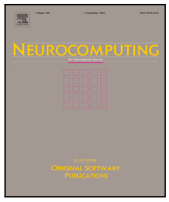
Neurocomputing 578 (2024) 127386

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



GAM: General affordance-based manipulation for contact-rich object disentangling tasks

Xintong Yang [a], Jing Wu [b], Yu-Kun Lai [b], Ze Ji [a,*]

[a] School of Engineering, Cardiff University, Queen's Buildings, The Parade, Cardiff, CF24 3AA, Wales, UK
[b] School of Computer Science and Informatics, Cardiff University, Abacws, Senghennydd Road, Cathays, Cardiff, CF24 4AG, Wales, UK

ARTICLE INFO

Communicated by Y. Cheng

Dataset link: 10.5281/zenodo.10654542

Keywords:
Contact-rich manipulation
Affordance theory
Reinforcement learning
Task-oriented grasping
Grasp filter

ABSTRACT

Picking up an entangled object is a difficult manipulation task due to its rich contact dynamics. Most existing solutions fail to produce grasp poses to enable reliable manipulation due to the dependence on simplified assumptions for the motion policies. Grasps generated by these methods tend to drop objects or cause undesired movements of non-grasped objects. To improve such object-disentangling tasks, we propose to extend the concept of reinforcement learning (RL)-based affordance to include arbitrary action consequences and implement a general affordance-based manipulation (GAM) framework.

In the GAM, we train an RL agent that uses more fine-grained actions and outperforms previous methods with a smaller chance of dropping objects and making contact with non-grasped hooks. Then, a manipulation affordance prediction (MAP) model is trained to estimate the performances of the RL agent. Finally, the manipulation affordance-based grasp filter (MAGF) selects grasp poses that afford the desired manipulation performances, showing substantial improvements in five challenging hook disentangling tasks in simulation. The experiments show (1) the limitation of TAG generators, (2) the effectiveness of filtering TAGs with predicted manipulation performances based on the general affordance theory, and (3) the importance of avoiding contact with non-grasped objects in contact-rich manipulation.

1. Introduction

Robot manipulation is one of the key research problems behind the automation of manufacturing, agriculture, construction, etc. [1]. This paper studies one of the challenging manipulation tasks of separating tangled objects, as shown in Fig. 1. Object entanglement can occur to U/C/S-shaped or similar objects in various scenarios, such as the mass manufacturing processes of such objects, assembly/disassembly processes of components of various shapes, daily uses of objects in irregular shapes, etc.

Automating such disentangling manipulation is challenging because of the intractably complex contact dynamics among objects during the manipulation process. It is unintuitive how to minimise the influences on surrounding objects and the chances of dropping the grasped object. A successful and failed example of separating a C-shape hook in simulation is shown in Fig. 2.

Existing solutions to such tasks [1–6] are commonly comprised of a task-agnostic grasp (TAG) pose estimator and a manipulation trajectory generator. They works rely on TAG poses and simple straight-up lifting motions that cannot guarantee separation in all situations. They (1) fail in cases where a straight-up lifting motion is insufficient, (2) fail to reduce disturbance to other objects, and (3) do not consider the link between grasp pose generation and the following manipulation task. In addition, TAG poses are not always suitable for manipulating tangled objects, as the choice of grasping poses highly depends on the downstream manipulation tasks [7,8].

To address these difficulties, this paper proposes a new manipulation framework based on a novel theory of general affordance, which provides a mathematical model for learning arbitrary action consequences and whether actions are affordable in terms of a particular metric or a combination of metrics. According to Gibson [9], The affordance of an object contains the knowledge of what can be done with the object by an agent [10]. In regard to the disentangling task, we seek to select TAGs based on the prediction of whether a TAG (action) can afford certain manipulation performances (consequences), such as specific values of success rate, object dropping rate and/or averaged movements of other objects.

Theoretically, we adopt the RL-based affordance definition [11] as the basis of our manipulation framework. Khetarpal et al. [11] define affordance as the set of state–action pairs in an MDP where the intents of the agent can be achieved. Here, the intent is the desired action

* Corresponding author.
E-mail address: jiz1@cardiff.ac.uk (Z. Ji).

https://doi.org/10.1016/j.neucom.2024.127386
Received 3 October 2023; Received in revised form 20 December 2023; Accepted 3 February 2024
Available online 7 February 2024

Fig. 1. Example objects that could become tangled.

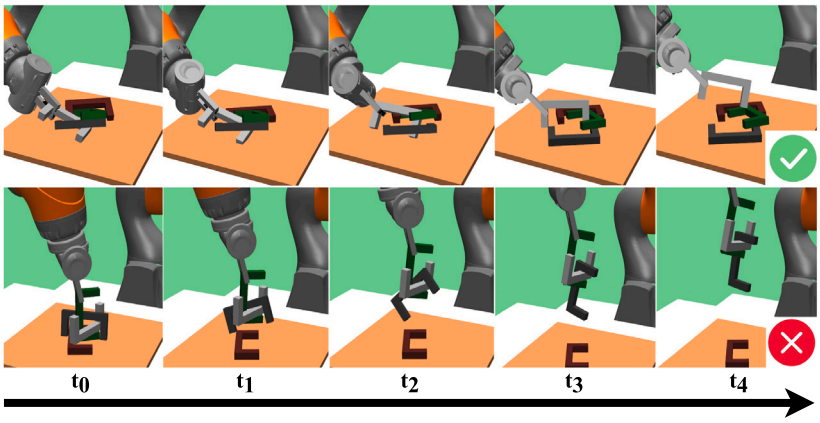


Fig. 2. Succeeded and failed examples of disentangling a hook in one of the simulated tasks in this paper. In the first row, the robot manages to rotate and lift up only the white hook.

consequence, and in [11] it is limited to the desired future state distribution. This paper extends it to include arbitrary action consequences, resulting in the concept of *general intent* and *general affordance*. The new definition allows the agent to select actions based on a broader range of action consequences. With the new theory, we propose the general affordance-based manipulation (GAM) framework that is able to select TAGs that afford the desired manipulation outcomes.

In addition, this paper applies reinforcement learning (RL) to generate more complex manipulation behaviours that separate a grasped object with minimised non-grasped object movements. This is because classic motion planners perform poorly in such contact-rich manipulation tasks, due to the difficulty of obtaining an accurate model of the object interaction processes [12], while RL policies can produce complex motions for manipulators without such dynamic models [4, 13].

A four-step approach is designed to implement GAM. First, we collect a set of 6 degree-of-freedom TAG poses and the latent features of the scene's point clouds associated with them using an off-the-shelf deep learning-based grasp estimator (GraspNet [7]). Secondly, we train a manipulation policy (deep double Q learning [14]) to separate the grasped hook with these TAGs. Thirdly, we train a manipulation affordance prediction (MAP) model to map the TAG poses and their latent point-cloud features to the performances achieved by the RL policy. Finally, we use the MAP model to select TAG poses with three grasp filtering strategies. Experiments demonstrate substantial improvements over baselines in five object disentangling tasks in simulation.[1]

This paper is organised as follows. Section 2 discusses related works. Section 3 proposes the general affordance theory. Section 4 introduces our implementation of GAM in detail. Section 5 gives experiment and training details, followed by experimental results and discussions in Section 6. Finally, Section 7 concludes the work.

[1] Codes are available at https://github.com/IanYangChina/GAM-paper-codes/.

