`PreAfford`: Universal Affordance-Based Pre-Grasping for Diverse Objects and Environments

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

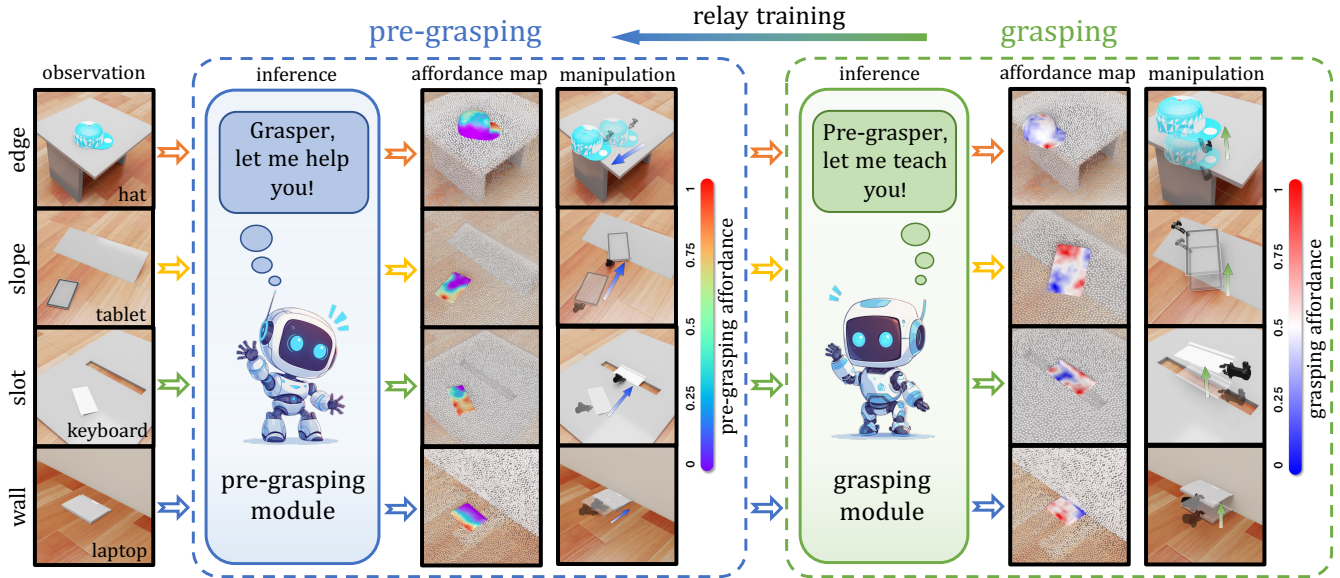Project website: https://air-discover.github.io/PreAfford

Fig. 1: **Illustration of `PreAfford`, demonstrating the application of a** *relay training* **paradigm where two synergistic modules cooperate to facilitate the manipulation of objects typically considered ungraspable.** The *pre-grasping* module assesses environmental features such as edges, slopes, slots, and walls to propose strategic pre-grasping actions that enhance the likelihood of a successful grasp. Simultaneously, the *grasping* module evaluates these actions and provides feedback in the form of rewards, which are used to refine and optimize the pre-grasping strategies. Two color bars represent the pre-grasping and grasping phases, respectively, with the color intensity reflecting the calculated affordance values; higher values denote more optimal interaction conditions.

Abstract— **Robotic manipulation with two-finger grippers is challenged by objects lacking distinct graspable features. Traditional pre-grasping methods, which typically involve repositioning objects or utilizing external aids like table edges, are limited in their adaptability across different object categories and environments. To overcome these limitations, we introduce `PreAfford`, a novel pre-grasping planning framework incorporating a point-level affordance representation and a relay training approach. Our method significantly improves adaptability, allowing effective manipulation across a wide range of environments and object types. When evaluated on the ShapeNet-v2 dataset, `PreAfford` not only enhances grasping success rates by 69% but also demonstrates its practicality through successful real-world experiments. These improvements highlight `PreAfford`'s potential to redefine standards for robotic handling of complex manipulation tasks in diverse settings.**

I. INTRODUCTION

Consider a robotic arm equipped with a two-finger gripper attempting to grasp a mobile phone lying flat on a surface, as illustrated in Fig. 2(a). The close contact between the phone and the table results in a lack of graspable features, creating an *ungraspable* scenario [4]. Inspired by human object manipulation strategies, an effective approach involves repositioning the object towards a table edge, allowing the phone to hang off slightly and thus become graspable. This technique, termed *pre-grasping* manipulation, is a critical preliminary step in enabling successful grasping in challenging scenarios [2, 5, 8, 9]. As depicted in Fig. 2(b-e), various environmental features, also known as extrinsic dexterity [10, 11], can be leveraged to reconfigure the object into a more favorable position for grasping.

