

Research on the integrated application of machine learning in Unity

Chuang Hu

South-Central Minzu University College of Computer Science, Wuhan, 430079, China

hccreate11@gmail.com

Abstract. In recent years, the introduction of Machine Learning Agents has enabled more individuals to apply machine learning within the renowned Unity game engine. As application development has become increasingly complex, traditional artificial intelligence (AI) struggles to keep pace with the growing demands of modern applications. However, machine learning offers new avenues for developing smarter, more dynamic game characters. This paper aims to provide a comprehensive review of the various applications of machine learning within the Unity ecosystem, including game AI, intelligent character behavior, and reinforcement learning. Through a literature review and case studies, this paper explores the specific applications of machine learning algorithms in the Unity engine and analyzes the current state of development in different fields. The research findings indicate that the application of machine learning in Unity exhibits numerous developmental trends and leaves ample room for further application development in its strong areas. This technology can be innovatively applied across many fields, from implementing advanced NPCs and third-person game agents to anti-cheat mechanisms, from image recognition-based health systems to the design of complex constrained environments. The integrated application model of machine learning and Unity3D will bring more innovation and possibilities to future application development.

Keywords: Unity Machine Learning Agents, Machine learning, Unity, Artificial Intelligence.

1. Introduction

Unity is a real-time 3D development platform widely used in game development, virtual reality (VR), and augmented reality (AR), and its robust rendering engine and cross-platform capabilities have cemented its position as a preferred tool among developers. In modern technology, machine learning has become a pivotal component, pervading numerous domains including image recognition, speech processing, autonomous driving, and personalized recommendation systems. The integration of Unity with machine learning has the potential to significantly enhance the user experience of a diverse array of Unity applications [1]. In this paper, the existing literature on the integration of machine learning with the Unity platform is systematically reviewed to summarize the main research advances and application scenarios. In addition, technical challenges and potential solutions for applying machine learning techniques in the Unity environment are analyzed, and relevant cases of successful integration of machine learning into Unity-based applications are explored to highlight the benefits and insights gained. Future developments should concentrate on the advancement of emerging technologies, the exploration of potential use cases, and the examination of implications for broader domains such as

interactive media, immersive experiences, and intelligent systems. Therefore, the paper aims to present a comprehensive overview of the current status and future direction of the integration of machine learning with the Unity platform, which will provide some insights for related research.

2. Relevant Technical Basics

2.1. Concepts and Classification of Machine Learning

Machine learning models represent a specific form of artificial intelligence that enables computers to learn and enhance their capabilities through experience, without the need for explicit programming. These models employ statistical algorithms and mathematical models to analyze and identify patterns in data, subsequently utilizing these patterns to make predictions or decisions about new data. There are a wide variety of machine learning models, including supervised learning, unsupervised learning, and reinforcement learning [2]. Supervised learning involves training a model with labeled data, where the model learns to predict outcomes based on input features. Unsupervised learning involves training a model with unlabeled data, where the model learns to identify patterns or groupings in the data. Reinforcement learning involves training a model to make decisions based on feedback from its environment. Some common machine learning models include decision trees, neural networks, support vector machines (SVM), and random forests. Each of these models has its own strengths and weaknesses, and the choice of model depends on the specific problem being solved and the nature of the data. Machine learning models are widely used in a variety of fields, including finance, healthcare, marketing, and computer vision. They have proven to be effective in solving complex problems and making accurate predictions, and continue to evolve as new algorithms and techniques are developed.

2.2. Unity Machine Learning Agents

Unity Machine Learning Agents (ML-Agents) is an open-source Unity plugin designed to facilitate the training of intelligent agents within game environments and simulations. This tool supports a range of machine learning methods, including reinforcement learning, imitation learning, and neuroevolution, which can be utilized to control agents via a straightforward Python API. Moreover, the plugin offers implementations of state-of-the-art algorithms based on TensorFlow, enabling game developers and hobbyists to efficiently train intelligent agents for 2D, 3D, and VR/AR games. Trained agents serve various purposes, such as controlling NPC behaviors in diverse setups (including multiple agents and adversarial scenarios), automating in-game testing, and evaluating pre-release game design decisions. ML-Agents proves advantageous for both game developers and AI researchers, providing a unified platform to test the latest AI advancements within the rich and dynamic environment of Unity[1]. This integration enhances the accessibility of results to a broader community of researchers and developers.

