Learning to Self-Reconfigure for Freeform Modular Robots via Altruism Proximal Policy Optimization

Lei Wu , **Bin Guo*** , **Qiuyun Zhang** , **Zhuo Sun** , **Jieyi Zhang** and **Zhiwen Yu**
Northwestern Polytechnical University
{leiwu, qiuyunzhang, jyzhang}@mail.nwpu.edu.cn, {guob, zsun, zhiwenyu}@nwpu.edu.cn

Abstract

The advantages of modular robot systems stem from their ability to change between different configurations, enabling them to adapt to complex and dynamic real-world environments. Then, how to perform the accurate and efficient change of the modular robot system, i.e., the self-reconfiguration problem, is essential. Existing reconfiguration algorithms are based on discrete motion primitives and are suitable for lattice-type modular robots. The modules of freeform modular robots are connected without alignment, and the motion space is continuous. It renders existing reconfiguration methods infeasible. In this paper, we design a parallel distributed self-reconfiguration algorithm for freeform modular robots based on multi-agent reinforcement learning to realize the automatic design of conflict-free reconfiguration controllers in continuous action spaces. To avoid conflicts, we incorporate a collaborative mechanism into reinforcement learning. Furthermore, we design the distributed termination criteria to achieve timely termination in the presence of limited communication and local observability. When compared to the baselines, simulations show that the proposed method improves efficiency and congruence, and module movement demonstrates altruism.

1 Introduction

Modular robots have advantages of low cost, robustness, and diversity [Yim *et al.*, 2000], compared to fixed-structure robots. They demonstrate potential applications in industrial manufacturing [Liu and Althoff, 2020], smart home [Spröwitz *et al.*, 2014], and other fields.

Modular robot system [Ahmadzadeh and Masehian, 2015] can usually be classified as lattice type, chain type and truss type according to connection mechanisms. These types of modular robots need dock-to-dock alignment and interconnecting, connectors significantly affects task performance. In practice, several reasons may result in the connector failure, including accumulation of manufacturing variations, lower sensor accuracy, and offsets from external forces [Swissler and Rubenstein, 2020]. Thus, the strong reliance on connectors with specific locations is a common problem in the past.

Free and diverse connectivities among living organisms [Gumbiner, 1996; Tennenbaum *et al.*, 2016] inspires a range of freeform module designs. Compared to lattice modular robots with a limited number of fixed connection positions, non-lattice freeform modular robots have continuous freeform connectors that do not need to be aligned and can be reconfigured more freely in a continuous configuration space. Freeform designs increase the efficiency of reconfiguration and reduce connection errors [Swissler and Rubenstein, 2020]. These new freeform modular robots have shown great potentials for solving realistic tasks. For instance, using freeform modular robots to construct dexterous robotic arms can adapt to narrow environments with dynamic obstacles [Zong *et al.*, 2022]. Therefore, the work focuses on such freeform modular robots.

Reconfiguration problem is a key and challenging issue for modular robot applications [Seo *et al.*, 2019]. Firstly, diverse ways in which modules are combined lead to a huge configuration space and the space size grows exponentially with the number of modules. For a system containing N modules, where each module has c connectors and w connection methods, the configuration space contains $(c \cdot w)^N$ different configurations. Secondly, since modules can move or dock simultaneously, the branching factor of the tree describing the configuration is $\mathcal{O}(mk)$ [Tucci *et al.*, 2018], where m is the number of possible movements and k is the number of modules that can move. Finally, search for a global optimal planning between any two configurations is NP-hard [Ye *et al.*, 2019] due to complex kinematic constraints created by the dependencies among modules. To address reconfiguration problem, existing methods usually adopts the configuration space search and the control rule design [Seo *et al.*, 2019].

For freeform modular robots, reconfiguration problem is more challenging. This is because the connection methods w and the possible movement methods k are infinite, i.e., both the configuration space and the movement space are continuous. As a result, during the reconfiguration process, the motion constraints are too complex, thereby making existing methods infeasible. Furthermore, discretizing the reconfiguration problem of freeform modular robots, leads to the combinatorial explosion and dissipates the particular advantages

*Corresponding author.

offered by freeform designs.

In this paper, we propose a distributed reconfiguration algorithm for freeform modular robots. The main challenge we faced is that due to the kinematic constraints of the system, the motion trajectories of each module will conflict during the decentralised reconfiguration process, which deteriorates the success rate and efficiency of reconfiguration.

To address this problem, we let modules learn to avoid conflicts through training. Since it is difficult to synchronise global configuration information in real time in modular robot systems [Tucci *et al.*, 2018], we can only use local configuration information for coordination between modules. In this case, the method of maximising joint rewards is not suitable [McKee *et al.*, 2020]. Modules also generate self-interested behaviour. For this mixed-motive problem, while several methods have been proposed in [Barbosa *et al.*, 2020; Hughes *et al.*, 2018; Wang *et al.*, 2018; Stastny *et al.*, 2021], most of them are only applicable to discrete action space problems, like matrix gaming or video games. Inspired by the altruism scale, we design a personalized coordination mechanism in proximal policy optimization to avoid conflicts caused by the continuous movement and docking of modules. We introduce personalized altruism factors for all modules to accommodate the dynamically changing dependencies among modules and find the optimal altruism of each module through meta-reinforcement learning [Günther *et al.*, 2020; Tang *et al.*, 2021].

Simulations show that by avoiding conflict, our method has better reconfiguration efficiency and average congruence, and module movements show altruism. Furthermore, our method is robust and generalizes well, satisfying the robust and scalable design principles of modular robot systems.

2 Related Works

2.1 Freeform Modular Robots

In freeform modular robot system, modules can move independently and have freeform attachment mechanisms to form amorphous configurations.

Two-dimensional freeform modular robots have been used as tools to study cellular movement patterns and the nervous system. Randomly oscillating particle robots [Li *et al.*, 2019] form amorphous systems resembling cell clusters through a loosely coupled connection mechanism. MarXbots [Mathews *et al.*, 2017] enables connection in almost any direction by means of three-finger connector and passive connection ring, on the basis of which a mergeable neural system is realized.