Research has shown that pre-grasping manipulations, including pushing [3], rotating [2, 4], and sliding [1, 12], can significantly enhance the success rate of grasping [6]. However, three key limitations remain to be addressed:

1) **Adaptability**. Existing studies primarily focus on task-specific settings, relying heavily on manually programmed criteria for pre-grasping success, limiting their

TABLE I: **Comparisons with prior pre-grasping methods.** This table offers a concise comparison of various pre-grasping studies, highlighting their key features and limitations. The last three columns evaluate each study's *adaptability* across different object categories and environments, *compatibility* with existing grasping pipelines via a defined procedure to bypass the pre-grasping step, and *deployability* in real-world experiments without the need for custom shape representations [1] or identical environments to those used in simulators [2]. Columns A, C, and D represent adaptability, compatibility, and deployability, respectively.

| method	end effector	pre-grasping manipulation	method	scenario	A	C	D
Ren *et al.* [3]	two-finger gripper	pushing	DRI	clustered objects	×	✓	✓
Sun *et al.* [4]	spherical	rotation	DRI	cuboid in corner	×	×	✓
Kappler *et al.* [5]	dexterous hand	pushing	data-driven approach	cuboid on table	×	✓	×
Chen *et al.* [6]	dexterous hand	finger contact	learning-based framework	ungraspable cases	✓	✓	×
Hang *et al.* [1]	two-finger gripper	sliding	integrated planning	thin objects on table	×	×	×
Chang *et al.* [2]	two-finger gripper	rotation	opt. for payload	transport tasks	×	×	×
Wang *et al.* [7]	two-finger gripper	pushing	DRI	clustered objects	×	✓	✓
Ours	**two-finger gripper**	**pushing**	**dual-module affordance map**	**ungraspable cases**	✓	✓	✓
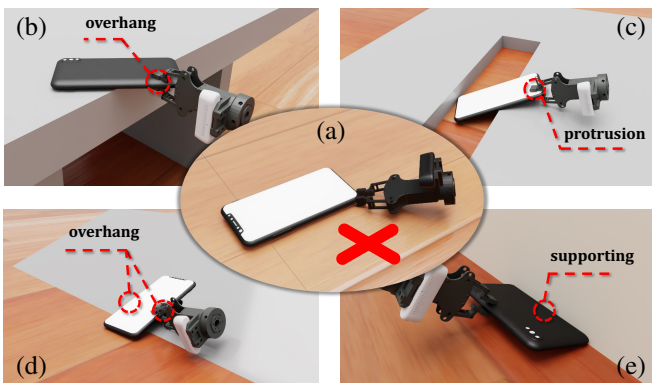
Fig. 2: **Pre-grasping leverages environmental features to enhance graspability.** (a) An object lying flat on the floor, ungraspable in its current position. (b) Side-grasping an object that overhangs a surface. (c) Grasping an angled part protruding from a slot. (d) Grasping the middle of a phone suspended at the foot of a slope. (e) Pinning a phone against a wall and grasping it from the opposite side.

ability to adapt across different object categories and environments [1, 3–5, 13].

2) **Deployability**. Pre-grasping planning methods often struggle to transition to real-world applications due to their dependence on privileged information from simulators or the need for customized representations for specific experimental setups [1, 2, 6].

3) **Compatibility**. To minimize the complexity and cost of robotic control, pre-grasping manipulations should be omitted for objects that are easy to grasp. However, mechanisms to assess the necessity of pre-grasping are often lacking in prior work [1–5, 13, 14].

To date, no approach has successfully addressed adaptability, deployability, and compatibility in a unified manner.

We introduce the `PreAfford`, which effectively addresses the above limitations. The adaptability of `PreAfford` arises from a novel *relay training* paradigm within a dual-module framework. This setup generates a robust reward function to improve the likelihood of grasping success, which is consistently evaluated by the grasping module. For deployability and real-world adaptability, `PreAfford` employs a point-level affordance visual representation [14–16] that relies solely on RGB-D data. Additionally, the `PreAfford` integrates a pre-grasping necessity check at the start of the inference process, enabling direct grasping of objects that are straightforward to handle, thereby enhancing compatibility and operational efficiency.

By training and testing on a large-scale offline dataset from ShapeNet-v2 [17] across five scenes, simulations demonstrated that `PreAfford` increases the grasping success rate by 69% for test object categories. The pre-grasping and grasping affordance maps, as shown in Fig. 5, indicate that our models possess a profound understanding of object geometries and the environmental features of the system. Additionally, `PreAfford` has demonstrated the ability to select appropriate pre-grasping policies in unseen, complex environments. The deployability of our framework is further confirmed through real-world experiments conducted across five different 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 that enhances pre-grasping manipulation strategies.
- Point-level affordance representation that enables detailed geometry awareness and seamless deployment.
- Extensive validation of `PreAfford` in both simulated and real-world settings, demonstrating its efficacy.

II. RELATED WORK

A. Pre-Grasping Tasks

Pre-grasping, a concept inspired by human behavior where objects are often pre-manipulated for easier handling [18, 19], involves robot manipulators adjusting the pose of an object *before* executing the final grasp [3, 5, 7, 13, 19, 20]. This adjustment is crucial for successful interaction with objects, especially when they are in challenging positions or orientations.

We compare previous pre-grasping strategies in Tab. I. One research direction involves altering the pose of an object to a more graspable configuration, often through rotation [1, 2, 21]. Another line of work focuses on pre-grasping with extrinsic dexterity [6, 10, 11], which leverages environmental features to facilitate an easier grasp [4–6]. Additionally, extensive studies have addressed the issue of cluttered objects on tables, where objects overlapped in such a way that they cannot be grasped without rearrangement [3, 7, 22].

