Task-Oriented Grasping Using Reinforcement Learning with a Contextual Reward Machine

Hui Li, Akhlak Uz Zaman, Fujian Yan, and Hongsheng He

***Abstract*—This paper presents a reinforcement learning framework that incorporates a Contextual Reward Machine for task-oriented grasping. The Contextual Reward Machine reduces task complexity by decomposing grasping tasks into manageable sub-tasks. Each sub-task is associated with a stage-specific context, including a reward function, an action space, and a state abstraction function. This contextual information enables efficient intra-stage guidance and improves learning efficiency by reducing the state-action space and guiding exploration within clearly defined boundaries. In addition, transition rewards are introduced to encourage or penalize transitions between stages which guides the model toward desirable stage sequences and further accelerates convergence. When integrated with the Proximal Policy Optimization algorithm, the proposed method achieved a 95% success rate across 1,000 simulated grasping tasks encompassing diverse objects, affordances, and grasp topologies. It outperformed the state-of-the-art methods in both learning speed and success rate. The approach was transferred to a real robot, where it achieved a success rate of 83.3% in 60 grasping tasks over six affordances. These experimental results demonstrate superior accuracy, data efficiency, and learning efficiency. They underscore the model's potential to advance task-oriented grasping in both simulated and real-world settings.**

***Index Terms*—Context-Aware System, Task-Oriented Grasping, Reward Machine, Reinforcement Learning**

I. Introduction

Robotic dexterity, the ability of a robot to manipulate objects with precision, adaptability, and control, akin to human hand dexterity, is essential for performing complex tasks across diverse applications, including aerospace, automotive, manufacturing, and warehousing [1]. While robots excel in structured environments and repetitive tasks, they remain constrained in unstructured and dynamic scenarios. Advancing robotic dexterity has the potential to bridge this gap and allow robots to handle complex tasks in uncertain environments [2]. The current methods struggle with dexterous manipulation, especially when handling objects of varying shapes, sizes, and materials.

Grasping an object represents the initial and foundational step of dexterous manipulation. A key factor in grasping tasks is the selection of an appropriate grasp topology, which defines the specific configuration of a robotic hand when interacting with an object. The grasp topology plays a pivotal role in minimizing redundant finger movements and streamlining the overall manipulation process. Selecting a suitable topology is a critical prerequisite for effective manipulation, as it ensures stable object acquisition and simplifies downstream control [3]. A firm or adaptive grasp stabilizes the object and enhances task efficiency and execution success.

Research in this area has explored various strategies for determining effective grasp topologies, including learning from human demonstrations to replicate natural grasp patterns [4], designing soft robotic hands for flexible and compliant interactions [5], and leveraging advanced learning-based techniques for data-driven optimization [6]. Selecting a grasp topology that aligns with object properties and task requirements reduces manipulation complexity and improves performance across diverse scenarios [7].

While numerous grasp types exist, they generally cluster into a limited set that is suitable for manipulating everyday objects and tools [8], [9]. This insight has led to the development of grasp taxonomy that categorizes and simplifies grasp poses [10], [11]. The taxonomy provides a systematic way to map object characteristics and task requirements to suitable grasp types, thereby improving planning and control.

The choice of grasp topology is influenced by both object features, such as size, shape, surface texture, and mass, as well as task-specific requirements, including the intended action, applied forces, and environmental constraints [12], [13]. For instance, precision tasks, like picking up small or fragile objects, typically utilize pinch or tripod grasps. In contrast, tasks requiring greater stability or force, such as lifting heavy items, benefit from power grasps. Tool-use scenarios introduce additional complexity and often necessitate specialized topologies, such as cylindrical grasps for tool handles or lateral grasps for flat items. Building on these insights, our previous work [14] successfully integrated object features and task requirements into grasp strategies and established a structured framework for the grasping process.

Deploying grasping tasks presents significant challenges due to environmental uncertainties, particularly in unstructured environments. The perception of target objects is often incomplete or inaccurate, which negatively impacts grasping performance. Traditional planning methods struggle to manage these complexities because they rely on precise context modeling, which is rarely feasible in real-world scenarios.

Task-oriented grasping can be formulated as a sequential decision-making problem in which an agent learns optimal behaviors through trial and error to maximize cumulative rewards. Reinforcement learning (RL) has shown strong potential in addressing such problems effectively [15]–[17].

RL has demonstrated particular success in solving complex

Hui Li, Akhlak Uz Zaman and Hongsheng He are with the department of Computer Science, The University of Alabama, Tuscaloosa, AL, 35487, USA. This research is funded by NSF #2420355.

Fujian Yan is with the School of Computing, Wichita State University, Wichita, KS, 67260, USA.

Correspondence should be addressed to Hongsheng He, e-mail: hongsheng.he@ua.edu.

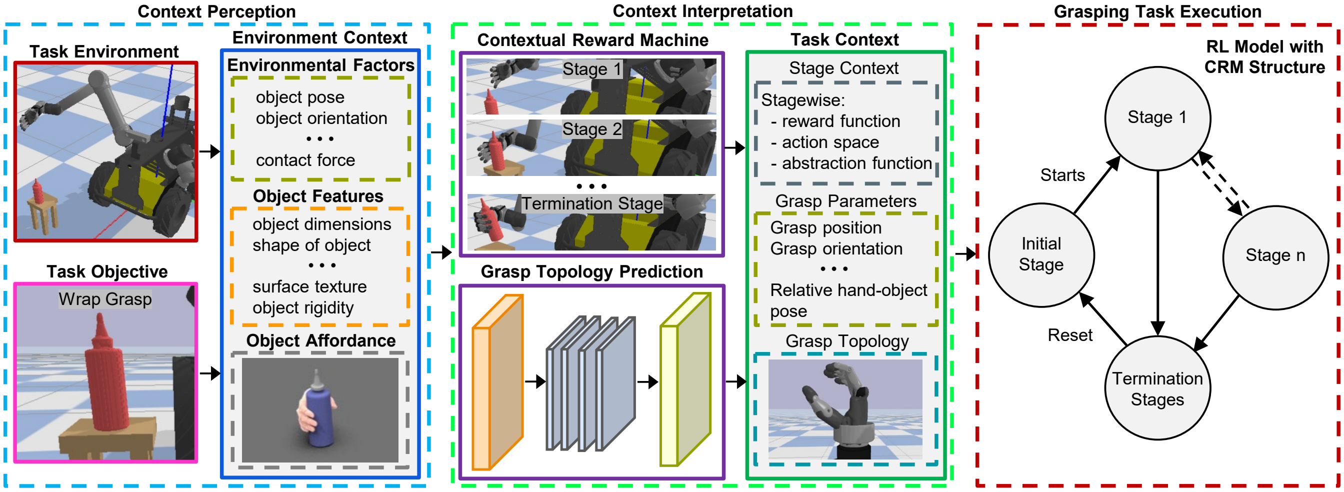

Figure 1. Context-aware task-oriented grasping framework with a contextual reward machine.

control problems in robotics, especially when integrated with Proximal Policy Optimization (PPO) [18]–[20]. PPO, a widely used RL algorithm, addresses many limitations of RL by improving training stability and sample efficiency [21]. Its simplicity and robustness make it well suited for high-dimensional and continuous action spaces, which enable effective policy optimization.

Despite these advantages, RL methods for grasping still face several challenges, including high computational demands and difficulties in achieving stable convergence. These limitations highlight the need for further development of RL techniques that can meet the demands of real-world applications.