Recently, several three-dimensional freeform modular robots have been proposed [Swissler and Rubenstein, 2020; Tu *et al.*, 2022]. Among them, FreeBOT [Liang *et al.*, 2020] and SnailBot [Zhao and Lam, 2022] all form a free connection between modules via a ferromagnetic spherical shell and an active connection system containing permanent magnets.

Freeform modular robots do not require precise alignment with specified connectors during reconfiguration process, thus can form a variety of configurations for diverse real-world tasks.

However, the reconfiguration problem of this type of modular robot is non-trivial, due to the larger configuration space and more conflicts among modules. To the best of our knowledge, there is no existing reconfiguration algorithm for this new type of modular robot. And our framework is applicable to many different types of module designs and motion primitives.

2.2 Reconfiguration Methods

Existing reconfiguration methods can be divided into search-based and control-based. Due to the complexity of reconfiguration problem, search-based methods [Ahmadzadeh and Masehian, 2015; Seo *et al.*, 2019] take a long time to find reconfiguration paths, especially for modular robots with limited computing resources.

More research attempts to design control rules for modular robots that do not require global information. Bionic [Bie *et al.*, 2018] or manual design rules [Moussa *et al.*, 2021; Kawano, 2020], calculation of gradients based on various virtual forces [Hourany *et al.*, 2021; Stoy, 2006], and settings with the help of scaffolding [Thalamy *et al.*, 2019; Thalamy *et al.*, 2020] have attempted to solve reconfiguration problems. Different from previous work, we address reconfiguration problem for the first time using multi-agent reinforcement learning(MARL). The motivation is that previous algorithms based on lattice type designs are not suitable for freeform designs with continuous configuration space and action space.

Similar to past methods, conflicts occur during distributed movements due to kinematic constraints. As shown in Figure 1, conflicts involved in past work mainly include collision (different modules are expected to occupy same position), disintegration (due to improper movement of key modules), and closure (motion trajectory is hindered by formed structure). Proper conflict resolution is necessary in many tasks [Ahmadzadeh and Masehian, 2015].

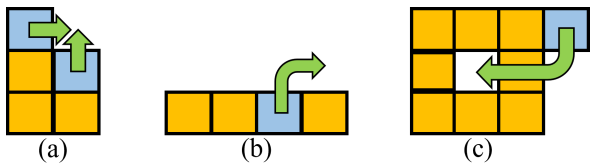

Figure 1: Diagram of conflicts. (a) Collision, (b) Disintegration, (c) Closure. Green arrows indicate motion planning of blue modules.

Conflicts may cause failure of reconfiguration. For example, closure will cause deadlock in reconfiguration process, and disintegration means failure. Moreover, conflicts can cause reduction in reconfiguration efficiency. In the existing work, the handling of conflicts means restriction on the movement of some modules [Hourany *et al.*, 2021]. For example, [Tucci *et al.*, 2018] avoid blockages by manually setting detection points.

We focus on avoiding negative effects of conflicts during the parallel reconfiguration of freeform modular robots. In continuous reconfiguration of freeform modular robots, conflicts are even more severe and conflict avoidance methods designed for discrete spaces are not applicable. Learning to avoid conflict is promising in MARL framework.

3 Problem Formulation

3.1 Freeform Modular Robot Abstraction

We model the main components of freeform modular robots as follows: Each module is regarded as a standard spherical shape, containing an active connector, passive connectors, and necessary sensing devices, driven by motion motors. Entire spherical shell can be used as a passive connector, and the active connector in module decides how to connect with passive connectors of touched modules. The motion motor of the module has continuous motion capability in two-dimensional space. The module can move on the surface of connected module but such movements are restricted to 2D only. Modules can communicate with each other within a range. As shown in Figure 2, these modules can be interconnected and combined into different geometric or topological configurations. Geometrically different configurations represent the distinct positional relationships among modules. Topologically different configurations represents the distinct connection relationships among modules. Different configurations are often derived from optimal configuration designs[Liu and Althoff, 2020; Whitman *et al.*, 2020; Hu *et al.*, 2022].

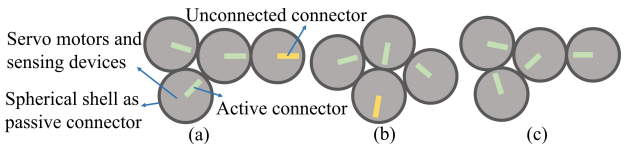

Figure 2: Different configurations formed by four modules. (a)Anatomy of freeform modular robots. (b)A configuration geometrically different from (a). (c)A topologically different configuration from (a).

3.2 Reconfiguration Problem

The reconfiguration problem of freeform modular robots studied in this paper is devoted to finding the optimal module motion sequence between two configurations so that the reconfiguration process can be completed as soon as possible. Consider a modular robot system $\{M_i\}$ consisting of n modules, given a pair of initial and target configurations, from the configuration space X. The goal of reconfiguration algorithm is to find an action sequence $\{u_{i,t}\}$ in allowed action space U so that the action sequence can convert x_{start} into x_{target}, with the shortest time T. Any given configuration x is uniquely determined by position information P_i of each module and topological connection information $\delta_{i,j}$ of all modules, which is the extension of incidence matrix [Chen and Burdick, 1998] and assembly incidence matrix [Chen and Yang, 1998] on freeform modular robot. During reconfiguration process, U contains actions that make motor torque and velocity vector of each module satisfy corresponding motion constraints. Due to exponential configuration space and tightly-coupled kinematic constraints, obtaining the optimal module action sequence is challenging.

3.3 Markov Decision Process

To solve reconfiguration problem, we adopt the method of MARL. We consider two general design principles for modular robots: (1) Each module has limited sensing capabilities and communicates with a limited range of modules, (2) Modules can move in parallel and distributed to complete reconfiguration quickly. To comply with above principles, we take the reconfiguration problem of modular robots as a Decentralized Partially Observable Markov Decision Process (Dec-POMDP). Each module M_i does not have direct access to environment state s_t but can obtain local observation $o_{i,t}$ through its observation model Z_i. At time step t, each module selects action $u_{i,t}$ based on its observation $o_{i,t}$, and policy π_i to form a joint action $\boldsymbol{u_t}$. The environment transits to s_{t+1} with the probability $\boldsymbol{P}(s_{t+1}|s_t, \boldsymbol{u_t})$. All modules have the same form of goal, i.e., to maximize respective discounted cumulative reward $\mathbb{E}[\sum_{t'=t}^{T} \gamma(t'-t)R_{i,t}]$, usually defined as value function $V_i(s_t)$. We approximate the value function $V_{i,t}$ with a neural network to cope with high-dimensional configuration space and action space of freeform modular robots.

