Reinforcement Learning Within the Classical Robotics Stack: A Case Study in Robot Soccer

Adam Labiosa[1*], Zhihan Wang[2*], Siddhant Agarwal[2], William Cong[1], Geethika Hemkumar[2], Abhinav Narayan Harish[1], Benjamin Hong[1], Josh Kelle[2], Chen Li[1], Yuhao Li[1], Zisen Shao[1], Peter Stone[2,3†], Josiah P. Hanna[1†]

***Abstract*— Robot decision-making in partially observable, real-time, dynamic, and multi-agent environments remains a difficult and unsolved challenge. Model-free reinforcement learning (RL) is a promising approach to learning decision-making in such domains, however, end-to-end RL in complex environments is often intractable. To address this challenge in the RoboCup Standard Platform League (SPL) domain, we developed a novel architecture integrating RL within a classical robotics stack, while employing a multi-fidelity sim2real approach and decomposing behavior into learned sub-behaviors with heuristic selection. Our architecture led to victory in the 2024 RoboCup SPL Challenge Shield Division. In this work, we fully describe our system's architecture and empirically analyze key design decisions that contributed to its success. Our approach demonstrates how RL-based behaviors can be integrated into complete robot behavior architectures.**

I. INTRODUCTION

In the field of robotics, reinforcement learning (RL) has enabled complex and impressive behaviors [1]–[3]. Despite the advances in RL, the training and deployment of RL for strategic decision-making on physical robots in partially observable, real-time, dynamic, and multi-agent environments remains a challenge.

One particular domain that exhibits these challenges is the RoboCup Standard Platform League (SPL) [4]. The SPL is part of the RoboCup initiative, which has driven advances in robotics over the past three decades [5]. In the SPL, teams of 5 or 7 humanoid NAO robots compete in soccer games. Each robot must be fully autonomous and act in real-time; and the presence of teammates and adversaries makes the domain highly dynamic. In addition, the competitive environment requires teams to quickly adapt to different opponents and improve their strategy between and within matches. Teams participating in the SPL typically rely on a classical robot behavior architecture with complex hand-coded behaviors, and RL has had little use at the behavior level.

Toward the use of RL in partially observable, real-time, dynamic, and multi-agent environments, we introduce an RL-based robot architecture and training framework that we evaluate in the RoboCup SPL domain. Using this architecture, our joint team across two universities, WisTex United, participated in and won the 2024 RoboCup SPL Challenge Shield Division. Over 8 games we won 7 and outscored opponents 39-7. To the best of our knowledge, our system represents the first successful use case of RL for high-level decision-making in the SPL domain. While specific to the SPL competition, our system design provides insights for roboticists seeking to apply RL in domains of similar complexity.

Our architecture is based upon a fairly standard classical robotics stack that decomposes perception, state estimation, behavior, and control into separate modules. Our main contributions are then to enable the use of RL as a central part of the behavior module that controls each robot's high-level, strategic decision-making. The architecture enjoys the robustness of a modular approach, uses separately trained RL policies to achieve flexibility and versatility, and allows for improvement at deployment time.

To effectively train behaviors, we adopt a sim2real approach and use simulators of different fidelities. A lower fidelity simulator enables extensive full field training, whereas a higher fidelity simulator enables the robot to learn more precise ball control in critical situations. Furthermore, instead of training a monolithic policy for all game scenarios, we decompose the overall behavior into four learned sub-behaviors with different action and observation spaces. During games, we heuristically select between behaviors to integrate human knowledge into our strategy and enable rapid adjustment.

In this paper, we fully describe the key components of our architecture and training framework and then empirically study the importance of key design decisions. Specifically, the main contributions of our work are:

- We detail our novel RL-based robot behavior architecture and training framework that led to winning the RoboCup SPL Challenge Shield Division.
- We identify and describe key design choices in the architecture: multifidelity RL training, behavior decomposition into sub-behaviors, heuristic selection of sub-behaviors during deployment, and usage of different action and observation spaces across sub-behaviors.
- We analyze our key design choices in a series of ablation experiments. Our experiments validate the effectiveness of key aspects of our architecture, complementing our victory in the 2024 SPL Challenge Shield Division.

