2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

RLAfford: End-to-End Affordance Learning for Robotic Manipulation

Yiran Geng*[1], Boshi An*[1], Haoran Geng[1], Yuanpei Chen[2], Yaodong Yang†[2], Hao Dong†[1]

Abstract— **Learning to manipulate 3D objects in an interactive environment has been a challenging problem in Reinforcement Learning (RL). In particular, it is hard to train a policy that can generalize over objects with different semantic categories, diverse shape geometry and versatile functionality. In this study, we focused on the contact information in manipulation processes, and proposed a unified representation for critical interactions to describe different kinds of manipulation tasks. Specifically, we take advantage of the contact information generated during the RL training process and employ it as unified visual representation to predict contact map of interest. Such representation leads to an end-to-end learning framework that combined affordance based and RL based methods for the first time. Our unified framework can generalize over different types of manipulation tasks. Surprisingly, the effectiveness of such framework holds even under the multi-stage and multi-agent scenarios. We tested our method on eight types of manipulation tasks. Results showed that our methods outperform baseline algorithms, including visual affordance methods and RL methods, by a large margin on the success rate. The demonstration can be found at `https://sites.google.com/view/rlafford/`.**

I. INTRODUCTION

Learning to manipulate objects is a fundamental problem in RL and robotics. An end-to-end learning approach can explore the reach range of future intelligent robotics. Recently, researchers have shown an increased interest in visual affordance [1], [2], [3], [4], [5], *i.e.*, a task-specific prior representation of objects. Such representations provide the agents with semantic information of objects, allowing better performance of manipulation.

The existing affordance methods for manipulation have two training stages [3], [5], [4], [6]. For example, VAT-Mart [5] first trains the affordance map with data collected by an RL agent driven by curiosity, and then fine-tunes both the affordance map and the RL agent. In Where2act [3] and many other works [7], [4], [6], affordance is associated with a corresponding primitive action for each task, such as pushing and pulling. Some recent works [1], [8] learn affordance through human demonstration. A significant drawback of those two-stage methods, which first train the affordance map and then propose action sequence based on the learned affordance, is that the success rate of interaction is highly related to the accuracy of the learned affordance. Any deviations in affordance predictions will significantly reduce the task performance. Hence, a method that require neither task-specific action primitives nor seperate training stages would significantly surpass current state-of-the-art results.

1 CFCS, School of CS, Peking University
2 Institute for AI, Peking University
The first two authors contributed equally.
Corresponding to {hao.dong, yaodong.yang}@pku.edu.cn

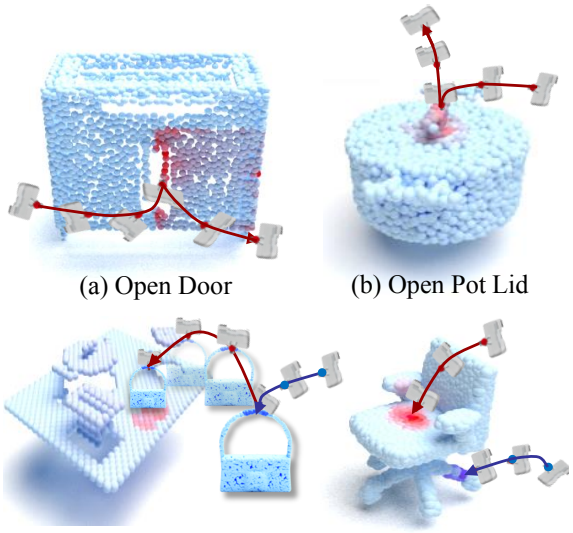

Fig. 1. **Affordance examples of different manipulation tasks.** (a-b): Agent-to-object affordance map. (c): Agent-to-object and object-to-object affordance map. (d): Dual agent-to-object affordance map.

In this paper, we investigate learning affordance along with RL in an end-to-end fashion by using contact frequency as a unfied representation for affordance. Therefore the affordance is not associated with a specific primitive action but rather the fundamental information in manipulation. In our method, the RL algorithm learns to utilize visual affordance generated from contact information to find the most suitable position for interaction. We also incorporate visual affordance in reward signals to encourage the RL agent to focus on points of higher likelihood. The advantages of end-to-end affordance learning are two-fold: 1) affordance can awaken the agent where to act as an additional observation and be incorporated into reward signals to improve the manipulation policy; 2) learning affordance and manipulation policy simultaneously, without human demonstration or a dedicated data colleting process simplifies the learning pipeline and can migrate to other tasks easily. Additionally, it helps the affordance and manipulation policy to adapt to each other, thus producing a more robust affordance representation.

Using contact information as affordance evidently supports multi-stage tasks, such as picking up an object and then placing it to a proper place, as well as multi-agent tasks, such as pushing with two arms [1], [9], [10]. These two types of tasks are difficult for two-stage affordance methods [3], [5], [1] because they need different pre-defined data collection for each human-defined primitive action. Also, by unifying all interactions as contacts, our method can effectively

represent both agent-to-object (A2O) and object-to-object (O2O) interactions, which is hard for other methods.