4 Approach

4.1 Reward Function Design

For reconfiguration problem, we design a reward function as,

$$R_{i,t} = c_p \times R_{i,t}^{top} + c_q \times R_{i,t}^{geo} - c_t \tag{1}$$

which is used by all modules. The first term is topological reward, the second term is geometric reward, and the third term is a tiny temporal reward, balanced by the hyperparameter c_p and c_q. The reward encourages modules to approach target configuration in both geometric and topological directions.

The topological reward is designed based on local topological information of configuration. For each module M_i, its connection relationship with other $n-1$ modules can be represented by $\delta_{i,j}$, where j belongs to $I \setminus \{i\}$. To encourage modules to make their local topological connections approach their local topological relationships in target configuration, the topological reward is designed as follows:

$$R_{i,t}^{top} = \frac{\sum_j (\|\delta_{i,j}^{t-1} - \delta_{i,j}^{G}\| - \|\delta_{i,j}^{t} - \delta_{i,j}^{G}\|)}{\sum_j \|\delta_{i,j}^{0} - \delta_{i,j}^{G}\|} \tag{2}$$

where $\|\delta_{i,j}^{t-1} - \delta_{i,j}^{G}\|$ and $\|\delta_{i,j}^{t} - \delta_{i,j}^{G}\|$ refer to the distance between local topology of M_i and target topology at time steps $t-1$ and t, respectively. This differential form avoids reward sparsity. The denominator is only related to the topology of local connection of module M_i in initial configuration and target configuration, which normalizes the reward, and avoids repeated adjustments to hyperparameters.

Utilizing the connection relationship can only promote topological similarity of configurations. To make configuration consistent in shape with target configuration, further geometric information needs to be introduced. Here, we use position information to design the geometric reward function:

$$R_{i,t}^{geo} = \|P_{i,t} - P_{i,t-1}\| \cos\varphi \tag{3}$$

where the first term is displacement vector of module and φ is the angle between two vectors with $P_{i,t-1}$ as the starting point and $P_{i,t}$, and P_i^G (the position vector of M_i in target configuration) as the endpoints respectively. Therefore, the geometric reward measures the effective distance that module moves in correct direction to target position per time step.

4.2 Altruism Proximal Policy Optimization

We process the continuous action space of freeform modular robots based on the Proximal Policy Optimization (PPO) algorithm. If we try to maximize the discounted cumulative rewards of each module according to PPO [Schulman *et al.*, 2017] , the objective function should be

$$L^{CLIP}(\theta_i) = \hat{E}_{i,t}[\min(k\hat{A}_{i,t}, clip(k, 1-\varepsilon, 1+\varepsilon)\hat{A}_{i,t})] \quad (4)$$

where, $k = \frac{\pi_{i,\theta}(v_i^t|s_t)}{\pi_{i,\theta_{old}}(v_i^t|s_t)}$ and $A_{i,t} = R_{i,t}+\gamma V_i(s_{t+1})-V_i(s_t)$ represent importance sampling factor and advantage function respectively.

This independent learning makes it difficult to perform collaboration because each module only regards other modules as part of environment. In the process of reconfiguration, each module moves towards its target quickly and selfishly, which causes the conflicts described above.

In addition, maximizing the reward sum of all modules is not a good solution as well. Even if it solves the credit assign problem, the network has difficulty in fitting dynamic interaction among modules in large-scale reconfiguration. Modules may learn unreasonable behaviors to maximize the overall reward that make the reconfiguration fail [McKee *et al.*, 2020]. More importantly, modules with limited perception ability are difficult to obtain overall reward synchronously in practical modular robot system [Ahmadzadeh and Masehian, 2015].

In this paper, we solve the reconfiguration of freeform modular robots from the perspective of hybrid motives, which make each module learn to coordinate and avoid conflicts.

Our approach is inspired by the altruism scale [Sawyer, 1966]. Evolutionary altruistic tendencies enable species to avoid social dilemmas through coordination [Bowles, 2006; De Dreu *et al.*, 2010]. For example, people avoid trampling and crowding by showing comity, although requires sacrifice of detours or slowing down. Therefore, we hope to introduce altruism mechanism in MARL to achieve cooperation among modules, and to avoid conflicts, thereby completing reconfiguration successfully and efficiently.

First, we measure the benefits of others. Neighbours of modules and their relationships are dynamically changing during the reconfiguration process. Borrowing from a similar mean-field idea that averages the actions of neighboring agents [Yang *et al.*, 2018], however, from the perspective of measuring the average reward of others [Nisbett and Kunda, 1985; Peng *et al.*, 2021], we define the mean-field reward of each module

$$R_{i,t}^{MF} = \frac{\sum_{j\in N_{i,t}} R_{j,t}}{|N_{i,t}|} \quad (5)$$

Where $N_{i,t} = \{j : \|P_i^t - P_j^t\| \leq d\}$ is a changing set containing the indices of other modules within a radius d of M_i. In this way, the mean-field reward represents average reward of neighboring modules that change dynamically around module M_i. Module M_i only needs to consider the impact on this single reward and does not need to separately consider the impact of each specific module, which reduces interaction cost and guarantees scalability. We can calculate the mean field advantage function $A_{i,t}^{MF} = R_{i,t}^{MF}+\gamma V_i^{MF}(s_{t+1})-V_i^{MF}(s_t)$, where the mean field value function $V_i^{MF}(s_t) = \mathbb{E}[\sum_{t'=t}^{T}\gamma(t'-t)R_{i,t}^{MF}]$ is approximated by another neural network.