2. Related work

Object manipulation has been a very interesting and important, but also challenging area of robotic research [1]. Recently, more researchers have started to study the link between grasping and manipulation, because deep learning-based task-agnostic grasp generators have achieved advanced performance [15] and manipulation is inevitably dependent on grasping [1].

Our work studies one of the difficult manipulation problems, disentangling objects, which, despite its value in daily life and industries, is a rather underdeveloped research area. Previously, Matsumura et al. [2] developed the first solution to picking up an object from a pile of potentially tangled objects. They proposed a Convolutional Neural Network (CNN) to predict whether or not a top-down grasp pose will result in picking up several tangled objects and used it as a filter to avoid picking from such grasp poses [2]. Similarly, Moosmann et al. [3] trained a CNN to predict whether an object is free from entanglement during a straight lifting-up motion and avoid grasping entangled objects [3]. The same team further developed supervised [16] and reinforcement learning [4] approaches to manipulating and separating entangled objects given task-agnostic grasp poses. Another recent work proposed a topological solution to compute an entanglement score from a depth image and thus find top-down grasp poses that are free from entanglement [6]. Another work proposed a sophisticated set of designed rules to recognise and model entangled tubes, detect entanglement and find a disentangling solution [5].

Compared to these previous works, our work also requires a 6 DoF manipulation policy that disentangles a grasped tangled object, which has only been studied in [4]. On top of the manipulation policy, our main contribution is a manipulation affordance-based grasp filter that selects grasp poses that are more likely to result in higher success rates, less object dropping and fewer contacts with non-grasped objects.

We leverage the recent development of deep reinforcement learning (DRL) [17], specifically the deep double Q-learning method [14], to

train the manipulation policy that separates a grasped hook. We refer the readers to [13] for a recent review of DRL for robot manipulation.

Another relevant topic is affordance learning in robotics [10,18]. Most works in this area focus on learning whether a grasp is stable (the graspability) [19,20]. Very few have studied whether a grasp pose can afford certain manipulations in terms of detailed performance metrics [21,22]. Our work fits into the last category, studying how to learn and use manipulation performances as affordance. Recent work proposes to define affordance over the RL framework such that the concept of affordance can be integrated into various RL algorithms more easily [11]. Another paper then suggests extending such a definition over temporally extended actions and thus allowing the fast learning and planning of temporally extended partial dynamic models [23]. They limit the action consequences within the sense of dynamic prediction, and our work will extend the definition over arbitrary action consequences which enables more flexible action selection.

3. General affordance theory

This section describes a novel concept named *general affordance*, based on which a manipulation framework is developed for contact-rich object disentangling tasks in Section 4. Since the concept of affordance was introduced [9], there have been various computational frameworks developed to integrate this concept in computer science and robotics [18]. However, there has not been a consensus on which one is more generalisable and interpretable [10]. In this paper, we embrace a new mathematical definition of affordance based on the RL paradigm [11]. The following content will review this definition and then propose the new *general affordance* concept that generalises beyond system dynamic predictions.

The RL task is defined on the Markov decision processes (MDPs), described formally by a tuple $\{\mathcal{S}, \mathcal{A}, p, p_0, r, \gamma\}$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ the action space, p the dynamic transition distribution, p_0 the initial state distribution, r the reward function, and γ the discount factor [17].

Khetarpal et al. [11] define affordance using the concept of *intent*, which is the *desired* next state distribution of an action taken at a state, denoted by $I_a : \mathcal{S} \to \mathcal{S}_a^+$. Given a threshold ϵ_I, an intent is said to be satisfied iif. $d(I_a(s), p(\cdot|s,a)) \leq \epsilon_I$, where $d(\cdot)$ is a distribution distance metric. Hence, given a set of intents for every action and state, $\mathcal{I} = \cup_{a\in\mathcal{A}} I_a$, a distance metric and a threshold value, the affordance of an agent is defined as a subset of state–action pairs: $\mathcal{AF}_{\mathcal{I}} \subseteq \mathcal{S}\times\mathcal{A}$, such that $\forall(s,a) \in \mathcal{AF}_{\mathcal{I}}$, the intents are satisfied. Interesting problems arise with such a definition such as learning a partial dynamic model, learning the set of affordances, using the set of affordances for planning, constraining action space, etc.

The affordance definitions in [11,23] are based on the prediction of the atomic or temporally extended next state. In other words, the affordances of actions only capture the action consequences in terms of the state distributions. While in the real world, affordances should be able to describe non-dynamical action consequences that are likely to be long-term and delayed. For example, the cost of driving, the success or failure of a game, the amount of water spilt or poured into a cup, etc. Defining affordances over a broader range of action consequences will allow more useful and practical real-world applications (which this paper demonstrates with a complicated robotic manipulation task). Notice that a policy π now needs to be included as the long-term action consequence is always induced by some policy. Following this thought, the general intent can be defined with respect to a given measurement y. The following gives the new definition of *general intent* and *general affordance* for arbitrary action consequences.

Definition 1 (*General Intent* $I^y_{a,\pi}$)**.** Given a measurement $y : \mathcal{S}\times\mathcal{A} \to \mathcal{Y} \in \mathbb{R}^n$, where $\mathcal{Y}$ is the space of all possible measurement values and $\mathbb{R}^n$ is the n-dimensional real number space, the general intent w.r.t. measurement y for a policy π is defined as a mapping to a subset of measurement values that are desired (or intended) to be achieved by taking action a and following the policy π thereafter, denoted as $I^y_{a,\pi}(s) = y^{\pi+}(s,a) : \mathcal{S} \to \mathcal{Y}^+_{a,\pi} \in \mathcal{Y}$.

Note that, let y be the system transition function and $\mathcal{Y} = \mathcal{S}$, general intent becomes dynamical intent. For dynamical action consequences, intent satisfaction can be done by comparing some distance between the intent and true system state. However, for general intent, its satisfaction check requires a target measurement value distribution induced by some baseline policy $\hat{\pi}$, which could but not necessarily be the optimal policy.

Definition 2 (*Intent Satisfaction*)**.** Denote $y^{\hat{\pi}}$ as the action consequences w.r.t. the measurement y of taking an action a and following the baseline policy $\hat{\pi}$ thereafter, given a distribution distance metric $d(\cdot)$ and a threshold ϵ_I, the general intent $I^y_{a,\pi}$ is satisfied according to the standard of $\hat{\pi}$ iif.:

$$d(I^y_{a,\pi}(s), y^{\hat{\pi}}(s,a)) \leq \epsilon_I \tag{1}$$

Then, for a policy π, the general affordance is defined as follow:

Definition 3 (*General Affordances* $\mathcal{AF}^y_{\mathcal{I},\pi}$)**.** Given a set of general intents for a policy π for all actions and states in an MDP, $\mathcal{I}^y_\pi = \cup_{a\in\mathcal{A}} I^y_{a,\pi}$, a distribution distance metric and a threshold, the general affordances for an agent is a subset of state–action pairs, $\mathcal{AF}^y_{\mathcal{I},\pi} \subseteq \mathcal{S}\times\mathcal{A}$, such that $\forall(s,a) \in \mathcal{AF}^y_{\mathcal{I},\pi}$, its general intent (Eq. (1)) is satisfied according to the standard of a baseline policy $\hat{\pi}$.

