Visuo-Tactile Feedback-Based Robot Manipulation for Object Packing

Wenyu Liang[1,*], Fen Fang[1,*], Cihan Acar[1], Wei Qi Toh[2], Ying Sun[1], Qianli Xu[1], and Yan Wu[1,2]

***Abstract*—Robots are increasingly expected to manipulate objects, of which properties have high perceptual uncertainty from any single sensory modality. This directly impacts successful object manipulation. Object packing is one of the challenging tasks in robot manipulation. In this work, a new visuo-tactile feedback-based manipulation planning framework for object packing is proposed, which makes use of the on-the-fly multi-sensory feedback and an attention-guided deep affordance model as perceptual states as well as a deep reinforcement learning (DRL) pipeline. Significantly, multiple sensory modalities, vision and touch [tactile and force/torque (F/T)], are employed in predicting and indicating the manipulable regions of multiple affordances (i.e., graspability and pushability) for objects with similar appearances but different intrinsic properties (e.g., mass distribution). To improve the manipulation efficiency, the DRL algorithm is trained to select the optimal actions for successful object manipulation. The proposed method is evaluated on both an open dataset and our collected dataset and demonstrated in the use case of the object packing task. The results show that the proposed method outperforms the existing methods and achieves better accuracy with much higher efficiency.**

***Index Terms*—Manipulation planning, force and tactile sensing.**

I. Introduction

OBJECT packing, especially dense box packing, is a challenging task for robotic systems [1]. It requires the robot to plan and execute a series of complex actions to manipulate yet semi-known objects in order to stow them firmly into a confined space for space optimization [2].

Generally, two key actions are essential in the object packing task, namely pick-and-place and push. While pick-and-place constitutes the bulk of the action space, grasping alone is often insufficient to accomplish the dense box packing task due to geometrical constraints as illustrated in Fig. 1 and/or limited graspability of the object. In such cases, pushing action can help to move an object to the target location or to align with one another. It is therefore beneficial for robots to be able to localize object properties for optimal manipulation results. Such locations can be inferred using the affordance learning approach [3]. However, in the cases where objects appear visually similar but have different intrinsic properties [such as weight, center of mass (CoM), mass/density distribution, surface properties, etc.], successful robotic object grasping and planar pushing can be challenging by using visual affordance alone. Moreover, the differences in interactions between the end-effector and the objects are difficult to observe visually without significant displacements. These in turn affect the accurate inference of the graspable/manipulable regions.

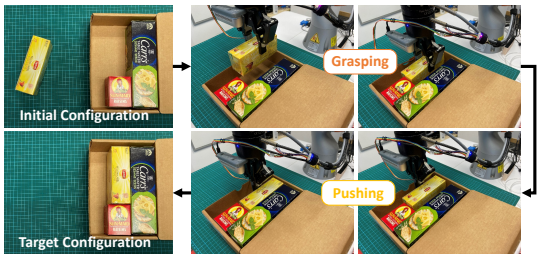


Fig. 1. Illustration of the object manipulation challenge for dense packing.

To address this challenge, physical interaction with objects is essential and the sense of touch can be used as a complementary sensory modality with vision. Such interaction allows the exploration of the object's intrinsic properties. In [4], force/torque (F/T) sensing is integrated into a deep grasp affordance prediction model to learn suction-based grasp affordance. Although it demonstrates enhancement in the success rate of suction-based object grasping, the proposed learning model works better for objects with CoM closer to their visual centroids potentially due to the dominance of visual percepts. Moreover, the F/T sensor only provides single-point information which may not be sufficient for affordance modelling of contact-rich manipulation, e.g., grasp and push. Tactile sensing, on the other hand, provides rich touch information that can be used to obtain a wide range of object-level information, e.g., localization, shape, mass, CoM, etc. [5].

While some state-of-the-art methods achieve good performance with the use of touch information, a considerable number of explorations are required to guarantee such accuracy. The need for a redundant number of explorations will lead to low efficiency and limited productivity. While efficiency is one important factor for robot manipulation, but most studies focus on prediction accuracy. In addition, little attention is placed on inferring multiple affordances with a unified model [6], [7].

To overcome the performance limitation of affordance inference and to improve the manipulation efficiency, an affordance-informed reinforcement-learning (RL)-based motion planning framework is proposed in this paper to construct and close the perception-action loop for object packing. An attention-guided multisensory multi-affordance learning model is designed to infer the manipulable regions from multiple sensory feedbacks. Here, an attention mechanism [8] is designed and employed to improve the performance of affordance inference in order to reduce the model's dependency on the quality of the support examples. Furthermore, the RL approach is proposed to select an optimal action for the next exploration

Manuscript received: September 16, 2022; Revised November 30, 2022; Accepted December 29, 2022. This paper was recommended for publication by Editor Hong Liu upon evaluation of the Associate Editor and Reviewers' comments. This work is supported by the Agency for Science, Technology and Research (A*STAR) through AME Programmatic Funding Scheme under Project #A18A2b0046.

* These two authors contributed equally to this work. *(Corresponding author: W. Liang*, e-mail: Liang_Wenyu@i2r.a-star.edu.sg)

[1] Institute for Infocomm Research, A*STAR, Singapore 138632.

[2] Institute of High Performance Computing, A*STAR, Singapore 138632.

Digital Object Identifier (DOI): see top of this page.

so that the number of explorations can be further minimized and thus the efficiency will be improved.

The main contributions of this paper include: (i) a manipulation planning framework with visuo-tactile feedback (especially the use of tactile information) that guides the robot to perform successful object manipulations; (ii) a unified deep multi-affordance learning model with a built-in attention mechanism to effectively encapsulate both vision and multimodal touch information of an object as an affordance map; (iii) the integration of multisensory information and affordance model with a deep RL (DRL) pipeline to significantly improve the efficiency of the manipulation motion planning.