[1]University of Wisconsin–Madison. [2]The University of Texas at Austin. [3]Sony AI. *Indicates equal contribution. †Indicates equal advising. All other authors listed in alphabetical order.
Correspondence to: labiosa@wisc.edu

II. BACKGROUND

In this section, we provide background on reinforcement learning and describe related work on enabling RL in robotics and other use-cases of RL to target similar domains.

A. Reinforcement Learning

Reinforcement learning algorithms enable an agent to learn optimal actions in sequential decision-making environments. We formalize this environment as a Partially Observable Markov Decision Process (POMDP) $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \mathcal{O}, \Omega, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transition function, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, $\mathcal{O}$ is the observation space, $\Omega : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{O})$ is the observation model, and γ is the discount factor. In a POMDP, the agent takes in the history of observations or a belief state and outputs an action. The objective is to maximize the expected cumulative reward, defined as $J(\pi) \coloneqq \mathbf{E}[\sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t, a_t)]$. It should be noted that even though we are interested in the multi-robot SPL domain, from the point of view of any single robot, the actions of other robots are represented as just part of the state transition function.

B. Related Work

Reinforcement Learning (RL) has significantly advanced robot learning, particularly via sim2real transfer for tasks like bipedal locomotion [2], [6]–[13]. However, these successes often focus on lower-level control and haven't addressed high-level decision-making in complex, dynamic, multi-agent domains like the RoboCup Standard Platform League (SPL). While some research explores hierarchical RL [14]–[16] or high-level policies in abstract simulation [17], [18], these typically involve simpler dynamics or platforms than the SPL. Our work differs by integrating RL for high-level strategy within a classical robotics stack, using manually decomposed sub-behaviors rather than a single monolithic or hierarchical policy, enhancing fine-grained control and transferability.

We utilize two simulation fidelities. While multi-fidelity simulation has been used with RL to improve sample efficiency or performance [19]–[23] and sometimes for sim2real transfer [24], [25], these works often don't target physical robots or train a single policy across increasing realism. In contrast, we train multiple, distinct policies specialized for different tasks across different, complementary simulation fidelities.

Within robot soccer [5], [26]–[28], many RL applications are limited to simulation [29]–[34] or use non-bipedal robots [35]–[40]. A notable exception learns agile bipedal soccer skills [41], but relies on external motion capture, unlike the fully autonomous SPL setting. While heuristics have been used for teamwork [42], our approach integrates learned RL policies with heuristic selection for strategic decision-making on physical robots in the challenging SPL environment.

III. DOMAIN CHALLENGES AND RL INTEGRATION IN ROBOCUP SPL

The RoboCup Standard Platform League (SPL) presents significant robotics challenges relevant to developing our RL approach. The SPL requires teams of fully autonomous humanoid robots (5v5 in the Challenge Shield Division) to play soccer using onboard perception and control, with limited, unreliable communication [43]. Robots must act in real-time under partial observability (uncertainty from vision/proprioception), coordinating amidst dynamic changes in ball/robot positions and unpredictable opponents. This demands rapid decision-making based on incomplete information. Like many teams, we leverage the B-Human codebase [43], a high-performing open-source system, as our foundation.

While RL is a promising approach, applying it to a testbed such as the SPL faces hurdles that, to our knowledge, have prevented its successful use previously. End-to-end learning from pixels to torques, while demonstrated in simpler settings [44], appears computationally impracticable for full 20-minute, multi-robot SPL games due to the scale and complexity. Integrating RL for high-level control by utilizing existing low-level skills also presents difficulties: the sim2real gap is significant, the open source high-fidelity simulator SimRobot is computationally slow and don't easily parallelize for RL training, and the domain's complexity makes training a single, monolithic RL policy to cover all situations intractable. Our architecture (Section IV-B) is designed to address these specific integration challenges.

IV. REINFORCEMENT LEARNING WITHIN A COMPLETE ROBOT SYSTEM

We describe our system (Fig. 1) and key design decisions enabling successful RL integration in the SPL.

