2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Safe multi-agent reinforcement learning for bimanual dexterous manipulation

Weishu Zhan and Peter Chin

Abstract— **Bimanual dexterous manipulation in robotics, essential for a wide range of applications, addresses the critical challenge of balancing intricate operational capabilities with assured safety and reliability. While Safe Reinforcement Learning is integral to the robustness of robotic systems, the area of safe multi-agent reinforcement learning (MARL), cooperative control of multiple robots has been scarcely studied. In this study, we explore MARL for safe cooperative control with multiple robot hands. Each robot must follow individual and collective safety guidelines to ensure safe team actions. However, the non-stationarity inherent in current algorithms hinders the precise updating of strategies to satisfy these safety constraints effectively. In this paper, we propose Multi-Agent Constrained Proximal Advantage Optimization (MACPAO), which considers the sequence of agent updates and integrates non-stationarity into sequential update schemes. This algorithm ensures consistent improvement in both rewards and adherence to safety constraints in each iteration. We tested MACPAO on various tasks with safety constraints and demonstrated that it outperforms other MARL algorithms in balancing reward enhancement and safety compliance. Supplementary materials and code are available at the provided link https://github.com/YONEX4090/MultiSafeHand.git.**

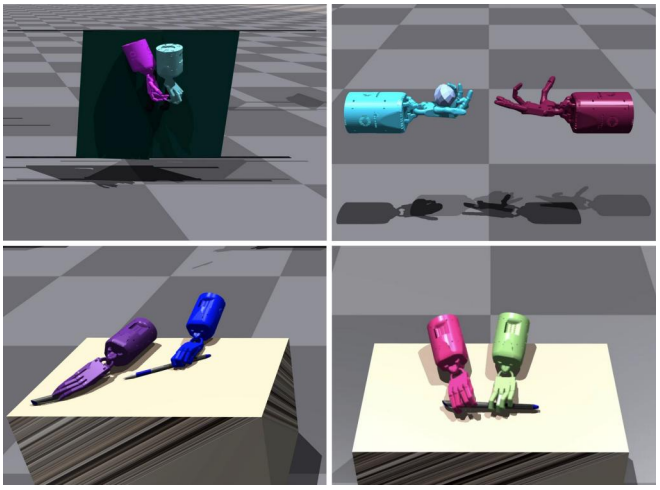

Fig. 1. We utilize simulation-trained Safe MARL methods to ensure the safe execution of diverse bimanual dexterous manipulation tasks, such as Hand Over, Close the Door, and Open/Close Pen Cap.

I. Introduction

Bimanual dexterous manipulation epitomizes human skill and sets a high standard for robots in sectors like healthcare and manufacturing [1]–[5]. The versatility of human hands sets a precedent for robotic designs [3]. However, current approaches typically separate dexterity from multi-agent systems, resulting in robots that lack agility, bound by challenges such as complex state spaces and detailed object-hand relations [6]. In MARL, it is critical to ensure safety by having agents strictly follow safety protocols while they optimize rewards in complex scenarios [7], [8].

Trust region learning methods have achieved notable success in reinforcement learning [9], particularly in solving complex tasks from single-agent control [10] to multi-agent applications [11]. These methods offer theoretical guarantees for consistent policy improvement, showing both superior and stable results. MARL advances beyond single-agent algorithms in terms of efficiency and effectiveness by enabling multiple agents to learn and coordinate their actions towards a shared goal. However, the implementation of most MARL algorithms poses significant challenges. These algorithms typically update all agents' policies simultaneously, preventing the observation of changes in other agents' policies and leading to non-stationarity problems in the environment [12]. Furthermore, they focus on optimizing policies for reward maximization, overlooking safety considerations. This oversight becomes particularly problematic in multi-agent environments, where ensuring safety is more complex due to each agent having to consider not only its own safety constraints but also those of others, to ensure joint behavior safety. Current solutions for safe multi-robot control in the context of effective learning algorithms are still limited [13]–[15].

In this paper, we present a series of steps to address the above problems. Firstly, we investigated the safety problem in bimanual dexterous control from the perspective of Safe MARL, addressing them through multi-agent policy optimization methods. So, we propose the Multi-Agent Constrained Proximal Advantage Optimization (MACPAO) method, which successfully extends constrained policy optimization techniques [14] to multi-agent settings, sequentially updating the policies of each agent. This method ensures monotonic improvements guarantee both the joint policies and individual agent policies. Notably, we applied clipping techniques inspired by Proximal Policy Optimization (PPO) [16] to optimize both reward and cost objectives, updating policies using data sampled from PPO, thus allowing our MACPAO algorithm to effectively overcome trust region constraints. Furthermore, we evaluated our method through a set of four dexterous manipulation tasks created in the Safe Multi-Agent Isaac Gym Benchmark (Safe MAIG) [17]. On dexterous manipulation tasks, MACPAO consistently

W. Zhan is with the Thayer School of Engineering, Dartmouth College, Hanover, NH 03755 (e-mail: weishu.zhan.th@dartmouth.edu).

P. Chin is with Thayer School of Engineering, Dartmouth College, Hanover, NH 03755 (e-mail: peter.chin@dartmouth.edu).

outperforms strong baselines in terms of performance and cost control, and demonstrated advantages in facilitating coordination among agents. The main contributions of our work are as follows:

- We propose the MACPAO algorithm, a novel approach to Multi-Agent Constrained Proximal Advantage Optimization, for solving Constrained Markov Decision Processes (CMDPs). It is the first safe sequential per-agent update algorithm that maintains a guarantee of monotonic policy improvement on both individual and joint agent policies.
- We demonstrate that our method maintains a guarantee of monotonic improvement for each agent's policy under a single rollout scheme, and further show that under theoretical guarantees, the tightness of the monotonic bound for joint policies in a single rollout algorithm is the most stringent, leading to effective policy optimization.
- We developed MACPAO, an algorithm using the deep actor-critic framework with a clipped surrogate approximation. We demonstrate that MACPAO surpasses current leading methods in both reward and cost efficiency.