The rest of this paper is organized as follows. Section II describes the related work, while Section III provides the definitions and robotic system description. Section IV details the proposed framework. Section V shows the experimental results. Finally, Section VI draws the conclusions.

II. Related Work

Affordance, first introduced by James Gibson [9], is a concept from psychology that describes the possibilities of an agent performing actions on an object [10]. Many researchers report the use of affordance in robotic manipulation. For example, the affordance models are successfully used in dexterous robotic grasping [3], vision-based robotic object pushing and grasping [6], contact point selection for vision-based planar pushing [11], and performing object positioning actions in a 2D space [12]. The affordance models have been shown to accomplish a range of difficult object manipulation tasks, but they are mainly based on visual information/perception.

In [13], a tactile-based grasp policy is introduced to improve robustness at a regrasping stage after performing visual-affordance-based grasping, where tactile information is only used as a corrective/reactive signal outside the affordance model. In [14], a regrasp planner with a slip detector using multi-sensing modalities is designed but limited for 1D grasp pose adjustment. In [15], a tactile-based rotation measurement and grasp-regrasp system is proposed to detect grasping failure and drive a robot to a stable grasp pose. These works show the successful applications of tactile sensing in robotic grasping.

In [16], [17], [18], some RL algorithms are successfully applied in robot grasping and/or pushing. Although these works do not focus on addressing challenges due to various intrinsic object properties, they demonstrate the benefits of using RL in robot manipulation.

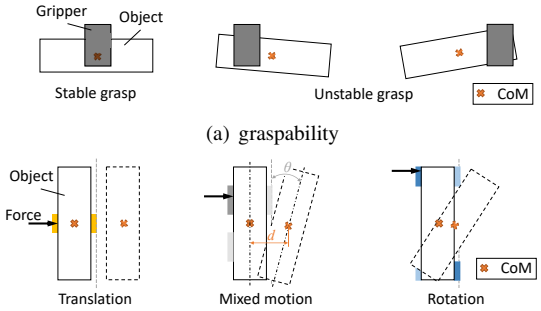
III. Definitions and System Description

A. Definitions of Affordances

The definitions of the affordances for graspability and pushabilities of an object are explained as follows.

Graspability: A stable grasp is defined as graspable. [see Fig. 2(a)].

Pushability: A set of three lower-level affordances is defined to describe the affordability of an object when it is pushed: (i) translational pushability; (ii) clockwise (CW) rotational pushability; and (iii) counter-clockwise (CCW) rotational pushability. Fig. 2(b) illustrates the pushability of an object.



(a) graspability

(b) pushabilities: translation, mixed motion, and rotation

Fig. 2. Illustrations of the definition of the affordances.

- *Translational Pushability:* A translational movement along the direction of a given push with consistent orientation is defined as translationally pushable [left of Fig. 2(b)]. The pushable zone (highlighted in yellow) is defined as the pushable surface of the object that only causes translation.

- *Rotational Pushability:* Although areas to both sides of the translationally pushable zone seem to be rotationally pushable, pushing some of these areas may result in both translation and rotation of the object [middle of Fig. 2(b)]. As such, a refined definition is needed. When rotational movement is caused by a given push of distance 3, if the translation distance of the object's CoM is within a threshold 3_{th} ($3_{th} = V3$, where V Ÿ 1 is a parameter to determine the threshold), the object is deemed rotationally pushable [right of Fig. 2(b)]. Qualified areas to the left and right of the translationally pushable zone are defined as CW and CCW rotationally pushable zones (highlighted in dark blue for CW and light blue for CCW), respectively.

B. Robotic System Description

Figure 3 shows the robotic system used in this work. This system consists of a KUKA LBR iiwa 14 R820 7-degrees-of-freedom (7-DoF) manipulator, a RealSense D435 eye-in-hand camera, a Robotiq FT 300 F/T sensor (6-axis measurement), a Robotiq 2F-85 2-finger gripper, and two XELA uSKin XR1944 tactile sensors mounted on the gripper's fingers. The tactile sensor has 16 taxels arranged in a 4-by-4 array and each taxel can detect the applied forces in the 3-axis (i.e., a total of 48 outputs from 1 tactile sensor).

IV. Framework Design

The overall object manipulation planning framework is depicted in Fig. 4, which mainly consists of a manipulation location classifier, a multi-affordance representation learning model,

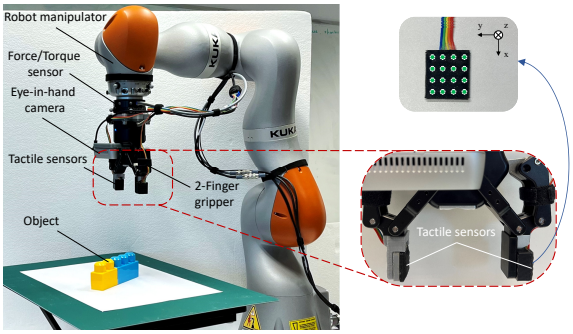


Fig. 3. Robotic system setup.

and an RL-based motion planning module. Both the location classifier and the affordance model are trained before they are used by the motion planning module. The location classifier is designed to digitize the possible manipulation locations (i.e., several labelled regions) on an object based on visual information, and its training is independent of the affordance model training. The output of the location classifier is the location map whose format is the digitalization of the location coding. As this is not a challenging task, to save inference time, the classifier can be trained with a shallow convolutional neural network (CNN), AlexNet [19], on our manually labelled image-position pairs, which provides the robot with a discrete action space in object manipulation. The affordance model is designed to predict the potential manipulable locations of a specific affordance, given an object with an unknown configuration and random orientation. The motion planning module is designed to plan the optimal action so that the end-effector can reach the desired manipulable location with minimum explorations/steps.