Then, we introduce the sociological mechanism of the altruism scale [Sawyer, 1966] to measure each module's tendency to benefit others, and form the altruism reward

$$R_{i,t}^{AS} = R_{i,t} + \alpha_i R_{i,t}^{MF} \quad (6)$$

where α_i is an altruism factor in the interval $(-1, 1)$ that measures each module's attitude that benefits others. The meaning of marginal value of α_i in sociological experiments is described as follows: people with $\alpha_i = -1$ tend to maximize the difference between their own and others' benefits i.e., the competitive personalities. People with $\alpha_i = 0$ don't care about others' benefits i.e., the selfish personalities. And people with $\alpha_i = 1$ maximize the mutual benefit of themselves and others i.e., the completely cooperative personalities. Note that altruism factors are more suitable for reconfiguration, which is a non-zero-sum game, than the ring metric of social value orientation [McKee *et al.*, 2020; Peng *et al.*, 2021] introduced by previous work in non-strict zero-sum game problems.

The vector $\boldsymbol{\alpha}$ is the set of α_i, representing the distribution of altruistic tendencies across the population. And the altruism factor α_i is an personalised attribute of each module M_i. Heterogeneity and diversity can improve performance [McKee *et al.*, 2020]. Here we verify the rationality of introducing altruism factors through two preliminary experiments.

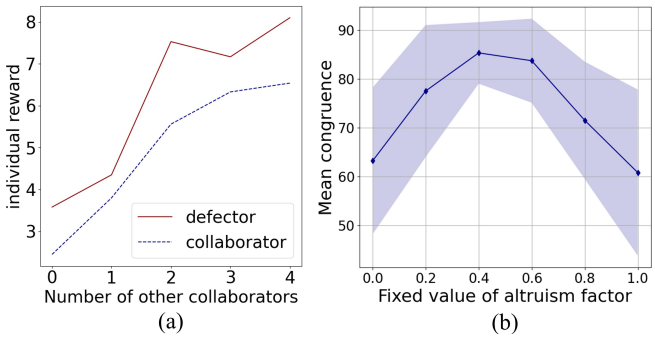

Figure 3: Priori experiments. (a)Schelling diagram of reconfiguration problem. (b)Reconfiguration performance varies with altruism. The shaded region shows the standard error of the mean.

We train PPO by way of CTDE(centralized training and decentralized execution) [Lowe *et al.*, 2017] to maximize R^{AS}:

$$L_{\alpha_i}^{AS}(\theta_i) = \hat{E}_{i,t}[\min(k\hat{A}_{i,t}^{AS}, clip(k, 1-\varepsilon, 1+\varepsilon)\hat{A}_{i,t}^{AS}] \quad (7)$$

where altruism advantage function $A_{i,t}^{AS} = A_{i,t} + \alpha_i A_{i,t}^{MF}$.

By fixing the value of α_i to 1, we force the agent to learn cooperative behaviors. By fixing the value of α_i to -1, we force the agent to learn betrayal behaviors. In this way, two policy sets are formed. Draw a Schelling diagram as Figure 3, where the horizontal axis represents the number of other modules that take cooperative behaviors, and the vertical axis represents the average payoff of module M_i. From the Figure 3, it can be seen that for modules, adopting a betrayal strategy

can only improve short-term benefits, and adopting appropriate cooperative behaviors can improve overall long-term benefits. Therefore, the reconfiguration problem of modular robots is essentially a social dilemma problem [Schelling, 1973; Hughes *et al.*, 2018]. Furthermore, different from the sequential social dilemma problem previously posed in discrete video games, the reconfiguration problem for modular robots is a continuous action space problem.

Similarly, we set up an equally spaced set of $\boldsymbol{\alpha}$ fixed to all modules. Test results (as shown in Figure 3) indicate that there is an optimal α_i that achieves the best performance.

But it is impractical to set each α_i manually. Therefore, referring to the meta reinforcement learning [Günther *et al.*, 2020; Peng *et al.*, 2021; Tang *et al.*, 2021], we take Equation 8 as the optimization goal, and perform another layer of training to optimize an appropriate personalised α_i for each module. A suitable set of personalised $\boldsymbol{\alpha}$ can lead to the best overall reconfiguration performance.

$$L_i^G(\theta_i|\theta_1,\theta_2\cdots) = \mathbb{E}[\sum_t \frac{\sum_j R_{j,t}}{N}] \tag{8}$$

In this way, the model will perform gradient descent on the episode scale according to Equation 9 to optimize the altruism factor of each module, and perform gradient descent on the time step scale to maximize R^{AS} simultaneously.

$$\begin{aligned}&\nabla_{\alpha_i} L_i^G(\theta_i^{new}) = \nabla_{\theta_i^{new}} L_i^G(\theta_i^{new}) \nabla_{\alpha_i}\theta_i^{new} \\ &= \mathbb{E}[\nabla_{\theta_i^{new}} \min(kA^G, clip(k, 1-\epsilon, 1+\epsilon)A^G)] \\ &\cdot [\nabla_{\theta_i^{old}} \log \pi_{\theta_i^{old}}(u_i|s)] A^{AS}_{\alpha_i,t}\end{aligned} \tag{9}$$

where old and new parameters of policies (θ_i^{old} and θ_i^{new}) are obtained by optimizing before and after in Equation 7, respectively. The gradient of optimization target $L_i^G(\theta_i^{new})$ is rewritten with respect to the altruism factor α_i through the chain rule and the Taylor series derivation. In this way, personalised altruism factors of modules can be optimized in training.

4.3 Observation and Action Space

The observation of module includes proprioceptive and environmental perception, i.e., the position vector P_i, the velocity vector v_i, the altruism factor α_i and active-passive connection relationship $\delta_{i,I\setminus\{i\}}$. Module can also sense the position P_j and velocity v_j of other modules j within the radius d. Although all modules share the same parameters of policy network when decentralized execution, each module act differently on its own unique observations.

We directly incorporate the connection/disconnection of active connectors into action space, which can be represented by a tuple $< \sigma_1, \sigma_2, \delta_{link} >$. The linking action of each module is represented by δ_{link}. The torques of two motion motors are σ_1 and σ_2 to drive module to move in any direction on the plane. In addition, the maximum movement speed v_{max} of modules is artificially limited.

4.4 Termination Criteria

The reconfiguration motion requires explicit termination criteria so that modular robot system can continue task using target configuration. When training or testing in simulator, we can directly judge whether the reconfiguration is completed based on global information provided by simulator, and terminate the reconfiguration process when specified time step is reached. However, in real deployment of modular robots, it is not suitable to directly specify the termination time step. At the same time, the global information of configuration is difficult to obtain, due to the limited communication and observation capabilities of modules.

