Learning Efficient Policies for Picking Entangled Wire Harnesses: A Solution to Industrial Bin Picking

Xinyi Zhang\textsuperscript{1*}, Yukiyasu Domae\textsuperscript{2}, Weiwei Wan\textsuperscript{1} and Kensuke Harada\textsuperscript{1,2}

Abstract—Wire harnesses are essential connecting components in manufacturing industry but are challenging to be automated in industrial tasks such as bin picking. They are long, flexible and tend to get entangled when randomly placed in a bin. This makes the robot struggle to pick a single one from the clutter. Besides, modeling wire harnesses is difficult due to the complex structures of combining deformable cables with rigid components, making it unsuitable for training or collecting data in simulation. In this work, instead of directly lifting wire harnesses, we proposed to grasp and extract the target following circle-like trajectories until it is separated from the clutter. We learn a policy from real-world data to infer the optimal action and grasp from visual observation. Our policy enables the robot to perform non-tangle pickings efficiently by maximizing success rates and reducing the execution time. To evaluate our policy, we present a set of real-world experiments on picking wire harnesses. Results show a significant improvement in success rates from 49.2% to 84.6% over the tangle-agnostic bin picking method. We also evaluate the effectiveness of our policy under different clutter scenarios using unseen types of wire harnesses. The proposed method is expected to provide a practical solution for automating manufacturing processes with wire harnesses.

I. INTRODUCTION

Bin picking is a vital task in manufacturing industries that enables a robot to pick objects randomly placed in a 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. Although robotic bin picking has been researched for decades \cite{1}, \cite{2}, \cite{3}, \cite{4}, \cite{5}, \cite{6}, some objects (e.g., wire harnesses) can still be challenging when automating this process. A wire harness is an indispensable component used in almost every electric drive product. Fig. 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 modeling wire harnesses; (3) The length of a wire harness often exceeds the operation range of a robotic manipulator, making it difficult to extract one from the bin. To successfully perform the pin picking using wire harnesses, a robot must predict the intertwined patterns from visual observations and plan the sophisticated motions for isolation. For this reason, the manufacturing industries still rely on human workers to grasp and separate entangled wire harnesses. Developing an intelligent system to automate this process is highly demanded.

Existing works on industrial bin picking have primarily focused on rigid parts. These tangle-agnostic methods in grasping from the clutter only consider the collisions between objects and the fingertips of the gripper \cite{1}, \cite{4}, \cite{5}, \cite{7}, \cite{8}. For picking simple shaped objects, the robot usually lifts the target object 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 manipulable regions in the robot’s working cell are limited for completely lifting them. Simply adapting the tangle-agnostic bin picking strategies shows unsatisfied performance (see Fig. 1(c)). Previously, some studies have addressed the entanglement problems but for picking curved rigid parts by avoiding the potentially tangled parts \cite{9}, \cite{10}. However, there remain problems for densely cluttered wire harnesses where the bin often contains no isolated objects as Fig. 1(b) shows. Since the complex structures and deformable nature of wire harnesses cause difficulties in obtaining 3D models, it is unsuitable to adopt model-based bin picking policies or train in simulation. Therefore, learning policies for picking wire harnesses requires real-world data. However, collecting a large-scale dataset using a real robot is time-consuming. Annotating ground truth labels for the data is also challenging since we lack the metrics for the entanglement.

In this paper, we (1) design an effective motion to extract wire harnesses from the clutter and (2) learn a policy to solve the entanglement with higher success rates and lower execution time. The key components of our system are:

- We propose a post-grasping action for disentangling the target object from the clutter. 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 from the clutter within the limited robot’s reach range. Fig. 1 shows this process.

- We learn a bin picking policy to infer the optimal grasp and a post-grasping action from a depth image. The learned policy enables the robot to prioritize grasping the untangled objects; avoid grasping at the bad positions (e.g., the ends of the object); reason the extracting distance to reduce the execution time for a successful non-tangle picking. Additionally, we train the network using real-world data by leveraging active learning for a satisfying converge.

Our contributions are three-fold. (1) We develop a unique bin picking system that can disentangle wire harnesses from dense clutters. (2) 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. (3) We learn a policy using real-world data to infer the optimal actions, which further improves the efficiency of bin picking. Real-world bin picking experiments suggest our policy can significantly improve the average success rates and reduce operation time than baselines.