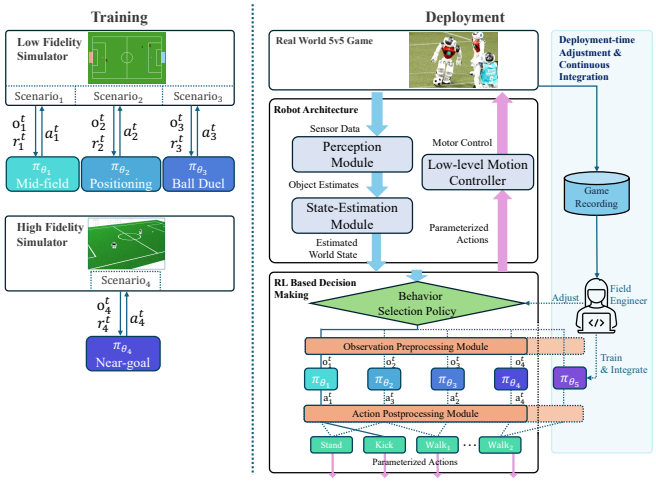


Fig. 1: Architecture of our training and deployment system. Left: Multi-fidelity training setup using AbstractSim (low-fi) and SimRobot (high-fi). Right: Deployment architecture built on the B-Human classical stack (Perception, State-Estimation, Low-level Control), integrating our RL-based decision making. The RL module uses heuristic policy selection based on the estimated world state to choose among specialized sub-behaviors executed by the controller.

Policy	Action Space	Action Space Description	Observation Space
MID-FIELD	$[\Delta\Theta]$	Adjusts desired kick angle (global frame); clipped.	[Ball, Can kick?, Goal ctr, Goalposts, Sides, Last 3 ball pos]
BALL DUEL	$[\Delta X, \Delta Y, \Delta\Theta]$	Egocentric velocity control (x, y, theta).	[Ball, Can kick?, Closest teammate, Goalposts, Sides, Last 3 ball pos]
NEAR-GOAL	$[\Delta X, \Delta Y, \Delta\Theta]$	Same as BALL DUEL.	[Ball, Opp Goalposts, Last 3 ball pos]
POSITIONING	$[\Delta X, \Delta Y, \Delta\Theta, Stand]$	Similar to BALL DUEL, plus Stand action.	[Ball, Strat pos, Defenders, Goalposts, Sides, Last 3 ball pos]

TABLE I: Action and observation space details for each sub-policy.

A. Robot Architecture and Simulation

We built upon the classical B-Human architecture [43], leveraging its perception, localization, and motion primitives. This avoids end-to-end learning challenges, allowing our RL policies to operate at a high level, processing estimated game state (e.g., ball/robot positions) and outputting parameterized actions for low-level skills (Table I).

To manage the sim2real gap and training costs, we employed a multi-fidelity simulation strategy. We developed AbstractSim (Fig. 1 top left), a fast, low-fidelity 2D simulator abstracting robot kinematics, enabling efficient training of broad behaviors across the field. For critical scenarios requiring precision (e.g., near the goal), we used the high-fidelity, physics-based SimRobot simulator (Fig. 1 middle left), despite its slower speed.

B. RL Behavior Decomposition and Selection

Instead of a monolithic policy, we decomposed behavior into four sub-policies trained with PPO [45], [46], each specialized using different simulators and action/observation spaces (Table I). Each policy is instantiated as a neural network, and the output is used as input to a low-level skill.

The BALL DUEL policy, trained in a 2 vs. 0 AbstractSim environment, develops ball control skills through velocity-based maneuvering. Despite the absence of opponents in training, its proficiency in ball handling makes it effective in real-world contested situations.

The MID-FIELD policy addresses the BALL DUEL policy's limitations in walking and kicking for less contested scenarios. Developed in a 1 vs. 0 AbstractSim environment, it outputs a kick angle parameterizing B-Human's walk-and-kick skill, which incorporates obstacle avoidance. This policy sacrifices precise velocity control for enhanced movement speed and kicking accuracy.

The NEAR-GOAL policy is designed for critical situations near the goal requiring decisive, precise movement. Trained using a 1 vs. 0 scenario in the high-fidelity SimRobot simulator, it learned subtle strategies like making small lateral movements to effectively bump the ball towards the goal, proving more efficient than actively kicking.

Finally, the POSITIONING policy guides the robot's movement when a teammate is closer to the ball. Trained in AbstractSim, it considers the ball's position and a manually defined strategy position, aiming to keep the ball in view while avoiding opponents.

A heuristic policy selector dynamically switches between these behaviors based on game state: POSITIONING is active if a teammate is closer to the ball; NEAR-GOAL activates within the opponent's goal box; BALL DUEL engages if an opponent is very close (≤ 0.5m) to the ball; otherwise, the default MID-FIELD policy is used. This modular, heuristic approach provided flexibility, allowing us to tune activation regions (e.g., for NEAR-GOAL) and integrate improvements during the competition, contributing to our performance.

