Digital Twin-Based Transfer Learning for Collaborative Robot Systems: A Proof of Concept

Supat Roongpraiwan[§], Zongdian Li[§], Mohsen Pourghasemian[‡], Haris Gacanin[‡], and Kei Sakaguchi[§]
[§]Department of Electrical and Electronic Engineering, Tokyo Institute of Technology, Tokyo, Japan.
[‡]Chair for Distributed Signal Processing, RWTH Aachen University, Aachen, Germany.
Emails: *{supat, lizd, sakaguchi}@mobile.ee.titech.ac.jp, †{mohsen, harisg}@dsp.rwth-aachen.de

***Abstract*—Nowadays, the increasing trend toward digitalization has driven the extensive adoption of collaborative robotic automation across industries, yet a significant limitation is the robots' adaptability to unexpected and dynamic environments. This research introduces a Digital Twin (DT)-based Transfer Learning (TL) approach that combines DTs and Machine Learning (ML) to enhance adaptability in collaborative robot systems. The proposed system uses DT cyberspace for pre-training ML algorithms and leverages TL to apply this knowledge to real-world applications. This innovative approach efficiently trains state-of-the-art ML models, delivering exceptional performance while reducing the required time and data resources. The proof-of-concept experiments, employing the proposed DT-based TL to control soccer robots, demonstrate a remarkable 96% reduction in training time while maintaining a high level of adaptability, achieving a 70% goal accuracy rate in dynamic scenarios.**

***Index Terms*—digital twin, collaborative robot system, machine learning, transfer learning, proof of concept**

I. Introduction

In recent years, the trend toward digitalization has increased the integration of collaborative robotic automation across various industries. Robotic automation adoption has proven to bring significant benefits, such as increased productivity, cost reduction, and improved product and service quality [1]. As a result, collaborative robots become increasingly popular, enabling industries to optimize processes, achieve high levels of productivity, and provide greater profits to meet the evolving demands of our digitized world.

Despite the numerous benefits of the current collaborative robot systems, there are still several notable limitations. One of the most significant limitations is the robots' limited ability to adapt to unexpected and dynamic environments [2]. Although collaborative robot systems can be programmed to perform specific tasks with precision and consistency as they help each other to achieve a wide range of complex objectives, they often struggle to handle unanticipated situations that are out of the programming scope. This results in robots instigating errors or unexpected outcomes in their tasks, which further leads to operational disruptions, productivity losses, and potential accidents. This lack of adaptability presents a considerable challenge when applying collaborative robot systems in real-world scenarios. In many instances, human supervision and intervention are needed to supplement the robots' capabilities and ensure the smooth operation of the system. This reliance on human involvement not only introduces additional costs and complexity but also discourages industries from realizing the full potential of collaborative robot systems in uncertain scenarios [3].

To overcome this adaptability challenge and enable robots to handle dynamic scenarios, this research aims to identify effective approaches for enhancing the robots' adaptability. In pursuit of this objective, we propose a DT-based TL method. By harnessing the power of DTs, we can leverage both the physical environment and its cyber reflection for efficient ML training. Our proposed DT-based TL framework can effectively train complex ML models, achieving outstanding performance while reducing the required training time and data. To validate the effectiveness of our proposed learning framework, we conduct a proof of concept using the soccer robot scenario, which achieves up to 96% reduction of training time while maintaining high-level adaptability with a remarkable performance of 70% goal accuracy in the physical world.

II. ML-based Collaborative Robots

ML-based collaborative robots have emerged as a notable advancement, significantly elevating the adaptability of collaborative robots. Traditionally, robotic tasks heavily relied on manual programming and inflexible algorithms. However, the incorporation of ML, such as Imitation Learning (IL) and Reinforcement Learning (RL), disrupts conventional robotic approaches. The IL enables robots to imitate complex expert behavior, facilitating rapid skill acquisition for proficient task execution [4]. Meanwhile, the RL allows robots to continuously optimize actions in response to changing environments by maximizing cumulative rewards [5]. Integration of these ML techniques with collaborative robots leads to significant benefits, as demonstrated by [6]–[8], which not only enhances the robots' robustness and adaptability but also elevates the overall performance of collaborative robot systems.