The termination criteria is as follows. All modules run the same termination criteria. Consider two modules A and B in contact with each other. If the active connector of A is connected to the spherical shell of B, then A is said to be a child module of B and B is a parent module of A. Each module judges whether its local connection relationship is consistent with target configuration based on its local observations, and transmits in-position signal Ω to parent module when it matches and receives Ω of all child modules. In this way, Ω will be distributed step by step from the leaf modules(without child module) to the root module(without parent node). Then root module pass termination signal Ω_2 to child modules step by step. Each module that receives Ω_2 immediately terminates its reconfiguration process. Only one active connector per module means there is at most one loop per configuration. If a root or leaf module is absent, virtual ones are generated among modules on the ring. This asynchronous termination criterion avoids the dependence on real-time updates of global configuration information.

5 Simulation

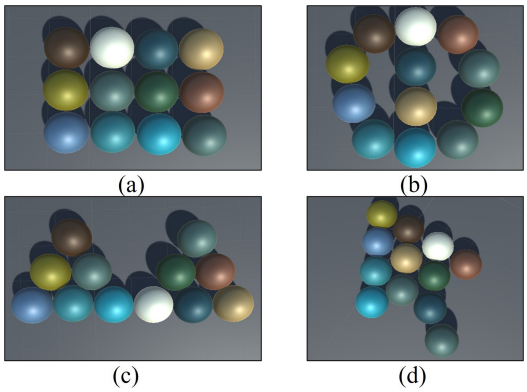

Figure 4: Configurations: (a) rectangle, (b) Φ, (c) triangle, and (d) arrow. They are paired as reconfiguration tasks.

5.1 Learning to Reconfigure

Referring to the setting of [Tucci *et al.*, 2018], we construct a series of desired two-dimensional configurations in unity-ml [Juliani *et al.*, 2018] (see Figure 4), as the basic environment for experiments. The geometric and topological features of these configurations, including hole-containing structures or bridging, are prone to conflict during reconfiguration and are suitable for demonstrating algorithm performance.

We implement the above method based on the PPO of Rllib [Liang *et al.*, 2018] and share the same policy in all modules through the parameter sharing [Christianos *et al.*, 2021],

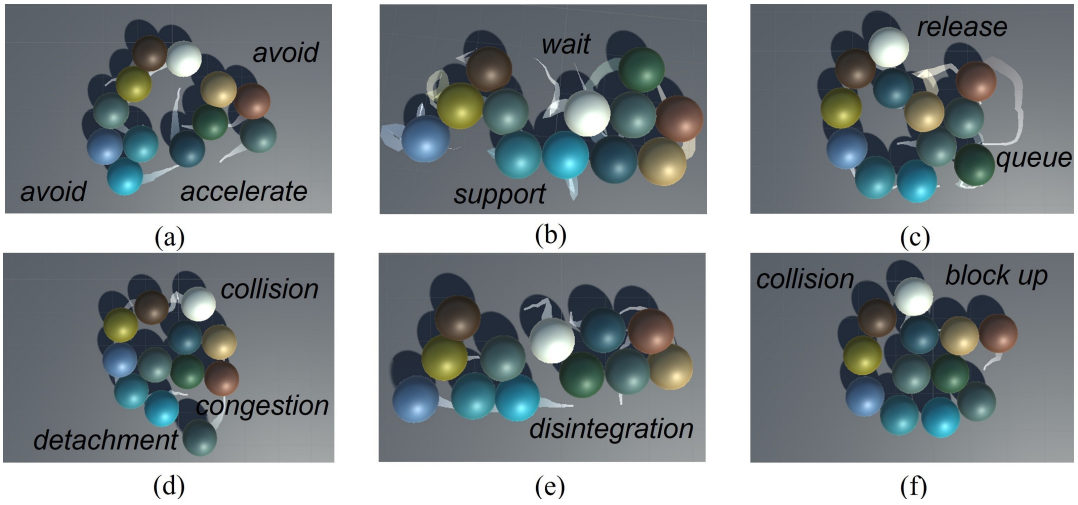

Figure 5: Visualisation of altruistic behaviour in the reconfiguration process. *Top*: Modules trained by APPO exhibit altruistic behaviors in Rectangle2Triangle, Rectangle2Arrow and Rectangle2Φ, respectively. *Bottom*: Conflicts arise with modules trained by baselines in different reconfiguration tasks.

	Rectangle2Triangle			Rectangle2Arrow			Rectangle2Φ		
	MC	AT	CR	MC	AT	CR	MC	AT	CR
IPPO-1	68.1±5.6	7.2±0.7	42.8±3.7	73.9±2.1	10.2±1.3	38.2±1.1	62.4±7.1	15.8±1.8	67.1±3.4
IPPO-2	65.8±3.7	10.9±2.0	69.8±2.6	78.6±5.7	13.4±1.9	63.9±2.6	57.2±4.9	0	74.9±2.7
CoPO	85.3±5.7	5.8±1.8	18.7±1.9	90.2±3.1	6.5±0.6	14.9±4.9	82.3±3.5	7.1±1.6	27.3±5.0
MFPPO	77.9±4.4	5.8±1.5	32.7±1.4	85.3±5.8	8.9±1.5	23.1±4.7	60.5±1.5	0	59.5±5.5
APPO	**92.1±2.3**	**5.7±0.9**	**11.6±0.8**	**92.3±1.6**	**5.2±1.7**	**10.8±0.5**	**90.8±6.9**	**5.4±1.4**	**19.4±0.6**

Table 1: Comparison of main results

which guarantees the scalability of the method. Since each module has separate local observations, modules take different actions when test.

5.2 Settings

We compare multiple baselines that can handle continuous motion control, including Independent Proximal Policy Optimization (IPPO), Mean Field Proximal Policy Optimization (MFPPO) and Coordinated Policy Optimization (CoPO). IPPO-1 maximizes the individual reward according to Equation 4, and similarly, IPPO-2 maximizes the mean-field reward. MFPPO is implemented based on MFRL[Yang *et al.*, 2018]. The state of neighbor modules is encoded into the value function. In this way, all neighbors of the module are equivalent to an agent and can perceive the state of neighbors. CoPO[Peng *et al.*, 2021] implements the sociological mechanism of homogeneous social value orientation and the two-layer optimization based on meta-gradient to simulate self-driven particles systems such as traffic flow.

