Institute of Engineering and Technology (IET)

“Revolutionizing Robotics: Unleashing the power of Reinforcement Learning in Real World Applications"

PREPARED BY

Jayani Abhiram (2021btech050)

Poorna Bhati (2021btech084)

FACULTY GUIDE

Dr. Rajlakshmi Nayak

April 2024

TABLE OF CONTENTS

CERTIFICATE

This is to certify that the project work entitled "**Revolutionizing Robotics: Unleashing the power of Reinforcement Learning in Real World Applications**" submitted by **Poorna Bhati (2021btech084) and Jayani Abhiram (2021btech050)** towards the requirements for course *PR1103 Minor Project* is the record of work carried out by them under our supervision and guidance. I believe the submitted work has reached the level required for acceptance.

Date of Submission: 01/05/2024

-------------------------------	-----------------------------------
Dr. Rajlakshmi Nayak	Dr. H.P. Agrawal
Assistant Professor	Assistant Professor, EEE
Coordinator, Office of International Affairs	Exam Coordinator, IET
Institute of Engineering & Technology	Institute of Engineering & Technology
JK Lakshmipat University Jaipur	JK Lakshmipat University Jaipur

ABSTRACT

Reinforcement learning (RL) is helping robots become smarter by allowing them to learn and adapt in real-world situations. In this paper, we explore how RL is transforming robotics, enabling robots to handle difficult tasks and challenges more effectively. We discuss how robots can learn from their experiences independently, without constant human guidance. This ability makes robots more efficient and safer, especially in industries like manufacturing, healthcare, and entertainment. However, applying RL in real-life situations presents challenges. One major challenge is the gap between what robots learn in practice simulations and what they encounter in the real world. [i]We need to ensure that what robots learn in one situation can be applied to others. Additionally, it's crucial to ensure that robots make decisions that people can trust and understand. Overall, this paper aims to demonstrate how RL is revolutionizing robotics and how we can address challenges to enhance robot performance. We achieve this by using specific methods and strategies such as policy based and value based methods tailored to each application. Our ultimate goal is to create robots that seamlessly collaborate with people, enhancing safety and efficiency for everyone.[ii]

PROBLEM STATEMENT

Despite 90% of robot training happening in simulations, the gap between simulation and reality hinders real-world use. This research paper tackles the challenge by developing methods for robots trained with reinforcement learning to seamlessly transfer skills to real-world scenarios. Additionally, this explores the techniques to make robot decision-making transparent and trustworthy, paving the way for robots that can collaborate with humans as reliable and understandable partners.

INTRODUCTION

In recent years, robots and reinforcement learning (RL) have teamed up to do amazing things. RL helps robots learn from their mistakes and adapt to different situations, making them smarter and more versatile. This partnership has big potential in many areas like manufacturing, healthcare, and entertainment. Traditionally, robots followed strict rules to do their jobs, but they struggled with changing environments. RL changes that by letting robots learn by doing, just like how we learn from trying new things. This means robots can keep getting better without needing constant instructions. RL also helps robots deal with the gap between what they learn in simulations and what they face in real life. But using RL in important areas like safety means making sure robots make decisions we can trust and understand. In this paper, we explore how RL is changing robotics and what challenges we need to tackle to make it even better. Our goal is to make robots that fit seamlessly into our lives, making things easier and safer for everyone.[iii]

Reinforcement learning (RL):

Reinforcement learning is like teaching a robot how to do things by letting it try and learn from its mistakes. The robot does stuff, and depending on whether it does good or bad, it gets rewards or punishments. [iv]The robot's aim is to figure out the best way to do things over time, so it gets more rewards.

