Contact Coverage-Guided Exploration for General-Purpose Dexterous Manipulation

Zixuan Liu[1,2*], Ruoyi Qiao[1,2*], Chenrui Tie[1,2], Xuanwei Liu[1], Yunfan Lou[1],
Chongkai Gao[1,2], Zhixuan Xu[1,2], Lin Shao[1,2†]
[1] School of Computing, National University of Singapore [2] RoboScience
[*] Equal contribution; listed in random order [†] Corresponding author

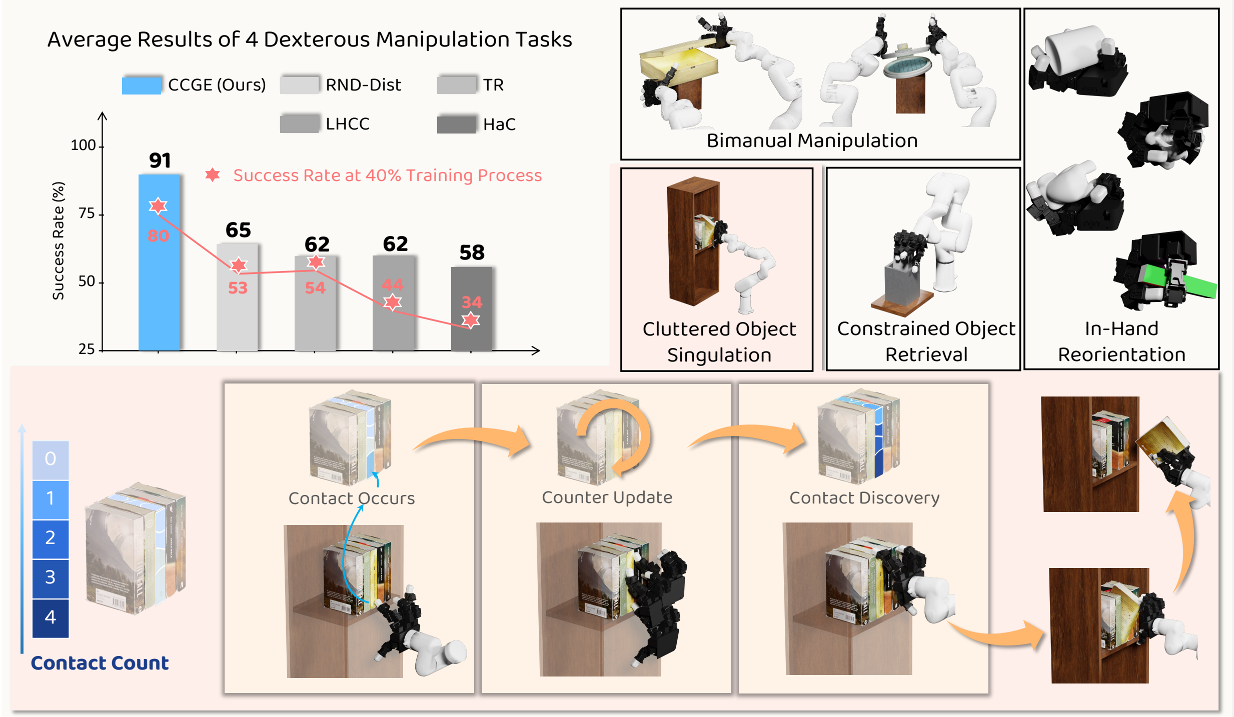


Fig. 1: CCGE is a general exploration method, which utilizes contact coverage over the target object to guide dexterous hands towards under-explored object regions. CCGE achieves strong performance with training efficiency in diverse manipulation tasks.

***Abstract*—Deep Reinforcement learning (DRL) has achieved remarkable success in domains with well-defined reward structures, such as Atari games and locomotion. In contrast, dexterous manipulation lacks general-purpose reward formulations and typically depends on task-specific, handcrafted priors to guide hand-object interactions. We propose Contact Coverage-Guided Exploration (CCGE), a general exploration method designed for general-purpose dexterous manipulation tasks. CCGE represents contact state as the intersection between object surface points and predefined hand keypoints, encouraging dexterous hands to discover diverse and novel contact patterns, namely which fingers contact which object regions. It maintains a contact counter conditioned on discretized object states obtained via learned hash codes, capturing how frequently each finger interacts with different object regions. This counter is leveraged in two complementary ways: (1) to assign a count-based contact coverage reward that promotes exploration of novel contact patterns, and (2) an energy-based reaching reward that guides the agent toward under-explored contact regions. We evaluate CCGE on a diverse set of dexterous manipulation tasks, including cluttered object singulation, constrained object retrieval, in-hand reorientation, and bimanual manipulation. Experimental results show that CCGE substantially improves training efficiency and success rates over existing exploration methods, and that the contact patterns learned with CCGE transfer robustly to real-world robotic systems. Project page is https://contact-coverage-guided-exploration.github.io.**

I. Introduction

Deep Reinforcement Learning (DRL) has proven effective for complex robotic control by autonomously discovering control policies through large-scale interaction and exploration. Its success is most evident in domains where a simple and reusable learning signal is available. For example, Atari games provide the game score as a direct reward [4], and robot locomotion commonly relies on broadly applicable dense objectives such as velocity tracking [23, 37] or reference kinematics imitation [30, 17]. These default reward formulations have enabled rapid algorithmic progress, scalable training in simulation, and transfer to real-world systems.

In contrast, dexterous manipulation lacks a canonical, plug-and-play reward. Existing DRL approaches in dexterous manipulation rely heavily on task-specific reward shaping derived from task priors, which often fail to generalize across tasks. For instance, in-hand reorientation rewards progress toward a target pose, typically augmented with proximity or contact-based shaping to stabilize finger–object interaction [1, 52, 33]. Cluttered object singulation frequently uses stage-structured rewards to encourage approach, separation, and lifting [2, 49, 11]. For functional grasping, tool manipulation tasks, and bimanual manipulation, rewards are frequently engineered around task-specific state variables in addition to reaching and grasping terms [35, 26, 31, 8, 9, 46, 16, 10, 24, 48]. While effective in specific settings, these methods rely on strong task- or embodiment-specific assumptions about how the hand should interact with the object. As a result, their applicability is often limited to particular tasks and does not naturally generalize across diverse dexterous manipulation scenarios.This motivates a fundamental question: *can we define a universal default reward that supports learning across a wide range of dexterous manipulation tasks?*

A general-purpose reward for dexterous manipulation must guide agents toward interaction strategies that are broadly useful across tasks. This makes **contact exploration** a fundamental prerequisite: in the absence of task-specific priors, an agent must first develop a rich repertoire of hand–object interactions. Prior works in DRL with general exploration rewards are often called *intrinsic rewards*, and they largely fall into two categories. *State novelty method* encourages visiting less-explored states [42, 43, 22, 6, 3, 45, 44], while *dynamics novelty method* rewards high prediction error in learned forward prediction models [41, 28, 29]. However, encouraging exploration towards the less-visited state space does not explicitly account for physical contact—the defining characteristic of dexterous manipulation [12, 55]. As a result, directly applying these methods often leads to task-irrelevant behaviors, such as pushing objects away or moving the hand freely in space without meaningful interaction.

Several works explicitly incorporate contact into exploration. Some reward novelty in hand–object distance [39], but this formulation does not incentivize contact itself, as moving around an object without touching it can still yield high novelty. Other works use haptics-based curiosity [34, 18], using prediction error of contact forces to promote interaction in gripper-based manipulation. However, contact forces in dexterous manipulation are often highly non-smooth, exhibiting force spikes and velocity discontinuities, which makes force prediction an unstable and unreliable exploration signal [20]. Effective exploration for dexterous manipulation therefore requires explicitly reasoning about hand–object contact events.

We propose Contact Coverage-Guided Exploration (CCGE), a contact-centric exploration framework that explicitly models and incentivizes hand–object interaction on novel contact patterns, namely which fingers contact which object regions. CCGE abstracts objects into surface regions and tracks contact coverage between fingers and object regions throughout training (Figure 1). To address the sparsity of contact, CCGE combines two complementary signals: a post-contact count-based reward that encourages novel finger–region contacts, and a pre-contact energy-based reaching reward that guides the hand toward regions likely to yield new interactions. Together, these signals ensure exploration is both contact-focused and continuously guided.

Moreover, the utility of a contact pattern depends strongly on the task phase and object configuration. A single global contact counter would induce cross-state interference, suppressing exploration in one configuration due to progress made in another. To enable state-aware exploration, CCGE conditions its contact counters on both the current and goal object states. The high-dimensional state space is discretized using learned hash codes, and an independent counter is maintained for each cluster. This design allows the agent to rediscover and reuse effective contact strategies across different task configurations without interference. We evaluate CCGE on a diverse suite of dexterous manipulation tasks, including cluttered object singulation, constrained object retrieval, in-hand reorientation, and bimanual manipulation. Across all tasks, CCGE consistently achieves higher success rates and faster convergence than existing exploration methods, and the learned policies exhibit robust contact behaviors that transfer effectively to real-world systems.

In summary, this work makes two key contributions. **First**, we introduce **CCGE**, a contact coverage-guided exploration reward that explicitly models and encourages diverse hand–object contact patterns across task regions. **Second**, through extensive quantitative and qualitative experiments both in simulation and the real world, we demonstrate that CCGE significantly improves training efficiency and final success rates across a wide range of dexterous manipulation tasks. More broadly, CCGE serves as a principled reward exploration for general-purpose dexterous manipulation, providing a general and task-agnostic exploration signal without reliance on handcrafted heuristics or task-specific priors. By guiding robots to systematically discover diverse and meaningful contact patterns, CCGE enables efficient learning of interaction strategies that underpin a wide range of manipulation tasks. Our implementation and code will be made publicly available.

II. Related Work

A. Intrinsic Rewards in DRL for Dexterous Manipulation

Intrinsic rewards for exploration aim to improve learning efficiency by encouraging novelty in the agent's experience. Such methods typically fall into two categories: state novelty and dynamics novelty. State novelty approaches assign higher intrinsic rewards to rarely visited states [42, 43, 22, 6, 3, 45, 44]. In dexterous manipulation, the effectiveness of state novelty depends critically on how contact is represented. A natural choice is to use contact forces as input; however, force spikes and discontinuities [20] make force-based representations noisy and unstable, often leading to erratic exploratory behavior. Alternatively, novelty can be defined using hand–object distance [39], but this formulation does not directly incentivize physical interaction, as high novelty can be achieved without making contact.

