IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 11, Issue 11, November 2024

DOI: 10.17148/IARJSET.2024.111119

Survey On Designing ML Agents

Pranjal Kumar[1], Rishav Anand[2], Sahil Raj Shrivastav[3], Tejas V[4]

Department of Artificial Intelligence and Machine Learning, Dayananda Sagar Academy of Technology and Management, Bangalore, Karnataka[1-4]

Abstract: Artificial intelligence offers valuable methods for crafting complex problem-solving scenarios, with recent advancements allowing the development of agents capable of human-level or even superhuman performance. Reinforcement learning (RL), particularly through tools like the Unity ML-Agents toolkit, enables developers to incorporate machine learning-driven behaviors into game environments without needing specialized expertise. This paper reviews and compares various reinforcement learning techniques, detailing their application across two distinct training environments. We assess these methods in terms of training pace, generalization capabilities, and cumulative reward accumulation, with a focus on evaluating how combined extrinsic and intrinsic rewards influence training effectiveness in sparse reward settings. Our findings aim to support developers in selecting optimal reinforcement strategies to save time during training while enhancing performance and robustness. Results indicate that agents trained in sparse environments achieved faster progress with a mix of extrinsic and intrinsic rewards, while agents relying solely on extrinsic rewards struggled to complete tasks and exhibited suboptimal learning behaviors. Additionally, we discuss the role of exploration-exploitation trade-offs, curriculum learning, and reward shaping in improving agent performance.

Index Terms: Unity, ML-Agents, Reinforcement Learning, Sparse Reward Environment, Artificial Intelligence, Machine Learning, Intrinsic Rewards, Extrinsic Rewards, Agent Training, Exploration-Exploitation, Curriculum Learning, Reward Shaping, Game Development, Autonomous Agents, Performance Evaluation, Generalization, Behavior Modeling, Policy Optimization.

I. INTRODUCTION

Reinforcement Learning (RL) has become a crucial approach for training agents to handle challenging, sequential decision-making tasks. This technique has demonstrated exceptional success in complex strategic games, such as StarCraft and DoTA2, where deep RL agents have achieved and even surpassed expert human performance. Training deep RL agents requires substantial data and computational power, as these agents rely on neural networks and vast training experiences to improve performance. For instance, advanced agents like MuZero and Agent-57 necessitate decades' worth of simulated gameplay to master environments like Atari games, and OpenAI Five required over 45,000 years of simulated experience to excel in its domain.

While these data demands are feasible in video game simulations, real-world applications often involve higher costs and require more efficient use of data. As a result, improving data efficiency has become a vital goal for RL in real-world settings. Self-supervised learning has made strides in data efficiency, especially in vision and language tasks where data scarcity or limited labelling is an issue. These self-supervised approaches leverage intrinsic structures within data, like image patches or temporal proximities, to generate additional training signals. This allows for significant gains in learning efficiency and performance, especially when resources for labeled data are limited.

Inspired by these advances, we employ enhanced representation learning for RL by developing models that predict future representations and maintain consistency across augmented data views. Specifically, we extend a model-free agent with a dynamic model that forecasts future representations based on temporal consistency and data augmentation, focusing on representations rather than raw data reconstruction. We evaluate this Self Predictive Representation (SPR) approach on the Atari 100k benchmark, where agents interact with the environment for a limited number of steps, providing a data-efficient comparison for evaluating these advancements.

II. STUDY ON DIFFERENT ML AGENTS

The Unity ML-Agents toolkit is an open-source plugin for the Unity game engine that enables developers to integrate reinforcement learning for training game agents. It uses PyTorch, a deep learning library, to train AI on CPUs and GPUs, allowing neural network-driven behaviors to control game agents within the Unity environment. The toolkit consists of three primary components: the agent, the environment, and the available actions the agent can take.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 11, Issue 11, November 2024

DOI: 10.17148/IARJSET.2024.111119

Through the Python API, agents share experiences during training, accessing player inputs, scripts, and neural networks to update their policies. A "Communicator" connects the agents to the trainers, facilitating the exchange of training data and behavioral adjustments.

