Prediction-based Hierarchical Reinforcement Learning for Robot Soccer

Zongyuan Zhang,[1] Tianyang Duan, [1] Zekai Sun, [1] Xiuxian Guan, [1] Junming Wang, [1]
Hongbin Liang, [2] Yong Cui, [3] Heming Cui [1]
[1] Department of Computer Science, The University of Hong Kong, Hong Kong, China.
[2] School of Transportation and Logistics, Southwest Jiaotong University, Chengdu, China.
[3] Department of Computer Science and Technology, Tsinghua University, Beijing, China.
{zyzhang2, tyduan, zksun, xxguan, jmwang, heming}@cs.hku.hk, hbliang@swjtu.edu.cn, cuiyong@tsinghua.edu.cn

***Abstract*—Robot soccer as a complex mixed cooperative-competitive task presents many challenges to multi-agent reinforcement learning (MARL), such as assigning long-term credits and effective exploration in high-dimensional and continuous state-action spaces. We propose Prediction-based Hierarchical Reinforcement Learning (P-HRL) for robot soccer. P-HRL consists of a coach for soccer tactics and a robot controller for robot motion control. To comprehensively evaluate the performance of P-HRL, we design various key performance indicators for robot soccer such as ball possession rate. Experimental results demonstrate that P-HRL has a better performance than the baseline MATD3, with 52% win rate, 22% draw rate, and 26% loss rate.**

***Index Terms*—Multi-agent Reinforcement Learning, Hierarchical Reinforcement Learning, Robot Soccer**

I. Introduction

Robot soccer has been developed to facilitate deploying multi-agent systems (MAS) from virtual environments to the real world [1]. In robot soccer, MAS controls a team of robots to cooperate and score goals. Several leagues on robot soccer have been successfully organized in recent years, such as RoboCup, IEEE Very Small Size Soccer (IEEE VSSS). Many research topics based on robot soccer have received extensive attention, such as the Internet of Things(IoT) [2] and path planning [3].

Multi-agent reinforcement learning (MARL) has achieved outstanding success on cooperative problems [4]. MARL controls the actions of multiple agents to maximize the return by continuously interacting with the environment. MARL enables end-to-end autonomous learning and can continuously improve its performance without reliance on domain knowledge [5]. The development of a MARL-based approach in robot soccer has the potential to outperform other non-reinforcement learning approaches [6]. However, robot soccer, as a cooperative-competitive task that requires controlling robots in the real world, brings more challenges to MARL. We identify the following challenges to MARL in robot soccer.

First, the problem of assigning long-term credit to multiple robots makes it difficult for MARL to learn to collaborate in robot soccer. Agents learn optimal policies by rewards interacting with the environment [7]. However, scoring is a sparse reward, and it is difficult to identify the contribution of each player to the goal. Previous work [8], [9] addressed the problem of reward sparsity by reward shaping. However, reward shaping still struggles to address the long-term credit assignment problem. The contribution of players to goals is difficult to measure and is associated with many factors such as the position, direction, speed of the ball and players. These factors increase the hyperparameter space of the reward function weights, making reward shaping difficult to achieve.

Second, the inherent high-dimensional and continuous state-action space of robot soccer increases the complexity of training. MARL faces the curse of dimensionality in robot soccer [10]: the dimensionality of the state-action space grows exponentially with the degrees of freedom and the number of robots. Consequently, it is hard for agents to effectively explore the environment and learn optimal policies.

Inspired by human soccer, where coaches choose tactics for players on the field, the paper designs a *coach* for robot soccer. The coach guides agents at the tactical level to help MARL devise different strategies for various opponents and on-field situations. Hierarchical reinforcement learning (HRL) provides a framework for improving each agent through coach. HRL learns the same task at multiple time scales, and the policies at each time scale jointly determine the behaviors of agents [11].

In the paper, we propose Prediction-based Hierarchical Reinforcement Learning (P-HRL), which tackles the problem of long-term credit assignment and exploration problem of high-dimensional continuous state-action. P-HRL consists of two hierarchical parts: a coach for soccer tactics and a robot controller for robot motion control. The coach decomposes the soccer game into a series of subtasks, while the robot controllers control robot to achieve these tasks. For the first challenge, P-HRL provides different subtasks and corresponding rewards for each robot. For the second challenge, the hierarchical structure allows the coach to focus only on learning soccer tactics and the robot controllers are dedicated to the subtasks. This division of labor enables both the coach and robot controllers to train within a simplified, lower-dimensional space.

