

Learning to Dexterously Pick or Separate Tangled-Prone Objects for Industrial Bin Picking

Xinyi Zhang¹, Graduate Student Member, IEEE, Yukiyasu Domae², Member, IEEE, Weiwei Wan¹, Senior Member, IEEE, and Kensuke Harada², Fellow, IEEE

Abstract—Industrial bin picking for tangled-prone objects requires the robot to either pick up untangled objects or perform separation manipulation when the bin contains no isolated objects. The robot must be able to flexibly perform appropriate actions based on the current observation. It is challenging due to high occlusion in the clutter, elusive entanglement phenomena, and the need for skilled manipulation planning. In this letter, we propose an autonomous, effective and general approach for picking up tangled-prone objects for industrial bin picking. First, we learn PickNet - a network that maps the visual observation to pixel-wise possibilities of picking isolated objects or separating tangled objects and infers the corresponding grasp. Then, we propose two effective separation strategies: Dropping the entangled objects into a buffer bin to reduce the degree of entanglement; pulling to separate the entangled objects in the buffer bin planned by PullNet - a network that predicts position and direction for pulling from visual input. To efficiently collect data for training PickNet and PullNet, we embrace the self-supervised learning paradigm using an algorithmic supervisor in a physics simulator. Real-world experiments show that our policy can dexterously pick up tangled-prone objects with success rates of 90%. We further demonstrate the generalization of our policy by picking a set of unseen objects.

Index Terms—Deep learning in grasping and manipulation, grasping.

I. INTRODUCTION

BIN picking is a valuable task in manufacturing to automate the assembly process. It deploys robots to pick necessary objects from disorganized bins, rather than relying on human workers to arrange the objects or using a large number of part

Manuscript received 15 February 2023; accepted 14 June 2023. Date of publication 30 June 2023; date of current version 6 July 2023. This letter was recommended for publication by Associate Editor F. Pecora and Editor M. Vincze upon evaluation of the reviewers' comments. This work was supported by Toyota Motor Corporation. (Corresponding author: Xinyi Zhang.)

Xinyi Zhang and Weiwei Wan are with the Graduate School of Engineering Science, Osaka University, Osaka 560-0043, Japan (e-mail: xinyiz0931@gmail.com; wanweiwei07@gmail.com).

Yukiyasu Domae is with the National Institute of Advanced Industrial Science and Technology, Tokyo 135-0064, Japan (e-mail: domae.yukiyasu@aist.go.jp).

Kensuke Harada is with the Graduate School of Engineering Science, Osaka University, Osaka 560-0043, Japan, and also with the National Institute of Advanced Industrial Science and Technology, Tokyo 135-0064, Japan (e-mail: harada@sys.es.osaka-u.ac.jp).

Our project is available at <https://xinyiz0931.github.io/tangle>.

This letter has supplementary downloadable material available at <https://doi.org/10.1109/LRA.2023.3291271>, provided by the authors.

Digital Object Identifier 10.1109/LRA.2023.3291271

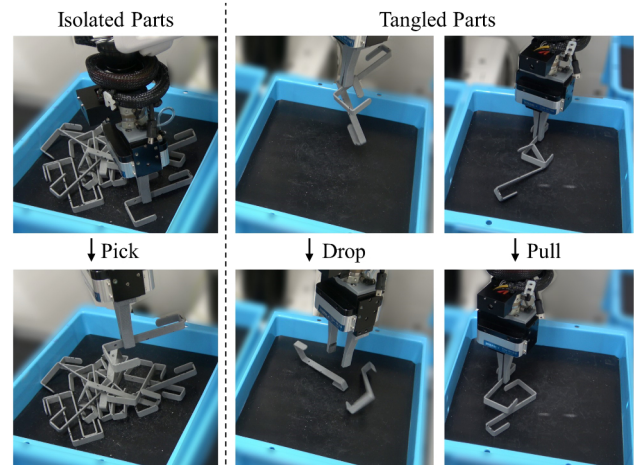


Fig. 1. Our policy learns to flexibly pick or separate tangled-prone objects for bin picking. The robot can search the untangled objects in the bin and pick them up. In cases where all objects are entangled, the robot can separate them by dropping them into another bin. Additionally, the robot can also perform pulling actions to disentangle the objects.

