PreAfford: Affordance-based Pre-grasping across Objects and Scenes

Kairui Ding[1,2], Boyuan Chen[2], Ruihai Wu[3], Yuyang Li[4], Zongzheng Zhang[1], Huan-ang Gao[1], Siqi Li[5], Yixin Zhu[4], Guyue Zhou[1,6], Hao Dong[3], and Hao Zhao[1]

https://air-discover.github.io/PreAfford

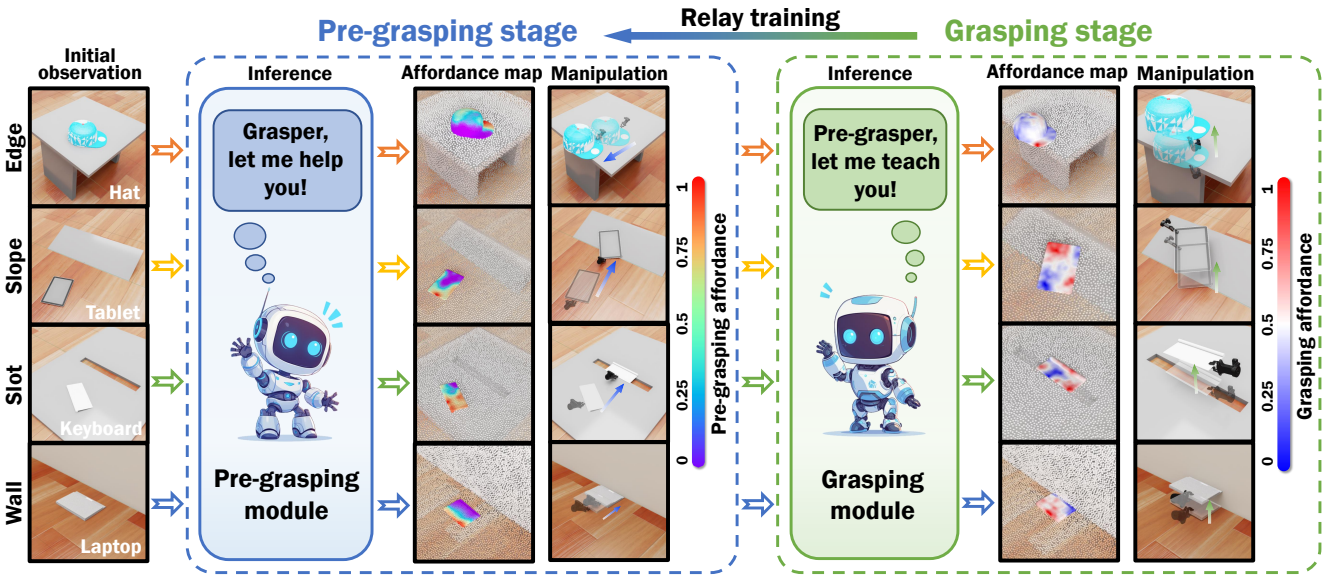

Fig. 1: **Demonstration of PreAfford.** Adopting a *relay training* paradigm, two successive modules collaborate to handle the grasping tasks of ungraspable objects. Considering different environmental features (edge, slope, slot and wall), the pre-grasping module proposes a pre-grasping strategy to facilitate a successful grasp, while the grasping module generates rewards to train (depicted as *teach* in the figure) the pre-grasping module. The two color bars correspond to the pre-grasping and grasping phases, respectively. They indicate the affordance values, with higher values suggesting better conditions for interaction.

***Abstract*— Robotic manipulation of ungraspable objects with two-finger grippers presents significant challenges due to the paucity of graspable features, while traditional pre-grasping techniques, which rely on repositioning objects and leveraging external aids like table edges, lack the adaptability across object categories and scenes. Addressing this, we introduce PreAfford, a novel pre-grasping planning framework that utilizes a point-level affordance representation and a relay training approach to enhance adaptability across a broad range of environments and object types, including those previously unseen. Demonstrated on the ShapeNet-v2 dataset, PreAfford significantly improves grasping success rates by 69% and validates its practicality through real-world experiments. This work offers a robust and adaptable solution for manipulating ungraspable objects.**

I. Introduction

Consider a robotic arm equipped with a two-finger gripper attempting to grasp a mobile phone lying flat on a surface (Fig. 2(a)). The close contact between the phone and the table results in a scarcity of graspable features, leading to an *ungraspable* scenario [1]. Ungraspable situations occur when an object's 6-DoF pose relative to its surroundings hinders direct grasping. Such challenges are common not only with flat objects like mobile phones, keyboards, and scissors but also with items such as hats or inverted bowls placed on a table, as the graspable edges are in intimate contact with the supporting surface.

Drawing from the experience of human object manipulation, a viable strategy is to move the flat objects to table edge, which enables an effective grasp from the hanging off part. This solution, known as a *pre-grasping* manipulation, plays a crucial role as a preliminary step for grasping manipulation in ungraspable situations [2].As shown in Fig. 2(b-e), different environmental features (a.k.a. extrinsic dexterity [3, 4]), can be utilized to transform the object into a favorable configuration for a successful grasp.

Studies have demonstrated that pre-grasping manipulations, including pushing [7], rotating [1, 6], and sliding [5] [10], can significantly improve the success rate of grasping manipulation [8]. However, three limitations still wait to be addressed: (i) **Adaptability.** Previous research mainly

* We thank DISCOVER Robotics for providing the hardware setups used in this research.

[1] Institute for AI Industry Research (AIR), Tsinghua University.
[2] Xingjian College, Tsinghua University.
[3] School of Computer Science, Peking University.
[4] Institute for Artificial Intelligence, Peking University.
[5] College of Control Science and Engineering, Zhejiang University.
[6] School of Vehicle and Mobility, Tsinghua University.