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 [2], [11], [12] and grasp planning [7], [8], [13]. Another approach does not require known object information. They can directly produce grasp poses for novel objects. Domae et al. proposed to plan grasps considering collisions between the gripper and the objects from a single depth image [1]. Several works also leverage deep learning to mitigate the challenges of complex physical interaction and environment uncertainties. Mahler et al. trained a model from synthetic data to produce collision-free grasps for daily objects [14]. Matsumura et al. proposed a learning-based method to plan robust grasps for industrial parts [5]. However, there remain challenges in handling difficult objects. Recently, several works tackled the challenges of those objects which are (1) difficult to be recognized, e.g., transparent or shiny objects [6], [15], (2) difficult to perform grasping, e.g., thin and elliptical objects [16], [17], and (3) involved with rich physical interaction, e.g., tangled-prone objects [9], [10]. So far, these approaches have focused on rigid objects. Leao et al. have proposed a method to pick up soft tubes by fitting the shape primitives to the clutter, but it does not work on dense clutters or objects with irregular shapes [18]. Objects with non-rigid nature and complex geometries like wire harnesses are relatively unexplored and remain challenging in the industrial bin picking domain.

B. Deformable Object Manipulation

Deformable object manipulation has primarily focused on two object classes: 1D (cable, wire, rope) and 2D (fabric, cloth). Several studies adopt specially designed motion primitives to accomplish various manipulation tasks such as knot tying/untying [19], [20], [21], spreading cloth [22] or whipping ropes [23]. 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. proposed to untangle herbs from the clutter using a two-finger gripper [24]. Takahashi et al. proposed a learning-based separation strategy for grasping a specified mass of small food pieces using a specially designed gripper [25]. Although some works have addressed the factory automation problems for wire harnesses [26], [27], [28], robotic wire harnesses picking is less studied. In this paper, we propose a novel and practical bin picking strategy to deal with wire harnesses.

III. 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 robot’s manipulable regions. To extract such objects, the possible positions of the gripper are required to remain in the outer part of the parts bin. In this way, rather than directly lifting, the target can be pulled out of the clutter. Such disentangling motion primitives must be designed to (1) provide enough space for effectively disentangling long objects and (2) handle the unpredictable tangle patterns. To meet these requirements, we propose two motion primitives for effectively disentangling a long and flexible object:

Helix motion: \( \psi_H = (H, \psi_H) \) where \( H \) denotes the helix trajectory represented by \((c_H, r_x, r_y, h_0, h)\) and \( \psi_H \) denotes the execution angle following the trajectory (see Fig. 2(a)). 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 from the clutter along a side angle. Fig. 2(a) shows that this 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 the process, making the workspace clean for the next picking. Let \( c \) denotes the base center of \( H \). Let \( r_x, r_y \) constrain the largest and smallest radius from the center. The helix starts after the gripper lifts the target and reaches \( h_0 \) and stops at \( h \) above the start point. As the execution angle \( \theta_H \) goes, the gripper keeps moving until it reaches the stop point. We experimentally determined the parameters of the trajectory \( H, \psi_H \) is automatically obtained through our learning-based method.

Spinning motion: \( \psi_H = (c_S, \psi_S) \) where \( c_S \) denotes the position of the gripper tip and \( \psi_S \) denotes the one-way rotation of the spinning (see Fig. 2(b)). The gripper spins to handle the occasionally unpredictable entanglement. As Fig. 2(b) shows, sometimes when the rigid components of a wire harness still slightly hang on each other after the helix motion, an extra spinning can help to separate them with less execution time. It also helps to handle the length of a wire harness by extracting it inside a limited working
Module I. Model-Free Grasp Detection: A grasp detection algorithm using a depth image without the object models.

Fig. 2. (a) Trajectory \( H \) of helix motion primitive. Six parameters \((c_x, r_x, r_y, h_0, h, \theta_H)\) are used to define the trajectory. (b) Illustration and parameters \( \theta_S, c_S \) of the spinning motion primitive.

Fig. 3. The extracting action can handle two properties of wire harnesses: tangle-prone and length. (a) The robot separates the entangled wire harnesses from a gentle angle by moving the gripper following a helix trajectory. (b) A spinning motion is performed when the extra objects hang on the connector.