Evaluation metrics include mean congruence(MC), average reconfiguration time(AT) and conflict rate(CR) . MC is the average of similarity of configuration obtained from reconfiguration method to target configuration, which represents the effectiveness of method. AT is the average time-consuming of modules that have completed the reconfiguration in test, which represents the efficiency of the method. CR is the proportion of modules that cause conflicts to the total number of modules. For easy detection in Unity, conflict detection are simplified into collision detection, loop formation detection and break detection. And modules that cause conflicts multiple times are only counted once.

5.3 Results

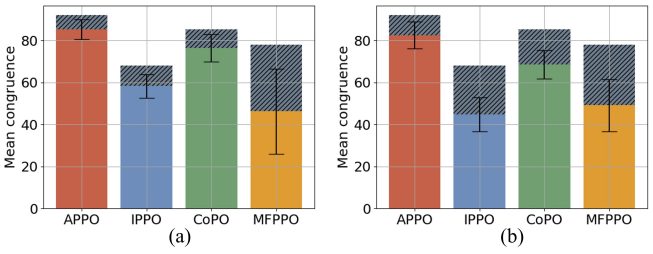

Figure 6: Results of generalization experiments. (a)Train on Rectangle2Triangle and test on Rectangle2Arrow. (b)Train on Rectangle2Triangle and test on Rectangle2Φ. The shaded region represents the performance gap between target task and source task.

It can be seen from Table 1 that compared with the baselines, our method shows advantages in terms of effectiveness and efficiency. In configuration pairs containing more complex configurations, our method still obtains high perfor-

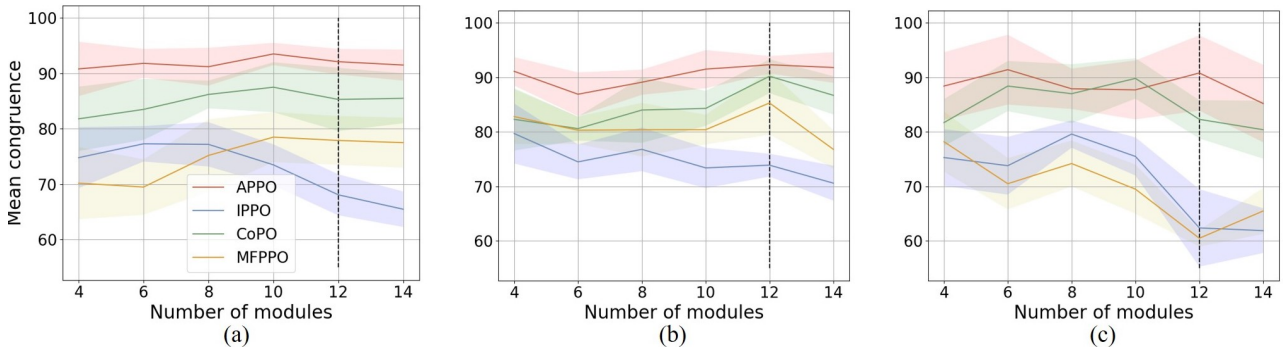

Figure 7: Performance varies with quantity. (a)Rectangle configuration to triangle configuration. (b)Rectangle to arrow. (c)Rectangle to Φ. The shaded region shows the standard error of the mean.

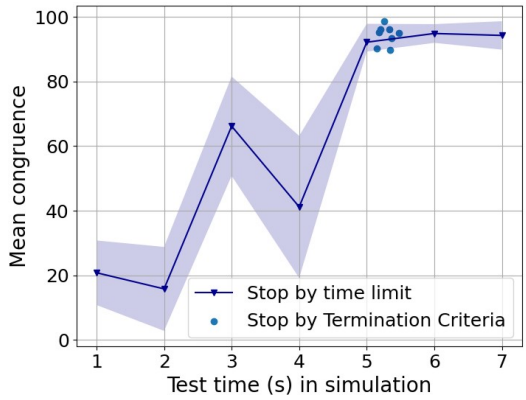

Figure 8: Mean congruence under different termination.

mance. Lower conflict rates guarantee reconfiguration performance. The personalised altruism factors enhance the diversity of modules and improve the efficiency of reconfiguration.

The absence of a cooperation mechanism in IPPO and MFPPO results in poor performance, primarily due to the increased likelihood of conflicts during the reconfiguration of complex configurations. The mean congruence of the CoPO is closed to our method, but the performance of the reconfiguration time is weaker. In non-zero-sum games, collaborative mechanisms based on non-strict zero-sum games do not fit the nature of the problem. The unweighted self-reward reduces the noise in the model training.

Figure 5 shows the reconfiguration process. The trained modules show different degrees of altruistic tendencies, and emerge various altruistic behaviors, such as fast passing, queuing, avoidance, etc. In contrast, snapshots of similar stages in the reconfiguration process of other methods show conflicts of freeform modular robots and their impact. This intuitively reveals our reconfiguration algorithm that by introducing the personalized altruism factor, facilitates diverse coordination among modules and thus spontaneously achieves conflict avoidance.

Figure 6 shows the performance of the trained model when tested on different configuration pairs. Our model shows good generalization ability on unseen configuration pairs, and strategies trained based on altruism factors can better avoid conflicts on different reconfiguration paths. This demonstrates the advantages of applying MARL to reconfiguration problem. The generalization of model can be further enhanced by the careful design of learning the commonalities of the configurations. Modules can handle zero-shot tasks in large-scale reconfiguration and even achieve adaptive reconfiguration.

And Figure 7 shows the performance of the model when the system has different numbers of modules. This variation of the number of modules could be caused by module communication or motor failure in a real environment, or adding more modules to perform more complex tasks. Our method exhibits robustness to the missing and adding of modules. This is owing to the fact that the algorithm runs distributed on autonomous modular robots.