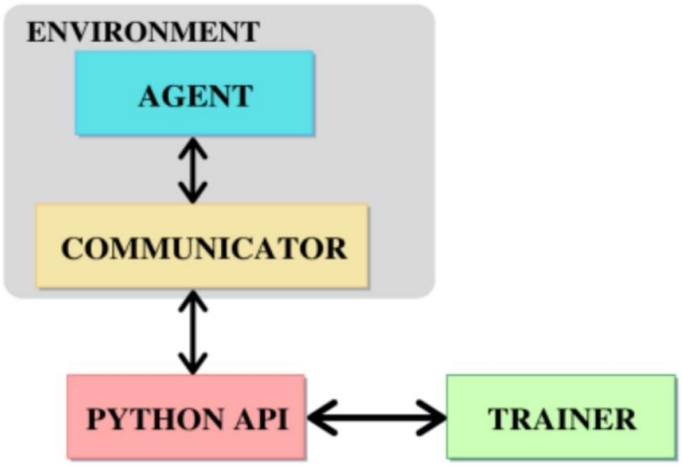

Figure 1: Flowchart of ML Agents

Supervised Learning Agent :

Supervised learning involves training an agent to map input data to specific outputs using labeled data pairs. During training, data is divided into sets for training and testing, allowing the model to learn from the training data and validate performance on test data. Supervised learning requires external labels to guide the agent, making it suitable for situations where clear input-output relationships exist. This approach has broad applications in machine learning, especially in cases where extracting precise information from large datasets is necessary.

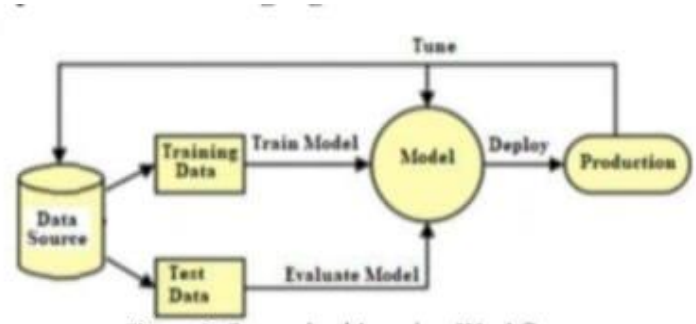

Figure 2: Supervised learning Workflow

Semi Supervise Learning agent:

Semi-supervised learning combines supervised and unsupervised learning techniques. It is beneficial in contexts where unlabelled data is abundant, but labeling is labor-intensive or costly. By training on both labeled and unlabeled data, semi-supervised learning enhances learning efficiency, making it useful for cases where limited labeled data is supplemented by large amounts of unlabeled data.

Unsupervised Learning agent:

Unlike supervised learning, unsupervised learning does not require labeled data. Instead, it allows agents to independently discover patterns or structures within the data, learning features that aid in classification or clustering tasks. This approach is valuable in exploratory data analysis, where the goal is to identify inherent groupings or reduce dimensionality without prior knowledge of categories.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 11, Issue 11, November 2024

DOI: 10.17148/IARJSET.2024.111119

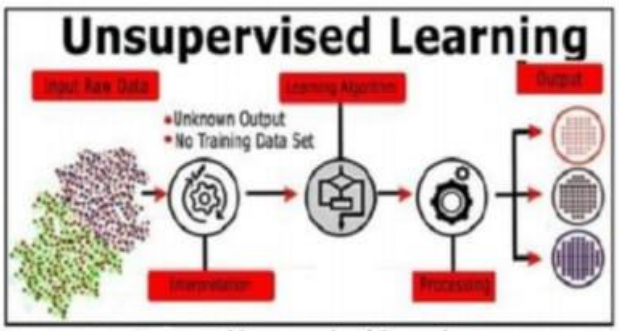

Figure 3: Unsupervised Learning

Multitask Learning Agent:
Multi-task learning aims to improve learning efficiency by training agents to perform several related tasks simultaneously. By leveraging similarities across tasks, multi-task learning encourages the agent to generalize knowledge across various domains, reducing the need for separate models. This technique is especially useful for applications where tasks are interrelated but not identical, allowing shared knowledge to improve performance.

III. MACHINE LEARNING APPLICATIONS

Table 1: Overview of Research Papers.

Ref no.	Research Paper	Authors/year	Methodology	Remarks
[1]	Unsupervised state representation learning in Atari	Ankesh Anand (2020)	This paper focuses on unsupervised learning techniques to extract state representations from Atari games. The approach leverages deep learning models to learn meaningful state representations without the need for explicit labels or supervision.	The research is significant as it highlights the potential of unsupervised learning in complex environments like Atari games, contributing to advancements in representation learning.
[2]	Learning representations by maximizing mutual information across views.	William Buchwalter. (2021)	The authors propose a method for learning representations by maximizing the mutual information between different views of the same data. This is achieved through a deep neural network that learns to represent data in a way that captures the shared information between views.	This work is important for its novel approach to representation learning, which can be applied to various domains such as computer vision and natural language processing.
[3]	Grandmaster level in StarCraft II using multi-agent learning.	Vinyals,Silver, D. (2020).	The research describes the use of multi-agent reinforcement learning to achieve grandmaster-level performance in the real-time strategy game StarCraft II. The approach involves training multiple agents to cooperate and compete in the game environment.	This paper demonstrates the potential of multi-agent systems in complex, dynamic environments, showcasing significant progress in AI capabilities.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 11, Issue 11, November 2024