3. Applications of Machine Learning to Unity

In recent years, research on the application of machine learning in Unity has been gradually increasing. Research has shown that integrating machine learning can significantly increase the intelligence of game AI, resulting in an improved user experience. Machine learning has also shown great potential in the areas of automated testing and content generation for Unity-based applications. Beyond game development, the AR (augmented reality) and VR (virtual reality) domains have also begun to leverage machine learning techniques to elevate the interactivity and level of immersion experienced by users. This emerging trend underscores the versatility and transformative impact of machine learning within the Unity ecosystem. Beyond game development, the AR (augmented reality) and VR (virtual reality) domains have also begun to leverage machine learning techniques to elevate the interactivity and level of immersion experienced by users. This emerging trend underscores the versatility and transformative impact of machine learning within the Unity ecosystem.

3.1. Challenges in Realizing Advanced NPCs in 3D Games

Implementing advanced Non-Player Characters (NPCs) in 3D games has been shown to be feasible, particularly in simple environments where they can effectively guide players. However, creating an NPC that can provide human-like game guidance in complex game environments presents significant challenges. Fröberg and Håkansson explored the potential of utilizing machine learning to build intelligent agents in Unity 3D, showing that in simple game environments, intelligent agents can successfully influence the player's path choices with a high success rate. However, it takes a lot of time and effort to achieve similar results in a complex environment with multiple agents [3]. Xu and Supli also noted that while data-driven approaches can enhance the intelligence of NPC behavior, achieving human-like intelligence for NPCs in complex digital media environments still faces numerous technical and design challenges [4]. Though the implementation of advanced NPCs has been proven feasible in simple scenarios, creating a human-like game guide in complex 3D game environments remains a formidable task, requiring the overcoming of numerous technical and design challenges.

3.2. Training Attempts in Simulated Scenarios

The application of machine learning in simulated training environments has shown significant promise, particularly for complex tasks and high-risk environments. The utilisation of simulated environments enables researchers to undertake the training of agents in the performance of a range of tasks in a safe and effective manner, without the concomitant costs and risks associated with real-world scenarios. For instance, Pontus Andersson successfully trained game agents to complete a series of complex movement tasks using the Unity ML-Agents toolkit[5]. This approach not only improved training efficiency but also enabled agents to perform impressively in diverse game environments without access to source code.

However, accurately simulating training environments remains an urgent issue. Although simulated environments provide a relatively safe and controlled training ground, they are often overly idealized and fail to fully reflect the complexity and uncertainty of the real world. Reza Mahmoudi and Armantas Ostreika noted in their research that while they successfully implemented reinforcement learning agents for obstacle avoidance in Unity, applying these agents in real-world environments still faced many challenges[6]. These challenges include dynamic environmental changes and unforeseen obstacles, which prevent agents from performing as expected.

Future research needs to focus on developing universal methods to realistically replicate any scenario in simulated environments. This involves not only higher precision modeling and simulation techniques but also the development of new algorithms to enable agents to adaptively learn in dynamic and complex environments. Dario Alilovic discussed in his paper the training of agents in platform games, proposing several training strategies applicable to different game environments[7]. While these strategies are effective in specific scenarios, further validation and improvement are required to address more diverse and complex real-world applications.

3.3. Training AI with Clear Rules: A Strength of Integrating ML and Unity

Training an AI to achieve high scores under basic rules is relatively straightforward, especially when the game's rules and objectives are clearly defined. By integrating Machine Learning with Unity, we can effectively create highly intelligent game agents. For instance, in their study, Lai et al. utilized the Unity ML-Agents platform to successfully train AI agents capable of executing complex tasks in a third-person shooter game through the Proximal Policy Optimization and Actor-Critic algorithms of Deep Reinforcement Learning. The agents demonstrated the capacity to identify adversaries in uncharted territory, pursue medical provisions in the event of injury, replenish ammunition, and neutralize multiple perilous targets. This research suggests that AI agents can undergo continuous improvement within the game environment via the ML-Agents platform, thereby demonstrating a notable enhancement in their intelligence [8].

However, training an AI to excel under basic rules might be detrimental to game players, as such AI agents could create an unfair competitive advantage, disrupting the fairness and enjoyment of the game. Therefore, applying this AI technology to cheat detection, rather than directly in the game, could be a more viable approach. Leveraging the efficient learning capabilities of AI agents, various cheating behaviors can be quickly identified and countered, thereby effectively maintaining the fairness of the game environment. Lukas et al. explored the application of AI in game anti-cheat mechanisms in their research, demonstrating the effectiveness of AI in cheat detection[9]. These examples illustrate that the combination of Machine Learning and Unity not only excels in training efficient AI agents but also shows immense potential in maintaining game fairness and detecting cheating behaviors.