However, the training of adaptive ML algorithms for collaborative robots requires a substantial volume of training data, which can be impractical to obtain in the real world. To address these challenges, two primary approaches have been developed as noteworthy solutions: simulation-based ML and TL.

Simulation-based ML approaches leverage simulated environments to generate the training data. Simulations allow for controlled and repeatable training scenarios, making it feasible to accumulate large datasets needed for ML model training. However, a notable drawback of this approach is the inevitable performance drop when transitioning from virtual simulations to real-world environments, as emphasized by [7] and [9].

On the other hand, TL is emerging as a promising technique in the field of robotics, as evidenced by [10]. This approach involves the transfer of knowledge acquired by one robot to another, resulting in improved performance for the newly trained robot. TL also offers a significant reduction in the time required for training [11]. By leveraging pre-existing knowledge and models, collaborative robots can adapt more rapidly to new tasks and environments, effectively reducing the learning curve. However, TL typically relies on pre-trained models from a source task, which may not always be readily available or applicable to all scenarios, and it requires a certain degree of similarity between the source and target tasks for effective knowledge transfer [12].

Recently, DT technology has gathered widespread attention, particularly in the industrial and manufacturing sectors. By establishing a connection between the physical and digital realms, the DT framework enables the seamless exchange of cyber-physical information, empowering informed decision-making and operational optimization [13], [14]. Additionally, DTs provide a platform for innovative experimentation, allowing organizations to explore ideas and assess improvements without affecting their physical systems. With access to cyber-physical data and experimental capabilities, DTs hold potential for applications in ML [15], [16]. In contrast to conventional simulation-based ML, DT offers not only an accurately reflected simulation platform but also the integration of real-world data to model training. Moreover, the DT can easily complement TL by transferring the accumulated knowledge from simulations to real-world experiences. Despite DT's great potential, it is rare to see practical implementations or demonstrations of DT-based learning systems in state-of-the-art studies. To provide our contributions, we develop a DT-based TL framework and conduct comprehensive proofs of concept to showcase its use cases and performance. Section III reveals more details about our proposed framework.

III. Digital Twin-based Transfer Learning

A. Concept

In this section, we present a DT-based TL approach designed for the optimization of collaborative robot systems. Leveraging the concept of a DT, we harness the power of virtual environments within cyberspace to pre-train ML algorithms, subsequently employing TL to transpose this acquired knowledge to real-world applications. This innovative approach demonstrates the capability to efficiently train state-of-the-art ML models, yielding exceptional performance while demanding reduced temporal and data resources within real-world operational settings.

B. General System Architecture

The general system architecture of the proposed DT-based TL method is illustrated in Fig. 1. Initially, ML is trained within a DT environment using simulated data. Upon completing this training, the model is stored within a central control center, serving as a centralized command node for overseeing and controlling all robots. During the deployment phase, the model efficiently processes sensor inputs from each robot and sends optimal control feedback tailored to the current conditions and system's requirements.

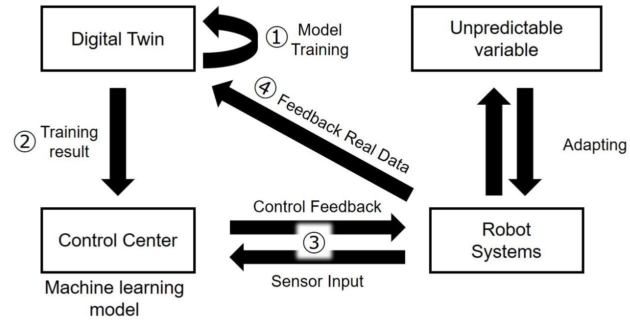


Fig. 1: The architecture of the DT-based TL method.

However, a model solely trained within the DT cyberspace may not seamlessly transition to real-world applications due to the inherent gap between simulated and actual environments. To overcome this, following the initial deployment, we collect real-world data that accurately reflects the true operating conditions and complexities of the physical space to the DT. This real-world data is then used for fine-tuning the model through TL, resulting in an enhanced version better suited for real-world situations. Once the model has undergone refinement, it is redeployed to control the robot system, enabling more effective adaption and high performance in real-world settings.

