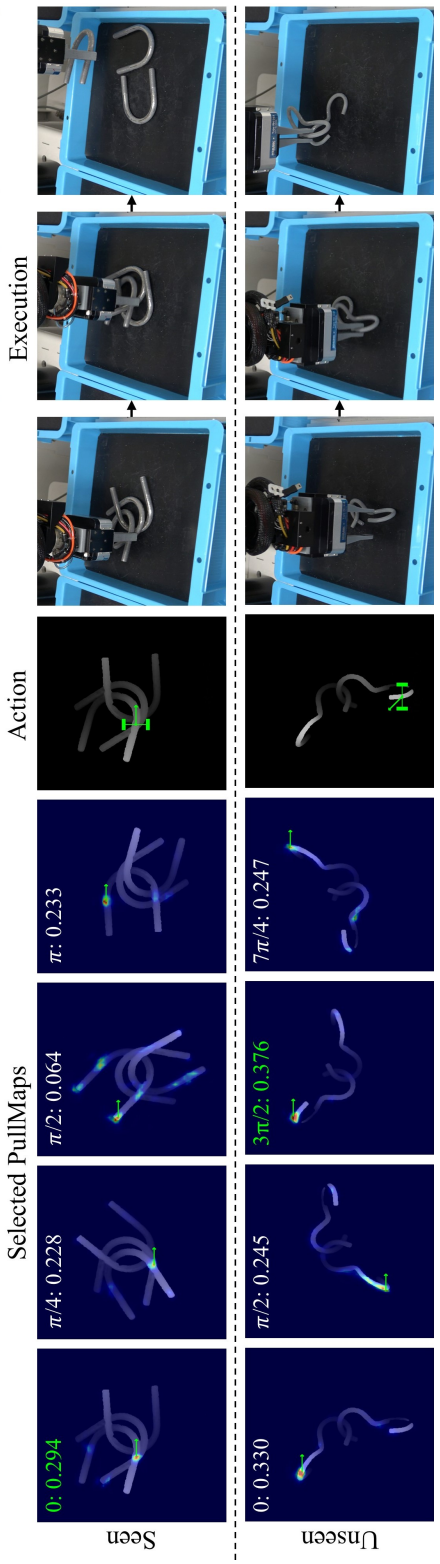


Figure 3.11: **Qualitative results of PullNet and the corresponding physical executions.** PullNet predicts the position and direction for pulling. We rotate the input depth image in eight directions and present four selected PullMaps with their maximum pixel value. The action is selected by the highest scores among all PullMaps. The green arrows denote the pulling directions.

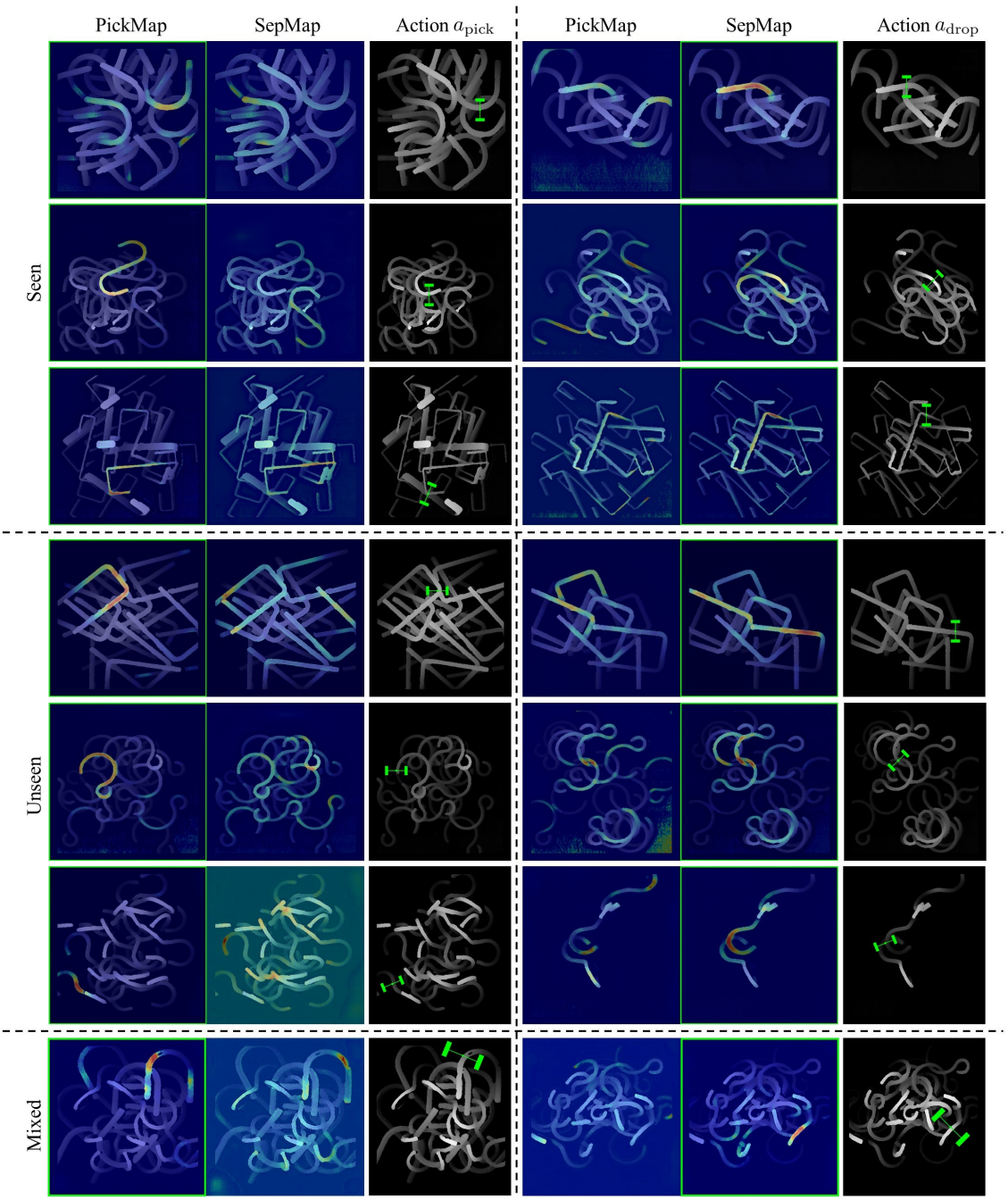


Figure 3.12: More visualized results using PickNet.

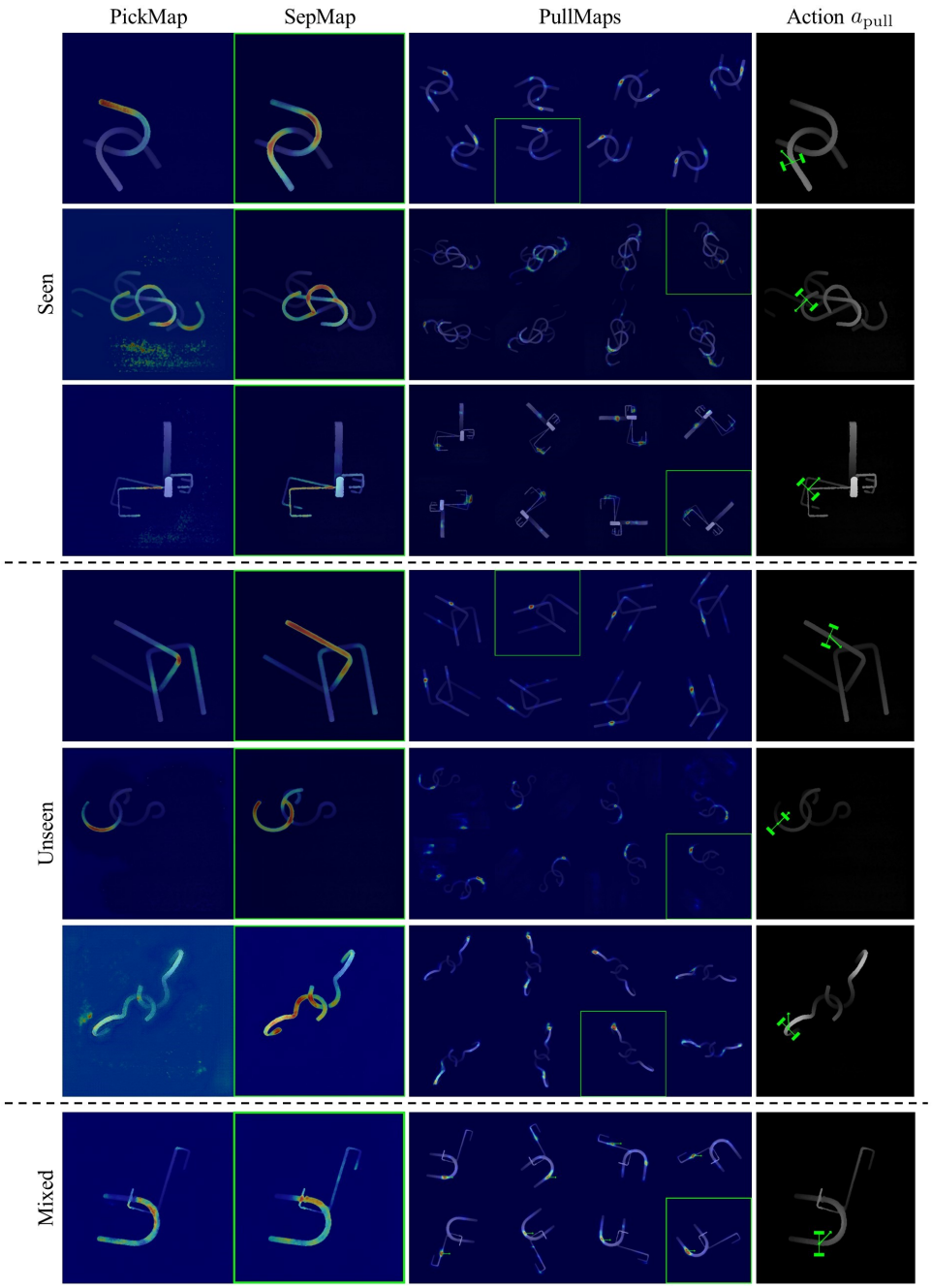


Figure 3.13: More visualized results where the bin contains only entangled objects using both PickNet and PullNet.

Chapter 4

Learning Efficient Policy for Entangled Wire Harnesses

4.1 Introduction

Chapter 2 and 3 tackle the problem of picking entangled rigid objects. However, using deformable objects in bin picking is very important but is still an open problem. For example, a wire harness is an indispensable component used in almost every electric drive product. Fig. 4.1(a) shows its appearance. It comprises a group of bundled wires and multi-conducted connectors and is used for transmitting signals and power. The structure of a wire harness also poses challenges in robotic bin picking: (1) The existence of both deformable and rigid components makes them easily form an entangled clutter in the bin; (2) The complex geometries and deformable nature cause difficulties in 3D modeling; (3) The length of a wire harness often exceeds the operation range of a robot, making it difficult to extract one from the bin. To successfully perform bin picking using wire harnesses, the robot must be equipped with the capability of effectively isolating each from the entanglement. For this reason, the manufacturing industry still relies on human workers to grasp and separate entangled wire harnesses. Therefore, developing an intelligent system to automate this process is highly demanded.

Existing works on industrial bin picking have primarily focused on rigid parts.

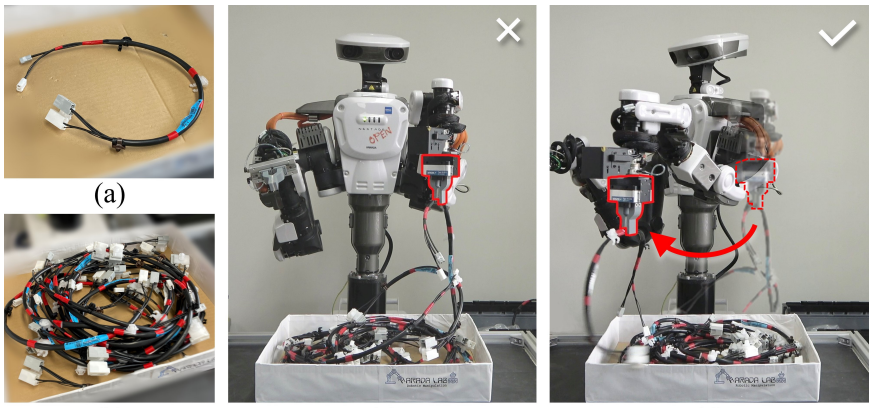


Figure 4.1: (a-b) Wire harnesses are composed of both deformable and rigid components. They get entangled easily in clutter and their length may exceed the robot arm's reach areas. (c) Directly lifting a wire harness causes entanglement. (d) We learn a bin picking policy to efficiently extract an entangled wire harness from an unstructured bin.

These methods grasp objects by avoiding collisions in highly cluttered environments [40, 44, 46, 57, 24, 112]. For picking simple shaped objects, the robot usually lifts the target in the vertical direction after a successful grasp. Different from those objects, wire harnesses involve complex entanglement when randomly placed in a bin. Besides, they are much longer than the rigid parts already automated in bin picking. The physical reach range of the robot in a bin picking working cell is limited for completely lifting them. Simply adapting the existing bin picking strategies shows unsatisfied performance (see Fig. 4.1(c)). Previously, some studies have addressed the entanglement problems but for picking curved rigid parts by avoiding the potentially tangled parts [49, 61]. However, there remain problems for densely cluttered wire harnesses where the bin often contains no isolated objects as Fig. 4.1(b) shows.

Deformable object manipulation has primarily focused on two object classes: 1D (cable, rope) and 2D (fabric, cloth). Several studies adopt specially designed motion primitives to accomplish various manipulation tasks such as knot tying/untying [113, 94, 87], spreading cloth [97] or whipping ropes [114]. Using deformable and long objects in industrial bin picking poses new challenges. The cluttered scenes are more complex due to the entanglement issues caused by their deformable nature. Ray et al. [92] proposed to untangle herbs from a pile using a two-finger gripper. Takahashi

et al. [93] proposed a learning-based separation strategy for grasping a specified mass of small food pieces. Motivated by modeling and manipulating deformable linear objects, studies on visually processing wire harnesses start with segmenting or generating synthetic data for wire harnesses with pure linear shapes. Although some works have addressed the factory automation problems for wire harnesses [115, 116, 117], robotic wire harnesses picking is less studied. This work proposes a novel and efficient bin picking strategy to deal with wire harnesses. For wire harnesses with complex geometries, obtaining precise models or training in simulation remains difficult. Alternatively, employing a real robot to collect large-scale data is time-consuming. Annotating ground truth labels is also challenging due to the lack of entanglement metrics.

This chapter tackles these challenge by (1) designing an effective motion to untangle wire harnesses in clutter and (2) learning a policy to perform bin picking tasks with higher success rates and lower execution time. The key components of our system are:

- A post-grasping action to untangle wire harnesses. Instead of lifting in the vertical direction, the robot separates the entangled objects in the horizontal direction. The action continuously follows a circle-like trajectory to extract the target within the limited robot's reach range. Fig. 4.1(d) shows this process.

- A bin picking policy to infer an optimal grasp and a post-grasping action from a depth image. Our policy can prioritize grasping the untangled objects, avoid grasping at the bad positions (e.g., the ends of the object) and reason the extracting distance to reduce the execution time for a successful picking. Additionally, we train the policy with real-world data by leveraging active learning for satisfying convergence.

Real-world experiments suggest our policy can significantly improve the average success rates and reduce operation time compared with baselines. Our contributions are three-fold.

- We develop a unique bin picking system that can disentangle wire harnesses from dense clutter.

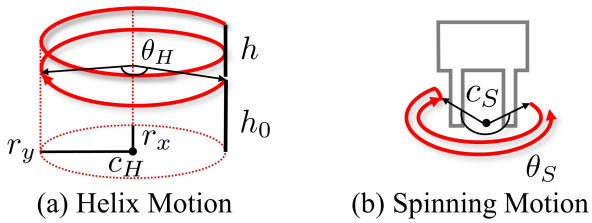
- Instead of lifting the target in the vertical direction after grasping, our policy proposes to simultaneously lift and move in the horizontal direction for separating wire harnesses.
- We learn a policy using real-world data to infer the optimal actions, which further improves bin picking efficiency.

Code, videos and datasets can be found at https://xinyiz0931.github.io/aspnet.

