2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Stimulate the Potential of Robots via Competition

Kangyao Huang[1], Di Guo[2], Xinyu Zhang[3], Xiangyang Ji[4], Huaping Liu[1†]

***Abstract*— It is common for us to feel pressure in a competition environment, which arises from the desire to obtain success comparing with other individuals or opponents. Although we might get anxious under the pressure, it could also be a drive for us to stimulate our potentials to the best in order to keep up with others. Inspired by this, we propose a competitive learning framework which is able to help individual robot to acquire knowledge from the competition, fully stimulating its dynamics potential in the race. Specifically, the competition information among competitors is introduced as the additional auxiliary signal to learn advantaged actions. We further build a Multiagent-Race environment, and extensive experiments are conducted, demonstrating that robots trained in competitive environments outperform ones that are trained with SoTA algorithms in single robot environment.**

I. Introduction

It has been demonstrated that competition can help improve the physical effort tasks [1]. For example, multiple race athletes often have the ability to achieve better results in competition that exceed their performance in individual training. Currently, competitive games have been well-studied in multi-agent reinforcement learning (MARL) field, like the professional-level performance of gaming agents implemented in StarCraft II [2], gFootball [3], and Honor of Kings [4]. In these studies, researchers pay more attention to the entire team performance, such as the win rate in a mixed-competitive game, converging to Nash Equilibrium (NE) in zero-sum games [5], or interactions and communication among cooperative agents [6]–[10]. However, the potential benefits of leveraging competition information to improve the individual performance is generally overlooked.

In this study, we focus on how to leverage the competition information among multiple robots to facilitate individual robot learning. Our approach aims to understand the connection between favorable actions and rewards. We propose a competitive learning framework which is able to help individual robot to acquire knowledge from the competition, fully stimulating its dynamics potential in the race. Specifically, the competition information among competitors is introduced as the additional auxiliary signal to enhance the learning process. As is shown in Fig.1, we find that under the competitive multi-agent environment, the individual robots can obtain higher rewards and continue to stimulate their dynamics potential compared with non-competitive environment, breaking through the baseline. The main contributions of the work are summarized as follows:

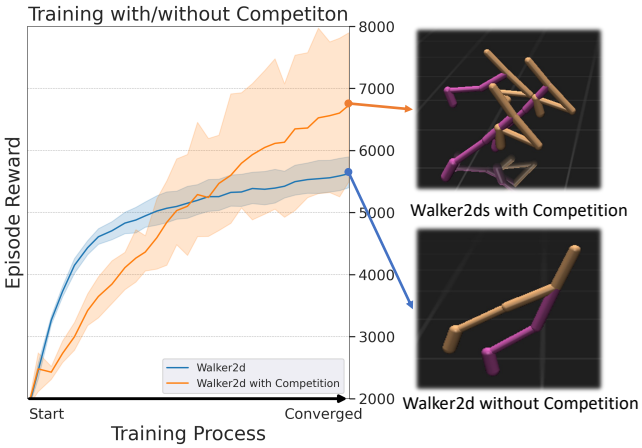


Fig. 1: Episode reward comparison between competitive and non-competitive `Walker2d` environment. The performance of `Walker2ds` trained with competition can reach 120% of the baseline.

- We propose a competitive reinforcement learning framework in which we use additional competition data among multiple robots to enhance the performance of an individual agent.
- We suggest that even by incorporating fundamental raw competitive data into the observation as supplementary auxiliary signals, and maintaining the reward mechanism unchanged, the untapped potential of individual robot can be further stimulated.
- We build a set of self-interest competition environments called Multiagent-Race [1]. We investigate how varying numbers of competitors and competitive signals influence the learning performance, and a 20% improvement has been gained over SoTA with the proposed framework (Table.I).

II. Related Work

A. Competition in Multi-agent Task

There are many types of competition tasks in MARL. Zero-sum games can be summarised in a linear program-

This work was supported by the National Natural Science Foundation of China under Grant 62025304.

[†]Huaping Liu is the corresponding author.
[1]K. Huang and H. Liu are with the Department of Computer Science and Technology, Tsinghua University, Haidian District, Beijing, 100084, P. R. China, huangky22@mails.tsinghua.edu.cn, hpliu@tsinghua.edu.cn
[2]D. Guo is with the School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing, China, guodi.gd@gmail.com
[3]X. Zhang is with the School of Vehicle and Mobility, Tsinghua University, Beijing, 100084, P. R. China, xyzhang@tsinghua.edu.cn
[4]X. Ji is with the Department of Automation, Tsinghua University, Beijing, 100084, P. R. China, xyji@tsinghua.edu.cn

[1]https://github.com/KJaebye/Multiagent-Race

ming (LP) formulation to address Nash equilibrium (NE) problem [11]. General-sum games might contain cooperation or team-level competition that might exist multiple NE points [12]. Many MARL studies are based on particle dynamics simulators [13], [14], such as MPE (Multi-Agent Particle Environments). Besides, a few are based on well-developed game engines [2], [4]. And some interactive tasks are based on robotics simulators. For example, researchers build a series of competitive and adversarial environments in MuJoCo [15], involving lots of physical confrontation environments, control strategies for physically simulated two-player competitive sports [16].

