2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Simulation-Assisted Learning for Efficient Bin-Packing of Deformable Packages in a Bimanual Robotic Cell

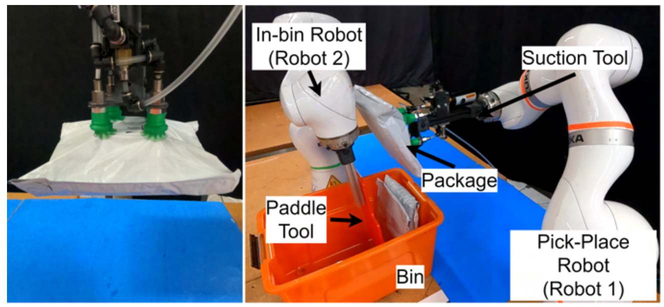
Omey M. Manyar*[1], Hantao Ye*[1], Meghana Sagare[1], Siddharth Mayya[2], Fan Wang[2], and Satyandra K. Gupta[1]

***Abstract*— Bin-packing is an important problem in the robotic warehouse domain. Traditionally, this problem has been studied only for rigid packages (e.g., boxes or rigid objects). In this work, we tackle the problem of bin-packing with deformable packages that have become a popular choice for fulfillment needs. We present a system that incorporates a dual robot arm bimanual setup, uniquely combining suction and sweeping motions to stably and reliably pack deformable packages in a bin. Additionally, we propose a comprehensive action prediction framework to optimize for bin-packing efficiency by predicting optimal actions for both robots involved. Our methodology leverages a two-pronged learning strategy, where initially, we train a model in a self-supervised manner to predict a scoring metric indicative of bin-packing efficiency and then leverage an online optimization scheme to compute optimal actions in real time. The model is pre-trained in simulation in MuJoCo and fine-tuned on small-scale data from a real-world laboratory setting. Our packing score prediction model predicts bin-packing score $\in [0, 1]$ with an MSE of 0.003. Real-world experiments validate our method's adaptability to novel scenarios and its effectiveness in packing operations. Project Website: https://sites.google.com/usc.edu/bimanual-binpacking/**

I. Introduction

Efficient bin-packing is a critical operation in warehouse logistics applications, where incoming packages with unknown characteristics must be swiftly organized into boxes or bins. The optimization of this process is crucial to fulfilling the growing needs of the market. Deformable packages constitute a substantial portion of the packages handled in the supply chain and logistics industry. Fig. 1 describes the type of packages we refer to as deformable packages. Unlike rigid packages that entail conventional packing strategies [1], deformable packages introduce unique challenges. Such packages exhibit bending and deformation that vary depending on their contents. Relying solely on visual cues to discern the objects within them proves infeasible. Moreover, traditional bin-packing methods mainly deal with packing homogeneous rigid contents inside a bin [2]. Thus, while the conventional goals of maximizing packing efficiency remain unchanged, the strategies employed for bin-packing must be refined to deal with the dynamic nature of deformable packages.

Our preliminary investigations into the packing problem revealed that humans often use both hands to pack bins rapidly and effectively. One hand is typically employed to stabilize packages within the bin through sweeping motions. In contrast, the other hand is used to drop packages into the bin rather than meticulously stow them. Drawing inspiration from this human strategy and addressing the aforementioned challenges in bin-packing, our work focuses on developing a bimanual robotic cell (Refer Fig. 1), leveraging the inherent advantages of using both hands concurrently. By employing one manipulator to stabilize and adjust the package and the other to perform placement, we aim to maximize packing efficiency.



(a) Deformable package held with suction cups (b) Bi-manual robotic cell for bin-packing

Fig. 1: (a): The deformable packages studied in this work. (b): The proposed bimanual cell. The in-bin robot stays inside the bin during the packing process, while the pick-place robot transports and places packages inside the bin

Since we have no prior information about the package characteristics, using a heuristic-based approach leads to inefficient performances and possible failures (Refer Section. VII-C). Thus, we aim to learn actions for the two robots that optimize bin-packing efficiency. These actions represent the parameterized goal poses for both robots, i.e., drop the package in the bin at an appropriate location and subsequently perform a sweeping action to achieve compact packing with the other robot. The bin-packing problem we examine in this work is online, i.e., packages are placed in boxes in order of arrival. Although past work has shown the benefits of reinforcement learning for the bin-packing problem, the focus of such methods is on problems where the sequencing of packages is crucial or cases where the objective is to learn the strategy of package placement (e.g., stacking packages, large packages at the bottom, etc.) [3]. Moreover, bridging the sim-to-real gap can be challenging in RL-based approaches [4], [5]. Hence, our proposed parameterized action prediction framework becomes a suitable choice for the given problem.

* Equal Contribution. [1]Realization of Robotic Systems Lab, University of Southern California, Los Angeles, CA, USA. [2] Amazon Robotics, North Reading, MA, USA. Address all correspondence to guptask@usc.edu

In a large warehouse setting, packages can have varying physical characteristics. Thus, relying solely on physical experimentation is impractical. To counter this, we propose a learning architecture pre-trained on simulation data and subsequently trained on small-scale real-world data to predict packing efficiency. While the simulation aims to impart effective inductive biases to facilitate robust real-world execution, our primary focus isn't on creating precise simulations. Instead, we emphasize bridging the sim-to-real gap by integrating an optimizer into our modeling loop. This optimizer computes optimal actions for both robots to achieve high packing scores. Our overall contributions can be summarized as follows:

- A system comprising a bimanual robotic cell that securely and consistently packs objects enclosed in deformable and pliable packages with a combination of suction and sweeping motion. We conduct real-world trials for 18 distinct objects enclosed within a generic deformable package commonly used in the industry
- An action prediction framework that computes optimal actions for the robots by querying a learned neural network model that predicts bin-packing score with an MSE of 0.003 for 100 real-world trials
- A simulation module in MuJoCo that emulates the interaction of deformable packages

II. Related Work

Bimanual object manipulation: Bimanual robotic cells have been extensively studied for handling various objects, including rigid, deformable, articulated, etc [6], [7]. Due to their inherent ability to mimic human manipulation skills, such robotic cells are preferred for complex manipulation tasks. However, they present motion and task planning challenges due to their increased complexity [8]. In [9], the authors study the bimanual manipulation of garments, where the objective is to fold clothes. Bimanual manipulation has also been applied in medical robotics, where intricate dextrous manipulation capabilities are essential [10]. Prior research often employs a primitive-based approach, predicting parameters for specific movement primitives [11], but none have addressed deformable packages.

Bin-Packing: Bin-packing in robotics has been a long-standing problem [12]. Traditionally, bin-packing refers to the problem of maximizing the number of objects packed into a bin [13]. Our focus lies in manipulation planning within this context, specifically addressing the online bin-packing problem of object manipulation and placement [1]. Past research has predominantly examined bin packing for homogeneous, rigid, regular-shaped objects [14]–[16]. However, the challenge of packing deformable objects remains largely unexplored [17], [18]

Deformable object manipulation: Recently, deformable object manipulation has gained significant interest [6]. In [19], the authors propose a system for manipulating deformable packages. However, the focus is on efficiently picking packages from a pile rather than stowing them in a bin. In [5], the authors model object interactions and demonstrate the robotic stowing task with a single robot arm. However, they focus on rigid objects easily perceived by an image or point cloud rather than objects hidden inside packages. Most of the other work mainly focuses on manipulating objects such as sheets [20], garments, elastic cables [7], etc. In [21], [22], authors study the bagging task but focus on manipulating the deformable bag for packing. Our work differs in its emphasis by modeling the impact of robot actions on bin-packing efficiency rather than inter-object interaction.

III. Bimanual Robot Setup for Packaging

Fig. 2 illustrates the deformable package bin-packing task facilitated by the proposed bimanual setup. This process involves picking a package, ensuring its safe transportation, and appropriately positioning it within the bin. The overarching goal is to optimize packing efficiency and process time. A crucial consideration for achieving high efficiency in this task lies in maintaining the stability of packages within the bin; this entails ensuring that in-bin packages remain upright relative to the bin's base and that new packages are safely deposited at their intended locations. Packages need to be upright and snugly packed due to the downstream requirements of the bin that would typically traverse across a large fulfillment center on a mobile platform or conveyor. Other packing strategies can lead to a potential risk of the package being dislodged from the bin during transportation.

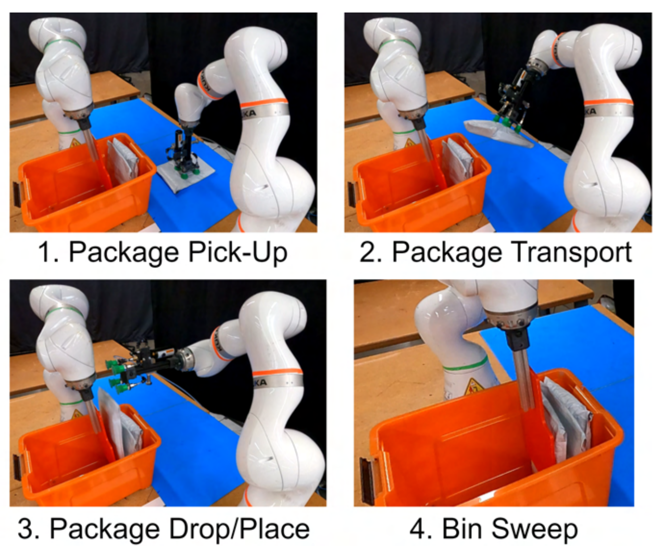


Fig. 2: Overview of the entire pick-and-place pipeline with the bimanual robotic cell.

Our setup (Refer Fig. 2) comprises two KUKA LBR IIWA robots equipped with joint torque sensors capable of operating under impedance control mode. The in-bin robot operates with perpetual compliance, while the pick-place robot exhibits compliance only during picking. The compliance of the pick-place robot is enabled to compute package characteristics online (Refer Section. VI). The in-bin robot is retrofitted with a rigid paddle-shaped tool to manipulate the in-bin packages, while the pick-place robot has a suction-based gripper for safe picking and transporting packages. In the context of bin-packing, both robots perform certain action primitives that enable the task objective. A RealSense D415 camera provides RGB-D image observations of the bin to extract a comprehensive bin-state representation.

IV. Problem Formulation

The objective of the bin packing problem is to optimize packing efficiency. Thus, given an observation O of the bin state S at a given instance, the goal is to compute robot actions a^1 and a^2 for the pick and place robot (Robot 1) and the in-bin robot (Robot 2), respectively. We assume that packages are handled one at a time in the order of arrival, following an online approach for bin packing.

