IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.
Cite as T-ASE paper.

RoboPacker: An Autonomous Robotic Packing System for General Objects

Zhenyu Wu, Ziwei Wang, Sichao Huang, Zhan Liu, Xiuwei Xu,
Haibin Yan*, and Jiwen Lu, *Fellow, IEEE*

***Abstract*—In this paper, we propose an autonomous robot packing system named RoboPacker designed to tightly store cluttered general objects into shipping boxes with high space utilization, which is a fundamental process in numerous industrial applications. However, achieving tight packaging for general objects often demands significant labor from human packers, particularly in high-throughput scenes. Compared to existing robot packing approaches, RoboPacker effectively overcomes challenges such as diverse object appearances, severe occlusion, and crowded packing spaces. Specifically, we propose an open-vocabulary shape estimation method to reconstruct complete point clouds for cluttered objects. We also design effective interactions with object clutter to gather informative visual clues for shape estimation under high uncertainty. Additionally, we introduce a hierarchical reinforcement learning framework to optimize packing order, location, and orientation for maximum space utilization. The robotic packing system integrates these techniques with feasible manipulation methods for real-world implementation. In this way, RoboPacker achieves efficient packing of novel and irregular objects, which is more suitable for real deployment environments. The Real-world experiments demonstrate RoboPacker can tightly pack 20 densely cluttered everyday objects from 8 seen and 4 novel classes into the 40×40×20 cm shipping box with a 73.3% success rate. The demonstration video can be found at https://gary3410.github.io/RoboPacker/.**

***Note to Practitioners*—In fields such as logistics warehousing and manufacturing, which have stringent requirements for high productivity, efficiently packing disordered items into limited spaces is a core operational process. However, manually packing objects generates high labor costs due to long hours of high-intensity work. While robots handle some tasks, fully automated general item packaging remains challenging due to variable object shapes, occlusions, and space constraints. The RoboPacker industrial-grade system addresses these issues by optimizing space use and reducing human reliance, which employs advanced computer vision and reinforcement learning to achieve general object packing in complex scenes. The modular design of RoboPacker integrates easily with existing robot arms, making it ideal for e-commerce, food, and pharmaceutical industries seeking intelligent upgrades. Future work will focus on expanding the system to enable more accurate and safer autonomous object packaging by incorporating additional sensors (e.g., tactile).**

***Index Terms*—Autonomous packing systems, open-vocabulary shape estimation, interactive object information collection, packing planning.**

Zhenyu Wu and Haibin Yan are with the School of Intelligent Engineering and Automation, Beijing University of Posts and Telecommunications. Email: {wuzhenyu, eyanhaibin}@bupt.edu.cn. (Corresponding author: Haibin Yan.)

Ziwei Wang is with the School of Electrical and Electronic Engineering, Nanyang Technological University. Email: ziwei.wang@ntu.edu.sg

Sichao Huang, Zhan Liu, Xiuwei Xu and Jiwen Lu are with the Department of Automation, Tsinghua University. E-mail: {huangsc20, liu-z19, xxw21}@mails.tsinghua.edu.cn, lujiwen@tsinghua.edu.cn.

I. Introduction

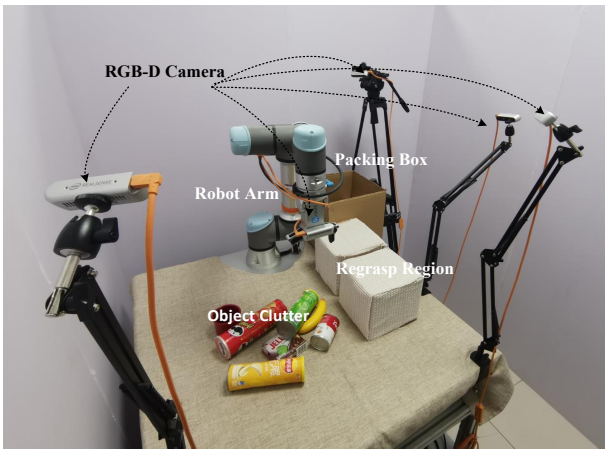
AUTONOMOUS robots have achieved human-level performance in many complex tasks such as driving vehicles [1], [2], playing games [3] and medical treatment [4], and they also reduced the workload of humans in industrial manufacturing and daily life. For example, with the fast growth in E-commerce, automation in warehouse management and cargo transportation has great potential in saving the labor cost with increased throughput and lower accident rate. Since object packaging is crucial in many industrial manufacturing tasks, an autonomous robotic packing system is demanded to achieve high efficiency and low cost [5], [6]. Fig. 1(a) provides the definition of the robotic packing task where cluttered objects are required to be stored in a confined space with high utilization ratio.

Currently, robotic picking and stowing systems [7] only loosely place a set of objects in oversized containers, which is infeasible in industrial and daily tasks because of numerous items and excessive packing box cost. Dense packing for general objects is proposed in [8] for warehouse automation. The overall system consists of the visual perception module that predicts the object geometry, the packing planning module that generates the object arrangement in the packing box, and the object manipulation module to pick and place objects based on plans. Although the robotic packing system in [8] achieves satisfactory performance in lab settings, it still faces several challenges in practical deployment scenarios. First, objects from novel categories that are not seen in the training stage may appear in deployment, and conventional geometry prediction methods fail to accurately estimate the object shape for out-of-distribution instances. Second, objects for packaging are usually piled in dense clutters in realistic scenarios, and directly performing visual recognition algorithms on clutters cannot extract effective information for object geometry prediction due to severe occlusion among objects. Third, searching for the optimal packing plan with the highest bin utilization is an NP-hard problem, and existing search algorithms lead to search inefficiency because of the extremely large search space.

In this paper, we propose an autonomous robotic packing system, RoboPacker, for general object packaging in practical industrial and daily tasks. Since practical robotic systems require strong generalization ability in geometry estimation for unseen categories, informative visual clue collection for densely cluttered objects, and effective search for satisfactory

IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.

(a) Task Definition



(b) Workspace Configurations

Fig. 1. (a) The definition of autonomous robotic packing. (b) The workspace configurations of RoboPacker.

packing plans, we present open-vocabulary shape estimation, interactive object information collection, and hierarchical reinforcement learning-based packing planning. The overall pipeline is demonstrated in Figure 2. For open-vocabulary shape estimation, we segment object instances in multi-view RGB images and assign instance labels to each point in the clutter point cloud to obtain the 3D object partitions. With cropped instances in multi-view RGB images, we extract features from the pre-trained CLIP visual encoder [9], which can be used to retrieve templates with similar geometry from a large template pool. Based on the 3D partial point cloud and the template, we predict the deformation regarding the template, including scaling factors and point-wise offsets, for shape recovery. For interactive object information collection, we compute the visual perception uncertainty consisting of segmentation entropy, multi-view object disagreement, and shape uncertainty in the top-down view. Based on this uncertainty, we generate effective interactions with the clutter to break up its structure for informative visual clue collection. For hierarchical reinforcement learning-based packing planning, we construct a top-level agent to select the next object for packaging and a bottom-level agent to determine the location and orientation of the selected instance. The reward function considers space utilization ratio to save packaging costs, stability to guarantee consistency between the planned and actual placement, perception error tolerance to enhance system robustness, and manipulation feasibility to boost practicability in system implementation. We build the robotic packing system by integrating the presented techniques with practical manipulation techniques including grasping, placement, and rearrangement. Extensive experiments show that our system achieves a space utilization ratio of 0.355 and a 73.3% success rate for 20 densely cluttered general objects from 8 seen and 4 novel classes. Our main contributions can be listed as follows:

1) To the best of our knowledge, we first construct an autonomous robotic packing system for densely cluttered general objects.
2) We present an open-vocabulary shape estimation approach to recover the complete point cloud for objects from both seen and novel categories.
3) We propose an interactive object information collection method to provide effective visual clues for shape estimation of densely cluttered objects.
4) We construct a packing planning framework to generate the sequence, location and orientation for object package in the packing box with high space utilization ratio.

II. Related Work

In this section, we briefly review three related topics including visual perception for densely cluttered objects, packing planning, robot picking and stowing.

A. Visual Perception for Densely Cluttered Objects

Visual perception for densely cluttered objects is very challenging due to the significant occlusion among instances and is also very important to complex robotic manipulation tasks such as grasping [10] and placement [11]. Object correlation mining discovers the possible occluded objects based on the learned geometric or semantic information, where graph neural networks [12] were leveraged to mine the implicit object correlation. Instance template matching computes the correspondence between the partial observation and the predefined object templates, and the definition of correspondence includes the centroid position [13] and difference of local shape descriptors [14]. By receiving the feedback of robot actions, performing physical interaction with the clutter [15]–[17] can efficiently discover the unseen target. Danielczuk *et al.* [18] presented push, suction and grasp as action primitives in the modeled Partially Observable Markov Decision Process, and the actions were taken iteratively until discovering the target object. Novkovic *et al.* [19] diversified the action primitives for interactive exploration, and they also leveraged reinforcement learning algorithms to efficiently search the optimal interaction policy. However, existing densely cluttered objects perception frameworks mainly focus on single-target search, such as retrieving categories or searching for target objects, which cannot satisfy the high generalization ability required for objects with novel appearances that appear in actual deployment scenes.