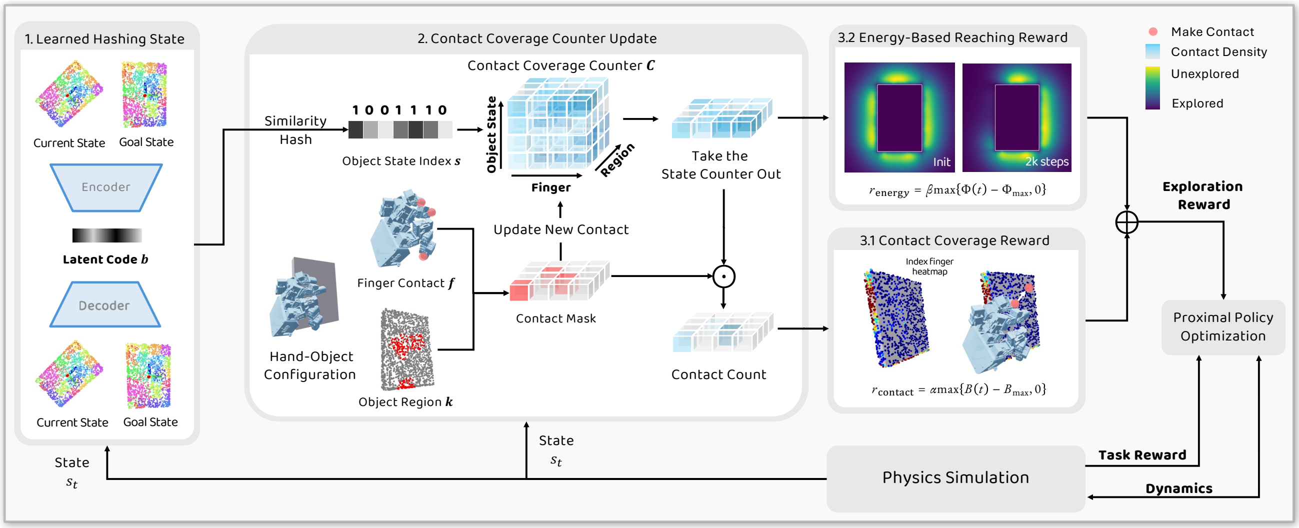


Fig. 2: **Overview of CCGE.** CCGE proposes a contact coverage-guided exploration method that explicitly models hand–object interactions, consisting of three key components: a learned state hashing module that discretizes continuous object states into compact state clusters, a contact coverage counter that records state-conditioned finger–region interactions, and a structured exploration reward. The exploration reward is further decomposed into a contact coverage reward, which encourages exploration of under-explored contact regions after contact occurs, and a pre-contact energy-based reaching reward, which guides the policy toward unexplored object regions, facilitating efficient contact discovery before physical interaction occurs. The current object state and the goal state are visualized as colored point clouds, with colors indicating different object surface regions.

In contrast, we define contact state as the intersection between object surface points and predefined hand keypoints. Compared to prior formulations based on force or distance, this representation explicitly reflects physical contact for better exploration. Compared to force- or distance-based formulations, our representation explicitly captures physical contact events and provides a more reliable, interaction-centric signal for exploration in dexterous manipulation.

Another line of work focuses on dynamics novelty, which encourages exploration by rewarding transitions that are difficult to predict under a learned dynamics model [41, 28, 29]. In robotic manipulation, HaC [34] predicts continuous contact forces and uses prediction error as an intrinsic reward to promote contact-rich behavior in parallel grippers. However, in dexterous manipulation settings, contact forces are often highly non-smooth, exhibiting force spikes and velocity discontinuities upon contact [20]. These characteristics make force prediction unreliable and can induce task-irrelevant or unstable interactions. To improve robustness, Huang et al. [18] proposes a simplified contact representation based on a gentleness metric that encourages low-force interactions during manipulation. While effective in simple scenarios such as gently touching an object, such coarse, hand-level metrics lack the spatial resolution to support the diverse and complex contact strategies required in general dexterous manipulation.

B. Task Priors in RL for Dexterous Manipulation

Beyond exploration strategies, prior work often introduces explicit task priors to improve training efficiency in dexterous manipulation. These approaches can be broadly categorized into reaching guidance and prior knowledge injection. Reaching guidance typically uses dense shaping rewards to encourage the hand to move toward the object [54, 49, 11]. While effective for grasping-oriented tasks, such guidance is often insufficient for more complex dexterous manipulation that requires rich and diverse hand–object interactions. Other approaches incorporate prior knowledge by explicitly encoding assumptions about how the hand should interact with the object. Examples include high-quality initialization [27, 13, 21], manually engineered robot-centric rewards [31, 8, 9, 2, 46, 24, 49], grasp generation modules [48, 50], learning from human videos [16], and expert demonstrations [35, 10, 26, 19]. Although effective in specific settings, these methods rely on strong task- or embodiment-specific priors, which limits their generalization across tasks and provides limited support for structured exploration that can autonomously exploit contact for dexterous manipulation.

III. The Contact Coverage Counter Design

A. Problem Formulation

We formulate dexterous manipulation as a Markov Decision Process (MDP), defined by the tuple $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{T}, \rho_0, \gamma\}$, where the state space $\mathcal{S}$ includes robot proprioception and object state information, and the action space $\mathcal{A}$ is the continuous low-level control commands for the dexterous hand. The reward function $\mathcal{R}$ consists of a task-specific reward that encourages task completion and an exploration reward designed by this work to promote effective discovery of contact strategies. We use Proximal Policy Optimization (PPO) [38] to solve this MDP for all tasks.

We introduce the contact coverage counter through three steps. We first describe how the object and the dexterous hand are represented by surface regions and robot hand fingers respectively to characterize contact interactions. Based on the object surface regions and hand fingers, we then formally define the contact coverage counter, which records contact occurrences at the level of finger–region pairs conditioned on object states, and describe how the contact coverage counter is updated. And finally, we describe how we divide the task space into different clusters to avoid cross-state interference.

B. Object and Hand Representations

To represent contact interactions in a hand-object manipulation, we first construct a discrete representation of the object surface regions and the hand fingers. Specifically, we uniformly sample M points $\{(\mathbf{p}_m, \mathbf{n}_m)\}_{m=1}^{M}$ from the object surface, where each point is associated with its position $\mathbf{p}_m$ and surface normal $\mathbf{n}_m$. These points are then clustered into K surface regions based on their spatial locations and normal directions, denoted as $k = \xi(m)$.

For the hand, we represent the hand as F fingers, where each finger is abstract using a set of predefined surface keypoints and corresponding normal $\{(\mathbf{p}_l, \mathbf{n}_l)\}$ on its finger links. As illustrated in Figure 3, these keypoints are attached to each finger link and located on their palmar surfaces. We select these specific hand keypoints because it has been extensively demonstrated in prior literature [47] that a sparse set of fingertip keypoints is sufficient to represent contact information across a wide range of tasks.

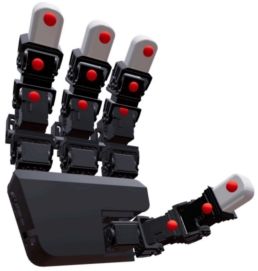


Fig. 3: **Hand Keypoint Representation.** We represent the dexterous hand fingers using a sparse set of keypoints (visualized as red spheres).

C. Object State Cluster via Learned Hashing

In dexterous manipulation, the same contact pattern may be expected to be reused across various spatial locations and time steps. If we only maintain a global counter, the previously explored patterns may have already been counted. This prevents the agent from seeking the same pattern under different spatiotemporal configurations. Thus, in this work, we use different counters in different object *state*, and want to use some clustering algorithm to discover different object states automatically during the training process.

To characterize the task-relevant configuration of the object, we define the object state $\mathbf{s}^{\text{obj}} = [\mathbf{s}^{\text{cur}}, \mathbf{s}^{\text{goal}}]$ as the point-cloud-based representation consisting of the object's current and goal configurations. Specifically, we represent the object using a fixed set of M synthetic points pre-sampled on a canonical object surface. The current state $\mathbf{s}^{\text{cur}}$ and goal state $\mathbf{s}^{\text{goal}}$ are then formed by transforming this point cloud according to the object's current and target poses, respectively. Then, we want to cluster the continuous and high-dimensional object states of the whole task space into different clusters $\{s_i\}_{i=1}^{S}$, where S is the total number of clusters, and assign an independent contact counter to each cluster. During training, each detected contact event increments only the counter corresponding to its specific state cluster. This spatiotemporal decoupling of counters ensures their mutual independence, effectively alleviating the exploration cross-state interference problem.

To this end, we follow Tang et al. [45] and employ an autoencoder structure to compress and discretize the state space into a finite set of clusters. The encoder maps $\mathbf{s}^{\text{obj}}$ to a D-dimensional latent code $\mathbf{b} \in (0,1)^D$, which is regularized toward binary values during training. The autoencoder is trained using the following equation:

$$\mathcal{L} = \left\| f(\mathbf{b}) - \mathbf{s}^{\text{obj}} \right\|_2^2 + \frac{\lambda}{D} \sum_{i=1}^{D} \min\left\{(1-b_i)^2,\ b_i^2\right\}, \tag{1}$$

where $f(\cdot)$ denotes the decoder, b_i is the i-th latent dimension of the latent code, and λ is a hyperparameter. This first term is a standard autoencoder reconstruction loss, and the second term drives each dimension away from 0.5 to be closer to either 0 or 1. We then binarize $\mathbf{b}$ via thresholding at 0.5 and project it to a compact H-bit hash using SimHash [7] with a fixed random projection matrix. The resulting hash is a discrete state index $s \in \{0, \ldots, 2^H - 1\}$, enabling tractable tracking of contact coverage across semantically similar object states while preserving computational efficiency.

The autoencoder is trained end-to-end alongside the PPO update. Whenever a contact event occurs, we first compute the object state's hash code to identify its cluster index s, then update the corresponding contact counter $\mathbf{C}_{s,f,k}$ as specified in Section III-D.

D. Contact Coverage Counter

We aim to record the interactions between fingers and the object surface regions, encouraging the robot to explore novel contact patterns defined by unique pairings of specific fingers and unexplored surface regions. To achieve this, we maintain a contact coverage counter $\mathbf{C} \in \mathbb{R}^{S \times F \times K}$. Each entry $\mathbf{C}_{s,f,k}$ records the number of contact occurrences between finger f and surface region k under state cluster s. Contact is detected at the finger level: if any keypoint on finger f contacts the object, the entire finger is considered in contact. This merged counter design empirically outperforms maintaining separate counters per keypoint.

