SA-DEM: Dexterous Extrinsic Robotic Manipulation of Non-Graspable Objects via Stiffness-Aware Dual-Stage Reinforcement Learning

Yanzhe Wang, Wei Yu, Hao Wu, Haotian Guo, Huixu Dong†, *Member, IEEE*

***Abstract*—We propose a novel framework for extrinsic dexterous manipulation in robotics, termed SA-DEM. This approach is grounded in dual-stage reinforcement learning and decouples the challenge of pose adjustment for ungraspable objects through non-grasping manipulation into two distinct phases: interactive mode decision-making and manipulation action planning. Notably, this framework innovatively incorporates the stiffness information of manipulated objects into the decision-making process, enabling the robot to autonomously perceive, decide, and plan manipulation strategies for objects with diverse physical attributes. The first phase of SA-DEM involves a high-level agent responsible for planning the grasping pose of objects and their interaction locations with the environment, based on the initial state of the objects, observations from environmental point clouds, stiffness representations, and prior knowledge of grasping regions. The second phase is executed by a low-level agent, which focuses on planning specific manipulation actions such as poking and flipping. These actions are derived from autonomous exploration during the training process, negating the need for manual customization. Both agents employ a hybrid discrete-continuous action space along with time-abstracted and spatially grounded representations centered around the point cloud, culminating in a unified actor-critic reinforcement learning framework.**

***Note to Practitioners*—By utilizing external resources like environmental contact and dynamic interaction, common low degree-of-freedom (DoF) parallel jaw grippers can achieve dexterity beyond their design capabilities. Existing research on extrinsic dexterous manipulation is often limited by constrained environments, manually designed meta-parameters, and specific interaction types, frequently overlooking the influence of object stiffness. This oversight results in methods designed for rigid objects facing challenges when manipulating soft ones. Our proposed reinforcement learning-based SA-DEM framework aims to address these challenges. Simulations and real-world experiments validate SA-DEM's effectiveness in zero-shot transfer and generalization. This method enables non-grasping manipulation of objects with varying geometries, masses, and stiffnesses, relying solely on the planar and vertical supports provided by the environment. SA-DEM shows great promise for intelligent object manipulation solutions in applications involving low-DoF grippers, such as industrial automation, warehousing, logistics, and service robotics.**

***Index Terms*—Extrinsic dexterity, Non-grasping manipulation, Reinforcement learning**

Yanzhe Wang, Wei Yu, Hao Wu, Haotian Guo and Huixu Dong are with Grasp Lab of Mechanical Engineering Department, Zhejiang University, Hangzhou 310058, China. (†The corresponding author Huixu Dong, e-mail: huixudong@zju.edu.cn).

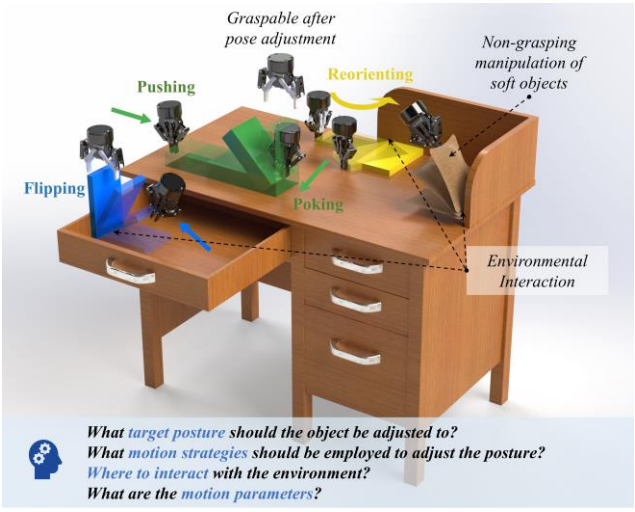

Fig. 1. Examples of non-grasping manipulation planned by the proposed method, including in-plane pushing, poking, and environment-assisted reorienting and flipping. Its innovation lies in autonomously determining the target posture and motion strategies based on the object's stiffness properties, while generating specific motion parameters.

I. INTRODUCTION

Human-like dexterous robotic hands pose significant technical and cost challenges, prompting a preference for simpler parallel jaw grippers, such as two-finger and three-finger designs, which are easier to implement and control in practical applications [1]. For objects that are difficult to grasp or in constrained workspaces, parallel jaw grippers can manipulate objects in ways that extend beyond traditional grasping techniques. This approach, referred to as "Extrinsic Dexterity," leverages interactions with the surrounding environment to achieve dexterous manipulation [2]. Through non-grasping actions such as planar pushing [3], flipping with the aid of flat surfaces [4], and lifting using vertical supports [5], grippers with fixed DoF can demonstrate manipulation capabilities comparable to those of multi-fingered robotic hands, all without the need to alter their hardware structure. These systems can achieve remarkable dexterous flexibility by coordinating the object, environment, and fingers.

Previous research on extrinsic dexterity has illustrated the ability of parallel jaw grippers to perform various non-grasping tasks. However, these studies predominantly rely on manually designed motions or primitives [6], [7] further make fixed assumptions about object properties and contact modes [8], [9].

The focus has primarily been on rigid objects, which limits manipulation planning to a narrow range of patterns [4] [5]. In contrast, real-world manipulation involves a complex and diverse range of objects. The contact dynamics and nonlinear changes that arise from interactions with materials of varying stiffness are challenging to predict and model accurately. Consequently, strategies designed for rigid objects may not be suitable for soft ones. Reinforcement learning offers a promising approach to overcoming the randomness and unpredictability of physical interactions through trial and error, enabling generalized manipulation capabilities across different objects [18]. Nevertheless, incorporating object stiffness into learning manipulation strategies presents significant challenges. This is primarily due to the difficulty in generalizing manually designed action primitives, which struggle to effectively address issues related to stiffness coupling. As a result, there is currently no flexible non-grasping manipulation method that adequately adapts to objects with varying stiffness properties.

To address these challenges, we proposed a method called SA-DEM for dexterous extrinsic manipulation of non-graspable objects via stiffness-aware dual-stage reinforcement learning. This approach decouples decision-making from execution planning in non-grasping manipulation, with high-level and low-level agents generating distinct output policies. First, a hybrid discrete-continuous action space is introduced, enabling both agents to produce outputs in the same action format. This unification enables the application of a unified actor-critic reinforcement learning framework to effectively address interactive decision-making and action planning. Both agents utilize a time-abstracted and space-grounded object-centric representation, with the high-level agent identifying the object's target location from the environmental point cloud and the low-level agent determining the gripper's contact location from the object point cloud. This cohesive architecture streamlines the design and ensures consistency across policies. Second, the constructed high-level agent integrates initial observations from various sources, including the environmental point cloud, the object point cloud, the stiffness point cloud, and the prior point cloud from the grasping region. By synthesizing this perceptual information, the agent infers the necessary target pose for the object and identifies the corresponding action types required by the low-level agent. To tackle the issue of sparse reward feedback, a proposed mapping method converts discrete rewards into a continuous space, thereby providing consistent and effective reward signals to enhance learning efficiency. Notably, by encoding the stiffness information and the prior grasping region into unified 3D point clouds, the agent eliminates the need to separately handle different modalities, thereby increasing the perceptual information density. Furthermore, the low-level agent functions as the action execution module, generating gripper contact points and action parameters based on the decisions made by the high-level agent and the real-time observations of the object point cloud. The action skills acquired by this low-level agent, such as planar poking and environmental flipping, are developed spontaneously during training. Finally, this approach autonomously explores environmental layouts and object attributes, generating long sequences of contact interactions based on the stiffness characteristics of the objects, thus allowing ungraspable objects to be adjusted into suitable poses for grasping.

In summary, the main contributions of this work include:

- A novel dual-stage reinforcement learning method that integrates object stiffness for the first time to achieve flexible, autonomous planning for non-grasping manipulation.
- A unified 3D spatial observation representation and a consistent actor-critic framework that effectively addresses pose decision-making and action interaction in extrinsic dexterous manipulation.
- A data-driven learning strategy for the high-level agent that integrates discrete sparse action decisions into a continuous reward representation, significantly enhancing training performance.
- The trained policies demonstrate exceptional perceptual decision-making capabilities and dynamic manipulation skills, enabling it to autonomously determine target poses for objects of varying stiffness and execute diverse actions such as pushing, poking, and flipping.

II. Related Work

A. Extrinsic Dexterity

Low-DoF robotic parallel jaw grippers can exhibit extrinsic dexterity through gravity, dynamic motion, or external contact, enabling them to perform complex manipulation tasks. Early research primarily imposed environmental constraints while manually designing interactive actions and controllers. For instance, Dafle et al. [2] developed a series of manually programmed open-loop trajectories for grasping postures tailored to specific objects. Eppner et al. [10] customized control strategies for sliding to the edge of a table or pushing against a wall, leveraging environmental constraints. Bimbo et al. [11] employed edge sliding to facilitate manipulation. However, these methods are limited to known environments and constrained interaction conditions.

Combining primitives can achieve more complex tasks by refining interactive actions and decomposing them into motion primitives. Eppner et al. [6] designed action primitives based on environmental measurements to inform grasping strategies. Hou et al. [7] proposed two motion primitives, rotation and compliant rolling, addressing the reorientation problem through hierarchical planning. Nonetheless, such approaches rely on fixed primitives, which can create gaps in actions that extend beyond predefined ranges.

With the introduction of contact-mode-guided manipulation methods by Cheng et al. [8], [9], the manual design of motion primitives has become unnecessary. These methods automatically enumerate contact modes between the environment and objects to identify effective manipulation sequences. Chavan et al. [12] provided a geometric definition and algebraic analysis of friction dynamics, abstracting a

feasible motion cone for planar pushing. However, these methods are limited by assumptions regarding interactions and specific ranges of contact adhesion.

Recently, learning-based methods for manipulation have gained attention due to their applicability in complex contact interactions [13], [14], demonstrating potential in skill complexity and operational flexibility[15], [16], [17]. For example, Chi et al. [18] employed diffusion models to represent robotic visual motion strategies, illustrating the feasibility of generating behaviors such as planar pushing and object pose adjustment. Kim et al. [19] developed a Tac2Pose-based controller [20] that leverages environmental interactions to control object poses. Zhou et al. [4] proposed a hybrid discrete-continuous action representation using reinforcement learning to implement a poking strategy for planar objects based on point cloud observations. However, this method provides only a single interaction mode and requires a predetermined target pose. Yang et al. [21] introduced a hierarchical reinforcement learning framework that processes depth images to output pixel-level Q-values, focusing solely on a single wall-flipping strategy, which limits the selection of interaction contact points.