We evaluate the P-HRL by simulating robot soccer games under IEEE VSSS rules [12]. The results show that P-HRL has better performance than the state-of-the-art baseline MATD3. In matches against the baseline, P-HRL has 52% win rate, 22% draw rate and 26% loss rate. P-HRL showed better cooperation with a 70.25% possession rate compared to 17.14% for baseline.

II. Prediction-based Hierarchical Reinforcement Learning (P-HRL)

Figure 1 shows the workflow of P-HRL. The key idea of P-HRL is to design a coach to guide the robot controller at the level of soccer tactics. We design a team policy in the coach and guided the robot controller to execute it.

A. Problem Formalization

The entire process of robot soccer can be formalized as a decentralized partially observable Markov Decision Process (Dec-POMDP). Specifically, it is formally defined as a tuple $< \mathcal{A}, \mathcal{S}, \mathcal{U}, \mathcal{O}, P, R, \gamma >$, where $\mathcal{A} = \{1, 2, \ldots, n\}$ denotes the set of agents (i.e., robots), $\mathcal{S}$ denotes the state space, $\mathcal{U}$ denotes the action space, P denotes the state-transition function, R denotes the reward function and γ denotes the discount factor. At each time step t, each agent selects an action $u_t^i \in \mathcal{U}$ following the policy $\pi_i (u \mid o_{i,t})$ to form a joint action $\mathbf{u_t} = (u_{1,t}, u_{2,t}, ..., u_{n,t})$. In the following paper, we refer to the agent's policy π_i as the individual policy, in parallel with the team policy π_i^T in coach. Note that each agent can only receive partial observable observations $o_{i,t} \in \mathcal{O}$ instead of the state $s_t \in \mathcal{S}$. Each agent then receives the reward $r_t = R(s_t, \mathbf{u_t})$ and the environment follows the state-transition function $P(s \mid s_t, \mathbf{u_t})$ into the next state s_{t+1}. The goal of each agent is to maximize the discounted accumulated rewards $\mathbb{E}\left[\sum_{k=t}^{T} \gamma^{k-t} r_k\right]$, where T is the maximum time step of a game.

B. Team policy in Coach

Inspired by human soccer, we aim to design a coach, which guides agents at the tactical level to adapt to different opponents and field situations. Formally, the coach is defined as a team policy $\pi^T (g_i, r_i^g \mid o_i)$ where $g_i \in \mathcal{G}$ is a subtask for each robot and r_i^g is a subtask reward. Each subtask is started and terminated using an initiation condition I_g and a termination condition β_g, respectively.

A basic principle of soccer tactics is to maintain possession of the ball because the team that has possession of the ball has the opportunity to attack and score, otherwise it can only passively defend. Thus, we divide subtasks into two types $\mathcal{G} = \{g^A, g^D\}$, where g^A is offensive subtask and g^D is defensive subtask. When our team has possession of the ball, the coach starts an offensive subtask; when the opponent gets possession of the ball, the coach starts a defensive subtask. A ball possession recognizer $f^{bp} : o_i \to \{0, 1\}$ is usually built into the robot to determine whether it has possession of the ball. If not, it can be determined by $f^{bp}(o_i) = \mathbf{1}[\|pos_i - pos_b\| < \epsilon]$, where $\mathbf{1}[\cdot]$ denotes the indicator function, pos_i and pos_b denote the position vectors of the robot and the ball, respectively, and ϵ is a threshold.

The offensive subtask aims to secure possession of the ball and score a goal, while the defensive subtask aims to get possession of the ball to prevent the opponent's attack. In the offense subtask, the coach encourages the ball-control robots (i.e., robots that satisfy $f^{bp}(o_i) = 1$) to score a goal and encourages the remaining robots (i.e., robots that satisfy $f^{bp}(o_i) = 0$) to explore the hotspot area to accelerate the convergence of individual strategies. In the defensive subtask, the coach encourages all robots to intercept the ball to get possession of the ball. The subtask reward r_i^g is defined as encouraging the robot to go to the target position:

$$r_i^g = \frac{pos_i - pos_g}{\|pos_i - pos_g\|} \cdot v_i, \qquad pos_g = \begin{cases} pos_b & \text{if} \quad g = g^A \text{ and } f^{bp}(o_i) = 1 \\ pos_z & \text{if} \quad g = g^A \text{ and } f^{bp}(o_i) = 0 \\ pos_b & \text{if} \quad g = g^D \end{cases} \tag{1}$$

where v_i is the robot speed vector and pos_z denotes the centroid of the high reward region, which is defined in the next subsection. Table I shows the definition of initiation conditions I_g and termination conditions β_g. Note that after T_p time step, the offensive subtask is restarted if our robots have possession of the ball.