One can notice that general intents include dynamical intents, whether it is for one-step or temporally extended dynamics. Similarly, general affordance includes the dynamical affordance definition proposed in [11,23]. Knowing the general affordances of an agent benefits a number of decision-making aspects, including action filtering, exploration region selection, planning action space reduction, etc. With this new definition, existing RL tools can use affordances to take actions based on not only dynamic transitions but also arbitrary action consequences and task performance metrics.

Lastly, it is worth noting that, by changing the *desired precision*, ϵ_I, the space of satisfied intents can be changed [11]. For instance, if $\epsilon_I = +\infty$ then any intent can be satisfied, which is unrealistic. This paper assumes $\epsilon_I = 0$, such that all intents are satisfied strictly according to the baseline performance.

In the next section, we implement this idea into a manipulation system where the manipulation performances of TAG actions are learnt and then used to select affordable TAGs to improve manipulation outcomes.

4. General affordance-aware manipulation

4.1. Overview

We consider a disentangling task in which a robot arm needs to grasp a hook stably and separate the grasped hook while keeping the movement of other hooks minimal. The problem is decomposed into three sub-problems: (1) task-agnostic grasp (TAG) pose estimation; (2) manipulation affordance-based grasp filtering; and (3) RL-based manipulation.

As shown in Fig. 3, our general affordance-based manipulation (GAM) framework relies on existing methods to generate TAG poses from a point cloud. Then, a grasping pose and the latent representation of the scene are used to predict the disentangling manipulation outcomes by the manipulation affordance prediction (MAP) model. Lastly, three grasp filtering strategies (i.e., intents) are defined for the TAGs. When a strategy (an intent) is satisfied, the robot proceeds to separate the grasped hook using the RL policy. The following subsections explain the three steps in detail.

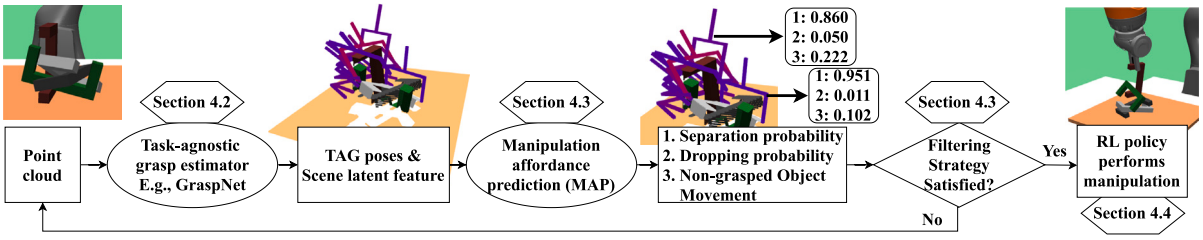


Fig. 3. General affordance-based manipulation (GAM) framework. Our approach is comprised of three sub-modules: (1) a task-agnostic grasp (TAG) generator (see Section 4.2) that proposes stable TAG poses; (2) the proposed manipulation affordance-based grasp filter (MAGF) that uses predicted manipulation performances to filter the TAG poses; (3) a reinforcement learning manipulation policy that controls the gripper to separate the grasped hook.

4.2. Task-agnostic grasp generation

The GraspNet-1Billion baseline [7] is adopted in our experiments to general TAGs, though other generators could also be used. We assume that the TAG poses provided by the generator (without any fine-tuning) are all reachable and stable. Unfortunately, this is not true in practice due to the difference between training and deployment environments. This means that in practice, most of the generated grasps are not reachable. In order to reduce the burden of computation and time spent on finding a reachable grasping pose in the training of the following manipulation policy, we pre-record a set of reachable and stable TAG poses along with their simulation system states and point cloud latent features. During the training and evaluation of the RL agent and the MAP model, we do not need to run the GraspNet model anymore, but simply load the stable grasps and reset the simulation to the corresponding recorded state.

To determine whether a grasp pose is reachable and stable, we conduct a basic manipulation stability test. The test starts with the gripper being moved to the grasp pose and closing its fingers. Then the gripper is moved by the RL agent by executing all the RL actions once (see action definition in Section 4.4). If the object remains within the finger at the end of the action sequence, then the grasp pose is stable and recorded (along with the latent representation of the scene and the simulation system state).

Although the recorded grasp poses are reachable by the robot and stable according to the basic stability test, not all of them afford the separation of the grasped object (some may be easier to drop the object), and each one would lead to a different amount of movements of the non-grasped objects. To address the above issue, we introduce the manipulation affordance prediction model and grasp filter.

4.3. Manipulation affordance-based grasp filter

As humans, we are able to choose grasping poses most suitable for different manipulation tasks, because we can predict whether a grasp pose (an action at a state) would lead to the preferred manipulation outcome (satisfied intent). Here, the proposed manipulation affordance-based grasp filter (MAGF) is inspired by the same underlying logic. The following will introduce the manipulation affordance prediction (MAP) network and the three example filtering strategies (action selection rules based on intent satisfaction).

MAP: Recall that, to learn general affordances, the action consequences need to be defined and a baseline distribution is required. Here, we define the consequences of grasp actions using three performance measurements:

- y_1: the probability of separating the grasped object.
- y_2: the probability of dropping the object.
- y_3: the average movements of non-grasped objects.

Note that it is possible to learn to predict whether a grasp is stable and release our method from the assumption of having a TAG generator. In other words, include the stability score of the grasp as another performance measurement. However, assuming access to a TAG generator significantly narrows down the solution space of grasps for the downstream manipulation. This also enables faster implementation as the RL policy will take unrealistically too long to learn both stable grasping and manipulation given the difficulty of the manipulation tasks.

Here, we adopt the RL policy that learns to disentangle the grasped hook as the baseline policy. Therefore, the MAP model learns to predict the manipulation performances that can be achieved by the RL policy, denoted as $y_1^{\hat{\pi}}$, $y_2^{\hat{\pi}}$, and $y_3^{\hat{\pi}}$. This means that given a set of intents (with desired manipulation performances y_1^+, y_2^+, and y_3^+), one can perform action filtering by comparing the predicted performances and the desired performances. In effect, this induces the grasps that afford the desired manipulation outcomes.

Filtering strategy: One can design different filtering conditions with the predicted manipulation performances to decide whether the manipulation policy should start disentangling the grasped object for a specific TAG at the current observation. This grounds down to the design of the distance metric d and corresponds to different intents and different sets of affordances. Formally, given a state s, a grasp pose is affordable iif. $d(y_i^+, y_i^{\hat{\pi}}(s,a)) \leq \epsilon_i$ according to the ith manipulation performance metric and an intent. Given different intents (desired value of measurement), with $\epsilon_I = 0$, we experiment with the following grasp filtering strategies and discuss their results.

I Filter away grasp actions that cannot achieve the desired separation success rate, i.e.: accept a grasp if $y_1^+ \leq y_1^{\hat{\pi}}$.

II Filter away grasp actions that cannot achieve the desired object dropping probability, i.e.: accept a grasp if $y_2^+ \geq y_2^{\hat{\pi}}$.

III Filter away grasp actions that cannot achieve the averaged movements of the non-grasped hooks, i.e.: accept a grasp if $y_3^+ \geq y_3^{\hat{\pi}}$.

IV Combine strategies I and III.

4.4. Reinforcement learning-based manipulation

We take an episodic RL approach to solving the hook separation task. A deep double Q-learning agent [14] is trained. The RL problem is formulated as follows.