To address these challenges, researchers have proposed multistage reinforcement learning approaches where each stage of a task is trained separately using specialized sub-networks that collaborate to determine an overall optimal policy [22]. Although this method achieves stable convergence, it remains computationally expensive. The reward machine framework has been introduced to solve the problems by organizing complex tasks into modular sub-tasks, each associated with a distinct reward function. This structure improves learning efficiency and reduces computation cost [23]. Although reward machines offer a structured approach to task decomposition, traditional implementations often lack the flexibility to adapt to dynamic environments or incorporate detailed contextual information. These constraints limit their applicability to real-world scenarios, where adaptability and context-awareness are essential for reliable robotic performance.

In this paper, we propose a context-aware task-oriented grasping approach that leverages a Contextual Reward Machine (CRM) to enhance efficiency and adaptability. The CRM decomposes grasping tasks into sequential stages with each stage defined by a stage-specific context including a reward function, an action space, and abstracted states. This structure guides intra-stage task progression and improves learning efficiency. Additionally, a transition reward mechanism is designed to facilitate smooth transitions between stages.

The general structure of the proposed method is illustrated in Fig. 1. In this approach, the context of the environment for a grasping task is continuously perceived and analyzed. The environmental context comprises object features, such as dimensions, shape, and texture. It also includes environmental factors, such as object pose, contact forces, obstacle positions, and object affordances. The object features and the task objective are processed by a pretrained grasp selection network to determine the appropriate grasp topology. Meanwhile, the environmental factors and object affordances are used by the CRM to identify the grasp location, determine the current stage, and retrieve the corresponding stage-specific context. They are also utilized by the RL agent to learn and optimize its policy. Task execution is carried out by an RL model, which integrates the grasp topology, grasp location, and stage-specific contexts under CRM framework. The model dynamically adapts to the changing conditions and performs robust, precise, and efficient grasps.

The main contributions of the paper include:

- ◇ We implemented a context-aware task-oriented dexterous grasping approach that enables adaptive and efficient grasping in unstructured environments.
- ◇ We designed and developed a Contextual Reward Machine that decomposes grasping tasks into sequential stages. It utilizes stage-specific contexts and transition rewards to enhance learning efficiency and adaptability.

II. Framework of Task-Oriented Grasping

Task-oriented grasping is inherently a context-aware process involving the perception of environmental context, the generation of grasp strategies based on that context, and the efficient, adaptive execution of those strategies. To address these challenges, environmental context was obtained through sensor fusion by integrating inputs from multiple sensors. A deep learning network was developed to generate grasp strategies, while a reinforcement learning model under CRM framework was designed and implemented to enable efficient and adaptive execution of grasping tasks.

A. Context Perception

Task-oriented grasping tasks require detailed environmental context. An RGB-D camera is used to capture object dimensions, shape, and surface characteristics. These object features guide the selection of an appropriate grasp topology. Environmental factors, such as the object's position, orientation, and nearby obstacles, are also captured with the same RGB-D camera. In parallel, force-sensing resistors (FSRs) are used to record contact forces and provide precise pressure data to improve task accuracy.

B. Grasp Topology Determination

A proper grasp topology facilitates smoother and more efficient manipulation. To simplify the selection process, we defined a grasp taxonomy comprising six primary topologies and developed a grasp selection network [14] that maps object features and task demands to the most suitable grasp topology.

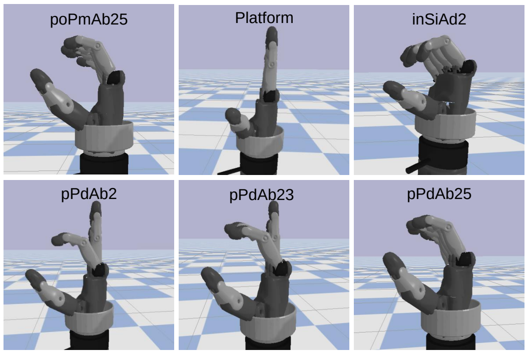

Figure 2. Grasp taxonomy with six grasp topology: the grasp topology names represent grasp attributes where "In," "po," and "p" indicate intermediate, power, and precision grasps. "Si," "Pm," and "Pd" refer to side, palm, and pad opposition. "Ab" and "Ad" signify abduction and adduction, and numbers define virtual finger groups.

The adopted taxonomy is illustrated in Fig. 2: (1) the platform grasp for holding, pushing, or pressing; (2) the power grasp (poPmAb25) for securely gripping objects; (3) precision grasps (pPdAb2, pPdAb23, pPdAb25) for tasks requiring fine dexterity; and (4) the intermediate grasp (InSiAd2) for levering or twisting actions.

The grasp selection network is a multi-class, multi-label Multilayer Perceptron (MLP) neural network that takes object features and task objectives as input and predicts the probability of each grasp topology in the predefined taxonomy. The topology with the highest probability is selected as the target grasp pose.

C. Execution of Grasping Tasks

We used a reinforcement learning approach for grasping task execution. Due to the inherent complexity of grasping, such tasks often exhibit limited flexibility and present optimization challenges. To reduce task complexity, we decomposed the grasping process into a sequence of manageable stages. Although grasping tasks can be decomposed in various ways, Fig. 3 shows a general decomposition framework. In the framework, the initial state represents the initial configuration of the task environment. In the approach stage, the robot hand moves to an appropriate position and orientation (grasp location) for grasping. During the grasping stage, the fingers adjust to establish a stable or adaptive grasp based on the task's requirements. Finally, the termination stages reflect task outcomes, such as grasp success, grasp failure, or the object being out of reach, and signify the completion of the task.

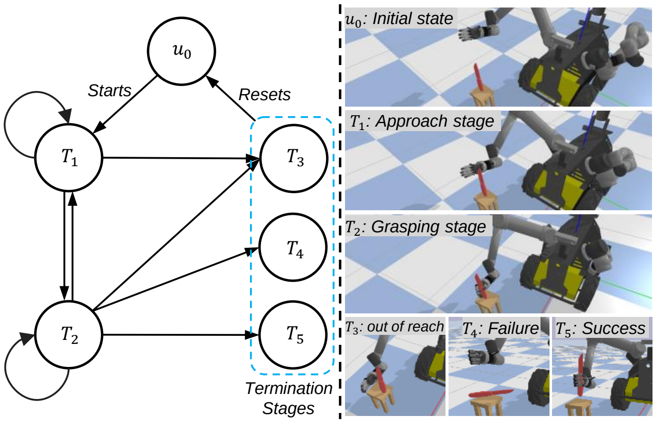

Figure 3. A general example of grasping task decomposition.

This method requires stage-specific learning mechanisms as each stage operates within a distinct context. To address this problem, we propose a Contextual Reward Machine that explicitly defines and manages these contexts.

III. Contextual Reward Machine

The CRM provides a structured and interpretable framework for addressing complex tasks by encoding task-specific knowledge into a hierarchical representation. This structure enables adaptive rewards based on task progress and stage transitions toward the desired goal. By guiding the agent through task-relevant stages, this approach improves learning efficiency and supports effective, goal-directed behaviors. Further details are provided in this section.

A. The General Framework of CRM

The general framework of the CRM extends the standard reward machine [23] by incorporating task context and a stage transition function, formally defined as

$$\mathcal{M} = (U, u_0, \Sigma, \delta, \mathcal{T}, R_{\mathrm{T}}) \tag{1}$$

where $\mathcal{T} = \{(A_i, r^{(i)}, \phi_i)\}_{i=1}^{k}$ represents a set of k stages (or sub-tasks), each characterized by task-specific knowledge. Each stage T_i consists of an action set $A_i \subseteq \mathcal{A}$, a state abstraction function $\phi_i : U \to U_i'$, which maps the global state space U to a stage-relevant abstract state space U_i' to simplify stage representation, and a stage reward function $r^{(i)} : U_i' \times A_i \to \mathbb{R}$ which defines the rewards for actions within the stage T_i.