The spinning motion is performed while the gripper is perpendicular to the robot workspace and executes a two-way spinning perpendicularly. Note that \( c_S \) is experimentally determined and \( \theta_S \) is determined in our policy.

IV. LEARNING BIN PICKING POLICIES

The goal of our bin picking policy is to pick up a single wire harness. Instead of separately estimating the entanglement level and then generating the action, our policy can directly infer the required action for the current entanglement situation. If the scene contains isolated objects, the robot can perform a lifting motion directly after grasping. Otherwise, the robot can evaluate the entanglement level and select disentangling action and the grasp pose to extract the target from the clutter. Given a top-down depth image \( o \) as observation, we formulate our bin picking policy \( \pi \) with a learned model parameterized by \( \tau \) using:

\[
a^*, g^* = \pi_\tau(o) \tag{1}
\]

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 shows the three essential modules in our policy:

Module II. Action Success Prediction (ASP): A trained model using real-world data to predict the success possibility of a disentangling action.

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

A. Model-Free Grasp Detection

We select Fast Graspability Evaluation (FGE) [1] - a model-free approach to detect 4-DoF collision-free grasps since obtaining 3D models for wire harnesses is difficult. FGE calculates pixel-wise graspability scores by convoluting a gripper’s template of contact areas and collision areas. A grasp composes a pixel location \( g = (u, v) \) on the depth map and a rotation angle \( \phi \) indicating the gripper’s orientation. We transform \((u, v, \phi)\) to \((g_x, g_y, g_z, g_\phi)\) denoting the grasp point and the gripper’s orientation at the robot coordinate frame. This module outputs the top \( N \) grasps ordered by their FGE scores.

B. Action Success Prediction (ASP)

1) Action Representation: We formulate the disentangling action \( a \) with a motion scheme \( \psi \) and two parameters as follows:

\[
a = (\psi, \theta_H, \theta_S) \mid \psi = \{\psi_H\} \text{ or } \{\psi_H, \psi_S\} \tag{2}
\]

where the robot only performs the helix motion \( \psi_H \) or performs the spinning motion \( \psi_S \) after \( \psi_H \). Note that directly executing \( \psi_S \) after grasping may not be effective since the extracting displacement of the target object is small. Here, we define the action complexity based on two aspects. First, action with two combined motion primitives where \( \theta_S \neq 0 \) are more complex than a single helix motion primitive where \( \theta_S = 0 \). Second, action with larger \( \theta_H \) involves more complexity. Therefore, two parameters \( \theta_H \) and \( \theta_S \) can indicate the action complexity and need to be learned through our policy. Note that since we fix the \( \theta_H \) experimentally, we only need to predict if the \( \theta_S \) is 0 or not. We formulate the complexity of the action \( a \) as \( A(a) \).
Let us consider a case when the robot performs \( a_{t_f} \) to extract an entangled object. 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 while \( a_{t_f} \) is a solvable action but with a higher action complexity. The optimal action can ensure a successful separation and at the same time with less execution time. Note that \( A(a_{\text{solv}}) \geq A(a_{\text{optimal}}) \). Therefore, it is necessary to search for the optimal action rather than solvable action.

2) Discrete Actions for Prediction: We propose six actions using the proposed motion scheme with different \( \theta_H \) and \( \theta_S \) denoted by \{\( \{a_h, a_{hs}, a_f, a_{fs}, a_{t_f}, a_{tfs}\}\)\}. To determine the parameters of each action, we employ the physical experiments to let the robot perform each action for 80 attempts. We hope their action complexity and task accuracy can be approximate as a linear relationship to downgrade the searching cost during learning. Table I shows the properties and success rates while \( a_{dl} \) denote the action of directly lifting. To the end, the helix motion performs three execution degrees \( \theta_H = \pi, 2\pi, 4\pi \). The robot spins by a degree \( \theta_S = \pi/2 \) when the spinning motion is needed. Otherwise, \( \theta_S = 0 \). Note that \{\( \{a_{dl}, a_h, a_{hs}, a_f, a_{fs}, a_{t_f}, a_{tfs}\}\)\} follows an ascending order of action complexity and success rate. Thus, actions with larger complexity lead to higher possibilities of solving the entanglement.