Let $a_i = \langle a_1^i, a_2^i \rangle \in \mathcal{A}$ be the action the bimanual system executes at state S for packing the i^{th} package, where $i \in [1, N]$ and N is the total number of packages. We also define a scoring function $\eta(\cdot, \cdot)$ that computes the packing efficiency of a bin, given the observation of the bin state and characteristics of the package that is queued to be placed in the bin. Thus, the bin packing problem can be represented as solving the following program:

$$\max_{a_i^*} \ \eta(O, p_i) \tag{1}$$

Here, a_i^* is the optimal action that maximizes the packing score $\eta(\cdot, \cdot)$ and p_i are the characteristics of the i^{th} package that is going to be placed in the bin. The actions here are robot end-effector poses for both the robots w.r.t a base frame of reference. Assuming that the bin dimensions and position are available, the actions for the robots can be precisely defined as $a_1^i = (x_1, \beta_1)$, and $a_2^i = (x_2, z_2, \beta_2)$. Here, x_1 & x_2 are the x movement of the corresponding robots, z_2 is the z movement of the robot-2, β_1 & β_2 are y euler angles w.r.t a nominal frame $T_{package} \in \mathbb{SE}(4)$. We fix the other degrees of freedom due to their redundancy in influencing $\eta(\cdot, \cdot)$ (Refer Section. VI-A for details). Additionally, we presume access to a time-optimal trajectory planner with collision-checking capabilities to facilitate the minimization of process time.

V. Approach

A. Packaging Pipeline as a Finite State Machine (FSM)

Our bimanual bin-packing pipeline can be represented as an FSM to facilitate control and task planning (Refer Fig. 3). Upon the arrival of a new package, the system transitions from pick pose detection to successful placement in the bin. The perception module provides the system with image and point cloud observations to compute nominal pick and drop locations. However, relying solely on perception module data can lead to failures due to calibration issues and the deformable nature of the packages. Hence, developing a strategy for learning the optimal actions for both robots involved in the bin-packing process is crucial. We outline the key states governing task completion as follows.

Pick State: The system estimates the pick pose of the package using the RGB-D image from the overhead camera.

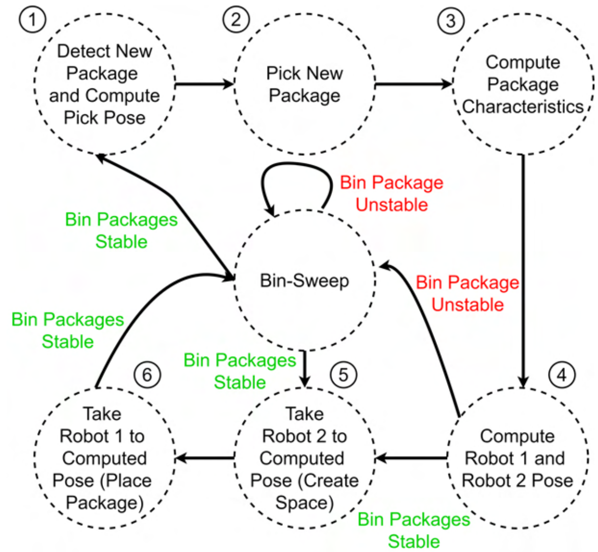


Fig. 3: The pick and place pipeline for bin-packing is represented as a simplified FSM due to its sequential nature. Such representation guides the system's high-level actions

We generate the segmentation mask of the package using YOLOv8 on the RGB image. Subsequently, an oriented bounding box is generated for the observed point cloud. Since packages are placed on the table, only yaw rotation is considered for the pick pose. The trajectory planner computes a feasible trajectory to the pre-pick pose. The pick-place robot moves linearly in the z-direction, interacting with the package under compliance control mode while recording external forces on its end-effector. The stiffness (K) and thickness of the package are determined using the recorded F_z on the robot's flange and the z-deviation (ΔZ) from its commanded goal pose.

Transport State: After picking up the package, the robot receives a drop location from the system. Subsequently, the transport robot follows a time-optimal trajectory to safely deliver the package, considering maximum velocity and acceleration constraints to prevent mid-trajectory drops. Force-torque data is recorded during this motion to determine the package's mass. This enables the characterization of package attributes $P_t = (m, l, b, h, K)$, where m is the mass in kgs, l, b, h are dimensions in meters, and K is stiffness in N/m.

Bin-preparation State: The in-bin robot ensures package stability and upright positioning within the bin. Once the bin is stable, the system provides the P_t and current state observation for the in-bin robot to compute the optimal action a_1. We observe that the height at which the paddle is placed while sweeping plays a crucial role for a successful sweep.

Package Drop State: Once both the robots are at their intended locations, the suction pressure is halted, allowing the package to drop into the available space. Subsequently, the in-bin robot performs a sweep motion under compliance control mode. During sweeping, the paddle aligns with the package while maintaining the z height computed in a_2. Sweeping is repeated until the packages are stable. This