4.2 Motion Primitives for Disentangling

When a robot tries to isolate small and rigid objects from a bin, it can lift them in a vertical direction after a successful grasp. However, this movement is insufficient for isolating long and flexible objects like a wire harness, whose length exceeds the bin picking workspace. To extract such objects, the required motion primitives must be designed to (1) provide enough space for effectively disentangling long objects and (2) handle various tangle patterns. Instead of directly lifting, the horizontal movement of the gripper can help pull the target object out. The possible positions of the gripper should also remain in the outer part of the parts bin during disentangling. In the end, two motion primitives are proposed for effectively disentangling a long and flexible object:

Helix motion: $\psi_H = (H, \theta_H)$ where H denotes the helix trajectory represented by (c_H, r_x, r_y, h_0, h) and θ_H denotes the execution angle following the trajectory (see Fig. 4.2(a)). c_H denotes the base center of H and r_x, r_y constrain the smallest and largest radius from the center. h denotes the height of H. The helix starts after the gripper lifts the target and reaches h_0. The stop point of the helix is determined by the execution angle θ_H. It is a post-grasping motion where the gripper simultaneously lifts and pulls following a helix-like trajectory. Let the gripper move around the bin while holding an entangled object. Part of this object is also moving outside the bin. When the gripper continuously moves like drawing circles, the grasped object can be disentangled softly along a side angle. Fig. 4.3(a) shows that this



(a) Helix Motion (b) Spinning Motion

Figure 4.2: **Formulation of motion primitives**. (a) Helix motion primitive $\psi_H = (H, \theta_H)$ where the helix trajectory is defined as $H = (c_H, r_x, r_y, h_0, h)$. (b) Spinning motion primitive $\psi_S = (\theta_S, c_S)$.

movement provides adequate space to pull the target (green) out of the entangled objects (yellow). Meanwhile, we also observe that other entangled objects remain in the bin during or after this process, making the workspace clean for the next picking.

Spinning motion: $\psi_S = (c_S, \theta_S)$ where c_S denotes the position of the gripper tip and θ_S denotes the one-way rotation angle of the spinning (see Fig. 4.2(b)). The robot performs a two-way spinning about the axis that is vertical to the robot workspace. The gripper spins to handle the entanglement that may be occluded from the observation. As Fig. 4.3(b) shows, when the rigid components of the wire harness still slightly hang on the others after the helix motion, an extra spinning can help separate them with less execution time. It can also handle the length of a wire harness by extracting it inside a limited working cell.

4.3 Learning Bin Picking Policies

The goal of our bin picking policy is to pick up a single wire harness at a time by inferring the optimal grasp and action from current entanglement situation. If the scene contains isolated objects, the robot prefers directly lifting them after grasping. Otherwise, the robot can infer disentangling actions and grasp poses to extract the target from the bin. Given a top-down depth image o as observation, we formulate our bin picking policy π with a trained model parameterized by τ using:

$$a^*, g^* = \pi_\tau(o) \tag{4.1}$$

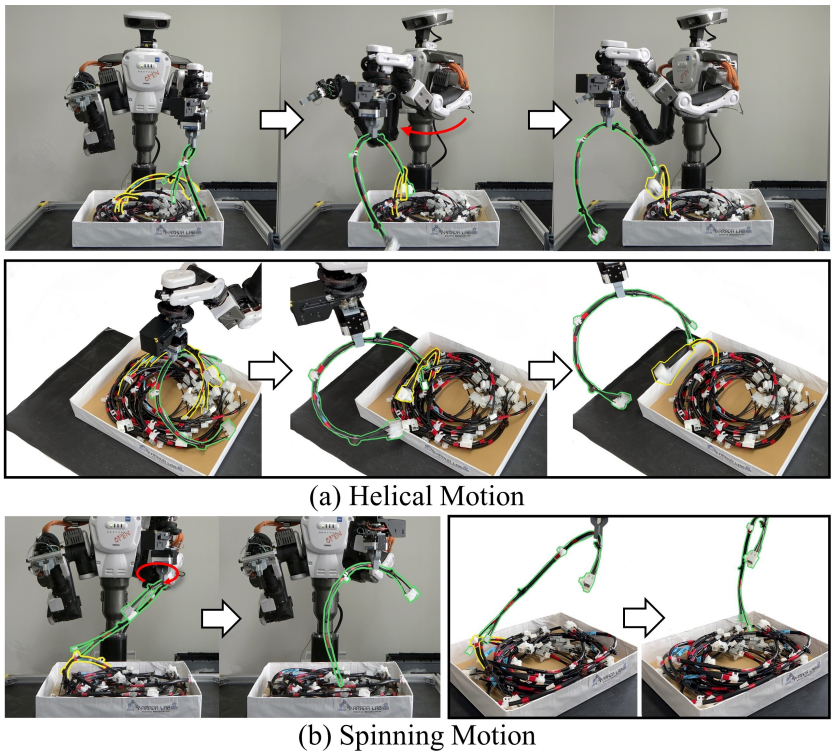


Figure 4.3: **The proposed motion primitives can handle two properties of wire harnesses: tangle-prone and length.** (a) The robot separates an entangled wire harness from a gentle angle following a helix trajectory. (b) A spinning motion is performed when the target's connectors slightly hang on the other objects.

where the outputs are an action a^* and a grasp g^* with the maximal task effectiveness. The action a comprises the proposed motion primitives. Fig. 4.4 shows the three essential modules in our policy:

Module I. Model-Free Grasp Detection: A grasp detection algorithm using a depth image without object models.

Module II. Action Success Prediction (ASP): A trained model using real-world data that predicts the success possibilities of the disentangling actions.

Module III. Action-Grasp Inference: A method to infer the action-grasp pair with the highest effectiveness using the trained ASP model.

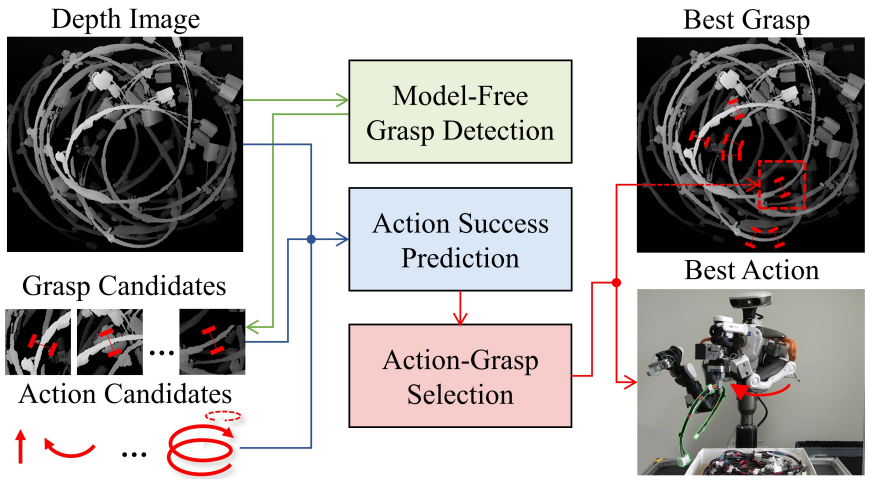


Figure 4.4: **Overview of our policy.** Given a depth image of an unstructured bin, Model-Free Grasp Detection module samples a set of non-collision grasp candidates. Then, Action Success Prediction module takes a depth image, grasp candidates and action candidates as input and evaluates the success possibility for each action-grasp pair. Finally, Action-Grasp Inference module ranks these pairs and outputs the optimal action and grasp.

4.3.1 Model-Free Grasp Detection

We select Fast Graspability Evaluation (FGE) [40] - a model-free approach to detect collision-free grasps. FGE calculates pixel-wise graspability scores by convoluting a gripper's template of contact and collision areas with the input depth map. A grasp composes a pixel location $g = (u, v)$ on the depth map and a rotation angle ϕ indicating the gripper's orientation. We transform (u, v, ϕ) to the grasp with four degrees of freedom (g_x, g_y, g_z, g_ϕ) denoting the grasp point and the gripper's orientation at the robot coordinate frame. This module outputs a set of grasps ordered by their FGE scores.

4.3.2 Action Success Prediction (ASP)

1) Action Formulation: We formulate each disentangling action a with a motion scheme ψ and two parameters as follows:

$$a = (\psi, \theta_H, \theta_S) \mid \psi = \{\psi_H\} \text{ or } \{\psi_H, \psi_S\} \tag{4.2}$$

Table 4.1: Action Parameters and Execution Details

	a_{dl}	a_h	a_{hs}	a_f	a_{fs}	a_{tf}	a_{tfs}
							
ψ	-	$\{\psi_H\}$	$\{\psi_H, \psi_S\}$	$\{\psi_H\}$	$\{\psi_H, \psi_S\}$	$\{\psi_H\}$	$\{\psi_H, \psi_S\}$
θ_H	0	π	π	2π	2π	4π	4π
θ_S	0	0	$\pi/2$	0	$\pi/2$	0	$\pi/2$
Time (s)	1.2	2.3	2.8	5	5.5	8.2	8.7
SR	31/80	47/80	60/80	65/80	66/80	70/80	72/80
$\mathcal{A}$	0	1	2	3	4	5	6

* Time (s) - Execution time of performing the action trajectory.
* SR - Success Rate of picking a single object.
* $\mathcal{A}$ - Action complexity.

where the robot only performs the helix motion ψ_H or performs the spinning motion ψ_S after ψ_H. Note that directly executing ψ_S after grasping may not be effective since the extracting displacement of the target object is small. Six separation actions $a_h, a_{hs}, a_f, a_{fs}, a_{tf}, a_{tfs}$ are crafted using two motion primitives and a direct lifting action a_{dl}. Table 4.1 shows their notations and illustrations. We use M to represent the collection of these seven actions.

2) Action Parameter Determination: To determine the parameters of each action and search for the best action, we define a numerical metric **action complexity** for exploring the trade-off between success rates and execution time. Let $\mathcal{A}(a)$ denote the action complexity of the action $a \in \{a_{dl}, a_h, a_{hs}, a_f, a_{fs}, a_{tf}, a_{tfs}\}$. It is defined by assuming that actions with larger θ_H or θ_S involve higher complexity. To reduce the search cost during exploration, we assume that the action complexity linearly scales with the success rate of each action. We find this linear relationship by executing 80 physical attempts for each action as Table 4.1 presents. Then, we use this hypothesis to determine the action parameters experimentally. Specifically, we predefine a set of possible values of θ_H, θ_S experimentally for our policy to select. θ_H can be selected from $\{0, \pi, 2\pi, 4\pi\}$ and θ_S can be selected between $\{0, \pi/2\}$. Note that the other parameters of the motion primitives $H = (c_H, r_x, r_y, h_0, h)$ and c_S are fixed in our policy. Finally, we assign integers 0 to 6 as the action complexity for

the discrete actions from a_{dl} to a_{tfs}. The action parameters and execution details are included in Table 4.1.

Our policy can explore the **optimal** action by minimizing the action complexity as much as possible. Let us consider a case when the robot performs a_{tf} to extract an entangled object. Suppose the target object is entirely disentangled after a full circle (a_f) while the robot still needs to perform the second circle. Thus, the current observation only requires a_f as the **optimal** action to ensure a successful separation with less execution time, while a_{tf} is a **redundant** action which can also solve the entanglement but costs more time. We can observe that an optimal action has lower action complexity than a redundant action. Thus, the optimal action is required to untangle the target with minimal action complexity.

3) Prediction Model: The inference of the optimal action without object models should be conditioned on the grasp locations. Action Success Prediction (ASP) is trained to predict if the action-grasp pair can successfully separate the target. ASP learns a function parameterized by τ:

$$p = f_\tau(o, g, a) \tag{4.3}$$

where the input is a depth image $o \in \mathbb{R}^{224\times224\times3}$ with triplicated depth values across three channels to match with the default input size of the image encoder's backbone, a pixel-wise grasp pose $g = (u, v) \in \mathbb{R}^2$, a categorical action $a \in \mathbb{R}^7$ and the output is a success possibility in the range of $[0, 1]$. We encode the image using a ResNet-50 backbone [109], the grasp point using a single fully-connected layer with 256 units, and the categorical action using a fully-connected layer with 14 units. Then we concatenate the output from all three branches and feed it to a fully-connected layer with 256 units and produce an action success possibility.

4) Training via Active Learning: The dataset for training ASP is entirely collected from real-world experiments. Each sample has a depth image o, a grasp location g, a labeled action a and a binary success metric $S = \{0, 1\}$. We execute each action for the clusters with 6, 10, 12 and 18 objects. We label each attempt with success ($S = 1$) or failure ($S = 0$) depending on if the robot picks a single wire harness. Due

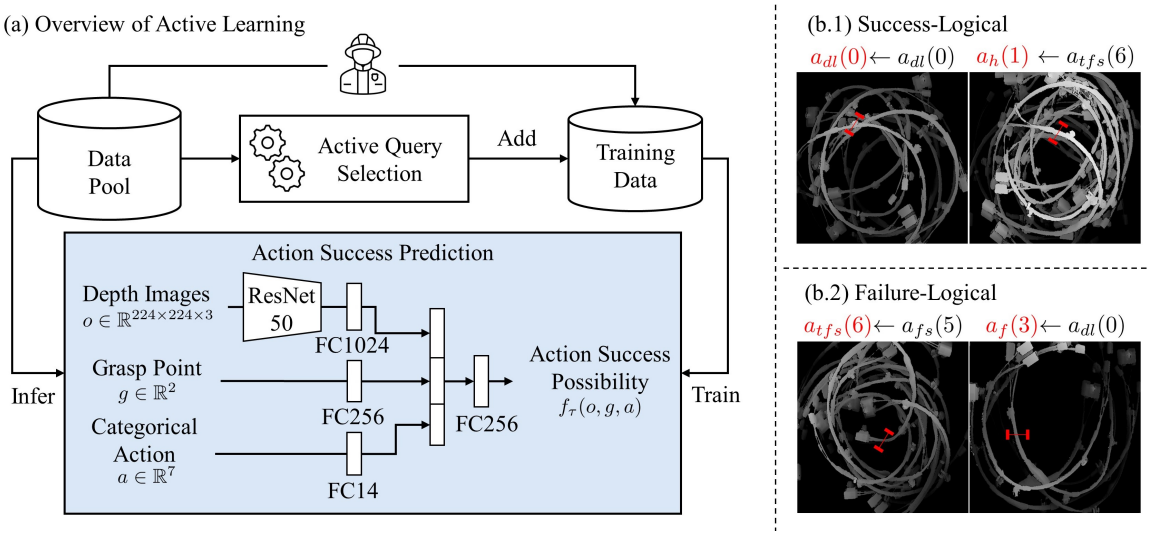


Figure 4.5: Overview of our proposed active learning.

to this data collection manner, some samples in the dataset are labeled with redundant actions instead of optimal actions. To deal with this problem, we leverage active learning to train the ASP model, making it possible to predict optimal actions using this dataset. Fig. 4.5(a) shows how the active learning works. Generally, we first select several samples manually as training data to train the model, use the trained model to predict the remaining samples, query and transfer samples for training and fine-tune the model repeatedly. Specifically, we manually select the initial training data with approximately optimal actions. Note the number of samples for each action is roughly equal. Let data pool denote the left samples except for the training data. After training, we query the samples in the data pool and transfer the **logical** samples to the training data. Here, a sample (o, g, a, S) can be determined as **success-logical** or **failure-logical** using the trained model τ and our proposed Action-Grasp Inference module (Section IV.C). Let $a_p = \texttt{ActionGraspInference}(o, g, M, \tau)$ denote the predicted action:

- Success-logical: $\mathcal{A}(a_p) \leqslant \mathcal{A}(a)$. For samples labeled with $S = 1$, the labeled action a is a redundant action compared with the predicted action a_p (Fig. 4.5(b.1)).

Algorithm 2: Active Learning Algorithm

input: Data pool, transfer ratio r, actions M
output: ASP model τ

```
Select training data from data pool
Train ASP model τ using training data
while data pool is not empty do
    N ← number of samples in data pool
    i ← 0
    while i ⩽ r × N do
        Randomly select {o, g, a, S} from data pool
        a_p ←ActionGraspInference(o, g, M, τ)
        if S = 1 and A(a_p) ⩽ A(a) then
            // Success-logical
            Move to training data, i = i + 1
        else if S = 0 and A(a_p) > A(a) then
            // Failure-logical
            Move to training data, i = i + 1
    Fine-tune ASP model τ using training data
```

- Failure-logical: $\mathcal{A}(a_p) > \mathcal{A}(a)$. For samples labeled with action a and failure $S = 0$, the predicted action a_p has higher action complexity (Fig. 4.5(b.2)).

During each iteration, as the number of logical samples increases, the model performance of predicting the optimal actions also improves. We define a transfer ratio r representing the ratio of the number of samples that would be transferred in each iteration to the number of samples in the current data pool. The iteration stops when the data pool is empty or early stops before overfitting. Algorithm 2 shows the detail of training ASP via active learning.

4.3.3 Action-Grasp Inference

At this point, we've obtained a set of grasp candidates, action candidates and the scores of each action-grasp pair. Our policy then needs to determine which action-grasp pair can be executed. This module infers all possible action-grasp pairs to

guarantee a successful picking with minimal action complexity:

$$a^*, g^* = \texttt{ActionGraspInference}(o, G, M, \tau) \tag{4.4}$$

where the inputs are a depth image o, a collection of actions M, grasp candidates G with FGE scores from the Model-Free Grasp Detection module and ASP model τ. This module first predicts the action success possibilities of all action-grasp pairs $P = f_\tau(o, G, M)$. If all possibilities in P are lower than the threshold p_{thld}, which means all action-grasp pairs cannot solve the entanglement, we select the grasp with the highest FGE score and the most complex action a_{tfs}. Otherwise, the best solution is determined by the action-grasp pair with the lowest action complexity. If multiple grasps share the same action complexity, we select the pair with the highest FGE score.