Significantly, the affordance model, fusing the vision and multimodal touch information, is used as one input state to the motion planning module. This affordance model predicts and highlights the manipulable information on top of the object image data for a specific affordance type, which allows faster learning convergence of the motion planning module. Note that with the motion planning module guiding the next explorations, the collected information is more meaningful and valuable than using a random exploration strategy. The proposed framework will improve the accuracy of affordance prediction and reduce the number of explorations needed for successful manipulation.

A. Attention-Guided Multi-Affordance Model

The objective of this model is to generate an affordance map $"$ via an end-to-end deep neural network (DNN). Here, the affordance map is represented by pixel-wise probability, i.e., the pixel/region associated with the highest value in the affordance map is considered to be the most suitable manipulation location given a query image $_q$ of an object and an affordance type $_q$.

Fundamentally, the proposed network is akin to a semantic segmentation network [20], which predicts a class for every pixel in an input image (a new image of the same object) via the extracted meaningful context from support explorations. Here, a manipulation location in an exploration/attempt is either a successful (positive) support exploration or a failure (negative) support exploration. Fig. 5 depicts the affordance model, which consists of the current query block (CQB), previous explorations block (PEB) and up-sampling block (USB). To overcome the challenges mentioned in Section I, achieve the purpose of multi-affordance prediction and further improve the affordance prediction performance, a branch to fuse the affordance type information is included in CQB and a channel attention block to allow the network to pay more attention to the manipulation region is designed in PEB. Thus, the proposed DNN is named "attention-guided network" (tAGN, where "t" is to emphasize the use of tactile information in the proposed framework).

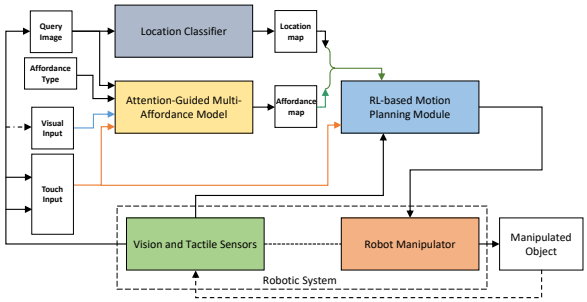


Fig. 4. Overall object manipulation planning framework.

1) Inputs and Output: For the model inputs, all the input images are resized to 88×88; the touch sensory information is standardized to have unit variance and zero mean; the affordance type is represented by a one-hot vector, namely, affordance vector. In this vector, only the $;^{th}$ entry is 1 which shows the indicated affordance type is the $;^{th}$ in the predefined affordance list given the current query. In each previous exploration, the manipulation location is labelled on an object image % by a circle with a radius of five pixels and different colors: red for failure and green for successful manipulations. Together with the sensory information of the previous attempts containing forces/torques $)$, tactile $)\,0$ and a Boolean variable > denoting the exploration outcome, the 8^{th} exploration is represented by 4_i={%$_i$- $)_i$-$)\,0_i$->$_i$}. For the output, an affordance map is outputted from the tAGN via the visualization by applying $\$\,!\,\$\,'\,"\ \ \%\_\ \ )$ wherefrom red to blue correspond to probability from high to low.

2) CQB: Given a query image of an object (obtained by the camera) and a requested affordance type, the CQB, which is a combination of a CNN and a multilayer perceptron (MLP), is used to extract and fuse visual features and affordance query information. The CNN has three convolutional layers, each layer has a filter followed by a max pooling layer. The outputs of CNN and MLP are concatenated and merged into a new feature vector with a dimension of 64. The vector is then repeated 11×11 times along 'height' and 'width' dimensions to form a 64×11×11 feature map.

3) PEB: This block consists of a CNN, a MLP and a channel attention module. It is noteworthy that the configurations of the CNN are the same as those of the CNN designed in CQB, but the *FC* layer is not connected directly after the CNN.

Remarkably, the images from the previous explorations are of different orientations, poses, and manipulable locations. The difference of manipulable locations on an object is the most important factor to affect the exploration result. Therefore, special attention on the locations' difference will help the model

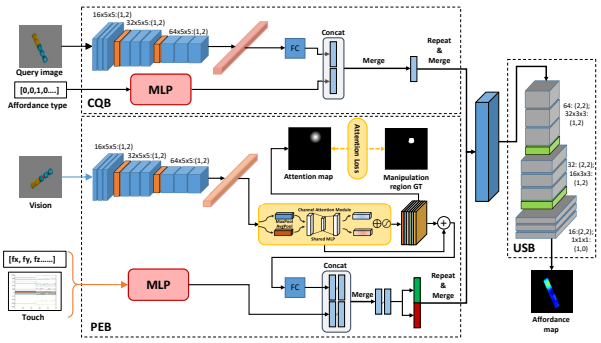


Fig. 5. Main architecture of the affordance model, tAGN: MLP is used to linearly convert the dimension of its input vector to 64, fully connected (*FC*) layer is used to generate a feature vector (dimension: 64×1) from a feature map.

to learn the best manipulation location on the current image, from experiences where different manipulable locations result in different exploration results. To this end, the feature map is fed into the channel attention module that is optimized by minimizing attention loss between the attention map and ground-truth manipulable region. Then after the attention module, the feature map is fed into a *FC* layer. For each previous exploration, the feature vectors from multisensory information will be then concatenated and merged into a new feature vector.

Following that, the new vectors of all previous explorations will be split into two groups, positive and negative, corresponding to the exploration results of success and failure, respectively. Subsequently, the average representation for each group, producing one positive and other negative representative vectors, is computed. These vectors are then repeated to form the feature maps with the dimension of 64×11×11.

4) USB: Three feature maps output from CQB and PEB are then concatenated and bi-linear interpolated [21] along concatenated dimension to finalize features for the subsequent USB. The USB consists of three groups of ConvTranspose and Conv2d layers. The ConvTranspose layer is an up-sampling layer which does not change the channel units but doubles the feature map size. The first two Conv2d layers are with filters which halve the channel unit but keep the feature map size constant. The last Conv2d layer has a filter and followed by a sigmoid layer to output a 2D 88×88 affordance map.