Overall, existing learning-based methods for extrinsic dexterity primarily concentrate on rigid body manipulation, often neglecting the substantial influence of object deformation on interaction outcomes. These approaches typically rely on specific manipulation modes or are limited to predefined actions, constraining their applicability in diverse scenarios. This highlights the necessity for a novel strategy integrating object characteristics into robotic systems' decision-making processes while offering various adaptive and responsive manipulation skills.

B. Non-Grasping Manipulation Planning

Grasping is a critical task in robotic manipulation, with extensive research dedicated to the generation and execution of grasping postures [22]. However, when the grasp configuration conflicts with the environment, failing direct grasping, non-grasping manipulation becomes necessary to adjust the object's pose. This adjustment leverages interactions among the robot, the object, and the environment to achieve extrinsic dexterity [23], [24], [25], [26]. The variability in contact introduces high-dimensional, non-convex planning challenges, leading to lengthy sequences of object pose adjustments. Posa et al. [27] formulated multi-contact dynamics as a linear complementarity problem, utilizing sequential quadratic programming to optimize local trajectories. However, this method is primarily applicable to objects with regular geometries.

Analyzing contact kinematics allows feasible contact paths to be identified within a multi-redundant configuration space. Huang et al. [28] transformed multi-rigid-body hybrid dynamics problems into combinatorial geometry problems, solving for effective motion polyhedral cones. Cheng et al. [29] proposed a planning method based on a Monte Carlo tree search to optimize object motion and contact modes. Other works [8] and [9] have also employed sampling methods to plan contact sequences. However, despite their ability to generate continuous motion paths, the frequent feasibility checks required by these methods can hinder planning efficiency.

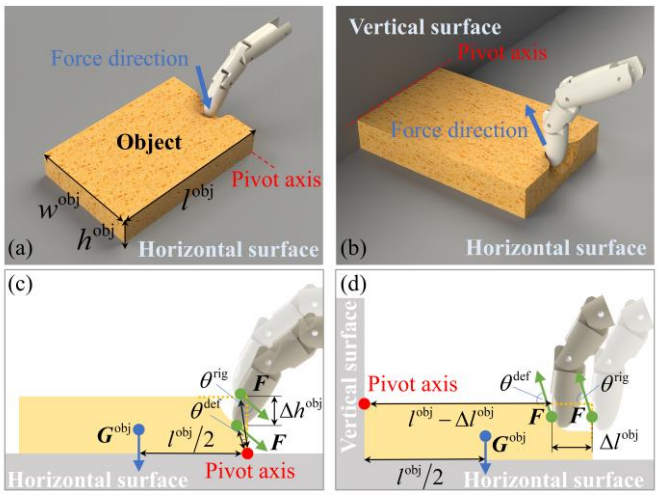

Fig. 2. The impact of object stiffness on deformation during contact for poking (a) and flipping (b), along with the simplified models (c) and (d). "rig" and "def" indicate rigid and deformable objects, respectively, assuming a consistent force angle. Δ represents the compression length.

Recent research has explored learning-based methods to tackle non-grasping manipulation tasks. Wu et al. [30] introduced a spatial action map and employed reinforcement learning for motion planning in mobile manipulation tasks. Liang et al. [31] proposed hybrid force-velocity controllers that learn to predict the success rates of candidate contact actions and plan motions to achieve target poses. Lee et al. [32] combined vision-based interactive policy distillation with reinforcement learning to tackle object stacking and assembly tasks. Zhou et al. [5] concentrated on a singular strategy for flipping objects in unfavorable configurations, which requires initial positioning and tracking of object poses.

In summary, existing methods typically rely on a predetermined target pose for the object, training manipulation strategies to generate actions that reduce discrepancies without adequately addressing the necessity for pose decision-making in general scenarios. The ability of an agent to autonomously determine the target pose for non-grasping manipulation adjustments based on environmental conditions and the object's initial state is critical for adapting to a broader range of applications.

III. Task Definitions and Assumptions

This research addresses a critical yet underexplored issue: the impact of object stiffness on non-grasping manipulation. In particular, for interaction actions such as poking and flipping, the stiffness of the object affects the deformation during contact, thereby altering the initial applied force, as illustrated in Fig.2.

To simplify the problem modeling, the moment arm is approximated as parallel to the corresponding edge while neglecting the plastic deformation of the object and assuming there is no relative slipping between the object, the environment, and the manipulator. Given the object's

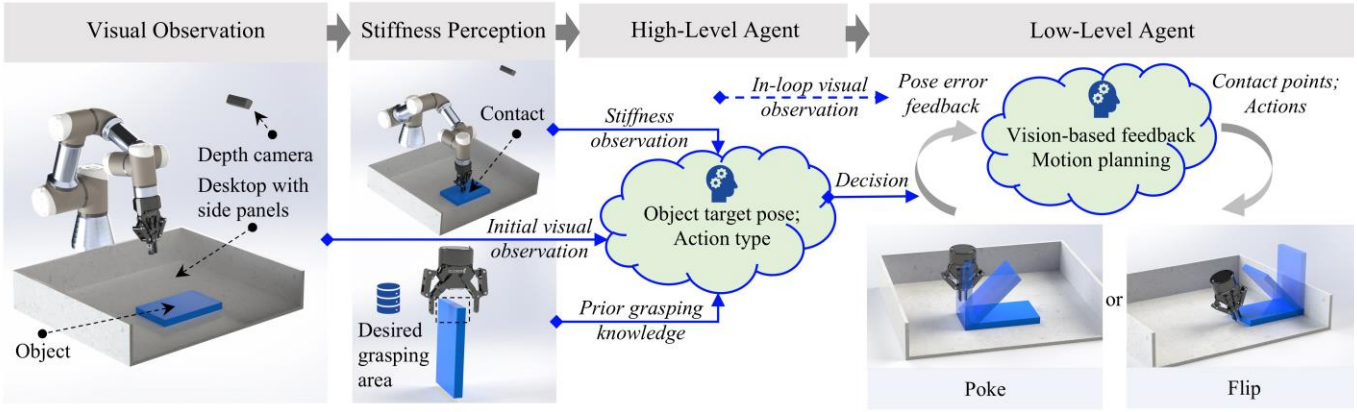

Fig. 3. An overview of SA-DEM framework. The implementation involves four steps: First, visual observation is conducted to obtain the initial point cloud of the environment and the object. Second, the stiffness information of the objects is perceived and represented as a point cloud of object deformation. Third, the high-level agent integrates visual observations, stiffness information, and priors on grasp positions to determine the target pose and the action type. Finally, the low-level agent continuously monitors the state of the object and plans manipulation actions until it reaches the target pose.

dimensions $l^{\text{obj}} \times w^{\text{obj}} \times h^{\text{obj}}$, gravitational force $\boldsymbol{G}^{\text{obj}}$, material Young's modulus E, and the cross-sectional area A of the contact region, the condition under which flipping can be more efficient than poking for deformable objects is expressed as

$$\left(l^{\text{obj}} - h^{\text{obj}}\right)\boldsymbol{F}\sin\theta^{\text{def}} > \left(l^{obj} - h^{obj}\right)\boldsymbol{F}^2 \sin 2\theta^{def}/2AE \quad (1)$$

where $\boldsymbol{F}$ is the interaction force. θ^{def} represents the angle between the interaction force and the moment arm for the soft objects. Consequently, the range of applied force that optimally favors flipping can be deduced as

$$\boldsymbol{G}^{\text{obj}} l^{\text{obj}}/2\left(d - \Delta d\right)\sin\theta^{\text{def}} < |\boldsymbol{F}| < AE/\cos\theta^{\text{def}} \quad (2)$$

In this context, for poking, d and Δd correspond to h^{obj} and Δh^{obj}, respectively. For flipping, d and Δd correspond to l^{obj} and Δl^{obj}, respectively.

The contact force in this hypothesis is constrained by an upper limit, indicating that the flipping strategy can achieve posture adjustment with a smaller interaction force compared to poking. Consequently, flipping is particularly effective for adjusting the posture of moderately deformable objects and thin items. However, additional pre-adjustment steps are required to align the object's vertical support face with the environment. In contrast, poking is beneficial for rigid objects and those with a certain thickness that cannot be grasped, as it allows for non-grasping adjustments around the initial posture while minimizing the cost of planar movement.

IV. Methodology

A. Framework Overview

This paper presents a dual-stage reinforcement learning-based framework for dexterous extrinsic manipulation called SA-DEM. This framework addresses challenges such as redirecting initially ungraspable object poses, making decisions regarding environmental interactions, and planning object manipulation in non-grasping postures. An overview of the framework is illustrated in Fig.3. The framework decouples complex and stochastic contact interaction problems into a hierarchical structure encompassing global decision-making and motion execution, without relying on manually designed motion primitives or explicit dynamic model assumptions.

In this framework, the high-level agent is tasked with perceiving and integrating observations of the environmental state, performing stiffness sensing, and gathering global information about desired object-grasping regions. This agent utilizes reasoning to determine appropriate interaction poses for objects relative to the environment, which includes identifying target poses after object redirection and specifying the types of non-grasping adjustments needed for robot manipulation.

The low-level agent, guided by the global decisions made by the high-level agent, continuously monitors current objects and environmental information. It deduces optimal contact positions and post-contact motion parameters for robot-object interaction. The low-level agent constructs a three-dimensional, vision-driven, closed-loop feedback planner using temporal abstraction and spatial grounding representations. This system progressively moves the object toward the target poses generated by the high-level agent.

Both agents in SA-DEM are developed based on a Q-learning-based off-policy algorithm. Each agent formulates a Markov Decision Process (MDP) characterized by states $s_t \in \boldsymbol{\mathcal{S}}$, actions $a_t \in \boldsymbol{\mathcal{A}}$, a reward function $r: \boldsymbol{\mathcal{S}} \times \boldsymbol{\mathcal{A}} \to \mathbb{R}$, and a discount factor γ. Within each agent, an actor network generates a deterministic policy $\pi(a_t|s_t)$, while a critic network computes the Q-function $Q_\pi(s,a) = \mathbb{E}_\pi[\sum_{i=0}^{\infty}\gamma^i r_{t+i} \mid s_t = s, a_t = a]$. These networks are intricately designed with tailored network structures and data flows optimized for specific tasks.