feeders. Existing studies have tackled some challenges in bin picking such as planning grasps under rich contact between the robot's hand and the objects in dense clutter [1], [2], [3], [4] and visual processing heavy occluded scenes [5], [6], [7], [8], [9]. However, objects with complex shapes still remain challenging for bin picking. These objects easily get entangled when randomly placed in a bin, making it difficult for the robot to pick up a single object at a time. It poses challenges in perception since the robot must be able to distinguish the isolated and potentially tangled objects in a cluttered environment. Manipulation is also challenging for planning effective and general separation motions due to the complexity of entanglement estimation and real-world executions.

Prior works have addressed this problem by avoiding grasping potentially entangled objects [10], [11]. However, these approaches use partial visual observation or simple geometrical features such as edges, making it challenging to be adopted in dense clutter. Other studies estimate the pose of object and evaluate the entanglement level for each object [12], [13]. Such a 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. Other studies utilize force and torque sensors to classify if the robot grasps multiple objects [14]. The entanglement detection naturally leads to the necessity for exploring separation strategies. Studies have proposed tilting the gripper to discard the entangled objects [15] or dragging the entangled object out of the clutter [12]. 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 [12] or verifying the entanglement by simulated execution [10]. They do not provide any general criteria for entanglement in cluttered environments.

To address these challenges, we present a novel bin-picking system that leverages self-supervised learning to flexibly and efficiently pick or separate various complex-shaped objects:

- We propose PickNet, which learns to map the visual observations of the unstructured bin to affordance maps that indicate the pixel-wise possibilities of potential actions: **picking** isolated objects or separating 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.
- We propose two efficient separation motion primitives 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 entanglement is rather lower, increasing action efficiency compared with dropping. We propose PullNet to infer the position and direction for pulling from visual observations.
- We train PickNet and PullNet 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.

Fig. 1 shows the proposed actions in our system. The contributions of this work are five-fold. 1) We propose a bin-picking system for tangled-prone objects that enlarges the accuracy, efficiency, flexibility and generalization. 2) We learn PickNet to distinguish untangled or tangled objects in clutter and infer the appropriate actions for them. 3) We propose two novel and efficient motion primitives for separating entangled objects: dropping and pulling. 4) We learn PullNet to infer the pulling actions without object models. 5) We develop an algorithm for simulated self-supervised data collection. We demonstrate the effectiveness of our method using both simulated and real-world experiments with an average success rate of 90%. We also test our method on unseen objects and shows impressive results.

II. RELATED WORK

A. Industrial Bin Picking

Industrial bin picking has been developed for decades. Prior works have primarily focused on model-based approaches such as 3D or 6D pose identification [5], [6], [9] and grasp planning [2], [7]. The model-free approach can directly detect grasps without object models. Domae et al. [1] proposed to plan grasps considering collisions between the gripper and the objects from a single depth image. Several studies leverage deep learning to mitigate the challenges from the complex physical phenomenon [3], [16]. To further improve the robustness of bin picking, Domae et al. [17] proposed a pipeline system for isolation, regrasping and kitting. However, there still remain challenges with objects that tend to get entangled when randomly placed in a bin. Matsumura et al. [10] proposed a learning-based approach to grasp avoiding potentially tangled objects. Zhang et al. [11] generated a feature map to represent the entanglement information from a depth image. These approaches focus on searching and grasping untangled objects and are insufficient for cases where all the objects are entangled in the bin. To address this problem, Leao et al. [15] proposed a method to pick up soft tubes by fitting shape primitives to clutter. Moosmann et al. [12] proposed to estimate the 6D pose of the target and then leverage reinforcement learning to plan separation motion. However, these approaches require prior knowledge of the object shape or model. On the contrary, our system does not require any prior knowledge of objects to flexibly pick isolated objects and separate entangled objects.