4.4 Experiments and Results

We conduct several real-world experiments to answer the following three questions: (1) How does the learned ASP model perform using active learning? (Section 4.4.1) (2) Does our bin picking policy perform more accurately and effectively than baselines? (Section 4.4.2) (3) How does our method qualitatively improve the performance of picking wire harnesses? (Section 4.4.3)

4.4.1 ASP Model Performance

Our dataset contains 722 samples. We set the ratio of active learning $r = 0.4$ and use a simple decision threshold of $p_{thld} = 0.5$ over the softmax of each action's success possibility to classify success (1) or failure (0). We train the network using binary cross-entropy loss function and the Adam optimizer. We stop training after three times of fine-tuning as it achieves the best performance. Fig. 4.6 shows the accuracy and loss during active learning. The gray curve refers to the Initial Model (IM) trained using manually determined samples, which would be potentially accurate but lack robustness due to fewer data. The green line indicates the Final Model (FM),

Table 4.2: Details and Validation Results of Active Learning

	IM	2nd	3rd	**FM**
# Samples in Training Data	282	453	558	618
# Samples in Data pool	428	257	152	92
Ratio of Success-Logical (%)	78.5	85.7	87.8	88.9
Ratio of Failure-Logical (%)	85.1	80.1	90.1	91.7

Table 4.3: Average Predicted Scores using Validation Samples

	a_{dl}	a_h	a_{hs}	a_f	a_{fs}	a_{tf}	a_{tfs}
$S = 1$	0.352	0.489	0.702	0.750	0.783	0.730	0.787
$S = 0$	0.257	0.375	0.581	0.636	0.678	0.606	0.685

which performs the best as the fine-tuning goes on since it converges to IM but with higher data-driven accuracy.

Moreover, Table 4.2 shows the details of each iteration in active learning. Row 1-2 shows the number of samples used as the training data and left in the data pool. Particularly, 92 samples left in the data pool after the final fine-tuning are used to validate all models by checking the number of logical samples. Row 3-4 shows the ratios of logical samples increase with the fine-tuning process. Finally, Table III validates our hypothesis that more complex actions correspond to higher success rates. We respectively present the average scores predicted by FM for each action. FM can correctly predict an ascending order of possibilities as the action complexity increases. We can observe that a_{fs}, a_{tf}, a_{tfs} share similar scores since the validation samples contain 18 objects at most. a_{tfs} does not show a significantly high score due to the accumulated low scores when all predictions fail and a_{tfs} is forced to be selected.

4.4.2 Bin Picking Performance

1) Physical Experiment Setup: We use a NEXTAGE robot from Kawada Industries Inc. The robot is required to grasp objects from the parts bin lying in front of it and transport them to another bin located on its left side. The robot's left arm

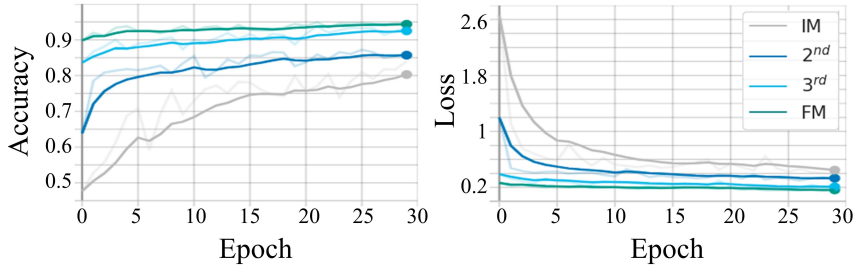


Figure 4.6: Accuracy and loss of each model during action learning.

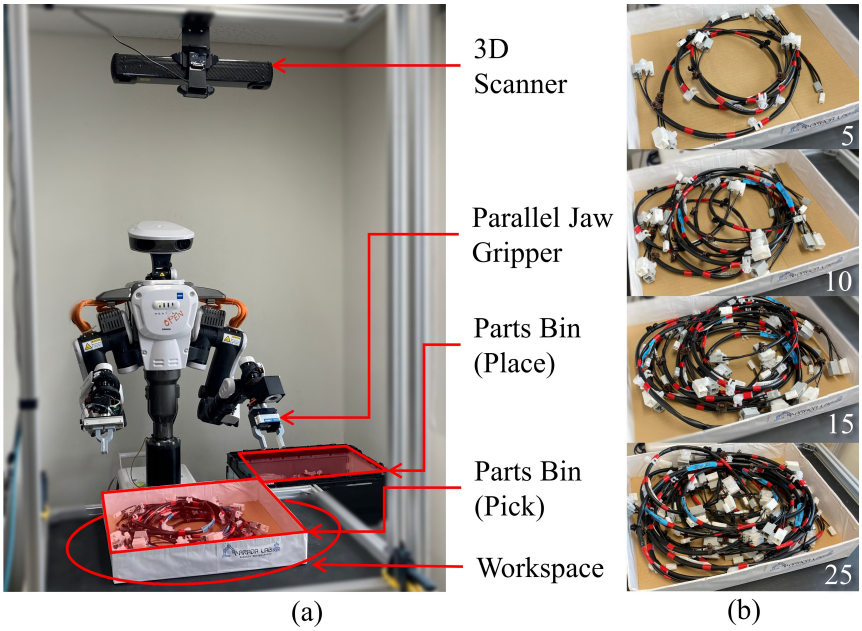


Figure 4.7: Physical experiment setup for bin picking.

operates over a workspace captured as a top-down depth image by a Photoneo PhoXi 3D scanner M. A two-fingered parallel gripper is attached at the arm tip. The setup is shown in Fig. 4.7(a). The length of the wire harness used in this work is 74cm. After performing the analysis and physical experiments, we fix the parameters of the proposed trajectory as $c_H = (0.525, 0.065)$[m], $r_x = 0.1$m, $r_y = 0.225$m, $h_0 = 0.32$m, $h = 0.14$m as well as the speed of the action since they yield high task effectiveness. We sample several waypoints on the trajectory and plan motions with a uniform velocity. We use a PC with an Intel Core i7-CPU and 16GB memory without GPU for real-world experiments and a PC with an Intel Core i5-6400 CPU, 16GB memory

and an Nvidia GeForce 1080 GPU for learning.

Three baselines are presented. **DL** (directly lifting) uses FGE to detect the grasp point and executes by directly lifting (a_{dl}). **RAND** executes a random action and the grasp with the highest FGE score. **TFS** only executes the most complex action a_{tfs} and the grasp of the highest FGE score. We also present three versions of our policy. **Ours-IM** is our policy using the initial model in active learning while **Ours-FM** uses the final model. **Ours-FM-R** denotes Ours-FM with a recovery module using force feedback. After performing the predicted action, we record the force from an F/T sensor mounted on the robot's wrist to determine if the grasped wire harness is still entangled. If there exists a sudden increase of force, the target is not disentangled and the robot places it back to the parts bin.

We leverage two metrics to evaluate the bin picking performance. **Success rate** refers to the number of successful attempts of picking up a single object divided by the number of attempts of placing. **PPH (Pickings Per Hour)** is the number of successful attempts the robot can execute in one hour. Additionally, **Avg. $\mathcal{A}$ (Average action complexity)** is evaluated how the action complexity predicted by our policy varies under different entanglement scenarios.

2) Task Design: We prepare two real-world bin picking tasks. **Consecutive picking** aims to empty the bin filled with respectively 5, 10, or 15 objects. The robot picks up objects one by one until the bin is empty. **Randomized picking** refers to picking up objects from the bin filled with respectively 18-20, 20-22 and 22-25 objects. After each picking, we reload the bin and shuffle the wire harnesses to provide randomness during the task. It can encourage the robot to confront different patterns of entanglement as much as possible. Fig. 4.7(b) shows the bins filled with different numbers of wire harnesses.

3) Comparisons with Baselines: Table 4.4 compares the performance of the three versions of our policy and three baselines in success rate and PPH. For consecutive picking where the goal is to empty the bin, Ours-FM and Ours-FM-R significantly increase the average success rate from 56.7% to 87.3% and 88.1% compared to DL. TFS achieves higher success rates than Ours-FM but has lower PPH since TFS only

Table 4.4: Performance of Bin Picking Experiments

		5 Objects			10 Objects			15 Objects		
	Method	SR (%)	PPH	Avg. $\mathcal{A}$	SR (%)	PPH	Avg. $\mathcal{A}$	SR (%)	PPH	Avg. $\mathcal{A}$
Consecutive Picking	DL	64.0	128	-	60.0	92	-	56.0	108	-
	RAND	88.0	115	2.3	92.0	117	2.5	76.0	99	2.8
	TFS	96.0	133	-	92.0	127	-	90.0	124	-
	Ours-IM	84.0	131	0.8	76.0	117	2.3	74.0	111	2.9
	Ours-FM	88.0	**156**	**0.8**	88.0	**140**	**2.8**	86.0	**143**	**2.3**
	Ours-FM-R	**89.8**	**154**	**0.8**	**89.8**	**142**	**2.8**	**84.4**	**123**	**2.9**
		18-20 Objects			20-22 Objects			22-25 Objects		
	Method	SR (%)	PPH	Avg. $\mathcal{A}$	SR (%)	PPH	Avg. $\mathcal{A}$	SR (%)	PPH	Avg. $\mathcal{A}$
Randomized Picking	DL	46.6	93	-	40.0	80	-	23.3	47	-
	Ours-FM	**86.7**	**113**	**2.9**	**80.0**	**112**	**3.3**	**73.3**	**103**	**4.3**
	Ours-FM-R	**92.6**	**108**	**2.6**	**91.7**	**103**	**4.5**	**76.9**	**107**	**4.6**

* SR - Success Rate of picking a single object.
* PPH - The number of Picking a single object Per Hour.
* Avg. $\mathcal{A}$ - Predicted average action complexity.

executes the time-consuming action a_{tfs}. Especially in the latter half of a continuous picking task when fewer objects remain in the bin, our policy can shorten the execution time by inferring adequate actions. Ours-FM-R also has lower PPH since this policy needs extra actions to place the entangled objects back in the parts bin. Furthermore, the average action complexities for the predicted actions using RAND, Ours-IM, Ours-FM and Ours-FM-R are also presented in Table 4.4. The average action complexity for 5 objects is significantly lower than that for 10 and 15 objects. As the number of objects in the bin increases, the action complexity of the predicted action increases. It demonstrates that entanglement frequently occurs when the bin contains more objects and requires more complex actions. We also observe that the failed attempts by baselines always drag objects outside the workspace, requiring human workers to rearrange after each attempt. Our policy helps maintain a relatively clean workspace during the consecutive picking thanks to the horizontal separation and our action-grasp inference algorithm.

For randomized picking, we compare the performance of Ours-FM and Ours-FM-R with a DL baseline as Table 4.4 shows. More objects are involved in this task than consecutive picking. Thus, the possibilities of encountering complex entanglement patterns become higher. Ours-FM completes the task with 80% accuracy and 109 PPH, almost twice higher than DL. The results suggest that our policy can grasp the tightly intertwined objects in dense clutter. All three proposed modules collaboratively contribute to efficient bin picking from perception to manipulation planning. However, as the number of objects increases, both metrics of Ours-FM decrease. Due to heavier occlusions and visual noise, the detected grasp candidates become fewer and some entanglement patterns can hardly be recognized from the depth image. Despite this, the most complex action a_{tfs} can still strive for success. Additionally, Ours-FM-R outperforms Our-FM in success rate especially when the number of objects increases thanks to the recovery module but has lower PPH. When the bin contains more than 22 objects, Ours-FM-R shows a higher success rate and PPH than Ours-FM, indicating the feedback module can help further improve the bin picking performance.

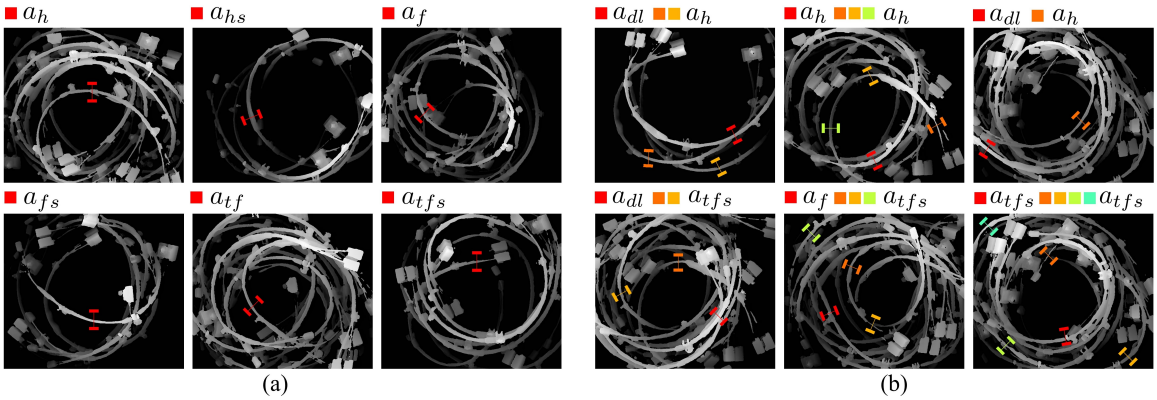


Figure 4.8: **Qualitative results.** (a) Ours-FM predicts the optimal action-grasp pairs for each action. (b) Ours-FM predicts the best action and grasp marked using red in real-world experiments. All action-grasp pairs are presented using the same colors.

4.4.3 Qualitative Analysis

1) Visualized Results: Visualized results of picking attempts with grasps, actions and input depth images are presented. First, Fig. 4.8(a) shows the predicted action-grasp pairs of each action. It demonstrates that our policy infers the actions not only by analyzing the object number in the scene but also by reasoning about the occlusions around the input grasp point. Additionally, if the robot grasps close to the wire harness's end, our policy tends to predict more complex actions since this case may require the gripper to handle the length by moving a larger distance. Then, Fig. 4.8(b) shows a set of successful pickings with the reasoned action-grasp candidates ranked by descending prediction scores. The optimal action-grasp pairs inferred by our policy are marked as red. Our policy can recognize the objects barely entangled with others that only require a_{dl}. As for the scenes that do not contain such objects, our policy can reason the entanglement situation and predict the proper actions. When the predicted scores of all action-grasp pairs are lower than p_{thld}, our policy executes a_{tfs} and grasp with the highest FGE score, where the target is likely on the top of the pile.

2) Novel Wire Harnesses: To demonstrate the breadth of our method, we utilize Ours-FM for two unseen wire harnesses. They differ from those used for training in lengths and structures but have similar components (e.g., deformable cables and

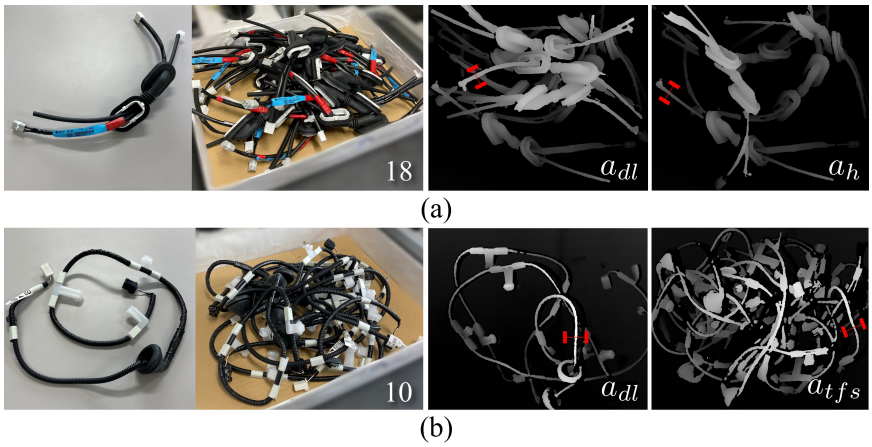


Figure 4.9: **Novel types of wire harnesses and the predicted action-grasp pairs by our policy.** (a) Short wire harnesses. (b) Long wire harnesses.

Table 4.5: Predicted Average Action Complexity (Avg. $\mathcal{A}$) for Two Types of Unseen Wire Harnesses

Type	Length (cm)	5 Objects	10 Objects	15 Objects
Short	45	0.7	1.3	1.7
Long	115	4.8	4.6	-

rigid connectors). Fig. 4.9 shows two novel wire harnesses and the corresponding action-grasp pairs predicted by our policy. Table 4.5 shows their length and the average action complexity of prediction with different object numbers. In the case of shorter objects (see Fig. 4.9(a)), our model does not predict actions with too higher complexity. The robot tends to select a_{dl} and a_h to pick up objects. Since this type of wire harness is less tangle-prone, the accuracy of picking them primarily relies on the grasp detection module while our policy can handle the potential entanglement. On the other hand, for long wire harnesses (Fig. 4.9(b)) whose length exceeds our bin picking working cell, Table 4.5 suggests that our policy tends to output more complex actions. However, even a_{tfs} is still insufficient to separate each. More complex manipulation strategies are needed for such objects.