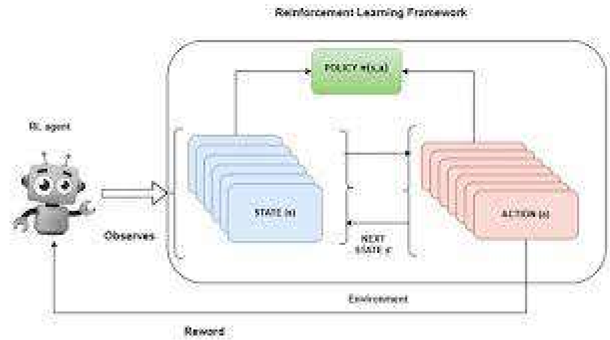

Figure 1: Reinforcement Learning Framework [v]

Here's a breakdown of the key components of reinforcement learning: Reinforcement learning is when a learner, like a robot, does things in a place and gets either good or bad feedback. The place changes as the learner does stuff, and this affects what happens next and how good the feedback is. The robot's goal is to figure out the best way to do things by trying different actions and seeing what happens. It wants to get as many rewards as possible over time, so it keeps learning and getting better at making decisions in its environment. In reinforcement learning, there are two main ways to teach robots: one is by focusing on values, and the other is by focusing on policies. With value-based methods, robots figure out how good different actions are in different situations. They use algorithms like Q-learning, where they learn the values of different actions they can take. Another method called Deep Q Networks (DQN) helps robots estimate these values using something called neural networks. With policy-based methods, robots learn the best way to act without worrying about values. They use methods like Policy Gradient, where they adjust how they act to get more rewards over time. Another method called Actor-Critic helps them improve their actions by combining value estimation and policy improvement. These methods give robots different ways to learn and make better decisions in different situations. Reinforcement learning is useful in many areas like robots, games, money, health, and more. It's been used to teach robots to play hard games like Go and StarCraft, save energy in buildings, move robot arms, and even drive cars without humans. Basically, it's a smart way for machines to learn, decide, and solve tricky problems by dealing with what's around them. This paper talks about how to use reinforcement learning (RL) in robots. It

explains different ways to teach robots and how to make them work well. Also, it discusses the problems faced in teaching robots and how to solve them to make them better. [vi]
Here are two example tasks. This updated version talks about new things like using robots for archery and what we can expect in the future. It also compares different ways of teaching robots and discusses their strengths and weaknesses. They made some improvements, like making certain points stand out more and explaining things in more detail. Even though they didn't come up with new teaching methods, they gathered a lot of information over the past few years to help make robots better. The study suggests that to make robots learn better, we need to use specific ways of teaching them. The main idea is to understand how to teach robots to do different tasks better.

METHODOLOGY

Advanced Reinforcement Learning Techniques for Robotics

Cutting-edge RL Algorithms

We're working hard to use the best reinforcement learning methods to make robots better in real-world situations. These advanced techniques help robots learn, adjust, and do well in tricky and changing environments, making them much more capable. Some of the promising new methods.

Deep Reinforcement Learning (DRL): Using a mix of reinforcement learning and deep neural networks has helped robots learn from complicated sensory information. Methods like Deep Q-Networks (DQN), Proximal Policy Optimization (PPO), and Soft Actor-Critic (SAC) have worked well in different robot tasks, like moving objects or controlling movements.[vii]

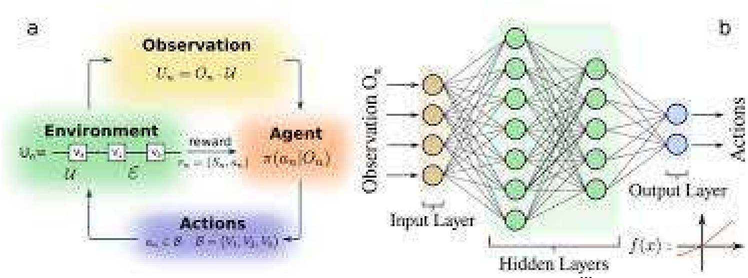

Figure 2: Deep Reinforcement Learning [viii]

The picture shows how a robot learns from its surroundings. In the first image (a), the robot gets information from its environment, thinks about it, and then does something that changes how the environment looks. In the second image (b), it shows how the robot's brain works. It takes in the information, processes it, and then decides what to do. This process keeps repeating, with the robot learning from the good and bad things that happen, and trying to make better decisions to get more rewards in the long run.