During training, contact detection operates at the level of hand keypoints and object surface points. For each finger f, we identify the closest interacting pair:

$$(l_f, m_f) = \mathop{\arg\min}_{\substack{l \in \text{finger } f \\ m \in \text{object}}} \|\mathbf{p}_l - \mathbf{p}_m\|_2, \tag{2}$$

where l_f is the finger keypoint nearest to the object and m_f is its closest object surface point. To avoid false positives from transient geometric proximity in simulation, finger-level contact is registered only when *both* geometric proximity and sufficient physical interaction are satisfied:

$$\mathbb{I}^{\text{contact}}(f) = \mathbb{I}\left[\|\mathbf{p}_{l_f} - \mathbf{p}_{m_f}\|_2 < \delta_{\text{dist}}\right] \cdot \mathbb{I}\left[\|\mathbf{F}_{l_f}\|_2 > \delta_{\text{force}}\right], \tag{3}$$

where $\mathbf{F}_{l_f}$ is the contact force measured at keypoint l_f, and δ_{dist}, δ_{force} are small thresholds. Point m_f is mapped to its surface region $k = \xi(m)$. The procedure including Equation 2 and Equation 3, which computes $\mathbb{I}_t^{\text{contact}}(f)$, k_f, and p_{l_f}, is referred to as CONTACTMATCH in Algorithm 1. When $\mathbb{I}^{\text{contact}}(f) = 1$, the counter $\mathbf{C}_{s,f,k}$ is incremented by one. This discrete approximation avoids expensive fine-grained collision detection while remaining robust to simulation noise. Counters persist throughout RL training and are never reset.

IV. Contact Coverage-Guided Exploration

Contact events are inherently sparse during manipulation: rewarding only novel contacts provides no guidance for motion in free space, while rewarding generic state novelty encourages manipulation-irrelevant behaviors (e.g., arbitrary hand motions unconnected to the object). To address this, CCGE decomposes exploration into two complementary signals derived from the same contact coverage statistics. The *post-contact* signal provides a sparse, interaction-focused reward that rewards only contact-relevant exploration. The *pre-contact* signal provides dense, continuous guidance by shaping motion toward spatial regions likely to yield novel contacts. Together, they deliver structured exploration during the training process. An overview of our pipeline is illustrated in Figure 2.

A. Contact Coverage Reward

To ensure exploration remains focused on interaction-relevant events, we provide exploration rewards *only* upon physical contact. At timestep t, for each finger f that contacts the object ($\mathbb{I}_t^{\text{contact}}(f) = 1$), we map the contacted point to its surface region k under the current state cluster s, and assign a count-based reward:

$$R_{\text{contact}}(t) = \frac{1}{F} \sum_{f=1}^{F} \mathbb{I}_t^{\text{contact}}(f) \cdot g\big(\mathbf{C}_{s,f,k}\big), \tag{4}$$

where $g(c) = 1/\sqrt{c+1}$ is a monotonically decreasing function. This reward explicitly incentivizes novel finger-region interactions while ignoring manipulation-irrelevant state variations in free space.

B. Energy-Based Reaching Reward

While the contact coverage reward directly encourages exploration after contact occurs, relying solely on post-contact rewards can be sample-inefficient since there is no guidance before the contact is made, thus the robot can only rely on random noise to discover novel contacts. To provide guidance in a pre-contact form, we design an energy-based reaching reward that encourages the policy to move toward under-explored contact regions before physical contact is established.

For each finger f, we define a contact energy function that measures how close the finger is to object surface regions that have low contact coverage of the current object state cluster:

$$\Phi_f = \sum_m g\big(\mathbf{C}_{s,f,\xi(m)}\big)\ \exp\left(-\frac{\left\|\mathbf{p}_{l_f} - \mathbf{p}_m\right\|_2}{\delta}\right), \tag{5}$$

where δ controls the spatial decay. Intuitively, this energy term measures the distance between the selected finger keypoint l_f and all points on the object surface, with a contact-count-based weight assigned for each hand-object contact to increase the weight for contacts with fewer counts.

The overall energy-based reaching reward is obtained by averaging over all fingers:

$$R_{\text{energy}}(t) = \frac{1}{F} \sum_{f=1}^{F} \Phi_f. \tag{6}$$

C. Prevent Premature Convergence

Although the abovementioned two rewards can guide the robot to make novel contacts, in DRL, during the training process, the agent may get stuck in a previously explored path, continuing go further on that path and never trying other ways besides it. This is well-known as the detachment [14, 56] and short-sighted [6] behaviors. To mitigate this phenomenon, in this work, we revise the above two rewards to make them only reward those steps that achieve higher rewards than previous steps in an episode. Specifically, the contact reward becomes:

$$R_{\text{contact}}^{\text{scaled}}(t) = \alpha[R_{\text{contact}}(t) - R_{\text{contact}}^{\max}]_+ \tag{7}$$

where α is a scaling coefficient, $R_{\text{contact}}^{\max}$ denotes the episodic cumulative maximum value and $[x]_+ = \max(x, 0)$ denotes the positive part operator.. Similarly, we change the reaching energy-based reaching reward to:

$$R_{\text{energy}}^{\text{scaled}}(t) = \beta[R_{\text{energy}}(t) - R_{\text{energy}}^{\max}]_+ \tag{8}$$

where β is a scaling coefficient and $R_{\text{energy}}^{\max}$ denotes an episodic cumulative maximal value. This formulation rewards only forward progress toward novel contact regions, suppresses oscillatory behavior around previously explored areas.

Together, these two components enable efficient exploration across both pre-contact and post-contact phases of dexterous manipulation. The algorithm is summarized in Algorithm 1.

V. Experiments

In this section, we evaluate CCGE on a diverse set of dexterous manipulation tasks both in simulation and real-world environments to answer the following key questions:

- **Q1** Does CCGE improve task performance and sample efficiency compared to existing intrinsic motivation methods across diverse kinds of manipulation tasks?
- **Q2** Does CCGE outperform RL methods that rely on task prior knowledge?
- **Q3** How does the proposed object state cluster alleviate the exploration saturation? What object state clusters does CCGE learn during the training process?
- **Q4** How do the two individual reward components proposed in CCGE contribute to the overall performance?
- **Q5** Can the contact patterns learned by CCGE in simulation be transferred to real-world robotic manipulation?
- **Q6** Can CCGE be applied to different dexterous robot hands and different hand keypoint selection mechanisms? (See Supp.)

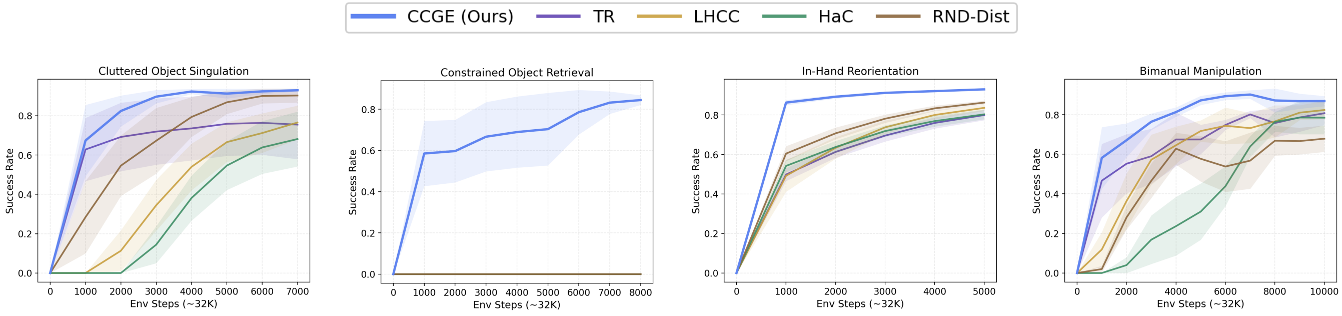


Fig. 4: **Learning Curves of 4 Dexterous Manipulation Tasks.** Our method, CCGE, leverages contact-guided exploration to achieve higher sample efficiency and success rates, particularly in "hard exploration" tasks like Constrained Object Retrieval where baselines fail.

Algorithm 1 Contact Coverage-Guided Exploration (CCGE)

```
Require: Surface regions $\{\mathbf{p}_k\}_{k=1}^{K}$, encoder $E$, decoder $f$, policy $\pi$, value $V$; $g(c) = \frac{1}{\sqrt{c+1}}$, decay $\delta$, scales $\alpha, \beta$, task reward $R_{\text{task}}$.
Initialize per-cluster counters $\mathbf{C}_s \in \mathbb{R}^{F \times K} \leftarrow 0$ for all discovered clusters $s$.
while not converged do
    for each interaction step $t$ do
        $s_t \leftarrow \text{SimHash}(\mathbb{I}[E([\mathbf{s}_t^{\text{cur}}, \mathbf{s}_t^{\text{goal}}]) > 0.5])$.
        for each finger $f = 1, \ldots, F$ do
            $(\mathbb{I}_t^{\text{contact}}(f), k_f, \mathbf{p}_{l_f}) \leftarrow \text{ContactMatch}(f)$.
            if $\mathbb{I}_t^{\text{contact}}(f) = 1$ then
                $\mathbf{C}_{s_t,f,k_f} \leftarrow \mathbf{C}_{s_t,f,k_f} + 1$.
            end if
            $\Phi_f = \sum_m g(\mathbf{C}_{s,f,k_f}) \ \exp(-\|\mathbf{p}_{l_f} - \mathbf{p}_m\|_2 / \delta)$.
        end for
        $R_{\text{contact}} \leftarrow \frac{1}{F} \sum_f \mathbb{I}_t^{\text{contact}}(f)\, g(\mathbf{C}_{s_t,f,k_f})$
        $R_{\text{energy}} \leftarrow \frac{1}{F} \sum_f \Phi_f$
        $R(t) \leftarrow R_{\text{task}}(t) + R_{\text{contact}}^{\text{scaled}}(t) + R_{\text{energy}}^{\text{scaled}}(t)$.
    end for
    Update $\pi, V$ with PPO using $R(t)$; update $f$ via Eq. (1).
end while
```

A. Experimental Setup

For **Q1**, we conduct extensive experiments on four simulated dexterous manipulation tasks designed to cover a wide range of contact-rich manipulation scenarios, including:

- **Cluttered Object Singulation**: The robot needs to extract a single book from a densely packed row on a bookshelf.
- **Constrained Object Retrieval**: The robot must retrieve a cube from a top-opening box by sliding it along the interior walls. This constraint stems from the narrow clearance between the cube and the box opening, which restricts feasible manipulation motions.
- **In-Hand Reorientation**: The robotic hand is tasked with rotating an object to a specified target orientation, with objects chosen from ContactDB [5] dataset.
- **Bimanual Manipulation**: Two robotic hands must coordinately flip open the hinged lid of a waffle iron, or open a box from ARCTIC [15] dataset.