V. EMPIRICAL ANALYSIS

In this section, we study the key decisions that led to our first-place finish in the RoboCup competition. We focus on three elements that we hypothesized contributed to our success: heuristic policy selection, training policies in different simulation fidelities, and utilizing distinct action spaces for the BALL DUEL and MID-FIELD policies. We conduct experiments on physical robots and in high-fidelity simulation (SimRobot).

A. Heuristic Policy Conditioning

Experiment	Physical Successes
Full Suite	$\mathbf{6/10 \pm 3}$
No MID-FIELD	$0/10 \pm 0$
No NEAR-GOAL	$4/10 \pm 3$
No BALL DUEL	$3/10 \pm 3$

Fig. 2: Evaluation of policy decomposition on success rate against a defender robot. Success is a goal, failure is an out of bounds or timeout of a minute. Higher is better. Confidence intervals are 95% bootstrapped.

The first experiment evaluates our policy decomposition and heuristic selection. We tested performance on physical robots against a weakened defender and goalie[1] with disabled kicking abilities in a 1 vs. 2 scoring evaluation. The results (Figure 2) show that the full suite of policies outperforms systems where one policy is removed, indicating that each policy plays a crucial role.

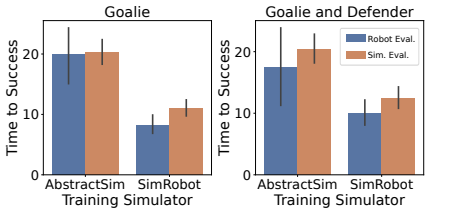
B. Simulation Fidelity

The second experiment examines the impact of simulation fidelity, comparing the NEAR-GOAL policy trained in high-fidelity SimRobot versus low-fidelity AbstractSim. We trained policies to convergence and tested them in two scenarios: goalie only, and defender and goalie together, starting the attacker with the ball near the goal box. The results (Figure 3) demonstrate that on physical robots, the SimRobot-trained policy achieves significantly

[1]The goalie code in our system is manually defined, as the behavior for this role is relatively simple to implement.

Experiment	Training Simulation	Physical Success	Simulation Success
Goalie	AbstractSim	$7/10 \pm 3$	$\mathbf{77/100 \pm 8}$
	SimRobot	$\mathbf{9/10 \pm 1.5}$	$62/100 \pm 9$
Goalie and Defender	AbstractSim	$4/10 \pm 3$	$\mathbf{62/100 \pm 9}$
	SimRobot	$\mathbf{9/10 \pm 1.5}$	$60/100 \pm 10$

(a) Simulation type success results. Higher is better.



(b) Simulation type time to success results. Lower is better.

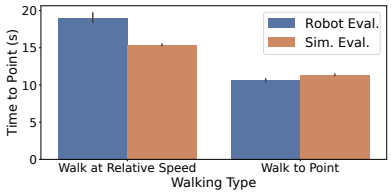
Fig. 3: Training simulation fidelity comparison for the NEAR-GOAL policy (AbstractSim vs. SimRobot training). Tested against goalie only or goalie+defender scenarios. Confidence intervals are 95% bootstrap.

higher success rates and shorter scoring times. Interestingly, AbstractSim-trained policies performed better in simulation, indicating the AbstractSim policy failed to generalize effectively despite apparent simulation success.

C. Action Spaces

Experiment	Physical Success	Simulation Success
Walk at Relative Speed	$\mathbf{7/10 \pm 3}$	$\mathbf{41/100 \pm 15}$
Walk to Point	$1/10 \pm 1.5$	$11/100 \pm 6$

(a) Evaluation of action spaces. Success moving the ball past the opponent with control. Failure is a timeout at a minute or losing control of the ball. Higher is better.



(b) Walking Type Experimental Results. Time to reach a point on the opposite side of the field as the robot. Lower is better.

Fig. 4: Results from action space experiments. In Figure 4a we show the success of dribbling around an opponent. In Figure 4b we show the time to walk to a point 4m away from the robot. Confidence intervals are 95% bootstrap.

The third experiment examines the trade-offs between the action spaces used in our BALL DUEL (walk-at-relative-speed) and MID-FIELD (walk-to-point) policies via two tests (Figure 4).

