University of St Andrews

School of Computer Science
KY16 9BF

Essay

Reinforcement Learning

The Development Towards General Purpose Robotics Since 2020

Florian Pfleiderer

February 17, 2025

Abstract

Reinforcement Learning (RL), a subset of Machine Learning (ML), has contributed to great advances in robotics since 2020, paving the way for the development of sophisticated general-purpose robotics. The ability of robots to optimise behaviour through trial and error using RL has enabled them to perform tasks that could only be done by humans previously, as they required independent decision-making and interaction with the environment. Recent approaches to system design and combinations of methods showed impressive real-world results in highly dynamic environments and real-time behaviour adaption during inference, suggesting great generalisation potential. Despite recent breakthroughs, advances need to be placed in context. Real-world applications like humanoid robots doing parkour give the impression that we are on the verge of seeing general-purpose robots co-existing with humans, but most deployments occurred in controlled, well-known environments, where excessive training using accurate simulations was possible. Robots still face many difficult challenges when faced with complex unknown environments, where little information is known a priori, and simulations are inaccurate. This essay aims to offer insights on recent advances RL has brought to the robotics domain and places them into context by reflecting on the actual development state towards general-purpose robots.

1 Introduction

The robotics domain has evolved rapidly in the past five years due to great advancements in Artificial Intelligence (AI) developments, especially in Machine Learning (ML). The subdomain of Reinforcement Learning (RL) has proven to be the main driver of robotic advances and has contributed greatly to recent successes in real-world applications. Although robotics already saw advances before 2020, most real-world achievements happened after that year and occured in more general and diverse environments. These achievements mark big steps towards general-purpose robots in our daily lives, although there is still a way to go.

Reinforcement Learning has had a big impact on robot development, because robots can learn through trial and error in simulation and real life, and receive positive feedback to achieve defined goals. Through these rewards, they can adjust their behaviour accordingly, similar to how humans learn motor skills. By designing a proper reward function, this method allows robots to perform many *intelligent* tasks, where they make independent decisions and adapt iteratively during the learning process. Although RL itself is not new, systems developed in recent years have achieved remarkable performance above human levels in tasks that are unpredictable and dynamic - an area where RL has previously had weaknesses - by improving algorithms and refining the overall system design. Recent breakthroughs have also shown adaptation during real-world inference, promising potential for efficient real-world learning. Furthermore, environments with multiple robots interacting, called multi agent systems, showed impressive zero-shot generalisation when doubling the amount of agents used in the environment, suggesting a great scaling potential for Multi-Agent Reinforcement Learning (MARL) in future systems. This leads to a vision that is followed by many robotic companies. Robots should be able to do independent work in areas where humans cannot or do not want to work, as well as aiding in many other fields such as healthcare, production, and disaster management.

Despite all these advances, looking closer at many successes in recent years puts them into perspective and context. Much has been achieved in controlled, well-simulated environments, as well as in specific tasks with clearly defined reward functions. Furthermore, most reward functions could be designed a priori and therefore optimised and adjusted according to the situation. General purpose robots, that can be deployed in diverse environments, are still a long way to go. RL still faces many challenges when it comes to real-world use, as most authors show only a few real-world trials to give a more successful impression. Environments that cannot be modelled for simulation, because they are unknown or complex, pose significant obstacles for the real-world applications and would require the agent to adapt reward functions in real time during exploration.

This essay aims to put recent advances, often hyped by the media, in context and to provide a realistic overview of the developments that ML has brought into the robotics domain. Chapter 2 gives an overview of the general terminology and covers alternative developments toward more efficient systems. Chapter 3 provides a technical description of the most common concepts used in RL. It also explains the domain of MARL, as it has huge potential for future developments. Chapter 4 dives into more detail of advances in the deployment and control of quadruped robots and evaluates two very recent studies that used Deep Reinforcement Learning (DeepRL), a combination of Deep Learning (DL) and RL, and MARL for impressive zero-shot sim-to-real performance and present systems with good generalisation potential. Chapter 5 puts all these advances in context by evaluating the actual state of developments and highlighting the open challenges in RL in robotics, as well as covering ethical concerns that need to be addressed before intelligent general purpose systems are created.

2 Reinforcement Learning Across Domains

Reinforcement Learning algorithms enabled great advances across many domains, which have seen advanced and successful real-world deployments much earlier than robotics, and this section aims to provide a comprehensive overview of alternative developments, because developments in different domains influence the overall improvements of RL algorithms, which indirectly help achieving advancements in robotics. To clarify common terms, this section also provides an overview of the terminology and how its used in robotics.