The above mentioned tasks are zero-sum or general-sum games that pay more attention to the entire team performance. However, in a self-interested game, each player strives to maximize their own utility or payoff without considering the overall outcome or cooperation with other players [17]. In the proposed work, we aim to facilitate individual robots in a continuous action space. There are a series of specialized algorithms dealing with continuous action spaces [18]–[24], and many benchmarks have been established [25]–[28]. In our work, we use Proximal Policy Optimization (PPO) [21] and its multi-agent variant to learn a continuous action task.

B. Learning from Competitive and Adversarial Data

Contrastive learning (CL) has been widely used in word and sentence embedding in NLP [29], image classification in CV [30], and implicit collaborative filtering in information retrieval (IR). It is able to extract meaningful representations through positive and negative data pairs. Furthermore, generative adversarial network (GAN) also has drawn significant attention in recent years [31] for selecting negatives, while the confrontation between the generator and discriminator may not converge to the ideal NE, and there is still potential for further exploration and improvement in the adversarial negative sampling method [32]. Moreover, some work introduces external disturbance from another adversarial robot to improve the robustness of robotic manipulation tasks [33].

Our idea shares fundamental similarities with the approach of adversarial learning and contrastive learning [34], [35]. We construct the competitive scenario that generates comparative data between opponents and learn features from it.

III. Problem Formulation

In general, agents can be trained to focus on specific skills by modifying the reward mechanism. In more cases, however, we do not want to change the reward mechanism since performance is somewhat sensitive to the reward, and inappropriate rewards might drown out correct reward signals. It comes to a problem: Can we acquire knowledge from raw comparative information, to surpass the results of normal training?

We first formulate the single-agent continuous action task as a Markov Decision Process (MDP), which can be described as a tuple $\langle S, A, P, R\rangle$. S is the state set of a system or environment. A is the action set that the agent can take. P is the state transition function, and $P\left(s'|s,a\right)$ represents the transition probability distribution of the system when transiting into the next state $s' \in S$ from state $s \in S$ after taking an action $a \in A$. R is the reward function. $r = R\left(s,a\right)$ is the reward given to agent after taking action $a \in A$ under $s \in S$. The aim is to find a policy π^*, to maximize the expected value of cumulative discount rewards:

$$\begin{aligned} \pi^* &= \arg\max_{\pi} J_\pi \\ J_\pi &= \sum_{t=0}^{\infty} \gamma^t \cdot \mathbb{E}\left[R(s_t, a_t)|s_0, \pi\right] \end{aligned} \tag{1}$$

where γ represents the extent to which the agent discounts future awards.

In this work, we consider the multi-agent scenario. We describe the corresponding MDP as a tuple $\langle \mathcal{S}, \mathcal{A}, P, R\rangle$ that includes N number of agents. $\mathcal{S}, \mathcal{A}$ are state and action tuples, respectively, where $\mathcal{S} = (S_1, S_2, \cdots, S_N)$, $\mathcal{A} = (A_1, A_2, \cdots, A_N)$. P and R remain the same as settings in MDP because agents are totally homogeneous and there is no physical interaction between competitors. In this regard, the objective function in (1) can be extended as

$$\hat{J}_\pi = \frac{1}{N} \sum_{i=1}^{N} \sum_{t=0}^{\infty} \gamma^t \cdot \mathbb{E}\left[R(s_t^i, a_t^i)|s_0^i, \pi\right] \tag{2}$$

where $s_t^i \in S_i$ and $a_t^i \in A_i$ are the state and action of the i-th agent. However, the above setting only facilitates the parallel exploration, but does not provide extra benefits due to the lack of interaction between agents.

In practical training, other agents for the i-th agent always can provide information which can be augmented to the state. We use o_t^i to denote the competitive information which can be observed from other agents, and may form a new state as

$$\bar{s}_t^i = [s_t^i, o_t^i]$$

and the objective function becomes

$$\bar{J}_{\bar{\pi}} = \frac{1}{N} \sum_{i=1}^{N} \sum_{t=0}^{\infty} \gamma^t \cdot \mathbb{E}\left[R(\bar{s}_t^i, a_t^i)|\bar{s}_0, \bar{\pi}\right] \tag{3}$$

and the optimization problem is changed as

$$\bar{\pi}^* = \arg\max_{\bar{\pi}} \bar{J}_{\bar{\pi}}$$

What we hope to address is therefore answer if the introduction of the extra state information o_t^i could be beneficial and lead to

$$\bar{J}_{\bar{\pi}^*} > J_{\pi^*}$$

If the results hold, then we may develop a set of new multi-agent competitive learning methods for the single agent. Please note that o_t^i can be raw measurement information or encoding features from observation of other agents.

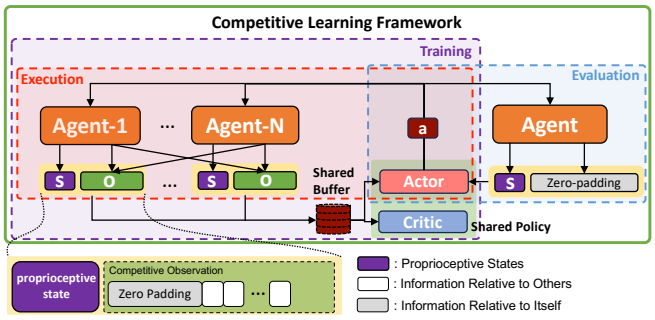


Fig. 2: Framework for learning knowledge from comparative information. Where a denotes actions, s represents the proprioceptive state, and o is the competitive observation.