The first test assesses moving the ball around an opposing robot (defender code with kicking disabled). The walk-at-relative-speed action space achieved significantly higher success than walk-to-point, demonstrating superior precise ball manipulation needed for dribbling.

The second test measures the time to walk 4m away. Qualitatively, the walk-to-point action space produced smoother and faster movement due to its stable desired location, whereas the walk-at-relative-speed action space, adjusting velocity frequently, resulted in slower traversal.

VI. DISCUSSION AND LIMITATIONS

Our SPL case study offers lessons for similar domains. Decomposing complex RL tasks into learnable sub-behaviors allows faster training and facilitates adjustments to the overall behavior post-training. Bootstrapping off of existing classical robotics stacks can also make RL more feasible with limited resources. Our approach also shows that matching simulator fidelity to the target task is crucial. For tasks requiring both global coverage and local precision, using multiple fidelities of simulation can enhance overall performance.

These lessons could generalize to other complex domains. For instance, in multi-robot disaster response, teams could use simplified simulators to develop general exploration policies and high-fidelity simulations to refine task-specific sub-behaviors (e.g., debris removal), integrating these using heuristic selection within classical frameworks to reduce computational burden compared to end-to-end training.

Our current approach faces limitations addressable by future work. Developing multi-agent training methods beyond hand-coded scenarios could improve complex team behaviors. Our heuristic selection ignores teammate policy choices, leading to potentially rapid role switching; communication or bidding systems could help. Other directions include learning the sub-behavior selection, better balancing simulator fidelities, and exploring human-in-the-loop methods.

VII. CONCLUSION

Robot soccer and the annual RoboCup competition is a research challenge task designed to spur innovation in building complete robot architectures that can operate in dynamic, partially observable, and adversarial domains. In this paper, we have described an RL approach for developing high-level behaviors for the NAO robot that won the Challenge Shield division of the 2024 RoboCup Standard Platform League competition. This work provides insights and lessons for using model-free RL as a primary driver of decision-making in dynamic, multi-agent and partially observable robot tasks where end-to-end RL may be intractable yet domain complexity suggests that manual programming of behaviors is likely suboptimal. In addition to describing our system, we conducted empirical analysis of three critical components: heuristic-based policy selection, varying simulation fidelity and different action spaces. The results of this analysis provide further lessons for the application of RL in domains with similar challenges. This work demonstrates the promise of RL for developing robot behaviors in complex, dynamic, partially observable, and multi-agent domains.

REFERENCES

[1] I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas *et al.*, "Solving rubik's cube with a robot hand," *arXiv preprint arXiv:1910.07113*, 2019.

[2] Z. Li, X. Cheng, X. B. Peng, P. Abbeel, S. Levine, G. Berseth, and K. Sreenath, "Reinforcement learning for robust parameterized locomotion control of bipedal robots," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 2811–2817.

[3] D. B. D'Ambrosio, S. Abeyruwan, L. Graesser, A. Iscen, H. B. Amor, A. Bewley, B. J. Reed, K. Reymann, L. Takayama, Y. Tassa, K. Choromanski, E. Coumans, D. Jain, N. Jaitly, N. Jaques, S. Kataoka, Y. Kuang, N. Lazic, R. Mahjourian, S. Moore, K. Oslund, A. Shankar, V. Sindhwani, V. Vanhoucke, G. Vesom, P. Xu, and P. R. Sanketi, "Achieving human level competitive robot table tennis," 2024. [Online]. Available: https://arxiv.org/abs/2408.03906

[4] D. Nardi, I. Noda, F. Ribeiro, P. Stone, O. von Stryk, and M. Veloso, "Robocup soccer leagues," *AI Magazine*, vol. 35, no. 3, pp. 77–85, 2014.

[5] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, and E. Osawa, "Robocup: The robot world cup initiative," in *Proceedings of the first international conference on Autonomous agents*, 1997, pp. 340–347.

[6] C. Tang, B. Abbatematteo, J. Hu, R. Chandra, R. Martín-Martín, and P. Stone, "Deep reinforcement learning for robotics: A survey of real-world successes," *arXiv preprint arXiv:2408.03539*, 2024.

[7] J. Siekmann, S. Valluri, J. Dao, L. Bermillo, H. Duan, A. Fern, and J. Hurst, "Learning memory-based control for human-scale bipedal locomotion," *arXiv preprint arXiv:2006.02402*, 2020.