C. Transfer Learning Flow

Our DT-based TL approach integrates TL by incorporating elements from both IL and RL. The training process begins with IL in cyberspace by imitating behaviors that are observed from the human expert demonstration. As the model gains proficiency, it transitions into RL phase, where it explores actions and strategies independently, learns from the outcomes, and changes its behavior based on the feedback rewards it receives. Subsequently, the model is transitioned into the physical space, where it undergoes a further refinement process by leveraging real-world environmental information from the sensor data, allowing the model to adapt and enhance its performance by actively interacting with the physical environment. This multifaceted knowledge integration ensures that the model combines insights and expertise from human experts, the virtual environment, and the physical environment, resulting in a robust and adaptable collaborative robot system that performs exceptionally well in real-world applications.

IV. Proof of Concept

A. Scenario

To validate the effectiveness of the proposed system architecture, this research conducts a proof of concept using the soccer robot scenario.

The proof-of-concept scenario is designed where multiple robots collaborate to score a goal, as shown in Fig. 2. The implementation involves using a DT-based TL method to train an ML model to control the soccer robots effectively. In this scenario, the soccer ball acts as a semi-uncontrollable

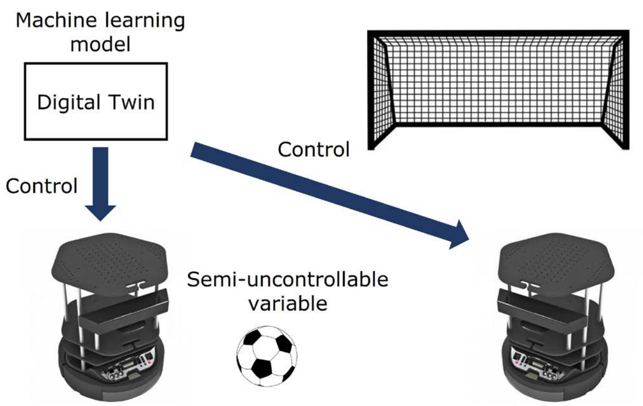


Fig. 2: Proof of concept using soccer robots.

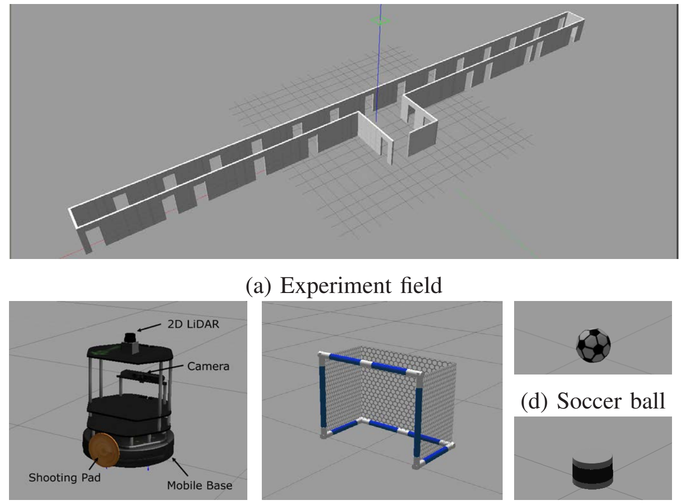
variable, where the robots can detect its position but are unaware of its precise movements after interacting with it. This approach allows the robots to adapt and respond to unpredictable movements of the soccer ball, ultimately leading to successful goal scoring.

The proof of concept using soccer robots consists of three main parts: 1) Single robot shoots from a stationary ball, which exhibits the robot system's ability to control and accurately shoot the ball; 2) Single robot shoots from a moving ball, which demonstrates the robot system's quick reactions and adaptability. 3) Multiple robots pass and shoot the ball, which showcases the robot system's ability to function as a team, collaborating together to complete the task. These proof-of-concept demonstrations highlight the efficiency of the proposed DT-based TL for collaborative robot systems.