II. RELATED WORK

Bimanual Dexterous Manipulation. In the field of robotics, the manipulation of dexterous hands is considered one of the most challenging and complex tasks in motion control, especially noted for its intricacy. These dexterous hands are particularly useful in unstructured environments and multi-contact scenarios, offering exceptional operational flexibility. However, this flexibility comes with its own set of challenges, including high-dimensional control and complex contact models [18], [19]. Moreover, the construction and control of multi-fingered dexterous hands represent significant technological challenges. For a long time, research in this area has largely depended on trajectory optimization and model prediction, heavily dependent on the accuracy of dynamic models [6], [20], [21]. Past successes in the design and control of dexterous hands have mostly focused on simplified control problems, or the development of controllers for relatively simple tasks, such as grasping [22], [23] or in hand rotating [24], often involving manipulators with fewer degrees of freedom [6], [25]. Recently, learning-based methods have made notable advancements in bimanual dexterous manipulation. These methods effectively address uncertainties in perception and are even adaptable to unseen objects [26]. Despite the rise of many benchmarks for learning-based robotic manipulation [27]–[30], most have not addressed safety constraints, making the safe manipulation of dexterous hands a relatively unexplored topic. In this work, we conduct a range of extensive parallel tests for safe bimanual dexterous manipulation tasks. Our aim is to advance research in the safe and complex manipulation of dexterous hands.

Multi-Agent Reinforcement Learning. MARL has gained prominence for its adept handling of complex, multi-agent tasks in dynamic environments. This approach enables agents to formulate adaptive decision-making strategies through interactions with both the environment and other agents [31], [32]. A significant advancement in MARL is the adaptation of policy gradient methods, notably the Multi-Agent Proximal Policy Optimization (MAPPO), a modified version of PPO with a centralized critic. This addresses key MARL challenges such as monotonic improvement, coordination, and credit assignment [33]. The diverse applications of MARL in robotics are demonstrated in collaborative navigation [34], [35], UAV formations [36], and distributed manipulation [37]. Innovations within MARL, like disentangled attention for bimanual tasks [38] and symmetry-aware actor-critic methods for handovers [39], demonstrate its adaptability in complex simulations. However, MARL algorithms encounter non-stationarity during simultaneous agent updates, leading to high gradient variance and the need for more samples for convergence [12]. In contrast, our proposed MACPAO algorithm, a MARL method with sequential agent updates, effectively mitigates the problems of non-stationarity and updates all agents using the same samples from a single rollout.

Safe Reinforcement Learning. In Safe Reinforcement Learning (Safe RL), the key challenge is balancing operational safety with high efficiency. This issue holds particular significance in practical scenarios such as robotics and autonomous systems. Numerous studies have introduced CMDPs [40] to frame the problem of safe control, aiming to maximize rewards while adhering to cost constraints. To address the learning challenge in CMDPs, one approach involves quadratic methods for policy optimization, such as CPO [14] and Projection-based constrained policy optimization (PCPO) [41]. CPO approximates constraint satisfaction through policy search within a trust region, while PCPO employs a two-step process: first finding the optimal policy, then projecting it back into the feasible set. These methods involve local policy searches based on conjugate gradients, leading to high computational costs. Another approach is Lagrangian-based methods like Primal-Dual Optimization (PDO) [42] and its variants, which utilize Lagrangian Duality to learn constraint-satisfying policies. Although these methods excel in safety, their performance in terms of rewards is less satisfactory. Compared with the prior work, our MACPAO algorithms offer three key advantages. First, they ensure monotonic strategy improvements without compromising on safety or performance, crucial for complex, dynamic environments. Second, MACPAO's sequential policy update scheme and theoretical guarantees enhance reliability and reward outcomes. Finally, our approach ensures adherence to safety constraints in every iteration.

III. PRELIMINARIES

A. Multi-Agent Constrained Markov Decision Process

Multi-Agent CMDPs model environments where multiple agents make decisions under constraints. In bimanual dexterous manipulation, each robot hand acts as an independent agent, coordinating to achieve common goals while ensuring safety. This is crucial for tasks like handling

fragile objects or operating near humans. Our scenarios are modeled as a constrained Markov game, denoted by a tuple $(\mathcal{N}, \mathcal{S}, \mathcal{A}, \mathcal{P}, R, \mathcal{C}, \gamma)$, where $\mathcal{N} = \{1, \ldots, n\}$ is the number of agents, $\mathcal{S}$ is the state space set, $\mathcal{A} = \mathcal{A}^1 \times \cdots \times \mathcal{A}^N$ is the joint action space, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the probabilistic transition function, R is the joint reward function, which returns one total credit for all agents when they select $\boldsymbol{a}_t \in \mathcal{A}$ in state s_t. Specifically, each agent is characterized by a series of cost functions represented as $\mathcal{C} = \left\{\left(c_i^j, b_i^j\right)\right\}_{1<j<k}^{i\in\mathcal{N}}$, where each c_i^j is auxiliary cost functions mapping from a state s_t and agent-specific action $\boldsymbol{a}_t$ to a real-valued cost, and b_i^j is the constraint bound. $\gamma \in [0, 1)$ is the discount factor.

In a multi-agent system at time step t, each agent i selects an action a_t^i according to its policy $\boldsymbol{\pi}^i\left(a^i \mid s_t\right)$, where s_t denotes the state. The trajectory is defined as $\tau = \{(s_0, \boldsymbol{a}_0), (s_1, \boldsymbol{a}_1), \ldots\}$. The actions of all n agents from a joint action $\boldsymbol{a}_t = \left\{a_t^1, \ldots, a_t^n\right\}$ and a joint policy $\boldsymbol{\pi}(\cdot \mid s_t) = \pi^1 \times \ldots \times \pi^n$. This joint policy induces a normalized state distribution $d^{\boldsymbol{\pi}}$, where $d^{\boldsymbol{\pi}}(s) = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t \mathcal{P}\left(s_t = s \mid \boldsymbol{\pi}\right)$. The reward state-action value and the state-value functions are defined: $Q^{\boldsymbol{\pi}}(s, \boldsymbol{a}) = \mathbb{E}_{\tau\sim\boldsymbol{\pi}}\left[\sum_{t=0}^{\infty}\gamma^t R\left(s_t, \boldsymbol{a}_t\right) \mid s_0 = s, \boldsymbol{a_0} = a\right]$ and $V^{\boldsymbol{\pi}}(s) = \mathbb{E}_{\tau\sim\boldsymbol{\pi}}\left[Q_{\boldsymbol{\pi}}(s, \boldsymbol{a})\right]$. The advantage function is $A^{\boldsymbol{\pi}}(s, \boldsymbol{a}) = Q^{\boldsymbol{\pi}}(s, \boldsymbol{a}) - V^{\pi}(s)$.