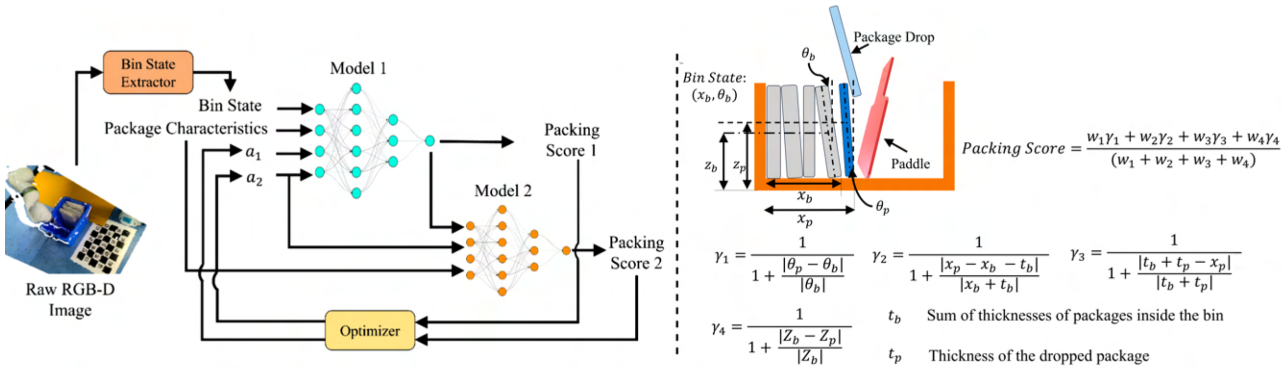


Fig. 4: Left: Our entire model and real-time optimization pipeline to compute optimal actions that maximize the packing score. Packing score prediction functions are modeled with a Multi-Layer Perceptron (MLP). Right: The packing score representation and bin state definition. Definitions remain the same for computing both packing score 1 and packing score 2

process iterates until the bin reaches full capacity.

B. Learning Packing Score Function

Defining a metric to quantify bin-packing quality can be approached in several ways. Our proposed bin-packing action prediction framework maintains flexibility in accommodating various metric definitions. However, for our specific downstream application, we required a metric tailored to our unique setup. We recommend practitioners to redefine the packing score metric to align with the specifications of their individual cells. Given our bimanual setup and the deformable nature of packages, we had to design a packing score that captures the influence of both robots on packing efficiency. Thus, we define packing score as depicted in Fig. 4. The γ_1 and γ_2 components promote bin stability by encouraging package configurations aligned with the last package in the bin. The γ_3 and γ_4 components ensure that the packing fraction of the bin is maximized and the configurations that disrupt the current bin state are penalized accordingly. Moreover, we define packing scores using the same elements for the pick-place robot and the in-bin robot. Throughout training data collection, these scores are computed using the point cloud observations (Refer Section. VI-A), and our objective is to predict these scores for a given action set at a nominal bin state.

One potential approach involves predicting these scores for given actions analytically. However, due to the deformable nature of packages, predicting package dynamics using finite element methods or analytical approaches poses significant challenges [6]. Thus, we aim to learn a model to predict the defined packing score given a new package and the corresponding robot actions for a given bin configuration. This model serves as the function approximator for $\eta(.,.)$. Our empirical investigations revealed that the final packing score of the bin for a package placement is influenced by the sequence of actions corresponding to the drop of the package (robot-1 actions) and the sweeping of the package (robot-2 actions). Suboptimal package placements during the suction robot's package-dropping phase hinder the in-bin robot's ability to achieve a high packing score.

Consequently, the model architecture for predicting the packing score must account for this sequential dependence. As illustrated in Fig. 4, our proposed architecture addresses this challenge by simultaneously optimizing the packing scores of both robots. Specifically, we feed the packing score of the bin after the transport robot's actions η_1 into the network responsible for predicting the packing score after the in-bin robot's actions η_2, ensuring coordinated optimization.

The model comprises two multi-layer perceptron networks, MLP 1 and MLP 2 (Refer Fig. 4), predicting the corresponding packing score. The input to MLP 1 is the robot actions a_1 and a_2, as well as the bin-state defined by the last package's orientation w.r.t bin's vertical face and the distance of the package's base from the bin's origin (Refer Fig. 4). The bin state is computed using point-cloud observation of the bin (Refer Section. VI-A for details). We also tried using PointNet [23] architecture as a state encoder. However, our minimalist state representation gave us satisfactory performance (Refer Section. VII). Moreover, this approach aids in reducing the overall model complexity, thereby improving the parsing time and memory footprint necessary for real-time execution. The MLP 2 takes the output of MLP 1, package characteristics, and the robot-2 actions a_2. This design choice is motivated by the observation that η_2 is conditioned only on η_1 and robot-2 actions a_2. Here, η_1 serves as a surrogate for bin-state post package drop is complete. The model is trained by optimizing the following loss function with L2-regularization:

$$\mathcal{L}_{packing}(w) = \sum_{i=1}^{2} \lambda_i \mathcal{L}^i_{packing} + \lambda_3 ||w||_2^2 \quad (2)$$

Where $\mathcal{L}^i_{packing}$ (Refer Eq. 3) is the Huber loss that motivates reducing mean as well as median error and is suitable for the packing score's data distribution,

$$\mathcal{L}^i_{packing} = \begin{cases} \frac{1}{2}(y^i - \hat{y}^i)^2 & if \left|(y^i - \hat{y}^i)\right| < \delta^i \\ \delta^i * ((y^i - \hat{y}^i) - \frac{1}{2}\delta^i) & \text{otherwise} \end{cases} \quad (3)$$

Here, $\mathcal{L}^1_{packing}$ and $\mathcal{L}^2_{packing}$ are the loss values for both robots' packing scores, $\lambda_1, \lambda_2, \lambda_3$ are regularization hyper-parameters, while δ^1 & δ^2 are thresholds at which change gets triggered between L1 & L2 loss.

C. Learning Optimal Robot Actions

The model described in Section. V-B predicts the corresponding packing scores for the actions executed by robot 1 and robot 2, given the bin and package state. Our objective is to solve the inverse problem, i.e., to compute the actions that can maximize both the packing scores. In order to achieve this, we devise an optimization loop in conjunction with the MLP model (Refer Fig. 4). The objective of this optimizer is to maximize the weighted mean of score 1 and score 2.