[8] G. A. Castillo, B. Weng, W. Zhang, and A. Hereid, "Reinforcement learning-based cascade motion policy design for robust 3d bipedal locomotion," *IEEE Access*, vol. 10, pp. 20 135–20 148, 2022.

[9] H. Duan, B. Pandit, M. S. Gadde, B. Van Marum, J. Dao, C. Kim, and A. Fern, "Learning vision-based bipedal locomotion for challenging terrain," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 56–62.

[10] R. Beranek, M. Karimi, and M. Ahmadi, "A behavior-based reinforcement learning approach to control walking bipedal robots under unknown disturbances," *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 5, pp. 2710–2720, 2021.

[11] C. Kouppas, M. Saada, Q. Meng, M. King, and D. Majoe, "Hybrid autonomous controller for bipedal robot balance with deep reinforcement learning and pattern generators," *Robotics and Autonomous Systems*, vol. 146, p. 103891, 2021.

[12] D. Qin, G. Zhang, Z. Zhu, T. Chen, W. Zhu, X. Rong, A. Xie, and Y. Li, "A heuristics-based reinforcement learning method to control bipedal robots," *Int. J. Humanoid Robot*, 2024.

[13] T. Li, H. Geyer, C. G. Atkeson, and A. Rai, "Using deep reinforcement learning to learn high-level policies on the atrias biped," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 263–269.

[14] O. Nachum, M. Ahn, H. Ponte, S. Gu, and V. Kumar, "Multi-agent manipulation via locomotion using hierarchical sim2real," *arXiv preprint arXiv:1908.05224*, 2019.

[15] T. Li, N. Lambert, R. Calandra, F. Meier, and A. Rai, "Learning generalizable locomotion skills with hierarchical reinforcement learning," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 413–419.

[16] T. Li, R. Calandra, D. Pathak, Y. Tian, F. Meier, and A. Rai, "Planning in learned latent action spaces for generalizable legged locomotion," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2682–2689, 2021.

[17] J. Truong, M. Rudolph, N. H. Yokoyama, S. Chernova, D. Batra, and A. Rai, "Rethinking sim2real: Lower fidelity simulation leads to higher sim2real transfer in navigation," in *Conference on Robot Learning*. PMLR, 2023, pp. 859–870.

[18] Y. Zhang, Y. Hu, Y. Song, D. Zou, and W. Lin, "Back to newton's laws: Learning vision-based agile flight via differentiable physics," *arXiv preprint arXiv:2407.10648*, 2024.

[19] S. Bhola, S. Pawar, P. Balaprakash, and R. Maulik, "Multi-fidelity reinforcement learning framework for shape optimization," *Journal of Computational Physics*, vol. 482, p. 112018, 2023.

[20] M. Cutler, T. J. Walsh, and J. P. How, "Reinforcement learning with multi-fidelity simulators," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 3888–3895.

[21] ——, "Real-world reinforcement learning via multifidelity simulators," *IEEE Transactions on Robotics*, vol. 31, no. 3, pp. 655–671, 2015.

[22] S. Khairy and P. Balaprakash, "Multi-fidelity reinforcement learning with control variates," *Neurocomputing*, p. 127963, 2024.

[23] J. J. Beard and A. Baheri, "Black-box safety validation of autonomous systems: A multi-fidelity reinforcement learning approach," *arXiv preprint arXiv:2203.03451*, 2022.

[24] V. Suryan, N. Gondhalekar, and P. Tokekar, "Multifidelity reinforcement learning with gaussian processes: model-based and model-free algorithms," *IEEE Robotics & Automation Magazine*, vol. 27, no. 2, pp. 117–128, 2020.

[25] G. Ryou, G. Wang, and S. Karaman, "Multi-fidelity reinforcement learning for time-optimal quadrotor re-planning," *arXiv preprint arXiv:2403.08152*, 2024.

[26] C. Hong, I. Jeong, L. F. Vecchietti, D. Har, and J.-H. Kim, "Ai world cup: robot-soccer-based competitions," *IEEE Transactions on Games*, vol. 13, no. 4, pp. 330–341, 2021.

[27] A. Smit, H. A. Engelbrecht, W. Brink, and A. Pretorius, "Scaling multi-agent reinforcement learning to full 11 versus 11 simulated robotic football," *Autonomous Agents and Multi-Agent Systems*, vol. 37, no. 1, p. 20, 2023.