3) Prediction Model: The inference of the optimal action without object models should be conditioned on the location where the robot grasps. We propose Action Success Prediction (ASP) to predict if the action-grasp pair can successfully separate the target, providing a data-driven metric for the entanglement. ASP learns a many-to-one mapping parameterized by \( \tau \):

\[
f_\tau: (\mathbb{R}^{224 \times 224 \times 3}, \mathbb{R}^2, \mathbb{R}^7) \rightarrow \mathbb{R}
\]

where the input is a 3-channel image \( o \) with triplicated depth values across three channels, a pixel-wise grasp pose \( g = (u, v) \), a categorical action \( a \) and the output is a success possibility in the range of \([0, 1]\). We encode the image using a ResNet-50 backbone [29], the grasp point using a single fully-connected layer with 256 units, and the categorical action using a fully-connected layer with 16 units. Then we concatenate the output from all three branches and feed it to a fully-connected layer with 256 units. The output of ASP is the action success possibility \( p = f_\tau(o, g, a) \).

4) Training via Active Learning: The dataset for training ASP is entirely collected by picking from different clutters containing 6, 10, 12 and 18 objects in the real world. Each sample has a depth image \( o \), a grasp \( g \), a labeled action \( a \) and a binary success metric \( S = \{0, 1\} \). However, this dataset has two properties that need to be considered for training: small and redundant in action complexity (some samples are labeled using solvable actions instead of optimal actions). Therefore, we leverage active learning to handle this dataset. 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. Let data pool denote the left samples except the training data. Specifically, we manually select initial training data following the fact that complex clutter patterns may need more complex actions. Note that the numbers of actions are approximately equal. After training, we query the samples in data pool and transfer logical samples to training data. Here, a sample \((o, g, a, S)\) can be determined as logical or illogical using the trained model \( \tau \) with the policy \( \pi \). Let \( a_p = \pi(o, g, a) \) denotes the predicted action:

- Logical: for the positive samples, \( A(a_p) \leq A(a) \) for the negative samples, \( A(a_p) \geq A(a) \).
- Ilogilcal: for the positive samples, \( A(a_p) > A(a) \) for the negative samples, \( A(a_p) < A(a) \).

We define a transfer ratio \( r \) for active learning. Each action query stage transfers \( r \) amount of samples in the current data pool. The iteration stops when data pool is empty or early stops before overfitting.

C. Action-Grasp Inference

At this point, we’ve obtained a set of grasp candidates, action candidates and the scores of each action-grasp pair. The policy then needs to determine which action-grasp pair can be executed. Algorithm II shows the inference of all possible action-grasp pairs to guarantee a successful picking with minimal action complexity. Let \( P = f_\tau(o, G, M) \) denotes a collection of predicted possibilities from the ASP module while the inputs are a depth image \( o \), a collection of actions \( M \) and grasp candidates \( G \) with FGE scores from Model-Free Grasp Detection module. Our algorithm takes

---

**Table I**

PROPERTIES AND EXECUTION DETAILS OF DISCRETE ACTIONS

| \( a_{dl} \) | \( a_h \) | \( a_{hs} \) | \( a_f \) | \( a_{fs} \) | \( a_{t_f} \) | \( a_{tfs} \) |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| \( \psi \) | \( \{\psi_H, \psi_S\} \) | \( \{\psi_H\} \) | \( \{\psi_H\} \) | \( \{\psi_H, \psi_S\} \) | \( \{\psi_H\} \) | \( \{\psi_H, \psi_S\} \) |
| \( \theta_H \) | 0 | \( \pi \) | \( \pi \) | 2\( \pi \) | 2\( \pi \) | \( 4\pi \) | \( 4\pi \) |
| \( \theta_S \) | 0 | \( \pi/2 \) | 0 | \( \pi/2 \) | 0 | \( \pi/2 \) |
| \( d (m) \) | 0.3 | 0.662 | 0.662 | 1.166 | 1.166 | 1.924 | 1.924 |
| \( t (s) \) | 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 |

---

**Algorithm 1:** Action-Grasp Inference

input: image \( o \), grasps \( G \), actions \( M \), ASP model \( f_\tau \)
output: best action \( a^* \), best grasp \( g^* \)