To test our method, we conducted experiments on eight representative robot tasks, including articulated object manipulation, object pick-and-place and dual arm collaboration tasks. The results showed that our method outperformed all the baselines, including those of RL and the two-stage affordance methods, and can successfully transfer to the real world. To the best of our knowledge, we are the first to investigate end-to-end affordance learning for robotic manipulation.

II. Related Work

A. Robotic Manipulation Policy Learning

The recent simulators and benchmarks have boosted the development of manipulation policy learning methods [11], [12], [13], [14]. For rigid object manipulation, there are already robust algorithms handling tasks such as grasping [15], [16], [1], planar pushing [17], [18] and object hanging [19]. However, it is yet difficult to manipulate articulated objects with multiple parts despite various attempts to approach this problem from different perspectives. For example, UMPNet [20] and VAT-Mart [5] utilized visual observation to directly propose action sequence, while some other studies [21], [22], [23] achieved robust and adaptive control through model prediction. The multi-stage and multi-agent manipulation settings are also challenging for current methods [24], [25], [14].

B. Visual Actionable Affordance Learning

Till now, several studies have demonstrated the power of affordance representation on manipulation [3], [5], grasping [2], [26], [1], [8], scene classification [27], [28], scene understanding [29], [30] and object detection [31]. The semantic information in affordance is instructive for manipulation. Some prior affordance learning processes for manipulation, such as Where2Act [3], VAT-Mart [5], AdaAfford [4] and VAPO [1], have two training stages. Specifically, they need to first collect interacting data to pretrain the affordance, and then train the policy based on the affordance. For methods [1], [2], [32] which train affordance and policy simultaneously, however, their affordance learning relies on human demonstration. Unlike them, our method requires neither pre-defined data collection process for different primitive actions / tasks nor any additional human annotations.

C. Comparison with Related Works

The related works mentioned above studied robotic manipulation in different problem settings, the difference includes observation, annotation and applicable scenarios. It is hard to compare these works given their distinct settings. Specifically, in the door-opening task, Maniskill [12] utilizes expert demonstrations for imitation learning. However, expert demonstrations are difficult to obtain since they are usually collected by human. Affordance methods, such as Where2Act [3], VAT-Mart [5], and VAPO [1] learn the affordance prior to its policy training and are not in an end-to-end fashion. They also output gripper pose as actions which in reality have no guarantee that the gripper can reach the position. Additionally, the related affordance studies mentioned here are designed for single-stage single-agent tasks, such as opening a door or grasping an object, which has no guarantee for multi-stage or multi-agent tasks. However, our method naturally allows RL policy to handle multi-stage tasks like picking-and-placing, and multi-agent tasks like collaborative pushing.

TABLE I
Comparison between our work and related works.

	W2A	VAT	MSkill	VAPO	UMP	**Ours**
No Demo	✓	✓			✓	✔
No Full Obs	✓	✓		✓	✓	✔
End-to-End			✓			✔
Multi-Stage						✔
Multi-Agent			✓			✔

Table I compares our method with five representative related works discussed above. Listed works are Where2Act (W2A) [3], VAT-Mart (VAT) [5], Maniskill (MSkill) [12], VAPO [1] and UMPNet (UMP) [20]. "**No Demo**" means the method does not need any expert demonstrations, such as human-collected trajectories, pre-defined primitive actions and human-designed interaction poses. "**No Full Obs**" indicates the method does not need state observations of objects such as the coordinate of door handle, since the accurate state of an object is difficult to obtain in real world. "**End-to-End**" signifies the method trains the policy in an end-to-end fashion, *i.e.*, no multiple training stages are involved and the actions of the policy can be directly applied to agents. "**Multi-Stage**" infers the method can complete multi-stage tasks which the agent needs to finish multiple dependent tasks sequentially. "**Multi-Agent**" suggests the method can be adapted to multi-agent tasks where agents need to cooperate one another to finish the work.

III. Methods

A. Method Overview

Visual-based RL is increasingly valued on robotic manipulation tasks, especially those requiring the agent to manipulate different objects with a single policy. Meanwhile, recent studies [33], [34], [8] identified the difficulty of learning observation encoders by RL from high-dimensional inputs such as point clouds and images. In our framework, we tackled this critical problem by exploiting underlying information through a process called "Contact Prediction".

In manipulation settings, contact is the fundamental way humans interact with an object. We believe that physical contact positions during interactions reflect the understanding of crucial semantic information about the object (*e.g.*, a human grasp a handle to open a door because the handle provides the position to apply force).