B. Object Singulation

Object singulation refers to separating an individual object in cluttered environments. Specific strategies for object singulation are required for achieving different tasks. Firstly, non-prehensile manipulation, such as pushing, is utilized to increase the grasp access when the objects are tightly placed or near the bin walls. Zeng et al. [18] proposed to learn the synergies between pushing and grasping to create enough space for grasping in the clutter. Danielczuk et al. [19] 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 through model-based [12] or model-free approaches [20]. Combining the advantages of these strategies, we propose two separation actions that can be planned without object models: a post-grasping action to drop the entangled objects and an action during grasping to pull out the entangled target.

C. Action Affordance Learning

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

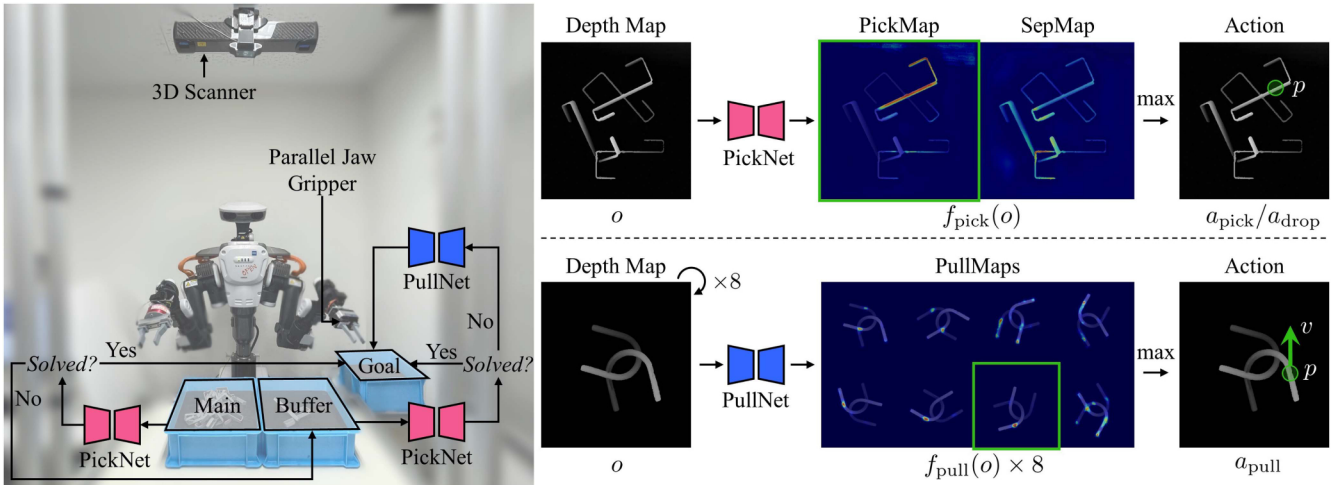


Fig. 2. Overview of our policy. The robot first uses PickNet to search untangled objects in the main bin and transport them to the goal bin. If such objects do not exist, the robot grasps the tangled objects, drops them in a buffer bin and uses PickNet to check if the separation succeeds. The robot then transports the isolated objects to the goal bin or separates the tangled objects by pulling, which is inferred by 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.

where each pixel indicates the possibilities of the grasp success [21]. Other studies also leverage action affordance for various tasks such as perceiving the 3D spatial structure of visual input for pick-and-place task [22], predicting the keypoints associated with visual input for cable untangling [23] or inferring both position and direction for manipulating articulated objects [24]. The direction of the applied action can be encoded by rotating the input image for the inference [18], [24]. In this letter, we employ action affordance learning in industrial bin picking to infer different actions (picking, dropping, pulling) for entangled objects.

III. 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 [1] 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 . 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 $\Phi: a \leftarrow \pi_{\Phi}(o)$.

IV. METHOD

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. 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 separated, we use a neural network PullNet to perform a pulling action and transport the singulated objects to the goal bin. The buffer bin helps to create an environment with few collisions for pulling. This process proceeds in iterations.