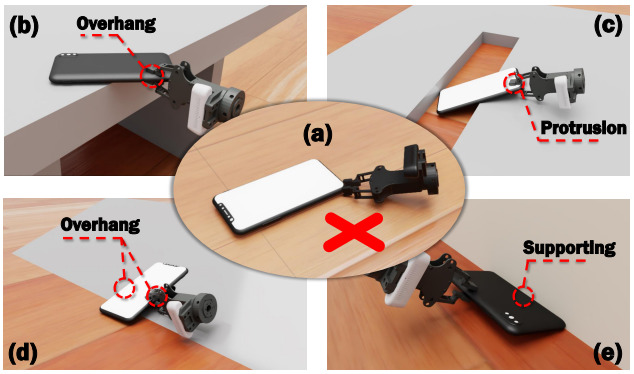

Fig. 2: Demonstrations for pre-grasping leveraging environmental features: (a) Ungraspable situation of lying flat on floor, (b) Side-grasping an overhanging part, (c) Grasping an angled part protruding from a slot, (d) Grasping the middle of the phone suspended on a slope's foot, and (e) Pinning the phone against a wall and grasping from the opposite side.

focuses on task-specified settings, which rely heavily on manually programmed criterion for pre-grasping manipulation success, and cannot adapt across object categories and environments [1, 2, 5, 7, 11]. (ii) **Deployablity.** Methods for pre-grasping planning often has difficulty to transfer to real-world experiments, due to their reliance on privileged information from simulators or the need to customize representations for specific experimental setups. [5, 6, 6, 8]. (iii) **Compatibility.** To reduce the cost of robotic control, pre-grasping manipulation should be skipped on easy-to-grasp objects. But the mechanism for checking the necessity of pre-grasping is often not enabled in previous work [1, 2, 5–7, 11, 12].

As none of the previous work has achieved adaptability, deployability, and compatibility simultaneously, we present `PreAfford` to fulfill the requirements. The adaptability of `PreAfford` mainly originates from a novel relay training paradigm along with a dual-module framework, resulting in an intuitive and robust reward function - the improvement in grasping success likelihood - which is robustly evaluated by the grasping module. High deployability and real-world adaptability are enabled by adopting a point-level affordance visual representation [12–14], which only takes RGB-D data as input. To enable a direct grasp on easy-to-grasp objects for compatibility, a pre-grasping necessity check is introduced at the beginning of our inference procedure.

By training and testing on a large-scale offline dataset based on ShapeNet-v2 [15] across 5 scenes, simulations demonstrated that a `PreAfford` increases the grasping success rate by 69% on test object categories. The pre-grasping and grasping affordance maps, as illustrated in Fig. 5, reveal that our models have a deep understanding of object geometries and environmental features of the system. Furthermore, `PreAfford` exhibits the capability to choose a pre-grasping policy in unseen complex environments. The deployability of our framework has been validated through real-world experiments across 5 setups.

To sum up, our key contributions are:

- A novel, adaptive, and deployable pre-grasping framework compatible with easy-to-grasp objects.
- A robust relay training paradigm for pre-grasping manipulation.
- Point-level affordance representation enabling detailed geometry awareness and seamless deployment.
- Extensive validation of `PreAfford` in simulated and real-world settings, demonstrating emergent capabilities.

II. Related Work

A. Pre-Grasping Tasks

Pre-grasping, a concept inspired by the observation that humans often pre-manipulate objects [16, 17] – for instance, sliding flat objects to the edge of a table for easier grasping – necessitates that robot manipulators adjust the pose of the object *before* the final grasp [2, 7, 9, 11, 17, 18].

In Tab. I, we briefly review previous pre-grasping strategies. One line of research focuses on changing the pose of the object to a configuration that can be grasped more easily, primarily through rotation [5, 6, 19]. Other works concentrate on pre-grasping with extrinsic dexterity [3, 4, 8], which utilizes surrounding environmental features to assist in achieving an easier grasp [1, 2, 8]. Extensive research has also been conducted on cluttered objects on tables, which cannot be grasped due to overlapping but can be grasped by rearranging the objects [7, 9, 20].

The key aspect of pre-grasping research is the design of the reward function for a pre-grasping manipulation. Since the *graspability* of an object posed in a certain environmental configuration cannot be easily defined, the reward is typically approached in two ways. The first approach involves

TABLE I: **Comparison of Related Pre-Grasping Research.** This table presents a concise comparison of various pre-grasping studies, highlighting their key features and limitations. The last three columns assess each study's *adaptability* across object categories and environments, *compatibility* with existing grasping pipelines through an explicit procedure to skip the pre-grasping step, and *deployability* in real-world experiments without requiring customized shape representations [5] or identical setups to simulators [6].

Method	End Effector	Pre-grasping Manipulation	Method	Scenario	Adaptability	Compatibility	Deployability
Ren et al. [7]	Two-finger gripper	Pushing	DRL	Clustered objects	×	✓	✓
Sun et al. [1]	Spherical	Rotation	DRL	Cuboid in corner	×	×	✓
Kappler et al. [2]	Dexterous hand	Pushing	Data-driven approach	Cuboid on table	×	✓	×
Chen et al. [8]	Dexterous hand	finger contact	Learning-based framework	Ungraspable cases	✓	✓	×
Hang et al. [5]	Two-finger gripper	Sliding	Integrated planning	Thin objects on table	×	×	×
Chang et al. [6]	Two-finger gripper	Rotation	Opt. for payload	Transport tasks	×	×	×
Wang et al. [9]	Two-finger gripper	Pushing	DRL	Clustered objects	×	✓	✓
Ours	**Two-finger gripper**	**Pushing**	**Dual-module affordance map**	**Ungraspable cases**	✓	✓	✓

manually programming the reward using methods such as pre-defined goal regions [5] or specific pose transitions of the object and gripper [1, 11]. However, these methods are often limited to the specific task settings in the research and lack the ability to adapt to unseen object categories and environments. The second approach relies on the output of pre-trained neural networks to estimate the *graspability* of an object at a given pose [6–8]. However, prior knowledge from simulators, like signed distance functions [8] or exact object and environment geometry [6], is used in these studies, making it difficult to transfer to real-world experiments, which requires the deep understanding of scenes [21–24].

Furthermore, we found that an option to skip the pre-grasping procedure for objects that can be grasped directly with high confidence of success is often neglected [6–8], thus making the pre-grasping framework redundant for most daily graspable objects and increasing the cost of robotic control. No previous work has simultaneously achieved adaptability across object-environment configurations, compatibility with graspable situations, and deployability in real-world experiments.

B. Point-Level Affordance for Robotic Manipulation

Affordance, defined as the action possibilities associated with an object or environment to an agent [25–29], plays a crucial role in robotic manipulation. In point-level affordance learning, we learn dense affordance maps as the actionable visual representations to suggest action possibilities at every point in point clouds of 3D objects. Recent literature has applied point-level affordance learning to various scenarios [13, 14, 26, 30–36], providing dense and actionable information for downstream executions.

Numerous empirical investigations have demonstrated that point-level affordance exhibits a robust geometry-aware capability, as evidenced by its effective generalization across both within-category [12, 37] and inter-category [14, 30, 35, 38] scenarios, showcasing its adaptability to novel objects.