IV. Methodology

A. Framework

We propose a framework to exploit competitive information among multiple agents, shown in Fig.2. The main purpose is to acquire knowledge from comparative data. We run a multi-agent competitive task to obtain additional comparable information and then distinguish positive or negative data for better training. It is akin to the contrasting representation between anchor/positive/negative samples. Using the `Walker2d` scenario as an illustration, shown in Fig.3, if a walker demonstrates exceptional performance, it must be the fastest participant in the race. Concurrently, the robot gathers distinctive relative details, like maintaining a consistently positive relative speed compared to others. This situation creates an auxiliary signal connecting high-speed running and favorable data (comprising state-action-reward pairs).

The framework includes mainly two parts: Training and Evaluation. We train robots with competitive observation during the training stage, while only proprioceptive state is required during the evaluation stage.

B. Policy Training & Experience Sharing

The policy training part conducts competitive tasks among multiple agents: agents collect full information involving proprioceptive state and competitive observation. The competitive information vector is a concatenation of relative observation against rivals.

We use the classical actor-critic algorithm PPO and its multi-agent variant [14] as the algorithmic benchmark because PPO has shown generally favorable results in continuous action control tasks. It becomes normal single-agent learning when there is only one robot in the race.

Since robots are completely homogeneous and exclusively self-interested, we adopt a shared policy among robots. During the training stage, data are sampled distributedly, but the policy is trained centrally since we only maintain one shared policy (actor) $\pi\left(a^i \mid \bar{s}^i; \boldsymbol{\theta}\right)$ for all agents, and one shared value (critic) $v(\bar{s}^i; \boldsymbol{\phi})$ to approximate value function $V_{\boldsymbol{\phi}}$, where $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ denote parameters of the shared policy/value network. Simultaneously, the experience replay buffer $\mathcal{D}$ is also shared across agents to aggregate all experience $\tau = \{\bar{s}, a, r\}$, $r = R(\bar{s}, a)$ is the reward given to agents after taking action a under newly designed state $\bar{s}$. In the training part, critic learns an optimal $\boldsymbol{\phi}^*$:

$$\boldsymbol{\phi}^* = \underset{\boldsymbol{\phi}}{\operatorname{argmin}} \frac{1}{\|\mathcal{D}\| T} \sum_{\tau \in \mathcal{D}} \sum_{t=0}^{T} \left(V_{\boldsymbol{\phi}}(\bar{s}_t) - R(\bar{s}_t, a_t)\right)^2 \quad (4)$$

Here the competitive observation o is introduced as an additional auxiliary signal where we have $\bar{s} = [s, o]$, helping the critic to get a more accurate estimation of experience. We also use GAE(General Advantage Estimation) [36] to estimate the advantages of actions $\hat{A}$. Then, we compute the objective function to update the policy:

$$\begin{aligned} L(\bar{s}, a, \bar{\alpha}, \theta^{new}, \theta^{old}) = \\ \min\Big(\frac{\pi_{\theta^{new}}(a \mid \bar{s})}{\pi_{\theta^{old}}(a \mid \bar{s})} \hat{A}^{\pi_{\theta^{old}}}(\bar{s}, \bar{\alpha}), \\ \operatorname{clip}\Big(\frac{\pi_{\theta^{new}}(a \mid \bar{s})}{\pi_{\theta^{old}}(a \mid \bar{s})}, 1-\epsilon, 1+\epsilon\Big) \hat{A}^{\pi_{\theta^{old}}}(\bar{s}, \bar{\alpha})\Big) \end{aligned} \quad (5)$$

We define the probability ratio between the shared new policy and the old one as $\frac{\pi_{\theta^{new}}(a|\bar{s})}{\pi_{\theta^{old}}(a|\bar{s})}$. $\bar{\alpha}$ is the action set of all agents, excluding the current agent. Hyperparameter ϵ is to limit the difference between old and new policies within a small range.

By sharing experience buffer $\mathcal{D}$, we attain a contrasting effect from competitive data during the critic training. This strengthens the association between rewards and positive/negative actions. The process described by (4) is similar to the step of labeling positive and negative samples for training in CL [30]. The difference lies in the fact that we utilize reinforcement learning reward measurement that naturally provides label-like signals to label competitive messages. Through incorporating the contrast provided by the multi-agent competition, competitive representation facilitates a clearer and more explicit understanding of the relationship between state and reward.

C. Robot Observation Construction

We tailor observation for the robot in order to appropriately introduce the competitive information. Different from the previous study that employs global information [13], we only consider the partial opponent observation that is more readily obtainable in real-world scenarios.