B. Packing Planning

Packing planning aims to generate the optimal object packing sequence, location and orientation for placement in the packing box. The minimization of height, surface area and volume have been leveraged as the objective for greedy search in bin packing [20]. El *et al.* [21] generalized squirrel search algorithm to search the best heuristics for large-scale instances in a reasonable time. Data-driven methods

IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.

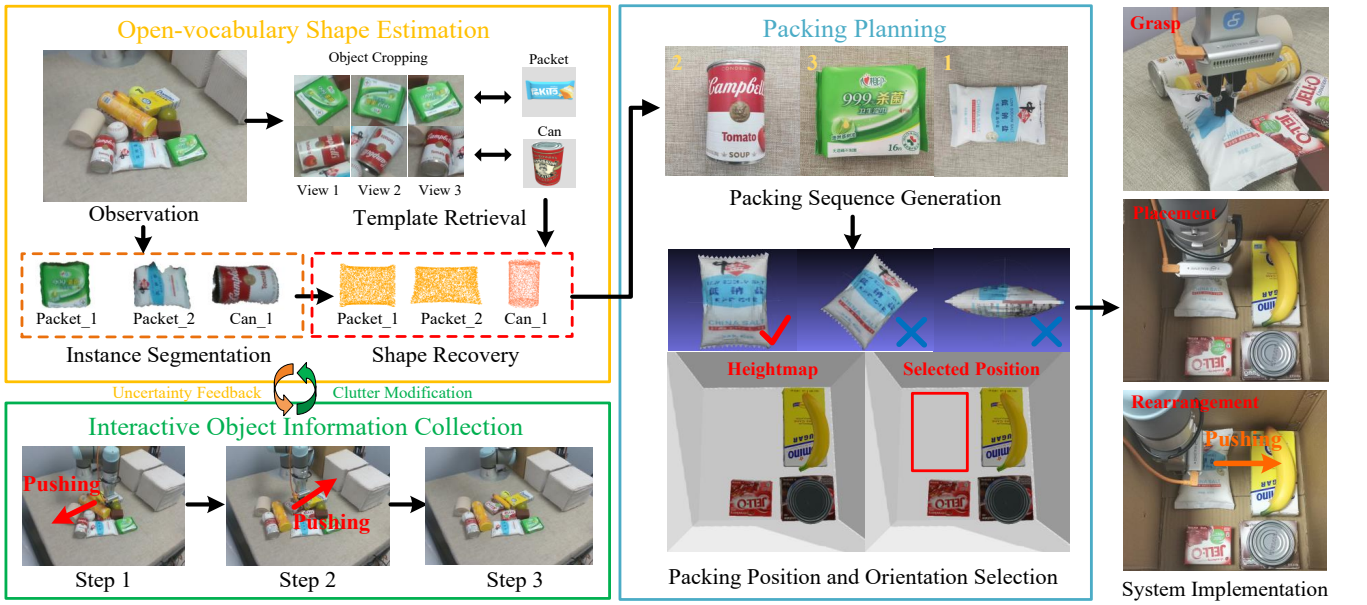


Fig. 2. Compositions of our robotic packing system including open-vocabulary shape estimation, interactive object information collection and hierarchical reinforcement learning based packing planning. Feasible manipulation techniques containing grasping, placement and rearrangement are integrated with the presented approaches to build the complete system.

[22] employ efficient search algorithms to avoid sub-optimal solutions and high computational costs in heuristic approaches. Since irregular objects instead of cuboids are usually packed in realistic applications, generating packing plans for general objects becomes important for many industrial manufacturing and daily life tasks. To yield the optimal packing sequence and location for general objects in realistic packing tasks, the objective of empty maximal space [23], heightmap minimization [24] further strengthens the stability and space utilization ratio by physically simulating the packing procedure. Huang *et al.* [25] utilize hierarchical reinforcement learning to achieve efficient packing planning generation. However, existing methods ignore the perception errors usually found in real-world packaging systems, which enforces the planned sequence and location to deviate from the optimal ones due to the discrepancy between the estimated and actual object shapes. Moreover, constrained by geometric restrictions in packed boxes, conventional planning approaches often overlook placement feasibility for robotic manipulation during plan generation.

C. Robotic Picking and Stowing

The Amazon Robotics Challenge 2017 (ARC) contains similar tasks including picking and stowing, which aims to place objects in an oversized container instead of limited space. Zeng *et al.* predicted the object-agnostic grasp affordance to pick a wide range of objects, and recognized known and unseen instances by cross-domain image matching. Schwarz *et al.* [26] heuristically selected the grasp with clutter graph construction and object pose estimation, and planned the placement with total stack height minimization by modeling objects as bounding boxes. However, they fail to generate effective packing plans to maximize the space utilization ratio due to the huge search space, and the necessary geometric shape estimation of all cluttered objects for packing plan generation cannot be achieved in conventional methods. General object grasping heuristically models the physical dynamics of both objects and grippers [27] or learns valid grasps in a data-driven manner [28] for isolated objects. Valid grasps may be non-existent due to the object occlusion in clutter, and auxiliary manipulation such as pushing [29] and non-targeted grasping [30] singulates the target object by breaking the clutter structure to prepare enough space for grippers. Yang *et al.* [29] marked the target object via presented visual segmentation modules and made the decision of action primitives based on the domain knowledge for successfully grasping. By adjusting the gripper according to the difference between the estimated in-hand object pose [31] and the planned one, the objects are placed in open space with the desired pose in many tasks such as rearrangement [32]. Because some object surface is inaccessible for the gripper in target grasping, extra action primitives containing reorientation and regrasp [11] are employed to generate feasible placement paths for achieving the planned object orientation. In this paper, we integrate feasible manipulation techniques including grasping, placement and rearrangement implement the generated packing plan, so that the general object package in limited box space can be achieved.

III. Approach

A. Open-vocabulary Shape Estimation

Fig. 3 demonstrates the framework of instance segmentation and shape recovery for shape estimation of all objects in the clutter. The shape of all objects in the dense clutter should be accurately estimated to provide necessary information for packing planning, so that a high space utilization ratio can be achieved with a tight package. Since the robotic packing system is usually deployed in diverse scenes, objects from

IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.

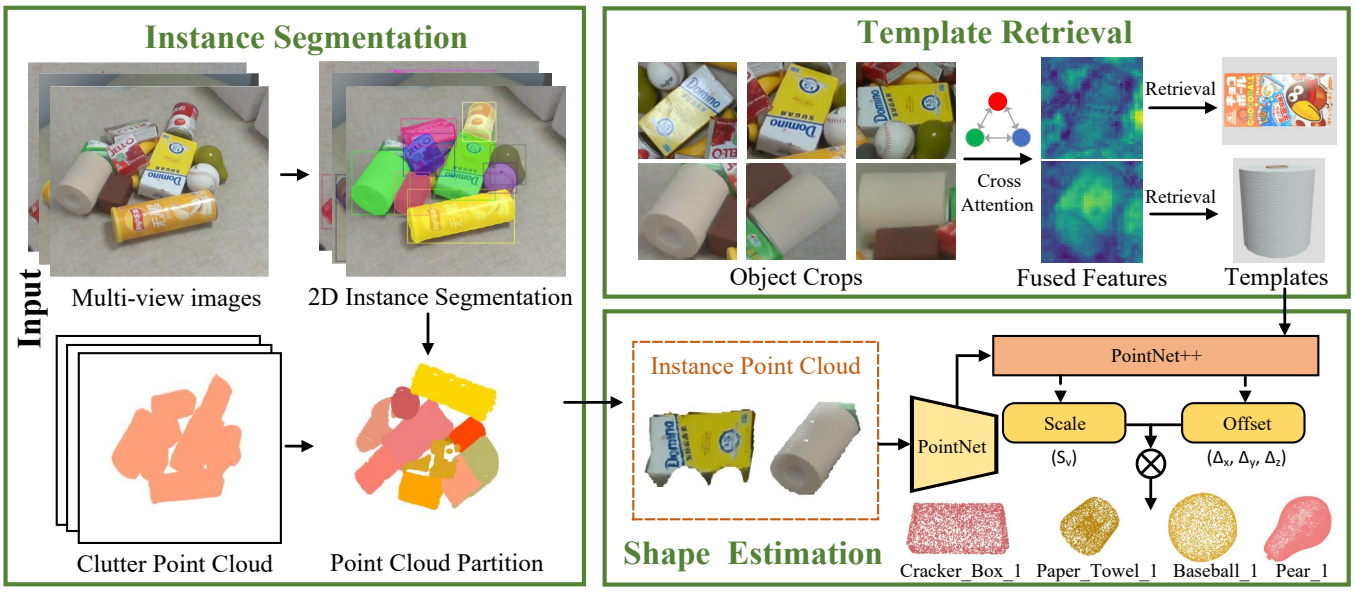


Fig. 3. The framework of open-vocabulary shape estimation, which aims to recover the complete shape of all cluttered objects based on the partial observation. The multi-view RGB images are input to the 2D instance segmentation model to acquire the object masks in each view, which are projected to the point cloud to obtain the instance-wise partition. The fused features of object crops in image patches are leveraged to retrieve the template in the large pool to provide rich geometric priors. The shape estimation network predicts the scaling factor and the point-wise offset with respect to the templates to recover the actual shape of each cluttered object.

categories that are unseen during the training stage may appear. Therefore, we present an open-vocabulary shape estimation method to precisely recover the geometry of objects from both seen and novel classes, which consists of three stages: instance-wise point cloud segmentation, object template retrieval, and deformation prediction regarding templates. The instance-wise point cloud segmentation partitions the clutter point cloud into observation for each object. Object template retrieval compares the features of partially observed object and templates in the large pool with aligned embedding space to select the similar one. Deformation regarding the templates are predicted by regressing the scaling factors and point-wise offset to acquire the fine-grained geometry of each object.