We proposed a novel end-to-end RL learning framework for manipulating 3D objects. As shown in Fig. 2, our framework is comprised of two parts. 1) *Manipulation Module* (*MA* Module) is a RL framework which uses the affordance map predicted by a *Contact Predictor* (*CP*) as an additional

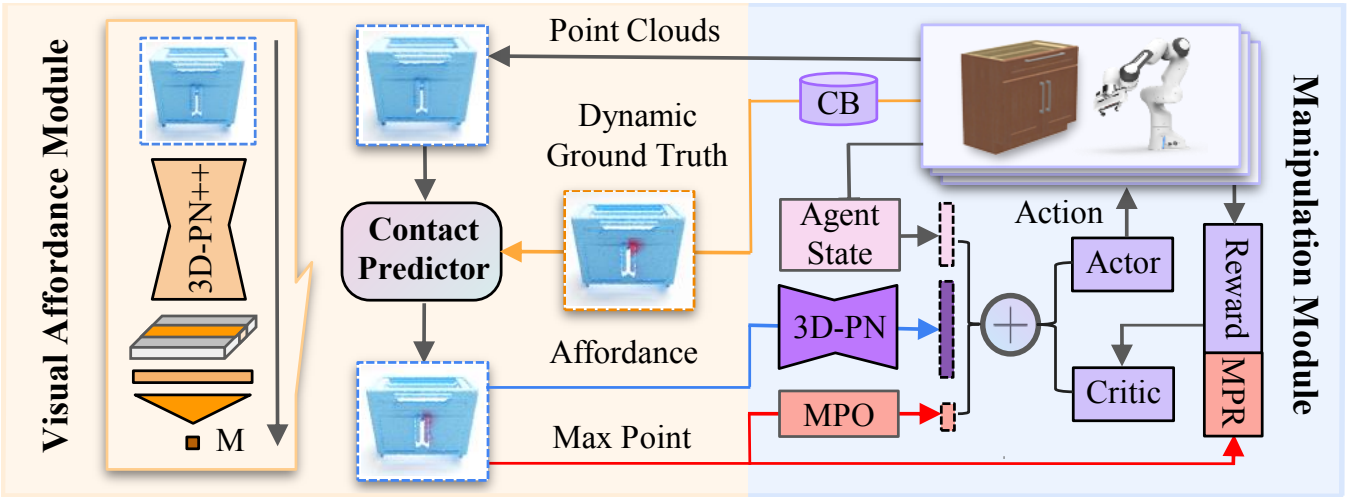

Fig. 2. **Training Pipeline of End-to-End Affordance Learning.** Our pipeline contains two main modules: *Manipulation Module* (*MA* Module) generating interaction trajectories and *Visual Affordance Module* (*VA* Module) learning to generate per-point affordance map M based on the real-time point cloud. The *Contact Predictor* (CP), shared across two modules, serves as a bridge between them: 1) *MA* Module uses the affordance map (indicated by the blue arrow) and *Max-affordance Point Observation* (*MPO*) (indicated by the upper red arrow) predicted by the CP as a part of the input observation. A *Max-affordance Point Reward* (*MPR*) feedback (indicated by the lower red arrow) is also incorporated in training *MA* Module; 2) *MA* Module maintains a *Contact Buffer* (*CB*) by collecting collision information and generating *Dynamic Ground Truth* (*DGT*) (indicated by the orange arrow), where *VA* Module uses the *DGT* as the target for training *CP*.

Algorithm 1 End-to-End Affordance Learning.

Require: E: the environment, CB: current contact buffer, RL: RL pipeline, i: current timestep.
Ensure: a: an action generated by RL pipeline

```
c ← collectContact(E, RL)            ▷ Get contact
CB ← insert(CB, c)                   ▷ Insert new contact into CB
PC ← getPointCloud(E)                ▷ Point Cloud
if i%k = 0 then                      ▷ Update CP every k timesteps
    DGT ← getMap(PC, CB)             ▷ Dynamic Ground Truth
    CP ← update(CP, DGT)             ▷ Update CP network
end if
M ← CP(PC)                           ▷ Get Affordance map
RL ← train(RL, E, M)                 ▷ Update RL network
a ← RL(E, M)                         ▷ Get the action
return a
```

observation and reward signal; 2) *Visual Affordance Module* (*VA* Module) is a per-point scoring network, which uses the contacts collected from RL training process as the *Dynamic Ground Truth* (*DGT*) to indicate the position of interaction.

Concretely, at every time-step t, the *MA* Module outputs an action a_t based on the robotic arm state s_t (*i.e.,* the angle and angular velocity of each joint) and the affordance map M_t predicted by the *VA* Module. After each time-step t, the contact position in RL training is inserted to the *Contact Buffer* (*CB*). After each k time-steps, we integrate the data in *CB* to generate the per-point score as the *DGT* to update the *VA* Module.

B. *Visual Affordance Module:* Contact as Prior

During robotic manipulation, physical contacts naturally happen between agent and object, or object and object. As contacts do not relate to any human-defined primitive action such as pull or push, the contact position is a general representation, providing visual prior for manipulation.

The RL training pipeline in Manipulation Module (*MA* Module) continuously interacts with the environment to collect 1) the partial point cloud observation $\mathscr{P}$, 2) the contact position under object coordinate. Based on this information, we measure how likely a contact between agent and object (A2O) or between object and object (O2O) is going to happen by the per-point contact *frequency* as the affordance during the current RL training. The Visual Affordance Module (*VA* Module) then learn to predict the per-point *frequency*. The training details of *VA* Module is as follow.

Input: Following the prior studies [4], [5], [3], the input for *VA* Module contains a partial point cloud observation $\mathscr{P}$.

Output: The output of the *VA* Module is a per-point affordance map M for each of the point from the input. The map contains A2O affordance and O2O affordance.