As illustrated at the bottom of Fig.2, we consider the new state of the i-th agent $\bar{s}^i = [s^i, o^i]$ to be a concatenation of proprioceptive state s^i and competitive observation o^i, or we call it competitive information which is obtained by comparing with other participants. The first part refers to the sensory feedback that robot receives from joints and muscles, providing measurements of motions and torques. The second part encompasses relative information that agents treat as competitive pressure. We denote $o^i = [o^{i1}, o^{i2}, ..., o^{ij}, ..., o^{iN}]$, where o^{ij} is the observation of the agent i regarding to agent j. Competitive observation can be any differentiable signals or any other comparable features.

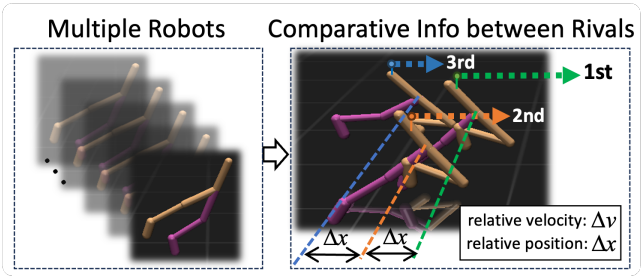


Fig. 3: The proposed Self-Interest Competition Environments **Multiagent-Race**. Three walker2d robots racing is illustrated as an example.

In this work, competitive information of the i-th robot is a concatenation of differences in velocities and displacements $o^{ij} = [x^j - x^i, v^j - v^i]$, which is related to all robots, including itself. x^i, x^j and v^i, v^j represent displacement and velocity of the i-th and j-th robot, respectively. Thus, the length of new state $\bar{s}$ depends on the number of participants in one race.

D. Evaluation

During the evaluation part, competitive messages are unnecessary, and agents exclusively rely on the proprioceptive state. To maintain alignment with the input dimensions of the neural network, we do zero-padding on the missing dimensions.

V. Emperical Results

A. Environment

We first build a series of self-interested and competitive environments named Multiagent-Race (**Race**). We extend the single-robot running to multi-robot racing. It is important to underline that environments are not the straightforward parallel environments of many solitary agents for sampling speed-up like what has been done in NVIDIA Isaac-Gym, but a real multiplayer competitive racing game, shown in Fig.3.

We provide six **Race** environments including `MultiAnt`, `MultiWalker2d`, `MultiHalfCheetah`, `MultiHopper`, `MultiSwimmer`, `MultiHumanoid`. In **Race**, self-interested agents face the same target: reach the maximum speed within a limited duration. Relative position and velocity along the desired orientation are treated as competitive information, fed to each robot. The rewarding for each type of robot maintains alignment with the Gym. We conduct experiments on four tasks: `MultiAnt`, `MultiHopper`, `MultiWalker2d`, and `MultiCheetah`. To avoid uncertainty, we take the average of over 10 trials using random seeds.

B. Baselines

We primarily consider several benchmarks for comparison to explore the effectiveness of our approach. $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ are used to represent shared parameters, then θ^i and ϕ^i denote separated parameters. X represents the robot number in **Race**. Our experiments are conducted in 2, 3, 4, and 5 robots race respectively, where $X = 2, 3, 4, 5$, to show the effect of different scales of competition.

S**A:** PPO training in a single-agent environment. We have $\pi\left(a \mid s; \boldsymbol{\theta}\right)$ and $v(s; \boldsymbol{\phi})$.

X**A-Sh-Decent:** training in X number of agents environment but with no competitive information. Here we define the knowledge-level information-sharing that agents share the same networks and experiences. The agent uses its decentralized local state as critic input. Where we have $\pi\left(a^i \mid s^i; \boldsymbol{\theta}\right)$ and $v(s^i; \boldsymbol{\phi})$.

X**A-Sh-Cent:** training in X number of agents environment but without competitive information. Policy and experience are shared. Agents use a centralized global state as critic input. We have $\pi\left(a^i \mid s^i; \boldsymbol{\theta}\right)$ and $v(s; \boldsymbol{\phi})$, in which s is the global observation, denotes the concatenation of $(s^1, ..., s^N)$.

X**A-Sp-Decent-Comp:** Training in X number of agents environment with competition. Every agent trains its own specific policy using local observation as value input. We can simply consider it to be a parallel training of SA with competitive observation. It is different from the proposed approach: although competitive information is applied to agents, no contrastive representation is formed because experiences are not shared. Therefore, the agent cannot improve itself by learning from others. Where we have $\pi\left(a^i \mid \bar{s}^i; \theta^i\right)$ and $v(\bar{s}^i; \phi^i)$.