(1) Instance-wise point cloud segmentation: We leverage the pre-trained Mask R-CNN framework [33] with fine-tuning to predict instance-wise masks on multi-view RGB images. Since objects from novel classes may exist, we do not predict mask categories. Masks with low confidence and relevance are filtered out. We then assign instance labels to the clutter point cloud generated from depth images. To avoid over-segmentation, point cloud partitions that significantly overlap in occupancy are merged.

(2) Object template retrieval: Since severe occlusion in the object clutter prevents the acquisition of sufficient visual clues, directly reconstructing the object geometry based on the extremely limited point cloud is infeasible. Leveraging geometric priors, providing a similar template for object shape estimation is beneficial. Because objects in deployment scenarios may come from novel categories unseen during training, we retrieve object templates from the large pool constructed in OpenShape [34] to enhance the generalization capability of shape estimation for diverse objects. RGB images in different camera views provide various geometric information to locate the accurate object template. Thus, we first fuse the features of cropped object patches across multi-view RGB images via cross-attention:

$$\hat{X^o} = \sum_{i=1}^{K} \sum_{j \neq i} \sigma\left(\frac{W_Q X_j^o W_K X_i^o}{\sqrt{d}}\right) W_V X_i^o \tag{1}$$

where $\hat{X^o}$ is the fused feature for template retrieval and X_j^o shows the CLIP [9] features of the cropped image patches for object o acquired from the i_{th} camera view. W_Q, W_K, W_V respectively represent the learnable query, key and value computation matrix, and K is the number of cameras observing the clutter. σ represents the softmax function, and d is the dimension of the feature vectors. The embedding spaces of the 3D point cloud representation, extracted by the backbone OpenShape [34], and the 2D image representation extracted by CLIP are aligned. We compare the template features acquired by OpenShape and the fused features of cropped object patches to retrieve similar templates. The computation matrix in cross-attention is trained with the following object template retrieval objective L_{otr} to minimize the distance between the fused features $\hat{X^o}$ of observed objects and those of the ground-truth complete point cloud from OpenShape $\overline{X^o}$:

$$L_{otr} = ||\hat{X^o} - \overline{X^o}||_2 \tag{2}$$

By imposing the similarity constraint, the cross-attention tends to generate fused features of observed objects that can precisely demonstrate the geometric appearances to retrieve more similar templates.

(3) Deformation prediction regarding templates: With the geometric priors provided by the retrieved templates, the shape

IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.

of observed instances can be recovered by re-scaling and point-wise offset in the following way:

$$\hat{S} = \alpha_s T + \Delta S \tag{3}$$

where $\hat{S}$ is the predicted shape and T means the point cloud of retrieved templates. $\alpha_s \in \mathbb{R}^3$ represents the scaling operation that scales the coordinates of templates in each axis. The point-wise offset $\Delta S \in \mathbb{R}^{p\times 3}$ indicates the shift of each point in the template with p points. The scaling factor α_s and the point-wise offset ΔS are predicted by shape recovery networks which consider the features of both the templates and the partially observed objects:

$$(\alpha_s, \Delta S) = \mathrm{MLP}([\phi(\tilde{X}^o), \psi(X^t)]) \tag{4}$$

where $\tilde{X}^o$ and X^t are features of the observed object and corresponding templates acquired via OpenShape. ϕ and ψ are transformation functions of the features for deformation prediction, which are parameterized by one fully-connected layer with activations. We also leverage 3D graph convolutional networks to implement the MLPs in the prediction. The training objective includes minimizing the Chamfer distance between the predicted shape and the actual one, and minimizing the average point-wise offset with hyperparameter η:

$$L_{est} = d_{ch}(\boldsymbol{S}, \hat{\boldsymbol{S}}) + \eta \frac{||\Delta \boldsymbol{S}||_2}{p} \tag{5}$$

where d_{ch} denotes the Chamfer distance between the predicted shape $\hat{\boldsymbol{S}}$ and the ground-truth shape $\boldsymbol{S}$, and p represents the number of points in the predicted shape. The objective encourages the shape recovery networks to reconstruct accurate object shapes from partial observations while preventing network overfitting.

B. Interactive Object Information Collection

The severe occlusion in the dense clutter prevents the packing system to recover satisfying object shape due to the extremely limited observation. Therefore, we generate interactive actions that are executed by robots to the clutter, and uncover more informative visual clues for estimation of the uncertain shape. Our goal is to obtain the maximal information gain with the least interaction cost. We leverage push actions to effectively break the clutter structure for accurate visual perception, because push actions are highly efficient in execution. Since the perception uncertainty provides guidance for effective interaction generation, we evaluate the perception uncertainty of the clutter in the top-down view. The perception uncertainty U_{ij} for the pixel in the i_{th} row and j_{th} column is composed of the segmentation entropy U_{ij}^{ent}, the object inconsistency U_{ij}^{inc} and the shape uncertainty U_{ij}^{sha} with hyperparameters λ_1 and λ_2 for balance:

$$U_{ij} = U_{ij}^{ent} + \lambda_1 U_{ij}^{inc} + \lambda_2 U_{ij}^{sha} \tag{6}$$

The segmentation entropy represents the Shannon entropy of object detection and foreground segmentation in the multi-view RGB-D images:

$$U_{ij}^{ent} = \sum_{k=1}^{K} \sum_{rs} u_{rs}^k \cdot \mathbb{I}(P_{td}(p_{rs}^k) = p_{ij}^{td}) \tag{7}$$

where $P_{td}(x)$ demonstrates the coordination where the center of pixel x in other views is mapped to the top-down view. p_{rs}^k and p_{rs}^{td} represent the pixel in the r_{th} row and s_{th} column in the k_{th} view and in the top-down view respectively. Meanwhile, $\mathbb{I}(x)$ is the indicator function that equals to one for true x and zero otherwise. The segmentation entropy contributed by p_{rs}^k is as follows:

$$u_{rs}^k = \sum_t p_{rs,f}^{tk,se} \log p_{rs,f}^{tk,se} + p_{rs,b}^{tk,se} \log p_{rs,b}^{tk,se} \tag{8}$$

where $p_{rs,f}^{tk,se}$ and $p_{rs,b}^{tk,se}$ respectively demonstrate the probability being classified into the foreground and background probability of the t_{th} bounding box that contains the pixel in the r_{th} row and s_{th} column for the k_{th}-view image. Accurately recognizing objects in the regions with high entropy for instance segmentation is usually difficult, because the instance segmentation networks are uncertain about the prediction. The object inconsistency depicts the number of predicted instances within one pixel:

$$U_{ij}^{obj} = N(\{c_{rs}^k | P_{td}(p_{rs}^k) \subset p_{ij}^{td}\}) \tag{9}$$

where c_{rs}^k means the predicted instance index of the pixel in the r_{th} row and s_{th} column from the k_{th} view, and $N(s)$ stands for the number of non-repeating elements in the set $\{s\}$. Regions with high object inconsistency represent contradictory segmentation masks, which also suggests high uncertainty in the prediction. The shape uncertainty for a pixel represents the standard errors of scaling factors and the shape distortion for the object whose observed point cloud occupies the pixel in the top-down view. Since the standard errors of scaling factors and the shape distortion are not statistically correlated, the shape uncertainty can be regarded as their linear combination:

$$\begin{aligned} &U_{ij}^{sha} = \sigma_{k,x}^2 + \sigma_{k,y}^2 + \sigma_{k,z}^2 + \alpha_d(\hat{x}^2 + \hat{y}^2 + \hat{z}^2) \\ &s.t.\ p_{ij}^{td} \subset P_{td}(PC_n) \end{aligned} \tag{10}$$

where PC_n means the point cloud observation of the n_{th} object. $\sigma_{n,x}$, $\sigma_{n,y}$, $\sigma_{n,z}$ respectively represent the standard errors of scaling factors in x, y, z axis for the n_{th} object, which are predicted by the extra branch in shape recovery networks. $\hat{x}$, $\hat{y}$, $\hat{z}$ are the principal axis length in different directions of the predicted shape, and α_d is the pre-defined deformability coefficient assigned to the retrieved templates. Objects in large size or with high deformability achieve higher uncertainty in the estimated shape. More details in APPENDIX C.