Network Architecture: The prediction is completed by a *Contact Predictor* (*CP*) that uses a PointNet++ [35] to extract a per-point feature $f \in \mathbb{R}^{128}$ from point cloud observation $\mathscr{P}$, the feature f is then fed through a Multi-Layer Perceptron (MLP) to predict the per-point actionable affordance [3].

Dynamic Ground Truth: To connect the RL pipeline in *MA* Module with the *VA* Module, we use a *Contact Buffer* *CB* to keep l record of history contact points, and to compute the *DGT*. Specifically, each object in the training set has a corresponding *CB*, it records contact positions on the object. To maintain the buffer size, the buffer randomly evicts one record whenever a new record of contact event is inserted. To provide training ground truth for *CP*, we calculate the *DGT* by first calculating the number of contacts within radius r from each point on the object point cloud, and then applying normalization to obtain *Dynamic Ground Truth DGT*. The

normalization is as follow:

$$DGT_t^i(p) = \frac{\sum_{q \in CB_t^i} I(|p-q|_2 < r)}{\max_{p'} \sum_{q \in CB_t^i} I(|p'-q|_2 < r) + \varepsilon}, \quad (1)$$

where DGT_t^i indicates the *Dynamic Ground Truth* for object i at time-step t, CB_t^i is the corresponding *Contact Buffer*. For A2O affordance, we only consider contacts between the end-effector and the object, while for O2O affordacne, we only consider contacts between objects.

Training: The CP is updated with DGT_t^i as below:

$$CP_t^* = \arg\min_{CP} \sum_i sr_t^i \left|\left| \sum_{p \in \mathscr{P}^i} CP(p|\mathscr{P}^i) - DGT_t^i(p) \right|\right|_2 \quad (2)$$

where sr_t^i is the current manipulation success rate on object i, $\mathscr{P}^i$ is i-th object's point cloud and CP_t^* is the optimal CP .

C. Manipulation Module: Affordance as Guidance

Manipulation Module (*MA* Module) is an RL framework able to learn to manipulate objects from scratch. Different from previous methods [5], [3], [12], our *MA* Module takes advantage of both the reward and observation generated by the *VA* Module.

Input: The input for *MA* Module includes, 1) a point cloud $\mathscr{P}$ of the real-time environment [3], [5]; 2) an affordance map M generated by *VA* Module; 3) the state s of the robotic arm. The state s consists of position/angle and velocity of each joint of the robotic arm; 4) a state-based *Max-affordance Point Observation* (*MPO*), which indicates the point with the maximum affordance score on $\mathscr{P}$.

Output: The output of the *MA* Module is an action a, which is then executed by the robotic arm. In our setting, the RL policy controls each joint of the robotic arm directly.

Reward from Affordance: We introduce the *Max-affordance Point Reward* (*MPR*) into our pipeline, where a point on the point cloud with maximum affordance score predicted by the *VA* Module is selected as the guidance for learning *MA* Module. We use the distance between robot end-effector and this selected point to compute an additional reward in the RL process. We found this reward from affordance could benefit the RL training thus improve the overall performance.

Network Architecture: The policy of the *MA* Module is a neural network π_θ with learnable parameter θ. The network consists of a PointNet [36] and a MLP. The PointNet extracts feature $f \in \mathbb{R}^{128}$ from the point cloud $\mathscr{P}$, affordance map M and additional masks m. The extracted feature f is then concatenated with s and fed to the MLP to obtain actions.

Training: We use Proximal Policy Optimization (PPO) algorithm [37] to train the *MA* Module. To improve the training efficiency by exploiting the high parallelism of our simulator, we deploy k different objects in the simulator, each object is replicated n times and given to one or two robotic arms. Hence, there are a total of $k \times n$ environments, each with a robotic arm (or two robotic arms in our multi-agent tasks) interacting with an object, as shown in Fig. 3.

IV. Experiment

A. Task Description

To evaluate our method, we designed three types of manipulation tasks: single-stage, multi-stage and multi-agent. In all tasks, a robotic arm or two robotic arms are required to complete a specific manipulation task on different objects.

The first type of tasks are single-stage tasks:

Close Door: A door is initially open to a specific angle. The agent need to close the door completely. We increase the difficulty of this task by applying an additional force on the door attempting to keep the door to the initial position and doubling the friction of the hinge.

Open Door: A door is initially closed. The agent need to open the door to a specific angle. This task can test whether the agent learn to leverage key parts like the handle to open the door, which is challenging.

Push Drawer: A drawer is initially open to a specific distance. similar to *close door*, the agent need to close the drawer on a cabinet completely.

Pull Drawer: A drawer is initially closed, similar to *open door*, the agent need to open the drawer to a specific distance.

Push Stapler: A stapler is on the desk, initially open. The agent need to push on the stapler and close it.

Lift Pot Lid: A pot is on the floor with its lid on. The agent need to lift the lid.

To show the agent can learn a policy in a multi-stage task, we use the pick-and-place task as follow:

Pick and Place: An object should be picked up and then placed on a table that already have several random objects on it, both the table and objects are randomly selected from the given datasets. The agent need to place the object stably on the table without collision.

To show our method can be generalized to multi-agent settings, we use the dual-arm-push task as follow:

Dual Arm Push: Two robotic arms need to be controled to push a chair to a specific distance and prevent the chair from falling over.