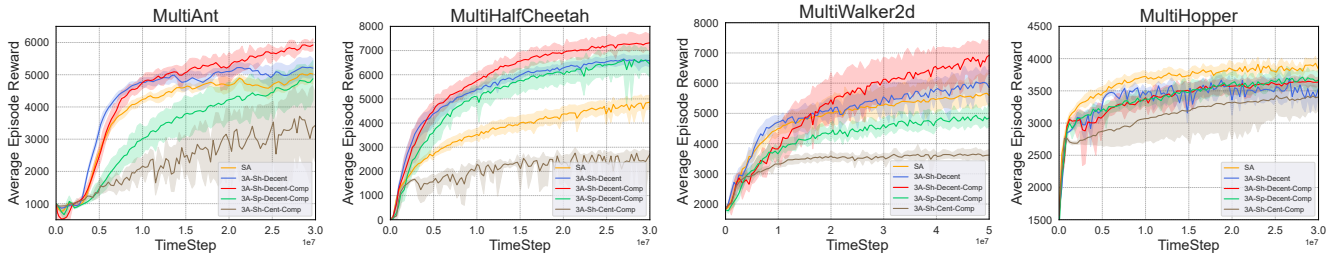
X**A-Sh-Decent-Noi:** Training in X number of agents environment with random noise. Previous studies argued that noise has a significant performance improvement on multi-agent learning rather than global information [37]. To validate that agents indeed acquire effective experience from competitive information, we replace with zero-mean random noise as a control group. Where we have $\pi\left(a^i \mid [s^i, n]; \boldsymbol{\theta}\right)$ and $v([s^i, n]; \boldsymbol{\phi})$, in which n denotes the noise.

X**A-Sh-Decent-Comp (Proposed):** Training in X number of agents environment with competitive input. Agents share policy and experience. Where we have $\pi\left(a^i \mid \bar{s}^i; \boldsymbol{\theta}\right)$ and $v(\bar{s}^i; \boldsymbol{\phi})$.

C. Experimental Details

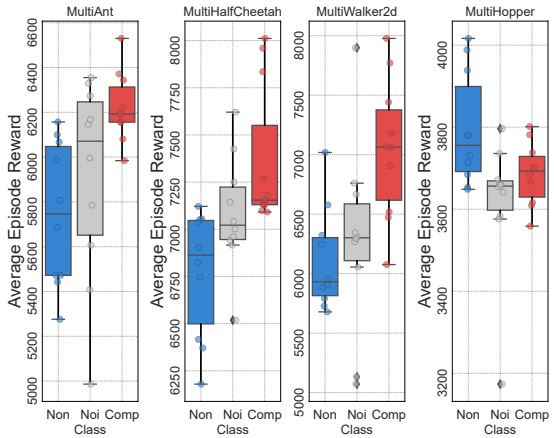
Networks: We use the same actor-critic network structure as stable-baseline3 [25] and TianShou [27]: 2 hidden layers MLP with 64 units each, and Tanh activation which is then fed into the Gaussian policy action out layer (except the `MultiCheetah` task where we use Beta distribution policy).

Algorithm Parameters: Routine hyperparameter settings: Adam optimizer with learning rate 0.0005 and linear learning rate decay strategy. The clipping parameter is 0.2, discounting factor is 0.995, and generalized advantage estimate parameter is 0.95. Our sampling number is larger than Tianshou and Stable-Baseline3 because we do not use the mini-batch update method for each iteration, but our optimization number is much less than theirs. However, this does not affect our comparison once they have converged.



(a) `MultiAnt` task (b) `MultiHalfCheetah` task (c) `MultiWalker2d` task (d) `MultiHopper` task

Fig. 4: Performance of different settings on 3-agent environments. SA: Yellow. 3A-Sh-Decent: Blue. 3A-Sh-Decent-Comp: Red. 3A-Sp-Decent-Comp: Green. 3A-Sh-Cent-Comp: Brown.



(a) (b) (c) (d)

Fig. 5: Comparison between 3A-Sh-Decent (**Non**: non-competitive information as inputs), 3A-Sh-Decent-Noi (**Noi**: noise information as inputs), and 3A-Sh-Decent-Comp (**Comp**: competitive information as inputs).

D. Role of Competition

We benchmark the proposed approach against baselines using 3-robot environments as a representative case. It is important to note that our method might yield better performance in environments with a different number of agents. Results are illustrated in Fig.4. Our proposed approach, training with competition and shared policy (3A-Sh-Decent-Comp), outperforms all baselines on task `MultiAnt`, `MultiHalfCheetah`, and `MultiWalker2d`. Besides, we notice that SA and 3A-Sh-Decent have a similar trend, but the latter slightly exceeds the former. This occurs because multiple agents gather more data leading to a more balanced distribution of data abundance and variance and increasing accuracy in the sampling results. Comparing 3A-Sh-Decent and 3A-Sh-Decent-Comp, we can conclude that robots learn valuable knowledge from additional comparative information. Especially in task `MultiAnt`, `MultiHalfCheetah`, and `MultiWalker2d`, we observe around 13%, 11%, and 16% improvement respectively.

However, it does not work on `MultiHopper`. In contrast, settings of competitive training seem to negatively affect the results. This phenomenon could be attributed to the `Hopper` task being overly simplistic, and the competitive observation might drown out the proprioceptive state. On the other hand, it appears that the dynamics of the `Hopper` has been thoroughly explored and understood, leading to little improvement under competition.