Our work further extends point-level affordance to guiding pre-grasping manipulations, which considers diverse geometries and diverse kinds of *environments*, demonstrating that point-level affordance handles more complicated scenarios with high accuracy and promising generalization ability.

III. Methodology

We formally define core concepts of `PreAfford` in Sec. III-A, describe the overall framework in Sec. III-B, elaborate the network architectures in Sec. III-C, introduce inference procedure in Sec. III-D and training losses in Sec. III-E, and show our approach to collect data in Sec. III-F.

A. Preliminaries

Pre-Grasping Tasks: The primary goal of pre-grasping manipulation is to alter the objects' poses to increase the likelihood of a successful grasp. We particularly focus on the exploitation of environmental features.

Recognizing that a single push action can achieve the repositioning of objects, we define a pre-grasping operation $\mathcal{P}$ as a push with offset $\Delta\vec{x}_1$ at contact point $\vec{p}_1$

$$\mathcal{P} = (\vec{p}_1, \Delta\vec{x}_1) = (x_1, y_1, z_1, \Delta x_1, \Delta y_1), \tag{1}$$

where $\vec{p}_1 = (x_1, y_1, z_1)$ and $\Delta\vec{x}_1 = (\Delta x_1, \Delta y_1)$ denote the contact point location and horizontal displacement, respectively. The pushing direction is by default set to be horizontal and the gripper remains closed.

In our research, the reward for a pre-grasping action is evaluated by the increase in the graspability score yielded by the grasping module, as shown in Eq. (7). We define a pre-grasping manipulation as *successful* if the score increases by more than 40%. Two safety-critical situations are considered as failure cases: (a) the object falling off the table, and (b) collision between the gripper and the slope or wall.

Scene-aware Grasping: Grasping tasks involve identifying an optimal manipulation, $\mathcal{G}$, based on environmental information $\mathcal{E}$ and object information $\mathcal{O}$. In contrast to object-centric grasping tasks, the environment plays a crucial role in scene-aware grasping, imposing constraints such as external visibility and kinematic feasibility [8, 34].

In `PreAfford`, $\mathcal{E}$ and $\mathcal{O}$ are dipicted as point clouds and a grasp $\mathcal{G}$ is defined by six parameters

$$\mathcal{G} = (p_2, \vec{\theta}_2) = (x_2, y_2, z_2, \alpha, \beta, \gamma). \tag{2}$$

where $p_2 = (x_2, y_2, z_2)$ is the contact point and $\vec{\theta}_2 = (\alpha, \beta, \gamma)$ is the Euler angles of the grasp orientation. When the gripper closes, it moves vertically by Δz_2 and and maintains its grip for a duration Δ_t. A grasp is labelled as *successful* if the object is lifted with a vertical displacement exceeding θ_v and the rotation $\sqrt{\alpha^2+\beta^2+\gamma^2}$ within θ_{r2}.

B. Overview of the `PreAfford` Framework

Fig. 3 illustrates the `PreAfford` framework. We divide the task into two main phases: pre-grasping and grasping, managed by the pre-grasping module and grasping module, respectively. Within each module, three specialized neural networks, inspired by *Where2Act* structure [30] are employed: an affordance network ($\mathcal{A}$), a proposal network ($\mathcal{P}$), and a critic network ($\mathcal{C}$).

The training and inference processes operate in opposite directions, and we intuitively call this *relay*. Offline training dataset is collected from simulation. In order to generate labels for training the pre-grasping module, a grasping network which can judge the success likelihood is trained firstly. Then pre-grasping network is trained together with information from simulation data and the output of grasping network. In the prediction phase, two modules form a closed-loop control. The pre-grasping module adjusts object's pose until it is suitable for grasping module to apply grasping.

C. Module Structure

Feature Extractors: Each network independently extracts features through its perception module, encoding object and environment point clouds into $f_s \in \mathbb{R}^{160}$ using a PointNet++ module with segmentation head [39]. We employ