The reward designs are listed on our website.

B. Dataset and Simulator

We performed our experiments using the Isaac Gym simulator [38]. We used Franka Panda robot arm as the agent for all tasks. Our training and testing data are the subset of the PartNet-Mobility dataset [39] and VAPO dataset [1]. For tasks **Close Door** and **Open Door**, we use four types of objects in the StorageFurniture category: *one door left*, *one door right*, *two door left* and *two door right*. For tasks **Pull Drawer** and **Push Drawer**, we use two types of objects in the StorageFurniture category: *drawer without door* and *drawer with door*. For tasks **Push Stapler** and **Lift Pot Lid**, we chose all *Stapler* and *Pot* from PartNet-Mobility dataset. For task **Pick and Place**, we chose three representative categories of objects from VAPO dataset to pick. We also selected four types of different tables: *Round Table*, *Triangle Table*, *Square Table* and *Irregular Table* while three daily items from PartNet-Mobility dataset were placed randomly on the

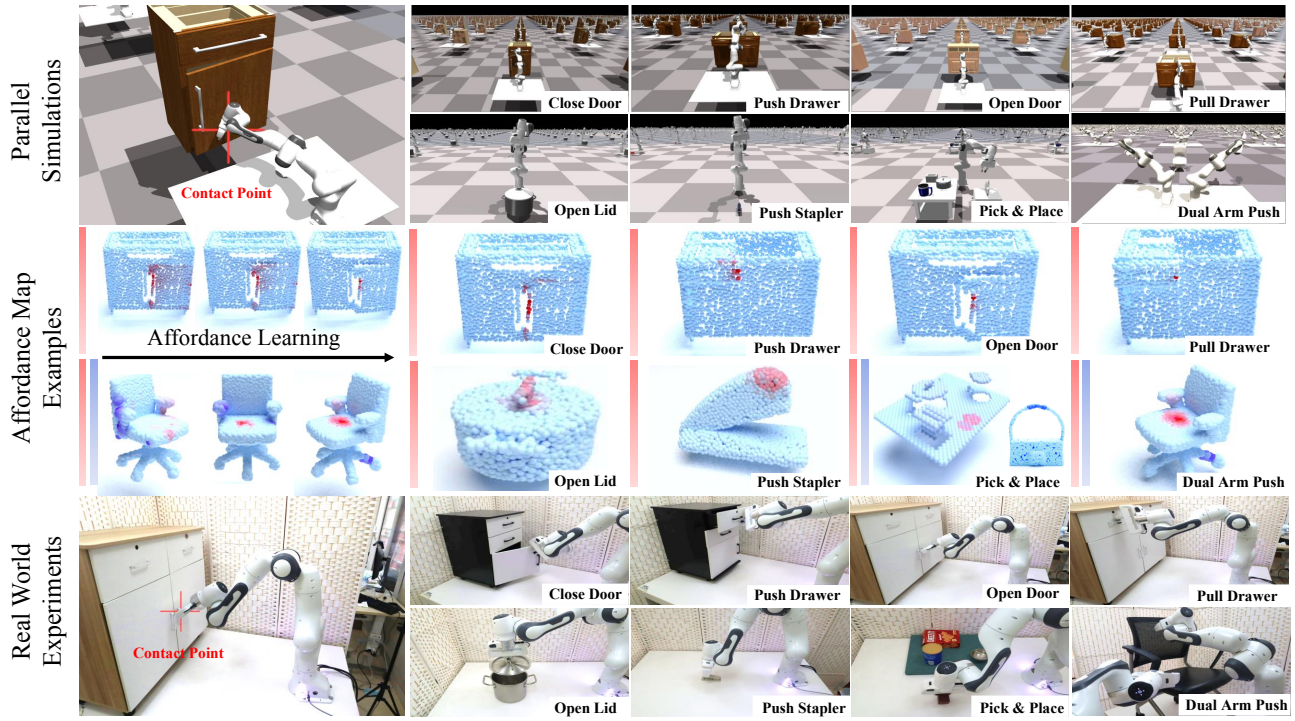

Fig. 3. **Experiment Settings and Affordance Learning Visualization.** Top: the tasks settings in simulators. Middle: the change in affordance maps during end-to-end training and the final affordance map examples. Bottom: the real-world experiments.

table. For task **Dual Arm Push**, we chose 60 *Chairs* from PartNet-Mobility dataset.

C. Baselines and Ablations

We compared our method with seven baselines:

- Where2act [3]: the original method only generates single-stage interaction proposals. To use this method as a baseline in our tasks, we implemented a multi-stage Where2act baseline (up to six steps). The object is gradually altered by pushing or pulling interactions produced by Where2act until the task is completed or the maximum number of steps have been taken. Unlike our own setting, this baseline used a flying gripper instead of a robotic arm.
- VAT-Mart [5]: We followed the implementation of VAT-Mart [5]. We implemented this method in our environment as a baseline with a flying gripper.
- RL: we used a point cloud based PPO as our baseline.
- RL+Where2act: we replaced the Contact Predictor with a pre-trained Where2act model that can output a per-point actionable score. The parameters in the Where2act model is frozen when training *MA* Module.
- RL+O2OAfford and RL+O2OAfford+Where2act: Similar to RL+Where2act, we replaced the O2O affordance map in our method with the map produced by a pre-trained O2OAfford [7] model.
- MAPPO: we used a point cloud based multi-agent RL algorithm (MARL): MAPPO [40] as our baseline.
- Multi-Task RL [37]: we adapted PPO to the multi-task setting by providing the one-hot task ID as input. To make this method comparable on the test set, both the test set and the training set were used in training process. So this is an oracle baseline.

