ADAPTIVE TRAINING AND GAMIFICATION FOR SIMULATORS

Gabriel Low[1], Ng Yan Xi Jericho[2]
St. Joseph's Institution[1], 38 Malcolm Road, Singapore 308274
Victoria School[2], 2 Siglap Link Singapore 448880

Mentors: Wee Hong Wei, Kenny[3], Ong Dao Nan[3]
Defence Science and Technology Agency[3], 1 Depot Rd, Singapore 109679

Abstract

The use of Adaptive Training within military training simulators has been gaining traction in the world. With the rapid development of Artificial Intelligence among other technologies, we want to use our work to delve into the possibilities of applying Adaptive Training in the military. Just in the last couple of years, the first few attempts at using Adaptive Training in a pilot training context were made. To further develop this field, we are exploring the use of two models alongside each other, a Machine Learning model and an Adaptive Learning model, to further improve the efficacy of military training using simulators. In this study, we evaluated various models and selected the most suitable Machine Learning model (Advantage Actor-Critic) and Adaptive Learning model (Intelligent Intervention System) that best fit our use case. An implementation approach to integrate the models into simulator architecture was also discussed. This study paves the way forward for the practical application of Adaptive Training in military training simulators in the future.

1. Introduction

The large success of Artificial Intelligence (AI) applications in the modern world, as seen in the rapid rise and large-scale dominance of OpenAI and ChatGPT, has brought many insights and questions about the full potential of Machine Learning (ML) architectures. One possible use case that has arisen from this success in AI is the use of Adaptive Learning (AL) in training.

AL is defined as a personalised approach to teaching that leverages on technology to tailor the learning experience to the needs of each individual student. Through the use of models and algorithms, it dynamically adjusts the delivery of the curriculum based on learner's comprehension and performance [1]. AL aims to increase the proficiency levels of learners with maximum efficiency by balancing cognitive workload and engagement. This balance between the two is known as the flow state. AL aims to achieve user flow state through personalisation. Content delivery is adjusted according to the user's engagement with the curriculum material, and AL models adjust themselves to best fit a trainee's perceived skill level, strengths and weaknesses. One such example is the language learning platform Duolingo, with over 500 million users worldwide. Duolingo's algorithm assesses the user's language proficiency level, determining strengths and weaknesses in their knowledge. Based on the defined strengths and weaknesses, it will generate similar questions targeting user's weaknesses, while ensuring the constancy of their strengths. By focusing on bridging the gap between the user's strengths and weaknesses, Duolingo personalises each learning experience based on the user's needs and preferences, which is difficult to achieve in a traditional classroom.

AL can be categorised into two types, micro- and macro- AL, which are explained below:

1. Macro-adaptive systems tailor course delivery according to the user's profile. This tailoring could be based on preselected preferences made by the user on how they would like to learn.
2. Micro-adaptive systems evaluate how the user interacts with the course content or material, adjusting content delivery and difficulty based on the user's perceived strengths and weaknesses.

Ultimately, micro-adaptive systems offer a higher level of adaptivity than macro-adaptive systems. This study will analyse the use of different models that fall under micro-adaptive systems. The use of a personalised learning experience has already been widely implemented in the academic and educational field, as we have seen from the success and effectiveness of Duolingo. AL thus holds great potential for success, with its use becoming more commonplace.

One current area of interest in the latest developments on AL has been in the context of military training. The prevalence and significance of simulators for military training has been ever increasing with the technological advancements in search for realistic yet low-stakes scenarios to prepare soldiers for actual warfare. Currently, military training simulators have become critical tools in training soldiers in a controlled and low-risk environment. The use of simulation in training also allows users to overcome spatial limitations through simulated environments.

Existing simulators perform well in terms of realism and performance, but would benefit from the implementation of adaptivity, which addresses individual learning needs of trainees. Their effectiveness is then limited as the current "one-size-fits-all" method fails to reinforce individual trainee weaknesses effectively, negatively impacting skill development and retention. For instance, weaker trainees within a cohort will find it more difficult to catch up as a standardised, static approach might be inadequate at targeting and addressing their weaknesses effectively. Similarly, stronger trainees with fewer weaknesses will also find it difficult to address their weaknesses and improve more effectively. With the implementation of adaptivity and the possibility of personalisation, course delivery can be adjusted for each trainee to ensure that no trainee falls too far behind from the rest of the cohort by emphasising on weaknesses, while at the same time ensuring no trainee feels held back by enhancing their strengths and finetuning their performances.