TABLE I
Definition of initiation and termination conditions

	Offensive subtask g^A	Defensive subtask g^D
Initiation condition I_g	$\exists o_i \ s.t. \ f^{bp}(o_i) = 1$	$\forall o_i \ s.t. \ f^{bp}(o_i) = 0$
Termination condition β_g	(1) $\forall o_i \ s.t. \ f^{bp}(o_i) = 0$ (2) After T_p time step.	$\exists o_i \ s.t. \ f^{bp}(o_i) = 1$

C. High-reward Area Prediction

We determine hotspot areas by predicting the ball position, because the reward function in robot soccer is directly correlated with the ball position. However, the complexity of observation presents a challenge for soccer position prediction. The observation contains three types of data: data related to the home team (i.e. our robots), data related to the away team (i.e. the opponent robots), and data related to the ball. In games, the position of the soccer is determined by the actions of players in both teams. A single network structure cannot effectively model the data under observation because the policy of the home player and the policy of the away player are mutually independent. Thus, we propose a soccer position prediction network (SPPN), which is a Mixture of Experts Network [13] and consists of the following three components:

Expert network: To effectively solve the problem of data independence in the observation, we use three expert networks in SPPN. Each expert focused on modeling only one type of

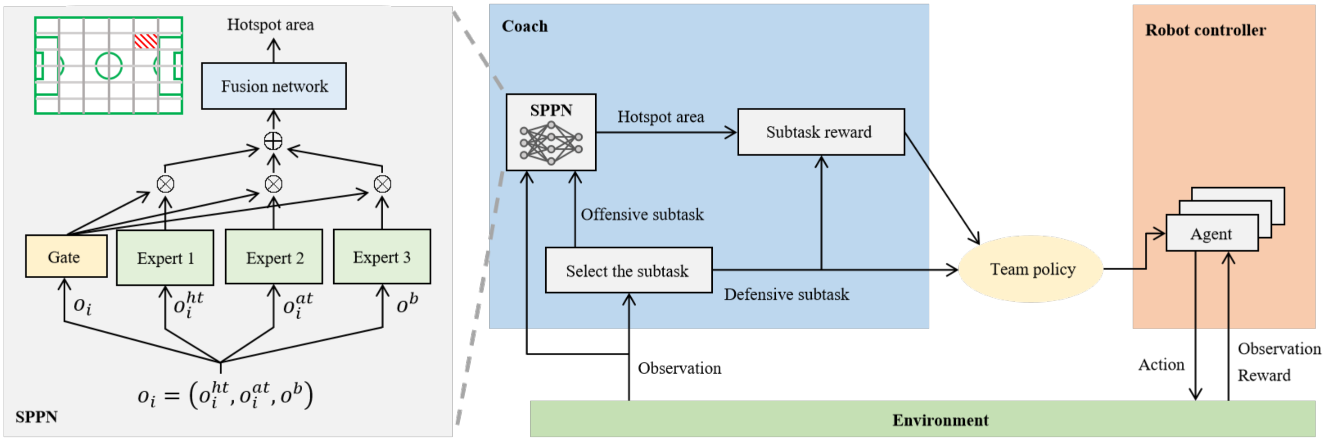


Fig. 1. The workflow of P-HRL. At each time step, the coach receives observation and starts (or continues) a subtask. If it is a new offensive subtask, the coach calculates the reward based on the predicted area by SPPN. In SPPN, the observation of the home team o_i^{ht}, away team o_I^{at} and ball o_i^b are input separately to different expert networks. Then, the coach calculates the corresponding subtask reward based on the subtask. In the robot controller, each agent outputs an action based on observation and reward.

data. We divided the data into three types and input them into expert networks respectively.

Gating network: Gating network is used to gate the output of expert networks. This allows modeling of the complex relationship between the three types of data in the observation.

Fusion network: Fusion network is used to map the output features of the gating network with the prediction results.