The transition function $\delta : U \times \Sigma \to \mathcal{T}$ determines whether a stage transition occurs and, if so, to which stage, based on the current state and Σ, the set of events triggering

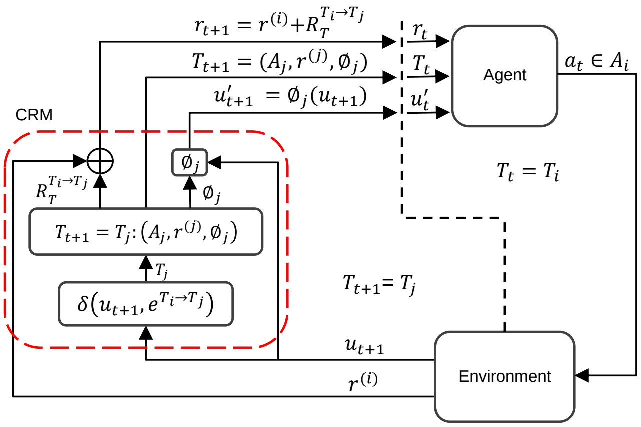
$r_{t+1} = r^{(i)}+R_T^{T_i \to T_j}$
$T_{t+1} = (A_j, r^{(j)}, \emptyset_j)$
$u'_{t+1} = \emptyset_j(u_{t+1})$
r_t
T_t
u'_t
Agent
$a_t \in A_i$
CRM
$\emptyset_j$
$R_T^{T_i \to T_j}$
$\emptyset_j$
$T_t = T_i$
$T_{t+1} = T_j: (A_j, r^{(j)}, \emptyset_j)$
T_j
$\delta(u_{t+1}, e^{T_i \to T_j})$
$T_{t+1} = T_j$
u_{t+1}
$r^{(i)}$
Environment

Figure 4. The Contextual Reward Machine: The dotted line separates the RL processes at timesteps t and $t+1$, illustrating the sequential interaction between the agent, environment, and the CRM.

stage transitions. A transition occurs when the current state satisfies one of these events. Upon a stage transition, the stage transition history set $\mathcal{H} \subseteq \mathcal{T} \times \mathcal{T}$ is updated as

$$\mathcal{H} = \{(T_i, T_j) \mid i, j = 1, 2, \ldots, k\} \tag{2}$$

where each transition $(T_i, T_j) \in \mathcal{H}$ is associated with a reward $R_{\mathrm{T}} : \mathcal{H} \to \mathbb{R}$, which quantifies the desirability of the transition.

The CRM within the RL framework operates iteratively and allows the agent to interact with the environment and adapt dynamically, as illustrated in Fig. 4. At timestep t, the agent operates within stage $T_t = T_i$. Based on the abstract state u'_t and action space A_i, the agent selects an action $a_t \in A_i$. The environment responds with the next global state u_{t+1} and an intra-stage reward $r^{(i)}(u'_t, a_t)$ which reflects the action's outcome.

The CRM processes the feedback to determine the next stage $T_{t+1} = T_j$ using the transition function $\delta(u_{t+1}, e^{T_i \to T_j})$, where $e^{T_i \to T_j} \in \Sigma$ represents the triggering event. If no transition occurs $(i = j)$, the global state u_{t+1} is abstracted to u'_{t+1} using ϕ_i. If a transition occurs $(i \neq j)$, the global state u_{t+1} is abstracted to the corresponding state u'_{t+1} of stage T_j using ϕ_j. These abstraction functions extract task-relevant features and simplify state representation.

The reward at timestep $t+1$ is computed as

$$r_{t+1} = r^{(i)}(u'_t, a_t) + R_{\mathrm{T}}^{T_i \to T_j} \tag{3}$$

and the cumulative reward is expressed as

$$R = \sum_{t=1}^{T} r_t(u'_t, a_t) + \sum_{(i,j) \in \mathcal{H}} R_{\mathrm{T}}^{T_i \to T_j} \tag{4}$$

where r_t represents the stage-specific reward at timestep $t \in [1, \ldots, T]$, and T is the total number of timesteps. The first term captures intra-stage rewards, while the second accounts for transition rewards. The stage knowledge in T_{t+1}, the abstract state u'_{t+1}, and the reward r_{t+1} define the context for the next timestep which supports efficient task decomposition and reward optimization for solving complex tasks.

B. PPO with CRM

To adapt PPO to the CRM framework, the objective function is revised to align with the hierarchical structure of CRM. For a stage T_i at timestep t, the clipped surrogate objective is

$$\begin{aligned} \mathcal{L}_{\mathcal{M}}^{\mathrm{CLIP}}(\theta) = \; & \mathbb{E}_{(u',a) \sim \pi_{\mathrm{old}}} \Big[\min \big(r_t^{\mathcal{M}}(\theta) A_t^{\mathcal{M}}, \\ & \mathrm{clip}(r_t^{\mathcal{M}}, 1-\epsilon, 1+\epsilon) A_t^{\mathcal{M}} \big) \Big] \end{aligned} \tag{5}$$

where $r_t^{\mathcal{M}}(\theta) = \frac{\pi_\theta(a_t|u'_t)}{\pi_{\mathrm{old}}(a_t|u'_t)}$ represents the probability ratio between the current and old policies based on the abstract state $u'_t = \phi_i(u_t)$.

The advantage function $A_t^{\mathcal{M}}$ is defined as

$$A_t^{\mathcal{M}} = r_t(u'_t, a_t) + \gamma V(u'_{t+1}) + R_{\mathrm{T}}^{T_i \to T_j} - V(u'_t) \tag{6}$$

where $r_t(u'_t, a_t) = r^{(i)}(u'_t, a_t)$ is the intra-stage reward, $R_{\mathrm{T}}^{T_i \to T_j}$ is the transition reward, and $V(u'_t)$ and $V(u'_{t+1})$ are value estimates for the current and next stages, respectively.

IV. CRM-PPO for Grasping Tasks

The CRM-PPO model divides grasping tasks into distinct stages, each defined by a specific context and transition mechanism. To optimize task performance, it is crucial to clearly specify these contexts and mechanisms for each stage. This section outlines the stage contexts and corresponding transition mechanisms for each stage of the grasping task.

A. Task Decomposition

We followed the task decomposition strategy shown in Fig. 3 to decompose grasping tasks using the general CRM framework (Eq. 1). The global state U is defined as the set of all possible states in the grasping environment, while $u_{\mathrm{initial}} = u_0$ represents the initial state corresponding to the environment's default configuration. Each grasping task begins from the initial state u_0, and upon task completion, the environment resets to u_0 to prepare for the next task.

The system transitions directly from the initial state to the approach stage without receiving any reward. During the approach stage, the robot moves its hand toward the object. If the object becomes out of reach in this stage (e^{aor}), the task transitions to the out-of-reach stage with a penalty $R_{\mathrm{T}}^{\mathrm{aor}}$. Arriving at the grasp location (e^{arrive}) transitions the task to the grasping stage, with a transition reward $R_{\mathrm{T}}^{\mathrm{arrive}}$. The approach stage cannot transition directly to the grasp-failure or grasp-success stages, as a failed or successful grasp requires hand-object interaction, which occurs only in the grasping stage.