5) Attention Module Optimization: The objective function for the attention module is defined by (1). It is composed of a binary cross-entropy loss (on a single manipulation point of the current frame) and $:$ attention losses (on previous exploration manipulation areas) which are mean squared error (MSE) losses.

$$J = g\log(\ '') + (1-g)\log(1-\ '') + \_\frac{1}{\#}\sum_{i=1}^{\tilde{\mathbb{K}}}\sum_{j=1}^{\tilde{\mathbb{N}}}(H_{ij}-\mathcal{H}_{ij})^2 \text{-} \quad (1)$$

where g is the ground truth map of single manipulation points with a pixel value of either 0 (failed manipulation) or 1 (successful manipulation), while the remaining points are with values other than 1 or 0, $''$ is the predicted affordance map, $\#$ is the number of pixels in each map, $:$ is the number of prior explorations, H_{ij} and $\mathcal{H}_{ij}$ are observed and predicted value on the $\mathcal{g}^{th}$ pixel of $\mathcal{8}^{th}$ prior exploration, respectively, and $\_$ is the loss balancing parameter.

The tAGN weights are optimized using Adam optimizer [22] in the training process, the training batch size is 64, and the learning rate is set as 0.0005 while the weight decay is 0.0002. The model training epoch is 1000. Data augmentations are implemented for training on the visual inputs for both CQB and PEB, e.g., horizontal and vertical flips.

B. Deep Reinforcement Learning-Based Motion Planning

To further reduce the number of explorations needed for a successful manipulation, RL is an effective way to find an optimal policy which tells the robot manipulator to take an action that can maximize the chance of successful manipulation. Essentially, it is to solve a motion planning problem.

1) Motion Planning Problem: As some physical properties of an object such as CoM are hidden from vision but observable through touch, in our work, our DRL state representation is a concatenation of visual and multimodal touch information. However, the touch information is not observable at the time of decision-making. The state-action pair here is stochastic, and the motion planning problem in this paper can be described as a partially observable Markov decision process (POMDP) [23]. The process is defined by the functions of $(($, $\ $, $\mathcal{T}$, $'$, $\$$). $($ is the collection of states that is the combination of the predicted affordance map from the affordance model, the location map from the location classifier and the tactile data. Before proceeding to the motion planning module, the hot region in the affordance map is roughly digitized as an index number with the location map, and then this number is used as the potential manipulation location. $\ $ is the set of allowed actions (i.e., movements to the manipulation locations) given a state. $\mathcal{T}$ is the transition function from the current state (B_q) to the next state (B_{q+1}) with action (O_q) at the current state. $'$ is the scalar reward computation given a state-action pair (B, O). $\$$ is the tactile sensor feedback computation function.

Remarkably, the tactile sensor input is computed as the temporal difference of three successive tactile information. The problem in the POMDP is that the current actions affect the next states and the future rewards, this can be solved by using deep reinforcement learning (DRL) algorithms.

2) Deep Q-Learning Network (DQN): In this work, the value-based algorithm, namely, DQN is employed to train the agent (i.e., the robot manipulator), and the Q-value update policy with learning rate $\mathbb{W}$ is expressed by

$$\&(B\text{-}O) = \&(B\text{-}O) + U[A + \mathbb{W}\max_{a'}\&(B'\text{-}O') - \&(B\text{-}O)]\bullet \quad (2)$$

Given the current affordance map and touch information, the agent needs to predict two actions, moving direction with an action space of eight (*left, right, up, down, left-up, left-down, right-up, and right-down*) and step size with an action space of five (from one interval to five intervals), to localize the successful manipulation location within shortest steps. An agent training process is designed to make the optimal prediction on the two actions simultaneously, and thus the output $\&$ of the DQN contains three items:

$$\begin{aligned}\&(B\text{-}O;\backslash) = +(B;\backslash^f\text{-}\backslash^s) + \&^{md}(B\text{-}O^{md};\backslash^f\text{-}\backslash^{md}) \\ + \&^{ms}(B\text{-}O^{ms};\backslash^f\text{-}\backslash^{ms})\text{-}\end{aligned} \quad (3)$$

where $\&$ is the combination of the state value $+$, the moving direction Q-value $\&^{md}$, and the moving step size Q-value $\&^{ms}$. Here, $+$ has a terminology-state value [24]. The three items can be called sub-items which share the same feature extractor (similar to the one used in the CQB and PEB of AGN) and inputs, and the $\backslash^f$ is the parameters of the feature extractor, $\backslash^s$, $\backslash^{md}$, and $\backslash^{ms}$ are parameters of the three branches. $\backslash$ is a collection of these parameters where $\backslash = \{\backslash^f\text{-}\ \backslash^s\text{-}\ \backslash^{md}\text{-}\ \backslash^{ms}\}$.

Figure 6 shows the architecture of the DRL-based motion planning module with the proposed overall framework. The DRL agent (i.e., robotic system) is optimized in the training process by minimizing the difference between the so-called current Q-value

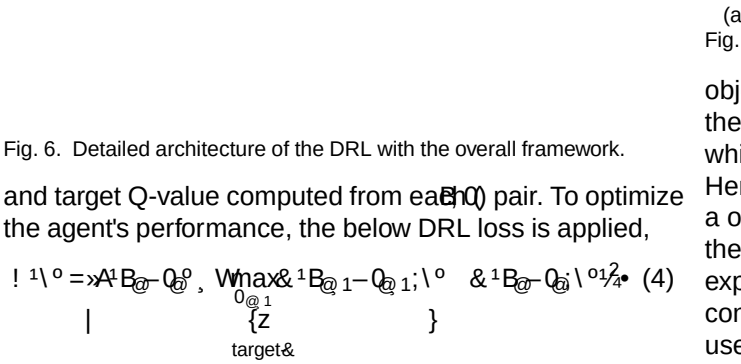


Fig. 6. Detailed architecture of the DRL with the overall framework.

and target Q-value computed from each (s,a) pair. To optimize the agent's performance, the below DRL loss is applied,

$$L_i(\theta_i) = \mathbb{E}_{(s,a,r,s')\sim U(D)}\left[\left(r + \gamma \max_{a'} Q(s',a';\theta_i^-) - Q(s,a;\theta_i)\right)^2\right] \quad (4)$$

where the target is $r + \gamma \max_{a'} Q(s',a';\theta_i^-)$.