A. PickNet: Learning to Pick or Separate

We learn PickNet f_{pick} 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 $o \in \mathbb{R}^{512 \times 512 \times 3}$ with triplicated depth values across three channels, the output is two heatmaps $f_{\text{pick}}(o) \in \mathbb{R}^{512 \times 512 \times 2}$: 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 action in the main bin, we select the heatmap with the highest value 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 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. We use a ResNet-50 backbone [25] with U-Net [26] skip connections for PickNet pre-trained on ImageNet [27]. We use an MSE loss during training.

B. 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 $o \in \mathbb{R}^{512 \times 512 \times 3}$ as input and generates a heatmap called PullMap $f_{\text{pull}}(o) \in \mathbb{R}^{512 \times 512}$ 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 $\pi i/4$, ($i = 0, 1, \dots, 7$) [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. We use a ResNet-18 [25] backbone followed by a bi-linear upsampling layer pre-trained on ImageNet [27]. The network architecture is similar to [23]. We use a binary cross-entropy loss for training. The pulling position is encoded as a 2D Gaussian.

C. 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, we propose an algorithmic supervisor that incorporates a set of entanglement representations, making it easier to control the collection process and adjust the dataset. 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 and 3) determine the sequence for picking demonstrations.

1) *Algorithmic Supervisor*: To distinguish if the object is entangled, we leverage the method for skeletonization and crossing annotation in [28] using the object models and poses. As Fig. 3 shows, 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 other objects, +1 is 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.

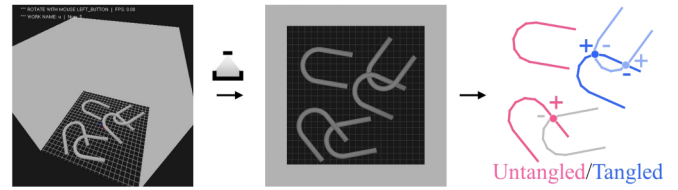


Fig. 3. Process of distinguishing untangled/tangled objects in our algorithm. Given the full state of all objects as input, our algorithm skeletonizes the objects and obtains a graph collection by projecting along the vertical angle to the bin plane. For each object, we annotate the under-crossings it 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.

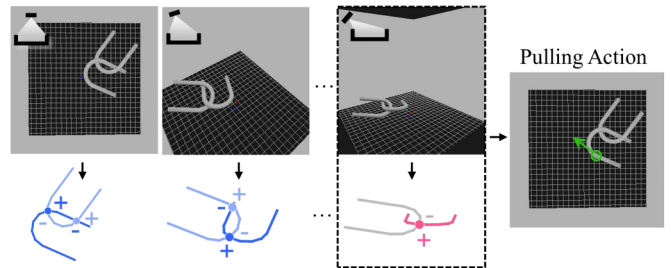


Fig. 4. Process of calculating feasible directions and corresponding objects for pulling in our algorithm. By projecting and labeling the crossing from multiple angles, the feasible pulling direction is determined as the vector along the projection angle where the corresponding graph collection contains untangled (pink) objects.

Our algorithm first randomly drops the objects in the bin and selects untangled objects to grasp. 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 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.

To plan pulling actions, we project objects from multiple angles to find feasible pulling directions (see Fig. 4). If the graph collection of a projection angle contains untangled objects, it is possible to pulling this object out of entanglement 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 vertical projection as the entanglement level 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 [1]. Specifically, we uniformly sample 48 projection angles as candidates in SO(3) space and define that the sampled angles should be in the range from $\pi/4$ [rad] to $\pi/2$ [rad] about the vertical axis to reduce the search cost.