To further evaluate the importance of different components of our method, we conducted ablation study by comparing our method with five ablations:

- Ours w/o MPR: ours without the max-point reward.
- Ours w/o MPO: ours without the max-point obversation.
- Ours w/o E2E: our method trained by a two-stage procedure. The *VA* Module is trained upon a fixed pretrained *MA* Module. The *MA* Module is then fine-tuned on the freezed *VA* Module.
- Ours w/o A2O Map: our method without agent-to-object affordance map in multi-stage tasks.
- Ours w/o O2O Map: our method without object-to-object affordance map in multi-stage tasks.

D. Evaluation Metrics

For each task, we trained all methods (ours, baselines and ablations) on the training set and saved checkpoints every 3200 time-steps within 160,000 total time-steps. Training curves show that all algorithms we compared have converged and have no signs of further improvement if we increase the total time step. After training, we chose the checkpoint with the largest average success rate on training set for comparison, the method was tested on eight different random seeds. We adpoted two metrics to measure the performance:

- Average Success Rate (ASR): The ASR is the average of the algorithm's success rate on all objects in the training / testing dataset.

TABLE II

QUANTITATIVE RESULTS OF SINGLE-STAGE TASKS. (MORE RESULTS ON OUR WEBSITE.)

Methods \ Tasks	Open Door				Pull Drawer				Push Stapler				Open Pot Lid			
	ASR		MP		ASR		MP		ASR		MP		ASR		MP	
	train	test	train	test	train	test	train	test	train	test	train	test	train	test	train	test
Where2act	22.8	14.1	6.8	8.3	19.0	12.9	2.3	0.0	16.4	14.4	13.0	13.0	10.5	5.4	8.7	4.3
VAT-Mart	23.2	21.9	31.8	33.3	5.5	5.1	0.0	0.0	21.9	20.9	17.4	13.0	27.4	21.5	17.4	17.4
Multi-task RL	18.8	9.2	11.4	5.0	0.1	2.4	0.0	2.8	34.9	30.2	30.4	26.1	35.2	32.6	21.7	17.4
RL	21.5	5.5	22.7	0.0	23.1	22.4	19.6	19.5	45.5	40.6	34.8	30.4	32.5	28.6	21.7	21.7
RL+Where2act	20.5	8.0	19.3	9.4	25.2	22.2	24.4	21.9	48.9	45.2	39.1	34.8	38.2	30.6	26.1	21.7
Ours	**52.9**	**32.6**	**61.4**	**41.7**	59.7	**58.6**	62.8	**63.3**	**69.5**	**53.2**	**47.8**	**39.1**	**49.5**	**44.6**	**34.8**	**30.4**
Ours w/o MPO	48.0	23.8	50.0	16.7	41.9	42.5	38.6	43.8	60.6	52.5	43.5	**39.1**	44.2	40.7	**34.8**	**30.4**
Ours w/o MPR	28.2	8.4	29.5	8.3	**62.3**	44.0	**65.9**	43.8	50.8	39.9	39.1	30.4	44.8	40.1	30.4	26.1
Ours w/o E2E	21.2	12.4	20.5	8.3	57.7	57.3	61.1	61.7	40.2	36.6	39.1	34.8	32.1	30.6	30.4	26.1

TABLE III

QUANTITATIVE RESULTS OF PICK-AND-PLACE.

Methods \ Metrics	ASR		MP	
	train	test	train	test
RL	25.2	22.1	19.2	11.5
RL+O2OAfford	26.1	22.2	19.2	11.5
RL+Where2act	28.6	23.5	23.1	15.4
RL+O2OAfford+Where2act	30.5	26.2	23.1	15.4
Ours	**46.5**	**39.2**	**30.7**	**26.9**
Ours w/o A2O Map	26.7	22.3	23.1	19.2
Ours w/o O2O Map	31.9	26.2	23.1	15.4
Ours w/o MPO	40.1	30.2	19.2	15.4
Ours w/o MPR	36.2	33.5	**30.7**	23.1
Ours w/o E2E	30.2	21.4	26.9	19.2

TABLE IV

QUANTITATIVE RESULTS OF DUAL-ARM-PUSH.

Methods \ Metrics	ASR		MP	
	train	test	train	test
MAPPO	7.8	9.0	0.0	0.0
RL	37.2	36.1	36.4	31.3
Multi-task RL	51.6	52.9	54.5	56.3
Ours	83.9	78.5	90.9	93.8
Ours w/o MPO	**95.9**	**96.3**	**100.0**	**100.0**
Ours w/o MPR	63.9	55.3	63.6	56.3
Ours w/o E2E	53.5	55.9	56.8	50.0

- Master Percentage (MP): We assume a policy is considered "stable" on an object if it has a success rate greater than 50% on that object. The master percentage is the percentage of objects which the algorithm is "stable" on. If an algorithm achieves a success rate over 50% on an object,, it is expected to succeed within two trials.