The critical element is the formulation of the reward function for pre-grasping manipulation. Defining the *graspability* of an object within a specific environmental setup is challenging, leading to two main approaches for constructing the reward function. The first method involves manually programming the reward, using strategies like pre-defined goal regions [1] or specific transitions in the pose of the object and gripper [4, 13]. Although effective within their specific experimental setups, these methods often lack the flexibility to adapt to new object categories and environments. The second approach uses the output from pre-trained neural networks to estimate the *graspability* at a given pose [2, 3, 6]. This method typically relies on simulator-derived prior knowledge, such as signed distance functions [6] or precise object and environment geometries [2], which poses challenges for real-world application due to the necessity for a deep understanding of the scene [23–26].

Furthermore, our findings indicate that many systems neglect the option to bypass the pre-grasping procedure for objects that can be directly grasped with a high confidence of success [2, 3, 6]. This oversight renders the pre-grasping framework redundant for most daily graspable objects and unnecessarily increases the complexity and cost of robotic control. To date, no previous research has successfully integrated adaptability across various object-environment configurations, compatibility with directly graspable situations, and deployability in real-world scenarios in a single framework.

B. Point-Level Affordance for Robotic Manipulation

Affordance, originally defined as the action possibilities linked to an object or environment for an agent [27–31], is vital in robotic manipulation. Point-level affordance learning involves creating dense affordance maps as actionable visual representations that suggest possible actions at every point on point clouds of 3D objects. Recent studies have applied point-level affordance learning to various scenarios [15, 16, 28, 32–38], providing detailed and actionable information for subsequent robotic actions.

Empirical research has shown that point-level affordance has a robust geometry-aware capability, effectively generalizing across both within-category [14, 39] and inter-category [16, 32, 37, 40] scenarios. This demonstrates its adaptability to new objects and situations.

In our research, we expand the application of point-level affordance to guide pre-grasping manipulations, considering various geometries and environmental contexts. This advancement shows that point-level affordance can manage more complex scenarios with high accuracy and strong generalization potential.

III. METHODOLOGY

We define core concepts of our method in Sec. III-A, describe the overall framework in Sec. III-B, elaborate on the network architectures in Sec. III-C, introduce the inference procedure in Sec. III-D, discuss 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 modify object poses to increase the likelihood of a successful grasp. Our approach particularly focuses on the exploitation of environmental features.

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

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

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. By default, the direction of pushing is set to horizontal, and the gripper remains closed during this operation.

In our research, the reward for a pre-grasping action is evaluated by the increase in the graspability score yielded by the grasping module (Eq. (6)). We define a pre-grasping manipulation as *successful* if the score increases by more than 40%. Two safety-critical situations are considered failure cases: (i) the object falling off the table and (ii) collision between the gripper and any obstacles such as walls.

Scene-aware Grasping: Grasping tasks involve identifying an optimal manipulation, $\mathcal{G}$, based on object and environment 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 [6, 36].

In our method, $\mathcal{O}$ is represented as point clouds and a grasp $\mathcal{G}$ is defined by six parameters:

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

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

B. Overview of the PreAfford

Figure 3 illustrates the PreAfford framework. We divide the task into two main phases: pre-grasping and grasping, which are managed by the pre-grasping module and the grasping module, respectively. Within each module, three specialized neural networks, inspired by the *Where2Act* structure [32], are employed: an affordance network $\mathcal{A}$, a proposal network $\mathcal{P}$, and a critic network $\mathcal{C}$.

Training and inference processes operate in opposite directions, a concept we refer to as *relay*. The offline training dataset is collected from simulations. A grasping network, which judges the likelihood of a grasp's success, is trained first to generate labels for training the pre-grasping module. Subsequently, the pre-grasping network is trained using both the simulation data and the outputs from the grasping network. During the prediction phase, the two modules form a closed-loop control system. The pre-grasping module adjusts the object's pose until it is deemed suitable for the grasping module to apply a grasp.
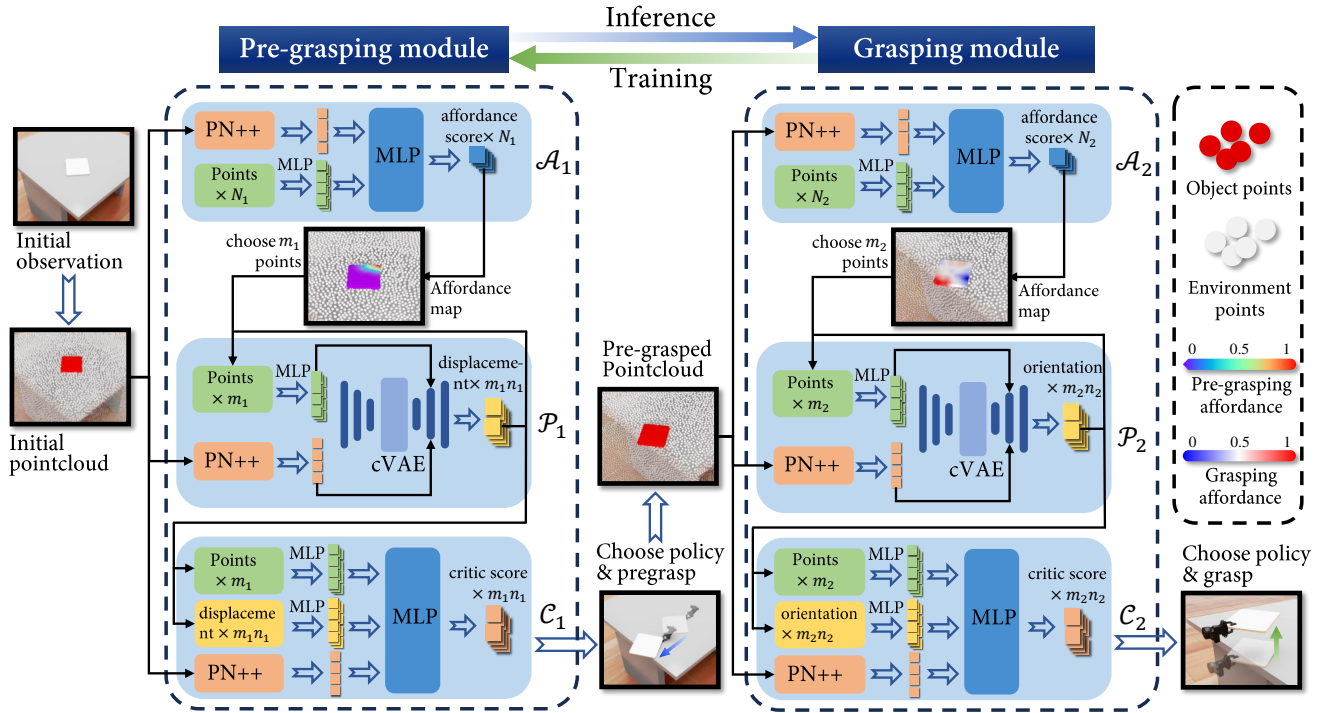
Fig. 3: **The framework of `PreAfford`.** The framework consists of two main modules, each incorporating three networks: an affordance network, a proposal network, and a critic network. These networks respectively handle tasks of choosing the contact point, generating a proposal, and evaluating the proposal. PointNet++ (PN++) and MLP are employed to process point clouds and facilitate decision-making. During the inference phase, both modules collaborate to develop strategies for pre-grasping and grasping. In contrast, during the training phase, the grasping module generates rewards for training the pre-grasping module, a process we refer to as *relay*.