Additionally, the static nature of current training simulators provides limited actionable insights for trainee improvement and might even result in trainees attempting to "game the system", with desirable performance being a result of muscle memory rather than true proficiency.

As such, the objective of this project is to discuss and evaluate several types of ML and AL models, to determine and suggest the best models for implementation in a military training simulator context.

2. Previous Work

Early applications of adaptive training have shown to increase overall proficiency of trainees using a naval simulator, compared to a control group using a conventional simulator [2]. The use of adaptive Computer-Generated Forces (CGF) for training combat pilots has also been explored. For instance, a study on Adaptive CGF for pilots training in air combat simulation demonstrated that an adaptive CGF, driven by a family of self-organising neural networks, could learn from real-time interactions with human pilots and extend existing doctrines [3]. This adaptive approach was shown to be more effective than a non-adaptive, doctrine-driven CGF in engaging trainee pilots in 1-v-1 dogfights, providing both quantitative results and qualitative assessments that highlighted the benefits of adaptivity in training simulation.

3. Methodology

We propose the use of two agents in conjunction, a ML agent and an AL agent. The ML agent acts as a form of "model student", and will learn to use and control the simulator as a virtual trainee. When it reaches a desired level of proficiency, it will fly alongside the actual trainee during simulator training. Both the ML agent's and the trainee's input data will be sent to the AL agent, who measures the discrepancy between the two sets of data and through which identifies trainee strength and weaknesses. Finally, the AL agent selects the appropriate intervention and adjusts the simulator as necessary.

We have selected this two-step training process for several reasons. Such an approach, while more complex, holds an advantage over using a single pedagogical model by producing more robust agents [4]. Leveraging on the adaptability and flexibility of the ML agent acting as the "model student" and providing its data to the AL agent, we can ensure that the AL agent will be able to adjust the environment to adapt to the needs of the trainee, while constantly receiving high quality data from the ML agent to better ascertain trainee performance. This enables dynamic, real-time interaction between the trainee and the AL agent. This not only provides a more personalised learning experience, but also allows the simulator to address the trainee's weaknesses as soon as they are recognised by the AL agent. The AL agent's role in measuring the discrepancy between the ML agent's actions and the trainee's input also allows for precise identification of the trainee's strengths and weaknesses, enabling more targeted interventions.

4. Evaluation of Machine Learning Models

AL methods are often built upon a ML algorithm that gathers data about a user, and learns how to best fulfil their learning needs over time. Hence, the involvement of ML will be necessary. The ML models discussed in this study will be evaluated against the following criteria:

1. Data size required for success: the data sample size required to be fed to the agent for it to reach an acceptable level of accuracy.

2. Precision: given that the proposed use case is for a military training simulator emulating real, high-stakes scenarios, agents will be expected to be able to perform their tasks at the highest level of precision and safety.
3. Scalability: the learning model needs to be scalable to support diverse training scenarios and a large number of users.

A type of ML commonly used for AL is called Reinforcement Learning (RL). In RL, agents interact with the environment at random, and adjust behaviour based on set objectives through a reward and punishment system that is designed by the developers [5]. As the agent goes through a long trial and error process, the actions of the agent slowly become more and more optimal. Agents collect data based on its actions and its resulting states, then make updates to agent policy to adjust to the collected data. The policy is a function that takes into account the current state and decides the new action to take, based on what earned rewards from previous trial and error data.

Instead of being fed a dataset, RL models interact with their environment and generate their own data in the process, altering their agent policy in accordance to the goal. Unlike other forms of ML, RL models do not require an initial dataset, but instead learn to work towards a goal using a system of rewards and punishments. RL models can be categorised into three distinct groups.

4.1 Value-Based Reinforcement Learning

Value-based RL models estimate the value of reward of certain actions in a given state, prioritising maximising the long-term reward. An example would be Q-learning, which is an off-policy, model-free RL algorithm that aims to learn the quality (Q-value) of actions in various states. Such models are effective when it comes to learning in a small action space. In a specific RL environment, action space is defined as the set of all effective actions of the agent [6]. The most prominent from this category would be the Q-learning model, which learns by learning the quality of actions, or their Q-value in various states, using the Bellman Equation.