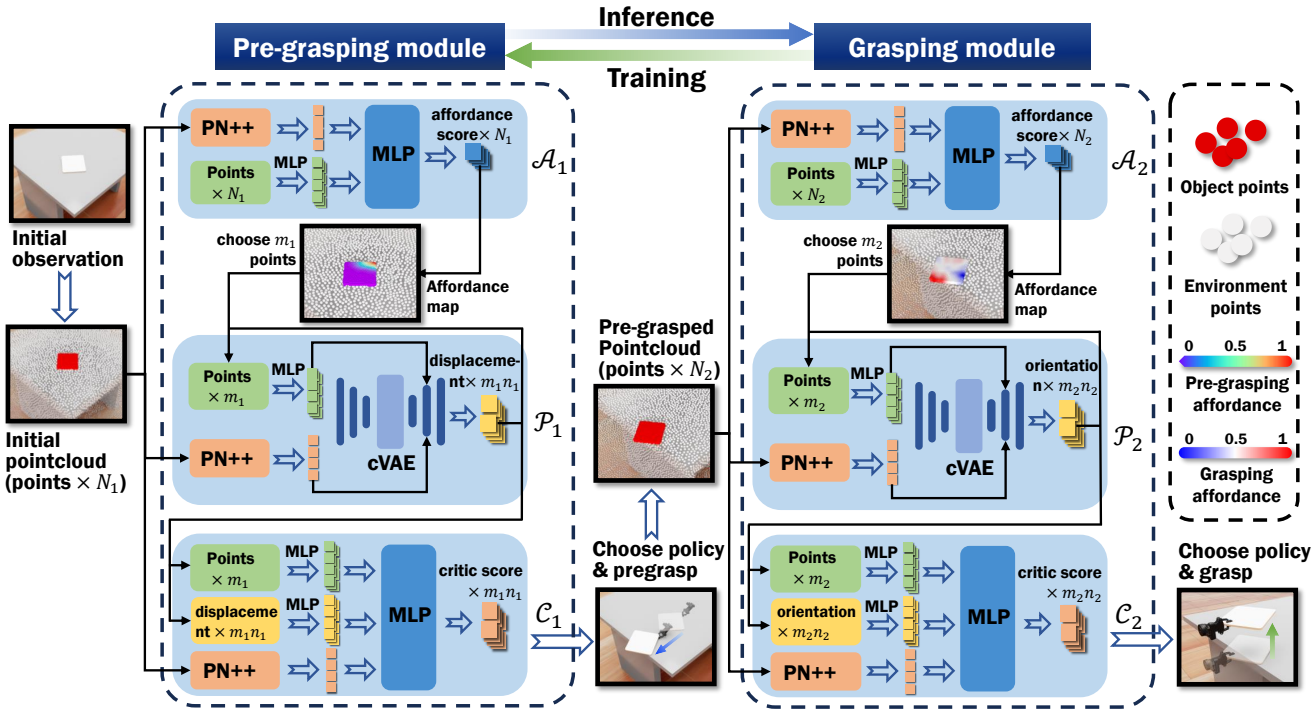

Fig. 3: **The framework of `PreAfford`.** Two modules are included, each containing an affordance network, proposal network, and critic network. The three networks are responsible for choosing contact point, generating proposal and evaluating the proposal, respectively. PointNet++ (PN++) and Multi-layer Perceptron (MLP) networks are used for processing point clouds and decision-making. In the inference phase, the two modules process point clouds to devise strategies for pre-grasping and grasping. Conversely, during training, the grasping module generates rewards to train the pre-grasping module, which we intuitively call *relay*.

various Multilayer Perceptron (MLP) networks for other input features, encoding the contact point p_i, the gripper displacement $\Delta\vec{x}_1$ and the gripper orientation $\vec{\theta}_2$ into $f_{p_i} \in \mathbb{R}^{32}$, $f_{M_1} \in \mathbb{R}^{32}$ and $f_{M_2} \in \mathbb{R}^{32}$.

Affordance Network: Both $\mathcal{A}_1$ and $\mathcal{A}_2$ predict an affordance score $\mathcal{A}_i(p_i|\mathcal{O}_i,\mathcal{E}) \in [0,1]$ for specific contact point p_i. The $\mathcal{A}_1$ score assesses the suitability of contact points for pre-grasping and the $\mathcal{A}_2$ score indicates the likelihood of success for grasping. It is implemented as an MLP that takes f_s and f_{p_i} as input. By aggregating the affordance scores, we generate an affordance map, shown as Fig. 6.

Proposal Network: The network $\mathcal{P}_1$ outputs the gripper displacement $\Delta\vec{x}_1$ at the specified point p_1. It was in a cVAE [40] structure with 32 hidden dims, in which the encoder maps f_s, f_{p_1} and f_{M_1} into Gaussian noise $z \in \mathbb{R}^{32}$ and the decoder reconstruct them. It's the same for P_2 except that it receives and reconstructs gripper orientation $\vec{\theta}_2$.

Critic Network: The network $\mathcal{C}_1$ scores a pre-grasping operation: $\mathcal{C}_1((p_1,\Delta\vec{x}_1)|\mathcal{O}_1,\mathcal{E}) \in \mathbb{R}$, where the pair $(p_1,\Delta\vec{x}_1)$ is proposed by A_1 and $\mathcal{P}_1$. And $\mathcal{C}_2$ evaluates the likelihood of success of a grasping operation: $\mathcal{C}_2(p_2,\vec{\theta}_2|\mathcal{O}_2,\mathcal{E}) \in [0,1]$. The critic network is a sinle-layer MLP that consumes f_s , f_{p_i} and f_{M_i}.

D. Inference

The inference pipeline of `PreAfford` includes 4 stages, we describe them respectively below.

Pre-grasping Necessity Check: Before pre-grasping step, a necessity check is applied to decide whether pre-grasping step could be skipped. To evaluate the *graspability*, $\mathcal{C}_2$ takes in manipulation proposals generated by $\mathcal{A}_2$ and $\mathcal{P}_2$, and gives success rate $\hat{c}_2$ by averaging the scores:

$$\hat{c}_2 = \frac{1}{n_2 m_2}\sum_{j=1}^{n_2}\sum_{k=1}^{m_2}\mathcal{C}_2(p_2^j,\mathcal{P}_2(p_2^j,z^k)|\mathcal{O}_2,\mathcal{E}). \tag{3}$$

If $\hat{c}_2$ passed a threshold θ_g, the object would be grasped directly following the proposal with the highest critic score.

Pre-Grasping Manipulation Inference and Implementation: At the inference stage, the affordance network $\mathcal{A}_1$ evaluates the affordance value for each point, and the points $p_1^j (j=1,2,\ldots,n_1)$ with the highest n_1 scores are selected as contact points. Next, the proposal network $\mathcal{P}_1$ generates m_1 pre-grasping manipulations $\Delta\vec{x}_1^{jk}(k=1,2,\ldots,m_1)$ for each p_1^j, each associated with a randomly generated normal distribution vector z_1^{jk}. Finally, the critic network $\mathcal{C}_1$ selects the optimal pair $p_1^{j^*}, \Delta\vec{x}_1^{j^*k^*}$ to execute, where

$$p_1^j(\mathcal{O}_1) = \underset{p_1}{\text{argmax}}^{(n_1)}\mathcal{A}_2(p_1|\mathcal{O}_1,\mathcal{E}), \tag{4}$$

$$j^*,k^* = \underset{j,k}{\text{argmax}}\ \mathcal{C}_1(p_1^j,\mathcal{P}_1(p_1^j,z_k)|\mathcal{O}_1,\mathcal{E}). \tag{5}$$

This process modifies the object's pose within the environment, yielding a new point cloud $\mathcal{O}_2$ that serves as input for the grasping module.

Grasping Manipulation Inference and Implementation: Same as mentioned in Sec. III-D, after pre-grasping step, the object would be grasped by the manipulation with the highest critic score produced by $\mathcal{A}_2$ and $\mathcal{P}_2$.

Closed-loop Control (optional): The inference pipeline of `PreAfford` forms a closed-loop control system that allows for additional pushing actions when a single pre-grasping manipulation fails to achieve a graspable object configuration. The process can be described as follows:

1) The grasping module first estimates the expected success rate of the grasp $\hat{c}_2$.
2) If the score is below the predefined threshold θ, the pre-grasping module forms a manipulation that pushes the object to a new position.
3) Go back to step 1 and re-evaluate $\hat{c}_2$ in the new state.
4) Repeat steps 2-3 until the $\hat{c}_2$ is high enough to attempt the grasp.