C. Module Structure

Feature Extractors: Each network within the modules independently extracts features through its perception module. Object and environment point clouds are encoded into feature vectors $f_o \in \mathbb{R}^{160}$ using a PointNet++ module equipped with a segmentation head [41]. Furthermore, various Multilayer Perceptron (MLP) networks are employed to process additional input features. These include encoding the contact point p_i into $f_{p_i} \in \mathbb{R}^{32}$, the gripper displacement $\Delta \vec{x}_1$ into $f_{M_1} \in \mathbb{R}^{32}$, and the gripper orientation $\vec{\theta}_2$ into $f_{M_2} \in \mathbb{R}^{32}$.

Affordance Network: The Affordance Networks, $\mathcal{A}_1$ and $\mathcal{A}_2$, compute an affordance score $\mathcal{A}_i(p_i|\mathcal{O}_i)$ within the range [0,1] for each specific contact point p_i. The score from $\mathcal{A}_1$ evaluates the suitability of contact points for pre-grasping, while $\mathcal{A}_2$ assesses the likelihood of successful grasping. These networks are implemented using Multilayer Perceptron (MLP) models that take feature vectors f_o and f_{p_i} as inputs. The collective affordance scores are then used to create an affordance map, as illustrated in Fig. 6.

Proposal Network: The Proposal Network $\mathcal{P}_1$ is designed to output the gripper displacement $\Delta \vec{x}_1$ at the specified contact point p_1. It utilizes a conditional Variational Autoencoder (cVAE) architecture [42] with 32 hidden dimensions. Within this structure, the encoder maps the feature vectors f_o, f_{p_1}, and f_{M_1} into a Gaussian noise vector $z \in \mathbb{R}^{32}$, which the decoder then reconstructs. Similarly, $\mathcal{P}_2$ operates with the same architecture but is focused on receiving and reconstructing the gripper orientation $\vec{\theta}_2$.

Critic Network: A network $\mathcal{C}_1$ evaluates the efficacy of a pre-grasping operation by assigning a score $\mathcal{C}_1((p_1, \Delta \vec{x}_1) \mid \mathcal{O}_1) \in \mathbb{R}$. This score considers the pair $(p_1, \Delta \vec{x}_1)$ proposed by $\mathcal{A}_1$ and $\mathcal{P}_1$. Similarly, $\mathcal{C}_2$ assesses the likelihood of success for a grasping operation, providing a score $\mathcal{C}_2(p_2, \vec{\theta}_2 \mid \mathcal{O}_2) \in [0, 1]$. Both critic networks utilize an MLP to process the input features f_o, f_{p_i}, and f_{M_i}.

D. Inference

The inference pipeline of the proposed method includes four stages, which are described respectively below.

Pre-grasping Necessity Check: Prior to the pre-grasping step, a necessity check is performed to determine whether this step can be skipped. This decision is based on an evaluation of *graspability*, where the network $\mathcal{C}_2$ assesses manipulation proposals generated by $\mathcal{A}_2$ and $\mathcal{P}_2$. The success rate $\hat{c}_2$ is calculated by averaging the scores according to the following equation:

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

If $\hat{c}_2$ exceeds the threshold θ_g, the object will be grasped directly according to the proposal with the highest score.