Model-Based Reinforcement Learning: Model-based reinforcement learning methods use learned or guessed models of the environment to learn faster and make better decisions. By using these models, robots can think ahead and make smarter choices, which helps them learn quicker and do better.[ix]

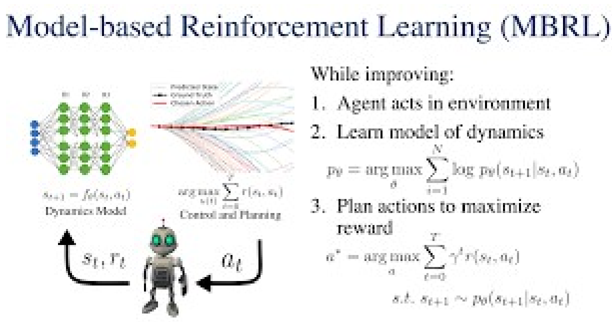

Figure 3: Model Based Reinforcement Learning [x]

The picture shows how Model-Based Reinforcement Learning (MBRL) works. In MBRL, the robot interacts with its surroundings and collects information like what it did, what happened next, and if it got a reward. This information helps the robot learn how the environment behaves.

Once it learns this, it can predict what might happen in the future and make smart decisions to get the best rewards. This way, the robot can try things out in a safe simulation instead of the real world, which saves time and money. The robot keeps getting better as it learns more, which helps it plan and make decisions even better.

Hierarchical Reinforcement Learning: Hierarchical reinforcement learning lets robots learn and think in different layers, making it easier for them to learn complicated behaviors and skills. Methods like Hierarchical Deep Q-Networks (h-DQN) and Hierarchical Actor-Critic (h-AC) help robots break down tasks into smaller parts and tackle them step by step.

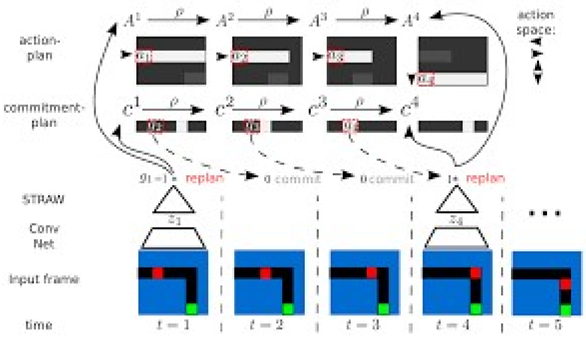

Figure 4: Hierarchical Reinforcement Learning [xi]

The picture shows a setup of computers and storage devices connected together. The computers, shown as rectangles, are grouped into sections called racks. They talk to each other through a network, like a local or wide area network. This setup helps with doing lots of tasks at the same time, keeping data safe, sharing the workload evenly, and making it easy to add more computers or storage space when needed.

Meta-Reinforcement Learning: Meta-reinforcement learning methods help robots quickly adjust to new tasks and places by using what they've learned before. This helps robots become better at learning, which makes them work more effectively and reliably on many different tasks.