B. Global Perception Integration and Decision-Making

The overall architecture of the proposed high-level agent is illustrated in Fig. 4.

Observation and Action Spaces. The observation space of the high-level agent comprises three main components: the environmental point cloud of the current scene $\boldsymbol{\mathcal{P}}^{\text{env}} =$

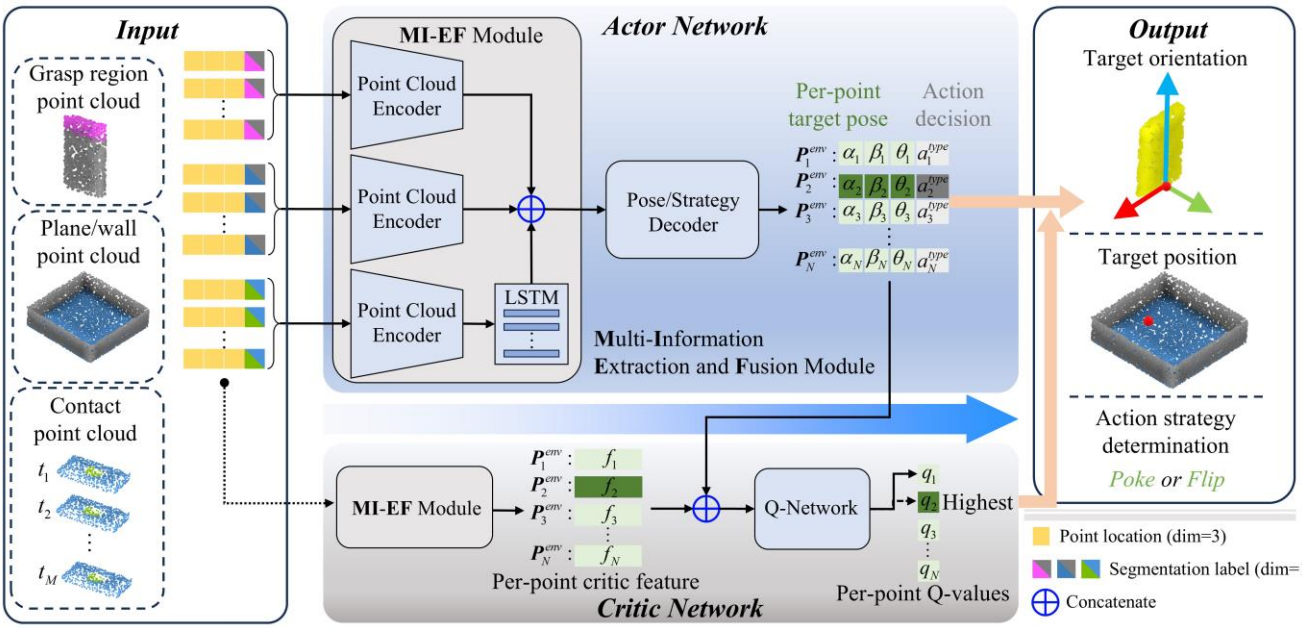

Fig. 4. Architecture of the high-level agent. This model uses point cloud observations that capture the environment, variations in object contact, and grasping region priors. The Actor generates an actor map from these observations, outputting the target pose and action type for each environmental point. Critic features connect to the actor map. A Q-network then produces a critic map. The output of the high-level agent consists of the point with the highest Q-value along with its associated target pose and action type.

$\{\boldsymbol{p}_i^{\text{env}}\}_{i=1}^{N_e}$, where $\boldsymbol{p}_i^{\text{env}}$ represents individual points in the environmental point cloud, segmented into plane regions and wall areas; the robot's tactile point cloud obtained from sequential object contacts under the current pose $\boldsymbol{\mathcal{P}}_t^{\text{con}} = \{\boldsymbol{p}_{t,i}^{\text{con}}\}_{i=1}^{N_t}$, where $\boldsymbol{p}_{t,i}^{\text{con}}$ denotes the points in the contact point cloud that are engaged by the gripper's fingers at time step t; and the point cloud representing expected grasp poses for the object sourced from an offline database $\boldsymbol{\mathcal{P}}^{\text{gsp}} = \{\boldsymbol{p}_i^{\text{gsp}}\}_{i=1}^{N_g}$, where $\boldsymbol{p}_i^{\text{gsp}}$ signifies points indicating graspable and non-graspable regions on the object. N_e, N_t, and N_g represent the number of points in each respective point cloud, respectively.

During environmental observation, the robot's end-effector moves to a reset position to acquire unobstructed and complete point clouds. The contact area point cloud in the tactile data $\boldsymbol{\mathcal{P}}_t^{\text{con}}$ consists of fingertip offset positions perceived by the robot after contacting the object, which is locally mapped to the initial object point cloud observed.

The action space is centered around the environmental point cloud, specifically identifying safe and stable grasping plane regions. This space encompasses three components: a repositioned target object position $\boldsymbol{T}^{\text{pos}}$ within the environmental point cloud, the reoriented target object orientation $\boldsymbol{T}^{\text{ori}}$, and the decision A^{type} for adjusting the object to the target pose. It integrates discrete object repositioning target positions with continuous reorientation directions and action-type decisions.

Model Architecture. Given the observation $s = \{\boldsymbol{\mathcal{P}}^{\text{env}}, \boldsymbol{\mathcal{P}}_t^{\text{con}}, \boldsymbol{\mathcal{P}}^{\text{gsp}}\}$, the high-level agent's actor network outputs the redirected object target orientation and robot action type strategy (poke or flip) for each point $\boldsymbol{p}_i$ in the environmental point cloud: $\pi_i^{\text{high}}(s) = \boldsymbol{a}_i^{\text{high}} = \{\boldsymbol{a}_i^{\text{pose}} \in \mathbb{R}^3, a_i^{\text{type}} \in \mathbb{R}\}$. The critic network computes the Q-value for each $\boldsymbol{p}_i$: $q_i = Q^{\text{high}}(s, \boldsymbol{a}_i^{\text{high}})$, representing the estimated return when using the actor's generated target orientation and action type at that point. The point corresponding to the maximum Q-value and the actor's output parameters determines the high-level agent's decision on where to place the object and which action type to execute.

To extract and integrate multiple masked information from point clouds, a Multi-Information Extraction and Fusion module (MI-EF) is introduced in the high-level agent, as illustrated in Fig.4. The MI-EF module comprises three point cloud encoders tailored to extract features from $\boldsymbol{\mathcal{P}}^{\text{gsp}}$, $\boldsymbol{\mathcal{P}}^{\text{env}}$ and sequential $\boldsymbol{\mathcal{P}}_t^{\text{con}}$. These encoders share the same structure but have independent parameters. To capture the dynamic characteristics of object shape changes during stiffness exploration, the MI-EF module incorporates an LSTM module. By embedding the feature sequences of $\boldsymbol{\mathcal{P}}_t^{\text{con}}$ over t time steps processed by the encoders into the LSTM module, the MI-EF module generates tactile encodings containing temporal tactile information and dynamic features. Subsequently, the encoded vectors from the three point clouds are concatenated to produce the fusion feature output of the MI-EF module. This feature, decoded by an actor's Pose/Strategy decoder composed of MLPs, represents target pose parameters $\boldsymbol{a}_i^{\text{pose}} = [\alpha_i, \beta_i, \theta_i]$, where $\alpha_i, \beta_i, \theta_i$ signify Euler rotation angles around the x, y, and z axes, respectively, constrained within $[0, 2\pi]$. Additionally, $a_i^{\text{type}} \in [-1, 1]$, with $a_i^{\text{type}} \leq 0$ corresponding to "poke" and $a_i^{\text{type}} > 0$ corresponding to "flip".

Using identical structures but independent parameters, the

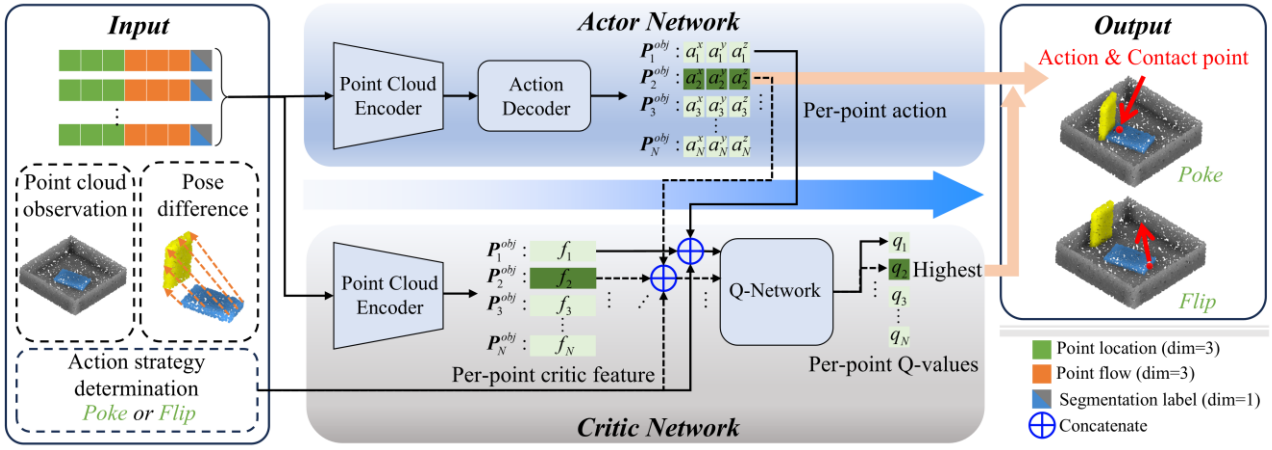

Fig. 5. Architecture of the low-level agent. This model employs point cloud observations of the environment and the object. The Actor processes point clouds and point flow to create an actor map that specifies motion parameters for each object point. Critic features connect to the actor map and the action decision of the high-level agent. These features are processed through a Q-network to generate a critic map. The output of the low-level agent includes the object point with the highest Q-value and its corresponding motion parameters.