4.2 Policy-Based Reinforcement Learning

Policy-based RL models aim to directly learn the optimal policy, or the most effective available strategy by mapping states to the probabilities of the agent selecting a particular action within an action space. Through the reward-punishment system, the agent alters the probabilities of taking actions in different states, increasing the probability of the agent picking “good” actions, and decreasing the probability of “bad” actions. One such model from this category is Proximal Policy Optimisation (PPO). PPO models aim to train AI using a method of “clipping”, which decreases the probability of undesirable behaviour and increases probability of desirable behaviours. They focus on stable updates to agent policy, leading to consistent improvements in behaviour. PPO has been researched and implemented by OpenAI, who have found it more effective for the ChatGPT use case compared to other models such as Trust Region Policy Optimisation (TRPO), among others [7].

4.3 Actor-Critic Approach

The actor-critic approach to RL comprises two neural networks, namely "actor" and "critic", which work in a feedback loop to determine the best action in each given state within an action space [8]. The actor first suggests a set of actions in a given state based on its current policy. The critic then estimates the value of the reward that will be received as a result of these actions, providing feedback to the actor agent. Upon the completion of an action, the actor and critic then adjust their policy and parameters accordingly.

A popular algorithm in this category is the Advantage Actor-Critic (A2C) model. The model's behaviour is influenced by policy gradient and value function approaches. It uses a stochastic policy that maps each state to a probability distribution over actions. Given the resulting state after the action, Advantage is calculated using the difference between the estimated reward by the critic and actual received reward. Adjustments are made to the actor network by updating policy, while adjustments are made to critic network to reduce the Advantage function by adjusting parameters, moving the estimated reward values closer to the actual values. The process of training repeats until the actor agent consistently chooses actions that maximise cumulative rewards.

Table 1: Evaluation of Machine Learning Models

Model Type	Data Size (relative)	Precision	Scalability
Q-Learning (Value-Based)	Small, but gets bigger as complexity of scenario increases.	Low, especially in complex action spaces.	Low: Simple and effective for small action spaces, however struggles to handle larger environments or scenarios.
Proximal Policy Optimisation (Policy-based)	Very large	High	High: Easily scalable to handle both simple and more complex scenarios.
Advantage Actor-Critic (Actor Critic Approach)	Moderate	Moderate	High: Can suit both simple and more complex scenarios.

Table 1 shows an overview of the pros and cons of each ML model. We have decided that the **Advantage Actor-Critic (A2C)** model is the optimal choice for application to a military training simulator. The A2C model has shown to be highly effective in tackling complex action states, as well as being data efficient. A2C excels in high dimensional and continuous action

spaces, such as flight simulators, where control inputs such as throttle, pitch and yaw require continuous adjustments.

Unlike Q-learning, which struggles with high-dimensional action spaces, A2C's actor-critic architecture effectively handles complex tasks by learning both the optimal policy (actor) and value function (critic) simultaneously. This results in better sample efficiency and faster convergence in environments with large state spaces and sparse rewards, typical in military training simulators. Furthermore, A2C's ability to reduce variance and stabilise learning makes it a reliable choice for high-stakes training scenarios, where precision and safety are paramount. Unlike PPO, which can be computationally expensive, A2C balances between performance and efficiency, offering both high adaptability and lower resource demands—ideal for military training simulators that need to operate in dynamic, real-time settings. Hence, the A2C model takes the best of both Q-learning and PPO models, while effectively addressing their pitfalls.

5. Evaluation of Adaptive Learning Models

AL algorithms gauge the user's proficiency in different aspects of the task at hand, identifying strengths and weaknesses that allow it to propose interventions to help the learner best improve their skills. In our use case, our agent will be fed the data from both the trainee and the ML agent "model student". The AL agent is then required to carry out two tasks:

1. Measure the differences between the datasets of the trainee and ML agent during training, determining trainee skill mastery.
2. Make appropriate interventions based on the determined strengths and weaknesses of the trainee.

<u>Task 1: Measure Performance Difference</u>

Given the simulator data from both the trainee and the ML agent, there are two approaches to measure the differences in performance between the two. The first would be a Formula-Based Approach which directly determines the difference between each corresponding data point. An example of a formula is shown below.

$$Performance\ Difference\ (PD) = \sqrt{\sum_{i=1}^{n}(T_i - M_i)^2}$$

Where:
T_i is the trainee's data for metric i
M_i is the ML agent's data for metric i
n is the number of performance metrics

The second approach would involve the use of an algorithm that leverages on computational methods to analyse data, identifying patterns or trends that indicate the trainee's strengths and weaknesses. Such algorithms include the Bayesian Knowledge Tracing (BKT), which tracks a learner's knowledge state over time and dynamically adjusts the curriculum to target weaknesses effectively.