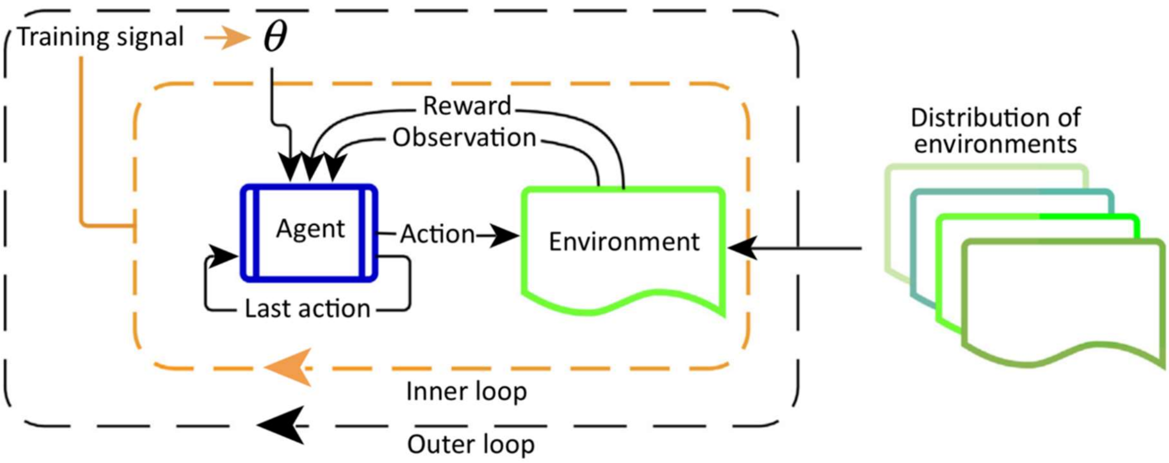

Figure 5: Meta Reinforcement Learning [xii]

This picture shows how we train a robot using reinforcement learning. The "Training pipe" is where the robot does its tasks and interacts with its surroundings. It takes actions based on what's happening, which leads to new situations and either rewards or punishments. The "Distribution of returns" shows how much rewards the robot gets over many tries, which helps us teach it better. We use this information to help the robot make better decisions as it learns, making it better at its job over time. This process keeps repeating, with the robot getting smarter and better at its tasks as it goes along.

Exploration Strategies and Challenges in Policy Representation:

Fancy ways of exploring, like being curious or wanting to learn new things, help robots find better ways to do stuff and learn faster when they're in places they don't know. These methods make robots try different things and learn from their experiences, which makes them better at their tasks. By using these advanced techniques, we can make big improvements in how robots work and solve tricky problems. These methods help robots become smarter, adapt to new situations, and do better in real-life situations, which leads to exciting advancements in autonomous robots. To make robots better using reinforcement learning, we need to solve some problems with how we tell them what to do. These problems involve figuring out the best ways to guide robots in making decisions and doing tasks. For robots to do their job well, they need to understand how things are connected and be able to adjust to changes quickly. They should handle different tasks easily and think about all the information around them, not just what's nearby. [xiii]They also need to manage a lot of information without getting confused. Lastly, they need to find the best solutions without taking too long. It's important to solve these problems so that robots can use reinforcement learning effectively and do their tasks better. By coming up with better ways to guide robots in making decisions, we can make them smarter and overcome challenges in how they work.

RESULTS AND DISCUSSION

Example: Pancake Flipping Task

In the quest to make robots better using reinforcement learning, we look at tasks like flipping pancakes. These tasks help us understand important problems robots face, like how different things are connected, how much information they can handle, and how smoothly they can do tasks.

Task Overview: In the pancake flipping task, you throw a pancake up, spin it around, and catch it with a frying pan. Because it's tricky to learn this task just by trial and error, we start by watching someone do it. This helps us get started with teaching robots how to do it using reinforcement learning.

Figure 6: Pancake flipping Robotic Arm [xiv]

Experimental Setup: In the experiment, we use a special robot arm with seven joints that can control the force used to flip a pancake. At the end of the arm, there's a real frying pan. We also use fake pancakes with shiny markers on them to see how well the robot does. We can track where the pancakes are and how they're positioned in real-time.

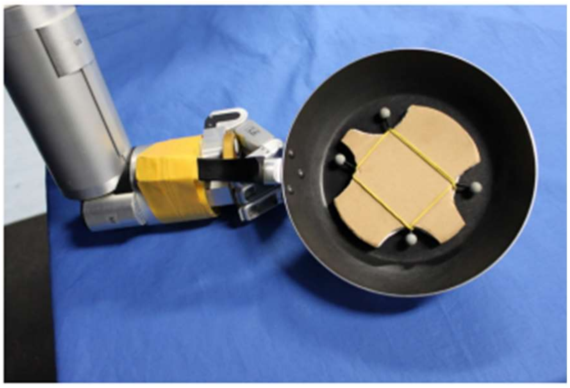

Figure 7:Setup of the Robotic Arm [xv]

Model-Free Reinforcement Learning: This approach directly learns the optimal policy for flipping pancakes through trial and error, without explicitly modelling the dynamics of the environment. Given the complexity of the task and the difficulty in accurately modelling pancake flipping dynamics, a model-free approach is often preferred.