2) *Training Datasets*: 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. The process is shown in Fig. 5. After each attempt, if only one object is lifted without pulling, PickMap is

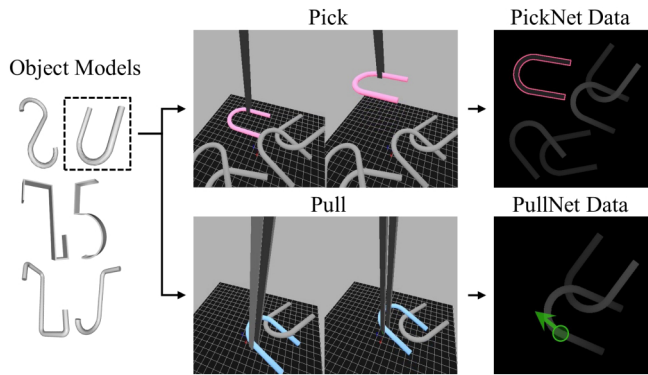


Fig. 5. Demonstrations in simulation and data examples.

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 the pulling action are recorded to train 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. After carrying out data augmentation methods, PickNet dataset contains 85,921 samples and PullNet dataset has 22,208 samples.

V. EXPERIMENTS AND RESULTS

A. Experimental Setup

We use a NEXTAG 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 16 GB 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 $u = (u_x, u_y, u_z)$, we fix $u_z = 0.2$ [cm].

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 [1]. **EMap** takes a depth image as input and produces the entanglement map evaluating where contains entangled objects using edge information [11]. **PickNet** is used in simulation to evaluate the ability to seek untangled objects. **PD** uses PickNet to detect graspable objects in the main bin and buffer bin. The tangled objects are transported to the buffer bin for separation. The robot performs dropping a maximum of three times and transports 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

TABLE I
RESULTS OF SIMULATED EXPERIMENTS

	Seen				Unseen			
	U	S	J	Avg.	U	S	J	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

robot transports one or multiple objects into the goal bin ($\# a_{\text{pick}} + \# a_{\text{pull}}$), “# 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 ($\# a_{\text{pick}} + \# a_{\text{drop}} + \# a_{\text{pull}}$). **Success rate** ($\frac{\# \text{Success attempts}}{\# \text{Goal attempts}}$) evaluates the ability to grasp and transport a single object. **Completion** ($\frac{\# \text{Success attempts}}{\# \text{Objects}}$) 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 the effectiveness of our policy in utilizing picking, pulling and dropping actions to complete the task. **Mean Picks Per Hour (MPPH)** evaluates the computation and execution speed of the system.

B. Simulated Experiments







In the simulation, we conduct a bin-picking task to evaluate the ability to seek untangled objects using FGE, EMap and PickNet. The methods detect the grasp position and feed to our simulator. The simulator locates the corresponding target object and automatically lifts it without gripper, excluding the irregular simulated physics phenomena in grasping or dynamical actions. The bin contains 30 objects and is replenished with the same number of objects after each attempt. We run 50 picking attempts for each object and evaluate the performance using the success rate only.

Table I 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 if 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. For unseen objects, our policy shows improvement in success rates compared to the baselines. The success rates of unseen objects on all methods are lower than seen objects, which could be attributed to the increased difficulty of the unseen set. The overall success rates are not impressive since when the bin contains no isolated objects, there is no separation motion to perform. It demonstrates the necessity of separation strategies to handle unsolvable cases relying solely on PickNet.

C. 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.







TABLE II
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			

1) *Comparison With Baselines:* The results of bin picking in the real world are shown in Table II. 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 that 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 completion results suggest that our methods outperform the baselines, demonstrating their ability to complete the task of 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 the dropping motion. As Table II shows, PD uses dropping action as the only separation strategy and achieves a similar success rate and task completion as our complete policy PDP for both seen and unseen objects. The dropping actions can 1) effectively disentangle the grasped objects and 2) reduce the degree of entanglement, thereby creating space for subsequent disentangling manipulation. Both PD and PDP benefit from this motion. However, the action efficiency of PD is significantly lower than that of PDP. Unlike pulling, dropping acts as an intermediate

TABLE III
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

action and cannot separate and transport the objects to goal bin simultaneously. Table III shows the number of dropping actions in real-world experiments. PD requires more dropping actions than PDP. In cases where the objects are still entangled after the first time of dropping in the buffer bin, PD repeatedly performs this action until the objects are singulated or it reaches three times. We also observe that dropping cannot solve some difficult entanglement cases without well-planned motions or modeling of physical effects. For these reasons, only relying dropping as the separation strategy results in lower action efficiency.