E. Ablation Studies

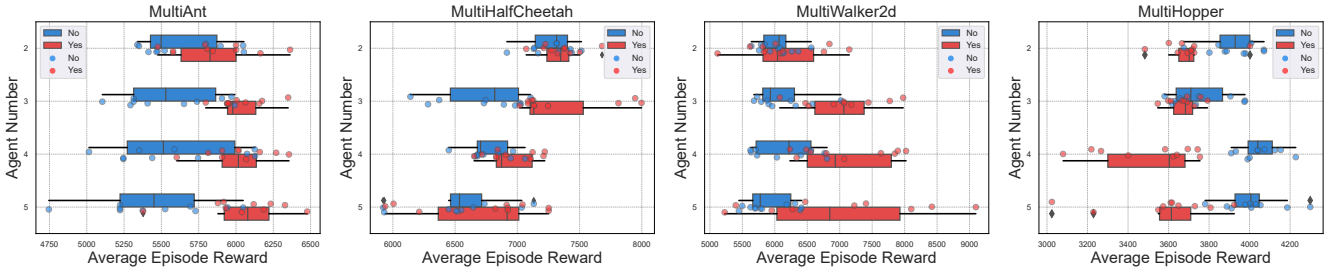
We perform a series of ablation studies to obtain a more profound understanding of how competitive scenarios yield valuable data for the learning process.

Policy and Buffer Sharing: We conducted controlled experiments on four environments depending on whether the policy network and experience reply buffer are shared or not, using the 3-robot **Race** game as an example. In 3A-Sp-Decent-Comp, robots enjoy independent policies, while robots share one policy and buffer in 3A-Sh-Decent-Comp. Results are shown in Fig.4. Irrespective of the varying tasks, agents with a shared network consistently demonstrate superior performance compared to their counterparts utilizing independent policy. This substantiates the explanation of (4) discussed above.

Moreover, as there is no interaction and cooperation between competitors, the value network need not coordinate the centralized global states of all robots. By contrast, centralized value inputs might confuse agents and harm the performance of evaluation, illustrated as brown lines in Fig.4.

Learning from Competition: There remains ongoing debate regarding the influence of external information on the effectiveness of training. Some studies have proved that adding more comprehensive information to the MARL task as additional states could improve the performance [14], [38]. However, others believe the network might treat external messages as noise that encourages more exploration [37], rather than learning valuable features from messages.

We employ random noise that is equivalent in length to the competitive information message as a controlled group, named XA-Sh-Decent-Noi. 3S-Sh-Decent is also taken as the comparison using fixed zero padding instead. We compare the performance of 3A-Sh-Decent, 3A-Sh-Decent-Noi, and 3A-Sh-Decent-Comp in Fig.5. Our results can corroborate the viewpoints presented in [37]: assigning noisy



(a) `MultiAnt` task (b) `MultiHalfCheetah` task (c) `MultiWalker2d` task (d) `MultiHopper` task

Fig. 6: Results on various numbers of agents with/without competitive info.

TABLE I: Comparison with SoTA PPO Baselines. Task1: `Ant`, Task2: `Walker2d`, Task3: `HalfCheetah`, Task4: `Hopper`. A: **OpenAI** [21], B: **Stable-Baselines3** [25], C: **Tianshou** [27]. XA-Sh-Decent-Comp is the proposed approach.

	SOTA PPO Baselines			SA	XA-Sh-Decent				3A-Decent-Noi	XA-Sh-Decent-Comp			
	A	B	C		2	3	4	5		2	3	4	5
1		1327±452	3900±850	4993±326	5468±371	5552±441	5987±512	5728±237	5781±365	5926±456	5960±207	6056±382	**6140±302**
2	3424	3479±822	4896±704	5579±488	6107±463	5902±665	6491±637	5711±441	6311±412	5876±1019	**7094±945**	6683±1379	6634±2039
3	1669	5819±663	7337±1508	4988±334	7284±289	6706±519	6671±293	6505±569	7063±363	**7342±213**	7197±498	6790±277	6873±702
4	2316	2410±10	3128±413	3834±313	3913±187	3701±201	**4059±165**	4002±205	3679±89	3715±54	3688±173	3598±267	3606±77

observations as inputs enhances the network's exploration capability, consequently leading to higher rewards. Simultaneously, we also prove that agents can acquire knowledge from competitive observation, leading to a higher reward surpassing those of the noise experiments.

VI. Discussion

We find there exists an unavoidable correlation between the results of games, the complexity of robots, and the number of racers involved. To explore how the number of racers affects the results, we conduct experiments with different numbers of participants, shown in Fig.6.

We compare the outcomes of utilizing competitive information in different environments and varying numbers of runners in **Race** tasks. For the sake of fairness, we record data after convergence. The results shown in Fig.6 indicate that training with competitive pressure can promote performance regardless of racer number, except `Hopper` because we believe its potential has been fully explored. Besides, in Table.I, we also compare the experimental results with the state-of-the-art PPO baselines. Our proposed method makes a great improvement on `Walker2d` and `Ant`, then a slight improvement on `HalfCheetch` task.

Furthermore, we discovered that on more challenging tasks like more complex robots, the advantages of utilizing competition are more pronounced. For example, robot `Ant` simulates in the three-dimensional environment which is more complex, robots achieve higher running speeds with an increase in the number of competitors. However, for agents with simpler structures, indiscriminately increasing the number of competitors could potentially lead to a decrease on final rewards. Declines are observed on 2D robots `Hopper`, `HalfCheetah`, and `Walker2d`. This is because robots with simple structures and straightforward dynamics can easily reach their ability boundary, while robots with complex structures and challenging tasks require more sophisticated controllers. We obtain the best performance on 2-`HalfCheetah`, 3-`Walker2d`, 5-`Ant`, and single-`Hopper`.