Algorithms Used:

Q-Learning: Q-learning is a model-free RL algorithm that learns the action-value function (Q-function) iteratively. In the context of pancake flipping, Q-learning involves maintaining a Q-table or Q-function that estimates the expected return for each possible action (flip) in each state (pancake stack configuration). The agent selects actions based on the learned Q-values and updates them using the Bellman equation. Q-learning is suitable for discrete action spaces like pancake flipping. **Deep Q-Networks (DQN):** DQN extends Q-learning to handle high-dimensional state spaces by approximating the Q-function using deep neural networks. In the pancake flipping task, DQN can learn directly from raw state representations (e.g., pancake sizes) without the need for handcrafted features. Experience replay and target network updates are used to stabilize learning and improve sample efficiency. **Policy Gradient Methods**: Policy gradient methods directly parameterize the policy (the strategy for selecting actions) and optimize it to maximize expected rewards. Algorithms like REINFORCE, Trust Region Policy Optimization (TRPO), and Proximal Policy Optimization (PPO) can be applied to learn a policy for pancake flipping. The policy outputs the probabilities of selecting each possible flip action, and policy parameters are updated through gradient ascent on expected returns. **Experimental Results:** It took about 60 tries to figure out the right way to flip the pancakes without dropping them. The picture shows one try where the pancake flipped perfectly and landed in the frying pan. In a video, you can see the pancake flipping over completely. Towards the end, you might notice the pan moving up and down a bit when the pancake lands. This happens because the robot is adjusting to catch the pancake properly. This bouncing motion was actually discovered by the robot learning algorithm to help catch the pancake better. Without it, the pancakes might bounce off the pan and fall. This shows how important these learning algorithms are for making robots that can adapt and be flexible.

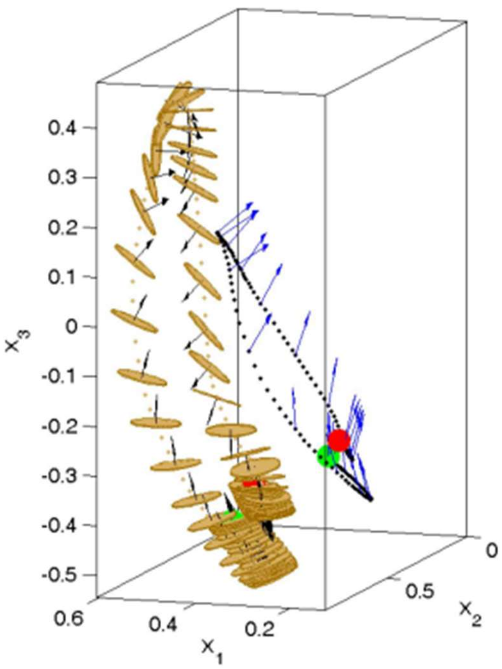

Figure 8: Execution of the Pancake Flipping [xvi]

This is a picture of a real experiment where a robot flipped a pancake. The pancake, shown in yellow, was thrown, caught in a frying pan, and spun 180 degrees. The path of the robot's hand is marked by a black dot, and the direction it's moving is shown by a blue arrow. Another black arrow shows the direction perpendicular to the pancake.

Significance of Approach: The suggested way to set up robot guidelines by combining different basic forces has many benefits. It helps robots learn how different movements are connected, makes the guidelines smaller, and ensures that robots can do tricky tasks safely.