3.4. Spotting Differences: The Best Application Mode for This Model

In the modern health sector, Unity3D stands out as one of the best platforms for developing virtual reality (VR) and augmented reality (AR) applications due to its powerful graphics processing capabilities and cross-platform compatibility. For example, Neng-Sheng Pai and colleagues integrated depth imaging and machine learning to develop a home fitness and rehabilitation support system using the Unity game engine [10]. This system utilized the Kinect v2 sensor to capture users' depth images and employed the AdaBoost algorithm for posture recognition, enabling a fitness rehabilitation game system that provided real-time feedback on the Unity platform. This method not only improved the accuracy of home rehabilitation training but also reduced the risk of injury due to incorrect movements [10]. Similarly, another study showcased Unity3D's superiority in developing complex interactive applications. Researchers developed a toothbrush monitoring system that used multimodal sensors and the Unity engine for 3D monitoring of brushing regions and force [11]. By capturing and analyzing users' brushing actions in real-time, this system could provide precise feedback, helping users improve their brushing habits and prevent oral diseases. The successful application of this system demonstrated Unity3D's powerful capability to handle complex data and provide real-time feedback, further proving its extensive application potential in health monitoring and education.

In short, Unity3D has become the preferred platform for developing VR and AR applications due to its powerful graphics processing capabilities and flexibility. In the health sector, machine learning's ability to "spot the difference" is particularly noteworthy, as it can meticulously analyze images and data to provide precise feedback and guidance. For instance, through depth image capture and machine learning algorithms, precise recognition and analysis of users' postures and movements can be achieved, providing valuable real-time feedback in fitness and rehabilitation training. This technology not only enhances training effectiveness and safety but also shows significant potential in other health monitoring and guidance applications. Through these examples, it is evident that spotting differences is indeed one of the best application modes for the model, especially in fields that require precise data analysis and real-time feedback.

3.5. Excelling in the Design Field

In layout design, machine learning technology has demonstrated its strong potential, especially when dealing with complex constraints. Taking the spatial layout of ski resort changing rooms as an example, Gao and colleagues' research indicated that by using Agent-Based Modelling (ABM) and machine learning algorithms within the Unity engine, the layout of changing rooms could be optimized considering psychological distance and crowd density, thereby enhancing visitor satisfaction and safety [12]. This method not only increased design efficiency but also provided data-driven decision support for designers.

In game design, the application of generative scenes further showcases the immense potential of machine learning. For example, Mariana's research utilized machine learning methods to generate procedural dungeons, which not only increased the game's playability and novelty but also significantly reduced the workload of designers [13]. By training models to recognize and generate complex dungeon structures, designers can create diverse scenes in games, fulfilling players' needs for

exploration and challenge. This generative design method is not only applicable to games but can also be extended to other design fields requiring high dynamic changes.

Furthermore, the comprehensive consideration of layout optimization through machine learning has been widely applied. Sun and Turkan's research pointed out that traditional design methods often rely on qualitative analysis, whereas the introduction of ABM and machine learning can better simulate and optimize architectural layouts, especially when dealing with complex human behavior patterns[14]. The application of this technology not only improved space utilization but also significantly enhanced user experience and satisfaction.

4. Existing Challenges and Future Developments

4.1. Present limitations

While the theoretical advantages of using machine learning in Unity environments are substantial, practical applications still require further optimization and improvement. One of the primary challenges is the prolonged training time. For instance, when training advanced NPCs in large and complex environments, sparse rewards make it difficult for agents to associate their actions with successful outcomes. Whether the training process can be sustained long enough to yield good results is another concern. Unity's team-created example environments of similar complexity require more than three times the training duration to achieve successful behaviors [3]. Moreover, most experiments presented in this paper are conducted in relatively simple environments, such as basic multiplayer games and tasks guiding players through levels [3]. However, real-world scenarios are typically more complex, demanding stronger generalization abilities and intricate behaviors. Therefore, the applicability of this method in broader and more complex game environments remains to be further validated and expanded.

Additionally, there are several limitations to this approach, such as platform constraints. Currently, the ML-Agents library is restricted to the Unity game engine and cannot be applied to other engines. This limitation significantly reduces the method's generalizability and hinders its application across different development environments. Future research needs to explore how to implement similar reinforcement learning agent training methods in various game engines to enhance their applicability. Some applications of the agents also face scalability challenges. For example, the anti-cheat discussed in this paper can only distinguish between human players and bots, failing to detect other types of cheating behaviors, such as the use of assistive scripts [9]. To effectively prevent cheating in practical applications, future research must expand the agents' detection capabilities to encompass a broader range of cheating behaviors.