The interaction with the clutter is generated for the prediction of shapes with high uncertainty. We select the start point of interaction by considering the average perception uncertainty in square regions with fixed size, and the region whose center is the pixel in the i_{th} row and j_{th} column is denoted as A_{ij}. The square regions with the highest average perception uncertainty are selected to generate an interaction if grippers can descend significantly to effectively break the clutter structure for visual clue discovery. The square region A_{ij}^* for start point selection is formulated as follows with the hyperparameter h_0:

$$A_{ij}^* = \arg\max_{A_{ij}} \ \overline{U}(A_{ij}) \cdot \mathbb{I}(h_{ij}^{max} - h_{ij}^{des} < h_0) \tag{11}$$

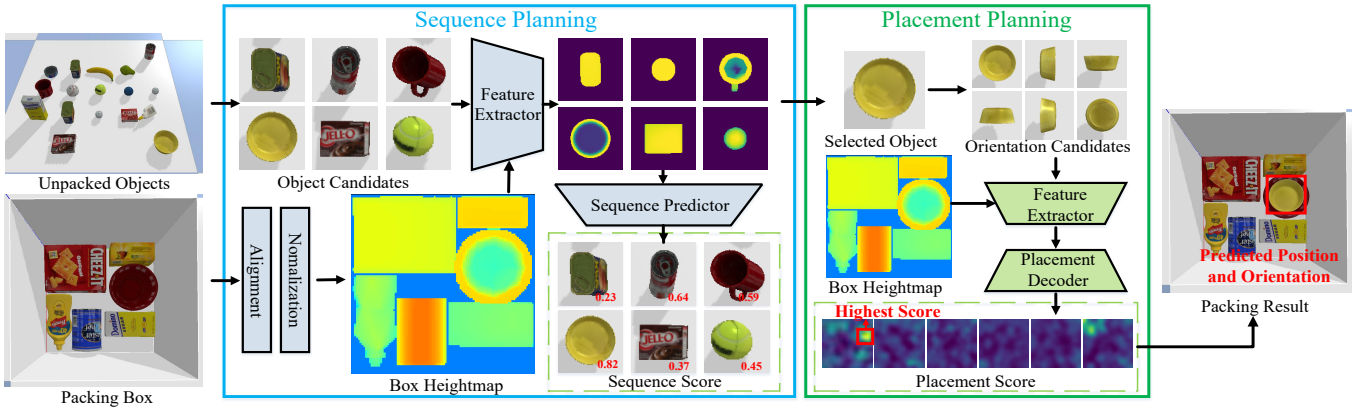


Fig. 4. The pipeline of hierarchical reinforcement learning based packing planning that generates the object sequence, location and orientation to maximize the space utilization ratio.

where $\overline{U}(A_{ij})$ is the average perception uncertainty in the region A_{ij}. The largest height and the gripper descent height in A_{ij} are denoted as $h_{ij}^{\max}$ and h_{ij}^{des}, respectively. To ensure the validity of the interaction, we require a significant difference between the largest height and the gripper descent height, and the region with the highest perception uncertainty is selected for interaction to maximize information gain. We enumerate A_{ij}^{*} to select the position with the lowest gripper descent height as the start point, because grippers can be inserted into the clutter to the largest depth to most significantly change the clutter structure for visual clue discovery.

We utilize the direction from the start point to the pixel with the largest perception uncertainty as the push direction. When the start point is close to the pixel with the largest uncertainty, a random direction is chosen for push actions. The push distance is set to a constant and scaled twice if the distance between the start points of consecutive push actions is close. The interaction with the clutter is generated only when the largest average perception uncertainty of the square regions A_{ij} across all locations surpasses a threshold, because perception with high reliability does not require interactive object information collection to enhance efficiency.

C. Packing Planning

Packing planning generates object order, position, and orientation to maximize box space utilization based on object shape estimates. Additionally, it must consider stability, manipulation feasibility, and perception error robustness for real-world deployment. High spatial resolution for fine-grained positioning expands the search space, causing search inefficiency. Therefore, we follow [25] to leverage hierarchical reinforcement learning algorithms to effectively search for the optimal plan. However, existing methods ignore visual information noise and manipulation safety constraints in practical deployment scenes, resulting in generated plans that are not applicable to complex and diverse factory deployment environments. Compared to [25], we have improved the reward function design by introducing visual perception errors and manipulation feasibility to further ensure that the packing plan meets physical constraints, enabling it to better adapt to real-world deployment scenes. Hierarchical reinforcement learning usually consists of agents from two hierarchies including a top-level agent and a bottom-level agent. The top-level agent sets sub-goals for the bottom-level agent as the search objective, and the bottom-level agent searches the optimal sub-policy to accomplish the given sub-goal in each decomposed space. In packing planning, the top-level agent predicts the object index to be packed next based on the heightmaps of in-box objects and each unpacked object, and the bottom-level agent selects the optimal location and orientation for the selected object. Fig. 4 shows the pipeline of packing planning.

1) Sequence Planning: The top-level agent predicts the packing sequence which sets the sub-goal for placement planning. The states and actions for the sequence prediction agent are introduced as follows:

State: The state space represents all unpacked objects and the current object arrangement in the packing box. The standardized packing box heightmap in the top-down view and the six-view heightmaps of each remaining object in the clutter are leveraged as the input for the top-level agent. The six-view heightmaps are collected for each object at a pose where objects are placed stably on the plane, and the zero plane of the heightmap is set as the opposite plane of the object bounding box for each view. The top-down view heightmap is utilized to represent the object arrangement in the packing box.

Action: The action space of the sequence planning agent includes the selection of the optimal object to pack next. We extract the visual features of unpacked objects by convolutional neural networks based on the six-view heightmap of unpacked objects and the top-down view heightmap of the packing box. By concatenating feature maps across all objects, the sequence planning agent predicts the score for each unpakced object where the object with the highest score is selected for package. To keep the channel number consistency in convolutional neural networks, we pre-define the number of input channels (e.g. K=100), and set the feature maps of packed or non-existing objects to be all-zero matrix. The score of packed and non-existing objects are manually selected to zero so that they will not be selected to pack next. After the location and orientation of the selected object is decided by the placement planning agent, the heightmap of the packing box is modified and feature maps of the selected object are set to all-zero matrix in subsequent sequence planning.

2) Placement Planning: The bottom-level agent generates the location and orientation for placement, and we describe the states and actions in the following.

State: The state space is demonstrated by the index of the object for package and the current object arrangement in the packing box. Similar to the sequence agent, we use the six-view heightmap of the selected object for representation, and leverage the top-down view heightmap of the packing box to depict the object arrangement.

Action: We use heuristic discrete action spaces to improve search efficiency. The action space consists of the orientation and the location of the selected object in the packing box. We first scan the object with different rolls and pitches to acquire the heightmap of the selected object in the top-down and bottom-up views. The roll and pitch are equally discretized into n^2 grids for the search space of $[0, 2\pi] \times [0, 2\pi]$, and those in the i_{th} grid are represented by (ϕ_i, θ_i). We rotate the obtained heightmaps with various yaws, and ψ_j means the j_{th} discrete grid for yaw that is equally discretized into m grids. The top-down and bottom-up view heightmaps for the orientation $(\phi_i, \theta_i, \psi_j)$ are concatenated with the heightmap of the packing box for the placement planning agent to generate the placement score matrix W_{ij}. The element in the x_{th} row and y_{th} column of W_{ij} means the score of putting the object in the location (x, y) with the orientation $(\phi_i, \theta_i, \psi_j)$. The score of the invalid region is manually set to zero if a collision appears between objects and box walls. The orientation and horizontal location with the highest placement score is selected for the object package, and the vertical location of the object is decided by the lowest height where the selected object does not collide with other in-box objects.

3) Reward Function: We apply the Q-learning framework [35] to train our hierarchical agents, and the reward function includes compactness C, stability S, perception error tolerance T and placement feasibility F, which are demonstrated in Fig. 5. Denoting the packing state at the t_{th} step as s_t, the overall objective at s_t can be written as follows:

$$\max J(s_t) = \alpha_1 C + \alpha_2 S + \alpha_3 T + \alpha_4 F \tag{12}$$

We detail different terms in the overall reward and their goals in the following.

Compactness: The compactness depicts the ratio between the occupied volume to that of the smallest box that can contain all packed objects, where the bottom area is fixed:

$$C = \sum_{i=1}^{t} \frac{V_i^o}{LWH_t^{max}} \tag{13}$$

where V_i^o means the volume of the i_{th} packed objects in the box, and L and W are the length and width of the box respectively. H_t^{max} represents the maximum height of packed objects in the box after putting the t_{th} object into the container, which can be depicted by the largest value in the box heightmap. The compactness is expected to be maximized to enhance the space utilization ratio and save package cost. More details in APPENDIX C.

Stability: The stability represents whether the object is stable against the formerly packed objects and the packing box without pile shift. The analytical solution is intractable because of the inaccessible contact points among objects. We alternatively leverage the difference between the planned and the actual object location and orientation for stability checking, and the plan is regarded as stable for the difference less than the threshold:

$$S = \mathbb{I}(\delta_t^d < h_d) \cdot \mathbb{I}(\delta_t^r < h_r) \tag{14}$$

The differences between the actual and planned object position and rotation for the t_{th} object are denoted as δ_t^d and δ_t^r respectively, and the threshold for stability definition is h_d and h_r. The consistency between planned and actual object location and orientation means the balance of force and torque, and the small difference indicates that the state of in-box objects is only modified slightly with negligible decrease on space utilization ratio.