Replacing R with c_i^j, we define the cost action-value function $Q_{c_i^j}^{\boldsymbol{\pi}}(s, a^i)$, cost value function $V_{c_i^j}^{\boldsymbol{\pi}}(s)$, and cost advantage function $A_{c_i^j}^{\boldsymbol{\pi}}(s)$ for the auxiliary costs in an analogy to V^{π}, Q^{π} and A^{π}. The goal of MARL with respect to a CMDP is to maximize the expected total reward $J_R(\boldsymbol{\pi}) = \mathbb{E}_{\tau\sim(\mathcal{P},\boldsymbol{\pi})}\left[V_{\boldsymbol{\pi}}(s)\right] = \mathbb{E}_{\tau\sim(\mathcal{P},\boldsymbol{\pi})}\left[\sum_{t=0}^{\infty}\gamma^t R\left(s_t, \boldsymbol{a}_t\right)\right]$ and minimize cost $J_{c_i^j}(\boldsymbol{\pi}) = \mathbb{E}_{\tau\sim(\mathcal{P},\boldsymbol{\pi})}\left[\sum_{t=0}^{\infty}\gamma^t c_i^j\left(s_t, a_t^i\right)\right] \leq b_i^j$ to ensure safety. The ultimate goal is to search the optimal policy $\boldsymbol{\pi}^{\star}$ such that: $\boldsymbol{\pi}^{\star} = \arg\max_{\boldsymbol{\pi}} J(\boldsymbol{\pi})$.

B. System design

1) Shadow Dexterous Hand: The Shadow Dexterous Hand [43], a 24-DoF robotic manipulator, mimics the size and dexterity of an average male hand. It uses 40 pneumatic muscle actuators connected through tendons for movement. The hand features 4 DoF in the first, middle, and ring fingers, 5 DoF in the little finger and thumb, and 2 DoF in the wrist. Each DoF is controlled by position and equipped with a joint angle sensor for precise manipulation (Fig. 2).

2) Simulation Setup: In our work, the Isaac Gym physical simulation platform [44] is utilized to facilitate the training of bimanual dexterous manipulation tasks, , which are illustrated in Fig. 1. The simulation operates at a frequency of 120Hz, whereas the control frequency is maintained at 20Hz. An end-to-end reinforcement learning strategy is implemented and trained within this simulated environment. Furthermore, Fig. 3 shows we select YCB [45] dataset objects for our tasks, focusing on object pose, allowing flexible substitution with suitably sized YCB objects.

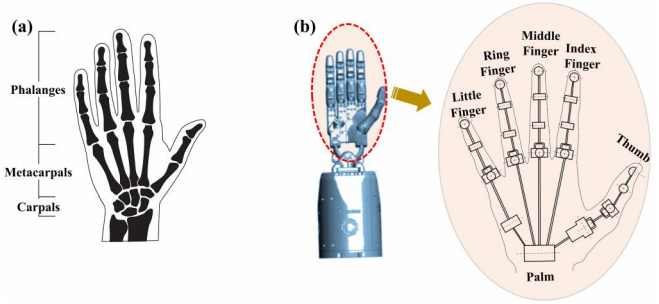

Fig. 2. Design of the joints on a Shadow Dexterous Hand

C. Dexterous Manipulation Tasks

To evaluate the safety and reliability of our described method for bimanual dexterous manipulation, we designed a system comprising three components: object datasets, tasks requiring strategic execution of safe dexterous operations (such as Hand Over, Close the Door, and Open/Close Pen Cap, as illustrated in Fig. 3), and the Safe MARL algorithm. These tasks involve challenging contact patterns and coordination, mirroring typical daily manipulation activities. We use Safe MARL to address these challenges, aiming to learn a safe control policy where each agent not only considers its own safety constraints to maximize its rewards but also keeps the safety constraints of others, ensuring their joint actions are safe. Specific safety definitions for each task are detailed separately.

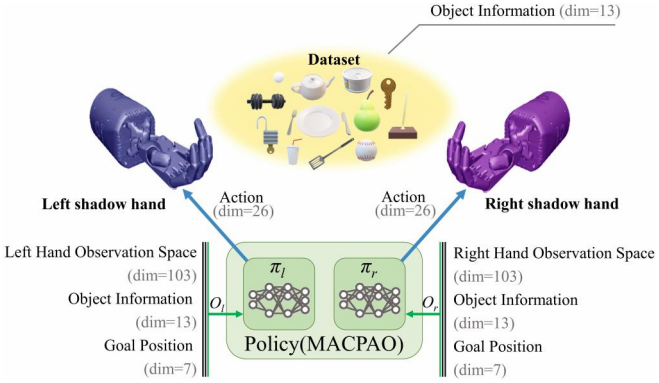

Fig. 3. The controller receives inputs comprising sensory data from the hands (indicated in green), object information and goal position. Outputs of the control, representing joint activations, are depicted in blue. Each hand is governed by its own individual controller.

1) Hand Over: (Fig. 4). This task includes a number of aspects such as object relocation, precise grasping, and hand safety within a constrained MARL framework. Successful collaboration is one where one hand must adeptly toss an object, while the other must catch it accurately, avoiding excessive force that could lead to accidental object damage, unstable grasp, or potential safety risks.

The state space is composed of the shadow hands' observation space (including shadow hands joint positions, velocities, forces, base velocity, base rotation, and the last action taken), ball information (position, linear velocity, angular velocity), and the goal position. The action space

Fig. 4. Hand Over

is the hands' actuated joint angles and the reward function is

$$r_t = d_{trans} + d_{rot}$$

$$d_{trans} := 20 * \exp\left(-0.2 * \|p_b - p_g\|_2\right)$$

$$d_{rot} := 10 \cos^{-1}\left(\text{clamp}\left(\|\Theta_{\text{diff}}\|, -1, 1\right)\right)$$

where d_{trans} is the translational distance between the ball position p_b and the goal position p_g. d_{rot} measures the angular difference between target and current orientations, using Θ_{diff} to represent rotation vector discrepancy, constrained within $[-1, 1]$ for validity. The cost function is provided as

$$c_t = \begin{cases} 1.0, & \text{for} \quad \left|\boldsymbol{ang}_{phalanges}\right| \geq 0.7 \\ 1.0, & \text{for} \quad \left|\boldsymbol{ang}_{metacarpals}\right| \geq 0.5 \\ 1.0, & \text{for} \quad \left|\boldsymbol{ang}_{carpals}\right| \geq 0.1 \\ 0, & \text{otherwise} \end{cases}$$

Where $\boldsymbol{ang}_{phalanges}$, $\boldsymbol{ang}_{metacarpals}$ and $\boldsymbol{ang}_{carpals}$ represent the motion degrees of the joints at the phalanges, metacarpals, and carpals, respectively(please refer to Fig. 2(a)). This function ensures that excessive joint movements, which could lead to instability or damage, are penalized. Similar cost functions are defined for other tasks, ensuring that each task-specific safety requirement is met. By integrating these cost functions into the learning process, our method ensures that the agents not only strive for high rewards but also adhere to strict safety constraints.