2.1 Terminology in Robotics

The technologies that accelerated the robotics domain can be traced back to the incredible evolution of AI technologies and algorithms in the past decade. While AI is a term that is widely used in different domains, it is necessary to specify it more clearly when it comes to its usage in robotics. AI in robotics is used to describe machines or systems that can perform tasks that humans classify as *requiring intelligence*. This means interacting with its surroundings, making independent decisions, understanding language, and crucially, learning from past experiences. [1]

Machine Learning is a subset of AI that focusses on the specific algorithms that enable robots to learn from data and past experiences to improve over time. It is the main driver behind the success of AI technologies, and different ML methods are used for commercial applications nowadays. In robotics, the ability to learn is crucial for advancing the field. Reinforcement Learning is itself another subset of ML and describes a specific method of learning: A reward and punishment system derived from learning theory in psychology is implemented to enable incremental improvements with every repetition of a task. RL enables robots to learn in interactive, dynamic, and unknown environments and generate their own training data through trial and error runs. [1] The most prominent use case of RL prior to 2020 was the success of AlphaZero, trained through RL to defeat world champions in games such as chess and go [2].

Neural networks evolved from a concept that can be traced down to a conference in 1956. It describes the mimicking of neurons in the human brain by modelling layers of nodes, resembling neurons and their connections mathematically. The concept has since been studied intensively and improved through research of probability and decision theory, development of the backpropagation algorithm, and recently, availability of the necessary computing power, as well as training data. DL further refines Neural Networks by using multiple processing layers, each layer consisting of nodes that are connected to each node of the neighbouring layers. This concept is powerful in pattern recognition of unlabelled data, as well as on labelled datasets. DL has helped image and speech recognition achieve great advances, both being used in robotics. Such algorithms have started to see increased success in their application in robotics in the last decade. [1]

2.2 Alternative Developments

Recent years have brought tremendous advances in incorporating different ML algorithms in various fields: Healthcare uses ML for disease diagnosis and drug development, ML is used for traffic light control, manufacturing has seen improvements through machine vision inclusions, monitoring is used in agriculture to work more efficiently, autonomous driving has seen exceptional test drives in partly unknown environments, and predictive algorithms help prevent issues before they occur, called predictive maintenance. Military robotics has also evolved with better surveillance, decision aiding, and drone control. [3]

Robots have not seen as many successful applications in practice, as they are incredibly complex due to the high Degrees of Freedom (DoF), which results in RL algorithms that become very large and lead to infeasible computational demands. Despite rapid hardware development, especially the evolution of Graphics Processing Units (GPUs) and the development of Tensor Processing Units (TPUs), hardware limitations still limit the applications of RL in robotics. There are different approaches to redesign hardware, most notable the research field of neuromorphic hardware which tries to use non-volatile devices to store data without power usage and moves away from the classical von Neumann architecture for computers. Another approach is called *neuro-evolution* and is a novel method for designing neural networks faster and more efficiently. This technique was not used much in robotics prior to 2020, but can be combined with gradient-based methods to increase the network efficiency. [4], [5]

In 2019, Nachum, Ahn, Ponte, *et al.* [6] achieved a successful real-world application in robotics. They managed to use multiple robot agents to manipulate objects in environments. The challenge here was to overcome the extensive real-world training that would have been needed to manage the interactions between the quadruped agents and the objects. Although they achieved this in 2019, the task was rather simple: Only two quadruped agents needed to coordinate to push a block into a specific position, as shown by markers on the floor. More complex tasks become exponentially more challenging, but this study indicated potential in MARL.

The COVID-19 pandemic has also had an impact on the development of robotic systems. It has shown the necessity of using robots to keep critical infrastructure running and provide essential services such as aiding in hospitals, warehouse logistics, providing delivery services and disaster management. By combining ML advances with robotics, agents are slowly going from performing scripted industrial tasks to independent decision making, manoeuvring unknown environments and social interactions. [7] Despite all advances in algorithms and hardware development, Singh, Kumar, and Singh [5] outlines significant limitations in both intelligence and self-learning abilities as of 2021, but argues that reinforcement learning is the most potent framework for future evolution in robotics.

3 Reinforcement Learning Theory

There are many different algorithms used in RL and this section should provide an overview of the most important concepts that are used as foundations for recent real-world successes.