B. Implementation

a) Implementation in cyberspace: In the DT implementation for soccer robots, the Robot Operating System (ROS) [17] is employed as the middleware for controlling the robot system, while Gazebo [18] serves as the robot simulator to construct a cyberspace of the DT. As depicted in Fig. 3, the simulation environment was composed of well-constructed models representing the experimental field, soccer robot, soccer goal, soccer ball, and 3D LiDAR sensor.

The soccer robot model in Fig. 3b represents a specific type of robot called "Kobuki robot" used in this research. The Kobuki robot is a popular mobile base platform, often used in research and educational settings [19]. Equipped with various sensors and actuators, the Kobuki robot exhibits the necessary mobility and manipulation capabilities for soccer-related tasks. To enhance its perception abilities, the Kobuki robot is equipped with a 2D LiDAR sensor and a camera. The 2D LiDAR sensor allows the robot to obtain a 360-degree view of its surroundings, while the camera provides visual information for object recognition and tracking. To enable the Kobuki robot to effectively interact with the soccer ball, modifications were made to its front end. A specially designed shooting pad was added to ensure a flat contact surface with the soccer ball during shooting actions. This modification allows the robot to apply force on the ball accurately and



(a) Experiment field

(b) Soccer robot (c) Soccer goal (d) Soccer ball (e) 3D LiDAR

Fig. 3: DT models of the game field and soccer robots.

efficiently, simulating kicking motions performed by human soccer players.

The 3D LiDAR sensor, as shown in Fig. 3e, serves as an external sensor providing essential surrounding information for the robot. This sensor operates by scanning its environment, capturing spatial details in a 360-degree horizontal and vertical field of view. This capability allows the soccer robots to effectively sense the positions and shapes of various objects, including the soccer ball. The position localization algorithm is implemented by Point Cloud Library (PCL) [20], which leverages Euclidean clustering [21], sphere fitting, and ball tracking techniques. This ensures precise localization of the soccer ball within the environment and is adapted from a specialized ball detection algorithm proposed by [22].

Fig. 4 illustrates the ML model training implementation system. This system is composed of three main components: the dynamic environment, the ROS-based DT bridge, and the training algorithm.

OpenAI Gym [23], a library of environments for RL training, is selected for the ML dynamic environment, providing a standardized interface for the ML algorithm to interact with environments. However, since OpenAI Gym does not natively support ROS, we incorporate OpenAI-ROS [24] serving as a bridge to translate ROS topic information into the OpenAI Gym environment, leveraging ROS's advanced features for robot control and communication while benefiting from OpenAI Gym's ML algorithm training capabilities.

For ML model training algorithm, we employ the Generative Adversarial Imitation Learning (GAIL) algorithm for IL from imitation package [25] and the Proximal Policy Optimization (PPO) algorithm for RL from stable-baselines3 package [26] to train the model. The GAIL [27] merges the Generative Adversarial Networks (GANs) [28] with IL, overcoming the need for high-quality demonstration data by training a generator network to produce realistic imitation trajectories and a discriminator network to differentiate real expert demonstrations from generated ones. The GAIL is effective with limited expert

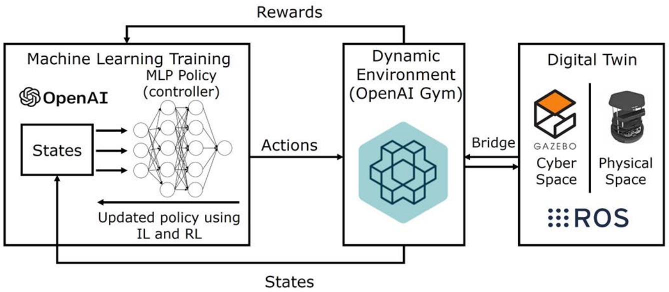


Fig. 4: Implementation of DT-based TL platform.

data, making it a valuable approach for real-world IL. On the other hand, the PPO [29] is a state-of-the-art RL algorithm ideal for training in complex environments. The PPO maintains policy updates with proximal constraints, ensuring stability and reliability, making it a popular RL algorithm choice for real-world decision-making problems.

Firstly, the model policy π is defined by a Multi-Layer Perceptron (MLP) with 2 fully connected hidden layers, each with 64 perceptrons per layer. As illustrated in Fig. 4, this policy takes the environment observation s as inputs and generates output actions a which can be written as $\pi(a|s,\theta)$, where θ represents the model parameters.