4.4.4 Haptic Feedback Evaluation

In Ours-FM, we determined both the manipulation policy and grasping pose from the obtained depth image of the pile. After a robot grasps the target wire harness, a robot just replays the predetermined manipulation policy. However, to increase the success rate of picking, we should obtain some sensor information after a robot grasps the target wire harnesses and modify the manipulation policy according to the sensor information. Therefore, we have implemented a new method **Ours-FM-R** utilized force feedback with our policy to evaluate the performance.

Specifically, Ours-FM-R combines our policy (Ours-FM) and a recovery module. Fig. 4.10 shows the workflow of this module. Ours-FM-R computes the best grasp and action the same as Ours-FM. After the robot completes the predicted action at q_s, we set a waypoint q_e before the robot moves to the parts bin for placing. During the gripper moves from q_s to q_e, we record the force F_z in the z-axis every 10ms. Then, F_z is used to determine if the grasped wire harnesses are still entangled when F_z contains a sudden increase over a threshold F_{thld}. If both conditions are satisfied, the robot places the entangled objects back in the parts bin for the next picking.

Fig. 4.11 shows the three examples of recorded force, the coordinate of the force sensor and the start/end positions of force recording. Red blocks denote the period of the force recording. Note that we set the threshold $F_{thld} = 0.2N$ based on the weight of a single wire harness. We only evaluate the force along the z axis (blue line) since the force change on the x or y axis is not significant and may be affected by other phenomena rather than entanglement. Fig. 4.11(a) shows no significant change in force when picking and placing untangled wire harnesses. Fig. 4.11(b) shows the process of picking and successfully separating entangled wire harnesses. We can also observe the force change during the separation process in the yellow block. The recorded force in the red block indicates the entanglement is solved. Fig. 4.11(c) shows the case where the separation cannot disentangle the wire harnesses. Then, the recovery module detects the entanglement and places the wire harnesses back in the parts bin. F_z has a sudden increase over F_{thld} in the red block. The robot returns the grasped wire harnesses to the parts bin as recovery. Fig. 4.11(c) also presents the robot grasping wire harnesses at the start point q_s and endpoint q_e of

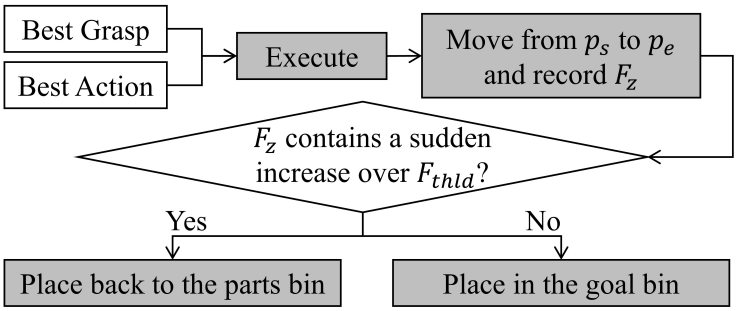


Figure 4.10: Overview of the added recovery module for the method **Ours-FM-R**.

force recording and the coordinates of the force sensor.

We evaluated Ours-FM-R in **success Rate**, **PPH** and **Avg.** $\mathcal{A}$ as the red blocks show in Table 4.4. We also present the detailed version of Table 4.4. It presents four numbers respectively denoting (1) # Success: number for attempts that successfully place a single object, (2) # Grasp Success: number of attempts successfully grasp an object, (3) # Place Attempts: number of attempts that the robot tries to place objects and (4) # Total Attempts: number of total attempts. Note that #Place Attempts equals # Total Attempts without recovery actions. The success rates presented in Table 4.4 is defined as $\frac{\text{\# Success}}{\text{\# Place Attempts}}$. Moreover, to conveniently go through the evaluation results, here, we compare two metrics: success rate and action efficiency (defined as $\frac{\text{\# Success}}{\text{\# Total Attempts}}$). Note the success rate and action efficiency are the same for Ours-FM.

In the consecutive picking, the success rates of Ours-FM and Ours-FM-R are similar since the predicted actions by our policy can solve most cases of entanglement. The recovery module of Ours-FM-R successfully recovered from the entanglement seven times in 150 attempts. In the randomized picking, the success rates of Ours-FM-R significantly increase when the number of objects increases. Among 90 picking attempts, the robot recovered from the entanglement 13 times. The entanglement detection we craft is more adaptive to the scenario where the bin contains more objects. Additionally, Ours-FM-R has a low average PPH and action efficiency than Ours-FM since extra picking attempts are needed to place the objects back in the parts bin. For the clutter containing 22-25 objects, Ours-FM-R outperforms Ours-FM in all metrics, indicating that the recovery module could help improve the performance

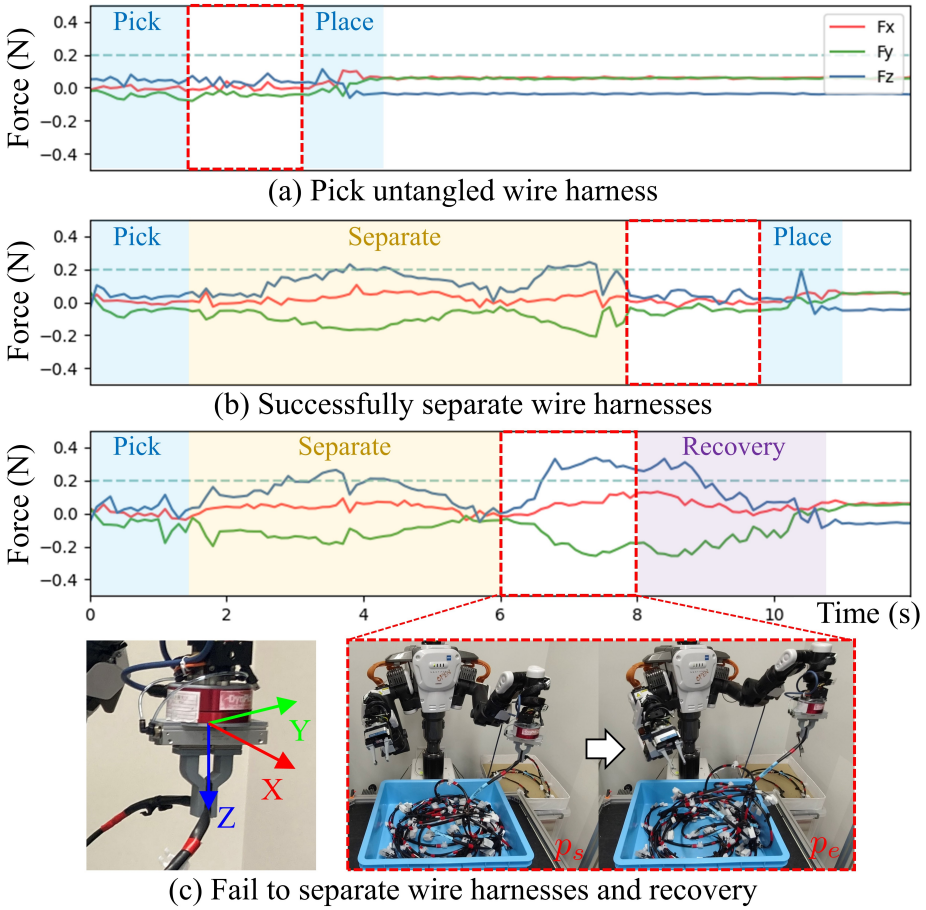


Figure 4.11: Recorded force for three cases using **Ours-FM-R**.

further. However, this method cannot detect other entanglement patterns that are difficult to notice from force signals. We also observe that the force during the separation is somewhat irregular. Thus, it is challenging to determine if the entanglement exists during the separation action only based on heuristics such as the one used in Ours-FM-R. More sophisticated entanglement detection methods should be explored in the future, maybe with the help of visual feedback.

4.4.5 Failure Modes and Limitations

We observe four failure modes in the physical experiments.

- Objects outside of the bin: The input image of the ASP model does not include

the complete objects.

- Grasp failure: The grasp failure rate is 2.1% (24/1170). A grasp fails when the robot grasps multiple objects in hand or grasps nothing. It mainly comes from vision sensor's noise and heavy occlusion.
- Tightly wedged objects: The target tightly inserts another one's cable bundles or rigid components, making it extremely difficult to be disentangled.
- Action prediction failure: Our policy sometimes predicts the wrong actions for separation due to visual noise or heavily occluded objects.

Our policy also has limitations. First, for long wire harnesses, the robot fails to extract them from the entanglement since their length exceeds the robot's reachable areas. Second, the training phase is unique and conditioned on the structure of the objects in the dataset. It would be difficult to adopt our current policy to wire harnesses with completely different geometries.

We divide the reasons causing failure modes and limitations into two categories and provide future extensions. (1) Poor visual prediction for heavily occluded clutter: We will extend our policy by using multi-sensory inputs other than vision-only predetermined policy and force-only feedback control. We will also consider online closed-loop learning and more effective recovery methods to further improve the robustness of our policy. (2) Insufficient motion primitives: the proposed motion primitives cannot solve some complex cases and the reach range of a single robot manipulator is limited. We will consider more effective motion primitives using dual-arm or involving dynamics. It would also be interesting to design more general motion primitives to utilize our policy on various wire harnesses with different geometries.

4.5 Summary

This chapter presents a novel bin picking system for grasping and separating entangled wire harnesses. We design an efficient post-grasping action for disentangling the target in clutter, learn a policy from real-world data to reason the extracting distance

and produce the optimal action and grasp from a single depth image. Real-world experiments suggest that our policy can successfully untangle the intertwined wire harnesses from different cluttered scenes and pick them up one at a time with high accuracy.

Chapter 5

Dynamic Manipulation with Haptic Feedback for Entangled Wire Harnesses

5.1 Introduction

This chapter addresses the problem of picking entangled wire harnesses and extend the work in Chapter 4. In Chapter 4, we observe some failure cases and we want to solve these failure and further improve the success rates in this work. We consider the challenges for developing a robust and efficient bin picking system for wire harnesses. Object recognition and grasp detection becomes challenging in such complex scenarios involving with rich contact and environmental uncertainties. In the case of the robot grasping the end of the objects, executing disentangling motions becomes insufficient. Simulated training or obtaining models for wire harnesses still remains an open problem while training in the real world is time-consuming. Additionally, the manipulable range of the robot in a standard bin picking cell has limited the maximum length of wire harnesses that can be handled. These difficulties led manufacturing industries to rely on human workers to manually separate entangled wire harnesses in the assembly processes.

In the previous work [3] (Chapter 4), a sequential policy is learned to perform a

circle-drawing trajectory to disentangle the wire harnesses. However, as the number of objects in a bin increases or when adapting to unseen objects types, the patterns of entanglement become unpredictable, making visual recognition and the circling motion insufficient. Moreover, wire harnesses often exceed the robot's reachable range, further diminishing the performance of quasi-static motion primitives for disentangling them. Therefore, more effective motion primitives and multiple sensing capabilities are highly demanded to ensure a robust, accurate and versatile bin picking system for wire harnesses.

Other studies on deformable object manipulation have successfully accomplished challenging manipulation tasks [94, 96, 89, 118, 119, 114, 113, 120, 97, 121]. Grannen et al. [96] proposed a method to untangle knots based on learned keypoints and bimanual manipulation. Seita et al. [122] learned a sequential policy that utilizes pick-and-place actions to smooth cloth. However, quasi-static manipulation have difficulties in dealing with heavy self-occlusion in 1D deformable objects and higher dimensions in cloth or fabric. Dynamic manipulation, which involves higher velocities and considers inertia effects, has shown effectiveness in manipulating deformable objects [113, 114, 97, 120, 121]. Chi et al. [114] developed an iterative policy for goal-conditional manipulation using visual feedback. Chen et al. [123] proposed a learning framework that enables a single arm to dynamically smooth cloth. Yamakawa et al. [113, 120] introduced an analytic control algorithm for performing high-speed manipulation tasks. Viswanath et al. [88] proposed a shaking motion to dynamically reduce loops and reveal knots in entangled cables. Building upon the advantages of these manipulation strategies, we have focuses on manipulation multiple deformable objects.

This chapter proposes a bin picking system for entangled wire harnesses with the following key components:

- Two motion primitives: swing and regrasping, specifically designed for disentangling of long wire harnesses. The swing motion with a high velocity can dynamically extract the target from the clutter. On the other hand, regrasping enables the robot to grasp the target at its middle section, creating sufficient space for disentangling process.

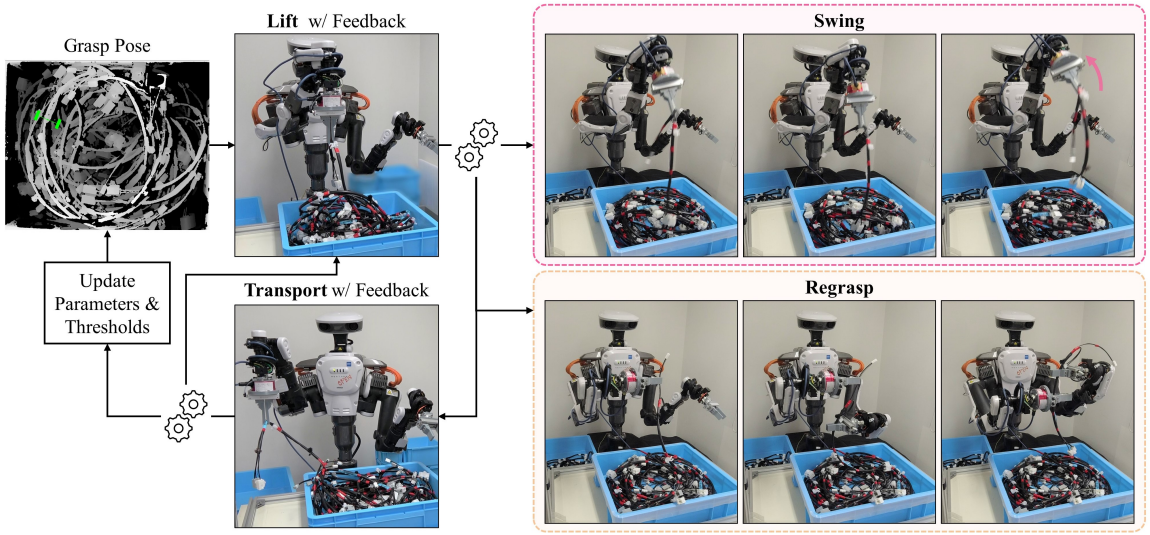


Figure 5.1: Overall process of picking entangled wire harnesses.

- A closed-loop system that utilizes haptic feedback to detect entanglement in real-time and tunes the system parameters online. Unlike open-loop policies without error recovery, our system closes the loop by incorporating force feedback, enhancing the robustness and efficiency for picking entangled wire harnesses.

The primary contribution of this work is a unique bin picking system for wire harnesses that leverages dynamic and bimanual manipulation as disentangling strategies.A haptic-guided closed-loop algorithm is proposed with failure detection and recovery in real-time. Real-world experiments suggest the proposed method can significantly improve the average success rates compared with our priors work.

5.2 A Closed-Loop System with Dynamic and Bimanual Manipulation

The goal of this study is to grasp wire harnesses individually from dense clutter. This section presents the manipulation planning of two disentangling motion primitives, the closed-loop workflow with force monitoring and online parameter tuning

process. These modules collaborate together to ensure the robustness, effectiveness and generalization of wire harnesses picking.

5.2.1 Dynamic Manipulation for Disentangling

We design two motion primitives. **Swing**, which involves high speed and acceleration, can effectively separate the entangled wire harnesses. **Regrasping** motion enables the robot to adjust the grasp pose from the end of the objects to the middle, making it effective for subsequent actions.

1) Swing: We use a parametric action primitive to describe the movement of the robot. The action space for the swing primitive is $a = (\theta, \omega, n)$, where $\theta = (\theta_3, \theta_4, \theta_5)$ is the moving angles for the i-th joint in one robot arm. ω denotes the permissible angular velocity across all joints while n indicates the number of times the swing motion is repeated. Specifically, θ_5 denotes the angle for the yaw rotation for the last joint of the robot arm, which can be seen as a "spinning" motion. Meanwhile, θ_4, θ_5 are roll and pitch angles for the last arm joint and can perform a "whipping" motion. Three joints of the robot arm moves simultaneously and dynamically extract the objects from the clutter. Note that we initially set the values of θ and they can be tuned during the execution. The swing motion is illustrated in Fig. 5.2(a-b).