3.1 Agent and Environment

In RL, the Agent-Environment Interface (AEI) is the foundational model used to design algorithms. The *agent* is the learning entity and interacts with the *environment* continuously. The interaction follows discrete steps, representing different *states* $S_t \in S$ of the environment. when the agent performs an action $A_t \in A(s)$, the environment enters a new state S_{t+1} and the agent receives a reward $R_{t+1} \in R \subset \mathbb{R}$, which will be used to perform the next action. [8]

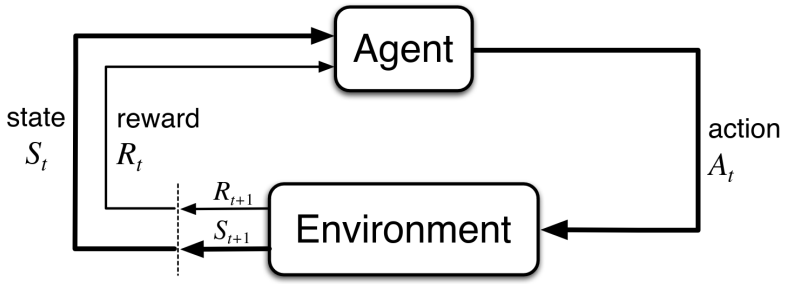

Figure 1: The interaction between agent and environment. Each completed cycle results in a new state with a new set of possible actions. [8]

3.2 Goals and Rewards

The reward $R_t \in \mathbb{R}$ is the scalar signal received after performing an action and entering a new state. It can be seen as a reflection of how *good* the action was and the agent will try to maximise the value of this reward over the long term. This concept of formulating the goal of maximising long-term reward is a distinctive feature that makes RL different from other ML algorithms. This approach makes RL usable in a wide variety of situations, it just depends on defining good reward signals: To teach walking, the reward can be given based on the distance covered in a specific direction, for searching specific objects, the reward can be negative for wrong objects and positive for right ones. The important takeaway is to design the reward in a way that the agent actually achieves the task, not by rewarding small steps along the way, which would be telling the agent *how* we would like the task to be achieved. RL should only set constraints to define *what* the agent should achieve. [8]

3.3 Returns and Episodes

The cumulative reward expected in RL is called the *return* and can be defined as:

$$G_t \doteq R_{t+1} + R_{t+2} + ... + R_T \tag{1}$$

Although this is highly simplified, it is suitable for applications where discrete steps are part of the interaction, like board games, where it is easy to define a clear final time step. These steps are called *episodes* and the tasks are called *episodic tasks*. Another

characteristic of episodic tasks is that every episode begins independently of how the previous one ended. Many real-world tasks cannot be broken down into episodes; these are called *continuing tasks* and Eq.1 is not applicable for those. The reason being the nature of continuous tasks, as the final time step would be $T = \infty$, making the series divergent. To counteract this, a new concept called *discounting* is introduced, which allows the agent to maximise the expected *discounted return*:

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ... = \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k \tag{2}$$

The parameter γ is the discount rate and determines whether present or future rewards are weighted more heavily. T is the *horizon*, which can be infinite. For $\gamma = 0$, the agent maximises immediate rewards and is called *myopic*, whereas for $0 < \gamma < 1$, future rewards are respected more strongly for higher values of γ, and the agents adjust their actions accordingly. Note that $\gamma = 1$ is possible, but for the series to be convergent, $T < \infty$ must apply in such cases. [8]

3.4 Policies and Value Functions

Value functions define how *desirable* a certain state is or define the value of each action in a certain state. The metric used for valuing the states or actions is the expected return, defined in Eq.2. Furthermore, each action has a certain probability $\pi(a|s)$, that is, performing an action $A_t = a$ when the agent is in state $S_t = s$. The mapping π from a state to all actions that follow it is called the *policy*. This mapping can be used to describe the expected reward for certain actions following a state:

$$q_\pi(s, a) = E_\pi \left[G_t | S_t = s, A_t = a \right] \tag{3}$$

The goal of reinforcement learning tasks is to find the optimal policies π_*, who share an optimal action-value (Q) function, so that the expected return is maximised:

$$q_*(s, a) \doteq \max_\pi q_\pi(s, a) \tag{4}$$

3.5 Markov Decision Process

The whole process of agents learning about the optimal policy by interacting with the environment in discrete steps can be modelled by a Markov Decision Process (MDP). It includes the environment S, the action space A, the probabilities of state-action transition P, the reward $R : S \times A \times S \to \mathbb{R}$ and the discount rate $\gamma \in [0, 1]$ can be noted as a tuple $\langle S, A, P, R, \gamma \rangle$. The constraint here is that the transition probabilities P need to be well defined. To help solving a MDP and derive optimal policies, the optimal value functions must satisfy *Bellman optimality equations*. [5] In practice, there will be much more uncertainty about states and transitions in the MDP. Thats why the partially-observable Markov Decision Process (POMDP) is introduced as a generalisation of the finite MDP and it includes an additional probability of observing certain values at certain states. A POMDP is more complex, but better suited for many real-world RL applications. [9]

3.6 Multi Agent Systems

MARL is an algorithm that enables multiple agents to act in the same environment, each agent solving their own MDP and optimising their policies. The key difference in MARL is that the next state and the expected reward are based on the results of all other agents actions within the environment. The resulting situation is called non-Markovian, and new methods need to be used to model such systems. [10] These methods include the introduction of Markov Games (MG), where the set of agents $N > 1$ is added to the tuple and the action space A expands into a list of spaces $\{A^i\}_{i \in \mathbb{N}}$ per agent and the reward expands to $\{R^i\}_{i \in \mathbb{N}}$. [9]