[28] 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.

[29] P. Stone, R. S. Sutton, and G. Kuhlmann, "Reinforcement learning for RoboCup-soccer keepaway," *Adaptive Behavior*, vol. 13, no. 3, pp. 165–188, 2005.

[30] M. Abreu, L. P. Reis, and N. Lau, "Designing a skilled soccer team for robocup: Exploring skill-set-primitives through reinforcement learning," 2023. [Online]. Available: https://arxiv.org/abs/2312.14360

[31] S. Huang, W. Chen, L. Zhang, S. Xu, Z. Li, F. Zhu, D. Ye, T. Chen, and J. Zhu, "Tikick: Towards playing multi-agent football full games from single-agent demonstrations," 2021. [Online]. Available: https://arxiv.org/abs/2110.04507

[32] F. Lin, S. Huang, T. Pearce, W. Chen, and W.-W. Tu, "Tizero: Mastering multi-agent football with curriculum learning and self-play," 2023. [Online]. Available: https://arxiv.org/abs/2302.07515

[33] S. Liu, G. Lever, Z. Wang, J. Merel, S. A. Eslami, D. Hennes, W. M. Czarnecki, Y. Tassa, S. Omidshafiei, A. Abdolmaleki *et al.*, "From motor control to team play in simulated humanoid football," *Science Robotics*, vol. 7, no. 69, p. eabo0235, 2022.

[34] S. Liu, G. Lever, J. Merel, S. Tunyasuvunakool, N. Heess, and T. Graepel, "Emergent coordination through competition," *arXiv preprint arXiv:1902.07151*, 2019.

[35] I. J. da Silva, D. H. Perico, T. P. D. Homem, and R. A. da Costa Bianchi, "Deep reinforcement learning for a humanoid robot soccer player," *Journal of Intelligent & Robotic Systems*, vol. 102, no. 3, p. 69, 2021.

[36] A. Merke and M. Riedmiller, "Karlsruhe brainstormers-a reinforcement learning approach to robotic soccer," in *RoboCup 2001: Robot Soccer World Cup V 5*. Springer, 2002, pp. 435–440.

[37] M. Riedmiller, T. Gabel, R. Hafner, and S. Lange, "Reinforcement learning for robot soccer," *Autonomous Robots*, vol. 27, pp. 55–73, 2009.

[38] X. Huang, Z. Li, Y. Xiang, Y. Ni, Y. Chi, Y. Li, L. Yang, X. B. Peng, and K. Sreenath, "Creating a dynamic quadrupedal robotic goalkeeper with reinforcement learning," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 2715–2722.

[39] Y. Ji, G. B. Margolis, and P. Agrawal, "Dribblebot: Dynamic legged manipulation in the wild," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5155–5162.

[40] Y. Ji, Z. Li, Y. Sun, X. B. Peng, S. Levine, G. Berseth, and K. Sreenath, "Hierarchical reinforcement learning for precise soccer shooting skills using a quadrupedal robot," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 1479–1486.

[41] T. Haarnoja, B. Moran, G. Lever, S. H. Huang, D. Tirumala, J. Humplik, M. Wulfmeier, S. Tunyasuvunakool, N. Y. Siegel, R. Hafner *et al.*, "Learning agile soccer skills for a bipedal robot with deep reinforcement learning," *Science Robotics*, vol. 9, no. 89, p. eadi8022, 2024.

[42] R. Ros, J. L. Arcos, R. L. De Mantaras, and M. Veloso, "A case-based approach for coordinated action selection in robot soccer," *Artificial intelligence*, vol. 173, no. 9-10, pp. 1014–1039, 2009.

[43] T. Röfer, T. Laue, F. Böse, A. Hasselbring, J. Lienhoop, L. M. Monnerjahn, P. Reichenberg, and S. Schreiber, "B-Human code release documentation 2023," 2023, only available online: https://docs.b-human.de/coderelease2023/.

[44] D. Tirumala, M. Wulfmeier, B. Moran, S. Huang, J. Humplik, G. Lever, T. Haarnoja, L. Hasenclever, A. Byravan, N. Batchelor *et al.*, "Learning robot soccer from egocentric vision with deep reinforcement learning," *arXiv preprint arXiv:2405.02425*, 2024.

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

[46] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021. [Online]. Available: http://jmlr.org/papers/v22/20-1364.html