3) *Does Pulling Help?* We can observe that PDP, incorporating both separation strategies of dropping and pulling, achieves the best performance particularly in terms of action efficiency and MPPH. Unlike dropping actions, pulling requires motion planning based on visual observation, making it more interpretable for skillful tasks such as object disentangling. As Table II shows, the success rates are relatively higher when using pulling compared to dropping, as pulling actions can resolve challenging entanglement cases. Pulling also contributes to the action efficiency. PDP follows a hierarchical strategy by first dropping the objects to create a relatively simple entanglement state and then planning pulling actions to further disentangle them. With the reduced degrees of entanglement by dropping in the buffer bin, pulling can efficiently disentangle and transport. Table III shows the successful pulling rates among all actions. Pulling and dropping in our policy PDP can be orchestrated together to 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 II demonstrates that our policies can be generalized to novel objects. Both PD and PDP can recognize isolated objects even if for the challenging non-planar object (the last column in Table II). Thanks to our efficient data collection algorithm, which allows us to collect a large-scale of synthetic data, 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 unseen objects pose additional challenges, such as heavy occlusion of non-planar objects. PickNet may occasionally misidentify isolated objects as entangled objects

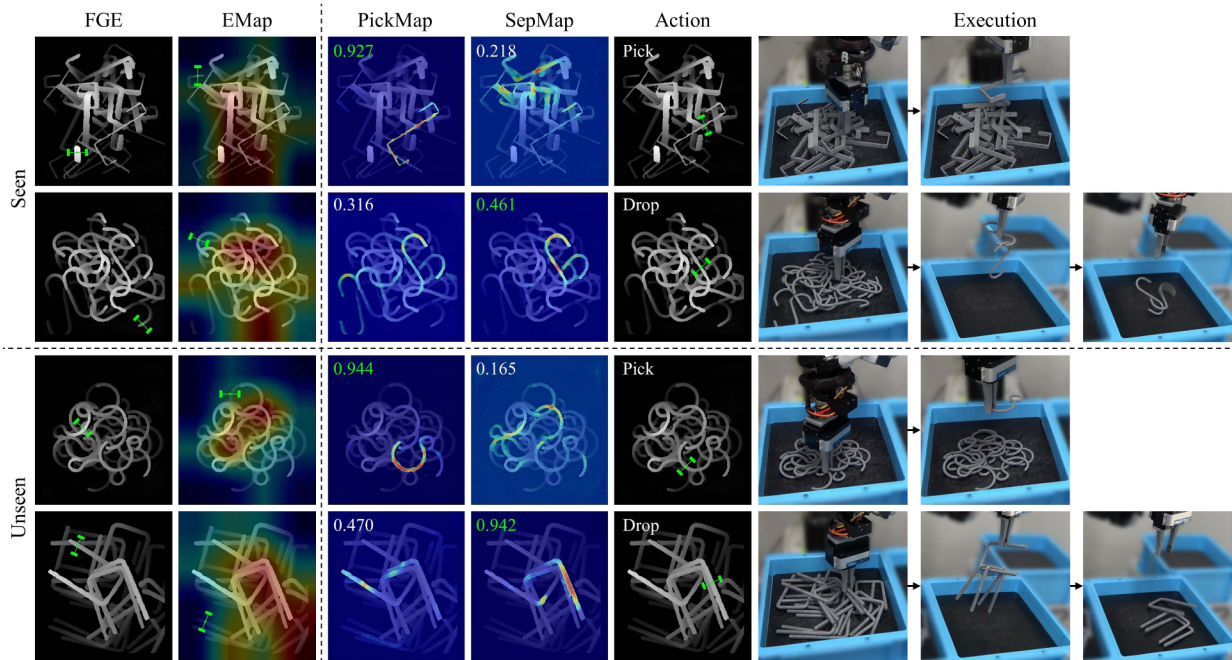


Fig. 6. Qualitative results using PickNet and the corresponding physical executions. Using the same depth map as input, we also present the detected grasps using FGE, the grasps and the entanglement maps using EMap (red regions show high possibilities of containing entangled objects). PickNet outputs PickMap and SepMap with their maximum pixel value as the affordance of picking or dropping.