Pre-Grasping Manipulation Inference and Implementation: During the inference stage, the affordance network $\mathcal{A}_1$ assesses each potential contact point's affordance value. The top n_1 points, denoted as p_1^j (for $j = 1, 2, \dots, n_1$), are selected based on the highest scores. Subsequently, for each selected point p_1^j, the proposal network $\mathcal{P}_1$ gener-

ates m_1 potential pre-grasping manipulations $\Delta\vec{x}_1^{jk}$ (where $k = 1, 2, \ldots, m_1$), each paired with a randomly generated Gaussian noise vector z_1^{jk}.

The critic network $\mathcal{C}_1$ then evaluates these manipulations to select the optimal pair $(p_1^{j*}, \Delta\vec{x}_1^{j*k*})$ for execution, based on the following criteria:

$$p_1^j(\mathcal{O}_1) = \underset{p_1}{\arg\max}^{(n_1)} \mathcal{A}_1(p_1 \mid \mathcal{O}_1), \quad (j^*, k^*) = \underset{j,k}{\arg\max}\, \mathcal{C}_1(p_1^j, \mathcal{P}_1(p_1^j, z_1^{jk}) \mid \mathcal{O}_1). \quad (4)$$

This procedure modifies the object's pose within the environment, resulting in a new point cloud $\mathcal{O}_2$, which serves as the input for the subsequent grasping module.

Grasping Manipulation Inference and Implementation: Following the pre-grasping step, as discussed in Sec. III-D, the object is ready to be grasped. The final selection of the manipulation technique is determined by the highest critic score. This score is produced by evaluating the combined output of the affordance network $\mathcal{A}_2$ and the proposal network $\mathcal{P}_2$. The manipulation with the highest score is then executed to grasp the object. This ensures that the chosen approach is optimal based on the current state of the object within the environment.

Closed-loop Control: The inference pipeline of `PreAfford` forms a closed-loop control system, enabling iterative refinement of the object's position to achieve a configuration suitable for grasping. This optional feature triggers additional manipulations when an initial pre-grasping action fails to render the object graspable. The closed-loop process operates as outlined below:

1) The grasping module initially assesses the expected success rate of the grasp, denoted by $\hat{c}_2$.
2) If $\hat{c}_2$ is below a predefined threshold θ, the pre-grasping module executes a manipulation to reposition the object.
3) The system then returns to step 1 to reassess $\hat{c}_2$ under the new object configuration.
4) This cycle (steps 2 and 3) repeats until $\hat{c}_2$ exceeds the threshold, indicating that the object is in a graspable position, at which point an attempt to grasp is made.

This feedback mechanism ensures that the system dynamically adapts to environmental changes, enhancing the likelihood of successful grasping through iterative adjustments.

E. Training and Losses

Critic Loss: The loss function for the critic network $\mathcal{C}_2$ is formulated based on the binary outcome (success or failure) of the grasping manipulation, denoted by r. The loss is defined as follows:

$$\mathcal{L}_{\mathcal{C}_2} = r \log(\mathcal{C}_2(p_2, \vec{\theta}_2)) + (1 - r) \log(1 - \mathcal{C}_2(p_2, \vec{\theta}_2)). \quad (5)$$

This binary cross-entropy loss measures the performance of $\mathcal{C}_2$ in predicting the success of grasping actions.

For $\mathcal{C}_1$, the loss is computed based on how effectively the pre-grasping manipulation improves the likelihood of a successful grasp. The success rates before and after the manipulation are evaluated by the grasping module, yielding $\hat{c}_2^{\text{before}}$ and $\hat{c}_2^{\text{after}}$, respectively. To encourage stable and safe manipulation actions, we apply penalties to the improvement in grasping success likelihood ($\hat{c}_2^{\text{after}} - \hat{c}_2^{\text{before}}$) based:

- **Displacement Penalty:** Defined as $p_d = \exp(-|\Delta\mathbf{x}_{\text{go}}|/a)$, where $\Delta\mathbf{x}_{\text{go}}$ represents the relative displacement between the grippers and the object, and a is a scaling coefficient.
- **Rotation Penalty:** Defined as $p_r = \exp(-\sqrt{\alpha^2 + \beta^2 + \gamma^2}/b)$, where α, β, and γ are the Euler angles describing the rotation, and b is a scaling coefficient.
- **Safety Penalty:** p_s is initially set to 1, but is reduced to 0 in scenarios such as the object falling off the table, or collisions involving the gripper with walls or slopes.

The total penalty term p is computed as the product of p_d, p_r, and p_s. The loss for $\mathcal{C}_1$ incorporates these penalties and is defined using the l_1 loss metric:

$$\mathcal{L}_{\mathcal{C}_1} = \left| \mathcal{C}_1(p_1, \Delta\vec{x}_1 | \mathcal{O}_1) - p \cdot (\hat{c}_2^{\text{after}} - \hat{c}_2^{\text{before}}) \right|. \quad (6)$$

Proposal Loss: The networks $\mathcal{P}_1$ and $\mathcal{P}_2$ are implemented as cVAEs, designed to generate appropriate actions for manipulation tasks. The training process exclusively uses data from *successful* pre-grasping and grasping manipulations to ensure the generation of effective actions. The loss function for these networks is a combination of geometric and Kullback-Leibler (KL) divergence losses. The geometric loss, $\mathcal{L}_{\text{geo}}$, quantifies the discrepancy between the reconstructed manipulation, M_i, and the ground truth, $\hat{M}_i$. The KL divergence, D_{KL}, assesses the deviation of the hidden layer distribution from a standard normal distribution. The total loss for each proposal network is expressed as: $\mathcal{L}_{\mathcal{P}_i} = \mathcal{L}_{\text{geo}}(M_i; \hat{M}_i) + D_{\text{KL}}(z(p_i, \hat{M}_i, \mathcal{O}_i) \| \mathcal{N}(0, 1))$.