DOI: 10.17148/IARJSET.2024.111119

[4]	Chess and Shogi by planning with a learned model.	Schrittwieser, J (2020)	: This paper discusses the various challenges faced when applying reinforcement learning (RL) to real-world problems. It covers issues such as sample efficiency, safety, and the transfer of RL algorithms from simulation to real-world environments.	The insights provided are crucial for advancing the application of RL in practical, real-world scenarios, guiding future research to address these challenges.
[5]	Challenges of real-world reinforcement learning..	Dulac-ArnoldHester, T. (2020)	This paper discusses the various challenges faced when applying reinforcement learning (RL) to real-world problems. It covers issues such as sample efficiency, safety, and the transfer of RL algorithms from simulation to real-world environments.	The insights provided are crucial for advancing the application of RL in practical, real-world scenarios, guiding future research to address these challenges.
[6]	A simple framework for contrastive learning of visual representations.	Hinton, G. (2020)	The authors propose a contrastive learning framework for visual representation learning. This method trains a model to distinguish between similar and dissimilar image pairs, improving the quality of learned representations.	The framework is praised for its simplicity and effectiveness, contributing significantly to the field of self-supervised learning.
[7]	Momentum contrast for unsupervised visual representation learning.	He, K., & Girshick, R. (2020)	his paper introduces Momentum Contrast (MoCo), an unsupervised learning method for visual representations. MoCo builds a dynamic dictionary with a queue and a moving-averaged encoder, which helps in contrasting different image representations effectively.	In unsupervised learning method for visual representations. MoCo builds a dynamic dictionary with a queue and a moving-averaged encoder, which helps in contrasting different image representations effectively.
[8]	A general platform for intelligent agents.	Mattar, M, Lange, D. (2021)	The paper presents the Unity platform, designed for developing and training intelligent agents. Unity provides a flexible and comprehensive environment for simulating complex tasks and testing various AI algorithms.	Unity has become a widely used tool in AI research and development, supporting advancements in areas such as reinforcement learning and robotics.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 11, Issue 11, November 2024

DOI: 10.17148/IARJSET.2024.111119

IV. STRUCTURE OF MACHINE LEARNING FRAMEWORK

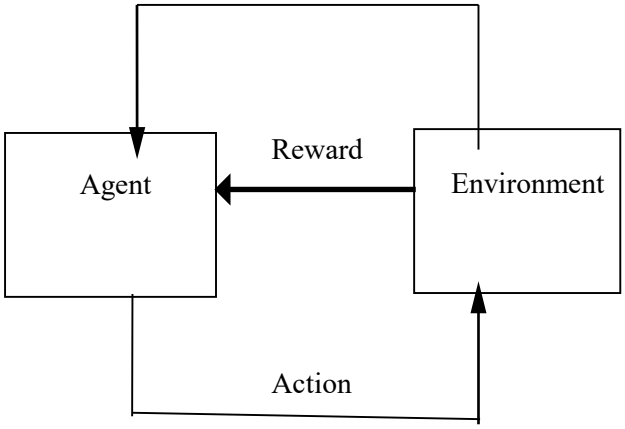

Figure 4: Structure of Machine Learning

The basic framework of Machine Learning is illustrated in a model where an agent interacts with its environment through sensor data, modifies the environment, and earns rewards for its actions. The environment is represented by states, which are features or parameters describing it. Sensor information determines these states. From a specific state SSS, a value function evaluates the possible actions an agent can take, predicting an associated reward. In Reinforcement Learning, an agent perceives the environment and learns the optimal strategy by taking actions in various conditions. This approach represents a simple behavior and learning algorithm where the agent iteratively optimizes its new states. The agent determines its current state (s∈Ss \in Ss∈S), selects an action (a∈Aa \in Aa∈A), potentially transitions to a new state, and receives a reward signal (r∈Rr \in Rr∈R). Through this process, the agent gathers useful experience regarding states, actions, transitions, and rewards to act more effectively, with the system's evaluation occurring simultaneously with the learning process. The primary goal of Reinforcement Learning is to learn how to associate states with actions that maximize cumulative rewards over time.