We divide the soccer field into several equal rectangular areas, and SPPN is used to predict the area where the ball will be located after T_p time steps. Formally, given an observation $o_i = \left(o_i^{ht}, o_i^{at}, o^b\right)$, where o_i^{ht}, o_i^{at} and o_i^b is the part of the observation related to the home team players, away team players and ball, respectively. The prediction area of SPPN can be formulated as follow:

$$\begin{aligned} pos_z = h\left(\sum_{k=1}^{3} g\left(o_i\right) f_k\left(o_i^k\right)\right), \\ \text{where } g\left(o_i\right) = softmax\left(f\left(o_i\right)\right), \\ \text{and } o_i^1 = o_i^{ht},\ o_i^2 = o_i^{at},\ o_i^3 = o^b \end{aligned} \tag{2}$$

where $f_i\left(\cdot\right)$ is the expert network, $g\left(\cdot\right)$ is the gating network, and $h\left(\cdot\right)$ is the fusion network.

D. Individual Policy in Robot Controller

In the robot controller, we modify MADDPG [14] to be used in the hierarchy structure of P-HRL to learn individual policies. The objective of each agent is to complete subtask g_i and maximize goal scores. Thus, the action-value function Q_{φ_i} is defined as:

$$Q_{\varphi_i}\left(o_{i,t}, \mathbf{u}_t\right) = \mathbb{E}\left[\sum_{k=t}^{T} \gamma^{k-t}\left(r_{i,k}^{g} + r_k\right) \mid g_i, o_{i,t}, \mathbf{u}_t\right] \tag{3}$$

where φ_i 1 denotes the neural network parameters of the action value function. We update the action-value function Q_{φ_i} by using the tuple $(o_i, g_i, r_i^g, \mathbf{u}, r, o_i')$ in the replay buffer $\mathcal{D}_\infty$ to minimize the loss:

$$\begin{aligned} \mathcal{L}\left(\theta^{Q_i}\right) = \mathbb{E}_{\left(o_i, g_i, r_i^g, \mathbf{u}, r, o_i'\right)\sim\mathcal{D}_1}\left[\left(Q_{\varphi_i}\left(o_i, \mathbf{u}\right) - y\right)^2\right], \\ \text{where } y = r + r_i^g + \gamma Q_{\varphi_i'}\left(o_i', \mathbf{u}'\right)|_{u_j' = \pi_{\phi_j'}\left(o_j, g_j\right)} \end{aligned} \tag{4}$$

where $Q_{\varphi_i'}$ is the target action-value function with network parameter φ_i', and each action u_j' in the joint action $\mathbf{u}'$ is given by the target individual policy $\pi_{\phi_j'}\left(o_j, g_j\right)$. Using the action-value function Q_{φ_i}, the gradient of individual policy network ϕ_i used to optimize the deterministic policy is shown below:

$$\begin{aligned} \bigtriangledown J\left(\phi_i\right) = \mathbb{E}_{o_i, u_{j\neq i}, g_i, r_i^g \sim\mathcal{D}}[\bigtriangledown_{\phi_i}\pi_{\phi_i}\left(o_i, g_i\right)\cdot \\ \bigtriangledown_{u_i} Q_{\varphi_i}\left(o_i, \mathbf{u}, g_i\right)|_{u_i = \pi_{\phi_i}\left(o_i, g_i\right)}] \end{aligned} \tag{5}$$

Target networks φ_i' and ϕ_i' are periodically updated by networks φ_i and ϕ_i, respectively.

E. Training Method

To allow the robot controller to focus on completing the subtasks assigned by the coach, P-HRL uses parallel training of the coach and robot controller. Algorithm 1 describes the training parallel for P-HRL.

To train the SPPN in coach and the robot controller simultaneously, we use two replay buffers $\mathcal{D}_1$ and $\mathcal{D}_2$. In an episode, the coach selects a subtask g (lines 5 to 10). Then, the robot controller controls robots to complete the subtask (lines 12). P-HRL performs parameter updates on the robot controller using samples from $\mathcal{D}_1$ (lines 13 to 14). Every T_p time step, P-HRL stores the observation o_t into replay buffer $\mathcal{D}_2$ for the training of SPPN. P-HRL uses the samples in $\mathcal{D}_2$ to generate the training set and labels to perform parameter updates on the SPPN (lines 15 to 18).