Perception error tolerance: Perception errors of object shape estimation can be caused by the severe occlusion in the dense clutter and the object deformability, and the inconsistency between the estimated and actual shape can decrease the space utilization ratio of the packing plan. Robustifying the packing plan to perception errors is necessary in the robotic packing system, and we require the margin to the nearest object should be positively correlated of the shape uncertainty:

$$T = ||\frac{d_t^{nea}}{U_t^{sha}} - d_0^{nea}||_2 \tag{15}$$

where d_t^{nea} is the distance to the nearest in-box object or box walls for the t_{th} packed instance, and the corresponding perception errors U_t^{sha} are with similar definition in (10) to represent the shape uncertainty. The slight difference is that we consider the standard errors of scaling factors and deformation for the t_{th} object instead of the pixel p_{ij}^{td} in the top-down view. d_0^{nea} is a hyperparameter to balance perception error tolerance and space utilization ratio, and large d_0^{nea} leads to more margins among objects with higher robustness to perception errors while the space waste is also more significant. In the reward, more tolerance should be provided for objects with more uncertain shape to prevent collision during placement.

Manipulation feasibility: Manipulation feasibility demonstrates whether the manipulator can achieve the planned location and orientation without collision among the in-box objects, box walls and grippers. The manipulation feasibility is set to one if there is at least one possible one-stage or two-stage collision-free placement path and is set to zero otherwise. The one-stage top-down placement paths to the planned location are selected if the object and the grippers are free of collision with the box walls and other in-box objects. Otherwise, the intermediate position is first selected for top-down placement followed by push actions to the planned location.

The existence of collision-free one-stage placement paths is judged by the rule that the heightmap pixels at the grippers should be smaller than the calculated gripper height in the final state, so that the grippers are collision-free during the descent. The existence of two-stage collision-free placement paths is equivalent to that of one-stage collision-free placement in

IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.

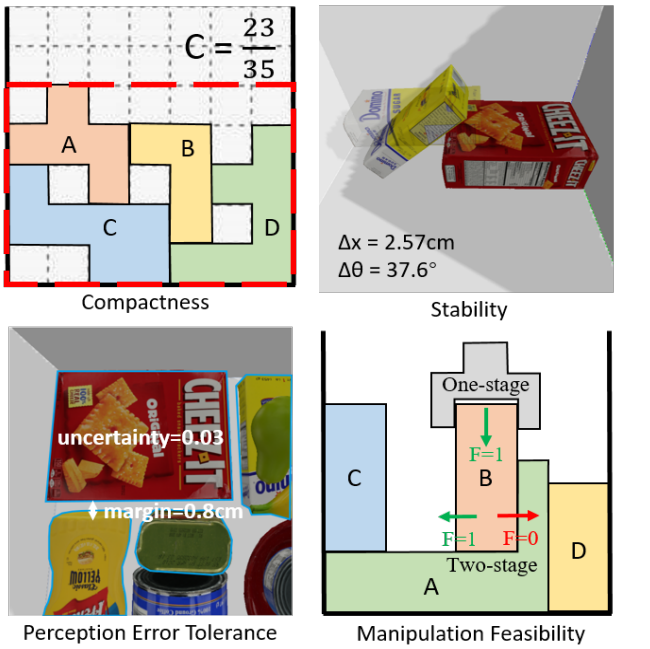


Fig. 5. Examples of compactness, stability perception error tolerance and manipulation feasibility in the reward of packing planning.

(a) Training Objects (b) Novel Objects in Testing

Fig. 6. (a) Objects adopted in training scenarios. (b) Novel objects in testing where some of them are from unseen classes. Numbers beside objects mean the deformability coefficients, and those without numbers are rigid objects.

intermediate positions, where $\boldsymbol{L}$ represents the region covered by grippers in descending:

$$\{\boldsymbol{L} | \sup_{x\in\boldsymbol{L}} h(x) \leq h_g, \sup_{x\in\overline{\boldsymbol{L}}} h(x) - \inf_{x\in\overline{\boldsymbol{L}}} h(x) \leq h_0^s\} \tag{16}$$

where $h(x)$ is the height value for the pixel x, and $\overline{\boldsymbol{L}}$ represents the region covered by closed grippers when pushing objects from the intermediate location to the planned packing position. h_g and h_0^s stand for the gripper height and the smoothness threshold. The first condition ensures that the existence of the top-down placement for the intermediate position, and the second condition prevents the height fluctuation during object pushing for manipulation stability.

IV. Experiments

In this section, we first introduce the algorithmic details of RoboPacker. Then we evaluate each modules of RoboPacker for autonomous general object packaging, including open-vocabulary shape estimation, interactive object information collection, and packing planning. Finally, we compare the performance in general object packaging between RoboPacker and baselines. More experimental results in APPENDIX E.

TABLE I
The Chamfer distance ($\times 10^{-3}$) between the predicted and the groundtruth shape w.r.t. different template settings.

Method	None	Training	Template	Ours
CD	27.7	2.7	0.7	0.3

TABLE II
The Chamfer distance ($\times 10^{-5}$) between the predicted and the groundtruth shape w.r.t. different camera numbers and feature fusion methods.

#Cameras	1	2	3	4	5
Sum	26.9	32.0	35.0	35.9	39.5
MaxPool	26.9	27.0	27.3	28.2	28.6
Ours	26.9	26.9	26.5	26.5	27.5

A. Algorithm Details

We employ the Mask R-CNN [33] framework for instance mask generation, where the backbone networks are pre-trained on the COCO dataset [36]. The segmentation confidence threshold and the IoU threshold for positive prediction are 0.35 and 0.8, respectively. Instances in different views that occupy more than 64 voxels are merged during point cloud partitioning. For deformation prediction, we adopt 3D graph convolutional networks for feature extraction, followed by three fully-connected layers to predict the deformation relative to templates. The hyperparameter η in the training objective (5) is set to 0.01. The interactive exploration is executed when the largest perception uncertainty shown in (6) is higher than 2.7, where the hyperparameters λ_1 and λ_2 are set to 0.1 and 0.2, respectively. All templates in the pool are divided into three deformability types: rigid, slightly deformable, and deformable, where the deformable coefficients α_d are set to 0, 1.05, and 1.85, respectively. The size of the square region A_{ij} for start point generation in interactive object information collection is set to 64×64. Random directions for pushing are selected if the distance between the start point and the pixel with the largest uncertainty is less than 2 cm. The initial pushing distance in interactive exploration is set to 5 cm and scaled twice when the distance between start points of two consecutive push actions is less than 2 cm. To avoid trivial exploration, the maximum number of push actions in one round of interactive exploration is set to 3.

For packing planning, we leverage ResNet18 [37] as the backbone for visual feature extraction in the top-level agent, which predicts the score for each unpacked object via three-layer fully-connected networks. In the bottom-level agent, the orientation search intervals for roll, pitch, and yaw are all set to $\pi/2$ by default. We employ a U-Net [38] architecture network to generate the score matrix with the same size as the box heightmap for each discrete orientation, enabling the selection of packing location and orientation. In the objective function, the hyperparameters $\alpha_1 \sim \alpha_4$ are set to 0.75, 0.25, 0.1, and 0.05, respectively. The thresholds h_d and h_r for stability judgement are 2 cm and π/6, and the hyperparameter d_0^{nea}, which balances perception error tolerance and space utilization ratio, is set to 50. To simulate the perception error e in packing planning module training, all objects are rescaled by a factor α, which is sampled from the normal distribution $\alpha \sim \mathcal{N}(1, e)$ with $e = 0.1$.

We follow [30] in target grasping, and the minimal height

IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.

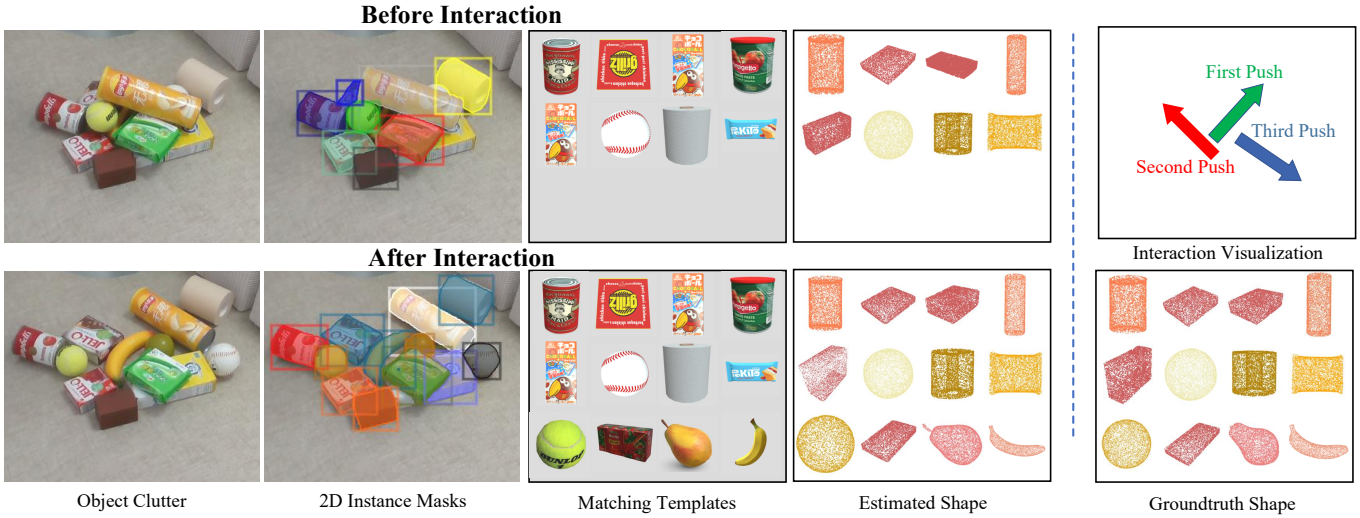


Fig. 7. The object clutter, 2D instance masks, matching templates, estimated shape before and after interaction. Segmentation masks in different colors represent different instances, and objects without masks are regarded as false positives. We also visualize the push actions in interaction and the groundtruth shape of objects.