The model is initially trained with the GAIL algorithm to imitate the expert behavior. In the GAIL algorithm, the generator network is defined as $G(\theta)$, which is the previously defined ML model policy $\pi(a|s,\theta)$, and the discriminator network is represented as $D(\phi)$, where ϕ are the discriminator network's parameters. The discriminator's role is to differentiate between the agent's generated trajectories and those of an expert. The objective of the GAIL algorithm is to encourage the generator to produce trajectories that are indistinguishable from those of the expert, as determined by the discriminator. This can be achieved by minimizing the discriminator's loss function (Eq. 1) and maximizing the generator's rewards (Eq. 3). The discriminator's loss function can be expressed as follows:

$$L(\phi) = \mathbb{E}_{\pi_G}\left[\log(D(\tau_t))\right] - \mathbb{E}_{\pi_E}\left[\log(1 - D(\tau_t))\right], \quad (1)$$

where $\mathbb{E}$ is the expectation operator, τ_t represents a trajectory which is a set of states and actions (s,a) at time step t, π_G is the generator policy, and π_E is the expert policy. In each training step, $D(\phi)$ updates its parameters based on the gradients of (1), then classifies the trajectory. The classification results from the discriminator, denoted as $D(\tau_t)$ are then employed in the rewards function, which is defined as:

$$R_t = -\log(1 - D(\tau_t)). \quad (2)$$

These rewards are then used to update $G(\theta)$ using the PPO algorithm.

After GAIL training is completed, the knowledge acquired from the IL can be transferred to the RL phase. The objective of RL is to maximize the expected cumulative reward over a trajectory, which can be expressed as:

$$J(\theta) = \mathbb{E}\left[\sum_{t=0}^{T} R_t\right], \quad (3)$$

where θ represents the policy parameters, t is the time step, T is the length of the trajectory, and R_t is the reward at time step t.

This is achieved through the PPO, which optimizes the policy by maximizing the surrogate objective within the proximal clipped region defined as follows:

$$L(\theta) = \mathbb{E}\left[\min\left(r_t(\theta)\hat{A}_t, \text{clip}\left(r_t(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_t\right)\right], \quad (4)$$

where $r_t(\theta)$ represents the probability ratio of taking an action in the new policy over the old policy, which is denoted by $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$, $\hat{A}_t$ is the advantage function that quantifies how much better a particular action is compared to the average action at a given state, and ϵ is a hyperparameter controlling the extent of the policy update. The first term $r_t(\theta)\hat{A}_t$ corresponds to the conventional Trust Region Policy Optimization (TRPO) method, while the second term limits the change to be within the proximal region defined by ϵ.

From (4), PPO updates the model policy in each rollout with the following optimization:

$$\pi_{new}(a|s,\theta) = \arg\max_{\theta}\left(L(\theta)\right). \quad (5)$$

During this RL phase, the model interacts with the environment, aiming to maximize cumulative rewards based on a task-specific reward function defined by the environment itself.

b) Implementation in physical environment: Moving to physical environment, as depicted in Fig. 5, our created cyberspace closely mirrors the actual physical space of the soccer robots proof of concept. The physical space also consists of the soccer robots, a soccer goal, a soccer ball, and a 3D LiDAR sensor.

The soccer robot system relies on the ROS TCP/IP protocols for multi-agent communications, ensuring standardized data sharing to facilitate effective coordination. However, the protocol's reliability can lead to latency when handling large messages. To overcome this challenge, the system adopts an edge computing approach. Specifically, the 3D LiDAR sensor is connected to an edge computer, enabling efficient ball detection. Meanwhile, the soccer robots independently process collision data from their 2D LiDAR sensors, offering immediate feedback for real-time decision-making. By distributing computation and transmitting of only essential input data to the control center, the system reduces the transmission message size, resulting in low-latency communication. These optimizations significantly enhance system responsiveness and performance during the soccer robots' proof of concept.

During the deployment phase, the system initiates by loading a pre-trained ML model from DT's cyberspace into the control center. It then utilizes input observations from the robots and external sensors to generate control feedback for robot management until the conclusion of an episode.