Once an action is determined at a certain state, the reward after moving t steps can be determined by

$$R(s,a) = \begin{cases} 1 - \frac{1}{1+t^2} & \text{if success manipulation location found,} \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

Remarkably, the main idea for the reward function design is to encourage the agent to make an optimal decision on two actions in the shortest steps. To achieve the optimal manipulation location, the agent may need to make multiple decisions. Putting t in the denominator is to ensure the action sequence from the agent is optimal (i.e., shortest steps).

After training, the parameters are optimized, the approximate action decision given the current state can be obtained by

$$a^* = \arg\max_{a} Q(s,a;\theta^*) \quad (6)$$

The output of the DRL algorithm is the action decision according to an object's a ordance map and the tactile feedback.

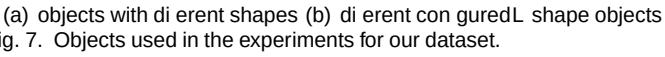
V. Experiments and Results

To verify and validate the e ectiveness of the proposed framework, several experiments are conducted on an open dataset named \YCBUSR" [4] and our dataset that was collected using the robotic system shown in Fig. 3. The use case on object packing is also demonstrated with this robotic system.

A. Experimental Setup

The open dataset is a suction-based object grasping dataset that contains nine classes of objects, with a total of 2868 samples. Also, the model proposed for the YCBUSR dataset, namely, deep a ordance prediction (DAP) model, is also used in the experiments for comparison purposes.

Further on, our dataset is collected on a robot manipulator with a 2- nger gripper instead of a suction cup. Such a robotic system o ers more DoF for object manipulation and thus our dataset includes data for both grasping and pushing. However, unlike the suction cup that is able to make contact with a large surface of the object, grippers can only manipulate certain parts of the object. This results in the collected samples for each



(a) objects with di erent shapes (b) di erent con gured L shape objects

Fig. 7. Objects used in the experiments for our dataset.

object being sparse in the 2- nger gripper scenario. As such, the average grasp success rate of our grasping setup is 17.37%, which is much lower than that of the YCBUSR dataset, 47.13%. Hence, our dataset provides a more challenging scenario for a ordance learning and manipulation planning. Signi cantly, these success rates can be considered as performance under zero exploration. In our dataset, ve objects with di erent shapes constructed by combinations of di erent building blocks are used to represent the real-world objects (as shown in Fig. 7). They are labeled as (long stick), (short stick), T, L, and X (cross). Each object has three di erent con gurations of mass distributions by adding the hidden weights into the hollows of the blocks. Fig. 7(b) shows example con gurations of the shape object. In total, 2442 and 3234 samples in grasping and pushing, respectively, are collected and used.

On both datasets, the performance of the a ordance prediction to be evaluated are: (i) how well a model can predict a ordances if it has seen an object's appearance before but has no knowledge about a particular mass distribution; and (ii) how well a model can generalize knowledge from seen objects to unknown objects (\seen" and \unseen", respectively hereafter). All the models are trained using the same datasets. Five-fold cross-validation is used in all evaluations. The mean and standard deviation (SD) of the area under the receiver-operating characteristic curve (AUROC) across all folds of cross-validation are selected as the metrics for evaluating the model performance. After the evaluation of the a ordance prediction, the overall framework (i.e., DRL with a ordance model) is evaluated on our dataset. Training and evaluation are carried out using PyTorch [25] on a workstation with an NVIDIA GeForce GTX 1080 GPU.

B. Experiments on A ordance Model

1) Results on Suction (YCBUSR) Dataset: As only image and F/T data (i.e., no tactile data) are used in the YCBUSR dataset, the input channels of the proposed tAGN are modi ed (i.e., removes the tactile input) to t the YCBUSR dataset without changing its main architecture. The modi ed model is named as AGN, which is applied to the YCBUSR dataset as well as our dataset (as another comparison model). The results on the YCBUSR dataset are listed in Table I. It is clear that the proposed AGN obtains higher AUROC than the DAP model in both \seen" and \unseen" settings, regardless of the number of explorations.

TABLE I
Comparison results (in AUROC: mean sd) on YCBUSR dataset

	DAP		AGN	
# of explorations	4	5	4	5
seen	0.903 0.03	0.910 0.03	0.913 0.03	0.938 0.03
unseen	0.898 0.02	0.905 0.02	0.907 0.03	0.930 0.03

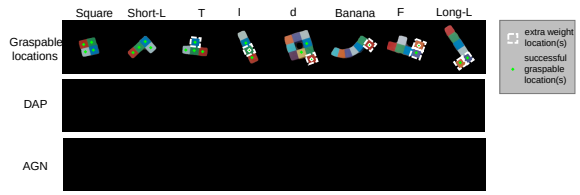


Fig. 8. Sample affordances predicted with five previous explorations.