2) Close the Door: (Fig. 5). This task involves robotic agents learning to close the door, a process that requires pushing or pulling on a handle and adjusting to factors like the door's inertia, force needed, and handle design.

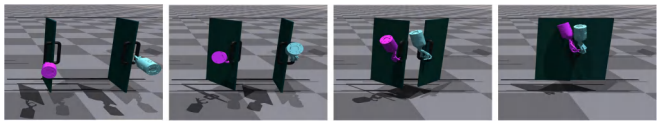

Fig. 5. Close the Door

The state space is composed of the shadow hands' observation space, door information, and the goal position. The action space is the hands' actuated joint angles and the reward function consists of three parts

$$r = 0.5 - d_{\text{left}} - d_{\text{right}} + 2 * (1 - d_{\text{lr}})$$

$$d_{\text{left}} = \|p_{\text{lhand}} - p_{\text{lhandle}}\|_2, d_{\text{right}} = \|p_{\text{rhand}} - p_{\text{rhandle}}\|_2$$

$$d_{\text{lr}} = \|p_{\text{lhandle}} - p_{\text{rhandle}}\|_2$$

Where d_{left} and d_{right} measure the distances from the left and right hands to their respective handles, and d_{lr} is the distance between the two handles. The cost function of this environment, interacting with any part of the door aside from the handle incurs a cost of 1 if such contact results in moving the door, and 0 otherwise.

3) Open/Close Pen Cap: (Fig. 6, Fig. 7). These tasks require the use of both hands to open or close a pen cap and prevent slippage or damage.

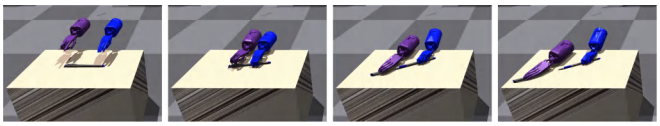

Fig. 6. Open Pen Cap

The state space is composed of the shadow hands' observation space, pen information, and the goal position. The action space is the hands' actuated joint angles and the open the pen cap reward function is

$$r = \exp\left(-10 * d_{\text{cap}}\right) + \exp\left(-10 * d_{\text{body}}\right) + d_{\text{cb}} * 5 - 0.8$$

$$d_{\text{cap}} = \|p_{\text{lhand}} - p_{\text{cap}}\|_2, d_{\text{body}} = \|p_{\text{rhand}} - p_{\text{body}}\|_2$$

$$d_{\text{cb}} = \|p_{\text{cap}} - p_{\text{body}}\|_2$$

Where d_{cap} and d_{body} denote the distances from the left and right hands to the pen cap and body, respectively, while d_{cb} represents the distance between the pen cap and body. The constraint in this environment is that the pen must not be lifted more than 0.01 cm off the desktop during the opening or closing of the pen cap. If the pen is lifted beyond this threshold, the cost is 1; otherwise, it remains 0.

Fig. 7. Close Pen Cap

The close the pen cap reward function is

$$r = \exp\left(-10 * d_{\text{cap}}\right) + \exp\left(-10 * d_{\text{body}}\right) + \left(0.37 - d_{\text{cb}}\right) * (-5) + 1.85$$

Safety constraints remain unchanged.

IV. METHOD

A. MARL Enhancement via Monotonic Sequential Updates

To establish a connection between MARL and sequential update scheme, consider a scenario where each agent is aware of its predecessor's actions in an arbitrary decision-making order. In this framework, individual agents' local advantages aggregate to form the collective advantage when each agent is aware of its predecessor's actions in a given sequence. This structured decision-making process facilitates the simplification of their joint strategy updates, where optimizing individual local advantages aligns with enhancing the overall collective advantage. Consequently, during policy updates, agents can disregard interference from others; the

local advantage function encapsulates the inter-agent dynamics. Fig. 8 illustrates this concept, assuming an arbitrary update sequence for agents (for simplicity, we consider the update order to be 1, 2, ..., n, without loss of generality). Each agent has access to its predecessor's actions and makes optimal decisions based on this information. Furthermore, the total state transition displacement encountered by agent i can be decomposed into the sum of state transition displacements caused by each updating agent. Changes induced by agents with higher priority will be encountered by more subsequent agents, thereby contributing more significantly to the non-stationarity problem.

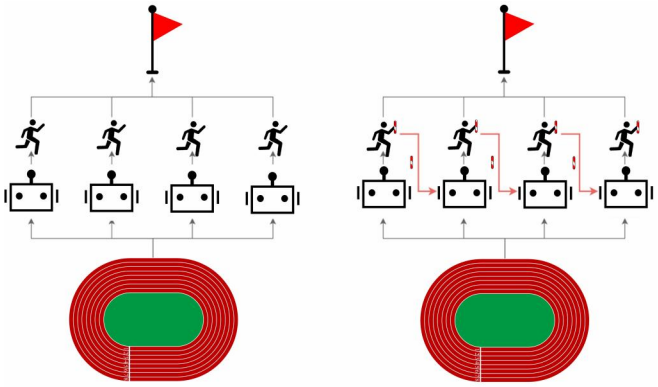

Fig. 8. Traditional vs. Sequential multi-agent learning: From simultaneous actions to ordered decision-making with predecessor consideration

To implement the sequential update paradigm for MARL, we define e^i as the set of agents updated before agent i, comprising $1, \ldots, i-1$, and denote $\bar{\pi}^i$ as the updated policy of agent i. The joint policy is denoted as $\hat{\boldsymbol{\pi}}^i$, which integrates both updated and existing policies across the agent spectrum. For initial and terminal configurations, we define $\hat{\boldsymbol{\pi}}^0 = \boldsymbol{\pi}$, representing the initial joint policy, and $\hat{\boldsymbol{\pi}}^n = \overline{\boldsymbol{\pi}}$, indicating the completely updated joint policy. In the sequential update scheme, each policy is updated individually during the training process by utilizing the outcomes of previous updates, expressed as $\hat{\boldsymbol{\pi}}^{i-1} = \prod_{j=1}^{i-1} \bar{\pi}^j \times \prod_{k=i}^{n} \pi^k$, to refine and improve the subsequent policy, $\hat{\boldsymbol{\pi}}^i$. Thus, the surrogate objective for agent i could be formulated [46] as

$$L^{\hat{\boldsymbol{\pi}}^{i-1}}\left(\hat{\boldsymbol{\pi}}^i\right) = J\left(\hat{\boldsymbol{\pi}}^{i-1}\right) + Z^{\boldsymbol{\pi}}\left(\hat{\boldsymbol{\pi}}^i\right) \tag{1}$$

where $Z^{\boldsymbol{\pi}}\left(\hat{\boldsymbol{\pi}}^i\right) = \frac{1}{1-\gamma}\mathbb{E}_{(s,\boldsymbol{a})\sim\left(d^{\boldsymbol{\pi}},\hat{\boldsymbol{\pi}}^i\right)}\left[A^{\boldsymbol{\pi}}(s,\boldsymbol{a})\right]$.