Regrasping: Regrasp can switch the grasp pose to the middle of the target after the robot grasps the end of the object. Regrasping relies on force feedback rather than vision. The process is illustrated in Fig. 5.2(c-d). Let the right arm of the robot, equipped with a force sensor, acts as the main arm while the left arm is the support arm. The main arm first grasps a wire harness and move to a pre-determine pose. Next, to determine the correct end of the object, the main arm's wrist spins by π [rad] and we record the torque signals of both poses. The correct object end is determined by the minimal torque. Then, the support arm move to the pose where its gripper is below the main arm's gripper, moves downward and ensures the object is securely held in the gripper. It then closes the gripper and pulls the object upward. Finally, the main arm moves to the pose where its gripper is below the support arm's gripper and grasp the objects. After the support arm returns to its the initial pose,

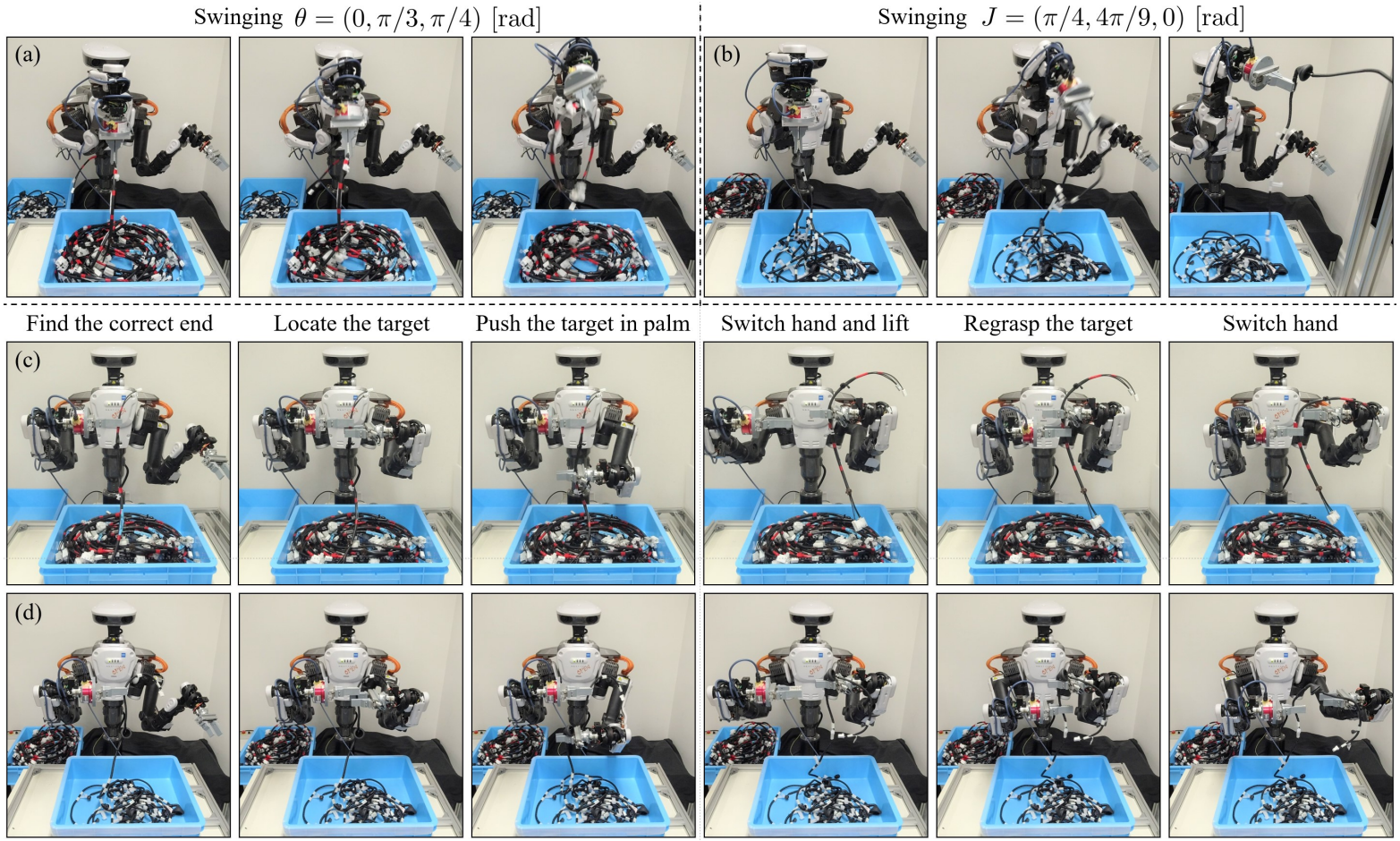


Figure 5.2: **Disentangling motion primitives.** (a-b) Swing motions using different parameters for two types of wire harnesses. The robot's movements can rapidly separate the target from entanglement. (c-d) Regrasping motions for two types of wire harnesses.

Algorithm 3: Workflow of A Picking Attempt

input: Depth map, F_{stop}, F_{fail}
Detect grasp pose from input depth map;
$N_{\text{transport}} \leftarrow 0, L \leftarrow$ empty list;
while *True* **do**
  `Lift` with force monitoring;
  if $F_z^0, F_z^1, ..., F_z^t < F_{\text{stop}}$ **then**
    Stop;
    `Swing`(θ, ω, n);
  else if $\dot{F}_z \to 0$ or $N_{\text{transport}} > 2$ **then**
    `Regrasp`;
  `Swing`$((0, 0, \hat{\theta}_5), \hat{\omega}, 2)$;
  `Transport` with force monitoring;
  $N_{\text{transport}} \leftarrow N_{\text{transport}} + 1$;
  if $F_z^0, F_z^1, ..., F_z^t < F_{\text{stop}}$ **or** $F_z^t < F_{\text{fail}}$ **then**
    Finish;
  L.append(F_Z^t);
  `Update`$(F_{\text{stop}}, F_{\text{fail}}, L)$;
  $\theta \leftarrow \theta + \delta\theta$;

the regrasping attempt is completed and the main arm successfully adjusts the grasp pose.

5.2.2 Closed-Loop System Workflow

The workflow of our proposed system is shown in Fig. 5.1 and Algorithm 3. First, we obtain the depth image of the bin filled with wire harnesses and detect a set of collision-free grasp [40, 3]. The robot then grasps the target and lifts it while monitoring the force F_z in z axis vertically to the workspace. If F_z exceeds the threshold F_{stop}, the robot immediately stops and performs the swing motion to disentangle the target. Otherwise, If the haptic feedback does not provide an stopping point, meaning either the robot grasps a single wire harness or the regrasping motion is needed. Thus, to determine if the robot should execute the regrasping motion, we use $F_z = \{F_z^0, F_z^1, ..., F_z^t\}$ over a time series t recorded during the lifting process. We apply a median filter to F_z and calculate the gradient $\dot{F}_z$. If $\dot{F}_z$ approximates

Algorithm 4: Online Parameter Tuning

Function `Update`($F_{\text{stop}}, F_{\text{fail}}, L$)**:**
  if not stop when `lift` and `transport` **then**
    `// Minimizes the gradient of` L
    $F_{\text{fail}} \leftarrow \arg\min_F \dot{L}$;
  if not stop when `lift` **and** stop when `transport` **then**
    $F_{\text{stop}} \leftarrow F_{\text{stop}} - \delta F$;

zero, indicating that the target is too long to exert any forces on the gripper, the robot leverages regrasping to change the grasp position to the middle. Then, the robot tries to transport the wire harness to the goal bin while monitoring the force. However, if F_z does not exceed F_{stop} during transporting and also does not exceed F_{fail} before dropping into the goal bin, the robot performs a successful attempt of picking a single wire harness. Otherwise, we increase the swing parameters θ, adjust the force threshold $F_{\text{stop}}, F_{\text{fail}}$ and restart from the beginning (lifting while monitoring F_z). Additionally, if the robot fails to disentangle the objects after two transporting attempts, it executes the regrasping motion.

5.2.3 Online Parameter Tuning

In Line 15-17 of Algorithm 3, we introduce an online parameter adjustment algorithm to improve the robustness of the robot. The initial force thresholds are manually set: F_{stop} represents the minimal force where the entanglement occurs, while F_{fail} approximates the weight of grasping a single object. Algorithm 4 outlines our online parameter tuning process.

First, before the robot drops objects into the goal bin, we monitor the force F_z^t. If the robot successfully transports only one object without any stops during both lifting and transportation ($F_z^t < F_{\text{fail}}$), we adjust the value of F_{fail}. After each attempt in this scenario, we obtain a list L of F_z^t. F_{fail} is updated by minimizing the gradient of L towards zero. The value of F_{fail} gradually converges to a value and the updating process is stopped when the gradient no longer changes. Next, if the robot does not stop during lifting ($F_z^t < F_{\text{fail}}$) but encounters a stop during transporting ($F_z^t < F_{\text{stop}}$),

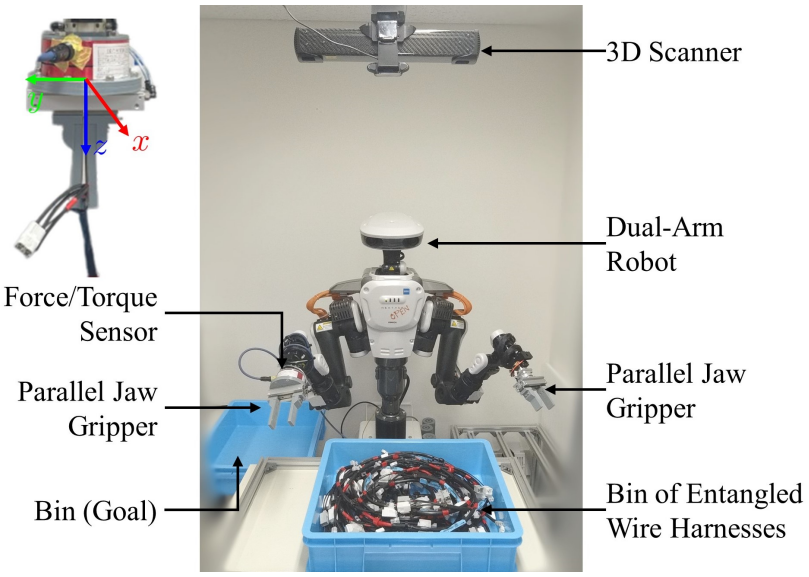


Figure 5.3: Our experimental setup.

it indicates that the threshold for detecting entanglement is not sensitive enough and should be tuned to a lower value. In this case, we set a residual force parameter δF and adjust the threshold as follows: $F_{\text{stop}} = F_{\text{stop}} - \delta F$.

In addition to the force thresholds, our algorithm also adjusts the swing parameters during each attempt. When the transporting process is unsuccessful, the robot will disentangle the grasped objects using increased swing angles. We increase the θ by a predefined residual angle $\delta\theta$, while ensuring that the adjustments remain within the velocity limits of the robot's arm. We also implement an additional motion $\texttt{Swing}((0, 0, \hat{\theta_5}), \hat{\omega}, 2)$ before the transporting process. This two-way spinning motion with pre-defined $\hat{\theta_5}, \hat{\omega}$ can can provide additional assurance.

5.3 Experiments and Results

5.3.1 Experiment Setup

We conduct real-world experiments using two types of wire harnesses measuring 74 cm and 120 cm in length, shown in Fig. 5.4. The bin used in the experiments is filled with a maximum of 8 and 40 objects of each one. We design two specific bin picking

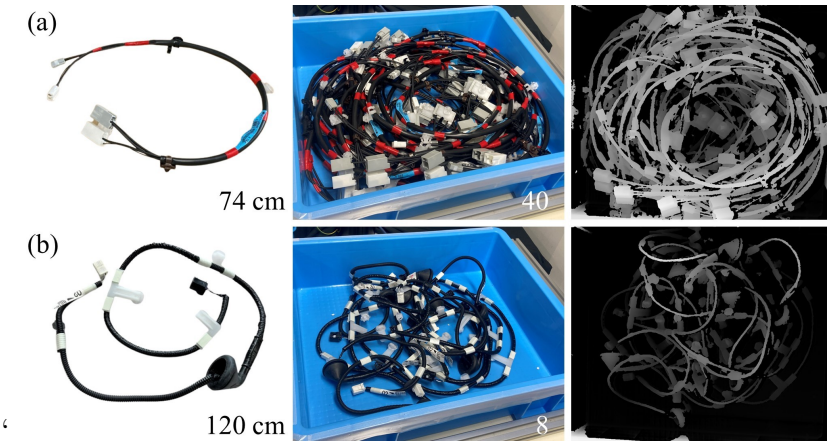


Figure 5.4: **Wire harnesses used in the experiments.** (a) A wire harness that also used to train ASPNet in [3]. (b) A challenging wire harness.

tasks for evaluation:

- **Emptying**: The goal is to completely empty the bin filled with entangled wire harnesses.
- **Standard**: After each successful picking, we reload the bin with the same number of wire harnesses and randomly shuffle them. This ensures that the robot encounters different entanglement patterns throughout the task.

Our experimental setup is shown in Fig. 5.3. We use a NEXTAGE robot from Kawada Industries Inc. The robot's arms operate within a workspace that is captured as a top-down depth image using a Photoneo PhoXi 3D scanner M. Each arm is equipped with a parallel jaw gripper at its tip. A force sensor DynPick WEF-6A200-4-RCD is mounted at the wrist of the robot's right arm. We use a PC equipped with an Intel Core i7 CPU, 16GB of memory and an Nvidia GeForce 1080 GPU for the physical experiments. We fix the parameters empirically for the experiments. The incremental angles for the swing motion is fixed at $\delta\theta = \pi/18$ [rad], while the incremental forces for online parameter tuning are set to $\delta F = 0.1$ [N]. The initial parameters for swing motion is $\theta_3, \theta_4\, \theta_5 = \pi/4, \pi/3, \pi/3$ [rad], $\omega = \pi/2$ [rad/s], $n = 2$. The initial force thresholds are $F_{\text{stop}} = 3$ [N], $F_{\text{fail}} = 1$ [N].

We present two baselines and two ablated version of our method:

Table 5.1: Success Rate Comparison

Object	Task	Method	# Objects	Success Rate (%)	# Attempts			
					Lift	Circle	Swing	Regrasp
	Emptying	Lift-G	15	53.6% (15/28)	15	-	-	-
		Circle-A	15	83.3% (15/18)	1	14	-	-
		Ours-G	40	**94.7% (36/38)**	10	-	29	7
		Ours-A	40	**97.4% (38/39)**	17	-	22	5
	Standard	Lift-G	25	23.3% (7/30)	7	-	-	-
		Circle-A	25	73.3% (22/30)	-	22	-	-
		Ours-G	40	83.3% (25/30)	2	-	28	5
		Ours-A	40	**86.7% (26/30)**	5	-	19	9
	Emptying	Circle-A	8	0.0% (0/20)	-	-	-	-
		Ours-A	8	**80.0% (16/20)**	-	-	5	12
	Standard	Circle-A	8	0.0% (0/20)	-	-	-	-
		Ours-A	8	**65.0% (13/20)**	-	-	9	10

- **Lift-G**: This open-loop method uses Fast Graspability Evaluation (FGE) [40] to detect collision-free grasps and directly lifts the target after grasping, without incorporating haptic feedback.
- **Circle-A**: This open-loop method, described in [3], leverages ASPNet to infer the lowest action complexity of each grasp and execute a circling motion to disentangle the wire harnesses.
- **Ours-G**: Our closed-loop policy incorporates dynamic and bimanual manipulation with haptic feedback for real-time adjustments. It utilizes the FGE algorithm [40] for grasp detection.
- **Ours-A**: Our closed-loop policy optimizes the grasp pose using ASPNet [3]. Instead of simply selecting the top rank from FGE, ASPNet evaluates the action complexity of each grasp and selects the lowest one.

5.3.2 Comparisons with Baselines

Table 5.1 presents the performance comparison among our methods and the baselines. In the emptying task, both Ours-G and Ours-A demonstrate significant improvements

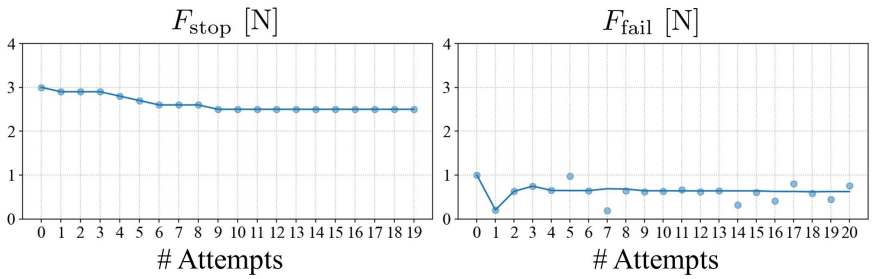


Figure 5.5: Results of the online tuning procedure.

in success rates compared to the baselines. The average success rates of emptying task achieve 96.1% and 80% respectvely for two types of wire harnesses. Since Lift-G and Circle-A without haptic feedback are unable to handle dense clutter, we evaluate their performance using less objects instead. Notably, our policy outperforms the other methods, even under a higher degree of entanglement due to effective disentangling motion primitives. Especially, for wire harnesses with a length of 120 cm, our policy improves the success rate from 0% to 80%. The swing motion plays a crucial role in separating the objects, and the regrasping motion leads to a remarkable success rate increase for longer wire harnesses.