We use a UFactory xArm robotic arm integrated with a 16-DOF LEAP Hand [40] for all experiments except for **Q6**. The task settings are visualized in Figure 1, and more task details can be found in Supp. For the PPO policy, the observation space comprises robot proprioceptive information, object states, goal states, and binary contact signals for each link of the dexterous hand. The action space consists of delta end-effector poses for the robotic arm and delta joint angles for the dexterous hand, except for In-Hand Reorientation (see Supp. for details), with all joints operating under position control. The simulation experiments are performed in Isaac Gym [25], which provides GPU-accelerated, large-scale parallel simulation for reinforcement learning. We use MLP to implement the PPO actor and critic, and the autoencoder for the state clustering.

B. Comparison with Intrinsic Exploration Baselines

To answer **Q1**, we compare the learning efficiency and final performance of CCGE against four baseline methods that incorporate intrinsic exploration rewards, evaluated across four challenging dexterous manipulation tasks mentioned above. These methods include:

- **TR (Task Reward):** Reinforcement learning with only task reward, without any exploration guidance.
- **LHCC (Learned-Hash-Codes Count):** A count-based exploration method that encourages visitation of novel states using learned hash codes [45, 53]
- **HaC (Haptics Curiosity):** An intrinsic reward based on prediction error over haptic signals [34], encouraging exploration through unexpected dynamics feedback.
- **RND-Dist (Random Network Distillation with Hand-Object Distance):** A curiosity-based exploration reward used in general RL that applies random network distillation using hand-object distance as the curiosity state [39].

All methods use identical policy architectures, observation spaces, and training budgets to ensure a fair comparison. Performance is evaluated using task success rates and learning curves as a function of environment steps.

Table I reports final success rates and sample efficiency measured by the number of environment steps required to reach a 70% success rate. If a method does not reach 70%, we use the maximum training steps for that task (e.g., 8.0 for Constrained Object Retrieval). Figure 4 presents learning curves across 4 challenging dexterous manipulation tasks.

Across all tasks, CCGE consistently outperforms all baseline methods in both learning efficiency and final task performance, **significantly improving average success rates**. Notably, in the Constrained Object Retrieval task, **CCGE is the only method that achieves successful task completion**, attaining an average success rate of **88%**, whereas all baseline methods fail to achieve any successful outcomes. This result highlights CCGE's ability to drive meaningful exploration in tasks with strong contact constraints.

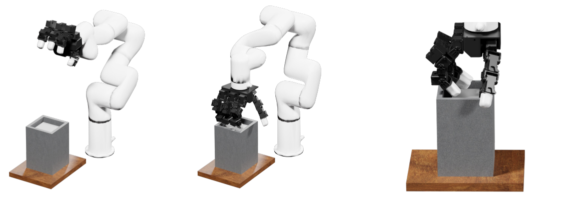
Moreover, CCGE explicitly promotes exploration of diverse and under-explored object–finger contact patterns, resulting in more structured and task-relevant exploration. As a result, **CCGE exhibits the lowest variance across random seeds among all compared methods**, as reported in Table I, indicating more stable and reliable learning behavior.

In terms of sample efficiency, Figure 4 shows that **CCGE achieves substantially faster learning speed**. In Cluttered Singulation and In-hand Reorientation, CCGE (the blue line) reaches over 80% success within approximately $3M \sim 9M$ environment steps , whereas baseline methods require significantly more interaction or plateau at lower performance. As presented in Table I, CCGE requires the fewest environment steps to reach 70% success across all tasks, reducing sample complexity by up to 2-3× compared to intrinsic-reward baselines such as LHCC, HaC, and RND-Dist.

C. Comparison with Task-Specific Prior Knowledge

To answer **Q2**, we evaluate whether CCGE reduces reliance on task-specific priors that are commonly used to facilitate exploration in dexterous manipulation. We focus on the Object Retrieval task and compare CCGE against two baselines: **Task Reward (TR)** and **TR-PrePose** (augment TR with a carefully designed pre-contact hand initialization).

Table II shows the results. Without any prior, TR completely fails to solve the task, resulting in 0% success. Introducing a carefully designed pre-contact hand initialization in TR-PrePose enables partial task completion, yielding an average success rate of 33%. In contrast, CCGE achieves a substantially higher success rate of **88%** without relying on any task-specific initialization.

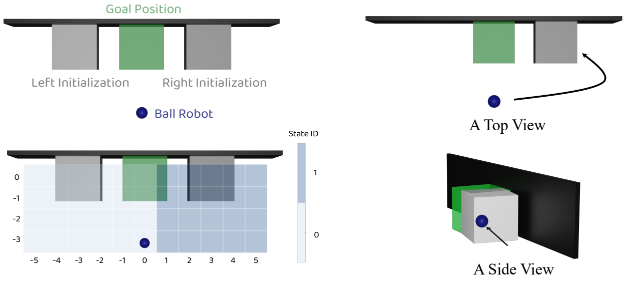


(a) Default Initialization (b) Pre-Contact Initialization (c) Pre-Contact Initialization (A Zoomed-in Side View)

Fig. 5: **Visualization of default initialization and pre-contact initialization in Constrained Object Retrieval.**

Beyond final performance, CCGE also demonstrates significantly improved learning efficiency. CCGE reaches the 30% success threshold in 2.0 × 32M environment steps, which is nearly 3× faster than TR-PrePose. These results show that CCGE not only outperforms methods that depend on strong hand-engineered task-specific priors, but also learns more efficiently and consistently. Overall, this indicates that CCGE can effectively replace manually designed priors that are typically required for constrained manipulation tasks. By autonomously discovering effective contact patterns, CCGE enables reliable learning from canonical initial states.

D. Object State Clustering Ablation



(c) Visualization of State Distribution (A Top View) (b) Push Box Trajectory

Fig. 6: **Push Box Task.** (a) The box is initialized at either end of the wall. (b) The actuated ball pushes the box to the goal (right initialization is used as an example). (c) Visualization of two learned object state clusters (state ID 0 and 1).

To answer **Q3**, we investigate whether conditioning exploration on object state is necessary for CCGE and whether it mitigates exploration saturation. To isolate this effect, we design a simple yet diagnostic task, termed **Box Push**.

In the Box Push task, a simplified point-mass agent is required to push a box—initialized at either end of a wall—toward a central goal location (Figure 6). Successful task execution requires state-dependent contact strategies: the agent must push from the left when the box is initialized on the left side, and from the right when the box is initialized on the right. This task provides a controlled setting in which to examine whether exploration mechanisms can adapt to different object configurations. We use two methods for experiments:

- **Single-State**: A global contact coverage counter that does not distinguish between different object states.
- **CCGE (Ours)**: A contact coverage counter conditioned on learned object state clusters.

As shown in Table III, the **Single-State** achieves a lower success rate and typically converges to a policy that solves the task from only one initial configuration. We attribute this behavior to cross-state counter interference. For example, under the left initialization, the task reward encourages the ball to preferentially explore pushing the box from the right surface, quickly accumulating contact counts in that region. Once the task reward is no longer obtainable, the policy becomes less inclined to explore the right surface due to the high count values. However, for the right initialization, pushing from the right is a natural and effective solution. This

TABLE I: **Quantitative Results on Four Dexterous Manipulation Tasks.**

Method	Success Rate (%) ↑					Steps (×32M) Needed to Achieve 70% Success Rate↓			
	Singulation	Retrieval	InHand	Bimanual	Avg.	Singulation	Retrieval	InHand	Bimanual
TR	77±33	0±0	79±9	92±5	62±12	2.4±2.3	8.0±0.0	3.4±1.3	2.3±2.0
LHCC	77±17	0±0	81±7	90±7	62±8	5.4±1.6	8.0±0.0	3.1±1.2	3.5±1.8
HaC	68±28	0±0	80±8	85±14	58±13	6.0±1.3	8.0±0.0	3.2±1.2	4.5±3.4
RND-Dist	91±7	0±0	78±11	89±11	65±7	3.0±1.7	8.0±0.0	2.9±1.2	4.1±3.7
CCGE (Ours)	**94±1**	**88±6**	**88±5**	**95±3**	**91**±4	**1.6±0.8**	**2.6±2.7**	**1.8±0.9**	**1.7±1.1**

TABLE II: **Comparison with Task-Specific Prior.**

Setting	Success Rate (%) ↑	Steps (×32M) at 30% SR↓
TR	0±0	8.0±0.0
TR-PrePose	33±40	5.6±3.0
CCGE (Ours)	**88±6**	**2.0**±2.0

TABLE IV: **Reward Ablation Studies**

Method	Success Rate (%) ↑
TR	77
CCGE-Contact	89
CCGE-Energy	91
CCGE (Ours)	**94**

TABLE III: **Ablation of Object-State Conditioning.**

Variant	Success Rate (%) ↑
Single-State	18 ± 11
CCGE (Ours)	**100 ± 0**

mismatch leads to suppressed exploration in a configuration where the same contact pattern is actually beneficial, resulting in cross-state counter interference.

In contrast, **CCGE** achieves high success across both initializations. By maintaining independent contact coverage counters for each learned object state cluster, CCGE enables state-specific exploration and prevents premature saturation of the exploration signal. Figure 6c visualizes the learned object state clusters, showing that the hash-based representation partitions the state space into meaningful and task-relevant configurations automatically.

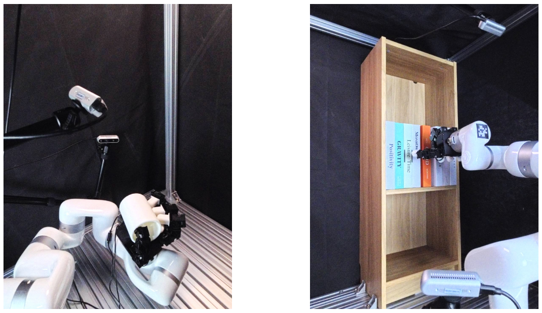
E. Ablation Studies

To answer **Q4**, we conduct ablation studies to analyze the contribution of individual components in CCGE. We focus on two aspects: (i) the design of the exploration reward, and (ii) the configuration of the learned object state representation. All ablation experiments use the same training setup as in the main results. Ablations related to the object state representation are put to Supp.

Here, we consider ablations that isolate the roles of the post-contact and pre-contact exploration signals:

- **CCGE-Energy**: removing the post-contact contact coverage reward while retaining the energy-based reaching reward R_{energy}.
- **CCGE-Contact**: removing the pre-contact energy-based reaching reward while retaining the contact coverage reward R_{contact}.

Table IV reports the results on Cluttered Object Singulation task. While both partial variants outperform the task-reward-only baseline, neither matches the success rate of the complete method. The full CCGE achieves consistently strong results across two tasks, indicating that both exploration components are essential to the overall performance.