While in theory, the perfect knowledge of an environment could help finding the optimal value function, the virtual model of the environment is usually incomplete. Even if there were exact replicas of real-world scenarios, it would then be impossible to compute. [11] For this reason, RL deals with approximating such optimal solutions in different ways to achieve realistic computational demands. Finding this approximation is very hard, the more dynamic and complex an environment gets, which is especially a problem for real world robotic applications. Despite those difficulties, there have been advancements in recent years, and Tang, Abbatematteo, Hu, *et al.* [11] have shown that RL and deep neural networks have performed well even at higher dimensional environments. This combination is called DeepRL. [8] The next chapter will outline recent successes using different algorithms and techniques to optimise RL problems.

4 Recent Successes in Real World Applications

For anyone following the media coverage around robotics, the past 4 years have given the impression that the domain has made big steps towards developing a general purpose robot and there is lots of hype, but also fear towards robotics. This Chapter introduces different successful deployments of RL algorithms in real-world scenarios that have been achieved since 2020.

4.1 Quadruped Robots

The robotics company Boston Dynamics[1] had a lot of media coverage of their successes, showing the growing public interest in these areas. The sales launch of their quadruped robot *spot* sparked lots of attention and can be seen as the first commercial milestone of applied reinforcement learning in quadruped robots. [12] This was followed by the commercial launch of *ANYmal* by ANYbotics[2] in 2022, a similar quadruped robot especially targeted at inspection tasks in oil, gas, and chemical industries. [13] In the same year,

Figure 2: Inspection robot in use at an oil offshore platform. [13]

Stretch, a wheeled robot for warehouse and distribution appliances, was launched, received great interest from the delivery and production industry and sold out quickly. In 2023, Boston Dynamics released a new demo of *Atlas*, a dynamic humanoid robot, now capable of running, jumping, and carrying heavy items. Especially Atlas achieved high media coverage due to its human-like capabilities and looks. [12] The two companies are not the only players in the field; Swiss-Mile[3], founded in 2023 as a spin-off from ETH Zurich already launched its quadruped competitor and UniTree[4], a Chinese competitor, builds quadruped and humanoid robots. Furthermore, a company called Agility Robotics[5] opened the first humanoid robot factory called *RoboFab* in September 2023, where their robot called *Digit* is being tested. They also claim that the robot should work safely alongside humans to aid in meeting increasing production demands. Robots working independently alongside humans in a safe manner will be a huge leap in the field, and they have set up a future testing area by building RoboFab. [14]

A common algorithm used is called Proximal Policy Optimisation (PPO), which performs well in a wide range of hyperparameter settings. This speeds up training, as parameter tuning can be kept to a minimum. [11] These developments look incredible to unknown viewers, but most seem more advanced than they are, as testing and training environments

[1] https://bostondynamics.com/
[2] https://www.anybotics.com/
[3] https://www.swiss-mile.com/
[4] https://www.unitree.com/
[5] https://agilityrobotics.com/

are well known and well defined. This allows operators to design dense reward functions that lead to quick learning. Moreover, their simulation environments are very good representations of the environment space, which greatly aides the zero-shot sim-to-real transfer. Moreover, they are incredibly energy consuming, which makes Atlas, for example, not run much longer than an hour [15]. Chapter 5 will outline the difficulties that the robotics domain faces when tackling more generalised real-world applications, where reward functions cannot be easily designed and environments cannot be modelled accurately.

4.2 Highly Dynamic Environments

A research group at Google DeepMind has used RL to achieve intermediate-level performance in table tennis, a highly dynamic and unpredictable environment. Robot table tennis has very demanding requirements on coordination, control, and decision making and has seen much progress after 2020 with hitting specified targets and even simple rallying achieved in 2023, but this approach struggled playing previously unseen opponents with different styles. [16]

To overcome the limitations of RL in this dynamic environment, they use a combination of RL and Imitation Learning (IL). IL works by capturing demonstrations of tasks and learning a mapping that serves as a policy. The limitations here are, that the demonstrations can only cover small subsets of all possible motions, so refinement of the policy is needed. [17] This combinations allows the use of real play data to produce a set of initial ball positions for the task. The robot is then trained in simulation and finally deployed on real hardware. D'Ambrosio, Abeyruwan, Graesser, *et al.* [18] then use the inference data to increase the set of conditions and the training-deployment cycle is repeated. The agent architecture is laid out as follows: The systems consist of Low Level Controllers (LLCs) that resemble a physical skill library. These low level policies are trained separately from each other so that each skill like forehand cross and other hitting styles gets trained individually to avoid forgetting previously learnt skills. The High Level Controller (HLC) is triggered by the moment the opponent hits the ball, takes as input the ball state, the style policy, the library of LLCs and other variable about the current state. The HLC then makes a decision in the subsequent time step and selects a LLC. This selection process is also taken under the physical constraints of the system, so it does differ between the theoretical best policy and the policy with the highest confidence of execution. During testing against human opponents, the results show that the agent not only adapted different policies during the games, but the HLC also developed different strategies against the beginners and selected different LLCs depending on the skill level, shown in Fig. 3. This indicates the ability of the system to distinguish between skill levels and adapt to playing styles and strategies.[18]