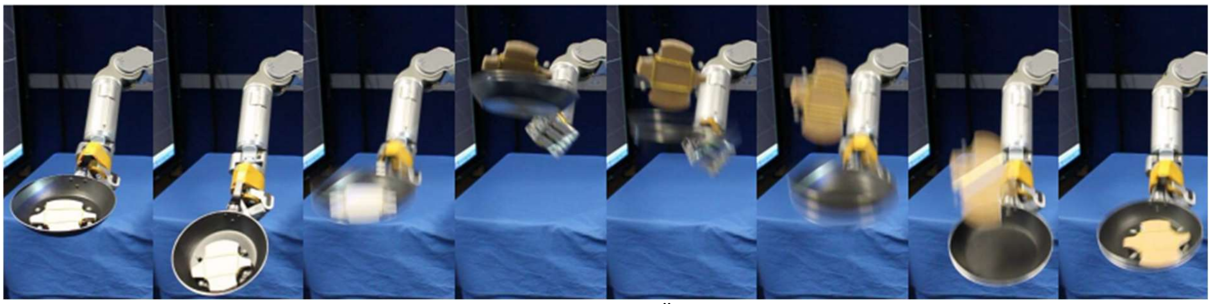

Figure 9: [xvii]

In short, the pancake flipping example shows how using creative ways to set up guidelines can help solve important problems and make it easier to use reinforcement learning in real-world robot tasks.

Another Example: Archery-Based Aiming Task:

This example primarily tackles the challenges of multi-dimensionality and convergence speed as outlined. It demonstrates that combining reinforcement learning (RL) with regression leads to a highly efficient algorithm, exemplified through the rapid acquisition of archery skills by the humanoid robot, iCub.[xviii] The objective is to devise an integrated methodology for enabling the iCub humanoid robot to master the art of archery. Following instructions on bow handling and arrow release, the robot autonomously learns to shoot the arrow precisely to hit the target's center. Two learning algorithms are introduced and compared for acquiring this bi-manual skill: one based on Expectation-Maximization reinforcement learning and the other employing chained vector regression, known as the Augmented Reward Chained Regression (ARCHER) algorithm. Both algorithms govern the coordination of the two hands' motion, while an inverse kinematics controller manages the arm movements. This approach is assessed using the upper body of the iCub robot, involving 38 Degrees of Freedom (DoF). The image processing component detects where the arrow strikes the target, relying on Gaussian Mixture Models for color-based target and arrow tip detection. Further implementation details can be found.[xix]

Description of the Archery Task: The archery task poses several challenges, including bi-manual coordination, precise movement with minimal force, tool manipulation, and integration of various components into a unified task. The primary focus is on acquiring the bi-manual coordination required to control arrow direction and velocity accurately.

Algorithms Used:

Q-Learning: Q-Learning is a model-free RL algorithm that aims to learn the optimal action-value function Q ∗ (s,a), where s is the state and a is the action. In the archery-based aiming task, the state could represent the current position of the target, and the action could represent the angle and force at which the arrow is shot. By updating Q-values based on the rewards received, the agent learns the best actions to take in different states to maximize its cumulative reward.; **Deep Q-Networks (DQN):** DQN is an extension of Q-Learning that uses a neural network to approximate the Q-function. This allows for handling high-dimensional state spaces, such as images of the archery target. In this setup, the neural network takes the image of the target as input and outputs Q-values for different actions. By training the network to minimize the temporal difference error between predicted and target Q-values, the agent learns to aim accurately.**Policy Gradient Methods**: Unlike value-based methods like Q-Learning, policy gradient methods directly learn the policy function π(a|s), which specifies the probability of taking each action in a given state. Algorithms like REINFORCE or Proximal Policy Optimization (PPO) can be used. In the archery task, the policy network takes the target state as input and outputs a probability distribution over actions (angles and force). By adjusting the policy parameters to maximize the expected cumulative reward, the agent learns to shoot accurately.