(a) In-Hand Reorientation (b) Cluttered Object Singulation

Fig. 7: **Real-World Experiment Setup.**

F. Real-World Experiments

To answer **Q5**, we validate sim-to-real transfer on a platform comprising a uFactory xArm and a 16-DoF LEAP Hand [40], with two RealSense D435 cameras for sensing (Fig. 7). We focus on two representative real-world dexterous manipulation tasks: In-Hand Manipulation, which evaluates the policy's ability to perform precise, contact-rich object reconfiguration within the hand, and Cluttered Singulation, which tests robust exploration and interaction under multi-object interference. These tasks are selected to stress complementary aspects of CCGE, including fine-grained contact control and structured exploration in complex environments. We distill the privileged-state teacher into a student policy that operates on raw point clouds and proprioception. Detailed metrics and qualitative results are provided in the Supplementary.

VI. Limitations and Conclusion

Despite its effectiveness, CCGE has several limitations that point to promising directions for future work. One important direction is to incorporate additional sensing modalities, such as force–torque or tactile sensing, to further enrich the exploration signal. In addition, while CCGE demonstrates promising transfer to real-world systems, most evaluations in this work are conducted in simulation. Accelerating reinforcement learning directly on real-world robotic systems remains an important avenue for future investigation. In this work, we introduce **Contact Coverage-Guided Exploration (CCGE)**, a general exploration reward that explicitly incentivizes structured hand–object interactions for dexterous manipulation. By modeling contact coverage between fingers and object regions and combining sparse post-contact rewards with dense pre-contact guidance, CCGE enables efficient, task-agnostic exploration across diverse manipulation tasks without relying on handcrafted shaping. Extensive experimental results demonstrate that CCGE consistently improves learning efficiency and robustness, positioning it as a principled default reward for general-purpose dexterous manipulation.

References

[1] OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39 (1):3–20, 2020. 2

[2] Fengshuo Bai, Yu Li, Jie Chu, Tawei Chou, Runchuan Zhu, Ying Wen, Yaodong Yang, and Yuanpei Chen. Retrieval dexterity: Efficient object retrieval in clutters with dexterous hand. *arXiv preprint arXiv:2502.18423*, 2025. 2, 3

[3] Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. 2

[4] Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of artificial intelligence research*, 47:253–279, 2013. 1

[5] Samarth Brahmbhatt, Cusuh Ham, Charles C. Kemp, and James Hays. ContactDB: Analyzing and predicting grasp contact via thermal imaging. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6 2019. 6, 13

[6] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. In *Proceedings of the 7th International Conference on Learning Representations*, 2019. 2, 5

[7] Moses S. Charikar. Similarity estimation techniques from rounding algorithms. In *Proceedings of the Thiry-Fourth Annual ACM Symposium on Theory of Computing*, STOC '02, 2002. doi: 10.1145/509907.509965. 4

[8] Yuanpei Chen, Chen Wang, Li Fei-Fei, and Karen Liu. Sequential dexterity: Chaining dexterous policies for long-horizon manipulation. In *Proceedings of the 7th Conference on Robot Learning*, pages 3809–3829. PMLR, 2023. 2, 3

[9] Yuanpei Chen, Yiran Geng, Fangwei Zhong, Jiaming Ji, Jiechuang Jiang, Zongqing Lu, Hao Dong, and Yaodong Yang. Bi-dexhands: Towards human-level bimanual dexterous manipulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(5):2804–2818, 2024. doi: 10.1109/TPAMI.2023.3339515. 2, 3

[10] Yuanpei Chen, Chen Wang, Yaodong Yang, and Karen Liu. Object-centric dexterous manipulation from human motion data. In *Proceedings of the 8th Conference on Robot Learning*, pages 3828–3846. PMLR, 2025. 2, 3

[11] Zeyuan Chen, Qiyang Yan, Yuanpei Chen, Tianhao Wu, Jiyao Zhang, Zihan Ding, Jinzhou Li, Yaodong Yang, and Hao Dong. Clutterdexgrasp: A sim-to-real system for general dexterous grasping in cluttered scenes. In *Proceedings of the 9th Conference on Robot Learning*, pages 885–905. PMLR, 2025. 2, 3

[12] Xianyi Cheng, Eric Huang, Yifan Hou, and Matthew T. Mason. Contact mode guided motion planning for quasidynamic dexterous manipulation in 3d. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2730–2736, 2022. doi: 10.1109/ICRA46639.2022.9811872. 2

[13] Sudeep Dasari, Abhinav Gupta, and Vikash Kumar. Learning dexterous manipulation from exemplar object trajectories and pre-grasps. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3889–3896, 2023. doi: 10.1109/ICRA48891.2023.10161147. 3

[14] Adrien Ecoffet, Joost Huizinga, Joel Lehman, Kenneth O Stanley, and Jeff Clune. Go-explore: a new approach for hard-exploration problems. *arXiv preprint arXiv:1901.10995*, 2019. 5

[15] Zicong Fan, Omid Taheri, Dimitrios Tzionas, Muhammed Kocabas, Manuel Kaufmann, Michael J. Black, and Otmar Hilliges. ARCTIC: A dataset for dexterous bimanual hand-object manipulation. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 6, 13

[16] Irmak Guzey, Yinlong Dai, Georgy Savva, Raunaq Bhirangi, and Lerrel Pinto. Bridging the human to robot dexterity gap through object-oriented rewards. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3344–3351, 2025. doi: 10.1109/ICRA55743.2025.11128690. 2, 3

[17] Tairan He, Zhengyi Luo, Wenli Xiao, Chong Zhang, Kris Kitani, Changliu Liu, and Guanya Shi. Learning human-to-humanoid real-time whole-body teleoperation. in 2024 ieee. In *RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8944–8951. 1

[18] Sandy H Huang, Martina Zambelli, Jackie Kay, Murilo F Martins, Yuval Tassa, Patrick M Pilarski, and Raia Hadsell. Learning gentle object manipulation with curiosity-driven deep reinforcement learning. *arXiv preprint arXiv:1903.08542*, 2019. 2, 3

[19] Léonard Hussenot, Robert Dadashi, Matthieu Geist, and Olivier Pietquin. Show me the way: Intrinsic motivation from demonstrations. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, page 620–628. International Foundation for

Autonomous Agents and Multiagent Systems, 2021. 3

[20] Changwei Jing, Jai Krishna Bandi, Jianglong Ye, Yan Duan, Pieter Abbeel, Xiaolong Wang, and Sha Yi. Contact-aware neural dynamics, 2026. URL https://arxiv.org/abs/2601.12796. 2, 3

[21] Gagan Khandate, Siqi Shang, Eric T Chang, Tristan L Saidi, Johnson Adams, and Matei Ciocarlie. Sampling-based Exploration for Reinforcement Learning of Dexterous Manipulation. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023. doi: 10.15607/RSS.2023.XIX.020. 3

[22] J Zico Kolter and Andrew Y Ng. Near-bayesian exploration in polynomial time. In *Proceedings of the 26th International Conference on Machine Learning*, pages 513–520, 2009. 2

[23] Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Science Robotics*, 2020. 1

[24] Toru Lin, Kartik Sachdev, Linxi Fan, Jitendra Malik, and Yuke Zhu. Sim-to-real reinforcement learning for vision-based dexterous manipulation on humanoids. In *Proceedings of the 9th Conference on Robot Learning*, pages 4926–4940. PMLR, 2025. 2, 3

[25] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, et al. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*, 2021. 6, 12

[26] Zhao Mandi, Yifan Hou, Dieter Fox, Yashraj Narang, Ajay Mandlekar, and Shuran Song. Dexmachina: Functional retargeting for bimanual dexterous manipulation. *arXiv preprint arXiv:2505.24853*, 2025. 2, 3

[27] Ashvin Nair, Bob McGrew, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Overcoming exploration in reinforcement learning with demonstrations. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6292–6299, 2018. doi: 10.1109/ICRA.2018.8463162. 3

[28] Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the 34th International Conference on Machine Learning*, pages 2778–2787. PMLR, 2017. 2, 3

[29] Deepak Pathak, Dhiraj Gandhi, and Abhinav Gupta. Self-supervised exploration via disagreement. In *Proceedings of the 36th International Conference on Machine Learning*, pages 5062–5071. PMLR, 2019. 2, 3

[30] Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel van de Panne. Deepmimic: example-guided deep reinforcement learning of physics-based character skills. *ACM Trans. Graph.*, July 2018. 1

[31] Aleksei Petrenko, Arthur Allshire, Gavriel State, Ankur Handa, and Viktor Makoviychuk. DexPBT: Scaling up Dexterous Manipulation for Hand-Arm Systems with Population Based Training. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023. doi: 10.15607/RSS.2023.XIX.037. 2, 3

[32] C. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 77–85, 2016. URL https://api.semanticscholar.org/CorpusID:5115938. 14

[33] Haozhi Qi, Brent Yi, Sudharshan Suresh, Mike Lambeta, Yi Ma, Roberto Calandra, and Jitendra Malik. General in-hand object rotation with vision and touch. In *Proceedings of The 7th Conference on Robot Learning*, pages 2549–2564. PMLR, 2023. 2

[34] Sai Rajeswar, Cyril Ibrahim, Nitin Surya, Florian Golemo, David Vazquez, Aaron Courville, and Pedro O. Pinheiro. Haptics-based curiosity for sparse-reward tasks. In *Proceedings of the 5th Conference on Robot Learning*, pages 395–405. PMLR, 2022. 2, 3, 6

[35] Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. In *Proceedings of Robotics: Science and Systems*, Pittsburgh, Pennsylvania, June 2018. doi: 10.15607/RSS.2018.XIV.049. 2, 3

[36] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. URL https://arxiv.org/abs/2408.00714. 14

[37] Nikita Rudin, David Hoeller, Philipp Reist, and Marco Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning. In *Conference on robot learning*, pages 91–100. PMLR, 2022. 1

[38] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 3

[39] Clemens Schwarke, Victor Klemm, Matthijs van der Boon, Marko Bjelonic, and Marco Hutter. Curiosity-driven learning of joint locomotion and manipulation tasks. In *Proceedings of the 7th Conference on Robot Learning*, pages 2594–2610. PMLR, 2023. 2, 6

[40] Kenneth Shaw, Ananye Agarwal, and Deepak Pathak. Leap hand: Low-cost, efficient, and anthropomorphic hand for robot learning. *Robotics: Science and Systems (RSS)*, 2023. 6, 8

[41] Bradly C Stadie, Sergey Levine, and Pieter Abbeel. Incentivizing exploration in reinforcement learning with deep predictive models. *arXiv preprint arXiv:1507.00814*, 2015. 2, 3

[42] Alexander L Strehl and Michael L Littman. A theoretical analysis of model-based interval estimation. In *Proceed-*

ings of the 22th International Conference on Machine Learning, pages 856–863, 2005. 2

[43] Alexander L Strehl and Michael L Littman. An analysis of model-based interval estimation for markov decision processes. *Journal of Computer and System Sciences*, 74 (8):1309–1331, 2008. 2