MI-EF module for the critic extracts features from input point clouds. Each point's critic feature vector, concatenated with policy parameters generated by the actor at that point, undergoes processing by a Q-network comprising MLPs to derive the Q-value for each point. The highest-scoring point among all environmental points determines the position, target pose, and action type decision parameters for the global decision output, thereby guiding the subsequent motion execution of the Low-level agent.

The update rule is defined as follows: Let ψ denote the network parameters of the actor π^{high}. The actor's loss function $\mathcal{L}_{\psi}^{\text{actor}}$ is set to the expected Q-value of all points $\boldsymbol{p}_i$, weighted by their selection probabilities $\mu(\boldsymbol{p}_i|s)$:

$$\mathcal{L}_{\psi}^{\text{actor}} = \mathbb{E}_{\boldsymbol{p}_i\sim\mu}\left(\mathcal{L}_{\psi,i}^{\text{actor}}\right) = \sum_{i=1}^{N_e}\mu(\boldsymbol{p}_i|s)\cdot\mathcal{L}_{\psi,i}^{\text{actor}} \tag{3}$$

with

$$\mathcal{L}_{\psi,i}^{\text{actor}} = -Q^{\text{high}}\left(s,\pi_{\psi,i}^{\text{high}}(s)\right) = -Q^{\text{high}}\left(f_i^{\text{high}},\boldsymbol{a}_i^{pose},a_i^{\text{type}}\right),$$

$$\mu(\boldsymbol{p}_i|s) = \mu(\boldsymbol{p}_i|\mathcal{P}^{\text{env}}) = \exp(\lambda q_i)/\sum_{k=1}^{N_e}\exp(\lambda q_k),$$

where λ denotes the temperature parameter utilized in the softmax operation, influencing the exploration strategy for determining the target location of the redirected object. N_e represents the count of points within the plane subset of $\boldsymbol{\mathcal{P}}^{\text{env}}$, specifically indicating available locations for placing objects.

Let ϕ denote the network parameters of the critic Q^{high}. Given the dataset $\mathcal{D} = \{s_t, a_t, s_{t+1}\}$, the Q-function's loss, as derived from the Bellman residual, is formulated as:

$$\mathcal{L}_{\phi,i}^{\text{critic}} = \mathbb{E}_{(s_t,a_t,s_{t+1})\sim\mathcal{D}}\left[\left(y_t - Q_\phi(s_t,a_t)\right)^2\right] \tag{4}$$

where

$$y_t = r(s_t,a_t) + \gamma\mathbb{E}_{p_i\sim\mu}\left[Q_\phi\left(f_i^{\text{high}}(s_{t+1}),\pi_{\theta,i}^{\text{high}}(s_{t+1})\right)\right]$$

Reward. To evaluate the effectiveness of object redirection poses $\boldsymbol{a}_i^{\text{pose}}$ produced by higher-level agents and non-grasping adjustment strategies a_i^{type}, the following criteria should be considered: (1) Whether the adjusted grasp region point cloud aligns with the expected grasp pose or not; (2) Whether the low-level agent can use a_i^{type} to adjust the object from its initial pose to the redirection pose $\boldsymbol{a}_i^{\text{pose}}$ or not; (3) Whether the redirected object interferes with the environment or not.

Given that the success rate of non-grasping pose adjustments is highly dependent on the adjustment strategy a_i^{type}, the tabletop environment $\boldsymbol{\mathcal{P}}^{\text{env}}$, and object properties, to avoid the inefficiencies associated with frequent calls to the low-level agent during training and the infrequency of effective reward feedback, this paper proposes a data-driven continuous distribution reward representation method. This method establishes a continuous distribution function for success rates that considers environmental layout, object stiffness coefficients, manipulation strategies, and discrete point success rates.

Define the object's graspable pose with its z-axis oriented vertically. Sample redirection poses uniformly within the feasible space $\mathcal{C} = \{(x,y,\theta^z)|x\in\mathbb{R}, y\in\mathbb{R}, \theta^z\in\mathbb{R}\}$. Use the low-level agent with a specified adjustment strategy a_i^{type} to reorient the object from any initial pose to the target redirection pose $\boldsymbol{a}_i^{\text{pose}}$. Conduct multiple random experiments to compute each sampled pose's average success rate S at various object stiffness coefficients. This yields mappings of stiffness coefficients, discrete poses, and success rates for both poke and flip strategies, i.e. $M:\mathcal{C}\rightarrow[0,1]$.

Directly switching between the discrete success rate mappings for these strategies complicates the training of high-level agents. To address this, we propose a continuous success rate distribution function that integrates the mappings from both strategies by transitioning through object stiffness coefficients. The function is defined as follows:

$$S = \begin{cases} S^{\text{flp}}, & k^s < k_l^s \\ \kappa\cdot\left(S^{\text{flp}} + f\cdot\left(S^{\text{pok}} - S^{\text{flp}}\right)\right), & k_l^s \le k^s < k_h^s \\ S^{\text{pok}}, & k^s \ge k_h^s \end{cases} \tag{5}$$

with

$$\kappa = \kappa_{\text{base}} + \kappa_{\text{adj}} \cdot \min\left(\left\| k^s - \frac{k_l^s + k_h^s}{2} \right\|, \kappa_{\max}\right),$$

$$f(k^s, k_l^s, k_h^s) = \left[1 + \exp\left(-\sigma_{\text{mod}}\left(\frac{k^s - k_l^s}{k_h^s - k_l^s} - \sigma_{\text{norm}}\right)\right)\right]^{-1},$$

where k^s is the stiffness coefficient, with k_l^s and k_h^s as transition thresholds. κ adjusts the extrema of the reward function, using weighted polarization to bias the reward transition towards a single strategy. This adjustment is regulated by three parameters: κ_{base}, the baseline weight; κ_{adj}, the adjustment magnitude; and $\kappa_{\max}$, which limits the upper bound of the shift. f is a smoothing function used to transition between the success rates of two strategies: S^{flp} and S^{pok}. (σ_{mod} is the modulation coefficient, and σ_{norm} is the normalized centroid.) Thus, the reward function r is defined as :

$$r = S \cdot \exp\left(-(\phi_1 \cdot D^{\text{ori}} + \phi_2 \cdot D^{\text{pos}})\right) + R^{\text{col}} - 1 \quad (6)$$

where D^{ori} represents the orientation distance between the redirected and suggested orientations based on quaternion, while D^{pos} denotes the position distance between the planned and suggested positions. The suggested pose is selected from the maximum success rate distribution region and incorporates environmental distance weights. For flipping, the closest supporting wall to the initial position is chosen. For poking, the pose with the shortest movement distance, i.e., the nearest flipping pose, is selected. R^{col} is the penalty term, representing the penalty score when there is an intersection between the planned object and $\boldsymbol{\mathcal{P}}^{\text{env}}$.

C. Adaptive Interaction Motion Planning

The overall architecture of the proposed low-level agent is illustrated in Fig.5.

Observation and Action Spaces. The observation space for the low-level agent includes the entire scene's point cloud, comprising environmental point cloud $\boldsymbol{\mathcal{P}}^{\text{env}} = \{\boldsymbol{p}_i^{\text{env}}\}_{i=1}^{N_e}$, object point cloud $\boldsymbol{\mathcal{P}}^{\text{obj}} = \{\boldsymbol{p}_i^{\text{obj}}\}_{i=1}^{N_o}$, and the high-order agent's action decision a_i^{type} (either "poke" or "flip"). Before each action, the robot's end-effector moves to a reset position that does not obstruct the camera's view, acquires the observation point cloud, and reads the high-order agent's action decision.

The action space is centered around the object point cloud $\boldsymbol{\mathcal{P}}^{\text{obj}}$ and consists of two components: the contact position $\boldsymbol{p}_i^{\text{obj}}$ of the gripper selected from $\boldsymbol{\mathcal{P}}^{\text{obj}}$, and the motion direction $\boldsymbol{a}^{\text{dir}}$. These parameters determine the location and manner of interaction between the robot and the object. The gripper first moves to a position at a specified normal distance from the target point. Then, it executes the action using a low-gain end-effector translation controller based on the action parameters decided by the agent. After the interaction, the gripper returns to the reset position, re-observes the scene, and proceeds to the next cycle until the object reaches the target pose or exceeds the step limit.

Model Architecture. Given the observation $s = \{\boldsymbol{\mathcal{P}}^{\text{env}}, \boldsymbol{\mathcal{P}}^{\text{obj}}\}$, the low-level agent's actor network determines the interaction direction of the robot for each point p_i in the object point cloud: $\pi_i^{\text{low}}(s) = \boldsymbol{a}_i^{\text{low}} = \{\boldsymbol{a}_i^{\text{dir}} \in \mathbb{R}^3\}$. The critic network computes the Q-value for each $\boldsymbol{p}_i$: $q_i = Q^{\text{low}}(s, \boldsymbol{a}_i^{\text{low}})$, representing the estimated return for interacting at that point with the action direction generated by the actor. The interaction method for non-grasping pose adjustments of the object is determined by the point with the maximum Q-value and the output parameters of the agent.

As shown in Fig.5, features are first encoded from the scene point cloud, which includes $\boldsymbol{\mathcal{P}}^{\text{env}}$ and $\boldsymbol{\mathcal{P}}^{\text{obj}}$, using a point cloud encoder. The encoded feature vectors are then decoded by an actor's action decoder, consisting of an MLP, to produce the action direction parameters $\boldsymbol{a}_i^{\text{dir}} = [a_i^x, a_i^y, a_i^z]$, where a_i^x, a_i^y, and a_i^z represent the components of the direction unit vector.

Using a similar architecture with independent parameters, the critic's point cloud encoder extracts features from the same input point cloud. The critic feature vector for each point is combined with the action direction parameters and the high-order agent's action type decision. This combination is processed by a Q-network, which includes an MLP, to compute the Q-value for each point. The point with the highest Q-value among all object points identifies the optimal interaction location and direction, facilitating a single-step adjustment of the object's pose.