1. \( P \leftarrow f_\tau(o, G, M) \);
2. \( L \leftarrow \{G, M, P\} \);
3. sort \( L \) in the descending order of \( P \);
4. if all \( P \) lower than \( \tau_{thld} \) then
   5. \( g^* \leftarrow \) top element in \( P \), \( a^* \leftarrow a_{tfs} \);
5. else
   6. sort \( L \) in the descending order of \( A \);
   7. \( a^*, g^* \leftarrow \) top element in \( P \);
   8. if exist multiple pairs containing \( a^* \) then
      9. \( g^* \leftarrow \) grasp with maximum FGE score;
10. return \( a^*, g^* \);
as input and produces the optimal action \( a^* \) and grasp \( g^* \). If all possibilities in \( P \) are lower than the threshold \( \rho_{thld} \), which means all action-grasp pairs cannot solve the entanglement, we select the grasp with the highest FGE score and the most complex action \( a_{fs} \). 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.

V. EXPERIMENTS AND RESULTS

We conduct several real-world experiments to answer the following three questions: (1) Does the proposed active learning learn a better action success predictor? (Section V-A) (2) Does our bin picking policy perform more accurately and effectively than baselines? (Section V-B) (3) How does our method qualitatively improve the performance of picking wire harnesses? (Section V-C)

A. ASP Model Performance

Our dataset contains 722 samples and we set the ratio of active learning \( r = 0.4 \). We trained the network using binary cross-entropy loss function and the Adam optimizer. We stopped training after three times of fine-tunings as it achieved the best performance. Fig. [5] 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), which performs the best as the fine-tuning goes since it converges to IM but with more data-driven robustness and accuracy.

Moreover, Table [1] shows the details of each iteration in active learning. Row 1-2 shows the number of samples used in training data and left in data pool. Particularly, 92 samples left in 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 ratio of logical samples in positive and negative data increase with the fine-tuning process. Finally, we also present the average scores predicted by FM for each sub-motion among both positive and negative samples as Table [1] shows. FM can correctly predict an ascending order of possibilities as the action complexity of sub-action increases. We can observe that \( a_{fs}, a_{f}, a_{tf} \) share similar scores due to the validation simples contains 18 objects at most. \( a_{fs} \) does not show a significantly high score due to the accumulated low scores when all predictions fail and \( a_{tf} \) is forced to be selected.

B. Bin Picking Performance

1) Physical experiment Setup: We use the 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 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 [6a]. The length of the wire harness used in this work is 74 cm. After performing the analysis and physical experiments, we fix the parameters of the proposed trajectory as \( c = (0.525, 0.065) \) [m], \( r_x = 0.1 \) m, \( r_y = 0.225 \) m, \( h_0 = 0.32 \) m, \( h = 0.14 \) m, \( \theta_2 = \pi/2 \) as well as the speed of the action since they yielded high task effectiveness. During the physical execution, we sample several points within the same distance from the trajectory and plan motions by the same speed at each interval. In physical experiments, we use a PC with an Intel Core i7-CPU and 16GB memory without GPU. In the learning phase, we use a PC with an Intel Core i5-6400 CPU, 16GB memory and an Nvidia GeForce 1080 GPU.

We evaluate our policy Ours-FM with four baselines. DL (directly lifting) uses FGE to detect the grasp point and executes by directly lifting \( (a_{dl}) \). RAND randomly executes a sub-action and the grasp with the highest FGE score. TFS only executes the most complex action \( a_{fs} \) and the grasp of the highest FGE score. Ours-IM denotes our policy except with the initial model (IM) in active learning.

We leverage two metrics to evaluate the bin picking performance. Success rate refers to the number of successful attempts for picking up a single object divided by the total number of attempts. PPH (Pickings Per Hour) is the number of successful attempts the robot can execute in one hour. Additionally, we present \( \mathcal{A} \) (Average action complexity) to evaluate if our model can recognize the complexities of actions for different entanglement levels.