In the grasping stage, the robot manipulates its fingers to attempt a grasp. Knocking the object out of reach in the grasping stage (e^{gor}) results in a transition to the out-of-reach stage with a penalty $R_{\mathrm{T}}^{\mathrm{gor}}$. Failure to grasp the object (e^{fail}) results in a transition to the grasp-failure stage with a penalty $R_{\mathrm{T}}^{\mathrm{fail}}$. Successfully grasping the object (e^{succ}) leads to the grasp-success stage with a reward $R_{\mathrm{T}}^{\mathrm{succ}}$.

The out-of-reach, grasp-success, and grasp-failure stages serve as termination stages, which conclude the current task and reset the environment to the initial state u_0 in preparation for the next one.

B. Action Space

The designed stages of a task-oriented grasping include the approach stage, the grasping stage, and the termination stages.

1) The Approach Stage: In the approach stage, the objective is to move the hand toward the grasp location while avoiding collisions with the object. Finger movements are disabled in the approach stage to reduce collision risks and improve sample efficiency. The action space is defined as

$$A_{\text{approach}} = [\Delta x, \Delta y, \Delta z]$$

where Δx, Δy, and Δz represent incremental changes along the x, y, and z axes, respectively.

2) The Grasping Stage: The action space for the grasping stage is defined as

$$A_{\text{grasp}} = [\Delta x, \Delta y, \Delta z, \theta_{\text{thumb}}, \theta_{\text{index}}, \theta_{\text{middle}}, \theta_{\text{ring}}, \theta_{\text{little}}]$$

where θ_{thumb}, θ_{index}, θ_{middle}, θ_{ring}, and θ_{little} represent the proximal interphalangeal (PIP) joint angles of the thumb, index, middle, ring, and little fingers, respectively. In the grasping stage, fine adjustments to the hand position are crucial for achieving an optimal grasp, even when the hand is close to the grasp location. To enable these adjustments, the hand's movements remain constrained by Δx, Δy, and Δz, but with reduced step sizes for finer control.

To enhance grasp quality, we developed a simplified hand model inspired by human hand kinematics and anatomy. The model captures key biomechanical features such as joint articulation and finger linkage and enables more natural and effective grasping strategies. This model focuses on finger flexion and extension while excluding finger spreading. Joint angles are constrained based on the PIP joint angle, θ_{PIP}, with the following relationships $\theta_{\text{DIP}} = \alpha_{\text{DIP}} \cdot \theta_{\text{PIP}}$ and $\theta_{\text{MCP}} = \alpha_{\text{MCP}} \cdot \theta_{\text{PIP}}$ where θ_{DIP} and θ_{MCP} represent the joint angles of the distal interphalangeal (DIP) joint and the metacarpophalangeal (MCP) joint, respectively. The constants α_{DIP} and α_{MCP} define proportional joint coordination. For the thumb, the relationship between the interphalangeal (IP) joint angle θ_{IP} and the trapeziometacarpal (TMCP) joint angle θ_{TMCP} is given by $\theta_{\text{IP}} = \alpha_{\text{TMCP}} \cdot \theta_{\text{TMCP}}$. Here, the constant α_{TMCP} defines the proportional coupling between the two joints, it reflects the biomechanical constraints of human thumb movement. The corresponding values for α_{DIP} are 0.77, 0.75, 0.75, and 0.57 for the index, middle, ring, and little fingers, respectively. The value of α_{MCP} is 0.67 for all fingers, while $\alpha_{\text{TMCP}} = 0.5$, as reported in [24]–[26].

The out-of-reach, grasp-success, and grasp-failure stages are termination stages and therefore have no associated action sets.

C. Observation Space

The observation space is defined as

$$\mathcal{O} = [n_{\text{c}}, o_{\text{dist}}, \boldsymbol{o}_{\text{object}}, o_{\text{cone}}, \boldsymbol{o}_{\text{relative}}, \boldsymbol{o}_{\text{force}}, \boldsymbol{o}_{\text{torque}}] \tag{7}$$

where each component represents a critical aspect of the grasping task. The variable n_{c} indicates the number of contact points between the robot hand and the object which reflects contact extent and contributes to grasp stability. The distance o_{dist} measures the proximity of the robot hand to the grasp location. The vector $\boldsymbol{o}_{\text{object}}$ represents the object position, which supports object out-of-range detection and task success or failure evaluation.

The vector $\boldsymbol{o}_{\text{relative}}$ describes the spatial relationship between the robot hand and the object which provides essential grasp configuration details. The vectors $\boldsymbol{o}_{\text{force}}$ and $\boldsymbol{o}_{\text{torque}}$ represent the summed magnitudes of contact forces and torques at all contact points along the x, y, and z axes. They enable the evaluation of force and torque equilibrium. Together, these components comprehensively describe the grasping task which covers grasp position, configuration, and stability.

The Boolean variable o_{cone} indicates whether all contact forces lie within the friction cone and ensures stability through frictional constraints. The friction cone is defined by the coefficient of friction μ and the normal force $\mathbf{F}_n$, which satisfies the condition: $\|\mathbf{F}_c\| \leq \mu \cdot \mathbf{F}_n$ where $\mathbf{F}_c$ is the contact force vector. Grasp stability is determined as $o_{\text{cone}} = 1$ if the above condition is satisfied for all contact points, and $o_{\text{cone}} = 0$ otherwise. This ensures that all contact forces remain within the friction cone which prevents slippage and enhances grasp stability.

Each stage of the grasping task has specific goals and therefore requires specific information. The state abstraction function ϕ_i extracts the task-relevant information from the observation space which reduces computational complexity and improves efficiency.

D. Reward Function

1) The Approach Stage: The objective of this stage is to move the robot hand as close as possible to the grasp location while avoiding collisions and ensuring the object remains within the workspace. The reward function for this stage is defined as

$$r_{\text{appr}} = r^{\text{dist}} - \rho_{\text{appr}} n_{\text{c}} \tag{8}$$

where ρ_{appr} is a constant coefficient, and the distance-based reward is defined as $r^{\text{dist}} = -\mathrm{e}^{|o_{\text{dist}}|}$, which is inversely proportional to the distance between the hand and the grasp location. The exponential form ensures rapid changes when the hand is far from the target and more gradual changes as it approaches the object. It encourages larger adjustments at greater distances and finer movements when closer. This enhances both the efficiency and accuracy of the task.

To reduce the risk of unintended collisions, which could cause the object to be knocked out of the workspace, a penalty proportional to the number of contact points n_{c} is applied. This reward design encourages precise and collision-free hand movements during the approach stage.

2) The Grasping Stage: The reward function for the grasping stage is designed to encourage stable and efficient grasping behaviors. The model is rewarded based on the number of contact points n_{c}, as a higher number of contact points leads to increased grasp stability. Additionally, the reward function evaluates equilibrium by minimizing net forces $\boldsymbol{o}_{\text{force}}$ and torques $\boldsymbol{o}_{\text{torque}}$ along the three axes at all contact points. The closer these values are to zero, the higher the reward, which indicates a more stable and secure grip. The reward function for the grasping stage is defined as

$$r_{\text{grasp}} = r^{\text{equil}} + \rho_{\text{grasp}} n_{\text{c}} + R_{\text{T}}^{\text{arrive}} \tag{9}$$

where the equilibrium reward is expressed as

$$r^{\text{equil}} = -\mathrm{e}^{|\boldsymbol{o}_{\text{force}}|} - \mathrm{e}^{|\boldsymbol{o}_{\text{torque}}|} \tag{10}$$

The coefficient ρ_{grasp} adjusts the reward's sensitivity to the number of contact points. The stage transition reward $R_{\text{T}}^{\text{arrive}} = R_{\text{T}}^{T_1 \to T_2}$ encourages the transition from the approach stage to the grasping stage. Its value is set higher than the maximum achievable reward in the approach stage r_{appr} to ensure that transitioning to the grasping stage is prioritized, while still maintaining a stable and controlled grip.