E. Training and Losses

Critic Loss: The loss function of the critic network $\mathcal{C}_2$ is determined based on the outcome (success or not) of the grasping manipulation, represented as r:

$$\mathcal{L}_{\mathcal{C}_2} = r\log(\mathcal{C}_2(p_2,\vec{\theta}_2)) + (1-r)\log(1-\mathcal{C}_2(p_2,\vec{\theta}_2)). \quad (6)$$

For $\mathcal{C}_1$, its loss is assessed by the extent to which the pre-grasping manipulation enhances the success rate of a grasp. We use the grasping module to evaluate, giving success rate $\hat{c}_2^{\text{before}}$ and $\hat{c}_2^{\text{after}}$. And to ensure stable and safe pushing actions, following situations are penalized in training process by multiplying a penalty term p to the grasping success likelihood enhancement $\hat{c}_2^{\text{after}} - \hat{c}_2^{\text{before}}$.

- Displacement penalty. $p_d = \exp(-|\Delta\mathbf{x}_{\text{go}}|/a)$, where $\Delta\mathbf{x}_{\text{go}}$ is the relative displacement of grippers and object, and a is a constant coefficient.
- Rotation penalty. $p_r = \exp(-\sqrt{\alpha^2+\beta^2+\gamma^2}/b)$, where (α,β,γ) represents the rotation in Euler angles, and b is a constant coefficient.
- Safety penalty. p_s is 1 by default, and will be set to be 0 if one of the following situations happens: Object falling off table, gripper colliding with a wall, gripper colliding with a slope.

The total penalty term p is the multiplication of p_d, p_r and p_s. We use l_1 loss for $\mathcal{C}_1$:

$$\mathcal{L}_{\mathcal{C}_1} = \left|\mathcal{C}_1(p_1,\Delta\vec{x}_1|\mathcal{O}_1,\mathcal{E}) - p\cdot(\hat{c}_2^{\text{after}} - \hat{c}_2^{\text{before}})\right| \quad (7)$$

Proposal Loss: $\mathcal{P}_1$ and $\mathcal{P}_2$ are implemented as cVAE. Aimed at generating proper actions, the proposal network has to reconstruct correct ones well, so only *successful* pre-grasping and grasping manipulations are used in training. The loss is set as the sum of geometric loss and KL divergence loss. The prior one measures the difference between reconstructed and the ground truth $\hat{M}_i$. The latter measures the difference between the hidden layer distribution and the normal distribution. The total proposal loss is:

$$\mathcal{L}_{\mathcal{P}_i} = \mathcal{L}_{\text{geo}}(M_i;\hat{M}_i) + D_{\text{KL}}(q(z|p_i,\hat{M}_i,\mathcal{O}_i,\mathcal{E})|\mathcal{N}(0,1)). \quad (8)$$

Affordance Loss: Affordance score indicates the suitability of contact point and help to choose the contact point. It evaluates how well the action proposed by $\mathcal{P}_i$ work and that can be scored offline by the critic network [30]. The affordance network is trained after the other two networks in each module. Data is labeled by the average critic score $\hat{a}_{p_i}$ of n_i motion generated by $\mathcal{P}_i$ with different Gaussian noise vector $z_j (j = 1, 2, \cdots, n_i)$:

$$\hat{a}_{p_i} = \frac{1}{n_i}\sum_{j=1}^{n_i} \mathcal{C}_i(p_i,\mathcal{P}_i(p_i,z_j)|\mathcal{O}_i,\mathcal{E}), \quad (9)$$

$$\mathcal{L}_{\mathcal{A}_i} = |\mathcal{A}_i(p_i|\mathcal{O}_i,\mathcal{E}) - \hat{a}_{p_i}|\,. \quad (10)$$

F. Data Collection

We use Sapien [41] to gather extensive data for pre-grasping and grasping tasks, selecting 5 hard-to-grasp and 5 easy-to-grasp object categories from ShapeNet-v2 for training, with over 10 shapes for each category. A INSPIRE-ROBOTS EG2-4C model is used for the gripper.

To generate an offline training dataset for the grasping module, objects are generated at random positions across four scenes, with chances of being posed *upon* environmental features. Grasping points and gripper orientations are randomly chosen from the object's surface and the hemisphere above the tangent plane at the point, respectively.

The pre-grasping module's dataset focuses on the hard-to-grasp categories. Objects are randomly posed within a uniformly distributed distance from the environmental feature. The gripper performs a pushing action with a normally-distributed displacement on random surface point. Collection efficiency is boosted by over 42% by specifying the gripper's displacement direction towards environmental features for 30% of cases (using domain-specific prior knowledge). The angle to the feature follows a Gaussian distribution.

For both datasets, the camera position is randomly generated, maintaining a distance between 3 and 5 meters from the object, with the camera oriented towards the object. Each dataset consists of 10,000 successful cases and 30,000 failure cases, as the negative operational space is hard to cover.

IV. Experiments

A. Environment Settings and Datasets

We tested `PreAfford` on 5 seen and 4 unseen categories of hard-to-grasp objects across 5 scenes to demonstrate its adaptability. These 5 scenes contain 4 scenes with a single

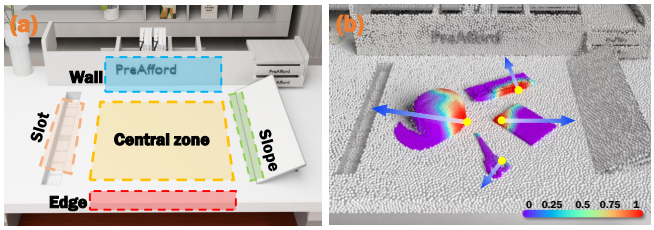

Fig. 4: **Multi-feature case.** (a) Rendered image of the complex environment. (b) Point cloud and affordance hot map. Our method can address the case that multiple environment features are simultaneously present.