More results can be found on our website.

E. Baseline Comparision and Ablation Study

From Tables II and III, the results of Where2act and RL show the visual affordance can improve the RL performance. However, our method achieves a more significant improvement over baselines in both training and testing sets. In dual-arm-push, as Table IV shows, our method outperforms both RL and MARL methods.

From all tables, we see the MPO, MPR and E2E components play important roles in our method except that E2E on dual-arm-push. The potential reason is that the predicted max affordance point on the object is changing during object movement, which may influence the RL training. This may be something worth looking into in the future.

Fig. 3 shows the change in affordance maps during end-to-end training and examples of final affordance maps. As training progresses, the affordance map gradually becomes more focused. More results can be found on our website.

F. Real-world Experiment

We used a digital twin system [41] for real-world experiment: The training process was in simulation, we then used some unseen objects to evaluate our method in real world. The input of the agent has two folds: 1) point cloud input from simulator, 2) agent state input from real world. The actions of the agent were computed upon the combination of the two input sources, and then were applied to the robotic arms both in the simulator and the real world. The experiment settings are shown in Fig.3. Experiments show that our trained model can successfully transfer to the real world. The video and more details can be found on our website.

V. CONCLUSION

To the best of our knowledge, this the first work that proposes an end-to-end affordance RL framework for robotic manipulation tasks. In RL training, affordance can improve the policy learning by providing additional observation and reward signals. Our framework automatically learns affordance semantics through RL training without human demonstration or other artificial designs dedicated to data collection. The directness of our method, together with the superior performance over strong baselines and the wide range of applicable tasks, has demonstrated the effectiveness of learning from contact information. We believe our work could potentially open a new way for future RL-based manipulation developments.

ACKNOWLEDGEMENT

This project was supported by National Key R&D Program of China (2022ZD0114900), Beijing Municipal Science & Technology Commission (Z221100003422004) and the National Natural Science Foundation of China (No. 62136001). We would like to thank Hongchen Wang, Ruihai Wu, Yan Zhao and Yicheng Qian for the helpful discussion and baseline implementation, and Ruimin Jia for suggestions in paper writing.

REFERENCES

[1] J. Borja-Diaz, O. Mees, G. Kalweit, L. Hermann, J. Boedecker, and W. Burgard, "Affordance learning from play for sample-efficient policy learning," in *ICRA*. IEEE, 2022, pp. 6372–6378.

[2] P. Mandikal and K. Grauman, "Learning dexterous grasping with object-centric visual affordances," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[3] K. Mo, L. J. Guibas, M. Mukadam, A. Gupta, and S. Tulsiani, "Where2act: From pixels to actions for articulated 3d objects," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 6813–6823.

[4] Y. Wang, R. Wu, K. Mo, J. Ke, Q. Fan, L. Guibas, and H. Dong, "Adaafford: Learning to adapt manipulation affordance for 3d articulated objects via few-shot interactions," *arXiv preprint arXiv:2112.00246*, 2021.

[5] R. Wu, Y. Zhao, K. Mo, Z. Guo, Y. Wang, T. Wu, Q. Fan, X. Chen, L. J. Guibas, and H. Dong, "Vat-mart: Learning visual action trajectory proposals for manipulating 3d articulated objects," in *ICLR*. OpenReview.net, 2022.

[6] Y. Zhao, R. Wu, Z. Chen, Y. Zhang, Q. Fan, K. Mo, and H. Dong, "Dualafford: Learning collaborative visual affordance for dual-gripper object manipulation," *arXiv preprint arXiv:2207.01971*, 2022.

[7] K. Mo, Y. Qin, F. Xiang, H. Su, and L. Guibas, "O2O-Afford: Annotation-free large-scale object-object affordance learning," in *Conference on Robot Learning (CoRL)*, 2021.

[8] Y.-H. Wu, J. Wang, and X. Wang, "Learning generalizable dexterous manipulation from human grasp affordance," *arXiv preprint arXiv:2204.02320*, 2022.

[9] A. Lobbezoo, Y. Qian, and H.-J. Kwon, "Reinforcement learning for pick and place operations in robotics: A survey," *Robotics*, vol. 10, no. 3, p. 105, 2021.

[10] D. R. Vyas, A. Markana, and N. Padhiyar, "Robotic grasp synthesis using deep learning approaches: a survey," in *Mathematical Modeling, Computational Intelligence Techniques and Renewable Energy*. Springer, 2021, pp. 117–130.

[11] F. Xiang, Y. Qin, K. Mo, Y. Xia, H. Zhu, F. Liu, M. Liu, H. Jiang, Y. Yuan, H. Wang *et al.*, "Sapien: A simulated part-based interactive environment," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 097–11 107.

[12] T. Mu, Z. Ling, F. Xiang, D. Yang, X. Li, S. Tao, Z. Huang, Z. Jia, and H. Su, "Maniskill: Generalizable manipulation skill benchmark with large-scale demonstrations," in *NeurIPS Datasets and Benchmarks*, 2021.