In each episode, the robot starts at a state where the gripper is in a grasping pose and a hook is gripped in hand. Then at each timestep, the RL agent selects an action to move the grasped hook and observe a new state of the environment and a reward. The episode is terminated when: (1) the object is no longer grasped; (2) the object is moved out of the workspace; or (3) when the episode length N is reached (in our experiments $N = 3$ timesteps). After the robot has performed N actions, it will perform a manually defined motion to move straight up for 0.1 m. If the grasped object is lifted up alone, then the episode succeeds. Otherwise, it is considered a failure (e.g., the object dropped during lifting up or other objects being hooked up together). In practice, we examine the finger width to determine whether the grasped object has been

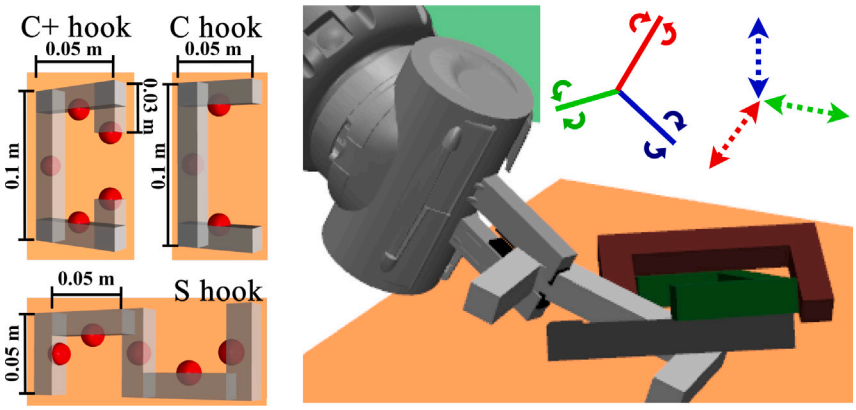


Fig. 4. Left: hook sizes and their keypoint (red) representations. Right: translation actions along the world frame directions (dash lines) and rotation actions about the gripper tip frame directions (solid lines).

dropped. The object is dropped when the finger width becomes too small. Then, to determine whether the hook is separated successfully, we examine the contacts between the hooks and the gripper fingers. If after the final lifting up motion, there *exist and only exist* contacts between the fingers and the grasped hook, then it is a success. The following will introduce the definitions of the state, action and reward.

State: The system state consists of the states of the hooks and the gripper. For each hook, its state consists of the Cartesian coordinates of the keypoints (see Fig. 4 left) and the quaternion of the hook centre in the world frame. For the gripper, its state consists of the Cartesian coordinates and the quaternion of the gripper tip, the width between the two fingers and the index of the hook being grasped.

Action: At each timestep, the RL agent selects one action from 12 discrete actions to move the gripper tip frame. The actions are small translations along the axes of the world frame (i.e. the robot base frame), and small rotations about the axes of the gripper tip frame (see Fig. 4 right). The first 6 actions translate the gripper tip frame in the axes' positive and negative directions for a fixed distance d_a, while the last 6 actions rotate the gripper tip frame about the axes in clockwise and counter-clockwise directions for a fixed angle δ_a. In our experiments, $d_a = 0.05m$ and $\delta_a = 30°$. This action space is named Cartesian movement (CM) action space for clearer comparison with the hemisphere movement (HM) action space proposed in [4].

Reward: At each timestep, the agent is given a reward, defined as a weighted negative value of the average movement of all non-grasped objects. This is because we believe that minimising such movement is an important constraint in object disentangling tasks. There are three termination conditions that will bring different terminal rewards. The first two conditions are when the grasped hook hand is dropped or moved out of the workspace, which gives the agent a punishment. The third condition is when an episode naturally ends at the last timestep, and a straight lifting-up motion is performed. A successful lifting up of the grasped hook will result in a positive reward, and a failed attempt will result in zero rewards. Only lifting up the grasped hook will be considered a success, and lifting up others will be considered a failure. This reward definition can be summarised by the following equation.

$$\boldsymbol{R}_t = \sigma \times d_t^{obj} + r_t^{step},$$

where, σ is the weight of the averaged non-grasped object movement, d_t^{obj}; and r_t^{step} is defined as

$$r_t^{step} = \begin{cases} 0, & \text{non-terminal or fail to lift up object} \\ a, & \text{successfully lift up object at the end} \\ b, & \text{grasped object dropped or out of workspace} \end{cases}$$

In our experiments, $\sigma = -1.0$, $a = 10$ and $b = -10$.

Network: The Q network is a three-layer MLP with size 256-256-256. Each layer is activated by ReLU and the output layer has no activation.

End-to-end baselines: We also design experiments to demonstrate that it is highly inefficient to learn both grasping and separation manipulation simultaneously through end-to-end RL. In the end-to-end setup, we extend the CM and HM action spaces with two extra actions to open and close the fingers. The robot no longer starts with a state where a hook is grasped, but instead always starts with a configuration where the gripper is facing towards the table (as shown by Fig. 5 left). To give sufficient exploration time, as the robot also needs to learn grasping, each episode is extended to a length of $N = 30$. The index of the grasped hook is removed from the state representation. The termination conditions remain the same, except dropping a grasped hook no longer terminates an episode. The reward function remains the same except that a positive reward c is given at each timestep when the robot has a hook grasped in hand. The non-grasped object movement is set to zero when there is no hook grasped in hand. Similarly, a straight lifting-up motion is performed when an episode reaches its maximum timestep, and then the finger width and contacts are used to determine whether a hook is successfully separated.

5. Task and training details

We use the Mujoco physic engine [24] to design five simulation tasks, in which a Kuka robot, equipped with a Rethink parallel-jaw gripper, is given a number of entangled rigid C/C+/S-shaped hooks (Fig. 5) and tasked to pick and lift only one of them. The sizes of the hooks are displayed in Fig. 4 left. Each hook has a square intersecting surface of size 0.014×0.014 m^2. The workspace is a square of size 0.3×0.3 m^2 (the orange square Fig. 5 left).

Data collection: As mentioned in Section 4.2, we collect a set of stable TAG poses for each of the five tasks, along with their point cloud latent representations and simulation states (250000 data points for each). Specifically, the hooks are randomised in a tangled configuration when a scene resets. They are then dropped together from a location $0.1m$ higher than the centre of the table. The GraspNet model then takes the partial point cloud from the top view camera and outputs 10 TAG poses with the best scores, along with their latent features. The simulator moves the robot to the grasping pose and closes the fingers. Then, each of the RL actions is performed once to confirm the stability of the grasp. If the object remains grasped, then the grasp pose, the latent feature, and the simulation state of the object being grasped are recorded as a data point. The scene resets again when the 10 TAGs are all tested.

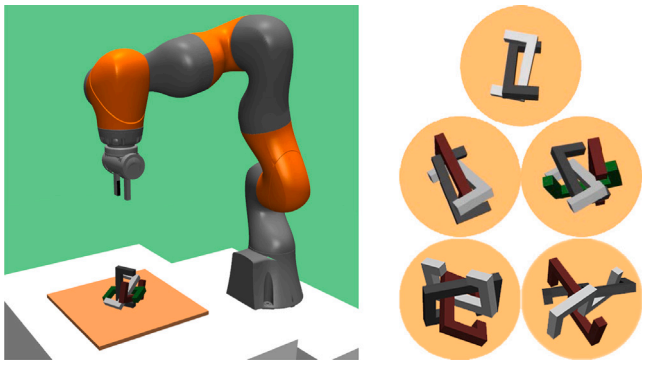


Fig. 5. Simulation environment and task visualisation.