2) Task Design: We prepare two types of bin picking tasks in the real world. Consecutive picking aims to empty the bin filled with respectively 5, 10, or 15 objects. The robot picks up the object 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 to encourage the robot
TABLE IV 
PERFORMANCE OF BIN PICKING EXPERIMENTS

| Baselines | 5 objects | 10 objects | 15 objects |
|-----------|-----------|------------|------------|
|           | Success Rate | PPH | Avg. A | Success Rate | PPH | Avg. A | Success Rate | PPH | Avg. A |
| Consecutive Picking | DL | 32/50 | 128 | - | 25/50 | 92 | - | 28/50 | 108 | - |
| RAND      | 44/50 | 115 | 2.3 | 46/50 | 117 | 2.5 | 38/50 | 99 | 2.8 |
| TFS       | 48/50 | 133 | 0.8 | 46/50 | 127 | - | 45/50 | 124 | - |
| Ours-IM   | 42/50 | 131 | 0.8 | 38/50 | 117 | 2.3 | 37/50 | 111 | 2.9 |
| Ours-FM   | 44/50 | 156 | 0.8 | 44/50 | 140 | 2.8 | 43/50 | 143 | 2.3 |

| Baselines | 18-20 objects | 20-22 objects | 22-25 objects |
|-----------|---------------|---------------|---------------|
|           | Success Rate | PPH | Avg. A | Success Rate | PPH | Avg. A | Success Rate | PPH | Avg. A |
| Randomized Picking | DL | 14/30 | 93 | - | 12/30 | 80 | - | 7/30 | 47 | - |
| Ours-FM   | 26/30 | 113 | 2.9 | 24/30 | 112 | 3.3 | 22/30 | 103 | 4.3 |

Fig. 6. Physical experiment setup for bin picking.

To confront the entanglement as much as possible. It can ensure the average entanglement level almost remains the same for each attempt. Fig. (b) shows the clutters filled with different numbers of wire harnesses.

3) Comparisons with Baselines: Table IV compares the performance of the proposed method and four baselines in success rate and PPH. For consecutive picking where the goal is to empty the bin, Ours-FM and TFS outperform other approaches with higher success rates under all different clutters. Particularly, Ours-FM significantly increases the average success rate from 56.7% to 87.3% compared to DL. TFS can achieve success rates even higher than Ours-FM but has lower PPH. The reason is that Ours-FM can select the optimal action and grasp instead of only executing the time-consuming \( a_{tf} \). 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. Furthermore, the average action complexities for the predicted actions using RAND, Ours-IM and Ours-FM are also presented in Table IV. The average action complexity for 5 objects is significantly lower than 10 objects and 15 objects. The entanglement frequently occurs in the clutter that contains more objects and requires more complex actions. Our policy can evaluate the entanglement levels of clutters and infer the optimal actions based on the evaluation. We also observe that the failed attempts by the baselines always damage the clutters (e.g., dragging 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. Additionally, Ours-FM achieves similar performance using all of the prepared clutters with 5, 10 and 15 objects while the success rate and PPH of baselines decrease as the number of objects increases. Thus, our policy can handle the wire harnesses picking with robustness and reliability, making it practical for the automated production processes.

For randomized picking, we compare the performance of Ours-FM with a DL baseline as Table IV 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 from such dense clutters. All three proposed modules collaboratively contribute to the efficient bin picking from perception to manipulation planning. Even if the visual observations are unpredictable for our policy, the most complex action \( a_{tf} \) can still strive for success. However, as the number of objects increases, both metrics of Ours-FM decrease due to the highly occluded scenes, thus, the detected grasp candidates become less and the entanglement patterns become unpredictable.

4) Failure Modes: We analyze the failure cases from the physical executions using our policy Ours-FM. First, when the input depth image does not contain a complete object, our model may misunderstand it into multiple graspable short objects. In this case, the robot may directly lift a potentially entangled object. Another apparent failure is when the entanglement modes are sticky (e.g., the connector is tightly wedged into the multi-cable bundle). It is difficult for a robot to disentangle them with a single arm.