3) The Termination Stages: The out-of-reach, the grasp-success, and the grasp-failure stages monitor task outcomes, where the reward is only related to the position of the object $\boldsymbol{o}_{\text{object}}$. The reward function for the out-of-reach stage is defined as

$$r_{\text{oor}} = R_{\text{T}}^{\text{aor}} \cdot \mathbf{1}_{\{e^{\text{aor}}\}} + R_{\text{T}}^{\text{gor}} \cdot \mathbf{1}_{\{e^{\text{gor}}\}} \tag{11}$$

where both transition rewards are negative penalties. The condition $R_{\text{T}}^{\text{aor}} = R_{\text{T}}^{T_1 \to T_3} < R_{\text{T}}^{\text{gor}} = R_{\text{T}}^{T_2 \to T_3}$ reflects greater task progress when transitioning from the grasping stage compared to the approach stage. The indicator function $\mathbf{1}_{\{e\}}$ equals 1 if the event e occurs and 0 otherwise. It ensures the right penalty is applied only when specific transitions occur. Here, $e^{\text{aor}} = e^{T_1 \to T_3}$ and $e^{\text{gor}} = e^{T_2 \to T_3}$.

The reward function for the grasp-failure stage is

$$r_{\text{fail}} = R_{\text{T}}^{\text{fail}} \tag{12}$$

where $R_{\text{T}}^{\text{fail}} = R_{\text{T}}^{T_2 \to T_4}$ is a smaller penalty compared to the out-of-reach stage, as it acknowledges partial task completion.

In addition to monitoring task outcomes, the grasp-success stage evaluates grasp quality through a friction cone analysis. The reward function of this stage is defined as

$$r_{\text{succ}} = R_{\text{T}}^{\text{succ}} + R_{\text{cone}} \cdot \mathbf{1}_{\{o_{\text{cone}}\}} \tag{13}$$

where $R_{\text{T}}^{\text{succ}} = R_{\text{T}}^{T_2 \to T_5}$ is a large reward for successfully transitioning to this stage, and R_{cone} is an additional reward given only when all contact points satisfy the friction cone condition. This reward design encourages both task completion and a stable grasp.

V. Experiments

The proposed method is evaluated in a simulated environment and compared with the state-of-the-art methods. To validate its real-world applicability, the method is transferred to a physical robot for performance testing. The evaluation results are analyzed and discussed in this section.

A. Experiment Setup

1) Environment Setup: The task environment for the grasping task is illustrated in Fig. 5. The target object is placed on a table in front of a dual-arm mobile robot. This robot comprises a Husky UGV (Unmanned Ground Vehicle) for mobility, two UR5e robotic arms, a Schunk SVH robotic hand (right), and a PSYONIC Ability Hand (left). Each robotic hand is mounted on a UR5e arm, and both arms are attached to the Husky UGV. This configuration enables coordinated manipulation.

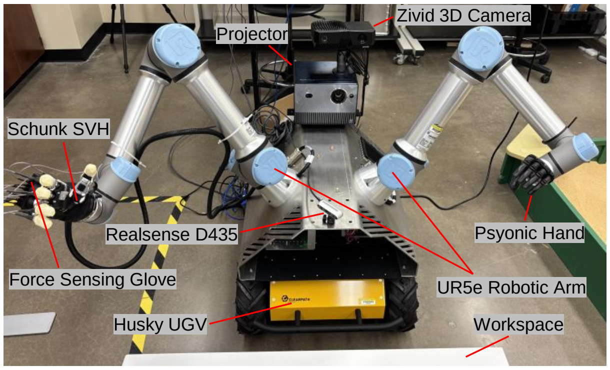

Figure 5. Real-world experiment setup for grasping tasks.

The robot integrates various sensors to enhance perception and interaction. The Schunk SVH hand is equipped with an ErgoGLOVE Force Sensing System for contact force detection, while the PSYONIC Ability Hand features built-in force sensors. A RealSense D435 depth camera provides visual input which enables precise object recognition, object pose estimation, and environmental awareness. Additionally, a Zivid One 3D camera and a projector are also part of the robot but are not utilized in this study.

A simulation environment was developed using PyBullet and OpenAI Gym to train and test the proposed model. It replicates the real-world setup of the robot, which enables seamless transfer of learned policies to the physical robot for performance evaluation and practical applications.

2) Dataset: The AffordPose dataset serves as a benchmark for robotic grasping tasks which emphasizes affordance-based pose estimation [27]. It provides 3D object models, annotated grasp poses, and corresponding affordance labels.

For this work, we refined the AffordPose dataset by removing redundant entries, classifying grasp poses based on the grasp taxonomy defined in Fig. 2, and computing grasp locations to support the proposed grasping task framework. The revised dataset comprises six unique grasp poses, seven distinct grasping objectives, and 20 diverse objects. As a result, the dataset contains a total of 26 different grasping tasks across various objects, purposes, and grasp configurations.

B. Performance Evaluation in the Simulation Environment

1) Evaluation Metrics: The proposed CRM-PPO model was evaluated against three baseline models using the benchmark dataset. The evaluation is divided into two parts: (1) performance assessment within the CRM framework and (2) comparison with state-of-the-art methods.

The first baseline, CRM-SAC, integrates the CRM framework with the Soft Actor-Critic (SAC) algorithm. This setup enables a direct comparison with CRM-PPO under identical conditions, which demonstrates the superior performance of PPO within the CRM framework. The second and third