For such an optimization scheme to be viable in real-time execution, a crucial design consideration is the convergence time. Furthermore, the loss landscape for our score prediction model (Refer Eq. 3) can be saddled with several local minima, leading to suboptimal action computations. Thus, we opt for the parallel basin-hopping method, a standard global optimization routine widely used for such problems [24]. This method reinstates numerous local optimizers with initial conditions in the vicinity of the current best local minima. The local optimizers use a gradient-based optimization scheme based on the L-BFGS-B algorithm, which supports bounded-constrained optimization. The bounds are placed on the domain of the action space. Thus, in this manner, we overcome the problem of being stranded in suboptimal local minima, and parallel basin-hopping ensures convergence can be achieved in the minimum possible time for it to be suitable for real-time deployment.

VI. Experiments

(a) Sample objects packed inside the package in real data collection

(b) Simulation data generation in MuJoCo

Fig. 5: Simulated and real data generation for model training. We replicate the setup in MuJoCo with deformable packages

A. Real-World Experiments

We collect the data necessary to train the packing score prediction model. 18 different types of objects (6 Rigid, 12 Deformable) are packed inside a LPDE polyethylene-based padded package (Refer Fig. 5a). We aim to collect data for packing scores for a given bin state and a package. Thus, we initially used 12 packages (9 Deformable + 3 Rigid) for training data collection. We uniformly generate a sequence of packages of different thicknesses and execute the bin-packing pipeline described in Section. V-A. During execution, we uniformly sample the values for actions a_1 (x_1 : $U(-0.02, 0.02)$ m, β_1 : $U(0.0, 20.0)$ degrees) and a_2 (x_2 : $U(0.0, 0.03)$ m, z_2 : $U(-0.07, 0.05)$ m β_2 : $U(0.0, 20.0)$ degrees) from the bin state. Additionally, we collect the raw point cloud of the bin to compute the bin state. The cropped point cloud data comprises the container, paddle, and package stacks. The force data collected while package pickup is used to compute the package stiffness and thickness using the impedance control parameters [1]. In this manner, we generate about 1000 bin-packing scenarios and compute corresponding packing scores. The data collection is executed in a self-supervised manner, thus obviating the need for expensive human labeling.

Generating Bin State Data and Ground Truth Packing Scores: To compute the bin state, we collect the raw point cloud of the bin. We then employ a density-based spatial clustering algorithm (DBSCAN) [25] and plane patch detection [26] to extract the position and orientation of the package stack. Once the suction robot drops a new package, to segregate this new package from the previous package stack and container, we perform a difference operation between the old and newly captured point cloud and subsequently perform DBSCAN and plane patch detection for computing pose of the new package. The same process is repeated after the bin-sweep operation is completed. Then, using these point clouds (refer Fig. 6), we compute the packing scores as per definitions in Fig. 4.

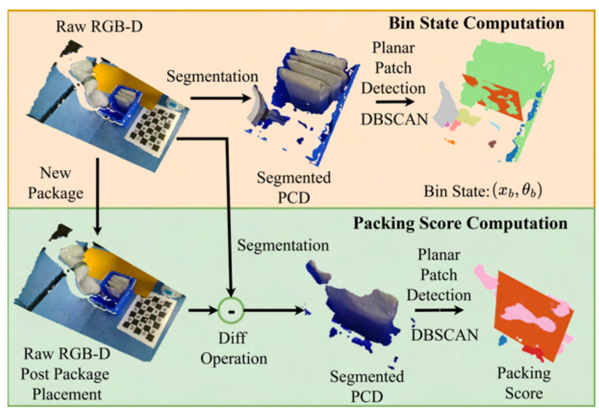


Fig. 6: The raw point cloud and the corresponding processed point clouds used for bin state and packing score computation. Process for computing packing score 1 & 2 is the same

B. Simulation Experiments

In simulation, we replicate the data collection strategy adopted in the previous section. We use MuJoCo 3.0 [27] and model each package as a flex object. We generate instances of packages with varying thicknesses ($U(0.005, 0.07)m$) and stiffnesses ($U(10, 8000)$ N/m) to ensure that the dataset represents the type of objects that can be encountered during real-world execution. In the simulation, the key is to generate

[1]Refer to our project website for implementation details: https://sites.google.com/usc.edu/bimanual-binpacking

Dataset	Size (Real Data)	MSE↓				MAE ↓				Max Error↓			
		Trained on Sim + Real		Trained on Real Only		Trained on Sim + Real		Trained on Real Only		Trained Sim + Real		Trained on Real Only	
		Score 1	Score 2	Score 1	Score 2	Score 1	Score 2	Score 1	Score 2	Score 1	Score 2	Score 1	Score 2
Training	695	**0.0025**	**0.0026**	0.0042	0.005	**0.012**	**0.062**	0.036	0.085	**0.040**	**0.070**	0.102	0.110
Validation	100	**0.0032**	**0.0035**	0.0067	0.0081	**0.063**	**0.062**	0.096	0.098	**0.161**	**0.182**	0.325	0.356
Test	100	**0.0033**	**0.0034**	0.0075	0.0081	**0.065**	**0.076**	0.098	0.101	**0.165**	**0.191**	0.372	0.395

TABLE I: The packing score prediction model's performance, evaluated with 5-fold cross-validation, shows that the model pre-trained on simulation data outperforms others across all metrics, reducing the maximum error by 46-48%

diverse data; thus, we sample different bin states ranging from empty to almost full bins. The paddle-shaped tool and the bin are modeled as rigid objects. We generate 20000 simulation scenarios to pre-train the packing score prediction model (refer Fig. 5a).