Proposition 1. *For agent i, let δ be the maximum value of $|A^{\boldsymbol{\pi}}(s,\boldsymbol{a})|$, and ω^j represent the maximum total variation distance $D_{KL}^{\max}$ between π^j and $\bar{\pi}^j$, where j belongs to $e^i \cup i$. We derive the inequality:*

$$\begin{aligned}
&\left|J\left(\hat{\boldsymbol{\pi}}^i\right) - L^{\hat{\boldsymbol{\pi}}^{i-1}}\left(\hat{\boldsymbol{\pi}}^i\right)\right| \leq \alpha^i + \beta^i \\
&\alpha^i = \frac{4\delta\omega^i}{1-\gamma}\left(1 - \frac{1}{1-\omega^j\sum_{j\in(e^i\cup\{i\})}}\right) \\
&\beta^i = \frac{1}{1-\gamma}\left[4\omega^i\delta + 2\sum_{j\in e^i}\omega^j\delta\right] \quad \text{is uncontrollable.}
\end{aligned} \tag{2}$$

The policy enhancement for agent i, as indicated by Eq. (1), depends on $Z^{\boldsymbol{\pi}}(\hat{\boldsymbol{\pi}}^i) > \beta^i$. However, uncontrollable elements within β^i, particularly $2\sum_{j\in e^i}\omega^j\delta/(1-\gamma)$, can hinder expected performance gains if $Z^{\boldsymbol{\pi}}(\hat{\boldsymbol{\pi}}^i) < \beta^i$, even after optimizing $Z^{\boldsymbol{\pi}}$. Nonetheless, a global monotonic improvement across the joint policy is achievable through cumulative analysis of all agents. We emphasize that ensuring individual agent improvements not only strengthens the joint policy's monotonic bound but also incrementally sharpens this bound with subsequent agent updates, mirroring challenges observed in MAPPO analysis.

The issue presented in Proposition 1 originates from neglecting the effect of sequential policy updates on the advantage function. To mitigate this, our methodology enhances policy evaluation by utilizing the advantage function $A^{\hat{\pi}^{i-1}}$ tailored to agent i's most recent policy update $\hat{\pi}^{i-1}$, rather than relying on the generic A^{π}. Such refinement improves the accuracy of policy evaluation, directly addressing the previously identified uncontrollable term. The surrogate objective for an individual agent i as: $L^{\hat{\boldsymbol{\pi}}^{i-1}}\left(\hat{\boldsymbol{\pi}}^i\right) = J\left(\hat{\boldsymbol{\pi}}^{i-1}\right) + \frac{1}{1-\gamma}\mathbb{E}_{(s,\boldsymbol{a})\sim\left(d^{\boldsymbol{\pi}},\hat{\boldsymbol{\pi}}^i\right)}\left[A^{\boldsymbol{\pi}^{i-1}}(s,\boldsymbol{a})\right]$, For the joint update of all agents, the surrogate objective is:

$$G^{\boldsymbol{\pi}}(\overline{\boldsymbol{\pi}}) = J(\boldsymbol{\pi}) + \frac{1}{1-\gamma}\sum_{i=1}^{n}\mathbb{E}_{(s,\boldsymbol{a})\sim\left(d^{\pi},\hat{\boldsymbol{\pi}}^i\right)}\left[A^{\boldsymbol{\pi}^{i-1}}(s,\boldsymbol{a})\right]. \tag{3}$$

It is important to note that Eq. (3) aggregates the expectations of the global advantage function under varying joint policies, diverging fundamentally from the advantage decomposition lemma presented in [47], which breaks down the global advantage function into individual local ones. Our approach emphasizes a holistic evaluation of advantage across multiple policy interactions, offering a new perspective on understanding the interplay between strategies, unlike the isolated consideration of local advantages.

By generalizing the result about the surrogate objective in Equation (3), We can now prove that the monotonic policy improvement guarantee.

Theorem 1. (Monotonic Improvement) *For each agent $i \in N$, let δ be the maximum value of $\left|A^{\boldsymbol{\pi}^{i-1}}(s,\boldsymbol{a})\right|$, and ω^j represent the maximum total variation distance $D_{KL}^{\max}$ between π^j and $\bar{\pi}^j$, where j belongs to $e^i \cup i$. We derive the inequality:*

$$|J(\overline{\boldsymbol{\pi}}) - G^{\boldsymbol{\pi}}(\overline{\boldsymbol{\pi}})| \leq \frac{4\gamma\delta}{(1-\gamma)^2}\sum_{i=1}^{n}\left(\omega^i\sum_{j\in(e^i\cup\{i\})}\omega^j\right) \tag{4}$$

According to Theorem 1, a bound is established on the discrepancy between the joint policy objective $J(\overline{\boldsymbol{\pi}})$ and its surrogate $G_{\boldsymbol{\pi}}$, which is a function related to the discount factor γ. By optimizing strategies for each agent sequentially, this bound can be incrementally tightened. This process, grounded in the contraction property theory proposed by [48], offers a more lenient condition for policy improvement, thereby enhancing the potential for performance gains.

B. MACPAO: Multi-Agent Constrained Proximal Advantage Optimization

In this section, we introduce the implementation of the MACPAO algorithm within a MARL framework. Utilizing a neural network $\Pi_\theta = \left\{\hat{\boldsymbol{\pi}}^i_{\theta^i} : \theta^i \in \Theta\right\}$, our algorithm iteratively updates policies by sampling from the environment, facilitating both evaluation and sampling processes. Addressing the challenge of minimizing objective Eq. (3), which involves unknown future state sequences and exhibits poor sampling efficiency, we draw inspiration from PPO [16]. To mitigate the instability in estimating the policy gradient for agent i, we consider a technique of truncating the ratio of joint policies from the preceding agent and once on the policy ratio of agent i. This culminates in our proposed **M**ulti-**A**gent **C**onstrained **P**roximal **A**dvantage **O**ptimization (MACPAO). approach, which formulates a practical optimization target derived from Eq. (3) as:

$$L_R^{\hat{\pi}^{i-1}_{\theta^i}}(\hat{\pi}_{\theta^i}) = \mathbb{E}_{(s,\boldsymbol{a})\sim\left(d^{\pi_{\theta^i}},\boldsymbol{\pi}_{\theta^i}\right)}\left[\min\left(f(s,\boldsymbol{a})A_R^{\hat{\pi}^{i-1}_{\theta^i}}, \operatorname{clip}\left(f(s,\boldsymbol{a}),1\pm\epsilon\right)A_R^{\hat{\pi}^{i-1}_{\theta^i}}\right)\right] \tag{5}$$