For the standard task where the robot must confront more complex entanglement patterns, our policy demonstrates a significant improvement in success rates for both types of objects. Different from the emptying task, where fewer objects remain in the bin at the later half of the task, the standard task keeps a higher degree of entanglement throughout the picking process. The real-time haptic feedback mechanism facilitates the recovery from failed disentangling actions. Every module in our proposed closed-loop system works collaboratively to achieve efficient bin picking from perception to manipulation planning. Additionally, Ours-A outperforms Ours-G in success rates overall since Ours-A can avoid grasping the ends of the objects, leading to more sufficient disentangling actions.

5.3.3 Benefits of Closed-Loop System with Haptic Feedback

Fig. 5.5 provides the adjustment of force thresholds throughout the consecutive picking process. The force threshold F_{stop} gradually decreases over time and eventually

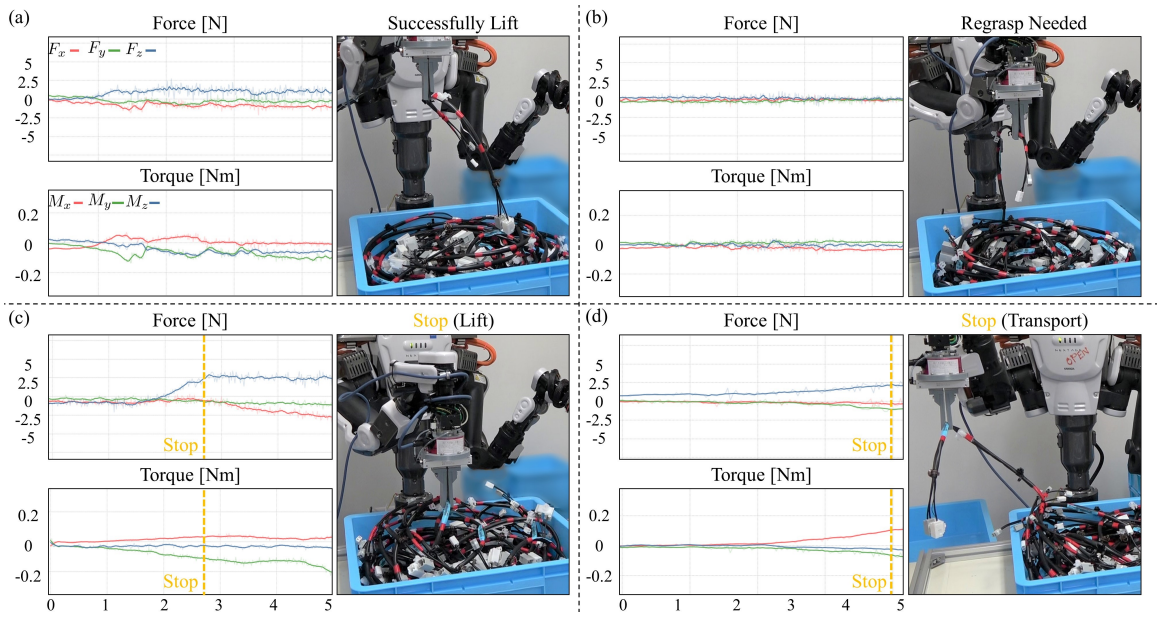


Figure 5.6: **Visualized outputs from the force sensor during different scenarios.** (a) The robot successfully grasps and lifts an isolated object without entanglement, as indicated by the smooth increase in F_z (blue line). (b) When grasping at the end of an object, F_z remains near zero without a significant increase. (c) The robot detects entanglement (marked in yellow) while lifting the target and immediately stops, as F_z shows a sharp increase exceeding the threshold $F_{\mathbf{stop}}$. (d) During transportation of the target to the goal bin, the robot stops after detecting entanglement, again indicated by F_z exceeding the threshold $F_{\mathbf{stop}}$.

stabilizes at a certain value. This value represents the minimum force at which entanglement occurs. On the other hand, the distribution of F_{fail} is more scattered, and we optimize it by minimizing the gradient to zero. The optimization process leads to the convergence of the threshold to a stable value, which closely approximates the weight of a single ob. The online parameter tuning acts as a valuable supervisor, enhancing the overall performance and generalization of our system when adapting to previously unseen objects.

We also visualize the force monitoring process during execution. In Fig. 5.6, we illustrate the robot's actions and the corresponding force readings. The robot utilizes force signals to detect whether the entanglement occurs (Fig. 5.6(c-d) or not (Fig. 5.6(a-b)). By incorporating force feedback, we effectively mitigate errors and mistakes

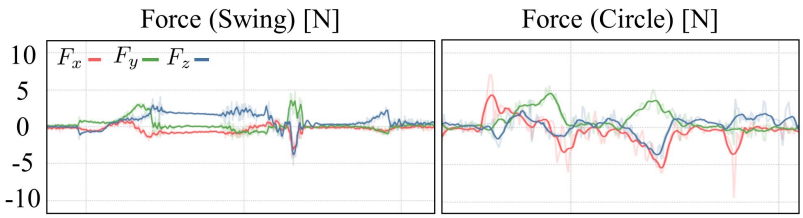


Figure 5.7: **Force comparison between swing and circling motions.** Swing motion exerts a moderate force on the wire harnesses, thereby preventing damage to them during the picking process.

that may arise when relying solely on visual predictions.

5.3.4 Benefits of Dynamic Motion Primitives

Table 5.1 also includes the attempt numbers for performing each disentangling motion primitive in the baselines and our methods. The results demonstrate that incorporating swing and regrasping achieves higher success rates compared to the circling motion. The swing motion effectively disentangles the target and loosens dense entanglements, particularly for longer wire harnesses. Additionally, regrasping enhances the accuracy of the swing motion by switching to a more suitable grasping position on the target. We can also observe that the longer wire harness has more regrasping attempts. This dual-arm manipulation can effectively addresses the issue of length and also allows for directly pulling the target from the entanglement.

In addition, we evaluate the force applied to the objects during the circling motion and our proposed dynamic motion primitives. The force readings show that the proposed motion primitives can successfully complete a picking attempt with a force of only 5 [N], which almost the same as the quasi-static circling motion. Applying less force to the objects reduces the potential damage to the wire harnesses, thereby minimizing wear and tear during the assembly process.

5.3.5 Benefits of ASPNet

Table 5.2 shows the normalized action complexity predicted by ASPNet [3]. ASPNet effectively predicts that longer objects require more complex actions. The result

Table 5.2: Normalized Action Complexity Predicted by ASPNet

# Objects	Action Complexity 	
5	0.133	0.800
10	0.467	0.767
15	0.483	0.800

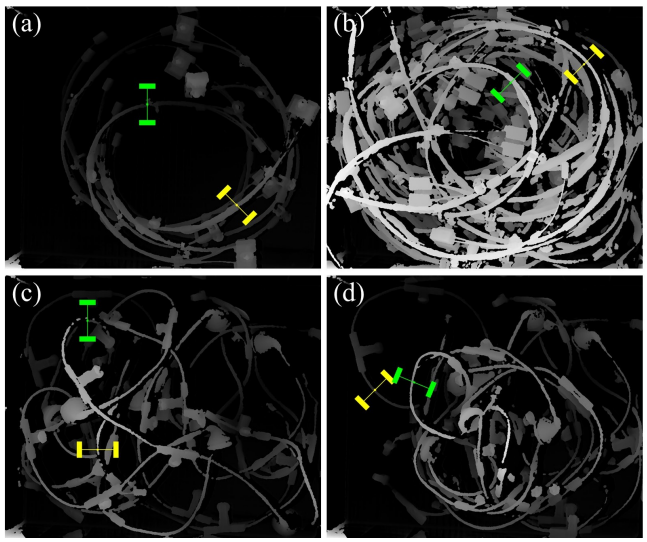


Figure 5.8: **Grasps computed from FGE (yellow) and ASPNet (green).** ASPNet tends to find objects located at the top of the heap and aims to grasp them at their middle part.

demonstrates that ASPNet can effectively predict the complexity of the entanglment from observation. Although we did not specifically match the action complexity with each action in this study, we leverage this learned vision model to assist in choosing more suitable grasps. In table 5.1, Ours-A completes the task with more lifting attempts and fewer swing attempts than Ours-G. This demonstrates ASPNet can seek objects of a lower level of the entanglement, making the picking efficiency higher than using FGE.

Figure 5.8 illustrates the grasp poses detected from the same depth image using the FGE (-G) [40] and ASPNet (-A) [3]. Grasp poses marked in green are detected

using ASPNet, which always find the objects located at the top of the clutter and grasps them at the middle. On the other hand, FGE detects grasp poses marked in yellow, which have a high score for avoiding collisions with the gripper but does not consider the entanglement issue, resulting in lower picking efficiency.

5.4 Failure Modes and Discussion

Table 5.3 presents the four failure modes and their corresponding frequencies when using our methods, Ours-G and Our-A.

(A) The robot transports nothing to the goal bin due to **grasp failure**.

(B) The robot transports nothing to the goal bin due to **swing failure**. Swing motion sometimes makes the other objects sprung out of the bin. Additionally, there are cases where the objects slipped from the gripper during high-speed swing motions.

(C) The robot transports nothing to the goal bin due to **regrasping failure**. After the main arm of the robot moves to the initial pose, in cases where the target is not aligned vertically with the workspace, the support arm cannot accurately locate the pose of the target.

(D) The robot transports multiple objects into the goal bin due to **recovery error**. Force monitoring fails to detect the entanglement.

Table 5.3 shows the frequency of each failure case. For long objects, the occurrence of regrasping failure and recovery failure is significantly higher compared to another type. It suggests that achieving robust and successful regrasping solely relying on force feedback without visual feedback is challenging.

5.5 Summary

This chapter presents a novel bin picking system specifically for grasping and separating entangled wire harnesses. Our closed-loop system utilizes dynamic manipulation

Table 5.3: Failure Cases in Ours-G/Ours-A and Their Frequencies

Failure Mode	Frequency	
		
(A) Grasp failure	2/127	1/40
(B) Swing Failure	1/127	2/40
(C) Regrasping failure	4/127	4/40
(D) Recovery Failure	5/127	4/40

with haptic feedback, enabling successful handling of complex entanglement scenarios. Through real-world experiments, we demonstrate the effectiveness of our policy in disentangling various wire harnesses with high success rates. In future work, we will address failure cases by incorporating vision-guided regrasping motion or vision-haptic fusion policies. Additionally, we will enhance the perception module to obtain more precise and interpretable visual representations of the entangled deformable objects. Moreover, we will focus on optimizing the parameters of dynamic motion primitives to ensure both accuracy and safety in bin picking.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

This dissertation has addressed the challenges of bin picking for entangled rigid and deformable objects for manufacturing processes. I have proposed unified and robust bin picking systems that incorporate dexterous manipulation and multi-modal perception, including (1) a topology-based method for generating non-tangle grasp positions, (2) a learned policy by predicting action affordances for flexible picking or separating, (3) an efficient policy using a circle-like trajectory to disentangling wire harnesses, and (4) the deployment of a dual-arm robot using bimanual and dynamic manipulation. I have studied the perception problems of vision and haptic signals to represent the entanglement and the manipulation problems of crafting effective disentangling motions that can be associated with the abstracted visual cues. Through experiments in both simulated and real-world scenarios, I have demonstrated the effectiveness of the proposed method by impressive success rates and reduced execution time. Overall, this dissertation contributes to the automation of assembly processes by providing effective solutions for picking both rigid and deformable tangled-prone objects. Taking the advantages of the analytic approaches and deep learning, the proposed methods leverage state-of-the-art techniques to improve the performance for such practical problems of manufacturing.

6.2 Future Work

Complex-shaped objects and deformable objects still pose challenges in bin picking. In the future, bin picking systems should strive to enhance their performance, adaptability, and ability to address more complex physical phenomena. In addition to linear-shaped objects, there is a need to further study non-linear or non-planar objects and wire harnesses with more complex structures to achieve full automation in the manufacturing industry. To achieve this objective, I will discuss several future work ideas that can expand on my research and address open problems in this field.

Fusion of Multiple Visual and Haptic Modalities. A precise perception recognition module is always vital for robotic bin picking. The robot should have a more comprehensive understanding of the objects and their interactions with the environments as we humans do. By extending the methods in the haptic-guided bin picking system, entanglement patterns can be further analyzed and abstracted in a higher-dimensional action space. It can lead to more accurate and robust motion execution. Investigating novel fusion techniques and developing learning algorithms that leverage multiple vision and haptic modalities will be a valuable research direction.

Tracing State of Cluttered Wire Harnesses. Prior works have abstracted away inferring the full state of the bulked objects from visual input, as tracing every object in dense clutter is challenging due to occlusion caused by adjacent objects or the objects themselves. While these methods have demonstrated effectiveness in bin picking, obtaining more precise visual recognition is always desirable. One idea is to first decrease the degree of entanglement and then trace the poses of wire harnesses, such as when the robot grasps and lifts objects. Additionally, exploring shape restoration techniques for occluded objects with multiple interrupted segments would be intriguing. A shape restoration for occluded objects with multiple interrupted segments would be interesting.

Simulated Training Using Deformable Multi-Material Objects. To further enhance the system's capabilities and generalize to unseen scenarios, simulated training using deformable multi-material objects is a promising idea. Leveraging research in the field of deformable object manipulation, a physics-based simulator can be

developed to handle deformable multi-material objects like wire harnesses. However, simulating the complex physical phenomena and real-world effects of multi-stiffness, multi-density, and cluttered objects poses significant challenges. It is also worth considering the possibility of initially learning bin picking policies in mixed simulated and real-world environments.

More Skillful Manipulation. Manipulation for separating entangled objects can be further developed. Specifically, more skillful bimanual manipulation can be implemented on robots, taking inspiration from how humans use two hands to complete such tasks. One arm can grasp the entangled object while the other arm approaches the tangled objects to remove them from the grasp. However, this presents the challenge of handling dynamic environments where the grasped object may not be static. Manipulation planning can be integrated with the aforementioned perception modules to reduce environmental uncertainties and improve execution robustness. Additionally, for extremely difficult cases, it may not be necessary to fully disentangle objects in a single movement; loosening entanglements in a multi-step approach shows promise.

In summary, future work should focus on advancing the field of bin picking for objects with complex shapes or properties. Let me summarize the main ideas to extend the methods in this dissertation: (1) Decreasing the degrees of entanglement by some motion primitives is useful. (2) Dynamic and bimanual manipulation with multiple steps is effective but it should be able to handle the non-static environments. (3) Developing simulators for such difficult objects are useful. These directions and ideas will contribute to the development of more robust and versatile systems capable of handling complex manipulation tasks in manufacturing and other domains.

References

[1] E. S. Ho and T. Komura, "Character motion synthesis by topology coordinates," *Computer Graphics Forum*, vol. 28, no. 2, pp. 299–308, 2009.

[2] E. S. Ho, T. Komura, S. Ramamoorthy, and S. Vijayakumar, "Controlling humanoid robots in topology coordinates," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 178–182, IEEE, 2010.

[3] X. Zhang, Y. Domae, W. Wan, and K. Harada, "Learning efficient policies for picking entangled wire harnesses: An approach to industrial bin picking," *IEEE Robotics and Automation Letters*, vol. 8, no. 1, pp. 73–80, 2022.

[4] S. Tsuji and A. Nakamura, "Recognition of an object in a stack of industrial parts.," in *Proceedings of the 4th International Joint Conference on Artificial Intelligence (IJCAI'75)*, pp. 811–818, Citeseer, 1975.

[5] W. A. Perkins, "Model-based vision system for scenes containing multiple parts.," in *Proceedings of the 6th International Joint Conference on Artificial Intelligence (IJCAI'77)*, pp. 678–684, Citeseer, 1977.

[6] K. Ikeuchi, B. K. Horn, S. Nagata, T. Callahan, and O. Feingold, "Picking up an object from a pile of objects," tech. rep., MASSACHUSETTS INST OF TECH CAMBRIDGE ARTIFICIAL INTELLIGENCE LAB, 1983.

[7] B. K. Horn and K. Ikeuchi, "Picking parts out of a bin," tech. rep., MASSACHUSETTS INST OF TECH CAMBRIDGE ARTIFICIAL INTELLIGENCE LAB, 1983.

[8] B. K. Horn and K. Ikeuchi, "The mechanical manipulation of randomly oriented parts," *Scientific American*, vol. 251, no. 2, pp. 100–113, 1984.

[9] K. Ikeuchi, H. K. Nishihara, B. K. Horn, P. Sobalvarro, and S. Nagata, "Determining grasp configurations using photometric stereo and the prism binocular stereo system," *The International Journal of Robotics Research*, vol. 5, no. 1, pp. 46–65, 1986.

[10] D. G. Lowe, "Three-dimensional object recognition from single two-dimensional images," *Artificial intelligence*, vol. 31, no. 3, pp. 355–395, 1987.

[11] R. M. Haralick, H. Joo, C.-N. Lee, X. Zhuang, V. G. Vaidya, and M. B. Kim, "Pose estimation from corresponding point data," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 19, no. 6, pp. 1426–1446, 1989.