Figure 8 shows the mean congruence to different termination time steps during test, where points represent the results obtained using the proposed termination criterion. The distributed termination criterion that can effectively terminate the reconfiguration process after reconfiguration is complete without external assistance or global information. This is important for realistic deployment of large-scale perception-limited freeform modular robot systems.

6 Conclusion

In this paper, a MARL algorithm based on altruism factors were designed to realize continuous reconfiguration control of freeform modular robots. Our approach showed general characteristics of modular robots with limited perception and parallel reconfiguration, demonstrating the research prospects for the automatic design of reconfigurable motion controllers. In addition, the introduction of personalised altruism factors and their optimisation are contributions to the automated solution of social dilemma problems.

As the first attempt to solve continuous reconfiguration problem of freeform modular robots and the first attempt to solve reconfiguration problem using MARL methods, our approach and experiments are still limited to two-dimensional reconfiguration. In the future, we will extend our algorithm in three-dimensional space and deploy it on the new freeform modular robot system that is under development.

Acknowledgements

This work was partially supported by the National Science Fund for Distinguished Young Scholars (No. 62025205), the National Natural Science Foundation of China (No. 62032020, 62102322) and the Natural Science Basic Research Program of Shaanxi (No. 2022JQ-623).

References

[Ahmadzadeh and Masehian, 2015] Hossein Ahmadzadeh and Ellips Masehian. Modular robotic systems: Methods and algorithms for abstraction, planning, control, and synchronization. *Artificial Intelligence*, 223:27–64, 2015.

[Barbosa *et al.*, 2020] João Vitor Barbosa, Anna H Reali Costa, Francisco S Melo, Jaime S Sichman, and Francisco C Santos. Emergence of cooperation in n-person dilemmas through actor-critic reinforcement learning. In *Adaptive Learning Workshop (ALA), AAMAS*, 2020.

[Bie *et al.*, 2018] Dongyang Bie, Yulin Wang, Yu Zhang, Che Liu, Jie Zhao, and Yanhe Zhu. Parametric l-systems-based modeling self-reconfiguration of modular robots in obstacle environments. *International Journal of Advanced Robotic Systems*, 15(1):1729881418754477, 2018.

[Bowles, 2006] Samuel Bowles. Group competition, reproductive leveling, and the evolution of human altruism. *science*, 314(5805):1569–1572, 2006.

[Chen and Burdick, 1998] I-Ming Chen and Joel W Burdick. Enumerating the non-isomorphic assembly configurations of modular robotic systems. *The International Journal of Robotics Research*, 17(7):702–719, 1998.

[Chen and Yang, 1998] I-Ming Chen and Guilin Yang. Automatic model generation for modular reconfigurable robot dynamics. *Journal of Dynamic Systems, Measurement, and Control*, 120(3):346–352, 1998.

[Christianos *et al.*, 2021] Filippos Christianos, Georgios Papoudakis, Muhammad A Rahman, and Stefano V Albrecht. Scaling multi-agent reinforcement learning with selective parameter sharing. In *International Conference on Machine Learning*, pages 1989–1998. PMLR, 2021.

[De Dreu *et al.*, 2010] Carsten KW De Dreu, Lindred L Greer, Michel JJ Handgraaf, Shaul Shalvi, et al. The neuropeptide oxytocin regulates parochial altruism in intergroup conflict among humans. *Science*, 328(5984):1408–1411, 2010.

[Gumbiner, 1996] Barry M Gumbiner. Cell adhesion: the molecular basis of tissue architecture and morphogenesis. *Cell*, 84(3):345–357, 1996.

[Günther *et al.*, 2020] Johannes Günther, Nadia M Ady, Alex Kearney, et al. Examining the use of temporal-difference incremental delta-bar-delta for real-world predictive knowledge architectures. *Frontiers in Robotics and AI*, 7:34, 2020.

[Hourany *et al.*, 2021] Edy Hourany, Christian Stephan, Abdallah Makhoul, Benoit Piranda, Bachir Habib, and Julien Bourgeois. Self-reconfiguration of modular robots using virtual forces. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6948–6953. IEEE, 2021.

[Hu *et al.*, 2022] Jiaheng Hu, Julian Whitman, Matthew Travers, and Howie Choset. Modular robot design optimization with generative adversarial networks. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 4282–4288. IEEE, 2022.

[Hughes *et al.*, 2018] Edward Hughes, Joel Z Leibo, Matthew Phillips, Karl Tuyls, Edgar Dueñez-Guzman, et al. Inequity aversion improves cooperation in intertemporal social dilemmas. *Advances in neural information processing systems*, 31, 2018.

[Juliani *et al.*, 2018] Arthur Juliani, Vincent-Pierre Berges, Ervin Teng, Andrew Cohen, Jonathan Harper, Chris Elion, Chris Goy, Yuan Gao, Hunter Henry, Marwan Mattar, et al. Unity: A general platform for intelligent agents. *arXiv preprint arXiv:1809.02627*, 2018.

[Kawano, 2020] Hiroshi Kawano. Parallel permutation for linear full-resolution reconfiguration of heterogeneous sliding-only cubic modular robots. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8281–8287. IEEE, 2020.

[Li *et al.*, 2019] Shuguang Li, Richa Batra, David Brown, Hyun-Dong Chang, Nikhil Ranganathan, Chuck Hoberman, et al. Particle robotics based on statistical mechanics of loosely coupled components. *Nature*, 567(7748):361–365, 2019.

[Liang *et al.*, 2018] Eric Liang, Richard Liaw, Robert Nishihara, Philipp Moritz, Roy Fox, Ken Goldberg, et al. Rllib: Abstractions for distributed reinforcement learning. In *International Conference on Machine Learning*, pages 3053–3062. PMLR, 2018.

[Liang *et al.*, 2020] Guanqi Liang, Haobo Luo, Ming Li, Huihuan Qian, and Tin Lun Lam. Freebot: A freeform modular self-reconfigurable robot with arbitrary connection point-design and implementation. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6506–6513. IEEE, 2020.

[Liu and Althoff, 2020] Stefan B Liu and Matthias Althoff. Optimizing performance in automation through modular robots. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4044–4050. IEEE, 2020.

[Lowe *et al.*, 2017] Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems*, 30, 2017.