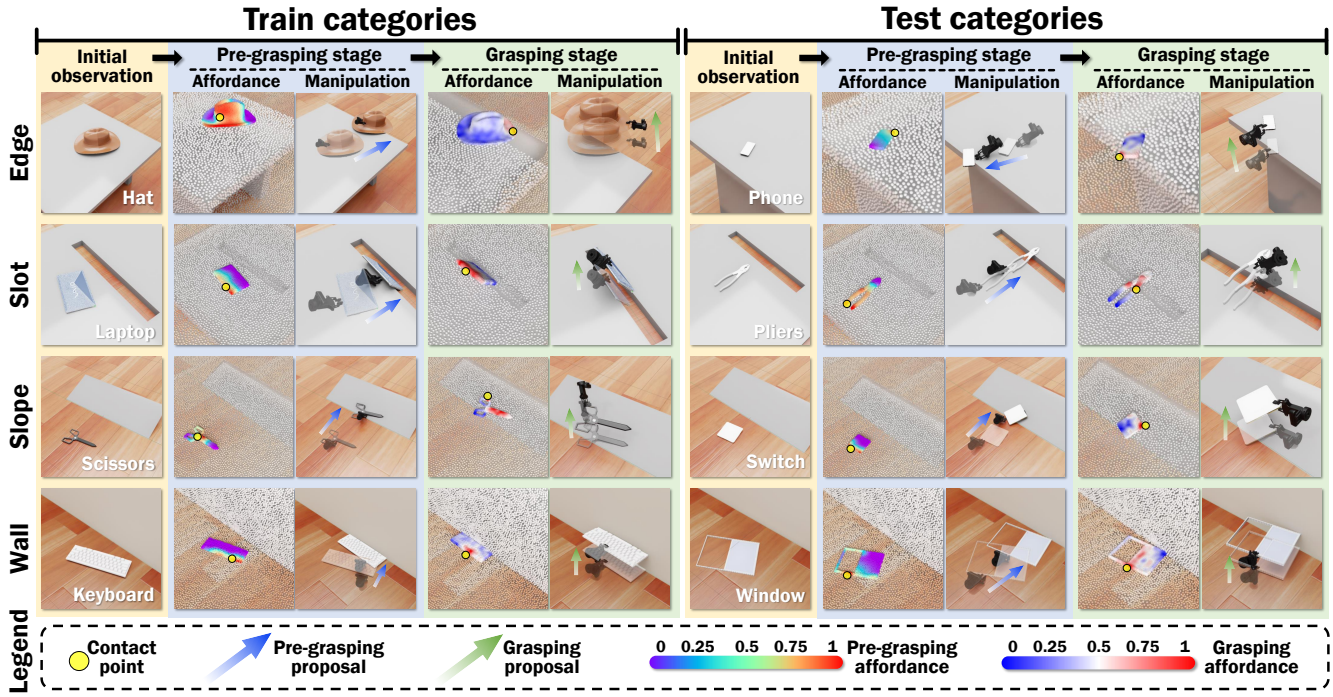

Fig. 5: **Qualitative results.** Both training and testing categories are showcased across four distinct scenarios: edge, slot, slope, and wall. Affordance maps illustrate potential effective interaction areas. It can be observed that `PreAfford` yields reasonable pre-grasping and grasping strategies across different object categories and scenes on both seen and unseen objects.

environmental features from edge, wall, slope, and slot, and one novel scene (shown in Fig. 4) with 4 environmental features simultaneously is also constructed for testing its adaptability to unseen complex environment (labeled as Multi below). We conducted 1,000 tests on each object-environment pair, and calculate the mean success ratio. Other experimental setups follow Sec. III-F.

B. Evaluation Metrics and Baselines

To evaluate the quality of a pre-grasping proposal, we measure the increase in the sample success rate of the grasping action. After performing a pre-grasping manipulation, a grasping proposal is generated using the grasping module, and its success is tested. This test is repeated 1,000 times for each object-environment pair to obtain a sample success rate. The efficacy of the pre-grasping manipulation is then calculated by comparing the increase in the sample success rate to the case without pre-grasping manipulation. For comparison, we employ four baselines: (a) W/o pre-grasping: a direct grasping manipulation without any pre-grasping action; (b) Random-direction Push: the contact point is proposed by the pre-grasping module of `PreAfford`, but the displacement is set to a random direction; (c) Center-point Push: the displacement is proposed by the pre-grasping module, but the contact point is set to the object's geometric center; and (d) Ours w/o closed-loop: an ablation study on `PreAfford` that eliminates the closed-loop control process. We demonstrate the compatibility of `PreAfford` on 5 training and 4 testing categories of easy-to-grasp objects.

C. Analysis

Efficacy and Adaptability: Fig. 5 presents the pre-grasping and grasping affordance maps predicted by the affordance networks and the proposed pre-grasping displacement given by $\mathcal{P}_1$. We can conclude that our network exhibits two key abilities: (a) **Environmental awareness:** The affordance map assigns higher values to the side opposite the target environmental feature, with the proposed direction mostly pointing towards the specific landscape. This indicates the network's ability to choose suitable pre-grasping policies given the point cloud of the surrounding environment. (b) **Dynamics awareness:** The network tends to propose pushes on the side of thin objects, with the pushing direction roughly passing through the object's center of mass. This suggests an

TABLE II: **Comparison with baselines.** Pre-grasping significantly improves grasping success rates by 52.9%. Closed-loop strategy further improves success rates by 16.4% over all categories.

Setting	Train object categories						Test object categories					
	Edge	Wall	Slope	Slot	Multi	Avg.	Edge	Wall	Slope	Slot	Multi	Avg.
W/o pre-grasping	2.3	3.8	4.3	3.4	4.0	3.6	6.1	2.3	2.9	5.7	6.0	4.6
Random-direction Push	21.6	10.3	6.4	16.8	18.1	14.6	24.9	17.2	12.1	18.4	23.0	19.1
Center-point Push	32.5	23.7	40.5	39.2	39.0	35.0	25.1	17.4	28.0	30.2	21.5	24.4
Ours w/o closed-loop	67.2	41.5	58.3	76.9	63.6	61.5	56.4	37.3	62.6	75.8	55.4	57.5
Ours	**81.4**	**43.4**	**73.1**	**83.5**	**74.1**	**71.1**	**83.7**	**47.6**	**80.5**	**83.0**	**74.6**	**73.9**

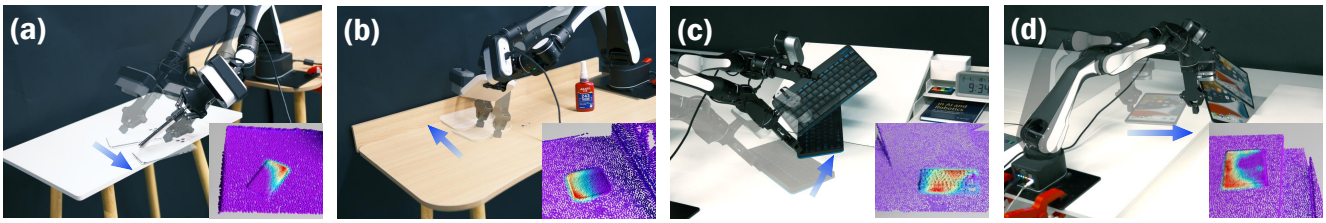

Fig. 6: **Real world pre-grasping manipulation.** Four pre-grasping cases with their affordance map are demonstrated: (a) Move a tablet to table edge, (b) Push a plate towards wall, (c) Push a keyboard up onto a slope and (d) Slide a tablet into a slot. In these affordance maps, areas marked in red signify the optimal locations for pushing. Point clouds are captured by Femto Bolt.