Affordance Loss: The affordance score is a critical measure in robotic manipulation that assesses the suitability of contact points, guiding the selection process during action proposals. It evaluates the efficacy of actions generated by the proposal networks $\mathcal{P}_i$. These scores are calculated offline using the critic networks, which gauge the performance of multiple actions under varying conditions. The affordance network is trained subsequent to the other networks in each module. The dataset for training is labeled with the average critic scores $\hat{a}_{p_i}$, computed from n_i actions generated by $\mathcal{P}_i$ using different Gaussian noise vectors z_j ($j = 1, 2, \ldots, n_i$). The affordance loss, $\mathcal{L}_{\mathcal{A}_i}$, measures the absolute difference between the predicted and the averaged critic scores:

$$\hat{a}_{p_i} = \frac{1}{n_i} \sum_{j=1}^{n_i} \mathcal{C}_i(p_i, \mathcal{P}_i(p_i, z_j) \mid \mathcal{O}_i), \quad \mathcal{L}_{\mathcal{A}_i} = |\mathcal{A}_i(p_i \mid \mathcal{O}_i) - \hat{a}_{p_i}|. \quad (7)$$
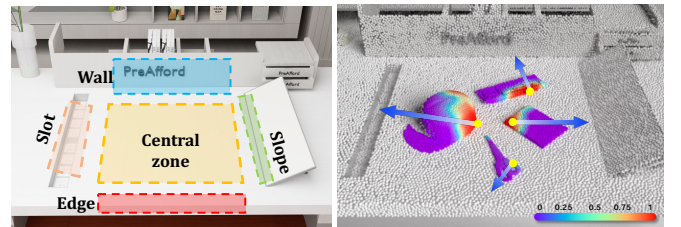
(a) A complex environment　　　(b) Affordance heatmap

Fig. 4: **Multi-feature scenario:** `PreAfford` effectively addresses scenarios where multiple environmental features are present simultaneously.

F. Data Collection

We used the Sapien simulation platform [43] to collect data for robotic pre-grasping and grasping tasks. Our dataset, drawn from ShapeNet-v2, includes 5 hard-to-grasp and 5 easy-to-grasp object categories, each with more than 10 unique shapes.

The INSPIRE-ROBOTS EG2-4C gripper model was used in grasping trials across four simulated scenes. Objects were randomly positioned, sometimes on environmental features like tables or ledges. Grasping points were randomly chosen from the object's surface, with gripper orientations determined by a hemisphere above each point's tangent plane.

The pre-grasping dataset concentrated on hard-to-grasp categories. Objects were placed at random distances from features, and the gripper performed a pushing action with a normally distributed displacement at random surface points. To improve data collection efficiency by about 42%, the displacement direction towards features was preset in 30% of trials based on domain knowledge, with interaction angles modeled by a Gaussian distribution.

Both datasets positioned the camera 3–5 meters away, oriented towards the object to simulate realistic robotic vision constraints. Each dataset included 10,000 successful and 30,000 unsuccessful interactions. The data-collection process takes about 8 days on 256 AMD EPYC 7742 processors.

IV. EXPERIMENTS

A. Simulation Settings and Datasets

We tested `PreAfford` on 5 seen (hat, laptop, scissors, keyboard, and tablet) and 4 unseen (phone, pliers, switch, and window) categories of hard-to-grasp objects across 5 scenes.

These scenes include four with a single environmental feature (edge, wall, slope, and slot) and one novel scene (shown in Fig. 4) that contains all four features simultaneously, which is constructed to test adaptability to complex and unseen environments. We conducted 1,000 tests on each object-environment pair and calculated the mean success ratio.

B. Simulation Evaluation Metrics and Baselines

To assess pre-grasping proposals, we measure the increase in grasping success rate following a pre-grasping manipulation. After each manipulation, a grasping proposal is generated and tested 1,000 times per object-environment pair for robust success rates. The effectiveness of the pre-grasping manipulation is evaluated by comparing the enhanced success rate against scenarios without pre-grasping.

We compare against four baselines: (i) W/o pre-grasping: direct grasping without pre-grasping actions; (ii) Random-direction Push: the contact point is set by our pre-grasping module, but displacement is random; (iii) Center-point Push: displacement is set by our module, but the contact point is at the object's geometric center; (iv) Ours w/o closed-loop: an ablation study eliminating closed-loop control from our method. Additionally, our method's compatibility is demonstrated across 5 training and 4 testing categories of easy-to-grasp objects.