Algorithm 1 Parallel training for P-HRL

Parameter: g^A: Offensive subtask. g^D: Defensive subtask.

```
Initialize replay buffer $\{\mathcal{D}_1, \mathcal{D}_2\}$, SPPN, and robot controller.
for episode = $1$ to $M$ do
  Reset the environment.
  for $t$ = $0$ to $T_{max} - 1$ and not $occur\_goal\_scoring$ do
    if our robots are in possession. then
      $g \leftarrow g^A$
      Use SPPN to predict area $pos_z$.
    else
      $g \leftarrow g^D$
    end if
    while $g$ dose not satisfy its termination condition defined in table I do
      Execute the joint action $\mathbf{u}$, calculate subtask reward $r_g$ using Eq 1, observe the external reward $r$ and next observation $o'$.
      Store $(o, g, r_g, \mathbf{u}, r, o')$ in replay buffer $\mathcal{D}_1$.
      Sample mini-batches from $\mathcal{D}_1$ and update robot controller by Eq. 4 and Eq. 5.
      if $t$ mod $Tp = 0$ then
        Store observation $o$ in replay buffer $\mathcal{D}_2$.
      end if
      Sample mini-batches from $\mathcal{D}_2$ and update SPPN by cross-entropy loss function.
    end while
  end for
end for
```

III. Evaluation

Our experiments are based on rSoccer [15] - a framework for studying reinforcement learning in robot soccer. Specifically, we use the IEEE VSSS multi-agent environment to simulate a robot soccer scenario. IEEE VSSS is a robotic soccer competition in which two teams of three robots compete against each other. According to the rules, the robots are controlled remotely by computers without human intervention; The robot is fixed-size (0.08 m × 0.08 m) with wheels but no hardware for dribbling or kicking the ball.

We use rSoccer built-in algorithm as the opponent. The hidden layer size for both the Critic and Actor networks is set to 64, with learning rates of 1×10^{-4} for each. The batch size is 1024, the discount factor γ is 0.95 and the soft update parameter is 0.01. The initial noise level is defined as 0.2, with a noise decay rate of 5×10^{-7}, and a minimum noise level of 0.05. The expert networks and fusion network of SPPN consist of three fully connected layers with 128 neurons. The training lasts until convergence (about 1×10^6 steps). The soccer field is equally divided into 6×6 rectangular areas for predicting hotspot areas. T_p is set to 30 time steps.

The evaluation focused on these questions:

- RQ1: How is P-HRL compared to baseline in terms of end-to-end performance?
- RQ2: How well does the P-HRL adapt to the new opponent compared to the baseline?
- RQ3: How does the coach contribute to the overall system?

A. Evaluation Metrics

Apart from the direct win-loss indicator of goals scored, we also define the following key performance indicators (KPIs) applicable to robot soccer:

Ball Possession Rate The last robot to control the ball is defined as having possession of the ball. We use the clock time method to measure a team's possession rate. For a team, the possession rate can be calculated as the sum of the time step that all robots on the team have possession of the ball as a percentage of the total time step of the game. Historically, it has been believed that higher possession is associated with scoring advantage [16]–[18].