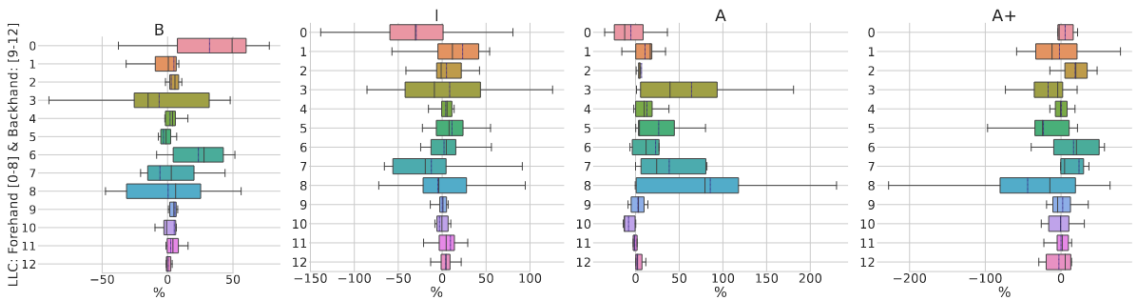

Figure 3: The change in policy adaptation after three games. The values show the percentage change in preferences for each LLC against all four skills levels. [18]

D'Ambrosio, Abeyruwan, Graesser, *et al.* [18] have developed a system that they expect to have broader implications beyond table tennis. Firstly, a hierarchical policy architecture

is successfully implemented, which consists of LLC skill descriptors and HLC skill selectors. This approach of combining different policies in a hierarchical structure allows for continuous updating of skill policies based on new real-world experiences. This approach can be used in many other scenarios where a skill set can be defined a priori. Secondly, they used IL and real-world data to define initial task conditions, which greatly reduces the simulation time, as the real-world data covers realistic initial conditions that would occur more frequently. Overall, they showed that system design is an area where much progress can be made in future research.

4.3 Multi-Agent Systems

Haarnoja, Moran, Lever, *et al.* [19] recently published an innovative paper about robot soccer, a long standing competition in AI and robotics since the introduction of the *RoboCup*[6], showcasing the full potential of MARL. Although many RoboCup competitors focus on transferring skills taught to a single robot to multiple models, their paper focused on simulating *1v1* games for training, including a second agent into the training process. Successes in the RoboCup are often achieved by highly optimised system components, targeting the special nature of the robot soccer game and sequencing together walking, kicking, and getting up tasks separately. Using a humanoid robot and approaching a more generalised training to learn the ball game is a far more difficult challenge. The policy learnt in the method of Haarnoja, Moran, Lever, *et al.* [19] managed to achieve impressive results against a manually designed baseline controller, with the robots turning and running speeds increasing by more than 100% and the time it took agents to get up after falling also reduced by over 50%. They also managed to keep the sim-to-real gap relatively small and the agent only performed about 15% worse than in simulation, still outperforming the baseline controller by a lot. Another impressive result was achieved, when their method was applied to *2v2* robot soccer without additional training. The agents were able to collaborate on tasks, such as letting the closer robot approach the ball, showing collaboration capabilities. This result promises great scaling capabilities for their system and has great potential to tackle the challenge of scaling MARL.

[6] https://www.robocup.org/

5 Breaking Down Recent Successes

Despite all the hype around Robots since 2020, all the groundbreaking advancements, and the media coverage that it created, different methods of RL are still very limited and do not see great commercial applications. Robotics companies are still far from profitable, and many have shut down after the costs exceeded expectations. The following sections will go into detail about the limitations and ethical concerns.