C. Simulation Results and Analysis

Efficacy and Adaptability: Figure 5 illustrates the pre-grasping and grasping affordance maps predicted by our networks alongside the displacement suggested by $\mathcal{P}_1$. Key findings include: (i) **Environmental awareness**: Our network effectively recognizes environmental features and
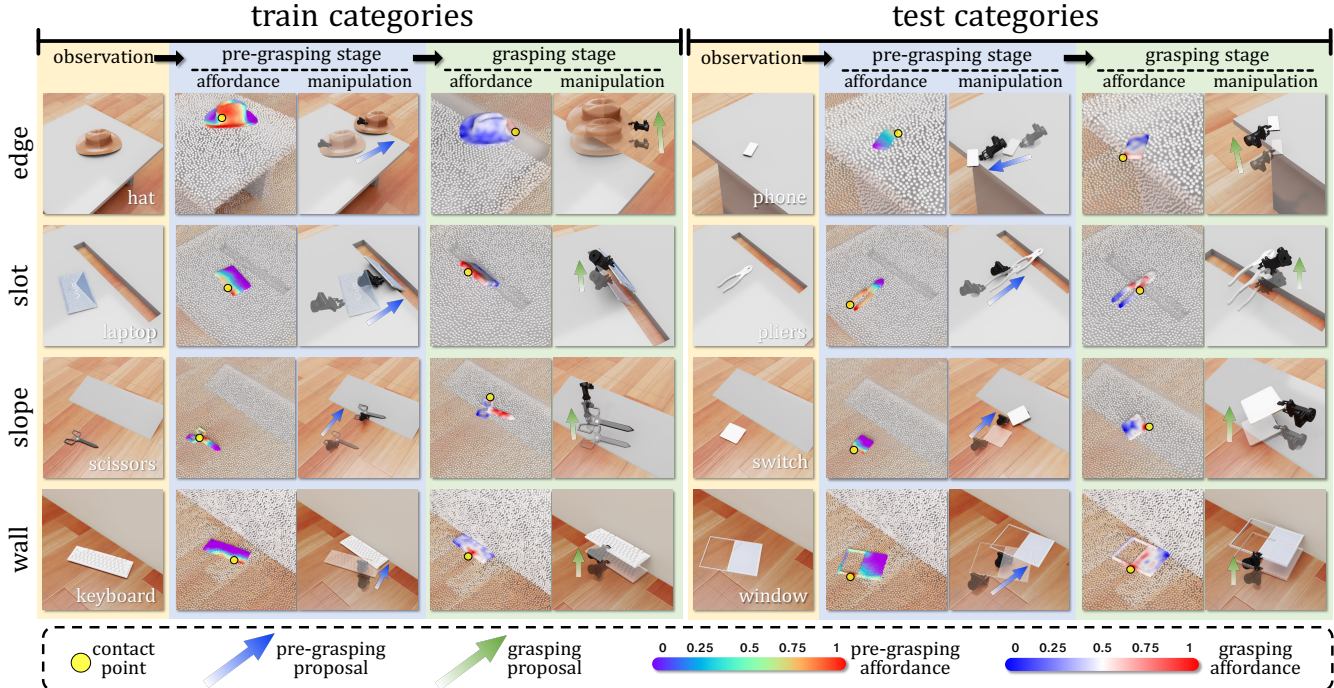

Fig. 5: **Qualitative Results.** We demonstrate pre-grasping manipulation on training and testing categories in four scenarios—edge, slot, slope, and wall. Affordance maps highlight effective interaction areas, showing `PreAfford`'s capability to devise suitable pre-grasping and grasping strategies for various object categories and scenes, including both seen and unseen objects.

proposes suitable pre-grasping actions, directing movements towards beneficial landscapes. (ii) **Dynamics awareness**: It also displays an understanding of object dynamics, favoring pushes that align with the center of mass to reduce rotation.

The adaptability of `PreAfford` is showcased by effective pre-grasping proposals across varied scenes and objects, as seen in Fig. 4, which handles complex environments with multiple features adeptly.

Table II quantitatively validates `PreAfford`, highlighting its superiority over direct grasping in challenging situations. Both Random-direction Push and Center-point Push strategies underline the importance of informed direction and contact point selection. Additionally, the ablation study without closed-loop control emphasizes the critical role of this feature in enhancing success rates.

TABLE II: **Success rates of grasping manipulation in percentage.** Pre-grasping increases grasping success rates by 52.9%. A closed-loop strategy further enhances this improvement by 16.4% across 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**

Compatibility: As detailed in Sec. III-D, cases where the estimated grasping success likelihood $\hat{c}_2$ falls below a threshold θ_g prompt a pre-grasping manipulation. The selection of θ_g is pivotal for ensuring compatibility with graspable objects and for activating necessary pre-grasping manipulations in ungraspable scenarios. Our tests in the Multi scene demonstrate that a θ_g of 0.8 strikes an optimal balance, as evidenced in Table III.

TABLE III: **Rate of performing a direct grasping on both graspable and ungraspable objects.** While $\theta_g = 0.8$, most ungraspable objects would be pre-grasped, while graspable objects not. G: graspable; U: ungraspable.

Metric	Train		Test	
	G	U	G	U
pre-grasping rate	14.7	84.1	23.4	77.5
success rate	83.7	74.1	80.0	74.6

D. Real-world Experiment Setup

We implemented our algorithm on the AIRBOT Play robotic arm, a compact six-degree-of-freedom manipulator. To capture RGB-D data, we used an ORBBEC Femto Bolt

(a) Object categories for testing.

(c) Slope.

(d) Slot.

(b) Multi-scene setup with three environmental features. Hardware includes an AIRBOT Play robotic arm, an INSPIRE-ROBOTS gripper, and a Femto Bolt RGB-D camera.