[Mathews *et al.*, 2017] Nithin Mathews, Anders Lyhne Christensen, Rehan O'Grady, Francesco Mondada, and Marco Dorigo. Mergeable nervous systems for robots. *Nature communications*, 8(1):1–7, 2017.

[McKee *et al.*, 2020] Kevin R McKee, Ian Gemp, Brian McWilliams, Edgar A Duéñez-Guzmán, Edward Hughes,

and Joel Z Leibo. Social diversity and social preferences in mixed-motive reinforcement learning. *arXiv preprint arXiv:2002.02325*, 2020.

[Moussa *et al.*, 2021] Mohamad Moussa, Benoit Piranda, Abdallah Makhoul, and Julien Bourgeois. Cluster-based distributed self-reconfiguration algorithm for modular robots. In *International Conference on Advanced Information Networking and Applications*, pages 332–344. Springer, 2021.

[Nisbett and Kunda, 1985] Richard E Nisbett and Ziva Kunda. Perception of social distributions. *Journal of Personality and Social Psychology*, 48(2):297, 1985.

[Peng *et al.*, 2021] Zhenghao Peng, Quanyi Li, Ka Ming Hui, et al. Learning to simulate self-driven particles system with coordinated policy optimization. *Advances in Neural Information Processing Systems*, 34:10784–10797, 2021.

[Sawyer, 1966] Jack Sawyer. The altruism scale: A measure of co-operative, individualistic, and competitive interpersonal orientation. *American Journal of Sociology*, 71(4):407–416, 1966.

[Schelling, 1973] Thomas C Schelling. Hockey helmets, concealed weapons, and daylight saving: A study of binary choices with externalities. *Journal of Conflict resolution*, 17(3):381–428, 1973.

[Schulman *et al.*, 2017] John Schulman, Filip Wolski, Prafulla Dhariwal, et al. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[Seo *et al.*, 2019] Jungwon Seo, Jamie Paik, and Mark Yim. Modular reconfigurable robotics. *Annual Review of Control, Robotics, and Autonomous Systems*, 2:63–88, 2019.

[Spröwitz *et al.*, 2014] Alexander Spröwitz, Rico Moeckel, Massimo Vespignani, Stéphane Bonardi, and Auke Jan Ijspeert. Roombots: A hardware perspective on 3d self-reconfiguration and locomotion with a homogeneous modular robot. *Robotics and Autonomous Systems*, 62(7):1016–1033, 2014.

[Stastny *et al.*, 2021] Julian Stastny, Maxime Riché, Alexander Lyzhov, Johannes Treutlein, Allan Dafoe, and Jesse Clifton. Normative disagreement as a challenge for cooperative ai. *arXiv preprint arXiv:2111.13872*, 2021.

[Stoy, 2006] Kasper Stoy. Using cellular automata and gradients to control self-reconfiguration. *Robotics and Autonomous Systems*, 54(2):135–141, 2006.

[Swissler and Rubenstein, 2020] Petras Swissler and Michael Rubenstein. Fireant3d: a 3d self-climbing robot towards non-latticed robotic self-assembly. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3340–3347. IEEE, 2020.

[Tang *et al.*, 2021] Yunhao Tang, Tadashi Kozuno, Mark Rowland, et al. Unifying gradient estimators for meta-reinforcement learning via off-policy evaluation. *Advances in Neural Information Processing Systems*, 34:5303–5315, 2021.

[Tennenbaum *et al.*, 2016] Michael Tennenbaum, Zhongyang Liu, David Hu, and Alberto Fernandez-Nieves. Mechanics of fire ant aggregations. *Nature materials*, 15(1):54–59, 2016.

[Thalamy *et al.*, 2019] Pierre Thalamy, Benoit Piranda, and Julien Bourgeois. *Distributed self-reconfiguration using a deterministic autonomous scaffolding structure*. PhD thesis, UBFC, 2019.

[Thalamy *et al.*, 2020] Pierre Thalamy, Benoit Piranda, and Julien Bourgeois. 3d coating self-assembly for modular robotic scaffolds. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11688–11695. IEEE, 2020.

[Tu *et al.*, 2022] Yuxiao Tu, Guanqi Liang, and Tin Lun Lam. Freesn: A freeform strut-node structured modular self-reconfigurable robot-design and implementation. In *2022 IEEE International Conference on Robotics and Automation (ICRA). IEEE*, 2022.

[Tucci *et al.*, 2018] Thadeu Knychala Tucci, Benoit Piranda, and Julien Bourgeois. A distributed self-assembly planning algorithm for modular robots. In *International Conference on Autonomous Agents and Multiagent Systems*, 2018.

[Wang *et al.*, 2018] Jane X Wang, Edward Hughes, Chrisantha Fernando, Wojciech M Czarnecki, Edgar A Duéñez-Guzmán, and Joel Z Leibo. Evolving intrinsic motivations for altruistic behavior. *arXiv preprint arXiv:1811.05931*, 2018.

[Whitman *et al.*, 2020] Julian Whitman, Raunaq Bhirangi, Matthew Travers, and Howie Choset. Modular robot design synthesis with deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10418–10425, 2020.

[Yang *et al.*, 2018] Yaodong Yang, Rui Luo, Minne Li, Ming Zhou, Weinan Zhang, and Jun Wang. Mean field multi-agent reinforcement learning. In *International conference on machine learning*, pages 5571–5580. PMLR, 2018.

[Ye *et al.*, 2019] Zipeng Ye, Minjing Yu, and Yong-Jin Liu. Np-completeness of optimal planning problem for modular robots. *Autonomous Robots*, 43(8):2261–2270, 2019.

[Yim *et al.*, 2000] Mark Yim, David G Duff, and Kimon D Roufas. Polybot: a modular reconfigurable robot. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, volume 1, pages 514–520. IEEE, 2000.

[Zhao and Lam, 2022] Da Zhao and Tin Lun Lam. Snailbot: a continuously dockable modular self-reconfigurable robot using rocker-bogie suspension. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 4261–4267. IEEE, 2022.

[Zong *et al.*, 2022] Lijun Zong, Guanqi Liang, and Tin Lun Lam. Kinematics modeling and control of spherical rolling contact joint and manipulator. *IEEE Transactions on Robotics*, 2022.