5.1 General Limitations

In many settings where successes have been achieved, like quadruped locomotion and object manipulation, the environment was well known and pre-designed dense reward functions could be used to train the agent. The success rate drops with more complex and diverse environments, where a priori designed reward functions are generally more sparse. Such tasks include generalised pick up and place scenarios like clean up tasks in households or exploration of unknown environments. This is one of many factors for why robotics is not as advanced as it seemed in successful demonstration videos over the past couple of years. The problem of sparse reward functions can be tackled using IL, as done by D'Ambrosio, Abeyruwan, Graesser, *et al.* [18] to set starting boundaries for learning. This approach does not work when no demonstrations can be given, as would be the case with exploration tasks in unknown environments. [11]

Sample efficiency is how many training samples an agent needs for optimal policy learning. The sim-to-real gap present in zero-shot applications requires further real-world training for better policy optimisation and real-life training is costly and often time-consuming, so there is a great need for more efficient algorithms to reduce the number of required episodes to achieve desired policy optimisation. Sample efficiency is not only a problem in real-world deployment, simulations can also be time consuming and fewer episodes would speed up the training process. This is especially important in more complex MARL systems.

Another limitation in modern-day robotics is real-world-learning in general. This is needed in complex scenarios where simulations do not represent the environments well enough to use zero-shot transfer methods. As simulations are very limited in any open world scenarios due to the high diversity of the settings, better real-world-learning methods are highly demanded. One problem that arises is the data collection. As mentioned, IL has limitations in generalised open world environments and, thus, training needs to work with reduced human impact. A crucial factor here is to maintain high safety standards, including algorithms for automatic resets and mechanisms to allow robots to explore environments in a safe manner. These two concepts are not present in most deployments, as they are done in lab settings. Another point regards the acceleration of training. Exploring alternatives to higher sample efficiency, such as introducing incremental, adaptive updates instead of entire policy changes to speed up training, is another area of research, but still challenging. [11]

Another limitation evident in recent successes is the short horizon of learning. Most successful deployments are trained for relatively short horizons, such as hitting a ball, collision avoidance and manipulation of objects. Algorithms that focus on reward assignment over long stretches of time are needed to create robots that can be used for varieties of general tasks. D'Ambrosio, Abeyruwan, Graesser, *et al.* [18] have shown a promising approach, learning small skills using separate policies and designing a compositional controller to deploy certain skills. While this could achieve success in more generalised applications than table tennis, where the skill set is easy to choose, approaches that rely less on domain knowledge and more general skillsets are needed. There is promising work

on goal-conditioned and unsupervised RL that tackle the problem, but this is not widely used. Both designed skillsets and unified skill learning do not tackle the task of combining learned skills. There have been several studies focusing on hierarchical RL and end-to-end approaches, moving towards the goal of general-purpose robots.

Tang, Abbatematteo, Hu, *et al.* [11] tried to benchmark real-world success, but found that the only source of information about the results are the authors and, often, only a single trial run by them. This makes it inherently difficult to judge real-world success, so standardised evaluation platforms and reproducible tests need to be introduced to measure future applications of real world robotics.

Looking at other developments in AI, it also stands out that foundation models, meaning large-scale RL models that serve as baselines for various domains, such as Large Language Models (LLM), are missing in RL in robotics. Tang, Abbatematteo, Hu, *et al.* [11] expect much more future integration of foundation models into RL in real-world applications. One notable achievement is to leverage LLMs to aid and automate sim-to-real design. They achieved comparable results when comparing a LLM-designed sim-to-real transfer with a human-made design. The quadruped agent with a LLM designed policy managed to outperform the latter in terms of forward velocity and managed to walk 15% further in more difficult conditions like on grass or wearing socks. They propose using their pipeline to automate difficult design aspects in the sim-to-real transfer. [11]

5.2 Multi-Agent Systems

Multi-Agent Systems face even harder challenges, as Markov Games are already hard to solve. Going from controlled to unpredictable environments presents significant challenges, but having multiple agents introduces much more sources for failures and inaccuracies, so the systems need to be designed much more robust to overcome potential defects. Apart from that, increasing the number of agents adds more complexity for finding optimal overall policy, as every agents action influences the environment state for all participating agents. Despite the promising result proposed by Haarnoja, Moran, Lever, *et al.* [19] it is still very difficult to scale the MARL algorithms and needs future research. [11]

5.3 Safety and Ethics

Although the domain has not reached that state, general purpose robots are just around the corner, and there are still many open challenges to overcome, until robots can be safely deployed to unknown environments, equipped with general skillsets and the ability to learn and adapt to many different circumstances, talking about safety and ethics is crucially important for the next decades. Going into detail is not within the scope of this essay, but the topic should be covered nevertheless, since military organisations around the world have great interest in robotics. For example, the US military agency DARPA helped in founding an early version of *Atlas* [20], developed by Boston Dynamics. While using robots for war to reduce human casualties could be morally acceptable, experts predict that robots are more likely support soldiers in battle. Such robots would lead to decisions about life or death taken solely by AI robots and to counteract such development, many of the leading robotic companies have released a signed pledge to commit to exploring technological features to counteract the risk of their robots being used for warfare. They also pledge to careful customer intent evaluation and demand the help of policymakers to aid this development towards a safe future. [21]