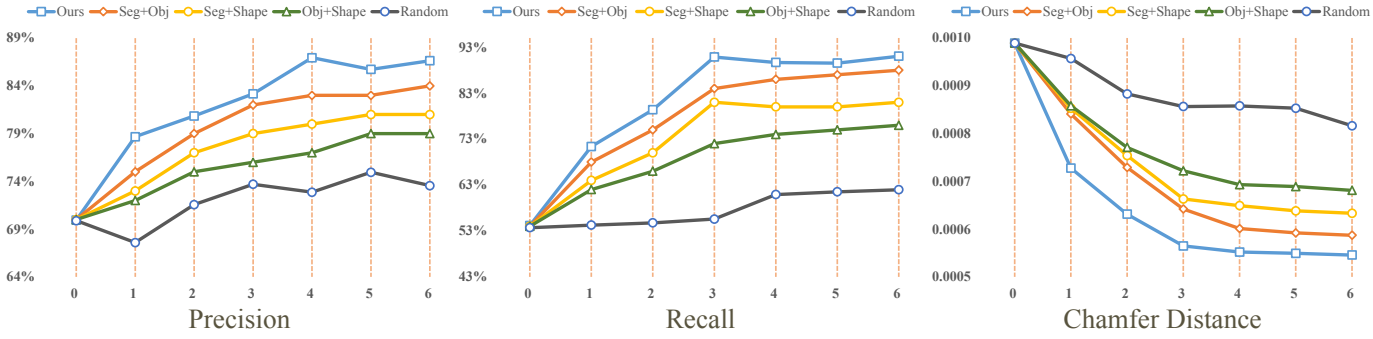


Fig. 8. Precision and recall of instance segmentation and the Chamfer distance between the predicted and groundtruth object shape regarding different uncertainty definition and the number of push actions. Three interactions can effectively uncover informative visual clues as the perception performance does not increase significantly with action steps over three.

differences between objects and contact points for grasping and pushing candidate generation are set to 1.5cm and 2.5cm respectively. For placement path generation, the smoothness threshold h_0^s in two-stage placement is assigned to 1cm. To evaluate the space utilization ratio of the packing box, we leverage the compactness and the success rate as the metric for physical experiments, and report the compactness and the maximal number of objects in the simulator. Compactness represents the ratio between the occupied volume to the volume of the minimal box that can contain all given objects, as depicted in Fig. 5. The success rate means the ratio of the cases that successfully packing all objects into the box with fixed sizes. The maximal number of objects demonstrate the object number that can be contained in the box with fixed sizes in unlimited settings, where the objects are generated with random index and scaling factors. We selected the latest irregular object packing as the baseline HM [8] to verify the superiority of RoboPacker. Further, we added the latest packing planning algorithm PackIt [39] as the baseline to verify the effectiveness of the hierarchical packing planning module. More deployment details are in APPENDIX D.

B. Evaluating Different Modules in the System

1) Open-vocabulary Shape Estimation: Since objects from novel classes usually appear in deployment, we retrieve the object templates from the large pool with the aligned representation for open-vocabulary shape estimation. Fig. 7 demonstrates an example of the instance segmentation masks, the retrieved templates, the estimated shape, and the groundtruth shape for the dense object clutter. Objects that are never seen in the training stage such as the paper roll and the sponge can be matched with the template whose geometry is still very similar to the object shape. The fused CLIP features of observed objects are aligned with the representation extracted by OpenShape method because of the large-scale pre-training, so that the generalization ability to unseen objects in template retrieval can be boosted. All experiments in the evaluation of the proposed open-vocabulary shape estimation are conducted with 10 pre-defined clutters. We show the quantitative influence of the template retrieval with the aligned pre-trained representation space on the open-vocabulary shape estimation. We conduct experiments in the scenarios where we directly predict the complete point cloud without considering templates (none), leverage the category-level templates of the training objects (training) and employ the retrieved template as the estimated shape without deformation (templates), and

IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.

Fig. 9. Visualization of the generated packing plan including the object sequence, orientation and location in the packing box. Our method learns effective packing policies such as putting cups on plates and placing balls in cups to enhance the space utilization ratio.

Table I reports the Chamfer distance between the predicted and the groundtruth. Without priors from the templates, the predicted object shape is significantly different from the groundtruth one because the limited observation of objects in the dense clutter fail to provide sufficient visual information for shape estimation. The templates retrieved from the large pool can enhance the accuracy of shape estimation, because the category-level templates of training objects cannot provide informative geometric priors for unseen objects. Deformation regarding the retrieved templates also contributes to the shape estimation accuracy due to the existed shape variances among different instances.

We also investigate the influence of camera numbers and multi-view object crop feature fusion methods, and we demonstrate the Chamfer distance between the predicted and the groundtruth shape with different experimental settings in Table II. We changed the number of cameras and mount them evenly on the same horizontal plane, and we adopt summation (sum) and maxpooling (maxpool) over features from different views for the fusion methods for comparison. Increasing the camera numbers can significantly enhance the performance of shape estimation when the camera number is low, because the marginal information provided by extra views is high for visual clue collection. The performance increase is slight for camera numbers over three, and adding more views for instance segmentation causes more computational cost. Therefore, we choose the number of cameras observing the object clutter to be three to achieve the satisfying trade-off between the segmentation accuracy and computational cost. Moreover, leveraging the cross-attention for feature fusion across views outperforms other methods.

2) Interactive Object Information Collection: Since shape estimation for densely cluttered objects usually suffers from severe occlusion, we generate interactions to clutter to break up the clutter structure to uncover informative visual clues. Fig. 7 shows an example of the the instance segmentation masks and the predicted complete shape before and after interactive exploration. The start point, direction and distance depicted by the arrows are decided according to the overall uncertainty in order to maximize the information gain. The dense regions with the highest ambiguity in perception are broken for informative visual clue discovery. Comparing the estimated shape before and after interaction, we can conclude that the interactive exploration can discover the occluded instances with more accurate shape recovery for subsequent packing planning.

TABLE III
THE SPACE UTILIZATION RATIO OF OUR GENERATED PACKING PLANS W.R.T. DIFFERENT PERCEPTION ERRORS.

Perception Errors	Compactness	#Objects
0	0.431	38.94
0.1	0.419	36.85
0.2	0.403	35.26

TABLE IV
THE SPACE UTILIZATION RATIO OF OUR GENERATED PACKING PLAN AND THAT WITHOUT CONSIDERING PERCEPTION ERROR TOLERANCE (PET) OR PLACEMENT FEASIBILITY (PF).

Method	Compactness	#Objects
Ours	0.419	36.85
Ours without PET	0.385	33.79
Ours without PF	0.396	34.66

The quantitative evaluation is conducted in 10 pre-defined easy clutter scenarios with 10-20 objects for each clutter. We leverage the precision and recall of instance recognition and utilize Chamfer distance between the predicted and groundtruth object point cloud to evaluate the generated interaction. Fig. 8 shows the performance variance with exploration steps, and other baselines include random pushing (Random), and push generation without segmentation entropy (Obj+Shape), without object disagreement (Seg+Shape) or without shape uncertainty (Seg+Obj). Our interactive exploration can enhance the recall by 37% (91% vs. 54%) and reduce the Chamfer distance by 61% (0.00069 vs. 0.00027) via only 3 push actions. The information gain revealed by recall, precision and Chamfer distance decreases rapidly when the number of actions is larger than three, and our interactive exploration efficiently mines the visual information in the clutter with least actions. With three push actions, the recall and Chamfer distance advantages compared with the random methods clearly show the effectiveness of uncertainty-based exploration on the recall and Chamfer distance (36% and 0.031). Different terms in the definition of the perception uncertainty all influence shape estimation accuracy, where segmentation uncertainty makes the most significant contribution because of the fine-grained informativeness measurement.

3) Packing Sequence and Location Generation: Fig. 9 illustrates an example of the generated packing plan including the object sequence, orientation and location. Our method learns effective packing policies such as putting cups on plates and placing balls in cups to enhance the space utilization ratio. Objects are separated by a suitable margin to achieve the robustness to perception errors and manipulation incapability without ineffective space utilization. We apply the compact-

10 Objects 15 Objects 20 Objects

Fig. 10. Visualization of realistic packing results across different numbers of ob

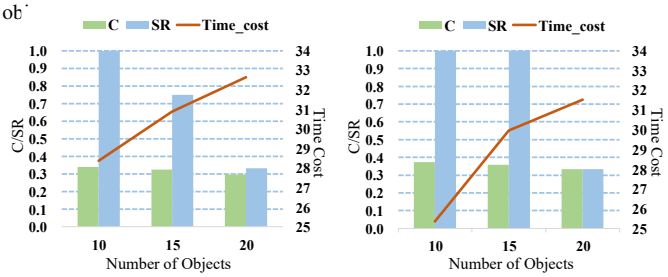


Fig. 11. The compactness (C), the success rate (SR) and the average time cost of our robotic packing system regarding different numbers of objects and box side lengths.

ness and the maximal number of packing objects to evaluate the space utilization ratio in the simulator across 50 object combinations, where the size of the packing box for evaluation is $40 \times 40 \times 40$cm.