Sample affordances predictions on the YCBUSR dataset can be found in Fig. 8. The masses of *(@DOA4* and *(>AC – !* shape objects are uniform and all previous explorations are successful grasp. Thus, almost all pixels on the affordance maps predicted by both models are in red or yellow (relatively high probability). The *)* , , *3*, and *0=0=0* shape objects contain one cell with extra weight. For the *)* shape, the extra weight is on its symmetry axis, therefore both models are able to generate sensible affordance maps easily. However, for the other three shapes, the cell with extra weight is biased to one side of the object. In such cases, the affordance map generated from the AGN is more convincing. The and *! >=6 – !* shape objects each contain two cells with extra weights biased to one end of the object, which make their mass distribution even more non-uniform and non-symmetric. Affordance map generation in this situation is very challenging, which needs the model to pay more attentions to the previous explorations. Both results shown in the table and figure clearly show that the proposed AGN model outperforms the DAP model, which indicates that the attention module in the PEB improves performance.

It is also noteworthy that the accuracy on both "seen" and "unseen" objects using the AGN with four explorations are equivalent to those using the DAP model with five explorations. In other words, the proposed AGN requires fewer explorations than the DAP model to achieve a competitively high accuracy (e.g., ¡ 0•9). It implies that the proposed AGN has the potential to use less number of explorations and thus improve efficiency.

2) Results on Grasping and Pushing (Our) Dataset:
- Experimental Results on Object Grasping: Evaluation results on our dataset are first compared among the DAP, AGN and tAGN models based on "seen" and "unseen" settings while the number of explorations is set as five. The results are tabulated in Table II. Significantly, one difference between the tAGN model and the other two models is that the tactile data on top of the F/T data is input to the tAGN model.

As listed in the table, the AUROC on our dataset for both the DAP and AGN models are lower than that on the YCBUSR dataset. This is reasonable because our dataset is more challenging than the YCBUSR dataset as mentioned previously. Nevertheless, consistent improvement in results can be observed for both using the proposed AGN model against the DAP model and using additional tactile inputs against only single-point F/T data. It is evident that the tactile sensing information can definitely improve the model prediction accuracy in comparison to those without the tactile data. Moreover, some tests are carried

TABLE II
COMPARISON RESULTS (IN AUROC: MEAN±SD) ON OUR GRASPING DATASET

	DAP	AGN	tAGN
seen	0.743±0.03	0.768±0.03	**0.792±0.03**
unseen	0.721±0.04	0.754±0.03	**0.769±0.03**

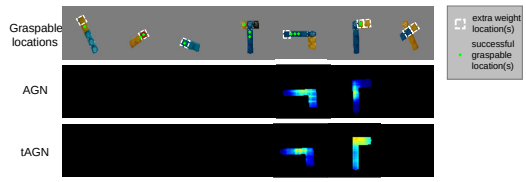


Fig. 9. Sample grasp affordance maps generated on our dataset.

out using the proposed tAGN model trained only with vision and tactile data, which show results of 0•803 ± 0•04 and 0•756 ± 0•04 (in AUROC) on "seen" and "unseen" objects, respectively. These imply that tactile data provides more information on affordance than F/T data.

Sample grasp affordance predictions on our dataset can be found in Fig. 9. It is obvious that tAGN makes more accurate predictions on the appropriate areas for stable grasping. It is also worth noting that the **X** shape object with the configuration shown cannot be stably grasped at all (i.e., not graspable) due to the limitation of the gripper aperture. Interestingly, tAGN gives a very accurate prediction in this case. This helps inform the robot to plan for other actions in place of graspability to successfully manipulate the object.

- Experimental Results on Object Pushing: Comparison results of the push affordance prediction on our dataset are listed in Table III. It can be clearly observed that tAGN performs much better than AGN. This can be mainly attributed to the fact that tactile inputs can provide not only more information about the contact force than single-point F/T inputs but also extra information about contact configuration (e.g. friction distribution in this case, stress distribution, contact surface area). The tactile sensor is much better at gathering distributed contact information than the F/T sensor and thus including the tactile sensing information in the AGN can greatly improve the accuracy and robustness of the push affordance prediction.

Figure 10 shows sample push affordance predictions on our dataset using tAGN. As can be seen, the proposed model can well predict the corresponding area for each affordance. By combining with previous results, it can be found that the **X** shape object is pushable but not graspable. The unified multi-affordance representation model can thus help the robot to replan the task sequences for object manipulation (e.g., moving an

TABLE III
COMPARISON RESULTS (IN AUROC: MEAN±SD) ON OUR PUSHING DATASE)

		AGN	tAGN
Translation	seen	0.448±0.03	**0.863±0.03**
	unseen	0.431±0.04	**0.846±0.03**
Rotate_CW	seen	0.512 ±0.04	**0.845±0.03**
	unseen	0.497±0.03	**0.812±0.04**
Rotate_CCW	seen	0.491±0.03	**0.843±0.04**
	unseen	0.462±0.03	**0.819±0.04**

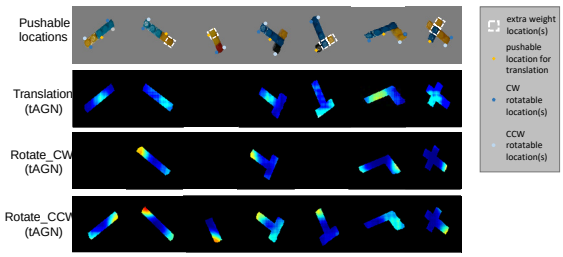
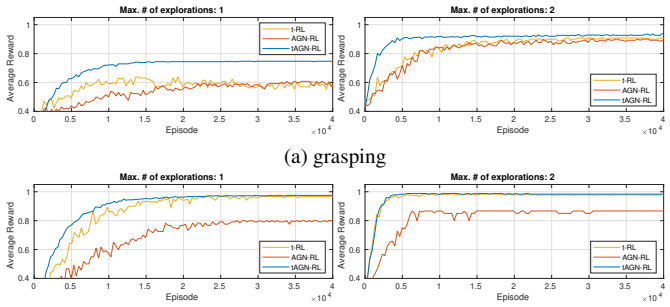


Fig. 10. Sample affordances predicted on our dataset for pushing.