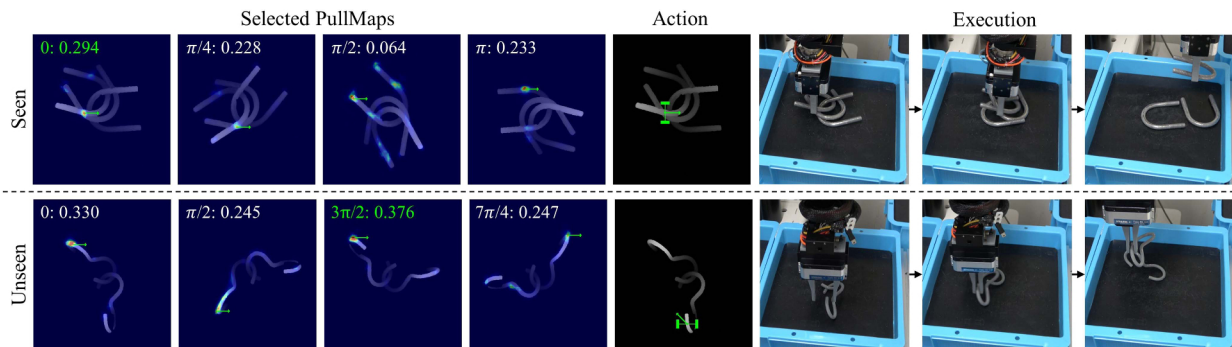


Fig. 7. Qualitative results of PullNet and the corresponding physical executions. PullNet predicts the position and direction for pulling. We rotate the input depth image in eight directions and present four selected PullMaps with their maximum pixel values. The action is selected by the highest score among all PullMaps. The green arrows denote the pulling directions.

and predicts redundant separation actions. Additionally, we present the visualized results using PickNet and PullNet in Fig. 6 in Fig. 7. We also visualize the results from FGE and EMap using the same observation as our policy. These visualizations demonstrate that our policy can accurately extract geometrical information for isolated objects and better represent entanglement compared to the baselines.

D. Failure Modes and Limitations

We observe some failure modes during physical experiments and investigate the limitations of our method. We divide them into three categories as follows.

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 positioned against walls, which left no space for grasping.

2) *Challenging Entanglement Patterns*: The proposed PickNet and PullNet have limitations. First, when the target object forms an endless chain with others, the robot cannot entirely lift and drop the whole chain 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 or bimanual manipulation 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 incorporate objects with various shapes. It will be interesting to collect data only using a minimal amount of objects based on the entanglement representation of their geometries.

VI. CONCLUSION

We propose a bin-picking system for efficiently picking tangled-prone objects. Our hierarchy method is learned using self-supervised simulated data, enabling the robot to perform picking or separation actions dexterously based on the visual observations. 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 extend our policies by leveraging various sensing or more skillful motion primitives for more complex-shaped or deformable objects.