Visual perception errors lead to inaccurate estimated object shape, and we evaluate the robustness to perception errors of our packing planning module in Table III. In order to prevent the performance fluctuation of the visual perception on the evaluation, we simulate the perception errors by deforming the object scale, which is introduced in implementation details. Increasing perception errors only slightly influence the space utilization ratio. The results shown in Table IV suggest that the space utilization ratio drops significantly without considering the tolerance to perception errors.

The spatial occupancy of grippers disables top-down placement into the packing box for tightly arranged objects, so that the in-hand objects should be placed in the intermediate position followed by push actions to achieve the goal. Therefore, the space between the intermediate and the goal position should be clear in the packing plan for successful execution of the above manipulation. Table IV depicts the space utilization ratio for our packing plan and that generated without considering the placement feasibility. The second optimal position is applied in the experiment if no collision-free placement path exists for the optimal one in the simulated environment. The reason why we utilize the simulated placement instead of the actual one is that the influence caused by the robot arm control should be removed for fair evaluation of packing planning algorithms. Without considering placement feasibility, the space utilization ratio sizably decreases because of the collision in placement.

C. Evaluating the Overall System

Table V shows the quantitative experimental results on 10 pre-defined clutter scenarios that verify the effectiveness of each presented technique on the space utilization ratio,

TABLE V
THE COMPACTNESS (C) AND SUCCESS RATE (SR) OF PACKING RESULTS, THE AVERAGE EXECUTION TIME (EXE.), INFERENCE TIME (INF.) AND TOTAL TIME (TOT.) FOR PACKING ONE OBJECT.

Method	C	SR	Exe.	Inf.	Tot.
Random	0.264	33.3	-	-	-
HM Search [8]	0.314	53.3	32.0	7.3	39.3
SHM Search [8]	0.321	60.0	32.2	6.5	38.7
PackIt Search [39]	0.306	46.7	32.3	7.6	39.9
Fixed Template	0.297	46.7	29.0	2.7	31.7
No Deformation	0.307	46.7	29.0	3.0	32.0
No Tolerance	0.334	60.0	27.8	3.3	31.1
No Feasibility	0.320	46.7	28.3	3.6	31.9
No Interaction	0.311	53.3	36.7	2.7	39.4
No Regrasp	0.281	40.0	23.1	3.5	26.6
One-stage Only	0.299	46.7	24.8	3.5	28.3
RoboPacker	0.355	73.3	26.9	3.4	30.3

where each clutter contains 10-20 objects. The latency is also provided to show the influence on efficiency. The baseline methods include our RoboPacker (a) leveraging the fixed templates of training objects in shape estimation (fixed template), (b) estimating object shape by templates without deformation (no deformation), (c) removing interactive exploration (no interaction), (d) leveraging heightmap maximization search method [8] (HM search), (e) leveraging stable heightmap maximization search method (SHM search), (f) leveraging PackIt search method (PackIt search) (g) removing the perception error tolerance in planning reward (no tolerance), (h) removing manipulation feasibility in planning reward (no feasibility), (i) removing regrasp operations (no regrasp), (j) placing objects only with one-stage top-down paths (one-stage only). We report the compactness and the success rate revealing the space utilization ratio, and also provide the performance of randomly stacking objects into the packing box (Random). The sizes of the packing boxes are $40\times40\times20$cm for evaluating the success of the packing process. All presented techniques in open-vocabulary shape estimation, interactive instance segmentation, packing planning and system implementation improve the space utilization ratio of robotic packing. Specifically, levering templates of training objects fail to provide informative geometric priors for unseen objects in deployment for shape estimation, and estimating the object shape without deformation fail to acquire the precise object shape. They both significantly degrade the space utilization ratio due to the infeasible packing plan caused by inaccurately estimated shape. Meanwhile, the execution time increases because the inaccurate visual perception reduces the efficiency of actual manipulation. Removing the learning based packing plan generation, the perception error tolerance and manipulation feasibility in planning reward underperforms our RoboPacker because of the inferiority of the search algorithm and the weak robustness to clutter occlusion and placement incapability. Moreover, searching with the heuristic algorithms such as heightmap maximization increases the inference time by a large margin due to the plan enumeration, and conventional reinforcement learning strategy such as PackIt cannot effectively search the plan in the large search space. Removing regrasp operations and two-stage placement disables the robotic packing systems completely, as the manipulation to achieve the planned location and orientation is infeasible. Our

RoboPacker outperforms random object stacking by a sizable margin, which verifies the effectiveness of autonomous robotic packing.

Fig. 10 visualizes several packing results across different object numbers, and Fig. 11 demonstrates the space utilization ratio and the average packing time cost of each object with different box sizes and object numbers. Enlarging the box sizes strengthens the space utilization ratio because the larger margin between objects in the packing plan leads to higher robustness to perception errors and manipulation infeasibility, and decreases the implementation latency because the regrasp operations and two-stage placement are less required with lower crowdedness in the box. Meanwhile, increasing the number of objects degrades the space utilization ratio because the grippers are more likely to be collided with other objects or the box walls due to the crowdedness.

V. Conclusion

Object packaging plays an important role in industrial applications and daily tasks, and requires a large amount of experienced human packers with exhaustive labor. In this work, we have proposed an autonomous robotic packing system, RoboPacker, to store general objects into a confined space with high utilization ratio. We design an open-vocabulary shape estimation pipeline to recover the point cloud for objects from both seen and novel classes, where we also generate interactions to the clutter to uncover more informative visual clues to enhance the shape estimation of cluttered objects. Meanwhile, we propose a hierarchical reinforcement learning based packing planning framework to acquire the order, location and orientation for object package with the goal of maximizing space utilization. Moreover, we build a robotic packing system by combining the presented methods with advanced manipulation techniques for realistic implementation. Extensive experiments in both the simulator and the real world demonstrate the effectiveness of the robotic packing system.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grant U22B2050, in part by the Beijing Natural Science Foundation under Grant L257008, and in part by the National Natural Science Foundation of China under Grant 62376032, and in part by the Singapore NRP DS-RFM Grant M25N4N2009.

References

[1] D. Coelho, M. Oliveira, and V. Santos, "Rlad: Reinforcement learning from pixels for autonomous driving in urban environments," *TASE*, vol. 21, no. 4, pp. 7427–7435, 2024.

[2] I. Cvišić, I. Marković, and I. Petrović, "Soft2: Stereo visual odometry for road vehicles based on a point-to-epipolar-line metric," *TRO*, vol. 39, no. 1, pp. 273–288, 2023.

[3] J. Schrittwieser, I. Antonoglou, T. Hubert, K. Simonyan, L. Sifre, S. Schmitt, A. Guez, E. Lockhart, D. Hassabis, T. Graepel, T. Lillicrap, and D. Silver, "Mastering atari, go, chess and shogi by planning with a learned model," *Nature*, vol. 588, no. 7839, pp. 604–609, 2020.

[4] H. Zhang, L. Pang, M. Bai, J. Yang, and J. Zhao, "Retinal surgical field realignment based on master-slave dual-arm surgical robot," *TASE*, vol. 21, no. 3, pp. 4743–4752, 2024.

[5] 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," *TASE*, vol. 21, no. 1, pp. 939–950, 2024.

[6] G. Tresca, G. Cavone, R. Carli, A. Cerviotti, and M. Dotoli, "Automating bin packing: A layer building matheuristics for cost effective logistics," *TASE*, vol. 19, no. 3, pp. 1599–1613, 2022.

[7] D. Morrison, A. Tow, M. McTaggart, R. Smith, N. Kelly-Boxall, S. Wade-McCue, J. Erskine, R. Grinover, A. Gurman, T. Hunn, D. Lee, A. Milan, T. Pham, G. Rallos, A. Razjigaev, T. Rowntree, K. Vijay, Z. Zhuang, C. Lehnert, I. Reid, P. Corke, and J. Leitner, "Cartman: The low-cost cartesian manipulator that won the amazon robotics challenge," in *ICRA*, 2018, pp. 7757–7764.

[8] F. Wang and K. Hauser, "Dense robotic packing of irregular and novel 3d objects," *TRO*, vol. 38, no. 2, pp. 1160–1173, 2022.

[9] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *ICML*, 2021, pp. 8748–8763.

[10] Y. Laili, Z. Chen, L. Ren, X. Wang, and M. J. Deen, "Custom grasping: A region-based robotic grasping detection method in industrial cyber-physical systems," *TASE*, vol. 20, no. 1, pp. 88–100, 2023.

[11] K. Wada, S. James, and A. J. Davison, "Reorientbot: Learning object reorientation for specific-posed placement," in *ICRA*, 2022, pp. 8252–8258.

[12] C. Xie, A. Mousavian, Y. Xiang, and D. Fox, "Rice: Refining instance masks in cluttered environments with graph neural networks," in *CoRL*, 2022, pp. 1655–1665.