(a) grasping

(b) pushing

Fig. 11. Training process of different RL frameworks on different actions (with different numbers of explorations, left: 1, right: 2).

object) if the default action for this task is not affordable.

C. Experiments on the Overall Framework

The proposed overall framework, the tAGN model plus DRL-based motion planning module, (namely, tAGN-RL), is applied to our datasets of both object grasping and pushing. For the object pushing, the performance of the overall framework on translational push is evaluated as it is more frequently used in the packing task mentioned in Section I. For comparison purposes, two frameworks with similar DRL architectures but different inputs are implemented and tested. These two frameworks are (i) touch-based DRL (without the use of affordance map), namely, t-RL; (ii) AGN plus the DRL-based motion planning (without the use of tactile feedback, i.e., only the vision and F/T information are used), namely, AGN-RL. It is noteworthy that the t-RL can be essentially considered as a kind of end-to-end RL as the touch information serves as the state and is directly input to the RL model. Also, the frameworks without the RL/DRL are used in the comparison, which are DAP, AGN, and tAGN.

The training processes of different DRL-based frameworks are shown in Fig. 11, where the n-greedy exploration policy is used. Obviously, all the DRL-based frameworks converge to higher rewards as the number of explorations increases. This also matches the facts found in the affordance model learning, i.e., more explorations are beneficial to better accuracy. By comparing AGN-RL with tAGN-RL, it is evident that not only the convergence speeds are slower but also the convergence values are smaller using the AGN-RL, especially on the pushing task. This is because the tactile sensor captures far richer contact information comprehensively, which is helpful in both affordance prediction and action selection. By comparing t-RL with tAGN-RL, it can be found that the tAGN-RL converges much faster in the grasping task and slightly faster in the pushing task. It also shows that higher rewards can be obtained using the tAGN-RL, especially on the grasping dataset with one exploration. These reveal that the generated affordance map by the tAGN with the fusion of the vision and touch information provides useful summarized information for the system to learn the optimal policy in a shorter time. It should be also noted that the AGN-RL achieves comparable convergence speeds and rewards to the ones of t-RL on the grasping task. This indicates that the affordance map can compensate for the limited single-point F/T information (in AGN-RL) to match the rich touch information provided by the tactile sensor (in t-RL).

TABLE IV
COMPARISON RESULTS ON OUR DATASET WITH MOTION PLANNING

# of explorations	DAP	AGN	tAGN	t-RL	AGN-RL	tAGN-RL
	Object grasping success rate (%)					
1	44.48	46.13	50.93	53.57	52.31	**60.77**
2	47.52	51.57	55.32	76.96	76.23	**83.08**
5	71.94	74.37	76.22	90.26	84.20	**91.64**
	Object translational pushing success rate (%)					
1	-	37.58	72.24	80.50	62.50	**82.50**
2	-	39.22	75.63	85.50	76.04	**88.54**
5	-	43.41	82.64	99.00	87.16	**100.00**

After the RL models are well-trained, the frameworks are tested on all five objects with different configurations on both grasping and translational pushing. Here, the frameworks without the RL-based module but a simple motion planning module directly utilizing the affordance model (DAP, AGN or tAGN) outputs are used in the comparison study. The success rates of using different frameworks are listed in Table IV. It can be found that higher success rates can be achieved as the number of explorations increases.

Moreover, it is obvious that the tAGN-RL achieves the best success rate on both grasping and translational pushing with two explorations among all the frameworks. By comparing with the frameworks without the use of RL (i.e., DAP, AGN, tAGN), the tAGN-RL shows great improvements in the success rates when they are using the same numbers of explorations (e.g., at least 27% improvement on grasping and 12% improvement on pushing for the two-exploration cases). Also, it is worth noting that the tAGN-RL can use just two explorations to achieve higher success rates (at least 6.8% and 5.9% improvements in grasping and pushing, respectively) than those frameworks without RL but using five explorations. It verifies the efficiency improvement of using RL. In summary, the proposed framework can accomplish tasks with high success rates and high efficiency.

D. Experiments on Real-World Objects

The proposed overall framework is tested on three real-world objects: tea bag box, toothpaste box, and massage gun (see Fig. 12). In the experiments, the tea bag box is set with two configurations: a full box of tea bags and a half box of tea bags at the side. The toothpaste box is packed with a toothpaste inside and its CoM is not in the geometric center of the box due to the non-symmetric shape of the toothpaste although the toothpaste

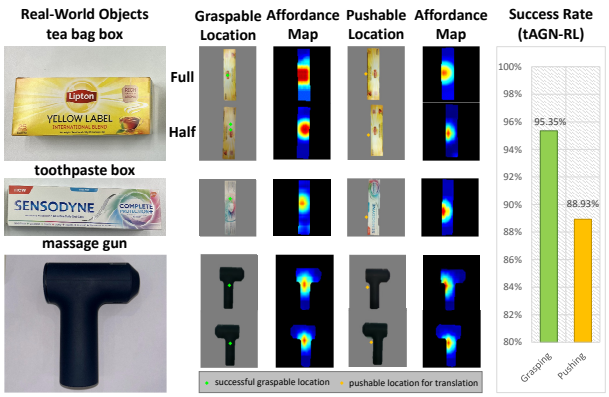


Fig. 12. Experimental results on real-world objects via the proposed framework.

box is visually symmetric. The massage gun has an internal movable component and is roughly an L-shaped object.

From Fig. 12, it is clear that the proposed tAGN can predict the object affordance correctly and the system can achieve the stable grasp and translational push tasks with high success rates over 95% and 85%, respectively while setting 2 as the number of explorations. It is evident that the results on the real-world objects match the ones on the building-block objects. The better grasping success rate on real-world objects can be attributed to the simpler shapes of real-world objects. In summary, it indicates that the proposed tAGN-RL can work on real-world objects well.