TABLE III: **Real world experiment results.**

Setting	Seen categories						Unseen categories					
	Edge	Wall	Slope	Slot	Multi	Avg.	Edge	Wall	Slope	Slot	Multi	Avg.
W/o pre-grasping	0	0	0	0	0	0	10	0	5	0	0	3
With pre-grasping	70	45	80	90	85	74	80	30	75	90	85	72

awareness of dynamic behavior that could minimize object rotation during the pushing process.

The adaptability of `PreAfford` is clearly illustrated by reasonable pre-grasping proposals across different scenes and object categories. Fig. 4 further demonstrates that when an object is placed in a complex scene integrating four environmental features, `PreAfford` can propose different pre-grasping policies based on the object's specific position within the scene.

Tab. II quantitatively validate our method. The results demonstrate that direct grasping attempts in ungraspable situations are highly unlikely to succeed. The results on Random-direction Push and Center-point Push strategies confirmed our observations regarding the environmental awareness and dynamics awareness of `PreAfford`, respectively, highlighting the importance of generating effective pushing directions and selecting appropriate contact points regarding surrounding landscape and dynamic features. Furthermore, the ablation study w/o closed-loop, underscores the significance of the closed-loop control process, demonstrating its contribution to the success rates.

Compatability: As described in Section Sec. III-D, cases with an estimated grasping success likelihood $\hat{c}_2$ below a threshold θ_g trigger a pre-grasping manipulation. The choice of θ_g is crucial for balancing compatibility with graspable objects and initiating necessary pre-grasping manipulations for ungraspable cases. Testing on both object categories in the Multi scene, we found that setting θ_g to 0.8 achieves a fair balance, as shown in Tab. IV.

TABLE IV: **Compatibility illustration.** Below shows the rate of performing a direct grasping manipulation on both graspable and ungraspable objects. While $\theta_g = 0.8$, most ungraspable objects would be pre-grasped, while graspable objects not.

Metric	Train categories		Test categories	
	Graspable	Ungraspable	Graspable	Ungraspable
Pre-grasping rate	14.7	84.1	23.4	77.5
Success rate	83.7	74.1	80.0	74.6

Real-world Experiment Setup: We deployed our algorithm on the AIRBOT Play robotic arm, a lightweight, compact, six-degree-of-freedom manipulator. To generate input point clouds, we utilized an ORBBEC Femto Bolt camera for RGB-D data. An INSPIRE-ROBOTS EG2-4C gripper is used as the end effector.

The experimental setup involves ten object categories, with five categories seen during training and five unseen during training, as shown in Fig. 7(a). To test `PreAfford`'s adaptability, we constructed five scenes featuring different environmental features, including edges, walls, slopes, slots, and a *Multi* scene that integrated all four features, shown in Fig. 7(b)-(f). Fig. 6 illustrates the pre-grasping and grasping manipulations, along with the corresponding pre-grasping affordance maps generated by our algorithm.

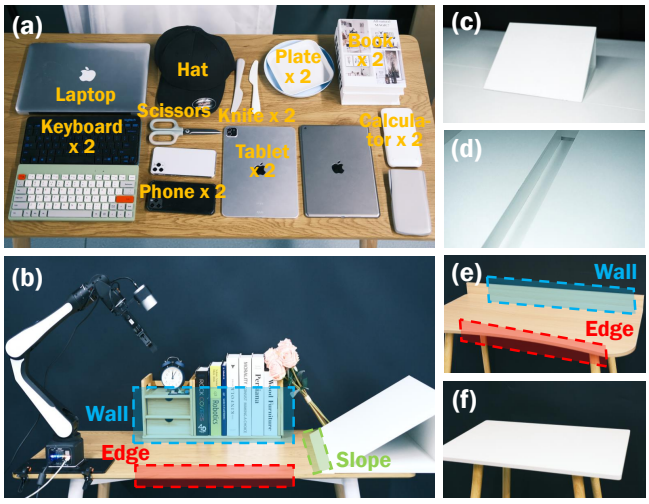

Fig. 7: **Real world experiment setups.** (a) object categories used for testing, (b) Multi scene with three environmental features (with hardware settings including an AIRBOT Play robotic arm, an INSPIRE-ROBOTS gripper, and a Femto Bolt RGB-D camera), (c) Slope scene, (d) Slot scene, (e) Wall scene (with edge) and (f) Edge scene. Note: (c-f) are not simulated.

Real-world Results: Tab. III presents the results of our real-world experiments. We select 10 categories of objects,

perform 4 tests on each object in each scene, (*i.e.*, 20 experiments per scene). Our proposed framework, `PreAfford`, highly improved the grasping success likelihood on those hard-to-grasp objects in diverse categories, demonstrating high deloyability in the real world.

V. Conclusion and Limitation

In this paper, we propose a novel two-stage affordance learning framework that achieves adaptability across object-environment configurations, compatibility on graspable objects, and deployability to real-world experiments. This framework is tested in both simulation and real-world experiments, validating its efficacy.

References

[1] Z. Sun, K. Yuan, W. Hu, C. Yang, and Z. Li, "Learning pregrasp manipulation of objects from ungraspable poses," in *ICRA*, 2020.

[2] D. Kappler, L. Chang, M. Przybylski, N. Pollard, T. Asfour, and R. Dillmann, "Representation of pre-grasp strategies for object manipulation," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2010.

[3] K. M. Lynch and M. T. Mason, "Dynamic nonprehensile manipulation: Controllability, planning, and experiments," *IJRR*, vol. 18, no. 1, pp. 64–92, 1999.

[4] N. C. Dafle, A. Rodriguez, R. Paolini, B. Tang, S. S. Srinivasa, M. Erdmann, M. T. Mason, I. Lundberg, H. Staab, and T. Fuhlbrigge, "Extrinsic dexterity: In-hand manipulation with external forces," in *ICRA*, 2014.

[5] K. Hang, A. S. Morgan, and A. M. Dollar, "Pre-grasp sliding manipulation of thin objects using soft, compliant, or underactuated hands," *RA-L*, vol. 4, no. 2, pp. 662–669, 2019.

[6] L. Y. Chang, S. S. Srinivasa, and N. S. Pollard, "Planning pre-grasp manipulation for transport tasks," in *ICRA*, 2010.