[12] K. Ikeuchi, "Generating an interpretation tree from a cad model for 3d-object recognition in bin-picking tasks," *International Journal of Computer Vision*, vol. 1, no. 2, pp. 145–165, 1987.

[13] R. C. Bolles and P. Horaud, "3dpo: A three-dimensional part orientation system," in *Three-dimensional machine vision*, pp. 399–450, Springer, 1987.

[14] E. Al-Hujazi and A. Sood, "Range image segmentation with applications to robot bin-picking using vacuum gripper," *IEEE transactions on systems, man, and cybernetics*, vol. 20, no. 6, pp. 1313–1325, 1990.

[15] F. Boughorbel, Y. Zhang, S. Kang, U. Chidambaram, B. Abidi, A. Koschan, and M. Abidi, "Laser ranging and video imaging for bin picking," *Assembly Automation*, 2003.

[16] S. Kristensen, S. Estable, M. Kossow, and R. Brösel, "Bin-picking with a solid state range camera," *Robotics and autonomous systems*, vol. 35, no. 3-4, pp. 143–151, 2001.

[17] K. Rahardja and A. Kosaka, "Vision-based bin-picking: Recognition and localization of multiple complex objects using simple visual cues," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS'96*, vol. 3, pp. 1448–1457, IEEE, 1996.

[18] A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3d scenes," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 21, no. 5, pp. 433–449, 1999.

[19] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE transactions on pattern analysis and machine intelligence*, vol. 24, no. 4, pp. 509–522, 2002.

[20] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[21] A. S. Mian, M. Bennamoun, and R. Owens, "Three-dimensional model-based object recognition and segmentation in cluttered scenes," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 10, pp. 1584–1601, 2006.

[22] O. Ghita and P. F. Whelan, "A bin picking system based on depth from defocus," *Machine Vision and Applications*, vol. 13, no. 4, pp. 234–244, 2003.

[23] A. Zuo, J. Z. Zhang, K. Stanley, and Q. J. Wu, "A hybrid stereo feature matching algorithm for stereo vision-based bin picking," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 18, no. 08, pp. 1407–1422, 2004.

[24] D. C. Dupuis, S. Léonard, M. A. Baumann, E. A. Croft, and J. J. Little, "Two-fingered grasp planning for randomized bin-picking," in *Proc. of the Robotics: Science and Systems 2008 Manipulation Workshop-Intelligence in Human Environments*, 2008.

[25] J. Kirkegaard and T. B. Moeslund, "Bin-picking based on harmonic shape contexts and graph-based matching," in *18th International Conference on Pattern Recognition (ICPR'06)*, vol. 2, pp. 581–584, IEEE, 2006.

[26] L.-P. Ellekilde, J. A. Jørgensen, D. Kraft, N. Kruger, N. Krüger, J. Piater, and H. Petersen, "Applying a learning framework for improving success rates in industrial bin picking," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1637–1643, IEEE, 2012.

[27] S. Fuchs, S. Haddadin, M. Keller, S. Parusel, A. Kolb, and M. Suppa, "Cooperative bin-picking with time-of-flight camera and impedance controlled dlr lightweight robot iii," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4862–4867, IEEE, 2010.

[28] D. Holz, M. Nieuwenhuisen, D. Droeschel, J. Stückler, A. Berner, J. Li, R. Klein, and S. Behnke, "Active recognition and manipulation for mobile robot bin picking," in *Gearing Up and Accelerating Cross-fertilization between Academic and Industrial Robotics Research in Europe:*, pp. 133–153, Springer, 2014.

[29] Y. Kita and Y. Kawai, "Localization of freely curved pipes for bin picking," in *2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA)*, pp. 1–8, IEEE, 2015.

[30] H.-Y. Kuo, H.-R. Su, S.-H. Lai, and C.-C. Wu, "3d object detection and pose estimation from depth image for robotic bin picking," in *2014 IEEE international conference on automation science and engineering (CASE)*, pp. 1264–1269, IEEE, 2014.

[31] J.-K. Oh, S. Lee, and C.-H. Lee, "Stereo vision based automation for a bin-picking solution," *International Journal of Control, Automation and Systems*, vol. 10, no. 2, pp. 362–373, 2012.

[32] B. Drost, M. Ulrich, N. Navab, and S. Ilic, "Model globally, match locally: Efficient and robust 3d object recognition," in *2010 IEEE computer society conference on computer vision and pattern recognition*, pp. 998–1005, IEEE, 2010.

[33] C. Choi, Y. Taguchi, O. Tuzel, M.-Y. Liu, and S. Ramalingam, "Voting-based pose estimation for robotic assembly using a 3d sensor," in *2012 IEEE International Conference on Robotics and Automation*, pp. 1724–1731, IEEE, 2012.

[34] S. Hinterstoisser, S. Holzer, C. Cagniart, S. Ilic, K. Konolige, N. Navab, and V. Lepetit, "Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes," in *2011 international conference on computer vision*, pp. 858–865, IEEE, 2011.

[35] M.-Y. Liu, O. Tuzel, A. Veeraraghavan, Y. Taguchi, T. K. Marks, and R. Chellappa, "Fast object localization and pose estimation in heavy clutter for robotic bin picking," *The International Journal of Robotics Research*, vol. 31, no. 8, pp. 951–973, 2012.

[36] J. J. Rodrigues, J.-S. Kim, M. Furukawa, J. Xavier, P. Aguiar, and T. Kanade, "6d pose estimation of textureless shiny objects using random ferns for bin-picking," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3334–3341, IEEE, 2012.

[37] L.-P. Ellekilde and H. G. Petersen, "Motion planning efficient trajectories for industrial bin-picking," *The International Journal of Robotics Research*, vol. 32, no. 9-10, pp. 991–1004, 2013.

[38] D. Buchholz, M. Futterlieb, S. Winkelbach, and F. M. Wahl, "Efficient bin-picking and grasp planning based on depth data," in *2013 IEEE International Conference on Robotics and Automation*, pp. 3245–3250, IEEE, 2013.

[39] K. Harada, K. Nagata, T. Tsuji, N. Yamanobe, A. Nakamura, and Y. Kawai, "Probabilistic approach for object bin picking approximated by cylinders," in *2013 IEEE International Conference on Robotics and Automation*, pp. 3742–3747, IEEE, 2013.

[40] Y. Domae, H. Okuda, Y. Taguchi, K. Sumi, and T. Hirai, "Fast graspability evaluation on single depth maps for bin picking with general grippers," in *2014*

IEEE International Conference on Robotics and Automation (ICRA), pp. 1997–2004, IEEE, 2014.

[41] J. Ichnowski, M. Danielczuk, J. Xu, V. Satish, and K. Goldberg, "Gomp: Grasp-optimized motion planning for bin picking," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5270–5277, IEEE, 2020.

[42] A. Doumanoglou, R. Kouskouridas, S. Malassiotis, and T.-K. Kim, "Recovering 6d object pose and predicting next-best-view in the crowd," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3583–3592, 2016.

[43] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes," in *Robotics: Science and Systems (RSS)*, 2017.

[44] K. Harada, W. Wan, T. Tsuji, K. Kikuchi, K. Nagata, and H. Onda, "Initial experiments on learning-based randomized bin-picking allowing finger contact with neighboring objects," in *2016 IEEE International Conference on Automation Science and Engineering (CASE)*, pp. 1196–1202, IEEE, 2016.

[45] J. Mahler and K. Goldberg, "Learning deep policies for robot bin picking by simulating robust grasping sequences," in *Conference on robot learning*, pp. 515–524, PMLR, 2017.

[46] R. Matsumura, K. Harada, Y. Domae, and W. Wan, "Learning based industrial bin-picking trained with approximate physics simulator," in *International Conference on Intelligent Autonomous Systems*, pp. 786–798, Springer, 2018.

[47] H. Liang, X. Ma, S. Li, M. Görner, S. Tang, B. Fang, F. Sun, and J. Zhang, "Pointnetgpd: Detecting grasp configurations from point sets," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 3629–3635, IEEE, 2019.

[48] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 421–436, 2018.

[49] R. Matsumura, Y. Domae, W. Wan, and K. Harada, "Learning based robotic bin-picking for potentially tangled objects," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7990–7997, IEEE, 2019.

[50] J. S. Dyrstad, M. Bakken, E. I. Grøtli, H. Schulerud, and J. R. Mathiassen, "Bin picking of reflective steel parts using a dual-resolution convolutional neural network trained in a simulated environment," in *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 530–537, IEEE, 2018.

[51] J. Mahler, F. T. Pokorny, B. Hou, M. Roderick, M. Laskey, M. Aubry, K. Kohlhoff, T. Kröger, J. Kuffner, and K. Goldberg, "Dex-net 1.0: A cloud-based network of 3d objects for robust grasp planning using a multi-armed bandit model with correlated rewards," in *2016 IEEE international conference on robotics and automation (ICRA)*, pp. 1957–1964, IEEE, 2016.

[52] G. Leão, C. M. Costa, A. Sousa, and G. Veiga, "Detecting and solving tube entanglement in bin picking operations," *Applied Sciences*, vol. 10, no. 7, p. 2264, 2020.

[53] J. Mahler, M. Matl, V. Satish, M. Danielczuk, B. DeRose, S. McKinley, and K. Goldberg, "Learning ambidextrous robot grasping policies," *Science Robotics*, vol. 4, no. 26, 2019.

[54] A. Zeng, K.-T. Yu, S. Song, D. Suo, E. Walker, A. Rodriguez, and J. Xiao, "Multi-view self-supervised deep learning for 6d pose estimation in the amazon picking challenge," in *2017 IEEE international conference on robotics and automation (ICRA)*, pp. 1386–1383, IEEE, 2017.

[55] M. Fujita, Y. Domae, R. Kawanishi, G. A. G. Ricardez, K. Kato, K. Shiratsuchi, R. Haraguchi, R. Araki, H. Fujiyoshi, S. Akizuki, *et al.*, "Bin-picking robot using a multi-gripper switching strategy based on object sparseness," in *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, pp. 1540–1547, IEEE, 2019.

[56] M. Ishige, T. Umedachi, Y. Ijiri, T. Taniguchi, and Y. Kawahara, "Blind bin picking of small screws through in-finger manipulation with compliant robotic fingers," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 9337–9344, IEEE, 2020.

[57] H. Tachikake and W. Watanabe, "A learning-based robotic bin-picking with flexibly customizable grasping conditions," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 9040–9047, IEEE, 2020.

[58] S. Song, A. Zeng, J. Lee, and T. Funkhouser, "Grasping in the wild: Learning 6dof closed-loop grasping from low-cost demonstrations," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4978–4985, 2020.

[59] Z. Tong, Y. H. Ng, C. H. Kim, T. He, and J. Seo, "Dig-grasping via direct quasistatic interaction using asymmetric fingers: An approach to effective bin picking," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3033–3040, 2021.

[60] M. Moosmann, F. Spenrath, J. Rosport, P. Melzer, W. Kraus, R. Bormann, and M. F. Huber, "Transfer learning for machine learning-based detection and separation of entanglements in bin-picking applications," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1123–1130, IEEE, 2022.

[61] X. Zhang, K. Koyama, Y. Domae, W. Wan, and K. Harada, "A topological solution of entanglement for complex-shaped parts in robotic bin-picking," in *2021*

IEEE 17th International Conference on Automation Science and Engineering (CASE), pp. 461–467, IEEE, 2021.

[62] X. Zhang, Y. Domae, W. Wan, and K. Harada, "Learning to dexterously pick or separate tangled-prone objects for industrial bin picking," *IEEE Robotics and Automation Letters*, vol. 8, no. 8, pp. 4919–4926, 2023.

[63] X. Zhang, Y. Domae, W. Wan, and K. Harada, "A closed-loop bin picking system for entangled wire harnesses using bimanual and dynamic manipulation," *arXiv preprint arXiv:2306.14595*, 2023.

[64] M. Danielczuk, J. Mahler, C. Correa, and K. Goldberg, "Linear push policies to increase grasp access for robot bin picking," in *2018 IEEE 14th international conference on automation science and engineering (CASE)*, pp. 1249–1256, IEEE, 2018.

[65] T. He, S. Aslam, Z. Tong, and J. Seo, "Scooping manipulation via motion control with a two-fingered gripper and its application to bin picking," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 6394–6401, 2021.

[66] S. Mathiesen, I. Iturrate, and A. Kramberger, "Vision-less bin-picking for small parts feeding," in *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, pp. 1657–1663, IEEE, 2019.

[67] Y. Inagaki, R. Araki, T. Yamashita, and H. Fujiyoshi, "Detecting layered structures of partially occluded objects for bin picking," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5786–5791, IEEE, 2019.

[68] K. Suzuki, Y. Yokota, Y. Kanazawa, and T. Takebayashi, "Online self-supervised learning for object picking: detecting optimum grasping position using a metric learning approach," in *2020 IEEE/SICE International Symposium on System Integration (SII)*, pp. 205–212, IEEE, 2020.

[69] Z. Dong, S. Liu, T. Zhou, H. Cheng, L. Zeng, X. Yu, and H. Liu, "Ppr-net: point-wise pose regression network for instance segmentation and 6d pose estimation in bin-picking scenarios," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1773–1780, IEEE, 2019.

[70] A. S. Periyasamy, M. Schwarz, and S. Behnke, "Synpick: A dataset for dynamic bin picking scene understanding," in *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*, pp. 488–493, IEEE, 2021.

[71] J. Yang, Y. Gao, D. Li, and S. L. Waslander, "Robi: A multi-view dataset for reflective objects in robotic bin-picking," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 9788–9795, IEEE, 2021.

[72] K. Kleeberger, C. Landgraf, and M. F. Huber, "Large-scale 6d object pose estimation dataset for industrial bin-picking," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2573–2578, IEEE, 2019.

[73] D. Fischinger, M. Vincze, and Y. Jiang, "Learning grasps for unknown objects in cluttered scenes," in *2013 IEEE international conference on robotics and automation*, pp. 609–616, IEEE, 2013.

[74] A. Saxena, J. Driemeyer, and A. Y. Ng, "Robotic grasping of novel objects using vision," *The International Journal of Robotics Research*, vol. 27, no. 2, pp. 157–173, 2008.

[75] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 705–724, 2015.

[76] D. Morrison, P. Corke, and J. Leitner, "Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach," in *Robotics: Science and Systems (RSS)*, 2018.

[77] M. Gualtieri, A. Ten Pas, K. Saenko, and R. Platt, "High precision grasp pose detection in dense clutter," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 598–605, IEEE, 2016.

[78] P. Jiang, J. Oaki, Y. Ishihara, J. Ooga, H. Han, A. Sugahara, S. Tokura, H. Eto, K. Komoda, and A. Ogawa, "Learning suction graspability considering grasp quality and robot reachability for bin-picking," *Frontiers in Neurorobotics*, vol. 16, 2022.

[79] S. Sajjan, M. Moore, M. Pan, G. Nagaraja, J. Lee, A. Zeng, and S. Song, "Clear grasp: 3d shape estimation of transparent objects for manipulation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3634–3642, IEEE, 2020.

[80] T.-T. Le and C.-Y. Lin, "Bin-picking for planar objects based on a deep learning network: a case study of usb packs," *Sensors*, vol. 19, no. 16, p. 3602, 2019.

[81] Z. Tong, T. He, C. H. Kim, Y. H. Ng, Q. Xu, and J. Seo, "Picking thin objects by tilt-and-pivot manipulation and its application to bin picking," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9932–9938, IEEE, 2020.

[82] A. Caporali, K. Galassi, R. Zanella, and G. Palli, "Fastdlo: Fast deformable linear objects instance segmentation," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9075–9082, 2022.

[83] A. Caporali, R. Zanella, D. De Greogrio, and G. Palli, "Ariadne+: Deep learning–based augmented framework for the instance segmentation of wires," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 12, pp. 8607–8617, 2022.

[84] M. Moosmann, M. Kulig, F. Spenrath, M. Mönnig, S. Roggendorf, O. Petrovic, R. Bormann, and M. F. Huber, "Separating entangled workpieces in random bin picking using deep reinforcement learning," *Procedia CIRP*, vol. 104, pp. 881–886, 2021.

[85] E. Moreira, L. F. Rocha, A. M. Pinto, A. P. Moreira, and G. Veiga, "Assessment of robotic picking operations using a 6 axis force/torque sensor," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 768–775, 2016.

[86] P. Hegemann, T. Zechmeister, M. Grotz, K. Hitzler, and T. Asfour, "Learning symbolic failure detection for grasping and mobile manipulation tasks," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4302–4309, IEEE, 2022.

[87] J. Grannen, P. Sundaresan, B. Thananjeyan, J. Ichnowski, A. Balakrishna, V. Viswanath, M. Laskey, J. Gonzalez, and K. Goldberg, "Untangling dense knots by learning task-relevant keypoints," in *Conference on Robot Learning*, pp. 782–800, PMLR, 2021.