VII. Results

A. Failure State Estimation and Packing Score Predictions on Real Data

Failure State Estimation: Prior to training a model to predict a packing score for a given state and action, we train a state estimation model that classifies a given state action pair into two categories : (1) feasible and (2) infeasible (Refer Fig. 7). Infeasible states are those with a packing score below 0.4. Our observations indicate that packing scores below 0.4 typically result in the system's inability to recover to a high packing score after the bin sweep, a trend consistent across both simulation and real-world data. Therefore, we focus on predicting packing scores only for feasible states. We identified 1138 infeasible cases out of 20,000 in simulation data and 105 infeasible cases out of 1000 in real-world data. We label these cases as infeasible. Consequently, we train the state estimation model with simulation and real datasets. The training dataset comprises feasible states (Real: 695 + Sim: 18,862) and infeasible states (Real: 55 + Sim: 1138), with the model trained using binary cross-entropy loss. We evaluated the model's performance on test data consisting of 200 feasible and 50 infeasible real-world cases. The model performs classification with an F1 score of 0.99 (Precision: 0.995 and Recall: 0.987).

Packing Score Prediction: Our packing score prediction model results are illustrated in Table. I. We train the model on a combination of real and simulated data. Initially, we pre-train the model with 18862 simulation data points. This model is then fine-tuned on a batch of 695 real-world data points. Model 1 and Model 2 (Refer Fig. 4) consist of 3 fully connected layers with ReLU activations. We learn the model parameters using an Adam-W optimizer with a fixed learning rate of 0.002 and batch size of 30. To demonstrate the impact of simulation, we present the results by training two model instances: (1) A model pre-trained on simulation data and then fine-tuned on real data and (2) A model only trained on 695 real data points. Table. I depicts that simulation data significantly boosts model performance for all metrics.

Our mean squared error (MSE) values of 0.0033 and 0.0034 for packing scores 1 and 2 suggest a strong fit to the test data. These low MSE values indicate that our model's predictions closely align with the observed packing scores. Cases with error > 0.15 occur infrequently, specifically in only 4 cases out of 100. Notably, these instances typically occur at the fringe, with packing scores between 0.4 and 0.5. All the models are trained on a system configured with intel i7 4.9 GHz and equipped with NVIDIA GeForce RTX 3060.

B. Action Prediction Performance

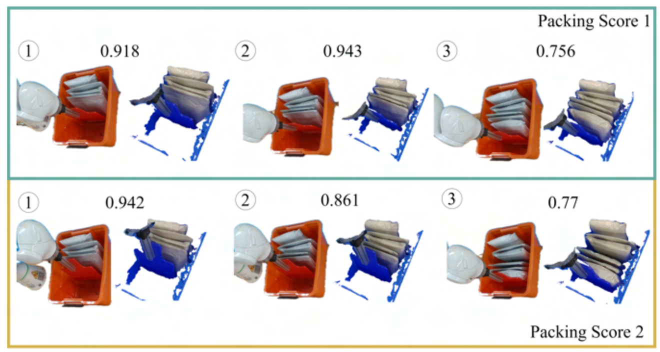


Fig. 7: Intermediate dropping and sweeping performances of the system achieving high-quality bins. We also demonstrate the bin packing instances when scores were lower than 0.8

No. of Trials	MSE ↓		MAE ↓		Max Error ↓		Latency (ms)↓
	Score 1	Score 2	Score 1	Score 2	Score 1	Score 2	
20	0.0019	0.0021	0.033	0.038	0.108	0.107	226

TABLE II: The performance of the action prediction framework in predicting bin packing score during online execution. The mean packing score during these trials was 0.88 for score 1 and 0.91 for score 2

To assess the effectiveness of the optimizer in predicting optimal actions, we conducted real-world trials involving 20 distinct bin states not previously encountered in our data collection. These trials utilized the remaining six packages from our dataset, ensuring a comprehensive evaluation. Subsequently, we executed the actions recommended by the action prediction framework for these novel bin states and recorded the resulting packing scores.

Table II presents a comparative analysis between the observed packing scores and those predicted by our framework. Our findings illustrate the optimizer's ability to effectively predict actions that maximize packing scores. Moreover, the calculated Mean Squared Error (MSE), Mean Absolute Error (MAE), and maximum error values between the predicted and observed packing scores underscore the robustness of our system. The max errors of 0.108 and 0.107 occurred in scenarios where the bin approached full capacity. The average bin packing scores during these trials were 0.88 for score 1 and 0.91 for score 2, reaffirming the efficacy of our system in consistently producing well-packed bins.

Additionally, our framework exhibits computation efficiency, with an average processing time of 226 milliseconds for computing actions for a given state. This time encompasses bin state computation, model parsing, and optimizer convergence, indicating the viability of our model for real-time execution and its applicability in practical scenarios.

C. Bin-packing Performance:

No of Trials	Type of Trial	Avg. Score 1 ↑	Avg. Score 2 ↑	Success Rate ↑
20	ours	**0.88**	**0.91**	**20/20**
	heuristic	0.76	0.82	16/20
	random	0.62	0.66	13/20

TABLE III: Our approach outperforms random and heuristic-based approaches with a high final packing score of 0.91

To demonstrate the need to learn optimal actions for our bimanual setup, we benchmarked our method against two approaches: (1) Random and (2) Heuristic-based. In random trials, we uniformly sample values for actions similar to Section. VI-A and record the bin-packing score and whether the package placement was successful or not. For the heuristic approach, we use package thickness and stiffness as parameters to decide the action values and assume package deformation plays no role in packing. As shown in Table III.