6 Conclusion

To conclude, the field of robotics has seen incredible achievements and successful real-world applications since 2020, that showed promising generalisation and real-time adaptation capabilities. By combining different methods like Imitation Learning and Reinforcement Learning (RL), or using hierarchical system design, recent papers managed to provide promising techniques to improve generalisation capabilities and sim-to-real efficiency. Scaling up systems to include multiple agents by including participating agents into the training process shows promising results for future developments. However, most successful deployments have been conducted in controlled, well known environments and many challenges still remain. RL still struggles in unknown environments, where only sparse reward functions can be designed. The complexity of real-world scenarios hinders the zero-shot sim-to-real transfer and makes real-world training necessary. Furthermore, multi-agent systems face significant increases in complexity when scaling them to include more agents. Robotics is going in a promising way toward general purpose usage, but the current state needs to be seen through a realistic lens, focusing on challenges ahead. In addition to that, building more sophisticated or even general-purpose robots requires careful regulation. This issue arises because military purposes of such machines are evident and there is a need for guidelines of future robot development and usage. As RL continues to evolve, overcoming these limitations will be crucial to unlocking the full potential of robotics in both commercial and general-purpose applications. The future of robotics, therefore, lies not only in technological advancements but also in addressing these ethical and practical concerns.

References

[1] J. Howard, "Artificial intelligence: Implications for the future of work," *American Journal of Industrial Medicine*, vol. 62, no. 11, pp. 917–926, Nov. 2019, ISSN: 0271-3586, 1097-0274. DOI: 10.1002/ajim.23037. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/ajim.23037 (visited on 10/04/2024).

[2] D. Silver, T. Hubert, J. Schrittwieser, *et al.*, *Mastering chess and shogi by self-play with a general reinforcement learning algorithm*, Dec. 5, 2017. DOI: 10.48550/arXiv.1712.01815. arXiv: 1712.01815[cs]. [Online]. Available: http://arxiv.org/abs/1712.01815 (visited on 10/08/2024).

[3] M. Soori, B. Arezoo, and R. Dastres, "Artificial intelligence, machine learning and deep learning in advanced robotics, a review," *Cognitive Robotics*, vol. 3, pp. 54–70, Jan. 1, 2023, ISSN: 2667-2413. DOI: 10.1016/j.cogr.2023.04.001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2667241323000113 (visited on 10/04/2024).

[4] K. O. Stanley, J. Clune, J. Lehman, and R. Miikkulainen, "Designing neural networks through neuroevolution," *Nature Machine Intelligence*, vol. 1, no. 1, pp. 24–35, Jan. 2019, Publisher: Nature Publishing Group, ISSN: 2522-5839. DOI: 10.1038/s42256-018-0006-z. [Online]. Available: https://www.nature.com/articles/s42256-018-0006-z (visited on 10/07/2024).

[5] B. Singh, R. Kumar, and V. P. Singh, "Reinforcement learning in robotic applications: A comprehensive survey," *Artificial Intelligence Review*, vol. 55, no. 2, pp. 945–990, Feb. 1, 2022, ISSN: 1573-7462. DOI: 10.1007/s10462-021-09997-9. [Online]. Available: https://doi.org/10.1007/s10462-021-09997-9 (visited on 10/04/2024).

[6] O. Nachum, M. Ahn, H. Ponte, S. Gu, and V. Kumar, *Multi-agent manipulation via locomotion using hierarchical sim2real*, Oct. 7, 2019. DOI: 10.48550/arXiv.1908.05224. arXiv: 1908.05224[cs]. [Online]. Available: http://arxiv.org/abs/1908.05224 (visited on 10/11/2024).

[7] Z. Zeng, P.-J. Chen, and A. A. Lew, "From high-touch to high-tech: COVID-19 drives robotics adoption," *Tourism Geographies*, vol. 22, no. 3, pp. 724–734, May 26, 2020, Publisher: Routledge _eprint: https://doi.org/10.1080/14616688.2020.1762118, ISSN: 1461-6688. DOI: 10.1080/14616688.2020.1762118. [Online]. Available: https://doi.org/10.1080/14616688.2020.1762118 (visited on 10/07/2024).

[8] R. S. Sutton and A. Barto, *Reinforcement learning: an introduction* (Adaptive computation and machine learning), Second edition. Cambridge, Massachusetts London, England: The MIT Press, 2020, 526 pp., ISBN: 978-0-262-03924-6.

[9] L. Canese, G. C. Cardarilli, L. Di Nunzio, *et al.*, "Multi-agent reinforcement learning: A review of challenges and applications," *Applied Sciences*, vol. 11, no. 11, p. 4948, Jan. 2021, ISSN: 2076-3417. DOI: 10.3390/app11114948. [Online]. Available: https://www.mdpi.com/2076-3417/11/11/4948 (visited on 10/03/2024).