The update rule is defined as follows: The actor's loss function $\mathcal{L}_{\psi}^{\text{actor}}$ is set to the expected Q-value of all object points $\boldsymbol{p}_i$, weighted by their selection probabilities $\mu(\boldsymbol{p}_i|s)$. This formulation is similar to Eq.(3), with the distinction that:

$$\mathcal{L}_{\psi,i}^{\text{actor}} = -Q^{\text{low}}\left(s, \pi_{\psi,i}^{\text{low}}(s)\right) = -Q^{\text{low}}\left(f_i^{\text{low}}, \boldsymbol{a}_i^{\text{dir}}\right),$$

$$\mu(\boldsymbol{p}_i|s) = \mu\left(\boldsymbol{p}_i|\boldsymbol{\mathcal{P}}^{\text{obj}}\right) = \exp(\lambda q_i)/\textstyle\sum_{k=1}^{N_o} \exp(\lambda q_k),$$

where λ denotes the temperature parameter utilized in the softmax operation, which influences the selection of interaction points on the object. N_o represents the count of $\boldsymbol{\mathcal{P}}^{\text{obj}}$, specifically indicating available locations for interaction. Additionally, the Q-function's loss $\mathcal{L}_{\phi,i}^{\text{critic}}$ is derived from $f_i^{\text{low}}(s_{t+1})$ and $\pi_{\theta,i}^{\text{low}}(s_{t+1})$, following the formulation presented in Eq.(4).

Reward. The goal of the low-level agent is to adjust the object from its current state to the target pose through the robot's interaction actions. Within the point cloud observation space, the objective is to transform the object's initial point cloud into the target point cloud. A "goal flow" representation is introduced to achieve this, which calculates the 3D vector difference between the target and initial points. Consequently, the reward function for point cloud alignment is defined as:

$$r = -\textstyle\sum_{i=1}^{N_o} \|\boldsymbol{p}_i' - \boldsymbol{p}_i\| / N_o \quad (7)$$

where $\boldsymbol{p}_i$ represents the current point, while $\boldsymbol{p}_i'$ denotes the target point. The point cloud of the object contains a total of N_o points.

V. EXPERIMENTS

To validate the effectiveness of SA-DEM, a series of simulations and real-world experiments are conducted in this section. These experiments evaluate the agents' capability to make autonomous decisions and execute non-grasping

manipulations.

A. Simulation Experiments

Environmental setup. The simulation environment is developed using Robosuite [33] and MuJoCo [34], as illustrated in Fig.6(a) and Fig.6(b). It includes objects randomly initialized in a bin on the desktop, four panels randomly arranged around the workspace and an end-effector. As shown in Fig.6(c), the experimental setup features object models sourced from [35], with 20 objects selected for their varied geometries and suitability for non-grasping tasks. These objects are divided into three groups: approximate cuboid (6 objects), approximate cylinder (6 objects), and unseen Instances (8 objects). The Approximate cuboid and cylinder objects form the training set, while the unseen instances, which are not present during training, are used to evaluate the agent's generalization capability.

Task setup. The non-graspable object is placed in a random stable SE(3) pose in the given environment. The task aims to determine a valid target-grasping pose and a non-grasping manipulation strategy, using a sequence of actions to align the object with the target pose. Tasks vary in difficulty and include a dataset with a single cuboid object, one with different types of cuboids, and a complete set. Each dataset is tested with rigid, soft, and rigid-soft hybrid properties. It is important to note that the simulator does not directly model object deformation. Instead, we weaken the moment arm of the applied forces based on the deformation principles outlined in Section 3. This reduction is controlled by a randomly generated stiffness coefficient specific to soft objects, while rigid objects remain unaffected by the applied forces. Two criteria define task success. First, the target pose must meet the grasping conditions. Second, the average distance between the adjusted object and the target pose point clouds must be less than 3 centimeters. A maximum of 10 adjustment steps is allowed.

Parameter setup. SA-DEM is built on Stable-Baselines3 [36], with TD3 and hybrid TD3 modifications. The high-level and low-level agents use PointNet++ segmentation backbones [37] for point cloud feature extraction. Each agent's actor and critic networks have separate point cloud extraction modules with no shared weights. Temporal features for MI-EF are extracted using an LSTM [38]. The network architecture and hyperparameters are detailed in Table 1.

Ablation and Baseline Experiments. To validate the effectiveness of SA-DEM's core modules and reward settings in the high-level agent, we designed the following algorithm variants for ablation experiments:

1) **Without stiffness awareness**: The MI-EF module excludes the stiffness observation-related encoding and LSTM components in this variant. The actor and critic networks only process the concatenated features of the point cloud from the grasp region and the plane/wall observation point clouds. This variant tests the value of stiffness awareness in decision-making when interacting with the environment.
2) **Without stiffness awareness memory**: The MI-EF module receives only point cloud observations from single-contact interactions in this variant. The encoded information from these point clouds is concatenated with the features of the grasp region and the plane/wall observation point clouds. This variant evaluates the effect of introducing temporal stiffness information and dynamic features.
3) **With discrete rewards (Without continuous rewards)**: This variant replaces the proposed continuous reward distribution with a discrete reward representation. No transition region is established between the success rates of the poke and flip strategies. Instead, the success rates are directly switched based on the target position. This variant assesses the advantage of the proposed reward representation method.

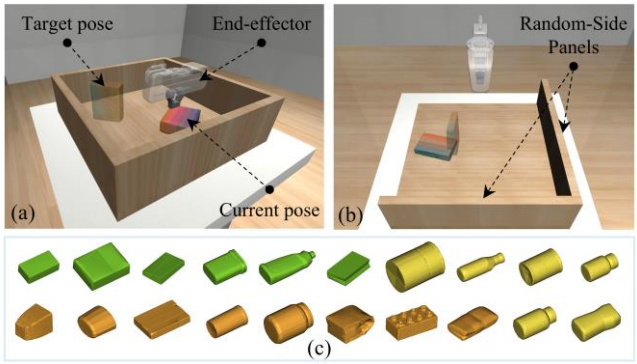

Fig. 6. Simulation environment and object set. (a) Basic layout. (b) Environmental generalization setup. (c) The green subset comprises approximately cuboidal objects, while the yellow subset consists of approximately cylindrical objects, together forming the training set. The orange subset represents unseen objects, serving as the test set.

TABLE 1: Model hyperparameters.

Hyperparameters	High-level	Low-level
Initial timesteps	5000	10000
Batch size	128	64
Actor update frequency	0.5	0.5
Critic update frequency	2	2
Temperature parameter	0.1	0.1
Discount factor	0.99	0.99
Learning rate	0.0001	0.0001
$\kappa_{\text{base}}/\kappa_{\text{adj}}/\kappa_{\max}$	0.9/0.2/0.5	
σ_{mod} / σ_{norm}	10/0.5	

The training curves for each method after 100k environment interaction steps are shown in Fig.7(d) using the complete object training set. The full SA-DEM method achieves a success rate of approximately 87%. In comparison, the "Without stiffness awareness" variant only reaches around 61%, demonstrating that relying solely on visual observations is insufficient for generating accurate action decisions. This underscores the importance of stiffness awareness for effective high-level agent interactions with the environment. The "Without stiffness awareness memory" variant achieves a

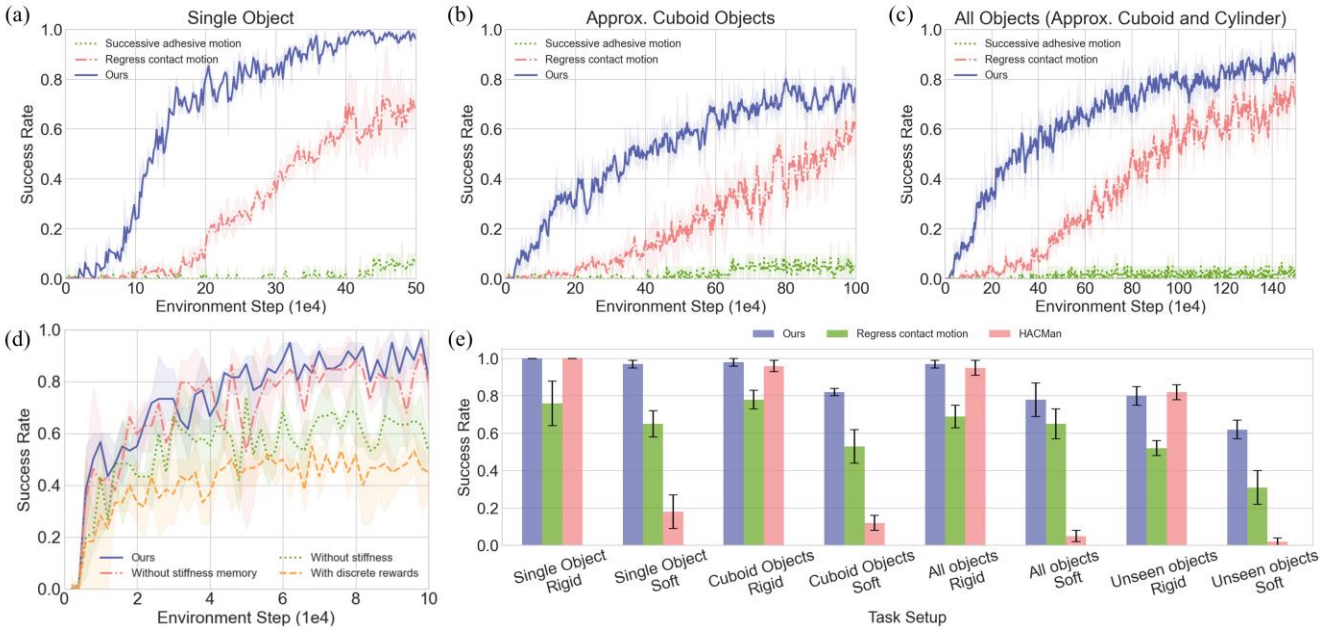

Fig. 7. Quantitative results of ablation experiments and baseline comparison. For the low-level agent: (a) Training curves for the single-object dataset; (b) Training curves for the approximate cuboid dataset; (c) Training curves for the complete training dataset. (d) Training curves for the module ablation of the high-level agent. (e) Baseline results for all tasks targeting rigid and soft manipulations.

success rate of about 81%, slightly lower than the full method, indicating that temporal stiffness perception captures additional stiffness information, aiding in training and decision-making. As for the reward function configuration, the "With discrete rewards" variant reaches only about 47% success, significantly lower than the continuous reward approach. This confirms the effectiveness of the proposed data-driven continuous distribution reward representation method, showing that continuous dense rewards are more conducive to training and faster convergence than discrete hierarchical rewards.