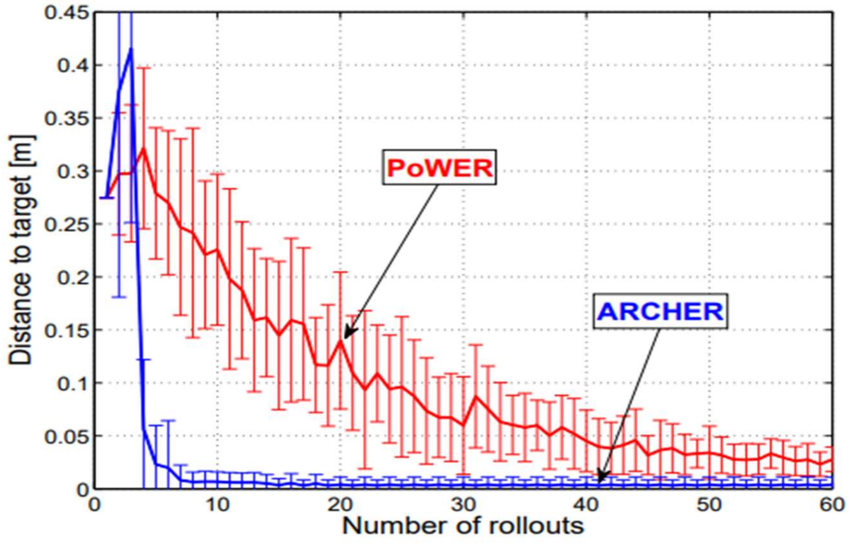

Figure 10: Comparison of the speed of convergence for the POWER and ARCHER algorithms [xx]

Comparison of the speed of convergence for the POWER and ARCHER algorithms. Statistics are collected from 40 learning sessions with 60 rollouts in each session. The first three rollouts of ARCHER are done with large random exploratory noise, which explains the big variance at the beginning.

Experimental Results on the iCub Robot: Real-world experiments with the iCub robot confirm the effectiveness of the ARCHER algorithm, achieving rapid convergence with fewer than 10 rollouts to hit the target centre. The local regression approach of ARCHER outperforms traditional RL algorithms, highlighting significant improvements in convergence speed facilitated by multi-dimensional rewards and prior knowledge.[xxi]

Figure 11: Experimental setup for the archery task [xxii]

Comparison of the three presented RL problems (rows 1–3) with three other robot skill learning problems (rows 4–6). DMP, Dynamic Movement Primitives; CoM, center of mass.

No.	Task	Dimensionality	Task dynamics	Noise and uncertainty	Best performing approach
1	**Pancake flipping**	High (positions and stiffness for each attractor)	Very high	High	RL + DMP
2	**Walking optimization**	Medium (continuous trajectory of CoM)	High	High	RL + evolving policy
3	**Archery**	Low (relative pose of hands)	Low (static shooting pose)	Medium	Regression
4	**Ironing** [3]	Medium (pose of end-effector)	Low (slow movement)	Low	Imitation learning + DMP
5	**Whiteboard cleaning** [6]	Medium (pose and force at end-effector)	Low (slow movement, medium forces)	Medium	
6	**Door opening** [3]	Medium (pose and force at end-effector)	Low (slow movement, high forces)	Medium	

Figure 12: Comparison of the three presented RL problems (rows 1–3) with three other robot skill learning problems (rows 4–6). DMP, Dynamic Movement Primitives; CoM, center of mass.

Real-Life Impact: Real-World Applications of Reinforcement Learning: Addressing Challenges and Transforming Robotics**.** The practical application of reinforcement learning (RL) in real-world scenarios has the potential to revolutionize robotics by overcoming various challenges. In this context, we explore the impact of RL in enhancing robotic capabilities and its implications for addressing key obstacles faced in the field.[xxiii]
Challenges Addressed: Combining reinforcement learning with regression techniques helps robots learn complicated tasks more quickly. This means they can pick up new skills faster and become better at challenging tasks.