In addition, an excessive number of competitors, however, introduce an excessive amount of competitive information, resulting in an increase in the observation dimension. The additional signals representing competition could overshadow the proprioceptive signals of the agent. This leads to challenges in robot learning when the number of runners becomes excessive.

Although our work primarily focuses on PPO as the core algorithm, our framework is adaptable to any other on-policy multi-agent algorithms. On-policy algorithms, by not relying on past experiences, can establish precise comparative features within a single batch of data sampled by the current policy. The framework cannot adapt to off-policy algorithms. The preservation of old experiences is common practice in off-policy methods, leading to the absence of a baseline value for evaluating the comparison of information along a long-term training process.

VII. Conclusion

In this work, we propose a competitive learning framework that can stimulate robots' potential. Our method effectively leverages the competition, allowing for increased exploration and exploitation of comparative data, even using raw data as additional input. Through extensive experimentation, we have empirically demonstrated that, under competitive learning among multiple self-interested racers, our method can surpass the majority of SoTA benchmarks, including Tianshou and Stable-Baselines3. In the future, how proprioceptive signal to additional signal ratio influences the training can be further explored. Besides, more experiments can be implemented to verify the effectiveness in the real world.

REFERENCES

[1] B. C. DiMenichi and E. Tricomi, "The power of competition: Effects of social motivation on attention, sustained physical effort, and learning," *Frontiers in Psychology*, vol. 6, 2015.

[2] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev, J. Oh, D. Horgan, M. Kroiss, I. Danihelka, A. Huang, L. Sifre, T. Cai, J. P. Agapiou, M. Jaderberg, A. S. Vezhnevets, R. Leblond, T. Pohlen, V. Dalibard, D. Budden, Y. Sulsky, J. Molloy, T. L. Paine, C. Gulcehre, Z. Wang, T. Pfaff, Y. Wu, R. Ring, D. Yogatama, D. Wünsch, K. McKinney, O. Smith, T. Schaul, T. Lillicrap, K. Kavukcuoglu, D. Hassabis, C. Apps, and D. Silver, "Grandmaster level in StarCraft II using multi-agent reinforcement learning," *Nature*, vol. 575, no. 7782, pp. 350–354, 2019.

[3] K. Kurach, A. Raichuk, P. Stanczyk, and M. Zajac, "Google research football: A novel reinforcement learning environment," *AAAI 2020 - 34th AAAI Conference on Artificial Intelligence*, pp. 4501–4510, 2020.

[4] H. Wei, J. Chen, X. Ji, H. Qin, M. Deng, S. Li, L. Wang, W. Zhang, Y. Yu, L. Liu, L. Huang, D. Ye, Q. Fu, and W. Yang, "Honor of Kings Arena: an Environment for Generalization in Competitive Reinforcement Learning," 2022. [Online]. Available: http://arxiv.org/abs/2209.08483

[5] S. Leonardos, G. Piliouras, and K. Spendlove, "Exploration-Exploitation in Multi-Agent Competition: Convergence with Bounded Rationality," *Advances in Neural Information Processing Systems*, vol. 31, pp. 26 318–26 331, 2021.

[6] J. K. Gupta, M. Egorov, and M. Kochenderfer, "Cooperative Multi-agent Control Using Deep Reinforcement Learning," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10642 LNAI, pp. 66–83, 2017.

[7] W. Cheng and W. Meng, "Collaborative algorithm of workpiece scheduling and AGV operation in flexible workshop," *Robotic Intelligence and Automation*, 2024.

[8] ——, "An efficient genetic algorithm for multi AGV scheduling problem about intelligent warehouse," *Robotic Intelligence and Automation*, vol. 43, no. 4, pp. 382–393, 2023.

[9] K. Huang, J. Chen, J. Oyekan, and X. Zhang, "Bio-inspired Multi-agent Model andăOptimization Strategy forăCollaborative Aerial Transport," *Lecture Notes in Electrical Engineering*, vol. 801 LNEE, pp. 591–598, 2022.

[10] K. Huang, J. Chen, and J. Oyekan, "Decentralised aerial swarm for adaptive and energy efficient transport of unknown loads," *Swarm and Evolutionary Computation*, vol. 67, 2021.

[11] Y. Yang and J. Wang, "An Overview of Multi-Agent Reinforcement Learning from Game Theoretical Perspective," 2020. [Online]. Available: http://arxiv.org/abs/2011.00583

[12] S. Gronauer and K. Diepold, "Multi-agent deep reinforcement learning: a survey," *Artificial Intelligence Review*, vol. 55, no. 2, pp. 895–943, 2022.

[13] I. M. Ryan Lowe, YI WU, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Advances in Neural Information Processing Systems*, 2017.

[14] 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," 2021. [Online]. Available: http://arxiv.org/abs/2103.01955

[15] T. Bansal, J. Pachocki, S. Sidor, I. Sutskever, and I. Mordatch, "Emergent complexity via multi-agent competition," *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, 2018.