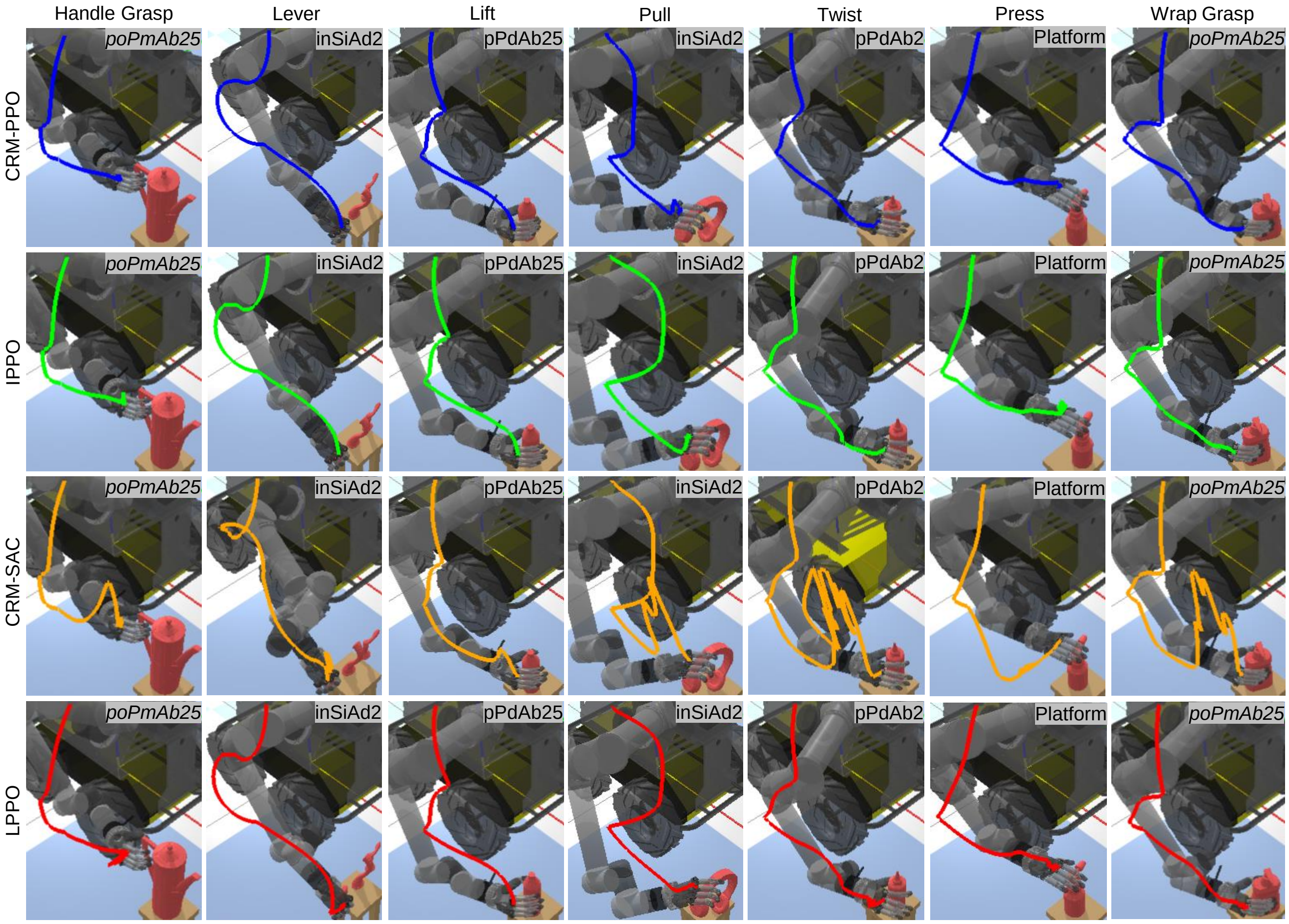

Figure 6. Trajectory visualization of grasping tasks performed by the proposed model and baseline models across different task objectives and grasp topologies.

baselines, LPPO (Learning-based PPO) and IPPO (Improved PPO), represent state-of-the-art approaches for grasping tasks based on stage-wise mechanisms. LPPO employs hierarchical dense rewards to enhance training efficiency and generalization across diverse object configurations [28]. IPPO employs a stage-wise sparse reward structure, which improves convergence speed and grasping accuracy [29]. The comparison with these baselines confirms the performance advantages introduced by the CRM framework.

The evaluation metrics include task success rate and average task completion time in timesteps.

2) Training Setup: The proposed model[1] is trained for approximately 12,000 episodes (equivalent to 5 million timesteps) using a discount factor of 0.99, a GAE lambda of 0.95, and a batch size of 64. A dynamic learning rate is applied, starting at 3×10^{-5} for the first 40% of training progress. Between 40% and 70% progress, the learning rate is reduced to 90% of its initial value. During the final 30% of training, it is further reduced to 80%.

During training, the robot hand performs grasping tasks with different task objectives and grasp topologies on a variety of objects. To simulate real-world uncertainties, random noise of $\pm 3\,\mathrm{mm}$ in object position, $\pm 11.5°$ in object orientation, and $0.02\,\mathrm{rad}$ in joint positions was applied. This domain randomization method enhances the model's robustness and generalization to real-world scenarios by introducing randomized variations in object pose and joint configurations during training.

For task objectives such as handle grasp, lift, lever, pull, and wrap grasp, a task is considered successful if the robot picks up the object and holds it steadily for 5 seconds without dropping it. For the twist objective, success is defined as applying sufficient torque in the twisting direction after grasping the object. In the press objective, success is achieved if the robot applies enough force in the pressing direction.

An early stopping mechanism is implemented to improve training efficiency. The training process is terminated when the average success rate of the most recent 100 episodes reaches or exceeds 99%.

Examples of trajectories generated by the proposed and baseline methods are shown in Fig. 6. The figure illustrates that the trajectory generated by the PPO-based method is smoother and more optimized compared to that of the SAC-based method. This difference may be due to SAC's lower efficiency in tasks with well-shaped rewards.

For the PPO-based method, both the proposed and baseline approaches exhibit similar trajectories. The proposed method completes the trajectory significantly faster and avoids over-

[1] This model is available at https://github.com/hhelium/DexMobile

Table I
THE TESTING RESULTS FOR THE PROPOSED AND BASELINE MODELS.

Models	**IPPO** [29]		**LPPO** [28]		**CRM-SAC**		**CRM-PPO (this paper)**	
	Success Rate	Episode Length	Success Rate	Episode Length	Success Rate	Episode Length	Success Rate	Episode Length
Lift	0.85	412.63	0.88	456.33	0.63	534.68	**0.88**	**342.43**
Pull	0.86	433.60	0.76	315.82	**1.0**	276.22	**1.0**	**217.34**
Press	0.76	415.82	0.29	793.22	0.33	787.64	**0.96**	**167.30**
Twist	0.74	436.20	0.67	531.34	0.49	599.17	**0.91**	**335.52**
Lever	0.70	466.80	0.90	367.67	**0.95**	307.24	0.93	**246.15**
Wrap-Grasp	0.89	348.88	0.67	372.49	0.60	377.41	**1.0**	**252.17**
Handle-Grasp	**0.98**	303.70	0.97	369.71	0.75	338.43	0.97	**281.09**
Overall	0.84	390.86	0.71	461.75	0.61	479.36	**0.95**	**273.07**

lapping or revisiting previous paths. This improvement is due to the stage-specific context and transition rewards, which effectively guide the model to complete the task efficiently and without redundancy.

3) Result Analysis: The proposed CRM-PPO model and the three baselines were trained five times each. For each model, the average success rate and average episode length were calculated as evaluation metrics, along with their standard deviations to reflect performance variability across runs. The training results are presented in Fig. 7, where solid curves represent average values and shaded areas indicate standard deviations.

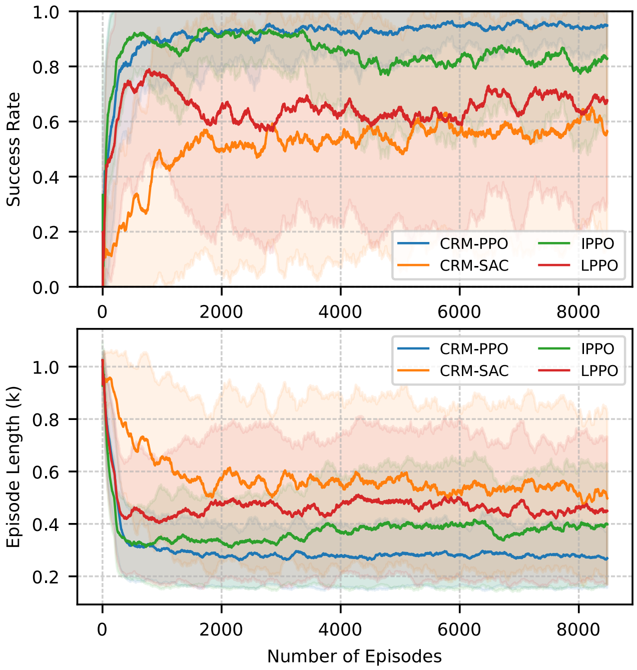

Figure 7. Comparison of different models in grasping tasks based on success rate and episode length.