[7] D. Ren, X. Ren, X. Wang, S. T. Digumarti, and G. Shi, "Fast-learning grasping and pre-grasping via clutter quantization and q-map masking," in *IROS*, 2021.

[8] S. Chen, A. Wu, and C. K. Liu, "Synthesizing dexterous nonprehensile pregrasp for ungraspable objects," in *ACM SIGGRAPH 2023 Conference Proceedings*, 2023.

[9] K. Mokhtar, C. Heemskerk, and H. Kasaei, "Self-supervised learning for joint pushing and grasping policies in highly cluttered environments," *arXiv preprint arXiv:2203.02511*, 2022.

[10] J. Wu, H. Wu, S. Zhong, Q. Sun, and Y. Li, "Learning pre-grasp manipulation of flat objects in cluttered environments using sliding primitives," in *ICRA*, 2023.

[11] A. Nguyen, D. Kanoulas, D. G. Caldwell, and N. G. Tsagarakis, "Preparatory object reorientation for task-oriented grasping," in *IROS*, 2016.

[12] Y. Wang, R. Wu, K. Mo, J. Ke, Q. Fan, L. J. Guibas, and H. Dong, "Adaafford: Learning to adapt manipulation affordance for 3d articulated objects via few-shot interactions," in *ECCV*, 2022.

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

[14] C. Ning, R. Wu, H. Lu, K. Mo, and H. Dong, "Where2explore: Few-shot affordance learning for unseen novel categories of articulated objects," in *NeurIPS*, 2023.

[15] 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.

[16] L. Y. Chang, G. J. Zeglin, and N. S. Pollard, "Preparatory object rotation as a human-inspired grasping strategy," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2008.

[17] L. Y. Chang, R. L. Klatzky, and N. S. Pollard, "Selection criteria for preparatory object rotation in manual lifting actions," *Journal of Motor Behavior*, vol. 42, no. 1, pp. 11–27, 2009.

[18] Y. Li, B. Liu, Y. Geng, P. Li, Y. Yang, Y. Zhu, T. Liu, and S. Huang, "Grasp multiple objects with one hand," *RA-L*, 2024.

[19] G. Lee, T. Lozano-Pérez, and L. P. Kaelbling, "Hierarchical planning for multi-contact non-prehensile manipulation," in *IROS*, 2015.

[20] M. Moll, L. Kavraki, J. Rosell, *et al.*, "Randomized physics-based motion planning for grasping in cluttered and uncertain environments," *RA-L*, vol. 3, no. 2, pp. 712–719, 2017.

[21] H.-a. Gao, B. Tian, P. Li, H. Zhao, and G. Zhou, "Dqs3d: Densely-matched quantization-aware semi-supervised 3d detection," in *ICCV*, 2023.

[22] H.-a. Gao, B. Tian, P. Li, X. Chen, H. Zhao, G. Zhou, Y. Chen, and H. Zha, "From semi-supervised to omni-supervised room layout estimation using point clouds," in *ICRA*, 2023.

[23] J. Lee, Y. Cho, C. Nam, J. Park, and C. Kim, "Efficient obstacle rearrangement for object manipulation tasks in cluttered environments," in *ICRA*, 2019.

[24] H. Zhang, Y. Lu, C. Yu, D. Hsu, X. La, and N. Zheng, "Invigorate: Interactive visual grounding and grasping in clutter," *arXiv preprint arXiv:2108.11092*, 2021.

[25] J. J. Gibson, "The ecological approach to the visual perception of pictures," *Leonardo*, vol. 11, no. 3, pp. 227–235, 1978.

[26] T. Nguyen, M. N. Vu, A. Vuong, D. Nguyen, T. Vo, N. Le, and A. Nguyen, "Open-vocabulary affordance detection in 3d point clouds," in *IROS*, 2023.

[27] X. Chen, T. Liu, H. Zhao, G. Zhou, and Y.-Q. Zhang, "Cerberus transformer: Joint semantic, affordance and attribute parsing," in *CVPR*, 2022.

[28] L. Cui, X. Chen, H. Zhao, G. Zhou, and Y. Zhu, "Strap: Structured object affordance segmentation with point supervision," *arXiv preprint arXiv:2304.08492*, 2023.

[29] P. Li, B. Tian, Y. Shi, X. Chen, H. Zhao, G. Zhou, and Y.-Q. Zhang, "Toist: Task oriented instance segmentation transformer with noun-pronoun distillation," *Advances in Neural Information Processing Systems*, vol. 35, pp. 17597–17611, 2022.

[30] K. Mo, L. J. Guibas, M. Mukadam, A. Gupta, and S. Tulsiani, "Where2act: From pixels to actions for articulated 3d objects," in *ICCV*, 2021.

[31] P. Li, T. Liu, Y. Li, Y. Geng, Y. Zhu, Y. Yang, and S. Huang, "Gendexgrasp: Generalizable dexterous grasping," in *ICRA*, 2023.

[32] S. Ling, Y. Wang, R. Wu, S. Wu, Y. Zhuang, T. Xu, Y. Li, C. Liu, and H. Dong, "Articulated object manipulation with coarse-to-fine affordance for mitigating the effect of point cloud noise," in *ICRA*, 2024.

[33] Y. Li, X. Zhang, R. Wu, Z. Zhang, Y. Geng, H. Dong, and Z. He, "Unidoormanip: Learning universal door manipulation policy over large-scale and diverse door manipulation environments," *arXiv preprint arXiv:2403.02604*, 2024.

[34] R. Wu, K. Cheng, Y. Zhao, C. Ning, G. Zhan, and H. Dong, "Learning environment-aware affordance for 3d articulated object manipulation under occlusions," in *NeurIPS*, 2023.

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

[36] K. Mo, Y. Qin, F. Xiang, H. Su, and L. Guibas, "O2o-afford: Annotation-free large-scale object-object affordance learning," in *CoRL*, 2022.

[37] R. Wu, C. Ning, and H. Dong, "Learning foresightful dense visual affordance for deformable object manipulation," in *ICCV*, 2023.

[38] Y. Ju, K. Hu, G. Zhang, G. Zhang, M. Jiang, and H. Xu, "Robo-abc: Affordance generalization beyond categories via semantic correspondence for robot manipulation," *arXiv preprint arXiv:2401.07487*, 2024.

[39] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *NeurIPS*, 2017.

[40] C. Cortes, N. Lawarence, D. Lee, M. Sugiyama, and R. Garnett, "Advances in neural information processing systems 28," in *NeurIPS*, 2015.

[41] 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 *CVPR*, 2020.