$$L_{c_i^j}^{\hat{\pi}^{i-1}_{\theta^i}}(\hat{\pi}_{\theta^i}) = \mathbb{E}_{(s,\boldsymbol{a})\sim\left(d^{\pi_{\theta^i}},\boldsymbol{\pi}_{\theta^i}\right)}\left[\max\left(f(s,\boldsymbol{a})A_{c_i^j}^{\hat{\pi}^{i-1}_{\theta^i}}, \operatorname{clip}(f(s,\boldsymbol{a}),1\pm\epsilon)A_{c_i^j}^{\hat{\pi}^{i-1}_{\theta^i}}\right)\right] \tag{6}$$

where $f(s,\boldsymbol{a}) = \frac{\bar{\pi}^i_{\theta^i}(a^i|s)}{\pi^i_{\theta^i}(a^i|s)}h(s,\boldsymbol{a})$, and $h(s,\boldsymbol{a}) = \operatorname{clip}\left(\frac{\prod_{j\in e^i}\bar{\pi}^j_{\theta^i}(a^j|s)}{\prod_{i\in e^i}\pi^j_{\theta^i}(a^j|s)}, 1\pm\frac{\epsilon}{2}\right)$. ϵ is the base clipping parameter. The surrogate objectives $L_R^{\hat{\pi}^{i-1}_{\theta^i}}$ (5) and $L_{c_i^j}^{\hat{\pi}^{i-1}_{\theta^i}}$ (6), acting as conservative estimates, enable updates from proximal policy samples, thus enhancing consistent, efficient learning.

V. Results

Task benchmark and methodology: We evaluate our MACPAO method against tasks of increasing complexity—Hand Over (elementary), Close the Door (intermediate), and Open/Close Pen Cap (advanced). We compare MACPAO's performance to baseline methods (HAPPO, MACPO, MAPPO-Lag [49]) using Average Reward per Episode and Cost metrics to demonstrate safety across varying task difficulties.

Baseline algorithms: For the UnSafe MARL algorithm, we exclusively utilized HAPPO, wherein the reward function does not incorporate information regarding cost. For the Safe MARL algorithms, we evaluated the performance of MACPO and MAPPO-Lag on dexterous manipulation system. MACPO promotes optimal and constrained policy formulation among agents in a multi-agent environment by the imposition of strong constraints coupled with a backtracking line search. Conversely, MAPPO-Lag extends credit assignment solely to reward signals and employs a traditional Lagrangian approach for adherence to constraints, optimizing policies within prescribed boundaries.

Evaluation metrics: In Safe RL, the superiority of any given agent is determined by the following precedence: on one hand, agents adhering to constraints are decidedly superior to those without constraints; on the other hand, among agents that do adhere to constraints, superiority is ascertained by comparing their cumulative returns.

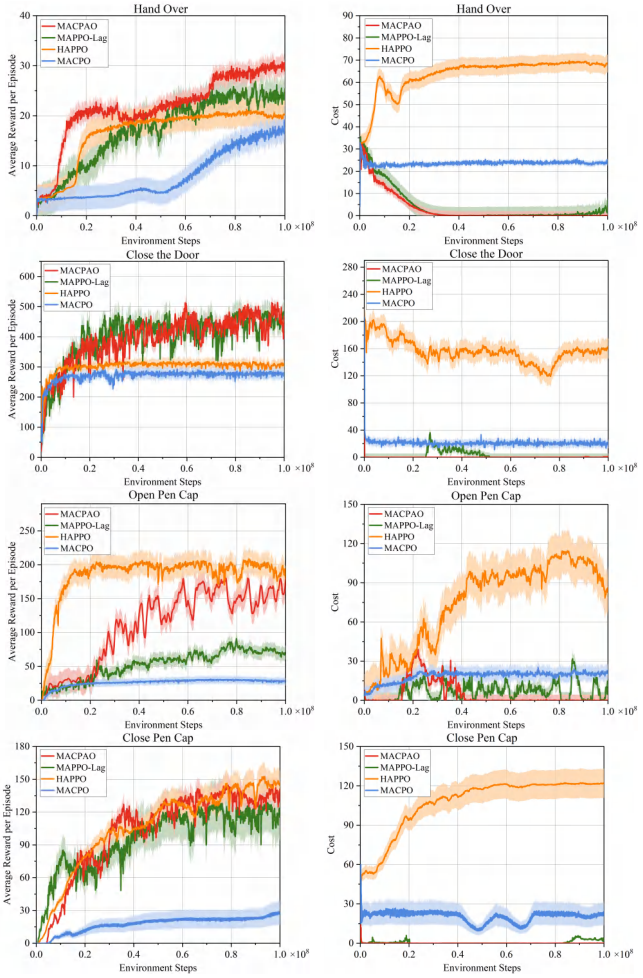

Fig. 9. In our study, we evaluate the performance of our MACPAO method across three tasks: Hand Over, Close the Door, and Open/Close Pen Cap, comparing it against both Safe and Unsafe MARL algorithms including MACPO, MAPPO-Lag, and HAPPO. Our findings demonstrate that MACPAO consistently achieves zero cost in the Hand Over task, indicating full compliance with safety constraints, and outperforms all considered baseline algorithms in terms of rewards. In the Close the Door task, MACPAO exhibits rapid convergence to zero cost, showcasing its ability to satisfy safety constraints while maintaining reward performance on par with MAPPO-Lag, yet with greater stability and higher performance compared to both MACPO and HAPPO. For the Open Pen Cap task, our method achieves nearly zero costs, aligning with safety requirements, and surpasses MACPO and MAPPO-Lag in reward metrics, although it falls short of the performance exhibited by the unsafe HAPPO algorithm. Across all tasks, the shaded areas in our graphs represent the standard deviation of scores from more than three trials, underscoring the robustness of our methodology.

Fig. 9 presents a detailed comparative analysis of the reward and cost efficacies of algorithms MACPAO, MAPPO-Lag, MACPO, and HAPPO across various continuous control tasks, employing five random seeds for each result. Each figure necessitates a threefold explanation: every plot sym-

bolizes a task of distinct complexity, and within each, an examination of four algorithms related to multi-agent control tasks is performed. Each task illustrates reward trajectories on the left (with higher values being more preferable) and cost trajectories on the right (where lower values are preferable).