[44] Adrien Ali Taïga, Aaron Courville, and Marc G Bellemare. Approximate exploration through state abstraction. *arXiv preprint arXiv:1808.09819*, 2018. 2

[45] Haoran Tang, Rein Houthooft, Davis Foote, Adam Stooke, OpenAI Xi Chen, Yan Duan, John Schulman, Filip DeTurck, and Pieter Abbeel. #exploration: A study of count-based exploration for deep reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 2, 4, 6, 15

[46] Yuhan Wang, Yu Li, Yaodong Yang, and Yuanpei Chen. Dexterous non-prehensile manipulation for ungraspable object via extrinsic dexterity. *arXiv preprint arXiv:2503.23120*, 2025. 2, 3

[47] Zhenyu Wei, Zhixuan Xu, Jingxiang Guo, Yiwen Hou, Chongkai Gao, Zhehao Cai, Jiayu Luo, and Lin Shao. $\mathcal{D}(\mathcal{R},\mathcal{O})$ grasp: A unified representation of robot and object interaction for cross-embodiment dexterous grasping. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4982–4988, 2025. doi: 10.1109/ICRA55743.2025.11127754. 4

[48] Yueh-Hua Wu, Jiashun Wang, and Xiaolong Wang. Learning generalizable dexterous manipulation from human grasp affordance. In *Proceedings of the 7th Conference on Robot Learning*, pages 3418–3433. PMLR, 2023. 2, 3

[49] Lixin Xu, Zixuan Liu, Zhewei Gui, Jingxiang Guo, Zeyu Jiang, Tongzhou Zhang, Zhixuan Xu, Chongkai Gao, and Lin Shao. Dexsingrasp: Learning a unified policy for dexterous object singulation and grasping in densely cluttered environments. *IEEE Robotics and Automation Letters*, 11(2):1346–1353, 2026. doi: 10.1109/LRA.2025.3641152. 2, 3

[50] Yinzhen Xu, Weikang Wan, Jialiang Zhang, Haoran Liu, Zikang Shan, Hao Shen, Ruicheng Wang, Haoran Geng, Yijia Weng, Jiayi Chen, Tengyu Liu, Li Yi, and He Wang. Unidexgrasp: Universal robotic dexterous grasping via learning diverse proposal generation and goal-conditioned policy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4737–4746, June 2023. 3

[51] Lixin Yang, Kailin Li, Xinyu Zhan, Fei Wu, Anran Xu, Liu Liu, and Cewu Lu. OakInk: A large-scale knowledge repository for understanding hand-object interaction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 16

[52] Zhao-Heng Yin, Binghao Huang, Yuzhe Qin, Qifeng Chen, and Xiaolong Wang. Rotating without seeing: Towards in-hand dexterity through touch. *arXiv preprint arXiv:2303.10880*, 2023. 2

[53] Chong Zhang, Wenli Xiao, Tairan He, and Guanya Shi. Wococo: Learning whole-body humanoid control with sequential contacts. In *Proceedings of the 8th Conference on Robot Learning*, pages 455–472. PMLR, 2025. 6

[54] Hui Zhang, Sammy Christen, Zicong Fan, Otmar Hilliges, and Jie Song. Graspxl: Generating grasping motions for diverse objects at scale. In *Proceedings of European Conference on Computer Vision*, pages 386–403. Springer, 2024. 3

[55] Mengchao Zhang, Devesh K. Jha, Arvind U. Raghunathan, and Kris Hauser. Simultaneous trajectory optimization and contact selection for contact-rich manipulation with high-fidelity geometry. *IEEE Transactions on Robotics*, 41:2677–2690, 2025. doi: 10.1109/TRO.2025.3554380. 2

[56] Tianjun Zhang, Huazhe Xu, Xiaolong Wang, Yi Wu, Kurt Keutzer, Joseph E Gonzalez, and Yuandong Tian. Noveld: A simple yet effective exploration criterion. In *Advances in Neural Information Processing Systems*, volume 34. Curran Associates, Inc., 2021. 5

Appendix

Appendix Contents

A. Implementation Details

1) Object and Hand Representations: We represent the manipulated object by a canonical surface point cloud $\{\mathbf{p}_m\}_{m=1}^{M}$ with associated outward normals $\{\mathbf{n}_m\}_{m=1}^{M}$. To obtain discrete surface regions used by our contact coverage counter, we cluster the canonical points into K regions using farthest-point sampling (FPS) initialization followed by K-means clustering. Point-to-center assignment uses a weighted combination of positional distance and normal disagreement:

$$d(m,k) = (1-\lambda)\,\|\mathbf{p}_m - \boldsymbol{\mu}_k\|_2 + \lambda\big(1 - \mathbf{n}_m^\top \bar{\mathbf{n}}_k\big),$$

where $\boldsymbol{\mu}_k$ and $\bar{\mathbf{n}}_k$ are the position and mean normal of region k, and λ corresponds to the normal weight. This yields a region label $\xi(m) \in \{1,\dots,K\}$ for each surface point.

2) Contact Coverage Counter: Because Isaac Gym does not provide rigid-body pairwise contact queries at keypoint granularity, we approximate keypoint contact using a distance–force criterion. For keypoint l_f, we compute the distance r_f to the nearest object surface point and the net contact force magnitude $\|\mathbf{F}_f\|_2$. A contact is registered if $r_f < \delta_{\text{dist}}$ and $\|\mathbf{F}_f\|_2 > \delta_{\text{force}}$, where $\delta_{\text{dist}} = 0.5\,\text{cm}$ and $\delta_{\text{force}} = 0.01\,\text{N}$. This binary signal updates the per-keypoint coverage counters over surface regions $\xi(m)$.

3) Energy-Based Reaching Reward: To encourage physically feasible approach directions, we weight each object surface point m by a directional term computed from the surface normal $\mathbf{n}_m$, the keypoint position $\mathbf{p}_{l_f}$, and the keypoint normal direction $\mathbf{n}_{l_f}$. Let $\mathbf{v}_{l_f,m} = \mathbf{p}_{l_f} - \mathbf{p}_m$ denote the line from surface point m to keypoint l_f. We first suppress back-facing points using

$$w^{\text{obj}}_{l_f,m} = \big[\cos(\theta^{\text{obj}}_{l_f,m})\big]_+ = \left[\frac{\mathbf{v}_{l_f,m}^\top \mathbf{n}_m}{\|\mathbf{v}_{l_f,m}\|\,\|\mathbf{n}_m\|}\right]_+,$$

and further prefer palm-facing configurations via

$$w^{\text{keypoint}}_{l_f,m} = \big[-\cos(\theta^{\text{keypoint}}_{l_f,m})\big]_+ = \left[-\frac{\mathbf{d}_{l_f}^\top \mathbf{n}_m}{\|\mathbf{d}_{l_f}\|\,\|\mathbf{n}_m\|}\right]_+,$$

where $[\cdot]_+ = \min(\max(\cdot,0),1)$. The final directional weight is the product $w^{\text{dir}}_{l_f,m} = w^{\text{obj}}_{l_f,m}\, w^{\text{keypoint}}_{l_f,m}$. Our energy-based reaching reward for keypoint l_f is then computed by summing energy over surface points, modulated by this directional weight and an exponential distance kernel:

$$\Phi_f = \sum_m g\big(\mathbf{C}_{s,f,\xi(m)}\big)\; w^{\text{dir}}_{l_f,m}\; \exp\!\left(-\frac{\|\mathbf{p}_{l_f} - \mathbf{p}_m\|_2}{\delta}\right),$$

with δ the kernel scale.

Additionally, we account for line-of-sight occlusions between keypoints l_f and object point m. Let $w^{\text{occ}}_{l_f,m} \in \{0,1\}$ be a binary visibility mask that equals 1 only if the segment along $\mathbf{v}_{l_f,m} = \mathbf{p}_{l_f} - \mathbf{p}_m$ is not blocked by any obstacle. In the implementation, $w^{\text{occ}}_{l_f,m}$ is computed via a ray–box intersection test against an oriented bounding box: we cast a ray from $\mathbf{p}_{l_f}$ toward $\mathbf{p}_m$ (treating the surface point as the endpoint at $t = 1$) and set $w^{\text{occ}}_{l_f,m} = 0$ if any valid intersection occurs for $t \in (0,1)$; otherwise $w^{\text{occ}}_{l_f,m} = 1$. We apply this by multiplicatively masking the distance kernel, yielding

$$\Phi_f = \sum_m g\big(\mathbf{C}_{s,f,\xi(m)}\big)\; w^{\text{dir}}_{l_f,m}\; w^{\text{occ}}_{l_f,m}\; \exp\!\left(-\frac{\|\mathbf{p}_{l_f} - \mathbf{p}_m\|_2}{\delta}\right).$$

This occlusion handling is necessary in cluttered object singulation and constrained object retrieval tasks, where nearby geometry should not contribute to the energy-based reaching reward if it is not directly reachable along the approach line.

B. Simulation Experiments Details

All simulation experiments are conducted with **2048 parallel environments** using Isaac Gym [25]. Policy updates are performed every **16 environment steps**. For each task, results are reported as the **mean and standard deviation over 5 random seeds**.

1) Cluttered Object Singulation: In this task, the robot is required to extract a single target object from a densely packed shelf. Each scene consists of 5 upright objects of the same size arranged in a single row. At the beginning of simulation creation, the position of the entire book grid is randomized along the edge of the shelf, and the target object is randomly

TABLE V: **Ablation of CCGE Reward Scale** on Constrained Object Retrieval.

Setting	Success Rate (%) ↑	Steps ($\times 32M$) at 70% SR↓
$\alpha = 50.0, \beta = 0.32$	17±35	7.0±2.0
$\alpha = 100.0, \beta = 0.64$	52±42	5.2±2.5
$\alpha = 400.0, \beta = 2.56$	**89**±4	2.8±1.6
$\alpha = 800.0, \beta = 5.12$	45±56	5.2±2.4
$\alpha = \mathbf{200.0}, \beta = \mathbf{1.28}$ **(Ours)**	88±6	**2.0±2.0**

TABLE VI: **Ablation of Learning Parameter for State Clustering** on Constrained Object Retrieval.