In order to train the manipulation affordance prediction (MAP) model, we need to collect training labels for these collected data points. As affordance is about what can be done with the object by an agent, such labels are most natural to come from the manipulation policy itself. In other words, what manipulation performances a grasp pose allows to happen can only be known after the robot actually manipulates the grasped object using that grasp pose. Thus, we first train the RL agent to separate the grasped hook with every collected TAG pose, and then evaluate the trained agent on all the TAG poses to collect training labels for the MAP model. The performances of the RL agent are presented in Section 6.2. After the MAP model is trained, we then apply it with the filtering strategies to select grasp poses before executing manipulation. The results are summarised in Section 6.3.

RL: The RL agent is updated exactly once on each interaction step for a total of $2e6$ timesteps (Assuming each episode terminates after 3 steps, there are more than $6e6$ episodes in total). We used a replay buffer of size $1e6$. Each agent performs $2e3$ random actions and optimisation steps at the beginning of the training. Mini-batch optimisation is performed with Adam [25] with a $1e-4$ learning rate, a batch size of 128 and a discount factor of 0.99. During the update, the value target is clipped to be within $[-50, 50]$. The target network is updated by copying all the main Q network's parameters every $1e3$ steps. We use ϵ-greedy as the exploration strategy, which decays the random action probability linearly from 1.0 to 0.05 in $5e4$ timesteps. The agent is evaluated for 30 episodes (without exploration) every $1e4$ timesteps to generate the performance figures.

MAP: To align with the GraspNet model, the input of the MAP network is a 1D vector comprised of a grasp pose and its associated scene representation. In our experiments, the grasp pose is flattened from the position (3D) and the rotation matrix (3×3) in the camera frame, while the latent feature (256-dim) is provided by the point auto-encoder of the GraspNet model (see Fig. 5-a of [7]). Note that this input could be changed to suit other TAG generators. The input first goes through a BatchNorm layer and then a three-layer (512-512-256) Multi-Layer Perceptron (MLP). The network output is a 3D vector, which is the predicted manipulation affordances. The three MLP layers are activated by ReLU. The probability outputs are activated by Sigmoid functions and trained with the Binary Cross Entropy loss. The average movement output has no activation function and is trained with smooth L1 loss. The MAP is trained by Adam [25] with a learning rate of $1e-3$ and a batch size of 1024 for $1e4$ optimisation steps.

6. Results

6.1. End2end reinforcement learning experiment

This subsection provides experimental results that demonstrate the ineffectiveness of training an RL agent to learn grasping and separation manipulation altogether. Fig. 6 displays the average (three seeds) evaluation results of the CM (blue curves) and HM (orange curves) agents trained end-to-end for the task with two C-shape hooks. Unsurprisingly, both agents perform very poorly in this training setting. Though the agent with our action space managed to grasp an object a few times, they both struggled to learn grasping as the number of grasped hooks remained very low. In short, these results confirm the inefficiency of learning to grasp and separate entangled hooks in an end-to-end fashion and the benefits of adopting an off-the-shell grasp generator, such as the GraspNet model adopted by this paper.

6.2. Reinforcement learning experiment

In this subsection, we discuss the training results of both RL agents (each averaged over 3 random seeds) with the CM and HM [4] actions, compared with the SLM baseline performances for all five tasks in Fig. 7. The straight-up lifting motion (SLM) baseline simply carries the grasped object up for $0.15m$, and its performance is evaluated with $5e4$ TAGs per task. The grasp filter is not applied in the experiments in this section.

First of all, the SLM baseline (green lines) achieves the lowest success rates, but near-zero object dropping rates and non-grasped object movements in all tasks. This is expected as the SLM motion only carries the grasped object up straightly, and all the TAG configurations have already passed the basic manipulation test (see Section 4.2).

Secondly, the RL policies both outperform the SLM baseline in terms of success rates in all task variations. However, they both perform worse than SLM in terms of object-dropping rates and non-grasped object movements. This indicates that there is a trade-off between motion flexibility and the risks of dropping objects and influencing other objects. However, the agent with our action space achieves smaller dropping rates and non-grasped object movements than [4], whose hemisphere actions move the grasped hook more drastically. This indicates that RL algorithms are able to learn to manipulate more carefully if the action space is more fine-grained.

Overall, the three manipulation policies achieve low separation success rates in all tasks, especially for the difficult ones (C+ and S hooks). The RL policies perform slightly better than the SLM baseline in terms of success rates. However, they achieve higher object-dropping rates and non-grasped object movements and become less advantageous as the tasks become more difficult. This reveals that a large proportion of the TAGs do not afford good manipulation performances. This could be potentially improved by learning a more complex RL policy, but we argue and demonstrate that this can also be largely resolved by selecting TAGs using the proposed MAGF module in the next section.

6.3. Grasp filter results

By evaluating our RL agent on all the recorded grasping poses, we obtain the manipulation performance labels to train the MAP model. To evaluate the manipulation performances with a grasp filter (GF), we use the trained MAP model to predict the manipulation outcomes of the CM policy given each recorded TAG. The algorithm discards the unsatisfactory TAGs and executes the CM policy when the selected filtering strategy is satisfied.

Baselines: The comparison includes the SLM baseline (on 50000 TAGs), both RL agents without a GF, and the CM policy with the proposed filtering strategies. Each strategy is evaluated with different desired measurement values, inducing different sets of intents. Four sets of intents are examined for the strategy I: $y_1^+ \in \{0.8, 0.9, 0.95, 0.98\}$. Three sets of intents are examined for strategy II: $y_2^+ \in \{0.1, 0.2, 0.3\}$. Three sets of intents are examined for strategy II: $y_3^+ \in \{0.1, 0.2, 0.3\}$. Four sets of intents are examined for strategy IV $(y_1^+, y_3^+) \in \{(0.8, 0.2), (0.8, 0.1), (0.9, 0.2), (0.9, 0.1)\}$. For each comparative case, the RL agent is evaluated with the three random seeds. Each seed run stops when a total

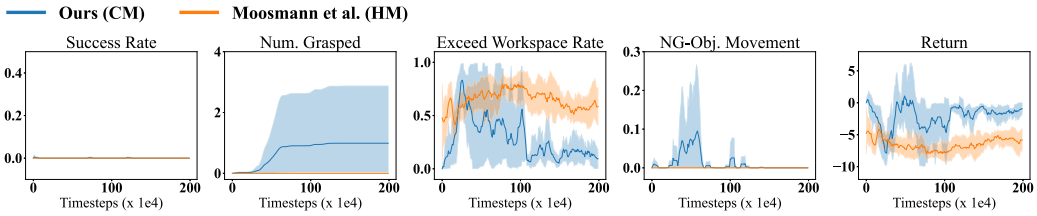


Fig. 6. Means and standard deviations of the testing results on the task with 2 C hooks for both RL agents without access to a TAG generator. **The results prove that it is much more inefficient to learn grasping and separation simultaneously for such hook separation tasks** From left to right: separation success rate, number of grasped objects, rate of the gripper moving out of the workspace, non-grasped object movements (when an object is grasped), and return.