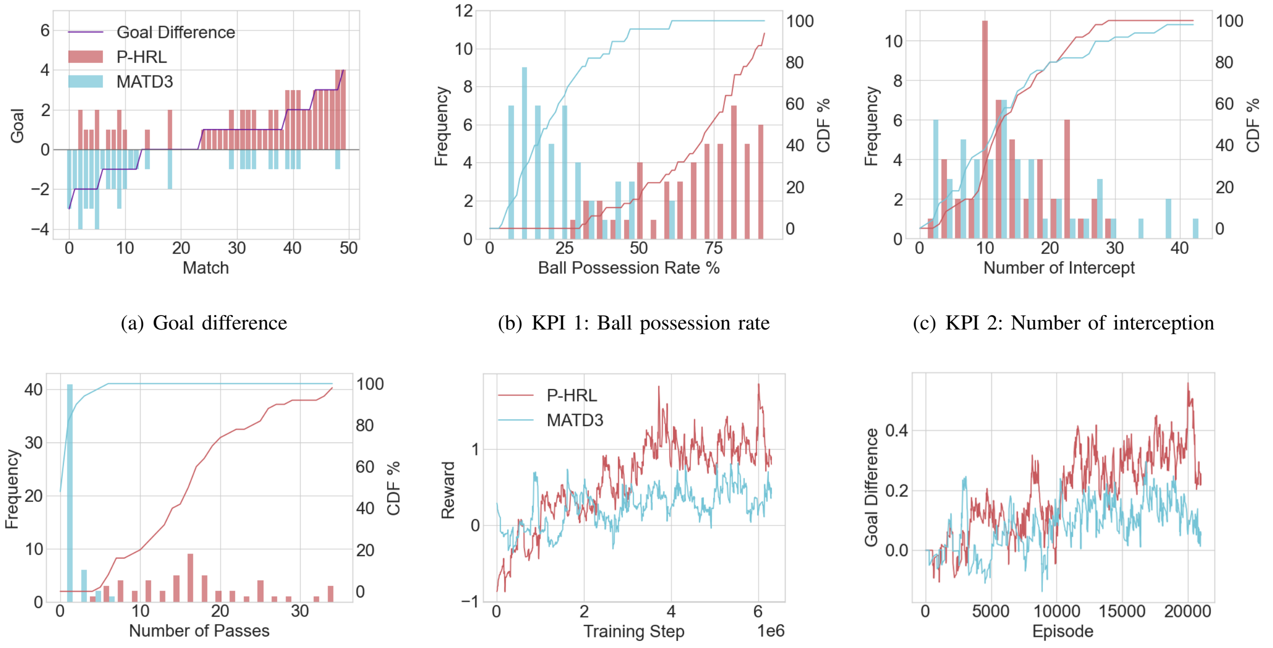
Number of Passes Passing is defined as the transfer of possession from one player to another player of the same team. The number of passes made by a team is the sum of all passes made by all players. The number of passes is a key indicator of the coordination between players.

Number of Interception Interception is defined as the transfer of possession from one player to an opponent's player. For a team, the number of interceptions is counted as the sum of interceptions made by all players. Frequent interceptions mean that players are more motivated to compete for ball possession.

B. End-to-end Performance

To evaluate the end-to-end performance, we use MATD3 [19], a popular MARL method that has been used in many multi-agent cooperative tasks, as the baseline. In the above environment, P-HRL controls one team of three robots and MATD3 controls another team for soccer matches. Inspired by Robust Adversarial Reinforcement Learning (RARL) [20], we further train P-HRL and MATD3 in a mutual confrontation. Based on the converged model output, two sets of agents compete against each other and are alternately trained: In the first stage, P-HRL is trained while keeping the MATD3 unchanged; In the next stage, P-HRL remains unchanged and MATD3 is trained. Repeat this sequence until convergence. The results are shown in Figure 2(e), 2(f).

For evaluation, the robots controlled by both algorithms play 50 matches in the environment as described above, each match lasting 2000 time steps. The results of the matches are shown in Figure 2. Out of the total 50 matches, P-HRL wins 26 matches, loses 13 matches and ties 11 matches. Figure 2(a) shows that P-HRL scores more goals than MATD3. Figure 2(b), 2(c), 2(d) show the differences in KPIs. P-HRL outperforms MATD3 in ball possession rate and the number of passes. P-HRL has an average of 70.25% possession and 14.32 passes per game, which is much higher than MATD3. In terms of interception rates, no significant differences are found.



(a) Goal difference

(b) KPI 1: Ball possession rate

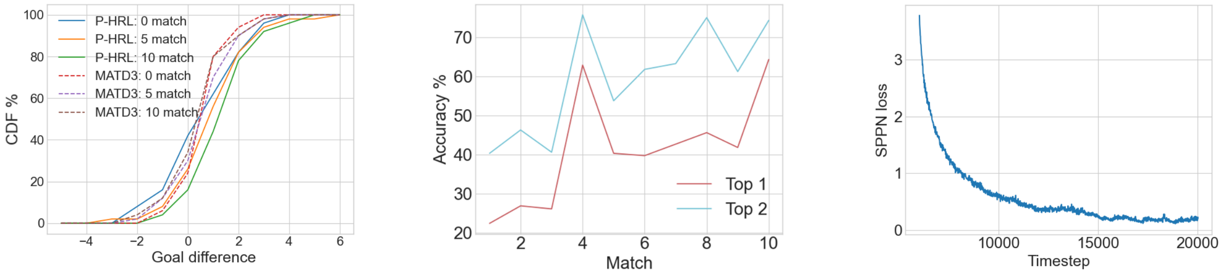
(c) KPI 2: Number of interception

(d) KPI 3: Number of passes

(e) Reward in training

(f) Goal difference in training

Fig. 2. (a) Results of P-HRL vs. MATD3 over 50 matches. The purple dash line represents the difference between the two (P-HRL minus MATD3). (b-d) The histogram shows the distribution of the average KPI per match; the dashed line is the cumulative distribution function (CDF). (e-f) Change of reward and goal difference with training. Smoothed by exponential moving average (EMA) with exponential smoothing constant K=0.95.