C. Qualitative Analysis

1) Novel Wire Harnesses: To demonstrate the breadth of our method, we utilize Ours-FM for the other two types of wire harnesses. They differ from those used for training in lengths and structures but have similar components (e.g.,
deformable cables and rigid connectors). Fig. 7(d) illustrates the optimal action-grasp pairs predicted using our policy. Table V shows the average action complexity of prediction with different object numbers. In the case of shorter objects (see Fig. 8(a)), our model does not predict actions with too large complexities. The robot tends to select $a_{dl}$ and $a_{h}$ to pick up objects. However, when the grasp point is located at the end of the target wire harness, our policy selects $a_{f_{fs}}$ as the optimal action. 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 potentially occurring entanglement. On the other hand, for long objects (Fig. 8(b)) whose length exceeds the capability of our manipulable working space, it is difficult to separate each even if our policy tends to select the complex actions. It suggests that long wire harnesses require more complex manipulation strategies to cope with visual prediction. Despite that, our policy can understand the entanglement levels and predict the corresponding actions for these unseen objects.

2) Visualized Results: We show the visualized results of picking attempts with grasps, actions and input depth images. First, Fig. 7(a) presents 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 grasp point is close to the wire harness’s end, our policy tends to predict more complex actions. Then, Fig. 7(b) presents the predicted actions from the validation samples using Ours-FM compared with the original labels. The notion for each image refers to labeled action $\rightarrow$ predicted action for the specified grasp poses marked as red in the image. For positive samples, our model can predict the optimal action which equals the labeled one or has a lower action complexity. For the negative samples where the labeled actions cannot solve the entanglement, our model can predict actions with higher action complexity. Then, Fig. 7(c) shows a set of successful pickings with the reasoned action-grasp candidates ranked from high score to low score. Our policy predicts the score of each action-grasp pair and infers the optimal pair marked as red. Our policy can recognize the grasp for objects on the top of the heap that only requires $a_{dl}$. As for the clutters that do not contain such objects, our policy can reason the entanglement situation of cluttering and predict the proper actions. When the bin contains heavily entangled objects so that all predicted scores are lower than $p_{thld}$, our policy can predict the pair with the highest FGE score and $a_{f_{fs}}$ to strive for success.

### TABLE V

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

Fig. 7. Qualitative results. (a) Ours-FM predicts action-grasp pairs for each action. (b) Ours-FM predicts the optimal action by depth images and labeled grasps from validation data. The notions above the images denote to labeled action $\rightarrow$ predicted action. (c) Ours-FM predicts the best action and grasp marked using red in real-world bin picking experiments. All action-grasp pairs are presented using the same colors. (d) Ours-FM predicts the best action and grasp for the unseen objects.

Fig. 8. We prepare two novel types of wire harnesses. Both their length and structure are different from the one in training data. Note that the length of the wire harness during training is 74cm. (a) Short wire harnesses. (b) Long wire harnesses.
This work proposes an efficient bin picking system for grasping and separating wire harnesses. We design the motion primitives for extracting the target from the clutter. We learn a policy to reason the extracting distance and produce the optimal action and grasp from a depth image. Real-world experiments suggest that our policy can effectively pick up wire harnesses one at a time with high accuracy. Our work is the first to propose a bin picking system coped with tangled-prone wire harnesses. It shows that automating this process is promisingly applicable to manufacturing industries.

However, our method is not without limitations. First, due to the limited manipulation plans performed by a single manipulator, our policy is not applicable to long wire harnesses and fails to solve the cases where the connectors of a wire harness are wedged with others. In the future, we will consider more skillful picking strategies such as dual-arm manipulation. Second, we also observe failure modes where the robot may mistakenly consider one wire harness into two. The reason might be the real-world data is still not sufficient. In the future, we will consider online closed-loop learning to train and collect data simultaneously to cope with the data efficiency problem. We will extend our policy by using multi-sensory inputs other than only visual observation to deal with unpredictable entanglement situations.

VI. CONCLUSIONS

REFERENCES

[1] Y. Domae, H. Okuda, Y. Taguchi, K. Sumi, and T. Hirai, “Fast grasping 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.

[2] 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.

[3] 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.

[4] K. Harada, W. Wan, T. Tsuji, K. Nakagoe, K. Nagata, and H. Ono, “Initial experiments on learning-based randomized bin-picking allowing finger contacts with neighboring objects,” in 2016 IEEE International Conference on Automation Science and Engineering (CASE), pp. 1196–1202, IEEE, 2016.

[5] 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.

[6] 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.

[7] 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.

[8] D. Buchholz, D. Kubis, J. 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. 857–882, IEEE, 2014.

[9] 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.

[10] 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.