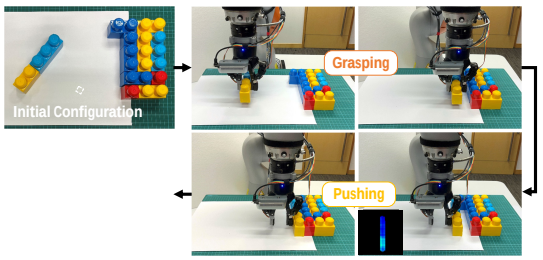
E. Use Case on Object Manipulation

In this use case, it is assumed that the action sequences and task plans are already known in advance, but the desired manipulation locations for grasping and pushing are needed to be determined by the proposed framework. It should be noted that object placement planning is not in the scope of this work.

Figure 13 shows the sequence of actions in two examples for successful object manipulation in a packing task. The object in each example can first be grasped stably based on the affordance model outputs and tactile feedback and then moved. Finally, the object can be placed tightly with another object as expected (i.e., matches the target image) with pushing.

VI. Conclusion

In this paper, a multisensory-based object manipulation planning framework using multi-affordance model and RL was developed for manipulation tasks of objects with unknown intrinsic properties (e.g., CoM, mass distribution). The integrated vision and multimodal touch information with the attention mechanism is proposed to improve the accuracy and robustness of the affordance prediction model. To further improve the accuracy and efficiency, the learned affordance model is integrated into a DRL-based motion planning pipeline. The proposed framework is then tested on the open (YCBUSR) dataset and our collected dataset. The results on both datasets show that the proposed method achieves better accuracy, especially for the push affordance prediction. It can be also concluded from the results that the proposed framework can effectively achieve the manipulation tasks with a high success rate and high efficiency. However, there are two limitations of the proposed method: (i) the generalization to new objects with very different shapes may be limited; (ii) the real-world objects used in this paper are with a limited range of shapes. Therefore, the possible directions for future work are (i) self-supervised learning with geometric knowledge embedding for further improvement on generalization capability; and (ii) training and investigation on more varieties of real-world objects.



(a) on building-block objects

(b) on real-world objects with half-filled tea bag box

Fig. 13. Robotic object manipulation using the proposed framework.

References

[1] S. Dong and A. Rodriguez, "Tactile-based insertion for dense box-packing," in *IROS 2019*, 2019, pp. 7953–7960.
[2] F. Wang and K. Hauser, "Dense robotic packing of irregular and novel 3-D objects," *IEEE Trans. Robot.*, pp. 1–14, 2021.
[3] P. Mandikal and G. Kristen, "Learning dexterous grasping with object-centric visual affordances," in *ICRA 2021*, 2021.
[4] M. Veres, I. Cabral, and M. Moussa, "Incorporating object intrinsic features within deep grasp affordance prediction," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6009–6016, 2020.
[5] Q. Li, O. Kroemer, Z. Su, F. F. Veiga, M. Kaboli, and H. J. Ritter, "A review of tactile information: Perception and action through touch," *IEEE Trans. on Robot.*, vol. 36, no. 6, pp. 1619–1634, 2020.
[6] H. Wu *et al.*, "Learning affordance space in physical world for vision-based robotic object manipulation," in *ICRA 2020*, 2020, pp. 4652–4658.
[7] A. Zeng *et al.*, "Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching," *Int. J. of Robot. Res.*, pp. 1–16, 2019.
[8] A. Vaswani *et al.*, "Attention is all you need," in *NIPS'17*, 2017, p. 6000–6010.
[9] J. J. Gibson, *The Senses Considered as Perceptual Systems*. Boston: Houghton Mifflin, 1966.
[10] P. Ardón *et al.*, "Building affordance relations for robotic agents - A review," in *IJCAI-21*, 2021, pp. 4302–4311.
[11] A. Kloss *et al.*, "Accurate vision-based manipulation through contact reasoning," in *ICRA 2020*. IEEE, 2020, pp. 6738–6744.
[12] T. Hermans, J. M. Rehg, and A. F. Bobick, "Decoupling behavior, perception, and control for autonomous learning of affordances," in *ICRA 2013*, 2013, pp. 4989–4996.
[13] F. R. Hogan *et al.*, "Tactile regrasp: Grasp adjustments via simulated tactile transformations," in *IROS 2018*, 2018, pp. 2963–2970.
[14] Q. Feng *et al.*, "Center-of-Mass-Based robust grasp planning for unknown objects using tactile-visual sensors," in *ICRA 2020*. IEEE, 2020, pp. 610–617.
[15] R. Kolamuri *et al.*, "Improving grasp stability with rotation measurement from tactile sensing," in *IROS 2021*. IEEE, 2021, pp. 6809–6816.
[16] A. Boularias, J. A. Bagnell, and A. Stentz, "Learning to manipulate unknown objects in clutter by reinforcement," in *AAAI-15*, 2015.
[17] D. Kalashnikov *et al.*, "Scalable deep reinforcement learning for vision-based robotic manipulation," in *Conference on Robot Learning*. PMLR, 2018, pp. 651–673.
[18] A. Zeng *et al.*, "Learning synergies between pushing and grasping with self-supervised deep reinforcement learning," in *IROS 2018*. IEEE, 2018, pp. 4238–4245.
[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun. of the ACM*, vol. 60, pp. 84 – 90, 2012.
[20] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR2015*, 2015, pp. 3431–3440.
[21] P. Smith, "Bilinear interpolation of digital images," *Ultramicroscopy*, vol. 6, no. 1, pp. 201–204, 1981.
[22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR 2015*, 2015.
[23] S. Brechtel, T. Gindele, and R. Dillmann, "Probabilistic decision-making under uncertainty for autonomous driving using continuous pomdps," *ITSC 2014*, pp. 392–399, 2014.
[24] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *AAAI-16*, vol. 30, no. 1, 2016.
[25] A. Paszke *et al.*, "Automatic differentiation in pytorch," in *NIPS'17*, 2017.