[88] V. Viswanath, K. Shivakumar, J. Kerr, B. Thananjeyan, E. Novoseller, J. Ichnowski, A. Escontrela, M. Laskey, J. E. Gonzalez, and K. Goldberg, "Autonomously untangling long cables," in *Robotics: Science and Systems (RSS)*, 2022.

[89] V. Lim, H. Huang, L. Y. Chen, J. Wang, J. Ichnowski, D. Seita, M. Laskey, and K. Goldberg, "Real2sim2real: Self-supervised learning of physical single-step dynamic actions for planar robot casting," in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 8282–8289, IEEE, 2022.

[90] V. Viswanath, J. Grannen, P. Sundaresan, B. Thananjeyan, A. Balakrishna, E. Novoseller, J. Ichnowski, M. Laskey, J. E. Gonzalez, and K. Goldberg, "Disentangling dense multi-cable knots," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3731–3738, IEEE, 2021.

[91] X. Huang, D. Chen, Y. Guo, X. Jiang, and Y. Liu, "Untangling multiple deformable linear objects in unknown quantities with complex backgrounds," *IEEE Transactions on Automation Science and Engineering*, 2023.

[92] P. Ray and M. J. Howard, "Robotic untangling of herbs and salads with parallel grippers," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2624–2629, IEEE, 2020.

[93] K. Takahashi, N. Fukaya, and A. Ummadisingu, "Target-mass grasping of entangled food using pre-grasping & post-grasping," *IEEE Robotics and Automation Letters*, 2021.

[94] W. H. Lui and A. Saxena, "Tangled: Learning to untangle ropes with rgb-d perception," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 837–844, IEEE, 2013.

[95] P. Sundaresan, J. Grannen, B. Thananjeyan, A. Balakrishna, J. Ichnowski, E. Novoseller, M. Hwang, M. Laskey, J. E. Gonzalez, and K. Goldberg, "Untangling dense non-planar knots by learning manipulation features and recovery policies," in *Robotics: Science and Systems (RSS)*, 2021.

[96] J. Grannen, P. Sundaresan, B. Thananjeyan, J. Ichnowski, A. Balakrishna, V. Viswanath, M. Laskey, J. Gonzalez, and K. Goldberg, "Untangling dense knots by learning task-relevant keypoints," in *Conference on Robot Learning*, pp. 782–800, PMLR, 2021.

[97] H. Ha and S. Song, "Flingbot: The unreasonable effectiveness of dynamic manipulation for cloth unfolding," in *Conference on Robot Learning*, pp. 24–33, PMLR, 2022.

[98] M. Moosmann, F. Spenrath, K. Kleeberger, M. U. Khalid, M. Mönnig, J. Rosport, and R. Bormann, "Increasing the robustness of random bin picking by avoiding grasps of entangled workpieces," *Procedia CIRP*, vol. 93, pp. 1212–1217, 2020.

[99] E. S. Ho and T. Komura, "Wrestle alone: Creating tangled motions of multiple avatars from individually captured motions," in *15th Pacific Conference on Computer Graphics and Applications (PG'07)*, pp. 427–430, IEEE, 2007.

[100] G. N. Reeke Jr, "Protein folding: Computational approaches to an exponential-time problem," *Annual review of computer science*, vol. 3, no. 1, pp. 59–84, 1988.

[101] H. Wakamatsu, A. Tsumaya, E. Arai, and S. Hirai, "Planning of one-handed knotting/raveling manipulation of linear objects," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, vol. 2, pp. 1719–1725, IEEE, 2004.

[102] M. Saha and P. Isto, "Manipulation planning for deformable linear objects," *IEEE Transactions on Robotics*, vol. 23, no. 6, pp. 1141–1150, 2007.

[103] V. Ivan, D. Zarubin, M. Toussaint, T. Komura, and S. Vijayakumar, "Topology-based representations for motion planning and generalization in dynamic environments with interactions," *The International Journal of Robotics Research*, vol. 32, no. 9-10, pp. 1151–1163, 2013.

[104] K. Klenin and J. Langowski, "Computation of writhe in modeling of supercoiled dna," *Biopolymers: Original Research on Biomolecules*, vol. 54, no. 5, pp. 307–317, 2000.

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

[106] M. Danielczuk, A. Kurenkov, A. Balakrishna, M. Matl, D. Wang, R. Martín-Martín, A. Garg, S. Savarese, and K. Goldberg, "Mechanical search: Multi-step retrieval of a target object occluded by clutter," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 1614–1621, IEEE, 2019.

[107] A. Zeng, P. Florence, J. Tompson, S. Welker, J. Chien, M. Attarian, T. Armstrong, I. Krasin, D. Duong, V. Sindhwani, *et al.*, "Transporter networks: Rearranging the visual world for robotic manipulation," in *Conference on Robot Learning*, pp. 726–747, PMLR, 2021.

[108] S. Y. Gadre, K. Ehsani, and S. Song, "Act the part: Learning interaction strategies for articulated object part discovery," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15752–15761, 2021.

[109] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[110] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pp. 234–241, Springer, 2015.

[111] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009.

[112] D. Buchholz, D. Kubus, I. Weidauer, A. Scholz, and F. M. Wahl, "Combining visual and inertial features for efficient grasping and bin-picking," in *2014 IEEE international conference on robotics and automation (ICRA)*, pp. 875–882, IEEE, 2014.

[113] Y. Yamakawa, A. Namiki, M. Ishikawa, and M. Shimojo, "Knotting manipulation of a flexible rope by a multifingered hand system based on skill synthesis," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2691–2696, IEEE, 2008.

[114] C. Chi, B. Burchfiel, E. Cousineau, S. Feng, and S. Song, "Iterative residual policy for goal-conditioned dynamic manipulation of deformable objects," in *Robotics: Science and Systems (RSS)*, 2022.

[115] J. Guo, J. Zhang, Y. Gai, D. Wu, and K. Chen, "Visual recognition method for deformable wires in aircrafts assembly based on sequential segmentation

and probabilisitic estimation," in *2022 IEEE 6th Information Technology and Mechatronics Engineering Conference (ITOEC)*, vol. 6, pp. 598–603, IEEE, 2022.

[116] X. Jiang, K.-m. Koo, K. Kikuchi, A. Konno, and M. Uchiyama, "Robotized assembly of a wire harness in a car production line," *Advanced Robotics*, vol. 25, no. 3-4, pp. 473–489, 2011.

[117] H. Zhou, S. Li, Q. Lu, and J. Qian, "A practical solution to deformable linear object manipulation: A case study on cable harness connection," in *2020 5th International Conference on Advanced Robotics and Mechatronics (ICARM)*, pp. 329–333, IEEE, 2020.

[118] Y. She, S. Wang, S. Dong, N. Sunil, A. Rodriguez, and E. Adelson, "Cable manipulation with a tactile-reactive gripper," *The International Journal of Robotics Research*, vol. 40, no. 12-14, pp. 1385–1401, 2021.

[119] Z. Ma and J. Xiao, "Robotic perception-motion synergy for novel rope wrapping tasks," *IEEE Robotics and Automation Letters*, 2023.

[120] Y. Yamakawa, A. Namiki, and M. Ishikawa, "Dynamic manipulation of a cloth by high-speed robot system using high-speed visual feedback," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 8076–8081, 2011.

[121] R. Hoque, D. Seita, A. Balakrishna, A. Ganapathi, A. K. Tanwani, N. Jamali, K. Yamane, S. Iba, and K. Goldberg, "Visuospatial foresight for multi-step, multi-task fabric manipulation," *arXiv preprint arXiv:2003.09044*, 2020.

[122] D. Seita, A. Ganapathi, R. Hoque, M. Hwang, E. Cen, A. K. Tanwani, A. Balakrishna, B. Thananjeyan, J. Ichnowski, N. Jamali, *et al.*, "Deep imitation learning of sequential fabric smoothing from an algorithmic supervisor," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 9651–9658, IEEE, 2020.

[123] L. Y. Chen, H. Huang, E. Novoseller, D. Seita, J. Ichnowski, M. Laskey, R. Cheng, T. Kollar, and K. Goldberg, "Efficiently learning single-arm fling motions to smooth garments," in *Robotics Research*, pp. 36–51, Springer Nature Switzerland, 2023.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisor, Professor Kensuke Harada. I am thankful for his acceptance and patience when I was just a rookie in robotics. I will always appreciate his trust, unwavering support, valuable suggestions throughout my research and my life. He has provided me a free and reassuring environment for research from the moment when I was a research student, Master student and till Ph.D. student. During moments of hitting barriers and disappointment in my projects or life, Professor Harada has consistently provided me with sharp and brilliant guidance, lifting me from the depths of frustration. He has not only taught me how to initiate and complete a cool research project, but has also taught me how to be a researcher with his excellent vision and insights in the field of robotics. Being a student of Professor Harada has definitely been the greatest fortune in my life. And I will continue my robotic research journey with him as my role model.

I would also like to thank Professor Weiwei Wan, whose enthusiasm in robotics research and brilliant ideas has always influenced on me. He has provided valuable insights on the motivation and writing techniques of my research. I am grateful to Mr. Yukiyasu Domae for his valuable guidance and counsel throughout my Master and Ph.D. I would like to also thank the rest of my thesis committee: Professor Kosuke Sato and Professor Youji Iiguni, for their insightful comments when completing my thesis.

I would like to also express my sincere thanks to Professor Yoshinori Hijikata for accepting me as a research student. His care and mentorship have been invaluable for a newcomer to this country. When I expressed my desire to change my research

topic, he provided understanding and valuable advice. I am also grateful to Professor Takuya Kiyokawa and Professor Ixchel G. Ramirez-Alpizar for their kind suggestions on my research.

I am deeply grateful to Toyota Motor Corporation for their financial support in my Ph.D. projects. I would like to also express my gratitude to the Kobayashi Foundation for awarding me the scholarship. Their support has allowed me to fully dedicate myself to my research pursuits.

My sincere appreciation goes to my labmates, particularly Ms. Ruishuang Liu, Mr. Qi Zhang, Mr. Hao Chen, Mr. Hao Chen, and Dr. Zhengtao Hu, for many research discussions, casual conversations about life and gossip, and shared moments of excitement when our papers were accepted, as well as supporting each other during frustrated times. I want to offer special thanks to Ms. Ruishuang Liu for her sharp comments on my research and life and her company from undergraduate to Ph.D. I would like to thank my basketball mates, Mr. Junbo Zhang, Mr. Bowen Yu, Mr. Chenxi Wang, Mr. Yu Tang, and Mr. Prashant Kumar, for the joyful times we spent every Friday afternoon on the basketball court. I would like to thank all the members and staff in Harada Laboratory for their help and support: Ms. Zhenting Wang, Ms. Yuan Gao, Mr. Kazuki Higashi, Mr. Masato Tsuru, Mr. Koshi Makihara, Mr. Hiroki Hanai, Mr. Hiroshi Tanaka, Ms. Mahiro Muta, Mr. Kodai Masunaga, Mr. Masashi Yo and Ms. Hitomi Yoshioka. Also, I also sincerely thank the graduated students from our lab who helped me a lot in both research and life: Dr. Tomohiro Motoda, Dr. Cristian Beltran, Dr. Ang Zhang, Dr. Yan Wang, Dr. Jingren Xu, Dr. Tomu Tominaga, Mr. Kyosuke Maeda, Mr. Ryo Matsumura, Mr. Kaidi Nie, Mr. Qiming He, Ms. Hitoe Ochi, Mr. Joshua C. Triyonoputro, Mr. Sho Kobayashi, Mr. Shogo Matsuoka, Mr. Shogo Hayakawa, Mr. Mizuki Takasu, Mr. Kento Nakatsuru, Mr. Syusei Nagato, Mr. Yuuga Nakamura, and Mr. Kazuki Iwao.

Finally, I want to thank my family and my friends for their love! I am deeply indebted to my parents, Ms. Yanhua Li and Mr. Li Zhang, for their unconditional love and support throughout my journey. From my childhood to the completion of my Ph.D., their care, wisdom and courage have made me the luckiest person in the world. I love them forever.

List of Publications

Journal Papers

1. **Xinyi Zhang**, Yukiyasu Domae, Weiwei Wan, Kensuke Harada. A Closed-Loop Bin Picking System for Entangled Wire Harnesses using Bimanual and Dynamic Manipulation. Robotics and Computer-Integrated Manufacturing, 2023 (Submitted, IF: 10.4).

2. **Xinyi Zhang**, Yukiyasu Domae, Weiwei Wan, Kensuke Harada. Learning to Dexterously Pick or Separate Tangled-Prone Objects for Industrial Bin Picking. IEEE Robotics and Automation Letters, vol. 8, no. 8, pp. 4919-4926, 2023 (IF: 5.2).

3. **Xinyi Zhang**, Yukiyasu Domae, Weiwei Wan, Kensuke Harada. Learning Efficient Policies for Picking Entangled Wire Harnesses: An Approach to Industrial Bin Picking. IEEE Robotics and Automation Letters, vol. 8, no. 1, pp. 73-80, 2023 (IF: 5.2. Presented at 2023 IEEE International Conference on Robotics and Automation).

4. Kaidi Nie, Felix von Drigalski, Joshua C. Triyonoptro, Chisato Nakashima, Yoshiya Shibata, Yoshinori Konishi, Yoshihisa Ijiri, Taku Yoshioka, Yukiyasu Domae, Toshio Ueshiba, Ryuichi Takase, **Xinyi Zhang**, Damien Petit, Ixchel G. Ramirez-Alpizar, Weiwei Wan, Kensuke Harada. Team O2AS Approach for Task-board Task of the WRC 2018. Advanced Robotics, vol. 34, no. 7-8, pp. 477-498, 2020 (IF: 2.0).

International Conference Papers (with Peer-Review)

1. **Xinyi Zhang**, Keisuke Koyama, Yukiyasu Domae, Weiwei Wan, Kensuke Harada. A Topological Solution of Entanglement for Complex-shaped Parts in Robotic Bin-picking. Proceedings of IEEE International Conference on Automation, Science and Engineering (CASE), pp. 461-467, 2021.

Local Conference Papers (without Peer-Review)

1. **Xinyi Zhang**, Yukiyasu Domae, Weiwei Wan, Kensuke Harada. Initial Experiments on Picking Entangled Wire Harnesses using Dynamic Manipulation. The Robotics and Mechatronics Conference (ROBOMECH), 2P2-E05, 2023.

2. Mizuki Takasu, **Xinyi Zhang**, Yukiyasu Domae, Weiwei Wan, Kensuke Harada. Bin-Picking for Potential Entangled Object by Linearing Image of the Pile. The conference of the Society of Instrument and Control Engineers System Integration Division (SICE SI), 1A2-A14, 2022.

3. **Xinyi Zhang**, Weiwei Wan, Yukiyasu Domae, Kensuke Harada. Learning Dexterous Bin Picking Policies for Picking and Separating Tangled-Prone Parts. The Conference of the Robotics Society of Japan (RSJ), 2J1-04, 2022.

4. **Xinyi Zhang**, Yukiyasu Domae, Weiwei Wan, Kensuke Harada. Efficiently Picking Tangled-Prone Parts by Learning a Sequential Bin Picking Policy. The conference of the Society of Instrument and Control Engineers System Integration Division (SICE SI), 1D1-02, 2021.

5. **Xinyi Zhang**, Keisuke Koyama, Yukiyasu Domae, Weiwei Wan, Kensuke Harada. Topology-based Grasp Detection Avoiding Entanglement for Robotic Bin-picking, The conference of the Society of Instrument and Control Engineers System Integration Division (SICE SI). 3D3-17, 2020.

6. **Xinyi Zhang**, Keisuke Koyama, Weiwei Wan, Yukiyasu Domae, Kensuke Harada. Motion Generation for Separating Tangled Objects in Robotic Bin-picking. The Conference of the Institute of Systems, Control and Information Engineers (SCI), OS07-4, 2020.

7. **Xinyi Zhang**, Damien Petit, Yukiyasu Domae, Ixchel G. Ramirez-Alpizar, Weiwei Wan, Kensuke Harada. Error Analysis and Adjustment on Randomized Bin-picking. The conference of the Society of Instrument and Control Engineers System Integration Division (SICE SI), 3E2-02, 2019.

8. **Xinyi Zhang**, Damien Petit, Yukiyasu Domae, Ixchel G. Ramirez-Alpizar, Weiwei Wan, Kensuke Harada. A Real-time Robotic Calibration Method for Vision-based Bin-picking. The Robotics and Mechatronics Conference (ROBOMECH), 1P1-C10, 2019.

Patents

1. 原田研介, **張馨芸 (Xinyi Zhang)**, 堂前幸康, 万偉偉, 森建郎. ワーク取出し装置. 特開2023-095329, 2023.

2. 原田研介, 万偉偉, 堂前幸康, **張馨芸 (Xinyi Zhang)**, 森建郎, 吹田和嗣, 五十嵐淳. ワーク取り出し装置、ワーク取り出し方法、プログラム及び制御装置. 特開2021-186542, 2021.