Reinforcement Learning often utilizes a framework modeled as a Markov Decision Process (MDP), which serves as the mathematical foundation for single-agent reinforcement learning. MDPs address sequential decision-making problems, where actions must be chosen at each state to navigate the system effectively. Such problems are prevalent in stochastic control theory and have deep mathematical roots. An MDP is typically defined by a tuple (S,A,P,R)(S, A, P, R)(S,A,P,R), where SSS represents the set of states, A(s)A(s)A(s) is the set of available actions for each state, P:S×A×S→[0,1]P: S \times A \times S \to [0, 1]P:S×A×S→[0,1] is the state transition probability function defining the likelihood of moving to a specific state after taking an action in a given state, and R:S×A→rR: S \times A \to r R:S×A→r specifies the immediate reward for performing an action in a particular state. The objective is to identify a policy that maximizes the total accumulated rewards over time, expressed as r0+γr1+γ2r2+…r_0 + \gamma r_1 + \gamma^2 r_2 + \ldotsr0+γr1 +γ2r2+…, where γ∈(0,1)\gamma \in (0, 1)γ∈(0,1) is the discount factor. This factor determines the relative importance of immediate versus future rewards, ensuring the cumulative return RtR_tRt remains finite when γ<1\gamma < 1γ<1.

Automated Robots:

Although most robots don't match the depictions popularized by pop culture, their capabilities are no less remarkable. With Reinforcement Learning (RL), robots improve their accuracy and efficiency over time, enabling them to perform tasks that were once time-consuming with increased speed and precision. They can also undertake hazardous tasks with significantly reduced risks compared to humans. For these reasons, robots, apart from needing occasional supervision and routine maintenance, have become a cost-effective and efficient alternative to manual labor. For instance, some restaurants deploy robots to serve food to tables, while grocery stores utilize them to monitor stock levels and initiate restocking orders. In various settings, robots have been widely used for assembling products, detecting defects, managing inventory, delivering goods, traveling across distances, organizing and reporting data, and handling objects of varying shapes and sizes. As robotic capabilities continue to evolve through testing and development, new features are being introduced, broadening their range of potential applications.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 11, Issue 11, November 2024

DOI: 10.17148/IARJSET.2024.111119

V. CHALLENGES IN EXISTING SYSTEM

The combination of deep learning (DL) and reinforcement learning (RL) has enabled advancements in fields such as autonomous driving, robotics, machine translation, and gaming. However, despite these achievements, current deep reinforcement learning (DRL) systems face significant challenges that limit their broader application. Key challenges include:

5.1 Opacity or Non-interpretability:

Modern neural networks often function as "black boxes," making it difficult to understand or interpret the processes within each layer. This lack of transparency creates limitations, as the networks do not resemble human cognitive processing or reasoning and lack intuitively understandable decision-making steps. Consequently, it remains challenging to decode how specific activation methods and layers contribute to final outputs, which affects model interpretability

5.2 Data Inefficiency:

Training neural networks to perform at optimal levels typically requires extensive data, which increases model complexity and demands high computational power. This is problematic for applications with high sample complexity; for instance, a DRL agent might require millions of frames in a video game setting to reach peak performance, whereas a human can often understand the game within a fraction of those frames.

5.3 Low Sample Efficiency:

Sample efficiency pertains to the quantity of experience needed for an algorithm or agent to learn effectively. Efficient algorithms should ideally learn quickly and make optimal use of experiences to improve policy and performance. However, many DRL methods struggle with sample efficiency, potentially missing valuable learning opportunities, which can slow down model performance improvements.

5.4 Issues of Reproducibility:

Due to the complexity and vast data requirements in neural networks, reproducing DRL models can be challenging even for their original developers. As networks grow more intricate, the ability to replicate or verify their performance becomes increasingly difficult. To address this, researchers are exploring approaches like minimal trace modeling, experiment tracking, logging, and using metadata platforms, along with novel architectures like Neurosymbolic AI, to enhance reproducibility and application-specific customization.

5.5 Implementation in Real Life Scenarios:

RL agents generally train by exploring simulated environments, which limits their ability to adapt to real-world situations without extensive retraining. In real-world scenarios, agents often rely on pre-trained data instead of live exploration, which may create a “reality gap” between the training simulation and actual deployment. Solutions to mitigate this issue include behavior imitation, using verified simulations, and algorithm optimization.

5.6 Hyperparameter configuration:

Configuring hyperparameters is critical for effective RL model training. In this study, we used a configuration file adapted from Unity’s ML-Agents toolkit, modifying parameters through trial and error to optimize training performance. Once an ideal setup was determined, the same parameters were applied consistently across all training methods, except for unique settings required by methods like GAIL, curiosity-driven learning, and behavior cloning.