Next, we assess the effectiveness of the action execution framework in SA-DEM's low-level agent. The manipulation strategies for both our method and the designed baseline comparisons are described as follows:

1) **Regress per-point motion (Ours)**: A contact point is selected from the object point cloud, with each point containing motion parameters regressed by the model. Based on the position and normal of the selected point, the end-effector moves along the normal direction to the contact point. Then, using a low-gain translation controller, it executes the motion parameters in three steps, each with a 2cm stride. After each step, the end-effector returns to a predefined starting position.
2) **Successive adhesive motion**: A contact point is selected from the object point cloud, where each point similarly contains motion parameters. However, unlike our approach, in this baseline, the end-effector does not return to the initial position after each step. Instead, the previous step's endpoint becomes the next's starting point. This creates a sequence of actions where the end-effector appears to "adhere" to the object.
3) **Regress contact motion**: This baseline does not select contact points from the object point cloud. Instead, the model directly regresses 6 DoF outputs, including three for the contact position and three for the motion parameters. In each step, the end-effector moves to the regressed contact position, executes the motion parameters, and returns to the predefined initial position.

The performance of the three methods is evaluated in mixed rigid-soft object manipulation tasks across three training scenarios: single-object interaction (rectangular box) over 500k environment steps, approximate cuboid objects over 1000k steps, and the complete training dataset over 1500k steps. The training curves are shown in Fig.7(a), 7(b), and 7(c), respectively. Results show that the "Successive adhesive motion" baseline struggles to learn effective action policies for these complex multi-step tasks. Our " Regress contact motion" approach demonstrates significantly better convergence and higher sample efficiency than the " Regress per-point motion " method. SA-DEM achieves success rates of 99.5%, 80.1%, and 90.6% under the three test conditions, outperforming the "Regress contact motion" baseline by 26.3%, 16.8%, and 11.9%, respectively.

Additionally, we introduce HACman [4] into the baseline methods to compare performance against the relevant state-of-the-art approach. HACman selects contact points from the point cloud and regresses motion parameters. Unlike our method, HACman operates solely as a low-level action executor, relying exclusively on visual perception without accounting for object stiffness information, and it only supports poking.

We evaluate the success rates of HACman, "Regress contact motion," and SA-DEM across non-grasping tasks on four datasets: single-object (models trained over 500k steps), approximate cuboid objects (models trained over 1000k steps),

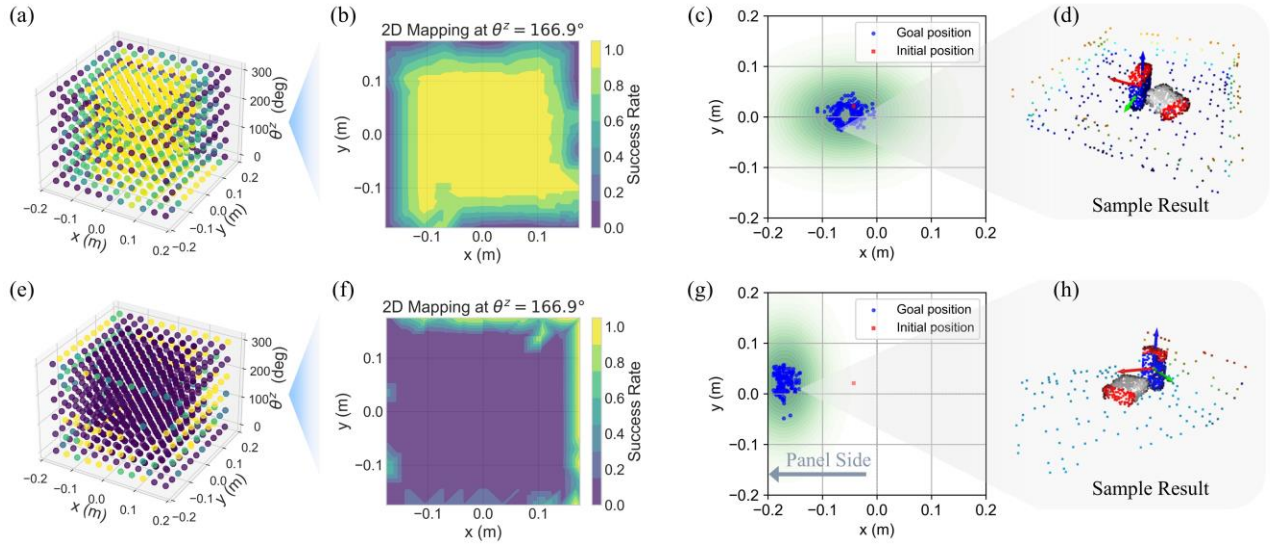

Fig. 8. Success rate distribution modeling based on the data-driven continuous distribution reward function for poke (a-d) and flip (e-h) strategies. (a)(e) show uniformly sampled success rate data; (b)(f) display 2D position plane slices at $\theta_z = 166.9^{\circ}$ from the continuous interpolated distribution; (c)(g) illustrate the decision location distribution of trained models; (d)(h) provide the decision states corresponding to the sampled points from the perspective of the point cloud.

complete training dataset (models trained over 1500k steps), and unseen object sets (models trained over 1500k steps). Each dataset is tested with three sets of rigid and soft tasks, each randomly repeated 100 times.

As illustrated in Fig.7(e), SA-DEM and HACman achieve similar success rates for rigid tasks. However, in soft tasks, our method significantly outperforms HACman. The lack of stiffness information greatly hampers the success rate in manipulating soft objects. For the "Regress contact motion" baseline, which also utilizes stiffness modality, SA-DEM consistently demonstrates superior performance across all tasks, highlighting that enhanced training efficiency improves manipulation outcomes. On the unseen dataset, our method achieves an overall success rate of 71% (approximately 80% for rigid objects and 62% for flexible objects). This result confirms that SA-DEM can effectively transfer to new manipulation objects under conditions of limited training data, showcasing good generalization capabilities. Furthermore, it outperforms HACman by a factor of 1.7 in overall success rate (approximately 82% for rigid objects and 2% for soft objects). These findings further validate the flexible decision-making and manipulation capabilities of SA-DEM, particularly emphasizing its robustness in tasks involving reduced applied torque, especially in rigid-soft coupling scenarios.

Validation of high-level agent success rate distribution. The following section details the implementation of the proposed method for data-driven continuous distribution rewards and analyzes the resulting success rate distribution.

During the training of the high-level agent, directly determining the success of each target pose through interactions with the environment would result in sparse rewards and low sample efficiency. A pre-constructed success rate distribution for each object's target pose is employed to mitigate this. Target reorientation poses are uniformly sampled within the feasible operation space, defined as $x \in [-0.2, 0.2]$, $y \in [-0.2, 0.2]$, $\theta^z \in [0, 2\pi]$. The sampling intervals are set as $\Delta(x, y, \theta^z) = (0.04, 0.04, \pi/3)$, striking a balance between success rate prediction accuracy and data collection efficiency.

For each sampled pose and action decision, 10 repeated interaction experiments are conducted, and the corresponding success rate is calculated. Figures 8(a) and 8(e) display the discrete success rate data for poke and flip manipulations on a rectangular box. Using interpolation, a continuous mapping between pose and success rate is generated. Figure 8(b) and 8(f) illustrate the success rate mapping for the slice corresponding to $\theta^z = 166.9^{\circ}$.

The results indicate that poke manipulations achieve higher success rates at a certain distance from the bin edge. In contrast, flip manipulations only succeed within a limited range near the edge. Furthermore, the success rate of flip manipulations is influenced by the panel's orientation and θ^z. Given the strong consistency of the data, it is possible to infer the success rate distribution for any random panel based on the success rate data for the side panel.

The high-level agent is trained using the reward function configured as described earlier. Under identical initial conditions, each action type is tested 200 times, and the target poses generated by the agent are collected. Figure 8(c) and 8(g) show the distribution of successful target poses in the 2D positional plane, with 191 data points (95.5%) for the poke strategy and 167 data points (83.5%) for the flip strategy. The results reveal that for the poke strategy, the agent predominantly plans target poses near the object's initial position, achieving object flipping with minimal movement cost. In contrast, for the flip strategy, the planned poses cluster near the side panel closest to the initial location. This indicates that the agent efficiently selects optimal interaction points with the environment, maintaining consistency in θ_z and minimizing action cost.

Evaluation of low-level agent action performance. This

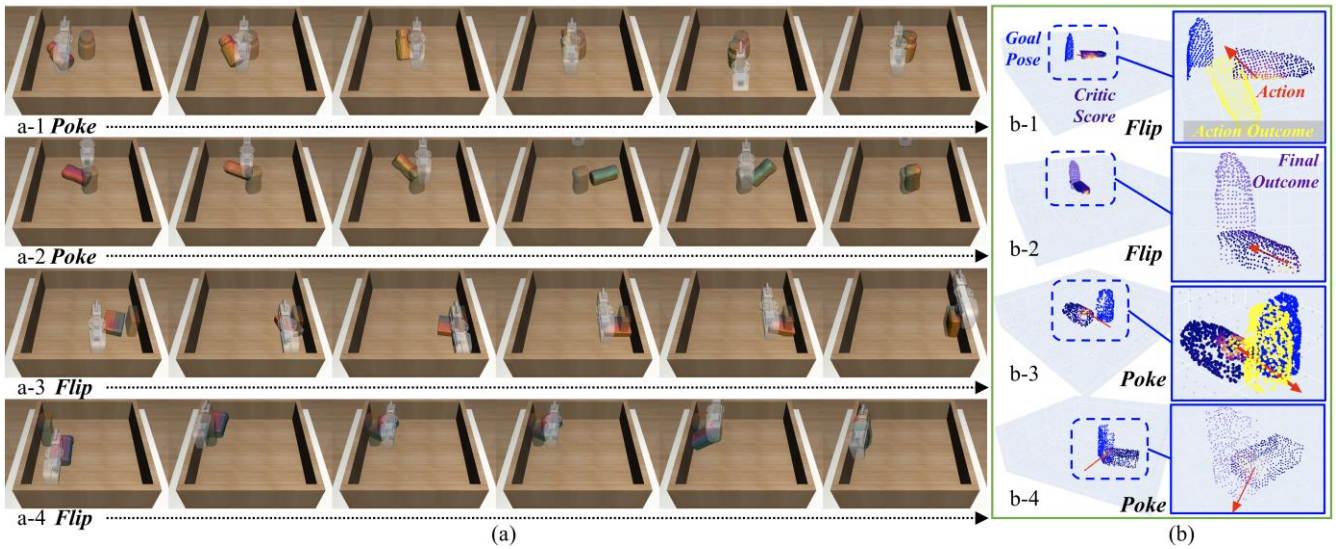

Fig. 9. Simulation frames and point cloud representations. (a) Random demonstrations of target pose alignment for non-grasping manipulation tasks, including poke and flip manipulations. (b) Point cloud representation in random demonstrations. The heatmap shows the critic score map, with lighter colors indicating higher scores. The red arrow marks the highest-scoring action. Blue represents the target point cloud, yellow the intermediate results, and purple the final outcome.

section illustrates the action details of the low-level agent, with Figure 9(a) presenting video frames of interaction processes involving four distinct objects. The interactions include poke manipulations on a medicine bottle and a container, as well as flip manipulations on a box and a canister. The demonstrated actions encompass planar pushing, downward flipping, lateral adjustments for directional correction, and lateral lifting. The low-level agent utilizes real-time feedback from environmental observations to generate closed-loop interaction points and action parameters. Autonomically combining these actions incrementally guides the object toward the target pose.