Lastly, in the model fine-tuning phase in the physical space, the system follows a similar workflow, but it also incorporates the use of the PPO RL algorithm to refine the ML model with real data from the physical environment. Fine-tuning ML models with real data facilitates the transfer of knowledge

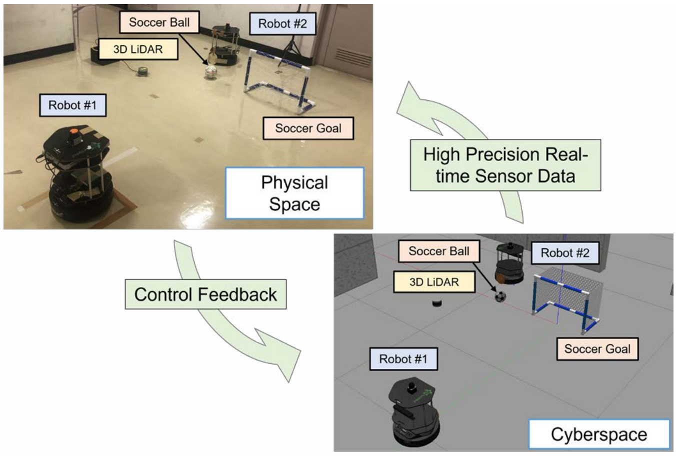


Fig. 5: Configurations of soccer game DT.

from cyberspace to physical space. This adaptation enables models to adjust to the complexities and variations of the real environment, bridging the gap between simulated and real-world scenarios, ultimately ensuring the superior performance in the physical world.

C. Results

In this section, we discuss the results of the soccer robots' proof of concept. Each proof-of-concept scenario is assessed based on the training parameters and their goal accuracy performance at each stage of training, including GAIL and PPO in cyberspace, as well as the deployment and fine-tuning in the physical space.

Table I provides insights into the training results in virtual cyberspace. Notably, as the complexity of the scenarios increases, there is an observable escalation in training requirements, including the amount of expert data and training time, along with a decrease in goal accuracy. Furthermore, the consistent improvement in accuracy through the transition from GAIL to PPO is also noteworthy. For instance, in the "multiple robots passing and shooting" scenario, the goal accuracy increases significantly from 33% with GAIL to 62% with PPO. This emphasizes the added value of TL from IL to RL, enhancing both model accuracy and robustness within a virtual environment.

Table II provides comprehensive results of the physical space deployment and fine-tuning of "single robot shooting from stationary ball" and "single robot shooting from moving ball" scenarios. Notably, the latter scenario presents higher complexity, as evidenced by the initial deployment's lower accuracy, with figures of 70% and 40%. However, significant improvement is achieved through fine-tuning with real-world data, resulting in goal accuracy of 95% and 70% for the respective scenarios.

Based on the results obtained from both the cyberspace (Table I) and physical space (Table II), the scenario of "single robot shooting from a moving ball" has been selected as the representative scenario. We have created Fig. 6 to provide a comprehensive analysis of model accuracy and training time in our experiment.

TABLE I: Model training results in cyberspace.

	Single robot shooting from stationary ball	**Single robot shooting from moving ball**	**Multiple robots passing and shooting**
GAIL	Expert data: 20 EP Total timesteps: 100k Training time: 10 hr Goal accuracy: 86%	Expert data: 20 EP Total timesteps: 200k Training time: 24 hr Goal accuracy: 82%	Expert data: 30 EP Total timesteps: 300k Training time: 36 hr Goal accuracy: 33%
PPO	Total timesteps: 50k Training time: 5 hr Goal accuracy: 98%	Total timesteps: 125k Training time: 15 hr Goal accuracy: 93%	Total timesteps: 200k Training time: 24 hr Goal accuracy: 62%