[16] J. Won, D. Gopinath, and J. Hodgins, "Control strategies for physically simulated characters performing two-player competitive sports," *ACM Transactions on Graphics*, vol. 40, no. 4, pp. 1–11, 2021.

[17] J. Blumenkamp and A. Prorok, "The Emergence of Adversarial Communication in Multi-Agent Reinforcement Learning," 2020. [Online]. Available: http://arxiv.org/abs/2008.02616

[18] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," *31st International Conference on Machine Learning, ICML 2014*, vol. 1, pp. 605–619, 2014.

[19] T. Erez, N. Heess, J. J. Hunt, D. Lillicrap, Timothy P.Silver, A. Pritzel, Y. Tassa, and D. Wierstra, "Continuous Control With Deep Reinforcement Learning," *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*, 2016.

[20] P. Abbeel, M. Jordan, S. Levine, P. Moritz, and J. Schulman, "Trust Region Policy Optimization," *32nd International Conference on Machine Learning, ICML 2015*, vol. 3, pp. 1889–1897, 2015.

[21] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," 2017. [Online]. Available: http://arxiv.org/abs/1707.06347

[22] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," *35th International Conference on Machine Learning, ICML 2018*, vol. 5, pp. 2976–2989, 2018.

[23] S. Fujimoto, H. Van Hoof, and D. Meger, "Addressing Function Approximation Error in Actor-Critic Methods," *35th International Conference on Machine Learning, ICML 2018*, vol. 4, pp. 2587–2601, 2018.

[24] L. Zhang, Y. Feng, R. Wang, Y. Xu, N. Xu, Z. Liu, and H. Du, "Efficient experience replay architecture for offline reinforcement learning," *Robotic Intelligence and Automation*, vol. 43, no. 1, pp. 35–43, 2023.

[25] A. Raffin, "RL Baselines3 Zoo," *GitHub repository*, 2020. [Online]. Available: https://github.com/DLR-RM/rl-baselines3-zoo

[26] S. Huang, Q. Gallouédec, F. Felten, A. Raffin, R. F. J. Dossa, Y. Zhao, R. Sullivan, V. Makoviychuk, D. Makoviichuk, C. Roumégous, J. Weng, C. Chen, M. Rahman, J. G. M. Araújo, G. Quan, D. Tan, T. Klein, R. Charakorn, M. Towers, Y. Berthelot, K. Mehta, D. Chakraborty, A. KG, V. Charraut, C. Ye, Z. Liu, L. N. Alegre, J. Choi, and B. Yi, "openrlbenchmark," 2023. [Online]. Available: https://github.com/openrlbenchmark/openrlbenchmark

[27] J. Weng, H. Chen, D. Yan, K. You, A. Duburcq, M. Zhang, Y. Su, H. Su, and J. Zhu, "Tianshou: A Highly Modularized Deep Reinforcement Learning Library," *Journal of Machine Learning Research*, vol. 23, 2022.

[28] Y. Fujita, P. Nagarajan, T. Kataoka, and T. Ishikawa, "ChainerRL: A deep reinforcement learning library," *Journal of Machine Learning Research*, vol. 22, 2021.

[29] T. Gao, X. Yao, and D. Chen, "SimCSE: Simple Contrastive Learning of Sentence Embeddings," *EMNLP 2021 - 2021 Conference on Empirical Methods in Natural Language Processing, Proceedings*, pp. 6894–6910, 2021.

[30] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," *37th International Conference on Machine Learning, ICML 2020*, vol. PartF16814, pp. 1575–1585, 2020.

[31] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative Adversarial Networks: An Overview," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 53–65, 2018.

[32] J. Wang, L. Yu, W. Zhang, Y. Gong, Y. Xu, B. Wang, P. Zhang, and D. Zhang, "IRGAN: A minimax game for unifying generative and discriminative information retrieval models," *SIGIR 2017 - Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 515–524, 2017.

[33] P. Jian, C. Yang, H. L. Di Guo, and F. Sun, "Adversarial Skill Learning for Robust Manipulation," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2021-May, pp. 2555–2561, 2021.

[34] L. Xu, J. Lian, W. X. Zhao, M. Gong, L. Shou, D. Jiang, X. Xie, and J.-R. Wen, "Negative Sampling for Contrastive Representation Learning: A Review," 2022. [Online]. Available: http://arxiv.org/abs/2206.00212

[35] P. H. Le-Khac, G. Healy, and A. F. Smeaton, "Contrastive Representation Learning: A Framework and Review," *IEEE Access*, vol. 8, pp. 193 907–193 934, 2020.

[36] J. Schulman, P. Moritz, S. Levine, M. I. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*, 2016.

[37] J. Hu, S. Hu, and S.-w. Liao, "Policy Regularization via Noisy Advantage Values for Cooperative Multi-agent Actor-Critic methods," 2021. [Online]. Available: http://arxiv.org/abs/2106.14334

[38] C. S. de Witt, T. Gupta, D. Makoviichuk, V. Makoviychuk, P. H. S. Torr, M. Sun, and S. Whiteson, "Is Independent Learning All You Need in the StarCraft Multi-Agent Challenge?" 2020. [Online]. Available: http://arxiv.org/abs/2011.09533