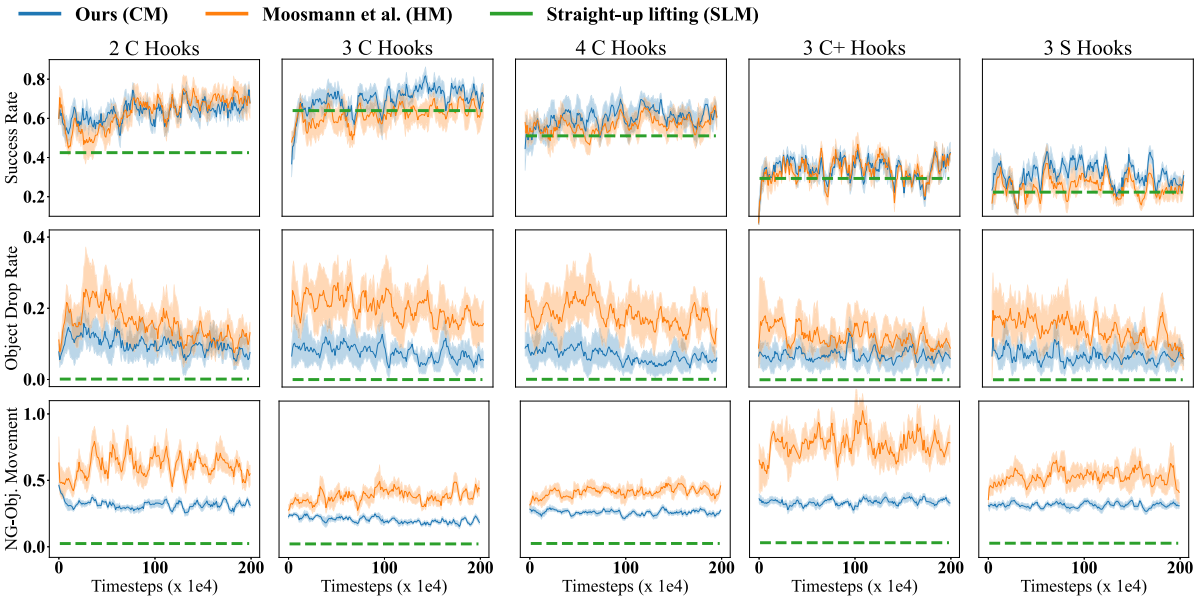


Fig. 7. Means and standard deviations of the testing results of both RL agents (blue and orange) and the performance of the straight-up lifting motion (green). **The performances, in general, reveal that a large proportion of the task-agnostic grasps is not suitable for the downstream manipulation**. The situation is exacerbated as the task becomes more difficult. Rows (top to bottom): success rates, object dropping rates, average non-grasped object movements (in meters). Columns (left to right): the tasks with 2, 3, 4 C hooks, 3 C+ hooks and 3 S hooks.

of 15000 TAGs are passed (that is, for each evaluation case, a total of 45000 different TAG poses are verified). Fig. 8 reports the success rates, object dropping rates, non-grasped object movements (NGOM) and proportions of discarded TAGs for each comparative case per task.

Performances without MAGF: Across all tasks, the three black colour bars in Fig. 8 show that the evaluation performances of the SLM baseline and both RL agents without GF are consistent with that reported in the last subsection. It indicates that a large proportion of TAGs do not afford satisfactory manipulation performances for such contact-rich tasks.

Performances with MAGF: The green, blue and pink colour sets show that the CM policy with strategies I, III and IV achieves substantially better manipulation performances, while the orange colour set shows that strategy II can improve the performance slightly over the baselines. Overall, these results firstly indicate that the MAP model successfully learns to predict the general action consequences in terms of manipulation performances, as the actual results match the threshold values. Secondly, the improved results, in general, demonstrate the effectiveness of applying the MAGF module before executing a manipulation policy. Thirdly, it showcases the successes of selecting TAGs according to certain individual (I, II, and III) or combined (IV) detailed manipulation performances. Lastly, the fourth histogram shows that, as the manipulation constraints become stricter, more TAGs will become invalid and need to be discarded.

The results also reveal an interesting correlation between the non-grasped object movements and success rates. In all the tasks with strategy I (green bars), a higher success rate is always accompanied by a smaller chance of dropping the hook and a smaller amount of averaged movements of the non-grasped objects. Those with strategy II (orange bars) show that a smaller chance of dropping the grasped object does not necessarily lead to a high separation success rate or smaller non-grasped object movements. On the other hand, those with strategy III (blue bars) show that restricting the non-grasped object movements actually does lead to higher success rates and lower object-dropping rates.

We hypothesise an explanation as follows. As non-grasped object movements are caused by contacts and forces among objects, the results indicate that reducing contact with other objects leads to better performances. This might be attributed to the logic that when the gripper and the grasped object have fewer contacts with other parts, the grasped object encounters less interference, making it easier to separate. On the contrary, it is more likely to be dropped when it has many contacts with other objects. In addition, merely preventing the dropped grasped object and disregarding minimising contact with other objects could lead the robot to overly prioritise not dropping the object, neglecting the more important objectives of separation and minimising contact.

6.4. Summary

In general, with these results, we can obtain the following observations:

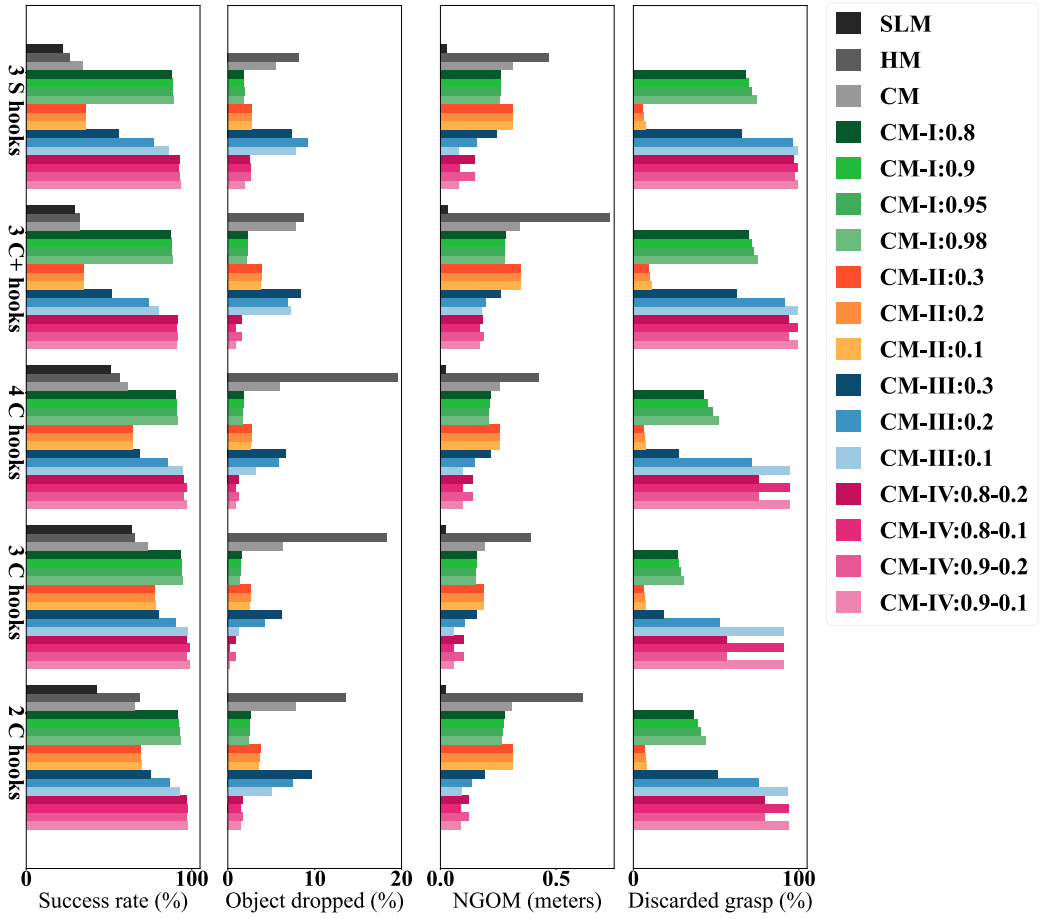


Fig. 8. Comparison of different filtering strategies. From left to right: success rates, object dropping rates, non-grasped object movements (NGOM) and proportion of discarded grasps. Each histogram shows the performances of the five tasks from top to bottom, with 3 S, 3 C+ and 4, 3, and 2 C hooks. SLM: straight-up lifting motion. HM: hemisphere movement. CM: Cartesian movement. Black colour set: baselines without MAGF. Green, orange, blue and pink colour sets: CM policy with strategy I to IV. **In general, the figure shows a clear advantage of selecting TAGs according to estimated manipulation performances.**

- A large portion of TAG poses are unsatisfactory for such contact-rich manipulation tasks, given the proportion of discarded TAGs reported in each strategy result (the rightmost histogram in Fig. 8).
- Learning to predict manipulation performances for TAGs provides an effective tool for a manipulation system to select grasps that can achieve substantially better results.
- Task-wise, more contacts made between the grasped object and others will result in higher object-dropping rates and more failures, and it leads to better separation performances by avoiding contacts than by simply avoiding dropping the grasped objects.

These results provide the following insights about robotic manipulation. First, TAG generators are insufficient for complex manipulation tasks, and more effort is required to develop task-relevant grasp generation and selection in a detailed manner. Secondly, albeit RL has achieved great success in many tasks, it clearly demands more effort to be improved and implemented for contact-rich manipulation, especially its interplay with grasp generation and selection. Lastly, for a robot to master contact-rich manipulations such as the example hook separation tasks, more effort and domain knowledge are needed to investigate and identify the key factors of the object interaction dynamics that impact manipulation performances for individual tasks, such as predicting the average non-grasped object movement and using it as an indicator for grasp filtering.

6.5. Remarks on scalability

As shown by the empirical results, the proposed framework is capable of handling tasks with up to four hooks with C and three hooks with C+ and S shapes. However, it is necessary to discuss further the scalability of the proposed method in terms of the number of objects and the shapes of objects.

First of all, as the number of objects within the scene increases, it would bring down the separation performance of the RL agent slightly, because more objects may lead to more entanglement. However, it is not as severe as influenced by the shapes of the objects. This can be observed from Fig. 7 as the performances of the tasks with 3 C+ and 3 S hooks are much worse than the tasks with 4 C hooks. Secondly, the shape of the object is the main cause of severe entanglement and, thus, low separation success rates. This is physically natural as complex-shaped objects are more difficult to separate from an entangled state. In addition, there are fewer grasping poses that afford good separation outcomes. Thus, the specific implementation of the proposed manipulation framework would have lower success rates if one keeps increasing the number of objects and the complexity of the shapes of the objects. Nonetheless, the decreased performance is expected to depend more heavily on the manipulation capability of the RL policy, which controls the gripper, while the proposed MAP module will still be able to identify those grasps that afford the desired manipulation outcomes. This is demonstrated by the pink colour bars in Fig. 8, which displays high performances in all situations even when the RL policy struggles to separate the grasped hook with most grasping poses, which are indicated by the blue colour bars.

Overall, the affordance prediction model would be much less influenced by the number of hooks and the shapes of hooks, because it concerns about predicting the performances of a particular manipulation policy. However, the performance of the RL policy can indeed be influenced severely by the number and shape of objects, mainly because, in this research, the RL policy uses state-based observations. The state representation designed for this research is not very scalable as it contains the keypoint poses of all objects in the scene, which can easily suffer from the curse of dimensionality as the number of objects increases or the insufficient information provided by the representation as the shape of the objects become too complex for keypoint representation. This suggests that, in the future, more complex representations for the RL policy and/or the MAP model could be employed to handle tasks with more objects of more complex shapes.

6.6. Remarks on real-world applications

Finally, this research does not provide experiments on a real-world robot platform. Therefore, a discussion of its possibility of being implemented in the real world is necessary. In particular, the proposed framework is not limited to simulation at all. However, the actual implementation of the framework in this paper is considerably difficult to directly deploy on a real robot.

Firstly, the theoretic idea of general affordance is applicable to all decision-making systems that are important in the real world, including the robotic manipulation system that this research is working on and many others, such as autonomous driving, unmanned aerial vehicles, games, etc. Therefore, similar to many other decision-making systems, the performance of real-world deployment of the proposed framework depends heavily on the perception ability of the system. In other words, it depends heavily on how accurately the system can capture task-relevant information from the real world.

Secondly, regarding the specific implementation of the GAM framework, the difficulty of being directly deployed on a real robot platform stems mainly from the design of the state representation of the objects for the RL manipulation policy. It is possible, but very difficult, to obtain the accurate keypoint poses for the hooks in the real world due to occlusions of entangled objects. However, the MAP model would be more likely to work in the real world because it takes point cloud features produced by a deep neural network as the input, which is not difficult to obtain. This highlights the significance of research in state estimation or representation learning for manipulation using deep neural networks.

7. Conclusion

This work studied a challenging contact-rich manipulation problem of separating an entangled hook following the classic two-stage approach: finding a grasp pose and then a manipulation trajectory. To improve over previous works, this paper proposed the concept of *general affordances*, which mathematically describes whether an action can afford an arbitrary consequence at a state. Based on this theory, this paper introduced the general affordance-based manipulation (GAM) framework, which enables a robot to learn any user-defined manipulation metric for task-agnostic grasps (TAGs) and then select TAGs that lead to better predicted manipulation results.

Briefly, the implementation of GAM consists of: (1) an RL policy that manipulates a grasped object, (2) a manipulation affordance prediction model that predicts manipulation performances given TAGs, and (3) a grasp filter that selects TAGs with different filtering strategies. Comprehensive experiments of five challenging hook separation tasks in simulation have shown that:

- TAGs are not sufficient for contact-rich manipulation tasks;
- It is possible to learn to predict manipulation performances given a TAG and a latent representation of the scene;
- Selecting TAGs with high predicted manipulation performances leads to significant manipulation improvement;
- It is possible to use the theory of general affordance to improve robotic manipulation.

For future research, more efforts are required to test the GAM with manipulation policies that take image/point-cloud inputs, which have a better potential to be deployed in the real world. Secondly, our experiments show that controlling how many contacts can be made is important for such contact-rich manipulation tasks. It would be valuable to invest in techniques for contact/tactile sensing and control. It is also promising to explore and implement other general affordance-based frameworks for robotics.