CONCLUSION

In summary, bringing reinforcement learning (RL) into robotics has big potential for changing how robots work in real life and tackling big problems. By looking at different learning tasks and what's ahead for RL in robotics, we've learned some important things. For example, combining RL with other methods like regression helps robots learn tough skills like archery faster. The ARCHER algorithm is one example that shows this works well, using feedback and previous knowledge to teach robots better. Looking ahead, we might see robots that can set their own goals and learn by themselves, maybe even understanding language. This could make learning more natural and efficient for robots. Overall, to make RL work in real life, we need to deal with challenges like handling lots of information and learning quickly. But with RL, robots can become more independent, adaptable, and better at doing their jobs in many different areas.

REFERENCES

[i] Petar Kormushev, Sylvain Calinon, and Darwin Caldwell, “Reinforcement Learning in Robotics: Applications and Real-World Challenges,” *Robotics* 2 (September 1, 2013): 122–48, https://doi.org/10.3390/robotics2030122.
[ii] Keerthana Sivamayil et al., “A Systematic Study on Reinforcement Learning Based Applications,” *Energies* 16, no. 3 (January 2023): 1512, https://doi.org/10.3390/en16031512.
[iii] Julian Ibarz et al., “How to Train Your Robot with Deep Reinforcement Learning: Lessons We Have Learned,” *The International Journal of Robotics Research* 40, no. 4–5 (April 1, 2021): 698–721, https://doi.org/10.1177/0278364920987859.
[iv] Emily, “Unleashing the Power of Reinforcement Learning in AI,” Aaron Hall, October 31, 2023, https://aaronhall.com/insights/unleashing-the-power-of-reinforcement-learning-in-ai/.
[v] Sivamayil et al., “A Systematic Study on Reinforcement Learning Based Applications.”
[vi] Ibarz et al., “How to Train Your Robot with Deep Reinforcement Learning.”
[vii] Ibarz et al.
[viii] Lorenzo Moro et al., “Quantum Compiling by Deep Reinforcement Learning,” *Communications Physics* 4 (August 6, 2021), https://doi.org/10.1038/s42005-021-00684-3.
[ix] “Debugging Deep Model-Based Reinforcement Learning Systems,” accessed April 12, 2024, https://www.natolambert.com/writing/debugging-mbrl.
[x] “Debugging Deep Model-Based Reinforcement Learning Systems.”
[xi] “The Promise of Hierarchical Reinforcement Learning,” accessed April 12, 2024, https://thegradient.pub/the-promise-of-hierarchical-reinforcement-learning/.
[xii] Yugesh Verma, “An Introductory Guide to Meta Reinforcement Learning (Meta-RL),” Analytics India Magazine, November 17, 2021, https://analyticsindiamag.com/an-introductory-guide-to-meta-reinforcement-learning-meta-rl/.
[xiii] Zihan Ding and Hao Dong, “Challenges of Reinforcement Learning,” in *Deep Reinforcement Learning: Fundamentals, Research and Applications*, ed. Hao Dong, Zihan Ding, and Shanghang Zhang (Singapore: Springer, 2020), 249–72, https://doi.org/10.1007/978-981-15-4095-0_7.
[xiv] Khruschev, Calinon, and Caldwell, “Reinforcement Learning in Robotics.”
[xv] Kormushev, Calinon, and Caldwell.
[xvi] Kormushev, Calinon, and Caldwell.
[xvii] Kormushev, Calinon, and Caldwell.
[xviii] Ibarz et al., “How to Train Your Robot with Deep Reinforcement Learning.”
[xix] “Robot Archer Learns How to Aim and Fire,” accessed April 4, 2024, https://www.zmescience.com/research/inventions/robot-archer-learns-how-to-aim-and-fire-28092010/.
[xx] Kormushev, Calinon, and Caldwell, “Reinforcement Learning in Robotics.”
[xxi] Tengteng Zhang and Hongwei Mo, “Reinforcement Learning for Robot Research: A Comprehensive Review and Open Issues,” *International Journal of Advanced Robotic Systems* 18, no. 3 (May 1, 2021): 17298814211007305, https://doi.org/10.1177/17298814211007305.
[xxii] Kormushev, Calinon, and Caldwell, “Reinforcement Learning in Robotics.”
[xxiii] Ding and Dong, “Challenges of Reinforcement Learning.”