(a) Goal difference

(b) SPPN accuracy

(c) SPPN loss

Fig. 3. Online learning results of P-HRL and MATD3 against MADDPG in 10 matches. (a) shows the CDF of the goal difference (P-HRL minus MADDPG and MATD3 minus MADDPG) in 50 matches for evaluation at 3 different checkpoints. (e-f) show the prediction accuracy and loss of the SPPN module of P-HRL.

C. Ability Against New Opponents

We investigate the performance of the model against new opponents. This simulates that in real robot training, the opponent's policy is unknown during the training phase. Once on the field, the robots are supposed to fine-tune their policies to counter their opponents.

We use MADDPG [14], a widely used actor-critic-based MARL algorithm, as the new opponent. Use the P-HRL and MATD3 models trained in the previous chapter as starting points. Let the two play against MADDPG separately to evaluate their adaptability. Note that neither model has been trained with MADDPG as the opponent, therefore, it is safe to say that they both play against a new opponent.

A total of 50 matches are played, each lasting 2000 timesteps. MATD3 trains the entire network, while P-HRL only trains the coach with the robot controllers unchanged. To avoid systematic errors, we save three checkpoints: before the start of training (match 0), halfway through training (match 5), and at the end of training (match 10). At these checkpoints 50 matches without training are played for evaluation.

The results in Figure 3(a) show that the overall number of goals scored is higher for P-HRL than MATD3 and slightly increases at 3 checkpoints in time sequence, while no significant trend is seen for MATD3. One possible reason for the increase could be the convergence of SPPN. Figure 3(b), 3(c) show the loss and accuracy of SPPN during training. It can be observed

TABLE II
MATCH RESULTS FOR DIFFERENT TYPES OF COACH (VS. MATD3 IN 50 MATCHES)

Coach	*Team Score : Opponent Score*	*Acc Top 1*	*Acc Top 2*
SPPN	**2.40±1.78 : 1.48 ± 1.05**	**0.70±0.06**	**0.85±0.04**
NN	2.15±1.22 : 1.55 ± 1.31	0.59±0.06	0.80±0.05
RN	1.85±1.35 : 2.15 ± 1.39	0.06±0.11	0.12±0.06

that the loss decreases rapidly and converges approximately within 15000 time steps. The accuracy of prediction increases within 10 matches and reaches a maximum of 66.17% top 1 accuracy and 82.35% top 2 accuracy, which is close to the model tested in end-to-end performance experiment with long training time (average of 67.49% top 1 accuracy and 83.00% top 2 accuracy over 50 matches). P-HRL has better overall performance than MATD3 in ball possession rate, the number of interceptions and the number of passes.

D. Ablation Study

To understand the effectiveness of the coach in the overall system, we keep the robot controllers unchanged and replaces the SPPN with the trained neural network (NN) model and the random nearest (RN) model, respectively. The structure of the NN coach is similar to SPPN with only one expert in the prediction network. The observed data are fed into the unique expert without being segmented. The RN coach selects two random adjacent areas of the current areas as the prediction results. For each setting, 50 games of 2000 time steps are performed. As shown in Table II, SPPN outperforms NN and RN in terms of accuracy of both goal difference (average 0.92 goal difference) and prediction accuracy (average 70% top 1 accuracy and 85% top 2 accuracy).

IV. CONCLUSION

In the paper, we identify two challenges for MARL applications in robot soccer: the long-term credit assignment, and high and continuous state-action spaces. We propose P-HRL, a hierarchical method in which the coach decomposes the soccer game into a series of subtasks, and the robot controller controls the motions of robots to complete these subtasks. Experiments demonstrate that P-HRL has better end-to-end performance than baseline, including more goals scored and better KPIs, and has a stronger ability to adapt to new opponents.