The proposed CRM-PPO model outperforms all baselines across all evaluation metrics. It records the highest average success rate and the shortest average episode length. The model reaches the early termination criterion after approximately 8,200 episodes on average, while none of the baseline models meet this criterion within the full training period. All five runs of CRM-PPO reach early termination and show the lowest standard deviation. These results reflect superior robustness, consistency, and sample efficiency. This strong performance results from its hierarchical reward structure, which provides clear stage-specific guidance and supports effective transitions between stages through transition rewards.

The performance of IPPO surpasses that of LPPO, primarily due to differences in reward design. IPPO assigns sparse rewards based on stage transitions, while LPPO defines continuous intra-stage rewards without explicitly encouraging transitions. In summary, IPPO provides only transition rewards, whereas LPPO offers only intra-stage rewards. IPPO outperforms LPPO because its sparse reward structure effectively guides the agent toward target stages. In contrast, LPPO focuses more on intra-stage optimization and lacks incentives for transitioning between stages, which results in lower overall performance despite its intra-stage guidance.

During early training, IPPO's performance approximates that of the proposed CRM-PPO model, as its sparse reward structure provides implicit transition-based guidance. Due to the absence of intra-stage rewards, IPPO does not support fine-grained exploration and adaptation, which reduces data efficiency. As training progresses, this limitation causes IPPO's performance to lag behind CRM-PPO.

The CRM-PPO model effectively integrates both approaches by combining stage-specific guidance and transition rewards, and it outperforms the baseline models across evaluation metrics. CRM-SAC performs the worst across all evaluation metrics, which indicates that PPO-based methods are better suited for hierarchical grasping tasks.

The proposed model and baseline models were evaluated using a benchmark dataset, with each model tested 1,000 times on randomly selected tasks. The results are summarized in Table I. The proposed model consistently outperformed baseline models in most tasks. Notably, it excelled in the challenging Twist task, where baseline models showed relatively low success rates. This task demands precise and coordinated actions, it demonstrates CRM-PPO's ability to handle complex scenarios due to its task-specific structured design, which provides effective execution guidance.

In the Lever and Handle-Grasp tasks, the proposed model achieved slightly lower success rates, trailing the best-performing baseline by 2% and 1%, respectively. Despite this, it surpassed all baselines in episode length and completed tasks more efficiently. These results confirm the effectiveness of the

CRM-PPO model in task-oriented grasping tasks.

C. Real-World Evaluation

The robots are integrated through ROS2 Humble on Ubuntu 22.04 with a low-latency kernel to support real-time performance. Inverse kinematics and collision avoidance are managed through MoveIt2. An Intel RealSense D435 depth camera is spatially aligned with the robot's coordinate frame via an eye-to-hand calibration, which enables accurate perception and interaction within the shared workspace. The depth images align with the color images, both with a resolution of 640 × 480 pixels. The image streams and force feedback data from the ErgoGLOVE Force Sensing System are synchronized and published to the proposed model through ROS2 topics. This setup ensures coherent sensory input for perception and interaction tasks.

Even though the simulation and the real robot are identical in design, a sim-to-real gap persists due to discrepancies in dynamic properties, environmental conditions, and sensor noise. To bridge this gap, domain randomization is employed [30]. The policy is trained across numerous variations of simulation parameters, such as joint position error, object pose error, and sensor noise. By randomizing these parameters over a wide range, the likelihood of the policy generalizing to real-world conditions increases significantly.

The robot perceives the object's pose and the contact forces between the Schunk hand and the object. The object pose is estimated using FoundationPose [31], which provides millimeter-level accuracy in pose estimation. Based on the object pose, the observations o_{dist}, $\boldsymbol{o}_{\text{object}}$, and $\boldsymbol{o}_{\text{relative}}$ can be calculated. The ErgoGLOVE Force Sensing System detects the contact force between the Schunk hand and the object and provides n_{c}. Since the force-sensing glove measures force along only one axis, $\boldsymbol{o}_{\text{force}}$, $\boldsymbol{o}_{\text{torque}}$, and o_{cone} cannot be detected and are set to zero.

In the simulation environment, the grasping force cannot be determined properly due to the agnostic nature of the target object's material properties, as it is represented by a 3D model. As a result, the grasping force cannot be accurately calibrated, and the object remains unaffected by any applied force. In contrast, in the real world, the contact force must be carefully controlled to avoid damaging the object or the robotic hand. To address this discrepancy and enable the transfer of the simulation model to a physical robot, we manually defined grasping force thresholds. For fragile objects, the contact force was constrained to a maximum of 1 N, with the robotic fingers halting motion upon reaching this threshold. For sturdy objects, the threshold was set between 1 N and 3 N, depending on factors such as the object's texture and weight.

The proposed CRM-PPO model is fine-tuned for 300 episodes across six affordances, including Handle Grasp, Lift, Press, Pull, Twist, and Wrap Grasp, each utilizing the corresponding grasp topology, as shown in Fig. 8. The achieved testing success rates for these tasks are 100%, 90%, 60%, 80%, 70%, and 100%, respectively. In the experiment, the grasping tasks for press and twist exhibited lower success rates of 60% and 70%, respectively. The low success rate for the twist task is attributed to the small, round-shaped handle of the screwdriver, which is prone to slipping when grasped using the topology inSiAd2. Additionally, the screwdriver, when placed on the table, can be easily knocked over with even light contact from the hand. The press task had the lowest success rate due to the limitations of the sensing glove. The eight force sensors on the glove cover only a limited area of the hand, and during pressing, the actual contact area often lacks sensor coverage, which can result in task failure. After sensor repositioning, the success rate improved significantly, while success rates for other tasks decreased. The overall success rate achieved was 83.3%, which can be further improved by adding more force sensors and further fine-tuning.

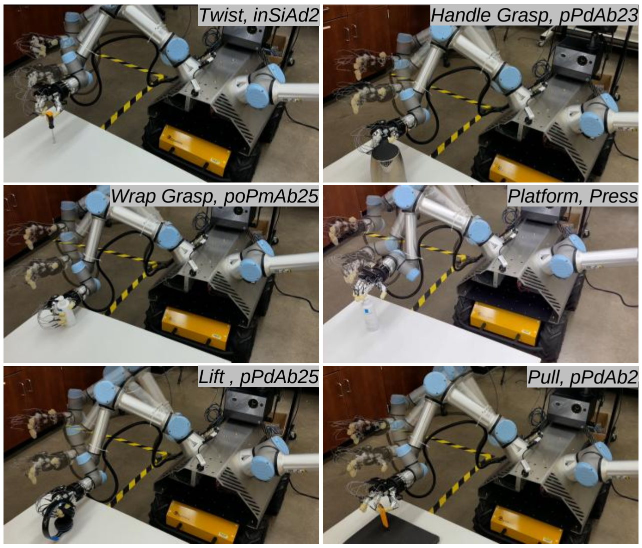

Figure 8. Real-world grasping tasks across different affordances and grasp topologies.