[13] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine, "Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning," in *Conference on Robot Learning*. PMLR, 2020, pp. 1094–1100.

[14] Y. Chen, Y. Yang, T. Wu, S. Wang, X. Feng, J. Jiang, S. M. McAleer, H. Dong, Z. Lu, and S.-C. Zhu, "Towards human-level bimanual dexterous manipulation with reinforcement learning," *arXiv preprint arXiv:2206.08686*, 2022.

[15] H. Cao, H.-S. Fang, W. Liu, and C. Lu, "Suctionnet-1billion: A large-scale benchmark for suction grasping," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 8718–8725, 2021.

[16] M. Breyer, J. J. Chung, L. Ott, S. Roland, and N. Juan, "Volumetric grasping network: Real-time 6 dof grasp detection in clutter," in *Conference on Robot Learning*, 2020.

[17] J. Li, W. S. Lee, and D. Hsu, "Push-net: Deep planar pushing for objects with unknown physical properties," *Robotics: Science and Systems XIV*, 2018.

[18] K.-T. Yu, M. Bauza, N. Fazeli, and A. Rodriguez, "More than a million ways to be pushed. a high-fidelity experimental dataset of planar pushing," in *2016 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2016, pp. 30–37.

[19] Y. You, L. Shao, T. Migimatsu, and J. Bohg, "Omnihang: Learning to hang arbitrary objects using contact point correspondences and neural collision estimation," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 5921–5927.

[20] Z. Xu, H. Zhanpeng, and S. Song, "Umpnet: Universal manipulation policy network for articulated objects," *IEEE Robotics and Automation Letters*, 2022.

[21] B. Abbatematteo, S. Tellex, and G. Konidaris, "Learning to generalize kinematic models to novel objects," in *CoRL*, ser. Proceedings of Machine Learning Research, vol. 100. PMLR, 2019, pp. 1289–1299.

[22] A. Jain and S. Niekum, "Learning hybrid object kinematics for efficient hierarchical planning under uncertainty," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 5253–5260.

[23] B. Eisner, H. Zhang, and D. Held, "Flowbot3d: Learning 3d articulation flow to manipulate articulated objects," *arXiv preprint arXiv:2205.04382*, 2022.

[24] Y. Zhu, J. Tremblay, S. Birchfield, and Y. Zhu, "Hierarchical planning for long-horizon manipulation with geometric and symbolic scene graphs," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 6541–6548.

[25] A. Mandlekar, D. Xu, R. Martín-Martín, S. Savarese, and L. Fei-Fei, "Learning to generalize across long-horizon tasks from human demonstrations," *arXiv preprint arXiv:2003.06085*, 2020.

[26] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 705–724, 2015.

[27] M. Dixit, S. Chen, D. Gao, N. Rasiwasia, and N. Vasconcelos, "Scene classification with semantic fisher vectors," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2974–2983.

[28] L. Zhang, X. Zhen, and L. Shao, "Learning object-to-class kernels for scene classification," *IEEE Transactions on image processing*, vol. 23, no. 8, pp. 3241–3253, 2014.

[29] S. Fowler, H. Kim, and A. Hilton, "Human-centric scene understanding from single view 360 video," in *2018 International Conference on 3D Vision (3DV)*. IEEE, 2018, pp. 334–342.

[30] C. Ye, Y. Yang, R. Mao, C. Fermüller, and Y. Aloimonos, "What can i do around here? deep functional scene understanding for cognitive robots," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 4604–4611.

[31] T.-T. Do, A. Nguyen, and I. Reid, "Affordancenet: An end-to-end deep learning approach for object affordance detection," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 5882–5889.

[32] T. Nagarajan, C. Feichtenhofer, and K. Grauman, "Grounded human-object interaction hotspots from video," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 8688–8697.

[33] M. Laskin, A. Srinivas, and P. Abbeel, "CURL: contrastive unsupervised representations for reinforcement learning," in *ICML*, ser. Proceedings of Machine Learning Research, vol. 119. PMLR, 2020, pp. 5639–5650.

[34] A. Stooke, K. Lee, P. Abbeel, and M. Laskin, "Decoupling representation learning from reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2021, pp. 9870–9879.

[35] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *NIPS*, 2017, pp. 5099–5108.

[36] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[37] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[38] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, and G. State, "Isaac gym: High performance GPU based physics simulation for robot learning," in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

[39] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su *et al.*, "Shapenet: An information-rich 3d model repository," *arXiv preprint arXiv:1512.03012*, 2015.

[40] C. Yu, A. Velu, E. Vinitsky, Y. Wang, A. Bayen, and Y. Wu, "The surprising effectiveness of ppo in cooperative, multi-agent games," *arXiv preprint arXiv:2103.01955*, 2021.

[41] K. Xia, C. Sacco, M. Kirkpatrick, C. Saidy, L. Nguyen, A. Kircaliali, and R. Harik, "A digital twin to train deep reinforcement learning agent for smart manufacturing plants: Environment, interfaces and intelligence," *Journal of Manufacturing Systems*, vol. 58, pp. 210–230, 2021.