[10] Z. Zhou, G. Liu, and Y. Tang, "Multi-agent reinforcement learning: Methods, applications, visionary prospects, and challenges," *IEEE Transactions on Intelligent Vehicles*, pp. 1–23, 2024, ISSN: 2379-8904, 2379-8858. DOI: 10.1109/TIV.2024.3408257. arXiv: 2305.10091[cs]. [Online]. Available: http://arxiv.org/abs/2305.10091 (visited on 10/03/2024).

[11] 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*, Sep. 16, 2024. DOI: 10.48550/arXiv.2408.03539. arXiv: 2408.03539[cs]. [Online]. Available: http://arxiv.org/abs/2408.03539 (visited on 10/04/2024).

[12] L. Ross. "How boston dynamics is leading the robotics revolution." (Feb. 15, 2024), [Online]. Available: https://www.thomasnet.com/insights/how-boston-dynamics-is-leading-the-robotics-revolution/ (visited on 10/10/2024).

[13] dwood. "World's only scalable ex-proof robot inspection solution now available to industry," ANYbotics. (Sep. 28, 2022), [Online]. Available: https://www.anybotics.com/news/anymal-x-commercial-launch-sprint-conference/ (visited on 10/10/2024).

[14] L. Kolodny. "Agility robotics is opening a humanoid robot factory, beating tesla to the punch," CNBC. Section: Technology. (Sep. 18, 2023), [Online]. Available: https://www.cnbc.com/2023/09/18/agility-robotics-is-opening-a-humanoid-robot-factory-.html (visited on 10/15/2024).

[15] S. M. Group. "DARPA's upgraded atlas robot has onboard power and wireless communication." (), [Online]. Available: https://www.techbriefs.com/component/content/article/31078-darpa-80-99s-upgraded-atlas-robot-h (visited on 10/15/2024).

[16] S. Abeyruwan, L. Graesser, D. B. D'Ambrosio, *et al.*, *I-sim2real: Reinforcement learning of robotic policies in tight human-robot interaction loops*, Nov. 22, 2022. DOI: 10.48550/arXiv.2207.06572. arXiv: 2207.06572. [Online]. Available: http://arxiv.org/abs/2207.06572 (visited on 10/10/2024).

[17] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne, "Imitation learning: A survey of learning methods," *ACM Comput. Surv.*, vol. 50, no. 2, 21:1–21:35, Apr. 6, 2017, ISSN: 0360-0300. DOI: 10.1145/3054912. [Online]. Available: https://dl.acm.org/doi/10.1145/3054912 (visited on 10/10/2024).

[18] D. B. D'Ambrosio, S. Abeyruwan, L. Graesser, *et al.*, *Achieving human level competitive robot table tennis*, Aug. 9, 2024. DOI: 10.48550/arXiv.2408.03906. arXiv: 2408.03906[cs]. [Online]. Available: http://arxiv.org/abs/2408.03906 (visited on 10/05/2024).

[19] T. Haarnoja, B. Moran, G. Lever, *et al.*, "Learning agile soccer skills for a bipedal robot with deep reinforcement learning," *Science Robotics*, vol. 9, no. 89, eadi8022, Publisher: American Association for the Advancement of Science. DOI: 10.1126/scirobotics.adi8022. [Online]. Available: https://doi.org/10.1126/scirobotics.adi8022 (visited on 10/11/2024).

[20] F. Van Allen. "The deadly, incredible and absurd robots of the US military," CNET. (Feb. 8, 2017), [Online]. Available: https://www.cnet.com/pictures/deadly-incredible-absurd-robots-the-us-military/ (visited on 10/15/2024).

[21] "General purpose robots should not be weaponized," Boston Dynamics. (Oct. 2022), [Online]. Available: https://bostondynamics.com/news/general-purpose-robots-should-not-be-weaponized/ (visited on 10/15/2024).

Acronyms

AEI Agent-Environment Interface. 5

AI Artificial Intelligence. 2, 3, 10, 12

DeepRL Deep Reinforcement Learning. 2, 7

DL Deep Learning. 2, 3

DoF Degrees of Freedom. 4

GPU Graphics Processing Unit. 4

HLC High Level Controller. 9, 10

IL Imitation Learning. 9–11, 13

LLC Low Level Controller. 9, 10

LLM Large Language Models. 12

MARL Multi-Agent Reinforcement Learning. 2, 4, 7, 10–12

MDP Markov Decision Process. 6, 7

MG Markov Games. 7, 12

ML Machine Learning. 1–5

POMDP partially-observable Markov Decision Process. 6

PPO Proximal Policy Optimisation. 8

RL Reinforcement Learning. 1–9, 11–13

TPU Tensor Processing Unit. 4