D. Sensitivity Analysis:

Number of Samples	Perturbation in Actions	Mean Change in Packing Score	Max Change in Packing Score
100	10 %	1.5 %	2.6%
	20 %	2.3 %	4.1 %
	30 %	4.6 %	10.2 %
	40 %	7.4 %	17.6 %
	50 %	18.2 %	24.5 %
	>50 %	25.3 %	34.7%

TABLE IV: Effect of adding noise in the actions computed by the optimizer. Here, a 50% change in action values corresponds to 2 cms in position values and 10^o in orientation values, respectively

The optimizer plays a pivotal role in predicting actions that maximize packing scores. Its effectiveness hinges on whether the loss landscape for action vs. packing score at a given state warrants such optimization. Indeed, if a random selection of actions consistently yields high packing scores, the necessity for an optimizer diminishes. To test this hypothesis, we conduct the sensitivity analysis of the optimizer.

In this analysis, we initially compute the optimal actions that maximize the packing score for the bin states in our test data. Subsequently, we systematically perturbed the action space by random increments and observed the corresponding effects on packing scores. Table IV presents our findings, revealing an exponential decrease in packing scores when optimal actions computed by the optimizer were randomly perturbed. Notably, instances where action values were perturbed by 50% resulted in an average packing score decrease of 18.2%. These results underscore the indispensability of optimization in our problem domain. They suggest that arbitrary action selections are unlikely to yield high packing scores consistently, reaffirming the critical role of the optimizer in maximizing packing efficiency.

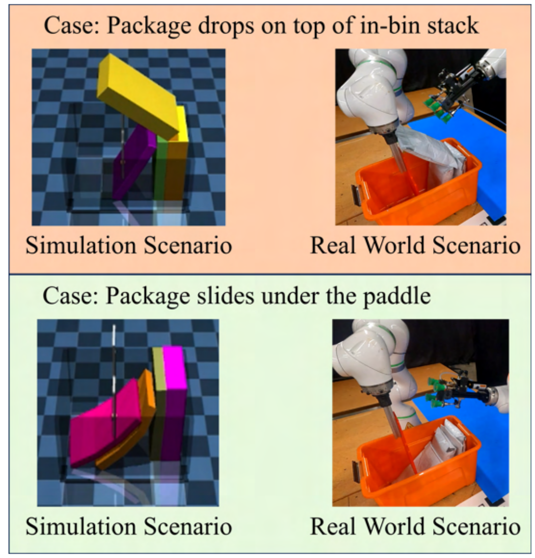


Fig. 8: The comparison between the real world and the corresponding simulation scenario, depicting the effectiveness of simulation to capture the essence of package characteristics

E. Simulation Results:

To demonstrate the fidelity of our simulation, we recreate the scenarios from our real-world dataset and conduct a qualitative and quantitative evaluation of the resulting discrepancies. Fig. 8 illustrates a few such cases for the bin state and action pairs that either lead to failure conditions that are crucial to recognize if we want to rely on the simulation data to provide us a robust enough representation of reality and a high-quality inductive bias for model training. Our simulation effectively replicates failure cases, including scenarios where (1) dropped packages disrupt in-bin arrangements, (2) the dropped package lands on top of other packages by colliding with the paddle tool, and (3) the bin sweeps by the in-bin robot that led to instability in the bin. These cases are the primary reasons for cases with low packing scores, and the ability of our simulation to replicate these scenarios underscores its proficiency in generating high-quality data.

Additionally, we simulate the real-world scenarios in our test dataset. We compute the corresponding values for both the packing scores and compare them with the true values. The low MSE values in the simulated vs. true packing scores error (Refer Table. V) demonstrate that simulation produces data that can provide a good inductive bias for pre-training the model. The max errors were 0.33 for Score 1 and 0.32 for Score 2, observed in cases with lower packing scores (<0.5). However, in these cases, the values of packing scores in the simulation were lower than the actual observed ones, thus making the sim data reflective of failures.

No. of Datapoints	MSE ↓		MAE ↓	
	Score 1	Score 2	Score 1	Score 2
1000	0.014	0.013	0.096	0.09

TABLE V: Performance of Sim vs Real in computing packing scores by recreating scenarios encountered in test data. The errors >0.2 occur in only 0.5% cases. However, all the failure cases are captured robustly with a low error.

VIII. Conclusions

Our study presents a comprehensive framework for addressing the evolving challenges in bin-packing within robotic warehousing, particularly focusing on the integration of deformable packages. By introducing a bimanual robotic cell and a novel action prediction methodology, we have demonstrated our system's ability to optimize bin-packing efficiency. The integration of real-world and simulated data, coupled with a self-supervised learning approach, underscores the robustness and adaptability of our methodology. Furthermore, our simulation pipeline is representative of the system's real-world behavior. Currently, our framework cannot recover from failures and requires human intervention. In the future, we would like to extend our learning framework to a safe learning methodology that can predict a safety metric for an action, thus obviating the need for human intervention.

Acknowledgment

This work is supported in part by Amazon Robotics. The opinions expressed are those of the authors and do not necessarily reflect the opinions of the sponsors.

References

[1] R. Shome, W. N. Tang, C. Song, C. Mitash, H. Kourtev, J. Yu, A. Boularias, and K. E. Bekris, "Towards robust product packing with a minimalistic end-effector," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 9007–9013.