Setting	Success Rate (%) ↑	Steps ($\times 32M$) at 70% SR↓
$H = 4, \lambda = 0.5$	90±2	2.8±1.5
$H = 4, \lambda = 1.0$	85±6	3.2±2.2
$H = 4, \lambda = 2.0$	52±43	4.2±3.1
$H = 5, \lambda = 0.5$	73±36	3.0±2.6
$H = 5, \lambda = 2.0$	72±36	3.0±2.6
$H = 6, \lambda = 0.5$	64±35	4.6±2.2
$H = 6, \lambda = 1.0$	71±36	4.0±2.4
$H = 6, \lambda = 2.0$	**91**±4	2.8±1.3
$H = \mathbf{5}, \lambda = \mathbf{1.0}$ **(Ours)**	88±6	**2.0±2.0**

selected among all the books. All non-target books remain fixed, while only the target book is movable. The observation space includes hand root poses and velocities; fingertip poses and velocities; target object positions and velocities; non-target object positions; relative hand–object poses; binary tactile signals at each hand link; and the previous action. The action space consists of delta end-effector poses for the robotic arm and delta joint angles for the dexterous hand, with all joints operating under position control. The task is considered successful when the target book sufficiently reaches the goal position.

2) Constrained Object Retrieval: In this task, the robot must retrieve a cube from a top-opening box by sliding it along the interior walls. The cube is lower than the top rim of the box, and the initial gap between the cube and the box is insufficient for inserting the LEAP Hand fingers, making direct grasping infeasible. As a result, successful retrieval requires contact-rich, constrained motions guided by interactions with the box interior. The observation, action spaces and control mode are identical to those used in Cluttered Object Singulation. The task is considered successful when the cube sufficiently reaches the goal position.

3) In-Hand Reorientation: In this task, the robotic hand is required to rotate an object to a specified target orientation. The task is evaluated under two settings: up-facing and down-facing. In the up-facing setting, objects includes the elephant, mug, bunny, duck, mouse, and teapot from ContactDB [5] dataset, as well as the letter R and letter S. In the down-facing setting, we use the elephant, mug, bunny, duck, mouse, and teapot, together with a slender cuboid (16 cm × 3 cm × 3 cm). To enable stable learning in the down-facing setting, we rescale the 6 ContactDB objects and assign each object a fixed initial pose and hand joint configuration, such that the object is initially grasped. For all episodes, the object initial pose (except for down-facing) and the goal orientation are randomly sampled. The observation includes hand joint positions, velocities, and forces; object poses and velocities; goal orientation and distance; and the previous action. The action space consists of absolute joint angles for the dexterous hand, with all joints operating under position control. The task is considered successful when the object sufficiently reaches the goal orientation.

TABLE VII: **Qualitative Results of using Allegro Hand.**

Method	Success Rate (%) ↑		Steps ($\times 32M$) at 70% SR↓	
	Singulation	Retrieval	Singulation	Retrieval
TR	87±5	0±0	3.2±0.4	8.0±0.0
LHCC	72±18	0±0	3.8±0.4	8.0±0.0
HaC	8±16	0±0	4.0±0.0	8.0±0.0
RND-Dist	55±45	0±0	3.4±0.8	8.0±0.0
CCGE (Ours)	**93**±4	**89**±4	**2.2±1.0**	**3.8±1.0**

TABLE VIII: **Sensitivity Analysis of Hand Keypoint Selection** on Constrained Object Retrieval.

Setting	Success Rate (%) ↑	Steps ($\times 32M$) at 70% SR↓
Low-Level Noise	87±3	4.4±1.2
High-Level Noise	86±3	2.2±0.4
Predefined Keypoints **(Ours)**	**88**±6	**2.0±2.0**

The results of the default upfacing setting are shown in Table I, and the results of the down-facing setting are shown in Table X.

4) Bimanual Manipulation: In this task, two robotic hands must coordinately manipulate articulated objects, including flipping open the hinged lid of a waffle iron or opening a box from the ARCTIC [15] dataset. Successful execution requires synchronized bimanual control to stabilize the object while actuating its articulated parts. The observation includes hand root poses and velocities for both hands; hand joint positions and velocities; object and articulated-part poses and velocities; relative hand–object and hand–hand poses; fingertip poses and velocities; binary tactile signals; and the previous action. The action space consists of delta end-effector poses for both arms and delta joint angles for both hands, with all joints operating under position control. The task is considered successful when the articulated joint sufficiently reaches the goal position.

C. Real-World Experiments Details

To answer **Q5**, we evaluate the policies on a real-world Cluttered Object Singulation task. Since the teacher policy is trained in simulation with privileged state information, we distill it into a vision-based student policy that conditions on proprioceptive inputs and fused point clouds.

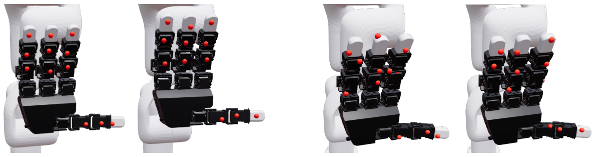
For real-world deployment, we reconstruct point clouds from two RGB-D cameras to mitigate occlusions. For each camera c, depth pixels are back-projected into a camera-frame point cloud using the camera intrinsics, then transformed into the robot base frame using calibrated extrinsics. Points from all views are subsequently concatenated. We then apply a fixed axis-aligned workspace crop and remove invalid depth

TABLE IX: **In-Hand Reorientation (Up-Facing) Experiments.**

Method	Success Rate (%) ↑									Steps (×32M) at 70% SR↓								
	Elephant	Mug	Bunny	Duck	Mouse	Teapot	Letter R	Letter S	Avg.	Elephant	Mug	Bunny	Duck	Mouse	Teapot	Letter R	Letter S	Avg.
TR	80±5	82±2	80±4	94±1	73±8	68±9	73±4	79±2	79±9	3.8±0.8	3.0±0.6	3.2±1.0	**1.0±0.0**	4.4±0.8	4.6±0.5	4.8±2.8	3.0±0.0	3.4±1.3
LHCC	84±3	80±7	78±2	94±2	82±3	72±4	79±3	83±2	81±7	2.6±1.0	3.4±1.0	3.6±0.5	1.2±0.4	3.4±0.5	4.4±0.2	5.1±2.9	2.4±0.5	3.1±1.2
HaC	80±5	81±3	79±2	93±2	77±11	72±4	77±8	83±2	80±8	3.2±0.8	3.2±0.4	3.4±0.5	1.6±0.8	3.8±0.8	4.6±0.5	4.9±2.6	**2.0±0.0**	3.2±1.2
RND-Dist	85±6	76±11	78±10	92±5	72±14	71±9	77±10	78±13	78±11	1.2±0.4	3.0±0.6	3.0±0.6	1.2±0.4	4.2±0.8	4.2±0.4	5.1±2.8	2.2±0.8	2.9±1.2
CCGE (Ours)	**93±1**	**89±1**	**88±3**	**96±1**	**83±3**	**81±2**	**88±3**	**85±2**	**88±5**	**1.0±0.0**	**1.0±0.0**	**1.6±0.8**	**1.0±0.0**	**3.2±0.4**	**2.4±0.5**	**2.0±0.7**	**2.0±0.7**	**1.8±0.9**

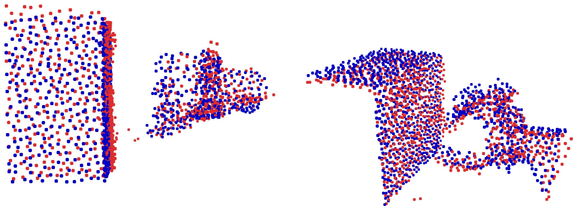
TABLE X: **In-Hand Reorientation (Down-Facing) Experiments.**

Method	Success Rate (%) ↑								Steps (×32M) at 70% SR↓							
	Elephant	Mug	Bunny	Duck	Mouse	Teapot	Cube (16x3x3)	Avg.	Elephant	Mug	Bunny	Duck	Mouse	Teapot	Cube (16x3x3)	Avg.
TR	80±4	90±8	71±3	61±7	80±5	92±1	58±10	76±14	3.2±1.0	3.6±2.4	6.8±1.5	8.0±0.0	3.6±1.7	**1.0±0.0**	7.6±0.8	4.8±2.8
LHCC	75±6	88±8	71±3	54±3	80±7	92±2	63±6	75±14	5.4±2.5	4.2±2.0	6.0±1.7	8.0±0.0	3.2±2.6	1.2±0.4	8.0±0.0	5.1±2.9
HaC	75±4	92±1	70±6	52±7	**83±3**	89±2	61±7	75±14	5.2±1.5	3.2±0.4	6.2±2.2	8.0±0.0	**2.2±0.8**	1.8±0.4	8.0±0.0	4.9±2.6
RND-Dist	76±8	93±2	67±4	62±4	82±3	91±2	59±8	76±14	4.6±1.9	3.2±0.4	8.0±0.0	8.0±0.0	2.4±0.8	1.4±0.5	8.0±0.0	5.1±2.8
CCGE (Ours)	**95±1**	**94±4**	**79±9**	**65±2**	76±10	**98±0**	**76±4**	**83±9**	**2.2±0.8**	**2.8±1.2**	**5.0±2.2**	**6.4±1.6**	3.4±2.0	**1.0±0.0**	**4.8±1.8**	**3.7±2.4**



(a) Low-Level Noise (b) High-Level Noise

Fig. 8: **Two Levels of Perturbation on Hand Keypoints.**



(a) Top-View Point Clouds (b) Side-View Point Clouds

Fig. 9: Visualization of point clouds in simulation (blue) and real world (real).

points. Finally, the fused point cloud is downsampled to a fixed size using farthest point sampling (FPS) without replacement. A comparison between the reconstructed point clouds in simulation and in the real world is shown in Fig. 9.

To provide target awareness, we apply SAM2 [36] on the RGB streams to obtain per-view target masks. Each point is augmented with a binary mask indicator and represented as a 4D vector (x, y, z, m), where $m \in \{0, 1\}$ indicates whether the point belongs to the target. The masked point cloud is processed by a PointNet encoder [32] to extract a permutation-invariant feature, which is concatenated with the proprioceptive observations and fed into the student policy. The student policy takes a two-step observation history as input. The complete observation space is summarized in Table XI. The action space is identical to the teacher policy, consisting of relative commands for a 6-DoF end-effector and 16 hand joints.

We use the teacher policies as oracles to provide supervision for student training. We roll out the teacher policy to collect 1,000 successful trajectories with randomly sampled target objects. The student policy is then trained via behavior cloning to predict the teacher's actions from proprioceptive inputs and point cloud observations, using an L1 loss between the predicted actions and the teacher actions. The real-world execution of the distilled student policy is shown in Fig. 10.

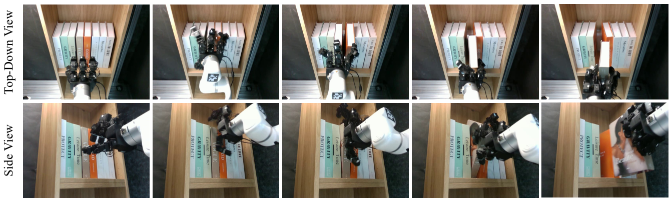


Fig. 10: **Real-world Policy Execution.** We show a temporal sequence (left to right) of the policy executing the shelf object singulation task.