Figure 9(b) depicts the rationale for selecting interaction points on the objects and the corresponding action decisions. The critic score map, derived from the object's point cloud, is visualized using color mapping: higher scores (indicated by more prominent colors) signify greater probabilities of selecting those contact points. The optimal contact point and its associated action parameters are subsequently employed for decision-making. In the case of the flip manipulations, the critic score map in Fig. 9(b-1) highlights the lower side of the bottle, with the action directed toward the side panel to facilitate lateral interaction and adjust the bottle's orientation on the plane. Following this adjustment, Figure 9(b-2) highlights the top of the bottle, indicating an upward action to lift it. Figures 9(b-3) and 9(b-4) showcase two instances of downward poke manipulations at the object's edge. The critic score map effectively reflects the agent's predicted interaction points, emphasizing shared features across objects that correspond to successful actions. This demonstrates the agent's ability to accurately predict and execute object manipulation actions to achieve the desired target pose.

B. Experiments in real-world scenarios

Platform setup: The experimental platform setup is shown in Fig.10(a). The UR5 robot is equipped with a GL gripper [39]. Three stationary Azure Kinect DK cameras capture and form the scene point cloud. The non-grasping manipulations are conducted within a fixed bin.

Task setup: A set of objects with varying shapes, stiffness, surface friction, and density is selected as the manipulation targets, as shown in Fig.10(b). These objects include rigid items such as boxes and a bottle, as well as softer objects like a sponge, a doll, a handbag, and a packaging bag, which are commonly encountered in daily life. The task objective is to autonomously plan the target grasping pose for each object and perform the alignment manipulations toward the target pose. We relax the constraint on the 6D point cloud distance, considering the alignment with the vertical direction of the target pose as a success. Our pose adjustment aims to achieve a grasp-friendly pose rather than a strict pose alignment.

Unlike the robot in the simulation environment, which moves from the normal of the point cloud to the contact point, the actions of the real robot encompass two distinct modes: for contact points on the object's top surface, it moves from the starting point to the target. In contrast, for contact points on the side surface, the robot first shifts to an initial position offset by a specified distance in the direction of relative motion. For the point clouds collected by the three cameras, we first perform coordinate alignment using the transformation matrices obtained from the calibration of the three camera coordinate systems. Then, we segment the point cloud within the bin container region and refine the point cloud shape matching using Iterative Closest Point (ICP) [40] for local refinement. Therefore, the alignment in the experiment is based solely on the shape of the point cloud without considering the matching of object image features.

Experiment result: The experimental results are presented in Table 2. Since the initial state corresponds to a non-graspable pose, and the desired grasping pose is vertical, the task involves

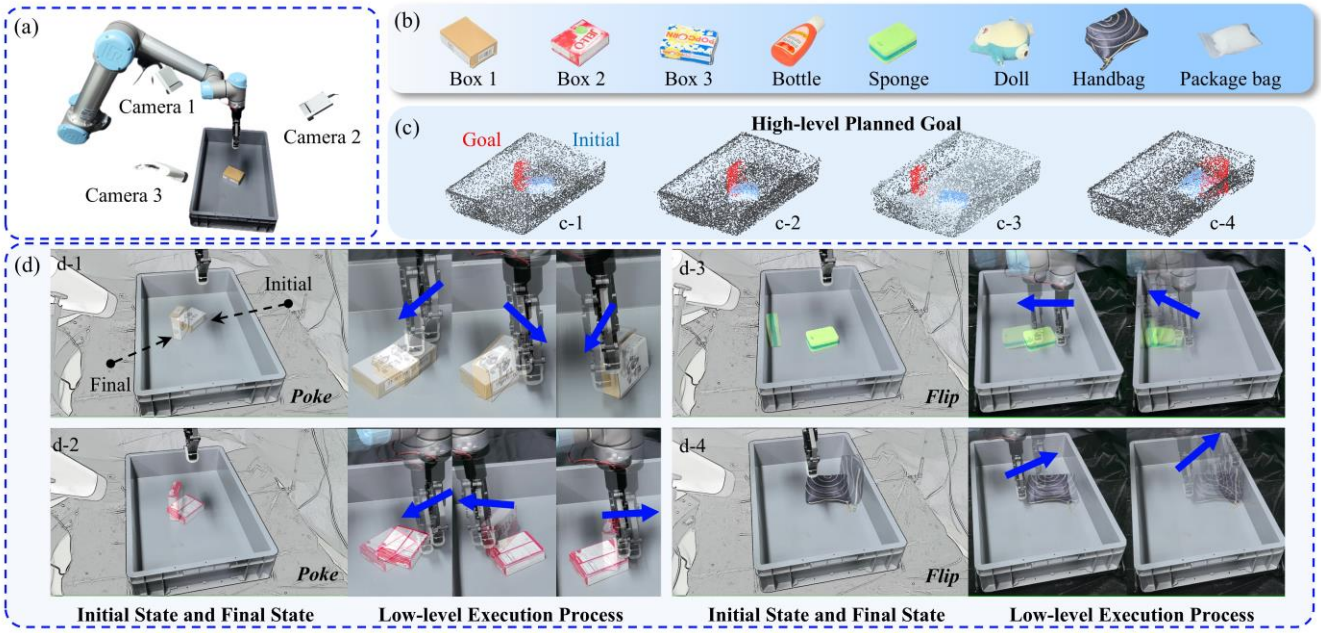

Fig. 10. Experimental setup in real-world scenarios. (a) The experimental platform, consisting of three cameras, one robot, and an environmental bin. (b) Eight objects to be manipulated, with varying properties, all being unseen objects. (c) Demonstrations of the high-level agent's planned target pose. The red point cloud represents the planned outcome, while the blue point cloud denotes the initial state. (d) The manipulation process of the low-level agent associated with c-1 to c-4.

TABLE 2: Experiment results in real-world scenarios.

Object	Stiffness	Decision	Count	Success rate	Total
Box 1	Rigid	Poke	8/10	80%	52.5%
Box 2	Rigid	Poke	5/10	50%	
Box 3	Rigid	Poke	6/10	60%	
Bottle	Rigid	Poke	2/10	20%	
Sponge	Soft	Flip	7/10	70%	42.5%
Doll	Soft	Flip	2/10	20%	
Handbag	Soft	Flip	5/10	50%	
Packaging bag	Soft	Flip	3/10	30%	

6D pose adjustment in a non-planar configuration. This task presents significant challenges, as each object has an unseen shape and physical properties. Even a small error in the interaction can lead to substantial changes in the object's pose. Examples of successful manipulations for rigid and soft objects are shown in Fig.10(c) and 10(d). The average success rate for the four relatively rigid objects is 52.5%. Among these, the success rates of the poke strategy for the three box-shaped objects are above 50%, while the success rate for the bottle is the lowest due to the instability of its contour and mass center, resulting in a high pose randomness after manipulation. For the four soft objects, all successful tests result from the flip strategy, with an average success rate of 42.5%. The success rate for the sponge and handbag exceeds 50%. The doll has a rounder contour, while the packaging bag has a high center of mass and a narrow base, making these two objects more challenging. Further comparison tests are conducted using the poke strategy, and the results showed that poke manipulations failed for all soft objects. This indicates the limitations of the single poke strategy for non-grasping manipulations, highlighting the need to consider the stiffness properties of the objects and utilize environmental interactions. Therefore, the results validate the rationality and effectiveness of the proposed agent, which integrates decision-making, poke strategy, and flip strategy.

VI. Conclusion

This study seeks to tackle a critical yet underexplored challenge in robotic extrinsic dexterity: the incorporation of object stiffness into the design of non-grasping manipulation strategies. We propose a novel dual-stage reinforcement learning framework, SA-DEM, which decomposes this task into decision-making and motion execution. By constructing a unified 3D observation space and considering the torque reduction effects during contact with objects of varying stiffness, the agent first autonomously learns to make decisions about interaction types. It then generates appropriate responses spontaneously, without relying on manually designed motion primitives or specific assumptions regarding contact modes. Experimental results in both simulation and real-world scenarios demonstrate our method's capability to perform poking actions on rigid objects using planar surfaces and flip actions on soft objects using environmental walls. It also exhibits strong generalization to unseen objects and effective transfer performance from simulation to reality. In simulated mixed rigid-soft tasks, SA-DEM achieves a success rate of 80% for rigid objects and 62% for soft objects. The overall success rate significantly surpasses the state-of-the-art method's,

yielding approximately a 1.7-fold performance improvement. In real-world scenarios, SA-DEM achieves an overall success rate of nearly 50% on unseen objects with varying stiffness, mass, and geometric properties, including 6D pose adjustments for particularly challenging shapes.

The proposed approach offers a novel perspective on enhancing robotic manipulation capabilities, showcasing the substantial potential of non-grasping manipulation for complex tasks. However, this study also has certain limitations. For instance, the manipulation success rate for direct transfer to real-world scenarios still has room for improvement. Failures are often attributable to errors in point cloud observations, such as inaccuracies inherent to depth cameras and registration errors, which hinder the precision of contact point selection. Additionally, some target poses for test objects are challenging (e.g., circular or narrow bases), making stability difficult even for human manipulation. Future work will enhance the framework's adaptability to a more diverse range of objects and responsiveness to increasingly complex environments.