VI. Conclusion

This paper presents a context-aware task-oriented dexterous grasping approach leveraging a Contextual Reward Machine framework. The CRM decomposes complex grasping tasks into modular sub-tasks with stage-specific contexts, which enables efficient learning and execution. By integrating Proximal Policy Optimization, the proposed method achieves significant improvements in learning efficiency, task performance, and adaptability. Extensive experiments in simulated and real-world environments validated the effectiveness and robustness of the proposed approach. The CRM-PPO model achieved a 95% success rate in simulation across 1,000 grasping tasks. When transferred to a real robot, it attained an 83.3% success rate over 60 real-world tasks. The proposed model exceeded the performance of state-of-the-art models in success rate and task completion time. Its ability to adapt to diverse grasping objectives, various grasp topologies, and dynamic conditions highlights its practical applicability in unstructured environments. The results demonstrate the potential of the CRM-PPO framework to advance robotic dexterity and manipulation.

References

[1] Y. Fan, L. Sun, M. Zheng, W. Gao, and M. Tomizuka, “Robust dexterous manipulation under object dynamics uncertainties,” in *2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE, 2017, pp. 613–619. 2

[2] J. W. James and N. F. Lepora, “Slip detection for grasp stabilization with a multifingered tactile robot hand,” *IEEE Transactions on Robotics*, vol. 37, no. 2, pp. 506–519, 2020. 2

[3] R. Wang, J. Zhang, J. Chen, Y. Xu, P. Li, T. Liu, and H. Wang, “Dexgraspnet: A large-scale robotic dexterous grasp dataset for general objects based on simulation,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 11 359–11 366. 2

[4] P. Mandikal and K. Grauman, “Dexvip: Learning dexterous grasping with human hand pose priors from video,” in *Conference on Robot Learning*. PMLR, 2022, pp. 651–661. 2

[5] R. Deimel and O. Brock, “A novel type of compliant and underactuated robotic hand for dexterous grasping,” *The International Journal of Robotics Research*, vol. 35, no. 1-3, pp. 161–185, 2016. 2

[6] H. Duan, P. Wang, Y. Huang, G. Xu, W. Wei, and X. Shen, “Robotics dexterous grasping: The methods based on point cloud and deep learning,” *Frontiers in Neurorobotics*, vol. 15, p. 658280, 2021. 2

[7] W. Shang, F. Song, Z. Zhao, H. Gao, S. Cong, and Z. Li, “Deep learning method for grasping novel objects using dexterous hands,” *IEEE Transactions on Cybernetics*, vol. 52, no. 5, pp. 2750–2762, 2020. 2

[8] T. Feix, J. Romero, H.-B. Schmiedmayer, A. M. Dollar, and D. Kragic, “The grasp taxonomy of human grasp types,” *IEEE Transactions on Human-Machine Systems*, vol. 46, no. 1, pp. 66–77, 2016. 2

[9] M. Santello, M. Flanders, and J. F. Soechting, “Postural hand synergies for tool use,” *Journal of Neuroscience*, vol. 18, no. 23, pp. 10 105–10 115, 1998. 2

[10] M. R. Cutkosky *et al.*, “On grasp choice, grasp models, and the design of hands for manufacturing tasks.” *IEEE Transactions on robotics and automation*, vol. 5, no. 3, pp. 269–279, 1989. 2

[11] I. M. Bullock, T. Feix, and A. M. Dollar, “The yale human grasping dataset: Grasp, object, and task data in household and machine shop environments,” *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 251–255, 2015. 2

[12] A. B. Rao, H. Li, and H. He, “Object recall from natural-language descriptions for autonomous robotic grasping,” in *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2019, pp. 1368–1373. 2

[13] T. Feix, I. M. Bullock, and A. M. Dollar, “Analysis of human grasping behavior: Correlating tasks, objects and grasps,” *IEEE transactions on haptics*, vol. 7, no. 4, pp. 430–441, 2014. 2

[14] H. Li, Y. Zhang, Y. Li, and H. He, “Learning task-oriented dexterous grasping from human knowledge,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 6192–6198. 2, 4

[15] R. S. Sutton, “Reinforcement learning: An introduction,” *A Bradford Book*, 2018. 2

[16] Y. Qin, B. Huang, Z.-H. Yin, H. Su, and X. Wang, “Dexpoint: Generalizable point cloud reinforcement learning for sim-to-real dexterous manipulation,” in *Conference on Robot Learning*. PMLR, 2023, pp. 594–605. 2

[17] P. Mandikal and K. Grauman, “Learning dexterous grasping with object-centric visual affordances,” in *2021 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2021, pp. 6169–6176. 2

[18] T. G. Thuruthel, E. Falotico, F. Renda, and C. Laschi, “Model-based reinforcement learning for closed-loop dynamic control of soft robotic manipulators,” *IEEE Transactions on Robotics*, vol. 35, no. 1, pp. 124–134, 2018. 3

[19] F. Stulp, E. A. Theodorou, and S. Schaal, “Reinforcement learning with sequences of motion primitives for robust manipulation,” *IEEE Transactions on robotics*, vol. 28, no. 6, pp. 1360–1370, 2012. 3

[20] W. He, H. Gao, C. Zhou, C. Yang, and Z. Li, “Reinforcement learning control of a flexible two-link manipulator: an experimental investigation,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 12, pp. 7326–7336, 2020. 3

[21] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” *arXiv preprint arXiv:1707.06347*, 2017. 3

[22] Y. Yang, “A deep reinforcement learning architecture for multi-stage optimal control,” *arXiv preprint arXiv:1911.10684*, 2019. 3

[23] R. T. Icarte, T. Q. Klassen, R. Valenzano, and S. A. McIlraith, “Reward machines: Exploiting reward function structure in reinforcement learning,” *Journal of Artificial Intelligence Research*, vol. 73, pp. 173–208, 2022. 3, 4

[24] M. Mentzel, A. Benlic, N. Wachter, D. Gulkin, S. Bauknecht, and J. Gülke, “The dynamics of motion sequences of the finger joints during fist closure,” *Handchirurgie, Mikrochirurgie, Plastische Chirurgie: Organ der Deutschsprachigen Arbeitsgemeinschaft fur Handchirurgie: Organ der Deutschsprachigen Arbeitsgemeinschaft fur Mikrochirurgie der Peripheren Nerven und Gefasse: Organ der V...*, vol. 43, no. 3, pp. 147–154, 2011. 6

[25] A. Roda-Sales, J. L. Sancho-Bru, and M. Vergara, “Studying kinematic linkage of finger joints: estimation of kinematics of distal interphalangeal joints during manipulation,” *PeerJ*, vol. 10, p. e14051, 2022. 6

[26] C.-E. Hrabia, K. Wolf, and M. Wilhelm, “Whole hand modeling using 8 wearable sensors: Biomechanics for hand pose prediction,” in *Proceedings of the 4th Augmented Human International Conference*, 2013, pp. 21–28. 6

[27] J. Jian, X. Liu, M. Li, R. Hu, and J. Liu, “Affordpose: A large-scale dataset of hand-object interactions with affordance-driven hand pose,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 14 713–14 724. 7

[28] A. A. Shahid, L. Roveda, D. Piga, and F. Braghin, “Learning continuous control actions for robotic grasping with reinforcement learning,” in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2020, pp. 4066–4072. 8, 9

[29] Z. Zhang, “Simulation of robotic arm grasping control based on proximal policy optimization algorithm,” in *Journal of Physics: Conference Series*, vol. 2203, no. 1. IOP Publishing, 2022, p. 012065. 8, 9

[30] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, “Domain randomization for transferring deep neural networks from simulation to the real world,” in *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2017, pp. 23–30. 10

[31] B. Wen, W. Yang, J. Kautz, and S. Birchfield, “Foundationpose: Unified 6d pose estimation and tracking of novel objects,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 17 868–17 879. 10