The results show that the specified safe MARL algorithms, including MACPAO, MAPPO-Lag, and MACPO, are adept at quickly conforming to safety constraints, evolving from initially non-compliant policies while maintaining exploration within acceptable policy frameworks. In particular, MACPAO demonstrates superior efficacy in adhering to safety, contrasting significantly with HAPPO, which often breaches constraint conditions, highlighting its intrinsic lack of safety. This observational data suggests notable variations in cost trajectories between MACPAO and other safe MARL algorithms, revealing differing approaches to constraint optimization.

Some methodologies, like HAPPO, prioritize reward maximization, often leading to high rewards but poor safety. These methods focus on aggressive exploration and rapid policy updates, neglecting safety constraints. For example, in the 'Hand Over' task, agents might achieve high scores by throwing objects forcefully, increasing the risk of damage or instability. In the 'Close the Door' task, agents sometimes applied excessive force, causing the door to slam shut and risking damage. In the 'Open Pen Cap' task, poor coordination led to the pen cap slipping or the pen being lifted off the table, violating safety constraints.

MACPAO addresses these issues by implementing adaptive step sizes in gradient ascent, significantly mitigating constraint challenges during initial training. This methodological innovation facilitates swift stabilization within safety parameters, enabling MACPAO to secure high rewards and marking a notable advancement over existing safe algorithms like MAPPO-Lag and MACPO. Despite these enhanced functionalities, MACPAO does not outperform the inherently unsafe HAPPO algorithm in reward metrics for tasks such as 'Open Pen Cap' and 'Close Pen Cap'. However, by integrating safety constraints into the learning process, MACPAO ensures both high performance and safe operation, effectively balancing reward optimization with safety adherence.

VI. Conclusion and Future Work

In this work, we introduced MACPAO, a novel algorithm designed to address the challenges of Safe MARL in bimanual dexterous manipulation tasks within robotics. By incorporating a strategic approach to manage the non-stationarity presented by multiple interacting agents, MACPAO significantly enhances cooperative control while ensuring strict adherence to safety constraints for each agent and the collective team. Our comprehensive evaluations demonstrate that MACPAO not only surpasses existing Safe MARL algorithms in achieving higher rewards but also maintains a steadfast commitment to safety compliance across various tasks with inherent safety requirements. This dual achievement underscores the effectiveness of MACPAO in navigating the complex balance between operational capability and safety, marking a significant step forward in the development of robust and reliable robotic systems for complex manipulation tasks.

Future research will target the enhancement of the bimanual dexterous manipulation system by adding deformable object manipulation and bridging the sim-to-real gap with more realistic sensory inputs. We will explore Learning from Demonstration to enrich training methodologies and improve robot body simulations for more varied task handling. Addressing the limitations of current meta/multi-task RL algorithms and enhancing sim-to-real transfer capabilities are also prioritized to expedite advancements in robotic manipulation and Safe RL fields.

References

[1] H. Moravec, *Mind children: The future of robot and human intelligence*. Harvard University Press, 1988.

[2] A. M. Dollar and R. D. Howe, "The sdm hand as a prosthetic terminal device: a feasibility study," in *2007 IEEE 10th International Conference on Rehabilitation Robotics*. IEEE, 2007, pp. 978–983.

[3] E. Mattar, "A survey of bio-inspired robotics hands implementation: New directions in dexterous manipulation," *Robotics and Autonomous Systems*, vol. 61, no. 5, pp. 517–544, 2013.

[4] L. Zhao, Y. Wu, J. Blanchet, M. Perroni-Scharf, X. Huang, J. Booth, R. Kramer-Bottiglio, and D. Balkcom, "Soft lattice modules that behave independently and collectively," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 5942–5949, 2022.

[5] L. Zhao, Y. Wu, W. Yan, W. Zhan, X. Huang, J. Booth, A. Mehta, K. Bekris, R. Kramer-Bottiglio, and D. Balkcom, "Starblocks: Soft actuated self-connecting blocks for building deformable lattice structures," *IEEE Robotics and Automation Letters*, vol. 8, no. 8, pp. 4521–4528, 2023.

[6] U. Kim, D. Jung, H. Jeong, J. Park, H.-M. Jung, J. Cheong, H. R. Choi, H. Do, and C. Park, "Integrated linkage-driven dexterous anthropomorphic robotic hand," *Nature communications*, vol. 12, no. 1, p. 7177, 2021.

[7] X. Deng, N. Li, D. Mguni, J. Wang, and Y. Yang, "On the complexity of computing markov perfect equilibrium in general-sum stochastic games," *National Science Review*, vol. 10, no. 1, p. nwac256, 2023.

[8] J. G. Kuba, M. Wen, L. Meng, H. Zhang, D. Mguni, J. Wang, Y. Yang *et al.*, "Settling the variance of multi-agent policy gradients," *Advances in Neural Information Processing Systems*, vol. 34, pp. 13 458–13 470, 2021.

[9] S. Kakade and J. Langford, "Approximately optimal approximate reinforcement learning," in *Proceedings of the Nineteenth International Conference on Machine Learning*, 2002, pp. 267–274.

[10] M. Andrychowicz, A. Raichuk, P. Stańczyk, M. Orsini, S. Girgin, R. Marinier, L. Hussenot, M. Geist, O. Pietquin, M. Michalski *et al.*, "What matters in on-policy reinforcement learning? a large-scale empirical study," *arXiv preprint arXiv:2006.05990*, 2020.

[11] S. V. Albrecht and P. Stone, "Autonomous agents modelling other agents: A comprehensive survey and open problems," *Artificial Intelligence*, vol. 258, pp. 66–95, 2018.

[12] P. Hernandez-Leal, M. Kaisers, T. Baarslag, and E. M. De Cote, "A survey of learning in multiagent environments: Dealing with non-stationarity," *arXiv preprint arXiv:1707.09183*, 2017.

[13] T. M. Moldovan and P. Abbeel, "Safe exploration in markov decision processes," *arXiv preprint arXiv:1205.4810*, 2012.

[14] J. Achiam, D. Held, A. Tamar, and P. Abbeel, "Constrained policy optimization," in *International conference on machine learning*. PMLR, 2017, pp. 22–31.

[15] L. Yang, Y. Zhang, G. Zheng, Q. Zheng, P. Li, J. Huang, and G. Pan, "Policy optimization with stochastic mirror descent," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 8, 2022, pp. 8823–8831.

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

[17] S. Gu, J. G. Kuba, Y. Chen, Y. Du, L. Yang, A. Knoll, and Y. Yang, "Safe multi-agent reinforcement learning for multi-robot control," *Artificial Intelligence*, p. 103905, 2023.

[18] W. G. Bircher, A. M. Dollar, and N. Rojas, "A two-fingered robot gripper with large object reorientation range," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 3453–3460.