REFERENCES

[1] M. Asada, P. Stone, M. Veloso, D. Lee, and D. Nardi, “Robocup: A treasure trove of rich diversity for research issues and interdisciplinary connections [tc spotlight],” *IEEE Robotics & Automation Magazine*, vol. 26, no. 3, pp. 99–102, 2019.

[2] E. Antonioni, V. Suriani, F. Riccio, and D. Nardi, “Game strategies for physical robot soccer players: a survey,” *IEEE Transactions on Games*, vol. 13, no. 4, pp. 342–357, 2021.

[3] S. Macenski, F. Martín, R. White, and J. G. Clavero, “The marathon 2: A navigation system,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 2718–2725.

[4] A. Perrusquía, W. Yu, and X. Li, “Multi-agent reinforcement learning for redundant robot control in task-space,” *International Journal of Machine Learning and Cybernetics*, vol. 12, no. 1, pp. 231–241, 2021.

[5] L. Buşoniu, R. Babuška, and B. D. Schutter, “Multi-agent reinforcement learning: An overview,” *Innovations in multi-agent systems and applications-1*, pp. 183–221, 2010.

[6] Z. Chen, H. Zhang, D. Guo, S. Jia, X. Fang, Z. Huang, Y. Wang, P. Hu, L. Wen, L. Chen *et al.*, “Champion team paper: Dynamic passing-shooting algorithm of the robocup soccer ssl 2019 champion,” in *Robot World Cup*. Springer, 2019, pp. 479–490.

[7] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, “Playing atari with deep reinforcement learning,” 2013.

[8] M. Abreu, L. P. Reis, and H. L. Cardoso, “Learning high-level robotic soccer strategies from scratch through reinforcement learning,” in *2019 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*. IEEE, 2019, pp. 1–7.

[9] T. F. de Medeiros, R. d. A. Marcos, and T. Yoneyama, “Deep reinforcement learning applied to ieee very small size soccer strategy,” in *2020 Latin American Robotics Symposium (LARS), 2020 Brazilian Symposium on Robotics (SBR) and 2020 Workshop on Robotics in Education (WRE)*. IEEE, 2020, pp. 1–6.

[10] T. T. Nguyen, N. D. Nguyen, and S. Nahavandi, “Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications,” *IEEE transactions on cybernetics*, vol. 50, no. 9, pp. 3826–3839, 2020.

[11] S. Pateria, B. Subagdja, A.-h. Tan, and C. Quek, “Hierarchical reinforcement learning: A comprehensive survey,” *ACM Computing Surveys (CSUR)*, vol. 54, no. 5, pp. 1–35, 2021.

[12] H. F. Bassani, R. A. Delgado, J. N. d. O. L. Junior, H. R. Medeiros, P. H. Braga, M. G. Machado, L. H. Santos, and A. Tapp, “A framework for studying reinforcement learning and sim-to-real in robot soccer,” *arXiv preprint arXiv:2008.12624*, 2020.

[13] D. Eigen, M. Ranzato, and I. Sutskever, “Learning factored representations in a deep mixture of experts,” 2013.

[14] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. Pieter Abbeel, and I. Mordatch, “Multi-agent actor-critic for mixed cooperative-competitive environments,” *Advances in neural information processing systems*, vol. 30, 2017.

[15] F. B. Martins, M. G. Machado, H. F. Bassani, P. H. Braga, and E. S. Barros, “rsoccer: A framework for studying reinforcement learning in small and very small size robot soccer,” in *Robot World Cup*. Springer, 2022, pp. 165–176.

[16] C. Lago and R. Martín, “Determinants of possession of the ball in soccer,” *Journal of sports sciences*, vol. 25, no. 9, pp. 969–974, 2007.

[17] J. Lago-Ballesteros and C. Lago-Peñas, “Performance in team sports: Identifying the keys to success in soccer,” *Journal of Human kinetics*, vol. 25, no. 1, pp. 85–91, 2010.

[18] E. J. Parziale and P. A. Yates, “Keep the ball! the value of ball possession in soccer,” *Reinvention: an International Journal of Undergraduate Research*, vol. 6, no. 1, pp. 1–24, 2013.

[19] J. Ackermann, V. Gabler, T. Osa, and M. Sugiyama, “Reducing overestimation bias in multi-agent domains using double centralized critics,” 2019.

[20] L. Pinto, J. Davidson, R. Sukthankar, and A. Gupta, “Robust adversarial reinforcement learning,” in *International Conference on Machine Learning*. PMLR, 2017, pp. 2817–2826.