[2] M. Agarwal, S. Biswas, C. Sarkar, S. Paul, and H. S. Paul, "Jampacker: An efficient and reliable robotic bin packing system for cuboid objects," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 319–326, 2021.

[3] Z. Yang, S. Yang, S. Song, W. Zhang, R. Song, J. Cheng, and Y. Li, "Packerbot: Variable-sized product packing with heuristic deep reinforcement learning," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 5002–5008.

[4] A. Kadian, J. Truong, A. Gokaslan, A. Clegg, E. Wijmans, S. Lee, M. Savva, S. Chernova, and D. Batra, "Sim2real predictivity: Does evaluation in simulation predict real-world performance?" *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6670–6677, 2020.

[5] H. Chen, Y. Niu, K. Hong, S. Liu, Y. Wang, Y. Li, and K. R. Driggs-Campbell, "Predicting object interactions with behavior primitives: An application in stowing tasks," in *7th Annual Conference on Robot Learning*, 2023.

[6] B. Shen, Z. Jiang, C. Choy, S. Savarese, L. J. Guibas, A. Anandkumar, and Y. Zhu, "Action-conditional implicit visual dynamics for deformable object manipulation," *The International Journal of Robotics Research*, vol. 43, no. 4, pp. 437–455, 2024.

[7] F. Liu, E. Su, J. Lu, M. Li, and M. C. Yip, "Robotic manipulation of deformable rope-like objects using differentiable compliant position-based dynamics," *IEEE Robotics and Automation Letters*, 2023.

[8] S. Thakar, A. Kabir, P. M. Bhatt, R. K. Malhan, P. Rajendran, B. C. Shah, and S. K. Gupta, "Task assignment and motion planning for bi-manual mobile manipulation," in *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2019, pp. 910–915.

[9] Y. Avigal, L. Berscheid, T. Asfour, T. Kröger, and K. Goldberg, "Speedfolding: Learning efficient bimanual folding of garments," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 1–8.

[10] Z.-Y. Chiu, F. Richter, E. K. Funk, R. K. Orosco, and M. C. Yip, "Bimanual regrasping for suture needles using reinforcement learning for rapid motion planning," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 7737–7743.

[11] R. Chitnis, S. Tulsiani, S. Gupta, and A. Gupta, "Efficient bimanual manipulation using learned task schemas," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 1149–1155.

[12] J. Shi and G. S. Koonjul, "Real-time grasping planning for robotic bin-picking and kitting applications," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 2, pp. 809–819, 2017.

[13] F. Kagerer, M. Beinhofer, S. Stricker, and A. Nüchter, "Bed-bpp: Benchmarking dataset for robotic bin packing problems," *The International Journal of Robotics Research*, vol. 42, no. 11, pp. 1007–1014, 2023.

[14] C.-Y. Weng, W. Yin, Z. J. Lim, and I.-M. Chen, "A framework for robotic bin packing with a dual-arm configuration," in *Advances in Mechanism and Machine Science*, T. Uhl, Ed. Cham: Springer International Publishing, 2019, pp. 2799–2808.

[15] F. Wang and K. Hauser, "Stable bin packing of non-convex 3d objects with a robot manipulator," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 8698–8704.

[16] S. Yang, S. Song, S. Chu, R. Song, J. Cheng, Y. Li, and W. Zhang, "Heuristics integrated deep reinforcement learning for online 3d bin packing," *IEEE Transactions on Automation Science and Engineering*, vol. 21, no. 1, pp. 939–950, 2024.

[17] Q. Zuo, X. Liu, L. Xu, L. Xiao, C. Xu, J. Liu, and W. K. V. Chan, "The three-dimensional bin packing problem for deformable items," in *2022 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, 2022, pp. 0911–0918.

[18] W. Ma, B. Zhang, L. Han, S. Huo, H. Wang, and D. Navarro-Alarcon, "Action planning for packing long linear elastic objects into compact boxes with bimanual robotic manipulation," *IEEE/ASME Transactions on Mechatronics*, 2022.

[19] S. Li, A. Keipour, K. Jamieson, N. Hudson, C. Swan, and K. Bekris, "Demonstrating Large-Scale Package Manipulation via Learned Metrics of Pick Success," in *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023.

[20] O. M. Manyar, J. Desai, N. Deogaonkar, R. J. Joesph, R. Malhan, Z. McNulty, B. Wang, J. Barbič, and S. K. Gupta, "A simulation-based grasp planner for enabling robotic grasping during composite sheet layup," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 930–937.

[21] A. Bahety, S. Jain, H. Ha, N. Hager, B. Burchfiel, E. Cousineau, S. Feng, and S. Song, "Bag all you need: Learning a generalizable bagging strategy for heterogeneous objects," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 960–967.

[22] L. Y. Chen, B. Shi, D. Seita, R. Cheng, T. Kollar, D. Held, and K. Goldberg, "Autobag: Learning to open plastic bags and insert objects," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 3918–3925.

[23] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 77–85.

[24] E. Heiden, D. Millard, E. Coumans, Y. Sheng, and G. S. Sukhatme, "Neuralsim: Augmenting differentiable simulators with neural networks," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 9474–9481.

[25] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. AAAI Press, 1996, p. 226–231.

[26] A. M. Araújo and M. M. Oliveira, "A robust statistics approach for plane detection in unorganized point clouds," *Pattern Recognition*, vol. 100, p. 107115, 2020.

[27] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 5026–5033.