VI. CONCLUSION

To delve deeper into the impact of our findings, we analyzed how different reward structures affect the efficiency of agent learning across tasks of varying complexity. Sparse reward environments pose significant challenges for reinforcement learning agents due to the minimal feedback available, often resulting in slower learning or difficulty in finding optimal strategies. However, when intrinsic reward mechanisms, such as curiosity-driven exploration, were introduced, agents showed improved adaptability and robustness. This facilitated better navigation through sparse environments, emphasizing the importance of enhancing sparse reward systems with complementary incentive structures to aid agent training.

In environments with dense rewards, the frequent feedback provided a clear and direct path for agents to optimize their actions. Proximal Policy Optimization (PPO), a commonly used baseline algorithm, performed effectively due to its balance between stability and efficiency, allowing agents to make consistent progress without being overly influenced by short-term variations in rewards.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 11, Issue 11, November 2024

DOI: 10.17148/IARJSET.2024.111119

Additionally, the Random Network Distillation (RND) method enhanced performance by fostering exploratory behaviors, which enabled agents to discover high-value strategies even in scenarios with abundant rewards. The combination of RND and PPO showcased the advantage of integrating exploration-focused methods with robust policy optimization algorithms.Another valuable insight from this study is the benefit of hybrid approaches in improving agent performance. By incorporating additional methods, such as intrinsic motivation or auxiliary learning objectives, alongside standard reinforcement learning techniques, we observed notable advancements in sparse reward settings. This finding highlights the value of customizing algorithms to align with the reward structure of a given environment, leading to more efficient learning and better generalization to novel situations,These results offer practical guidance for game designers and developers aiming to integrate AI-driven behaviors into their projects.

Choosing the right reinforcement learning approach based on the environment's reward dynamics can significantly reduce the time and resources required for training. Furthermore, understanding the strengths and limitations of different reinforcement learning strategies can help designers create environments that enhance agent performance, whether by adjusting reward systems or employing hybrid training methodologies.

REFERENCES

[1]. *Ankesh Anand, Evan Racah, Sherjil Ozair, Yoshua Bengio*, Marc-Alexandre Côté, and R Devon Hjelm. Unsupervised state representation learning in atari. In NeurIPS,(2020).
[2]. *Philip Bachman, R Devon Hjelm, and William Buchwalter*. Learning representations by maximizing mutual information across views. In NeurIPS,(2021).
[3]. *Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., ... & Silver, D.* (2020). Grandmaster level in StarCraft II using multi-agent learning.
[4]. *Schrittwieser , J, Antonoglou, Simonyan, K., Sifre, L., Schmitt & Silver, D*. (2020) , go, chess and shogi by planning with a learned model.
[5]. *Dulac-Arnold, G., Mankowitz, D., & Hester, T.* (2020). Challenges of real-world reinforcement learning.
[6]. *Chen, T., Kornblith, S., Norouzi, M., & Hinton, G.* (2020). A simple framework for contrastive learning of visual representations.
[7]. *He & Girshick, R.* (2020). Momentum contrast for unsupervised visual representation learning.
[8]. *Juliani, A., Berges, V. P., Vckay, E., Gao, Y., Henry, H., Mattar, M., & Lange, D*. (2021). Unity: A general platform for intelligent agents.
[9]. *Chen, T., Kornblith, S., Swersky, K., Norouzi, M., & Hinton, G*. (2020). Big self-supervised models are strong semi-supervised learners.
[10]. OpenAI, *Berner, C., Brockman, G., Chan, B., Cheung, V., Debiak, P., ... & Olsson, C*. (2021). Dota 2 with large scale deep reinforcement learning
[11]. *François-Lavet, V., Henderson, P., Islam, R., Bellemare, M. G., & Pineau, J.* (2021). An introduction to deep learning. Foundations and Trends in Machine Learning
[12]. *Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg*. (2022). Analysis of sentence embedding models using prediction tasks in natural language processing.
[13]. *Sutton, R. S., & Barto, A. G.* (2022). Reinforcement learning: An introduction (2nd ed.). MIT Press.
[14]. *Hafner, D., Lillicrap, T., Norouzi, M., & Ba, J.* (2020). Dream to control: Learning behaviors by latent imagination. International Conference on Learning Representations (ICLR).
[15]. *Badia, A. P., Piot, B., Kapturowski, S., Sprechmann, P., Vitvitskyi, A., Guo, Z. D., & Blundell, C.* (2020). Agent57: Outperforming the Atari human benchmark.