4.2. Future Developments

The future holds great promise for the integration of ML with Unity. Whether in real-life scenarios or gaming applications, simulated environments are crucial future application areas. For instance, using video games as training grounds for experimental machine algorithms is a significant development[5]. However, whether for machine learning algorithms or real-life product testing, creating a simulated environment in a short time frame remains challenging. Future researchers might consider how to rapidly construct a simulated environment or even how to establish unified standards for simulation scenarios. In addition, cheating and anti-cheating have always been strengths of AI [8][9]. However, as game cheaters' methods become more and more sophisticated with technological advances, particularly in FPS games with a large following of cheat users, how anti-cheating technology can outpace cheaters and more comprehensively capture cheating behaviors and evidence requires deeper investigation by future scholars. In the health domain, exemplary systems have been developed that combine the Unity game engine with deep image recognition and machine learning for home health and rehabilitation support, such as detecting tooth brushing techniques [10][11]. Many people have incorrect fitness postures as well as habits in their lives, and it is possible to extend the scope of such

applications from the fitness domain to all aspects of life, studying how they sit, how they play basketball, how they eat, and so on.

5. Conclusion

This study systematically explores the applications of machine learning in Unity, demonstrating its vast potential in enhancing game intelligence, maintaining clean game environments, improving interactive experiences, and design. Through the discussion, summary, and comparison of various related papers, the research not only reveals their respective strengths and weaknesses but also provides valuable guidance for practical applications and indicates directions for future research. As machine learning becomes more prevalent across various aspects of human society, the demand for intelligent applications will continue to grow. Whether in daily life or video games, more practical validations are needed to ensure their effectiveness and robustness in future applications. Currently, most applications and research remain in the experimental stage due to high training costs and the complexity of real-world scenarios. Overall, this study lays the groundwork for the future development of ML combined with Unity3D application models, anticipating more innovations and breakthroughs in the future.

References

- [1] Juliani, A., Berges, V. P., et al. (2018). Unity: A general platform for intelligent agents. arXiv preprint arXiv:1809.02627.
- [2] Alpaydin, E. (2020). Introduction to machine learning. MIT Press.
- [3] Håkansson, C. and Fröberg, J. (2021). Application of machine learning to construct advanced NPC behaviors in Unity 3D.
- [4] Xiqi, X. (2024). Game AI Evolution: Uniting Unity and Machine Learning for Enhanced Player Interaction and Experience.
- [5] Andersson, P. (2021) Future-proofing Video Game Agents with Reinforced Learning and Unity ML-Agents.
- [6] Alilovic, D. (2022). Developing a Machine Learning Model for Playing a Platformer Video Game in Unity and TensorFlow (Doctoral dissertation, Fakultet organizacije i informatike).
- [7] Mahmoudi, R. and Ostreika, A. (2023) Reinforcement Learning for Obstacle Avoidance Application in Unity ML-Agents. In IVUS, 214-221.
- [8] Lai, J., Chen, X.L. and Zhang, X.Z. (2019) Training an agent for third-person shooter game using unity ML-agents. International Conference on Artificial Intelligence and Computing Science, 317-332.
- [9] Lukas, M., Tomicic, I. and Bernik, A. (2022) Anticheat system based on reinforcement learning agents in unity. Information, 13(4): 173.
- [10] Pai, N.S., Chen, P.X., Chen, P.Y. and Wang, Z.W. (2022) Home Fitness and Rehabilitation Support System Implemented by Combining Deep Images and Machine Learning Using Unity Game Engine. Sensors & Materials, 34.
- [11] Li, H.C. and Jing, L. (2023) 3D Monitoring of Toothbrushing Regions and Force Using Multimodal Sensors and Unity. IEEE Access, 11: 94753-94771.
- [12] Gao, X., Zhang, L. and Huang, X. (2024) Exploring Spatial Arrangement of Locker Area in Ski Resort Hakk for Enhancing Interpersonal Distancing Using Unity Engine.
- [13] Lopes, M.W.R. (2020). Generating procedural dungeons using machine learning methods: an approach with Unity-ML.
- [14] Nopawong, E. and Praditsangthong, R. (2020) Developing a Dynamic Decision Making of an Enemy Character in a Tower Defense Game using Reinforcement Learning Technique in Unity ML-Agents. RSU International Research Conference 2020, 726-735.