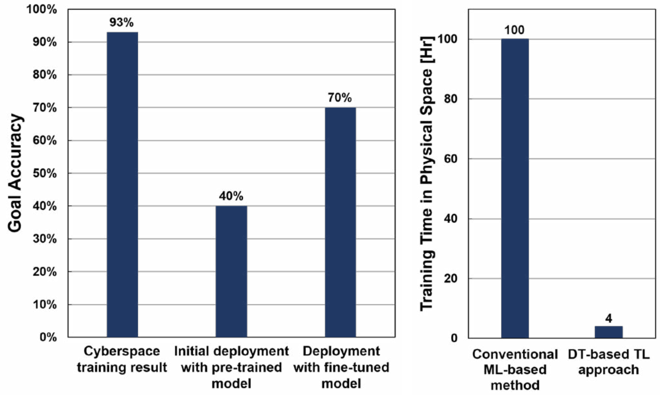
TABLE II: Deployment results in physical space using pre-trained and fine-tuned model.

	Single robot shooting from stationary ball	**Single robot shooting from moving ball**
Pre-trained model	Deployment: 20 EP Goal accuracy: 70%	Deployment: 20 EP Goal accuracy: 40%
Fine-tuned model	Fine-tuning timesteps: 4k Fine-tuning time: 2.5 hr Deployment: 20 EP Goal accuracy: 95%	Fine-tuning timesteps: 8k Fine-tuning time: 4 hr Deployment: 20 EP Goal accuracy: 70%

In Fig. 6a, the model achieved an impressive 93% goal accuracy during cyberspace training. However, upon deploying the same pre-trained model in the physical space, the goal accuracy decreases to 40%. This initial decrease in accuracy highlights the challenges of transitioning from simulation to reality. Nevertheless, fine-tuning with real data significantly improved the model, achieving 70% goal accuracy in the second physical deployment. This emphasizes the importance of using real-world data to bridge the cyber-physical performance gap, demonstrating the significance of our DT-based TL approach for successful deployment.

Furthermore, in Fig. 6b, the conventional RL method requires a model training time of up to 100 hours. In contrast, our approach only requires 4 hours of training time in the physical space. This demonstrates the remarkable effectiveness of our DT-based TL approach in reducing the training time by up to 96%. This significant reduction in training time not only enhances the efficiency of the model development process but also showcases the advantage of leveraging DT training and real data fine-tuning to achieve optimal performance in real-world environments.

In summary, our proof of concept showcases the potential of the DT-based TL approach for enhancing collaborative robot systems in practical applications. While initial accuracy drops when transitioning from cyberspace to the real world, fine-tuning with real data significantly improves performance. Moreover, our DT-based TL approach reduces training time by 96%, highlighting its efficiency and effectiveness.



(a) Goal accuracy (b) Training time

Fig. 6: Performance comparisons in training and deployment phases when migrating from cyberspace to physical space.

V. Conclusion

In conclusion, this research has successfully harnessed the power of DT and ML technologies to develop a DT-based TL approach for collaborative robot systems that demonstrated impressive performance and adaptability while drastically reducing training requirements. The proof of concept exemplified through the soccer robot scenario showcased the DT-based TL effectiveness in both cyber and physical environments, particularly in achieving high goal accuracy during final real-world deployment. The unique approach of knowledge transfer from the cyberspace to the physical space resulted in a remarkable 96% reduction in training time, presenting a clear advantage over the conventional methods. In fact, this DT-based TL approach offers significant potential for a wide range of applications beyond soccer robots, such as industrial automation, healthcare, and logistics, paving the way for efficient, customized collaborative robot systems to tackle real-world challenges.

Acknowledgment

This work was supported by the NICT-JUNO under Grant #22404, and by the U.S. National Science Foundation under JUNO3 Grant CNS-2210254.

References

[1] R. D. Atkinson, "Robotics and the future of production and work," *Information Technology & Innovation Foundation*, Oct. 2019. [Online]. Available: https://www2.itif.org/2019-robotics-future-production.pdf

[2] J. Arents and M. Greitans, "Smart industrial robot control trends, challenges and opportunities within manufacturing," *Applied Sciences*, vol. 12, no. 2, p. 937, 2022.

[3] C. Weidemann, N. Mandischer, F. Kerkom, B. Corves, M. Hüsing, T. Kraus, and C. Garus, "Literature review on recent trends and perspectives of collaborative robotics in work 4.0," *Robotics*, vol. 12, p. 84, 06 2023.