[19] N. Rahman, L. Carbonari, M. D'Imperio, C. Canali, D. G. Caldwell, and F. Cannella, "A dexterous gripper for in-hand manipulation," in *2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE, 2016, pp. 377–382.

[20] A. M. Okamura, N. Smaby, and M. R. Cutkosky, "An overview of dexterous manipulation," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 1. IEEE, 2000, pp. 255–262.

[21] V. Kumar, E. Todorov, and S. Levine, "Optimal control with learned local models: Application to dexterous manipulation," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 378–383.

[22] M. Ciocarlie, C. Goldfeder, and P. Allen, "Dimensionality reduction for hand-independent dexterous robotic grasping," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2007, pp. 3270–3275.

[23] J. Ye, J. Wang, B. Huang, Y. Qin, and X. Wang, "Learning continuous grasping function with a dexterous hand from human demonstrations," *IEEE Robotics and Automation Letters*, vol. 8, no. 5, pp. 2882–2889, 2023.

[24] H. Qi, A. Kumar, R. Calandra, Y. Ma, and J. Malik, "In-hand object rotation via rapid motor adaptation," in *Conference on Robot Learning*. PMLR, 2023, pp. 1722–1732.

[25] Y. Luo, K. Xie, S. Andrews, and P. Kry, "Catching and throwing control of a physically simulated hand," in *Proceedings of the 14th ACM SIGGRAPH Conference on Motion, Interaction and Games*, 2021, pp. 1–7.

[26] A. Rajeswaran, K. Lowrey, E. V. Todorov, and S. M. Kakade, "Towards generalization and simplicity in continuous control," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[27] S. Li, X. Ma, H. Liang, M. Görner, P. Ruppel, B. Fang, F. Sun, and J. Zhang, "Vision-based teleoperation of shadow dexterous hand using end-to-end deep neural network," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 416–422.

[28] A. Rajeswaran, V. Kumar, A. Gupta, G. Vezzani, J. Schulman, E. Todorov, and S. Levine, "Learning complex dexterous manipulation with deep reinforcement learning and demonstrations," *arXiv preprint arXiv:1709.10087*, 2017.

[29] B. R. Kiran, I. Sobh, V. Talpaert, P. Mannion, A. A. Al Sallab, S. Yogamani, and P. Pérez, "Deep reinforcement learning for autonomous driving: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 6, pp. 4909–4926, 2021.

[30] J. Ramírez, W. Yu, and A. Perrusquía, "Model-free reinforcement learning from expert demonstrations: a survey," *Artificial Intelligence Review*, pp. 1–29, 2022.

[31] H. He, J. Boyd-Graber, K. Kwok, and H. Daumé III, "Opponent modeling in deep reinforcement learning," in *International conference on machine learning*. PMLR, 2016, pp. 1804–1813.

[32] K. Son, D. Kim, W. J. Kang, D. E. Hostallero, and Y. Yi, "Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning," in *International conference on machine learning*. PMLR, 2019, pp. 5887–5896.

[33] C. Yu, A. Velu, E. Vinitsky, J. Gao, Y. Wang, A. Bayen, and Y. Wu, "The surprising effectiveness of ppo in cooperative multi-agent games," *Advances in Neural Information Processing Systems*, vol. 35, pp. 24 611–24 624, 2022.

[34] A. Malus, D. Kozjek *et al.*, "Real-time order dispatching for a fleet of autonomous mobile robots using multi-agent reinforcement learning," *CIRP annals*, vol. 69, no. 1, pp. 397–400, 2020.

[35] Y. Jin, Y. Zhang, J. Yuan, and X. Zhang, "Efficient multi-agent cooperative navigation in unknown environments with interlaced deep reinforcement learning," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 2897–2901.

[36] Z. Xia, J. Du, J. Wang, C. Jiang, Y. Ren, G. Li, and Z. Han, "Multi-agent reinforcement learning aided intelligent uav swarm for target tracking," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 1, pp. 931–945, 2021.

[37] G. Ding, J. J. Koh, K. Merckaert, B. Vanderborght, M. M. Nicotra, C. Heckman, A. Roncone, and L. Chen, "Distributed reinforcement learning for cooperative multi-robot object manipulation," *arXiv preprint arXiv:2003.09540*, 2020.

[38] M. Zhang, P. Jian, Y. Wu, H. Xu, and X. Wang, "Dair: Disentangled attention intrinsic regularization for safe and efficient bimanual manipulation," *arXiv preprint arXiv:2106.05907*, 2021.

[39] Y. Li, C. Pan, H. Xu, X. Wang, and Y. Wu, "Efficient bimanual handover and rearrangement via symmetry-aware actor-critic learning," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 3867–3874.

[40] E. Altman, *Constrained Markov decision processes*. Routledge, 2021.

[41] T.-Y. Yang, J. Rosca, K. Narasimhan, and P. J. Ramadge, "Projection-based constrained policy optimization," *arXiv preprint arXiv:2010.03152*, 2020.

[42] Y. Chow, M. Ghavamzadeh, L. Janson, and M. Pavone, "Risk-constrained reinforcement learning with percentile risk criteria," *Journal of Machine Learning Research*, vol. 18, no. 167, pp. 1–51, 2018.

[43] P. Tuffield and H. Elias, "The shadow robot mimics human actions," *Industrial Robot: An International Journal*, vol. 30, no. 1, pp. 56–60, 2003.

[44] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa *et al.*, "Isaac gym: High performance gpu-based physics simulation for robot learning," *arXiv preprint arXiv:2108.10470*, 2021.

[45] B. Calli, A. Singh, J. Bruce, A. Walsman, K. Konolige, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Yale-cmu-berkeley dataset for robotic manipulation research," *The International Journal of Robotics Research*, vol. 36, no. 3, pp. 261–268, 2017.

[46] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *International conference on machine learning*. PMLR, 2015, pp. 1889–1897.

[47] J. G. Kuba, R. Chen, M. Wen, Y. Wen, F. Sun, J. Wang, and Y. Yang, "Trust region policy optimisation in multi-agent reinforcement learning," *arXiv preprint arXiv:2109.11251*, 2021.

[48] R. Munos, T. Stepleton, A. Harutyunyan, and M. Bellemare, "Safe and efficient off-policy reinforcement learning," *Advances in neural information processing systems*, vol. 29, 2016.

[49] S. Gu, J. G. Kuba, M. Wen, R. Chen, Z. Wang, Z. Tian, J. Wang, A. Knoll, and Y. Yang, "Multi-agent constrained policy optimisation," *arXiv preprint arXiv:2110.02793*, 2021.