ACKNOWLEDGMENT

This work was supported in part by the China Postdoctoral Science Foundation under Grant 2024M762814; in part by the National Natural Science Foundation of China through the Youth Program under Grant 509109-N72401.

REFERENCES

[1] A. Billard and D. Kragic, "Trends and challenges in robot manipulation," *Science*, vol. 364, no. 6446, p. eaat8414, 2019.

[2] N. C. Dafle *et al.*, "Extrinsic dexterity: In-hand manipulation with external forces," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2014, pp. 1578–1585.

[3] P. Sodhi, M. Kaess, M. Mukadam, and S. Anderson, "Learning Tactile Models for Factor Graph-based Estimation," Mar. 28, 2021, *arXiv*: arXiv:2012.03768. Accessed: Jul. 30, 2024. [Online]. Available: http://arxiv.org/abs/2012.03768

[4] W. Zhou, B. Jiang, F. Yang, C. Paxton, and D. Held, "HACMan: Learning hybrid actor-critic maps for 6D non-prehensile manipulation," *arXiv preprint arXiv:2305.03942*, 2023.

[5] W. Zhou and D. Held, "Learning to grasp the ungraspable with emergent extrinsic dexterity," in *Conference on Robot Learning*, PMLR, 2023, pp. 150–160.

[6] C. Eppner and O. Brock, "Visual detection of opportunities to exploit contact in grasping using contextual multi-armed bandits," in *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, IEEE, 2017, pp. 273–278.

[7] Y. Hou, Z. Jia, and M. T. Mason, "Fast planning for 3d any-pose-reorienting using pivoting," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2018, pp. 1631–1638.

[8] X. Cheng, E. Huang, Y. Hou, and M. T. Mason, "Contact mode guided sampling-based planning for quasistatic dexterous manipulation in 2d," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2021, pp. 6520–6526.

[9] X. Cheng, E. Huang, Y. Hou, and M. T. Mason, "Contact mode guided motion planning for quasidynamic dexterous manipulation in 3d," in *2022 International Conference on Robotics and Automation (ICRA)*, IEEE, 2022, pp. 2730–2736.

[10] C. Eppner, R. Deimel, J. Alvarez-Ruiz, M. Maertens, and O. Brock, "Exploitation of environmental constraints in human and robotic grasping," *The International Journal of Robotics Research*, vol. 34, no. 7, pp. 1021–1038, 2015.

[11] J. Bimbo *et al.*, "Exploiting robot hand compliance and environmental constraints for edge grasps," *Frontiers in Robotics and AI*, vol. 6, p. 135, 2019.

[12] N. Chavan-Dafle, R. Holladay, and A. Rodriguez, "Planar in-hand manipulation via motion cones," *The International Journal of Robotics Research*, vol. 39, no. 2–3, pp. 163–182, 2020.

[13] M. A. Lee *et al.*, "Making sense of vision and touch: Learning multimodal representations for contact-rich tasks," *IEEE Transactions on Robotics*, vol. 36, no. 3, pp. 582–596, 2020.

[14] Í. Elguea-Aguinaco, A. Serrano-Muñoz, D. Chrysostomou, I. Inziarte-Hidalgo, S. Bøgh, and N. Arana-Arexolaleiba, "A review on reinforcement learning for contact-rich robotic manipulation tasks," *Robotics and Computer-Integrated Manufacturing*, vol. 81, p. 102517, 2023.

[15] I. Akkaya *et al.*, "Solving rubik's cube with a robot hand," *arXiv preprint arXiv:1910.07113*, 2019.

[16] T. Chen, J. Xu, and P. Agrawal, "A system for general in-hand object re-orientation," in *Conference on Robot Learning*, PMLR, 2022, pp. 297–307.

[17] H. Qi *et al.*, "General in-hand object rotation with vision and touch," in *Conference on Robot Learning*, PMLR, 2023, pp. 2549–2564.

[18] C. Chi *et al.*, "Diffusion policy: Visuomotor policy learning via action diffusion," *arXiv preprint arXiv:2303.04137*, 2023.

[19] S. Kim, A. Bronars, P. Patre, and A. Rodriguez, "TEXterity–Tactile Extrinsic deXterity: Simultaneous Tactile Estimation and Control for Extrinsic Dexterity," *arXiv preprint arXiv:2403.00049*, 2024.

[20] M. Bauza, A. Bronars, and A. Rodriguez, "Tac2pose: Tactile object pose estimation from the first touch," *The International Journal of Robotics Research*, vol. 42, no. 13, pp. 1185–1209, 2023.

[21] S.-M. Yang, M. Magnusson, J. A. Stork, and T. Stoyanov, "Learning Extrinsic Dexterity with Parameterized Manipulation Primitives," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2024, pp. 5404–5410.

[22] A. Ten Pas, M. Gualtieri, K. Saenko, and R. Platt, "Grasp pose detection in point clouds," *The International Journal of Robotics Research*, vol. 36, no. 13–14, pp. 1455–1473, 2017.

[23] Y. Cho, J. Han, Y. Cho, and B. Kim, "CORN: Contact-based Object Representation for Nonprehensile Manipulation of General Unseen Objects," *arXiv preprint arXiv:2403.10760*, 2024.

[24] M. Kim, J. Han, J. Kim, and B. Kim, "Pre-and post-contact policy decomposition for non-prehensile manipulation with zero-shot sim-to-real transfer," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2023, pp. 10644–10651.

[25] C. Ma *et al.*, "DexDiff: Towards Extrinsic Dexterity Manipulation of Ungraspable Objects in Unrestricted Environments," *arXiv preprint arXiv:2409.05493*, 2024.

[26] H. Zhang *et al.*, "Reinforcement learning based pushing and grasping objects from ungraspable poses," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2023, pp. 3860–3866.

[27] M. Posa, C. Cantu, and R. Tedrake, "A direct method for trajectory optimization of rigid bodies through contact," *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 69–81, 2014.

[28] E. Huang, X. Cheng, and M. T. Mason, "Efficient contact mode enumeration in 3d," in *Algorithmic Foundations of Robotics XIV: Proceedings of the Fourteenth Workshop on the Algorithmic Foundations of Robotics 14*, Springer, 2021, pp. 485–501.

[29] X. Cheng, S. Patil, Z. Temel, O. Kroemer, and M. T. Mason, "Enhancing Dexterity in Robotic Manipulation via Hierarchical Contact Exploration," *IEEE Robot. Autom. Lett.*, vol. 9, no. 1, pp. 390–397, Jan. 2024, doi: 10.1109/LRA.2023.3333699.

[30] J. Wu *et al.*, "Spatial action maps for mobile manipulation," *arXiv preprint arXiv:2004.09141*, 2020.

[31] J. Liang, X. Cheng, and O. Kroemer, "Learning Preconditions of Hybrid Force-Velocity Controllers for Contact-Rich Manipulation," Oct. 28, 2022, *arXiv*: arXiv:2206.12728. Accessed: Sep. 20, 2024. [Online]. Available: http://arxiv.org/abs/2206.12728

[32] A. X. Lee *et al.*, "Beyond pick-and-place: Tackling robotic stacking of diverse shapes," in *5th Annual Conference on Robot Learning*, 2021.

[33] Y. Zhu *et al.*, "robosuite: A modular simulation framework and benchmark for robot learning," *arXiv preprint arXiv:2009.12293*, 2020.

[34] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ international conference on intelligent robots and systems*, IEEE, 2012, pp. 5026–5033.

[35] W. Liu, T. Hermans, S. Chernova, and C. Paxton, "Structdiffusion: Object-centric diffusion for semantic rearrangement of novel objects," in *Workshop on Language and Robotics at CoRL 2022*, 2022.

[36] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021.
[37] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.
[38] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: LSTM cells and network architectures," *Neural computation*, vol. 31, no. 7, pp. 1235–1270, 2019.
[39] J. Li, K. Zhu, G. Lu, I.-M. Chen, and H. Dong, "Construction of a Multiple-DOF Underactuated Gripper with Force-Sensing via Deep Learning," *Robotics: Science and Systems*, 2024, [Online]. Available: https://api.semanticscholar.org/CorpusID:272560240
[40] J. Zhang, Y. Yao, and B. Deng, "Fast and robust iterative closest point," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 7, pp. 3450–3466, 2021.

Yanzhe Wang received the B.S. degree in vehicle engineering from Jilin University, Changchun, China, in 2017 and obtained Ph.D. degree at the State Key Laboratory of Fluid Power and Mechatronic Systems, School of Mechanical Engineering, Zhejiang University, Hangzhou, China 2023. He is currently a post-doctoral fellow at Grasp Lab, Institute of Aerospace Manufacturing Engineering, School of Mechanical Engineering, Zhejiang University. His current research interests include robotic motion planning, multimodal perception and manipulation .

Wei Yu received the B.E. degree in robotics engineering from Zhejiang University, Hangzhou, China, in 2024. He is currently working toward the M.S. degree in mechanical engineering with the Zhejiang University, Hangzhou, China. His research interests include robot visual perception for grasping and manipulation, and equivariant learning.

Hao Wu received B.E. degree in Mechanical Engineering from Zhejiang University in 2024. He is currently working with the Grasp Lab, Zhejiang University, focusing on mechanical design, soft robotics, and multimodal tactile perception.

Haotian Guo received the B.S. degree in Mechatronics from Katholieke Universiteit Leuven, Leuven, Belgium, in 2020, and the research-based M.S. degree in Biomedical Engineering from National Unviersity of Singapore, Singapore, in 2023. Since then, he has been with the Grasp Lab, Zhejiang Univeristy, China, focusing on mechanical design, machine intelligence and multi-modal perception.

Huixu Dong (S'17–M'18) received the B.Sc degree in mechatronics engineering from Harbin Institute of Technology in China, in 2013 and obtained Ph.D. degree at Robotics Research Centre of Nanyang Technological University, Singapore 2018. He was a post-doctoral fellow in Robotics Institute of Carnegie Mellon University and National University of Singapore. From 2022, he is a New Hundred-Talent Program faculty, directing Grasp Lab at Zhejiang University, China. He is an associate editor of IEEE Robotics and Automation Letter (IEEE RA-L), IEEE Transactions on Automation Science and Engineering (IEEE T-ASE), ICRA2023/2024/2025, IROS2022/2023/2024/2025 and AIM2022/2023/2024. His current research interests include robotic perception and manipulation in unstructured environments, robotic gripper/hand.