[4] M. Lopes, F. S. Melo, and L. Montesano, "Affordance-based imitation learning in robots," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007, pp. 1015–1021.

[5] C. Szepesvári, *Algorithms for Reinforcement Learning*, Jan. 2010, vol. 4.

[6] N. Liu, T. Lu, Y. Cai, R. Wang, and S. Wang, "Real-world robot reaching skill learning based on deep reinforcement learning," in *2020 Chinese Control And Decision Conference (CCDC)*, 2020, pp. 4780–4784.

[7] Z. Zhang, X. Wang, Q. Zhang, and T. Hu, "Multi-robot cooperative pursuit via potential field-enhanced reinforcement learning," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 8808–8814.

[8] Z. Chen, B. Subagdja, and A.-H. Tan, "End-to-end deep reinforcement learning for multi-agent collaborative exploration," in *2019 IEEE International Conference on Agents (ICA)*, 2019, pp. 99–102.

[9] W. Zhu, X. Guo, D. Owaki, K. Kutsuzawa, and M. Hayashibe, "A survey of sim-to-real transfer techniques applied to reinforcement learning for bioinspired robots," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 7, pp. 3444–3459, 2023.

[10] Z. Deng, H. Guan, R. Huang, H. Liang, L. Zhang, and J. Zhang, "Combining model-based Q-learning with structural knowledge transfer for robot skill learning," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 11, no. 1, pp. 26–35, 2019.

[11] J. Pan, X. Wang, Y. Cheng, and Q. Yu, "Multisource Transfer Double DQN Based on Actor Learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 6, pp. 2227–2238, 2018.

[12] S. Niu, Y. Liu, J. Wang, and H. Song, "A decade survey of transfer learning (2010–2020)," *IEEE Transactions on Artificial Intelligence*, vol. 1, no. 2, pp. 151–166, 2020.

[13] F. Tao, H. Zhang, A. Liu, and A. Y. C. Nee, "Digital twin in industry: State-of-the-art," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2405–2415, 2019.

[14] K. Nonomura, T. Yu, and K. Sakaguchi, "Reshape car-sharing system for super smart society: A digital twin-based method and implementation," in *2023 IEEE International Conference on Consumer Electronics (ICCE)*, 2023, pp. 1–5.

[15] M. Čech, A.-J. Beltman, and K. Ozols, "Digital Twins and AI in Smart Motion Control Applications," in *2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2022, pp. 1–7.

[16] A. Malik, P. Rajaguru, and R. Azzawi, "Smart manufacturing with artificial intelligence and digital twin: A brief review," in *2022 8th International Conference on Information Technology Trends (ITT)*, 2022, pp. 177–182.

[17] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, "ROS: an open-source Robot Operating System," in *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA) Workshop on Open Source Robotics*, Kobe, Japan, May 2009.

[18] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, 2004, pp. 2149–2154 vol.3.

[19] ROS Components, "Kobuki user guide," https://www.roscomponents.com/es/index.php?controller=attachment&id_attachment=329.

[20] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 2011.

[21] R. B. Rusu, "Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments," *Künstliche Intelligenz*, vol. 24, pp. 345–348, 2010.

[22] L. Huang, S. Chen, J. Zhang, B. Cheng, and M. Liu, "Real-time motion tracking for indoor moving sphere objects with a lidar sensor," *Sensors*, vol. 17, no. 9, 2017.

[23] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "OpenAI Gym," 2016.

[24] A. Ezquerro, M. A. Rodriguez, and R. Tellez, "OpenAI-ROS," http://wiki.ros.org/openai_ros.

[25] A. Gleave, M. Taufeeque, J. Rocamonde, E. Jenner, S. H. Wang, S. Toyer, M. Ernestus, N. Belrose, S. Emmons, and S. Russell, "imitation: Clean imitation learning implementations," arXiv:2211.11972v1[cs.LG], 2022.

[26] 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.

[27] J. Ho and S. Ermon, "Generative adversarial imitation learning," *CoRR*, vol. abs/1606.03476, 2016.

[28] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[29] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *CoRR*, vol. abs/1707.06347, 2017.