CRediT authorship contribution statement

Xintong Yang: Conceptualization, Data curation, Formal analysis, Methodology, Software, Visualization, Writing – original draft, Writing – review & editing. **Jing Wu:** Conceptualization, Methodology, Supervision, Writing – review & editing. **Yu-Kun Lai:** Conceptualization, Methodology, Supervision, Writing – review & editing. **Ze Ji:** Conceptualization, Funding acquisition, Investigation, Methodology, Resources, Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Information on the data underpinning the results presented here, including how to access them, can be found at DOI: 10.5281/zenodo.10654542.

Acknowledgements

Xintong Yang thanks the China Scholarship Council (CSC) for providing the living stipend for his Ph.D. programme (No. 201908440400). This work was partially supported by the Engineering and Physical Sciences Research Council (grant No. EP/X018962/1).

References

[1] M.T. Mason, Toward robotic manipulation, Ann. Rev. Control Robotics Auton. Syst. 1 (1) (2018).
[2] R. Matsumura, Y. Domae, W. Wan, K. Harada, Learning based robotic bin-picking for potentially tangled objects, in: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2019, pp. 7990–7997.
[3] M. Moosmann, F. Spenrath, K. Kleeberger, M.U. Khalid, M. Mönnig, J. Rosport, R. Bormann, Increasing the robustness of random bin picking by avoiding grasps of entangled workpieces, Procedia CIRP 93 (2020) 1212–1217.
[4] M. Moosmann, M. Kulig, F. Spenrath, M. Mönnig, S. Roggendorf, O. Petrovic, R. Bormann, M.F. Huber, Separating entangled workpieces in random bin picking using deep reinforcement learning, Procedia CIRP 104 (2021) 881–886.
[5] G. Leão, C.M. Costa, A. Sousa, G. Veiga, Detecting and solving tube entanglement in bin picking operations, Appl. Sci. 10 (7) (2020) 2264.
[6] X. Zhang, K. Koyama, Y. Domae, W. Wan, K. Harada, A topological solution of entanglement for complex-shaped parts in robotic bin-picking, in: 2021 IEEE 17th International Conference on Automation Science and Engineering, CASE, IEEE, 2021, pp. 461–467.
[7] H.-S. Fang, C. Wang, M. Gou, C. Lu, Graspnet-1billion: A large-scale benchmark for general object grasping, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 11444–11453.
[8] V. Ortenzi, M. Controzzi, F. Cini, J. Leitner, M. Bianchi, M.A. Roa, P. Corke, Robotic manipulation and the role of the task in the metric of success, Nat. Mach. Intell. 1 (8) (2019) 340–346.
[9] J.J. Gibson, The theory of affordances, Hilldale, USA 1 (2) (1977) 67–82.
[10] X. Yang, Z. Ji, J. Wu, Y.-K. Lai, Recent advances of deep robotic affordance learning: A reinforcement learning perspective, IEEE Trans. Cogn. Dev. Syst. (2023) 1, http://dx.doi.org/10.1109/TCDS.2023.3277288.

[11] K. Khetarpal, Z. Ahmed, G. Comanici, D. Abel, D. Precup, What can I do here? A theory of affordances in reinforcement learning, in: International Conference on Machine Learning, PMLR, 2020, pp. 5243–5253.
[12] S. Karaman, E. Frazzoli, Sampling-based algorithms for optimal motion planning, Int. J. Robotics Res. 30 (7) (2011) 846–894.
[13] R. Liu, F. Nageotte, P. Zanne, M. de Mathelin, B. Dresp-Langley, Deep reinforcement learning for the control of robotic manipulation: a focussed mini-review, Robotics 10 (1) (2021) 22.
[14] H. van Hasselt, A. Guez, D. Silver, Deep reinforcement learning with double Q-learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 30, No. 1, 2016, http://dx.doi.org/10.1609/aaai.v30i1.10295, URL https://ojs.aaai.org/index.php/AAAI/article/view/10295.
[15] Q.M. Marwan, S.C. Chua, L.C. Kwek, Comprehensive review on reaching and grasping of objects in robotics, Robotica 39 (10) (2021) 1849–1882.
[16] M. Moosmann, F. Spenrath, M. Mönnig, M.U. Khalid, M. Jaumann, J. Rosport, R. Bormann, Using deep neural networks to separate entangled workpieces in random bin picking, in: Advances in Automotive Production Technology–Theory and Application, Springer, 2021, pp. 238–246.
[17] R.S. Sutton, A.G. Barto, Reinforcement Learning: An Introduction, MIT Press, 2018.
[18] N. Yamanobe, W. Wan, I.G. Ramirez-Alpizar, D. Petit, T. Tsuji, S. Akizuki, M. Hashimoto, K. Nagata, K. Harada, A brief review of affordance in robotic manipulation research, Adv. Robot. 31 (19–20) (2017) 1086–1101.
[19] H. Wu, Z. Zhang, H. Cheng, K. Yang, J. Liu, Z. Guo, Learning affordance space in physical world for vision-based robotic object manipulation, in: 2020 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2020, pp. 4652–4658.
[20] P. Mandikal, K. Grauman, Learning dexterous grasping with object-centric visual affordances, in: 2021 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2021, pp. 6169–6176.
[21] H. Wu, G.S. Chirikjian, Can i pour into it? robot imagining open containability affordance of previously unseen objects via physical simulations, IEEE Robot. Autom. Lett. 6 (1) (2020) 271–278.
[22] K. Mo, L.J. Guibas, M. Mukadam, A. Gupta, S. Tulsiani, Where2act: From pixels to actions for articulated 3d objects, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 6813–6823.
[23] K. Khetarpal, Z. Ahmed, G. Comanici, D. Precup, Temporally abstract partial models, Adv. Neural Inf. Process. Syst. 34 (2021).
[24] E. Todorov, T. Erez, Y. Tassa, Mujoco: A physics engine for model-based control, in: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2012, pp. 5026–5033.
[25] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.

Xintong Yang is a research associate at the school of engineering at Cardiff University. He received His bachelor's and master's degrees in Mechanical Engineering and Industrial Engineering in Guangdong University of Technology in 2019 and his Ph.D. in Engineering from Cardiff University in 2023. His research interests include deep (reinforcement) learning-based robotic grasping and manipulation, deformable object manipulation and differentiable physics.

Jing Wu is a lecturer in computer science and informatics at Cardiff University, UK. She received B.Sc. and M.Sc. from Nanjing University, and Ph.D. from the University of York, UK. Her research interests are in computer vision and graphics including image-based 3D reconstruction, face recognition, machine learning and visual analytics. She is on the editorial board of Displays and serves as a PC member in CGVC and BMVC.

Yu-Kun Lai is a Professor in the School of Computer Science and Informatics, Cardiff University. He received his bachelor's and Ph.D. degrees in computer science from Tsinghua University, China, in 2003 and 2008 respectively. His research interests include computer graphics, computer vision, geometric modeling, and image processing. He is on the editorial boards of IEEE Transactions on Visualization and Computer Graphics and The Visual Computer.

Ze Ji is a Senior Lecturer (Associate Professor) with the School of Engineering, Cardiff University. He received his B.Eng. from Jilin University, China, in 2001,M.Sc. from the University of Birmingham, UK, in 2003, and Ph.D. from Cardiff University, UK, in 2007. Before this, he was working in industry (Dyson and Lenovo) on autonomous robotics. His research interests include robot manipulation, autonomous robot navigation, robot learning, computer vision, simultaneous localization and mapping (SLAM), and tactile sensing.