[13] A. G. Buch, L. Kiforenko, and D. Kraft, "Rotational subgroup voting and pose clustering for robust 3d object recognition," in *IEEE International Conference on Computer Vision*, 2017, pp. 4137–4145.

[14] W. Tao, X. Hua, K. Yu, X. Chen, and B. Zhao, "A pipeline for 3-d object recognition based on local shape description in cluttered scenes," *TGRS*, vol. 59, no. 1, pp. 801–816, 2020.

[15] P. Raj, L. Behera, and T. Sandhan, "Scalable and time-efficient bin-picking for unknown objects in dense clutter," *TASE*, vol. 21, no. 3, pp. 2289–2301, 2024.

[16] Z. Wu, Z. Wang, Z. Wei, Y. Wei, and H. Yan, "Smart explorer: Recognizing objects in dense clutter via interactive exploration," in *IROS*, 2022, pp. 6600–6607.

[17] N. A. Abd Rahman, K. S. M. Sahari, and N. A. Hamid, "An autonomous clutter inspection approach for radiological survey using mobile robot," *TASE*, vol. 20, no. 2, pp. 1212–1225, 2023.

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

[19] T. Novkovic, R. Pautrat, F. Furrer, M. Breyer, R. Siegwart, and J. Nieto, "Object finding in cluttered scenes using interactive perception," in *ICRA*, 2020, pp. 8338–8344.

[20] S. Roselli, F. Hagebring, S. Riazi, M. Fabian, and K. Åkesson, "On the use of equivalence classes for optimal and suboptimal bin packing and bin covering," *TASE*, vol. 18, no. 1, pp. 369–381, 2021.

[21] W. H. El-Ashmawi and D. S. Abd Elminaam, "A modified squirrel search algorithm based on improved best fit heuristic and operator strategy for bin packing problem," *Applied Soft Computing*, vol. 82, p. 105565, 2019.

[22] T. Tanaka, T. Kaneko, M. Sekine, V. Tangkaratt, and M. Sugiyama, "Simultaneous planning for item picking and placing by deep reinforcement learning," in *IROS*, 2020, pp. 9705–9711.

[23] A. G. Ramos, J. F. Oliveira, J. F. Gonçalves, and M. P. Lopes, "A container loading algorithm with static mechanical equilibrium stability constraints," *Transportation Research Part B: Methodological*, vol. 91, pp. 565–581, 2016.

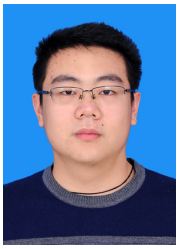
[24] F. Wang and K. Hauser, "Stable bin packing of non-convex 3d objects with a robot manipulator," in *International Conference on Robotics and Automation*, 2019, pp. 8698–8704.

[25] S. Huang, Z. Wang, J. Zhou, and J. Lu, "Planning irregular object packing via hierarchical reinforcement learning," *RAL*, vol. 8, no. 1, pp. 81–88, 2022.

[26] M. Schwarz, C. Lenz, G. M. García, S. Koo, A. S. Periyasamy, M. Schreiber, and S. Behnke, "Fast object learning and dual-arm coordination for cluttered stowing, picking, and packing," in *ICRA*. IEEE, 2018, pp. 3347–3354.

[27] N. Kitaev, I. Mordatch, S. Patil, and P. Abbeel, "Physics-based trajectory optimization for grasping in cluttered environments," in *ICRA*, 2015, pp. 3102–3109.

[28] Q. Lu, K. Chenna, B. Sundaralingam, and T. Hermans, "Planning multi-fingered grasps as probabilistic inference in a learned deep network," in *Robotics Research: The 18th International Symposium ISRR*, 2019, pp. 455–472.
[29] Y. Yang, H. Liang, and C. Choi, "A deep learning approach to grasping the invisible," *RAL*, vol. 5, no. 2, pp. 2232–2239, 2020.
[30] Z. Liu, Z. Wang, S. Huang, J. Zhou, and J. Lu, "Ge-grasp: Efficient target-oriented grasping in dense clutter," in *IROS*, 2022, pp. 1388–1395.
[31] K. Kleeberger, J. Schnitzler, M. U. Khalid, R. Bormann, W. Kraus, and M. F. Huber, "Precise object placement with pose distance estimations for different objects and grippers," in *IROS*, 2021, pp. 4639–4646.
[32] W. Goodwin, S. Vaze, I. Havoutis, and I. Posner, "Semantically grounded object matching for robust robotic scene rearrangement," in *ICRA*, 2022, pp. 11 138–11 144.
[33] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *ICCV*, 2017, pp. 2961–2969.
[34] M. Liu, R. Shi, K. Kuang, Y. Zhu, X. Li, S. Han, H. Cai, F. Porikli, and H. Su, "Openshape: Scaling up 3d shape representation towards open-world understanding," *NeurIPS*, vol. 36, 2024.
[35] T. Hester, M. Vecerik, O. Pietquin, M. Lanctot, T. Schaul, B. Piot, D. Horgan, J. Quan, A. Sendonaris, I. Osband, *et al.*, "Deep q-learning from demonstrations," in *AAAI*, vol. 32, no. 1, 2018.
[36] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *ECCV*. Springer, 2014, pp. 740–755.
[37] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
[38] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *MICCAI*, 2015, pp. 234–241.
[39] A. Goyal and J. Deng, "Packit: A virtual environment for geometric planning," in *ICML*, 2020, pp. 3700–3710.



Zhenyu Wu received his B.Eng. degree from the Beijing University of Posts and Telecommunications in 2021. He is currently a PhD student at the School of Intelligent Engineering and Automation, Beijing University of Posts and Telecommunications, China. His research interests include machine vision and intelligent robotics.

Ziwei Wang is currently an assistant professor in School of Electrical and Electronic Engineering, Nanyang Technological University. Before joining NTU, he is a postdoc fellow in Robotics Institute, Carnegie Mellon University, with Prof. Changliu Liu. He received the Ph.D and the B.S degrees from the Department of Automation, Tsinghua University in 2023 and the Department of Physics, Tsinghua University in 2018 respectively. His research goal is to design foundation models (FMs) for robotics including grounding FMs to the physical scene and deploying FMs in resource-limited robots. He has published over 30 scientific papers in TPAMI, IJCV, RAL, CVPR, ICCV, ECCV, NeurIPS, IROS and ICRA. He serves as a regular reviewer member for a variety of conferences and journals.

Sichao Huang received the B.S. degree in Automation from Tsinghua University, Beijing, China in 2020. He/She is currently working toward the M.S. degree in Automation with Tsinghua University, Beijing, China. His/Her research interests include reinforcement learning, computer vision and robot manipulation.

Zhan Liu received the B.Eng. degree in communication engineering from Beijing Jiaotong University in 2011 and the M.Eng. degree in communication engineering from Xi 'an Institute of Advanced Technology in 2013. He is currently working toward the Ph.D. degree at the Department of Automation, Tsinghua University. His research interests include computer vision and intelligent robotic applications.

Xiuwei Xu received the B.Eng degree from Tsinghua University in 2021. He is currently a PhD candidate in the Department of Automation at Tsinghua University. His research interests lie data/computation-efficient learning, 3D vision and robotic systems. He has published more than 10 scientific papers in TPAMI, TIP, CVPR, ECCV, NeurIPS, ICLR and IROS. He serves as a regular reviewer member for IJCV, TIP, TITS, TMM, CVPR, ICCV, ECCV, NeurIPS, ICML, ICLR, CoRL and IROS.

Haibin Yan (Senior Member, IEEE) received the B.Eng. and M.Eng. degrees from the Xi'an University of Technology, Xi'an, China, in 2004 and 2007, respectively, and the Ph.D. degree from the National University of Singapore, Singapore, in 2013, all in mechanical engineering. She is currently a full professor with the School of Intelligent Engineering and Automation, Beijing University of Posts and Telecommunications, Beijing, China. Her current research interests include human-robotic interaction, social robotics, industrial robot, and computer vision, where she has authored/co-authored more than 40 scientific papers in IEEE T-CYB/T-MM/T-CSVT/PR/CVPR/ICRA/IROS. She serves as the field Chair of international conferences ICME2022, VCIP2022, and ICME2020.

Jiwen Lu (Fellow, IEEE) received the BEng degree in mechanical engineering and the MEng degree in electrical engineering from the Xi'an University of Technology, Xi'an, China, in 2003 and 2006, respectively, and the PhD degree in electrical engineering from Nanyang Technological University, Singapore, in 2012. He is currently a full professor with the Department of Automation, Tsinghua University, Beijing, China. His current research interests include computer vision and pattern recognition, where he has authored/co-authored more than 160 scientific papers in PAMI/IJCV/CVPR/ICCV/ECCV. He serves as the co-editor-of-chief for Pattern Recognition Letters, an associate editor for IEEE Transactions on Image Processing, IEEE Transactions on Circuits and Systems for Video Technology,IEEE Transactions on Biometrics, Behavior, and Identity Science, and Pattern Recognition. He also serves as the General co-chair of ICME'2022, and the Program co-chair of FG'2023, VCIP'2022, AVSS'2021 and ICME'2020. He received the National Outstanding Youth Foundation of China Award. He is an IEEE/IAPR fellow.