TABLE XI: **Observation space for student policies.**

Name	Dimension
Point Cloud	2 × 1024 × 4
Proprioceptive	
Arm Joint Position	2 × 7
Hand Joint Position	2 × 16

For the in-hand reorientation task, we evaluate sim-to-real consistency via open-loop trajectory replay. Specifically, we execute in the real world the action sequences generated by the privileged teacher policy in simulation, with the initial object pose and hand configuration aligned to their simulated counterparts. As shown in Fig. 11, real-world rollouts exhibit object pose changes consistent with simulation, suggesting that the manipulation behaviors learned by CCGE in simulation can transfer to the real world.

D. Additional Experiment Results

1) Reward Scale Ablation Studies: We study the effect of the CCGE reward scale on the *Constrained Object Retrieval* task by varying the contact coverage reward scale α in Equation 7 and the energy-based reaching reward scale β in Equation 8, as reported in Table V. The results show that an

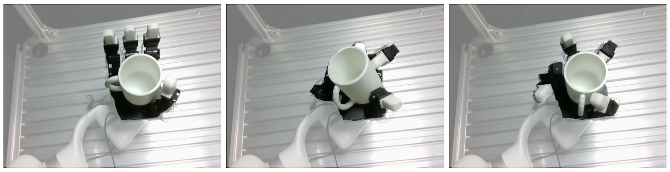


Fig. 11: **Real-World Trajectory Replay.** Simulated action sequences achieve consistent 90° z-axis rotations when replayed in the real world.

appropriate reward scale is crucial for both final performance and learning efficiency. Small reward scales lead to insufficient exploration and low success rates, while excessively large scales degrade performance and introduce training instability. Our chosen setting ($\alpha = 200.0$, $\beta = 1.28$) achieves a high final success rate while converging fastest to the 70% success threshold, demonstrating a favorable balance between exploration strength and training stability. Under this setting, the per-step exploration reward remains approximately 2 orders of magnitude smaller than the task reward, which we empirically find to be the most effective scale. We observe a consistent trend when tuning other exploration baselines. Therefore, in Table I of the main paper, we apply the same reward scale across all methods to ensure a fair comparison.

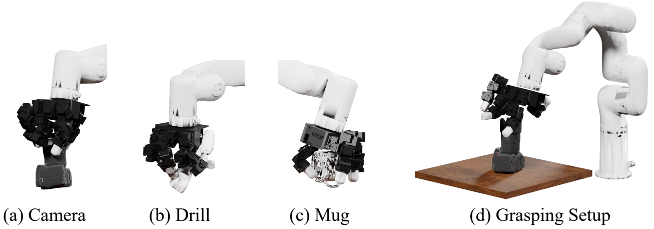
2) Learning Parameter Ablation Studies for Object State Clustering: We analyze the impact of the learning parameters in the object state clustering module by varying the binary regularization weight λ and the hash length H, which jointly determine the discretization behavior of the learned state representation. As shown in Equation 1, the regularization term controlled by λ encourages each binary code element b_i to approach either 0 or 1, while the hash length H defines the total number of object state clusters, i.e., $s \in \{0, \dots, 2^H - 1\}$. Specifically, H defines an upper bound on the number of state clusters, while λ implicitly controls the number of clusters that are effectively utilized by encouraging binarization [45].

As shown in Table VI, we observe that the original upper bound ($H = 5$, corresponding to 2^5 clusters) is relatively generous for most of the tasks including Constrained Object Retrieval. Reducing the upper bound to $H = 4$ still yields strong performance, and further decreasing λ (e.g., $H = 4$, $\lambda = 0.5$) improves both success rate and sample efficiency by allowing a larger number of effective state clusters to be used. In contrast, increasing λ overly constrains the representation, causing diverse object states to collapse into fewer clusters and leading to degraded performance. A similar trade-off is observed when increasing the cluster upper bound. For larger H, a stronger regularization is required to prevent over-clustering of the state space. For example, when $H = 6$, increasing λ to 2.0 effectively limits the number of active clusters and restores performance. These results highlight the importance of jointly tuning H and λ to balance state discrimination and sample efficiency.

3) Cross-Embodiment Experiments: To answer **Q6**, we conduct cross-embodiment experiments using the Allegro Hand on the Cluttered Object Singulation and Constrained Object Retrieval tasks. All other settings, including observation and action spaces, training procedures, and hyperparameters, remain identical to those used with the LEAP Hand.

TABLE XII: **Bimanual Manipulation Experiemnts.**

Method	Success Rate (%) ↑			Steps ($\times 32M$) at 70% SR↓		
	Waffle Iron	Box	Avg.	Waffle Iron	Box	Avg.
TR	88±5	95±4	92±5	3.4±2.3	1.2±0.4	2.3±2.0
LHCC	86±6	95±3	90±7	3.8±1.2	3.2±2.2	3.5±1.8
HaC	83±12	87±12	85±14	7.6±1.6	1.4±0.8	4.5±3.4
RND-Dist	80±10	**97**±1	89±11	7.2±2.8	**1.0**±0.0	4.1±3.7
CCGE (Ours)	**93**±3	**97**±2	**95**±3	**2.4**±1.2	**1.0**±0.0	**1.7**±1.1



(a) Camera (b) Drill (c) Mug (d) Grasping Setup

Fig. 12: **Objects and Setup in the Grasping Task.**

As shown in Table VII, CCGE consistently improves performance over all baselines on both tasks when transferring to the Allegro Hand. Notably, CCGE achieves substantial gains in success rate for object retrieval, where several baselines fail to solve the task under the same training budget. Meanwhile, CCGE also reduces the number of interaction steps required to reach the success threshold, indicating improved learning efficiency across embodiments. These results suggest that the contact exploration encouraged by CCGE remains effective under changes in different hands, supporting its robustness in cross-embodiment dexterous manipulation.

4) Sensitivity Analysis of Keypoint Selection: Our current hand keypoints are predefined on the *palmar face* of each hand link. To evaluate the robustness of CCGE to keypoint selection, we perturb the predefined keypoints and recompute new keypoints by projecting each perturbed point onto the nearest point on the corresponding link surface. We consider two levels of perturbation shown in Figure 8. For *low-level noise*, noises are sampled from a spherical shell of radius $[0,\, 1.0]$ cm, resulting into keypoints largely remain on the palmar face. For *high-level noise*, the shell radius is expanded to $[1.0,\, 2.0]$ cm, which may cause the resulting keypoints to shift to the side face of the link. Performance under these settings is reported in Table VIII.

As shown in Table VIII, CCGE maintains stable performance under both low- and high-level perturbations, with only minor variations in success rate and learning efficiency. Notably, even when keypoints shift away from the palmar face, the performance degradation remains limited. These results indicate that CCGE does not rely on precise keypoint placement, but instead benefits from the overall structure of contact coverage, demonstrating robustness to moderate spatial variations in keypoint definition.

TABLE XIII: **Grasping Experiemnts.**

Method	Success Rate (%) ↑				Steps (×$32M$) at 60% SR↓			
	Camera	Mug	Drill	Avg.	Camera	Mug	Drill	Avg.
TR	71±7	76±2	57±33	68±21	8.3±2.0	2.0±0.7	6.3±2.3	5.0±3.2
LHCC	71±5	75±2	70±2	72±4	7.0±1.6	3.0±1.2	6.3±0.5	5.2±2.2
HaC	71±2	54±2	65±8	63±16	6.0±0.8	6.8±3.4	9.0±0.8	7.2±2.6
RND-Dist	49±34	75±2	43±9	58±24	8.0±1.4	3.3±0.4	10.0±0.0	6.7±3.0
CCGE (Ours)	**75**±4	**77**±2	**84**±2	**79**±5	**5.8**±3.8	**1.5**±0.9	**2.3**±0.8	**3.2**±3.0

5) In-Hand Reorientation: Table I shows the average results on the up-facing setting over different objects introduced in Section B. The detailed results of each object are shown in Table IX. Table X reports additional results on the down-facing in-hand reorientation setting. Compared with extrinsic exploration baselines, CCGE consistently achieves higher success rates across most object categories, while also reaching the 70% success threshold with fewer interaction steps on average. Notably, CCGE maintains strong performance on geometrically complex objects such as the elephant and teapot, and also shows clear advantages on the slender cube, indicating its effectiveness in handling contact-rich reorientation that requires maintaining stable contacts against gravity. These results further demonstrate the robustness of CCGE in challenging in-hand manipulation scenarios where stable contact exploration is critical.

6) Bimanual Manipulation: Table I shows the average results over 2 different objects introduced in Section B. The detailed results of each object are shown in Table XII.

As shown in Table XII, CCGE achieves the most notable improvement on the geometrically more complex waffle iron, where coordinated bimanual interaction and structured contact exploration are more critical for successful manipulation. In contrast, on the simpler box object, CCGE maintains performance comparable to or better than existing baselines. These results suggest that CCGE is particularly beneficial in contact-rich and geometrically challenging bimanual scenarios.

7) Grasping: Grasping is a fundamental dexterous manipulation task in which excessive exploration can be detrimental to task performance. We additionally evaluate CCGE on a grasping task using the rescaled camera, mug, and drill objects from the OakInk [51] dataset. For all episodes, the object initial pose is randomly sampled. The observation space, action space, and control mode follow the same design as the preceding manipulation tasks. The task is considered successful when the object reaches the specified goal position.

As shown in Table XIII, several exploration-based baselines exhibit degraded performance compared to the task-reward-only (TR) baseline, suggesting that overly aggressive exploration may interfere with stable grasp acquisition. In particular, RND-Dist encourages novelty in hand–object distance even in free space, which guides the policy toward exploring non-contact behaviors and conflicts with the prolonged, surface-enveloping finger contact required for stable grasping. In contrast, CCGE consistently improves both success rate and sample efficiency across all objects, demonstrating its ability to encourage structured contact exploration without disrupting task execution.

TABLE XIV: **Real-world shelf object singulation results.** We report the success rates across 30 trials.

Method	Singulation Success ↑	Task Success (Grasp) ↑
TR Baseline	36.7%	3.3%
Ours (CCGE)	**76.7%**	**33.3%**

8) Real-World Experimental Results: We deploy the vision student policy on a real-world shelf object singulation task consisting of two phases: object singulation followed by grasping and transporting the object to a target position. For each policy, we conduct 30 trials. A singulation is considered successful if the object pose becomes pre-grasp feasible and at least half of the object is exposed. Final task success is defined as transporting the object to the specified target position. As summarized in Table XIV, the policy learned with CCGE exhibits more reliable behavior than the student policy trained with the Task Reward (TR) baseline, achieving higher singulation and grasp success rates.