(e) Wall with edge.

(f) Edge.

Fig. 7: **Real-world experiment setups.**

camera. The system's end effector is an INSPIRE-ROBOTS EG2-4C gripper.

The experimental arrangement included ten object categories, split into five known and five unknown categories during training, as depicted in Fig. 7(a). To assess the adaptability of `PreAfford`, we designed five distinct scenes incorporating various environmental features: edges, walls, slopes, slots, and a *Multi* scene combining all features, illustrated in Fig. 7(b)-(f). Fig. 6 demonstrates both the pre-grasping and grasping manipulations executed by the system, including the generated pre-grasping affordance maps.

E. Real-world Results and Analysis

Table IV shows the outcomes from our real-world experiments. We tested ten object categories, conducting four trials per object in each scene, resulting in 20 experiments per scene. Our framework, `PreAfford`, significantly enhanced the grasping success likelihood for hard-to-grasp objects across diverse categories, demonstrating its high deployability in real-world scenarios.

V. CONCLUSION AND LIMITATION

In this paper, we introduced a novel two-stage affordance learning framework that excels in adaptability across various

(a) move a tablet to table edge (b) push a plate towards a wall (c) push a keyboard up a slope (d) slide a tablet into a slot
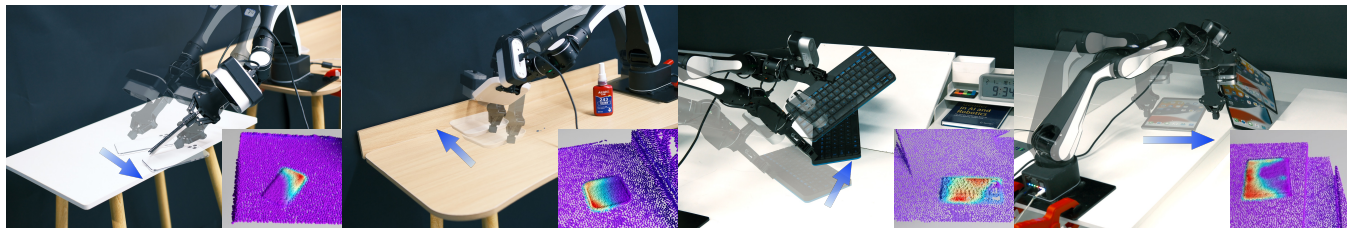
Fig. 6: **Real-world pre-grasping manipulations with affordance maps.** Red areas in the maps indicate optimal pushing locations. Point clouds are captured by Femto Bolt.
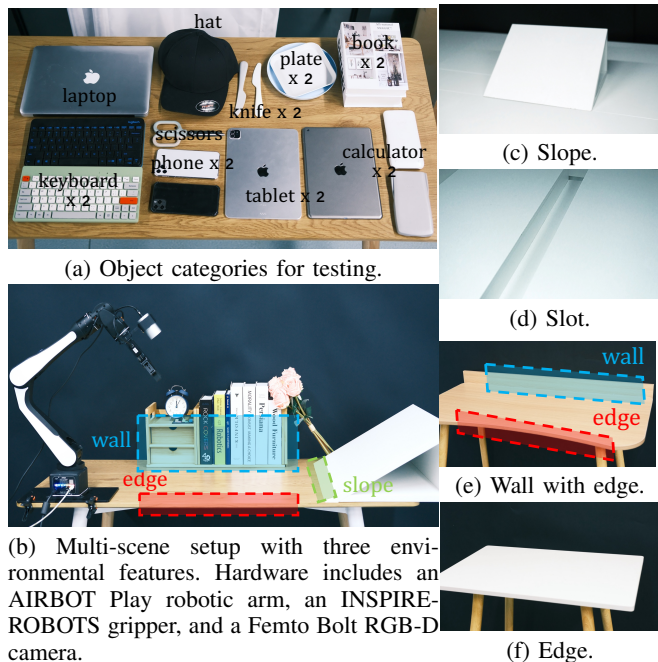
TABLE IV: **Real-world experiment results.** Experiments were conducted twice for each object in every scene, comparing direct grasping (without pre-grasping) to grasping after pre-grasping. Success rates are presented as percentages.

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

object-environment configurations, shows compatibility with graspable objects, and proves deployable through rigorous real-world testing. This framework has been validated in simulation and real-world settings, underscoring its effectiveness. However, our method still faces challenges, particularly in handling objects with highly irregular shapes or extremely dynamic environments. Future work will focus on enhancing the robustness and flexibility of our approach to address these issues.

REFERENCES

[1] 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.

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

[3] 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.

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

[5] 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.

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

[7] 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.

[8] W. Zhou and D. Held, "Learning to grasp the ungraspable with emergent extrinsic dexterity," in *CoRL*, 2023.

[9] J. E. King, M. Klingensmith, C. M. Dellin, M. R. Dogar, P. Velagapudi, N. S. Pollard, and S. S. Srinivasa, "Pregrasp manipulation as trajectory optimization.," in *RSS*, 2013.

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

[11] 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.

[12] 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.

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

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

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

[22] 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.

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

[24] 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.

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

[26] 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.

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

[28] 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.

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

[30] 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.

[31] 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.

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

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

[34] 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.

[35] 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.

[36] 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.

[37] 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.

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

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

[40] 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.

[41] 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.

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

[43] 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.