REFERENCES

- [1] 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 *Proc. IEEE Int. Conf. Robot. Automat.*, 2014, pp. 1997–2004.
- [2] D. Buchholz, D. Kubus, I. Weidauer, A. Scholz, and F. M. Wahl, "Combining visual and inertial features for efficient grasping and bin-picking," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2014, pp. 875–882.
- [3] R. Matsumura, K. Harada, Y. Domae, and W. Wan, "Learning based industrial bin-picking trained with approximate physics simulator," in *Proc. 15th Int. Conf. Intell. Auton. Syst.*, 2018, pp. 786–798.
- [4] H. Tachikake and W. Watanabe, "A learning-based robotic bin-picking with flexibly customizable grasping conditions," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 9040–9047.
- [5] 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," *Int. J. Robot. Res.*, vol. 31, no. 8, pp. 951–973, 2012.
- [6] C. Choi, Y. Taguchi, O. Tuzel, M.-Y. Liu, and S. Ramalingam, "Voting-based pose estimation for robotic assembly using a 3D sensor," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2012, pp. 1724–1731.
- [7] K. Harada, K. Nagata, T. Tsuji, N. Yamanobe, A. Nakamura, and Y. Kawai, "Probabilistic approach for object bin picking approximated by cylinders," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2013, pp. 3742–3747.
- [8] K. Harada, W. Wan, T. Tsuji, K. Kikuchi, K. Nagata, and H. Onda, "Experiments on learning-based industrial bin-picking with iterative visual recognition," *Ind. Robot: An Int. J.*, vol. 45, no. 4, pp. 446–457, 2018.
- [9] J. Yang, D. Li, and S. L. Waslander, "Probabilistic multi-view fusion of active stereo depth maps for robotic bin-picking," *IEEE Robot. Automat. Lett.*, vol. 6, no. 3, pp. 4472–4479, Jul. 2021.
- [10] R. Matsumura, Y. Domae, W. Wan, and K. Harada, "Learning based robotic bin-picking for potentially tangled objects," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 7990–7997.
- [11] 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 *Proc. IEEE 17th Int. Conf. Automat. Sci. Eng.*, 2021, pp. 461–467.
- [12] M. Moosmann et al., "Separating entangled workpieces in random bin picking using deep reinforcement learning," *Procedia CIRP*, vol. 104, pp. 881–886, 2021.
- [13] M. Moosmann et al., "Transfer learning for machine learning-based detection and separation of entanglements in bin-picking applications," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 1123–1130.
- [14] 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 Robot. Automat. Lett.*, vol. 1, no. 2, pp. 768–775, Jul. 2016.
- [15] G. Leão, C. M. Costa, A. Sousa, and G. Veiga, "Detecting and solving tube entanglement in bin picking operations," *Appl. Sci.*, vol. 10, no. 7, 2020, Art. no. 2264.
- [16] J. Mahler and K. Goldberg, "Learning deep policies for robot bin picking by simulating robust grasping sequences," in *Proc. Conf. Robot. Learn.*, 2017, pp. 515–524.
- [17] Y. Domae, A. Noda, T. Nagatani, and W. Wan, "Robotic general parts feeder: Bin-picking, regrasping, and kitting," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 5004–5010.
- [18] 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 *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 4238–4245.
- [19] M. Danielczuk et al., "Mechanical search: Multi-step retrieval of a target object occluded by clutter," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2019, pp. 1614–1621.
- [20] X. Zhang, Y. Domae, W. Wan, and K. Harada, "Learning efficient policies for picking entangled wire harnesses: An approach to industrial bin picking," *IEEE Robot. Automat. Lett.*, vol. 8, no. 1, pp. 73–80, Jan. 2023.
- [21] D. Morrison, P. Corke, and J. Leitner, "Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach," in *Proc. Robot.: Sci. Syst.*, 2018.
- [22] A. Zeng et al., "Transporter networks: Rearranging the visual world for robotic manipulation," in *Proc. Conf. Robot Learn.*, 2021, pp. 726–747.
- [23] J. Grannen et al., "Untangling dense knots by learning task-relevant keypoints," in *Proc. Conf. Robot Learn.*, 2021, pp. 782–800.
- [24] S. Y. Gadre, K. Ehsani, and S. Song, "Act the part: Learning interaction strategies for articulated object part discovery," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 15752–15761.
- [25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
- [26] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. 18th Int. Conf., Med. Image Comput. Comput.-Assist. Interv.*, 2015, pp. 234–241.
- [27] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.
- [28] W. H. Lui and A. Saxena, "Tangled: Learning to untangle ropes with RGB-D perception," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 837–844.