A formula-based approach is an intuitive and straightforward method to directly make comparisons between the AL agent and the trainee as it allows for real-time assessments with minimal resource requirements. However, an algorithmic approach provides greater opportunity to truly understand the trainee. By not only discerning differences in the raw data but also identifying patterns to determine the strengths and weaknesses of the trainee. A model such as BKT becomes incredibly strong in this aspect. BKT is a predictive model built to estimate the learner's mastery of skills based on their performances. This allows it to keep track of and determine which skills the learner displays proficiency in, and those that require interventions by the model. The use of BKT as a diagnostic tool in conjunction with an intervention engine allows trainees to better understand which areas of their flying require intervention, and provides the simulator the ability to make such interventions.

Task 2: Intervention

To personalise the trainee's learning experience, the use of an intervention engine is necessary. Three notable options are the Intelligent Intervention System, Behaviour Change Interventions, and Rule-Based Expert System.

5.1 Intelligent Intervention System

The Intelligent Intervention System (In2S) is feedback-based and designed based on learning analytics. It includes three types of intervention: instructional, supportive, and motivational [9]. The instructional intervention uses signal lights (red, yellow, and green) to guide learners through assessment tasks, giving immediate feedback on their performance. The signal light given depends on how well the learner did, for example, red for bad and green for good. Supportive interventions are presented via a dashboard, and motivational interventions incorporate gamification elements like leader boards and badges for user engagement. This system has been evaluated positively by learners, indicating its usefulness and potential for broader application.

5.2 Behaviour Change Interventions

Behaviour Change Interventions focus on understanding the target group, behaviours that need to change, and the context in which change will occur. For example, they provide objectives for the target group, and provide rewards for completing them. These interventions are developed systematically, considering mechanisms of behaviour change techniques. They are designed to be iterative, allowing for continuous testing and refinement. This approach ensures that interventions are tailored to the specific needs and contexts of the learners. However, in the context of a military training simulator that may involve large and complex scenarios, an objective-based intervention approach may not be the most suitable as it may not be sufficient to fully encapsulate the complexities of the battlefield.

5.3 Rule-Based Expert System

Rule-based expert systems are a type of AI that uses a set of "if-then" rules to solve problems or make decisions. These systems emulate the decision-making abilities of a human expert in a specific field. Such systems use a knowledge base containing the rules for the system's decision making. For example: "If the trainee has a lower-than-expected fuel level, then consider the possibility of a more efficient flight path." An expert module then applies the rules and assesses the trainee performance, before making a decision on the appropriate intervention based on the knowledge it has to best suit the learning needs of the trainee.

Table 2: Evaluation of Adaptive Learning Models

Model Type	**Pros**	**Cons**	**Scalability**
Intelligent Intervention System	The real-time feedback provides timely guidance to uses in correcting behaviour or actions.	Risk of providing too much assistance to users, thus creating an over-reliance on the AL system. Complex, costly and time consuming to implement.	High: adaptable across domains, and scalable for multiple users in the same simulation.
Behaviour Change Interventions	Encourages motivation due to having many objectives.	Too narrow-minded for a complex domain, as objectives may be too simplistic.	Low: Not as effective with complex domains like military simulators.
Rule-Based Expert System	Binary, straightforward "rules" ensure efficiency in assessment and selection of interventions.	Overly-simple for a military training context, may not provide a complete or fully accurate coverage of adaptivity.	Low: Poor scalability limits system's ability to work with changing or increasing number of variables to keep track of.

Table 2 shows an overview of the pros and cons of each AL model. When comparing these three options, the **Intelligent Intervention System (In2S)** stands out as the most suitable for applying AL to military training simulators. This is because military training often involves complex, dynamic scenarios that require real-time adaptation and personalised feedback. Given that interventions are made based on understanding differences in the simulator data of the trainee and the ML agent, the data-driven approach of IIS proves to be the best option to determine trainee proficiency, out of the three methods discussed.

Hence, for the AL agent, we propose the use of BKT in conjunction with In2S to (1) determine the skill level of the trainee, and (2) make the correct intervention to the simulator environment to best suit the trainee's learning.

6. Implementation

The ML and AL models involved will have to be compatible with one another, meaning that the ML model outputs must be interpretable by the AL model. The feasibility of the models being implemented into the simulator architecture will also have to be considered. Simulators are often built upon open standards such as High Level Architecture (HLA), or Distributed Interactive Simulation (DIS). In our case, we will be referencing HLA for implementation.

In HLA, the integration process requires the design of Federates (components in the HLA framework) for both the ML and AL agents. These Federates must adhere to the Federation Object Model (FOM), which specifies the shared data objects and interactions within the simulation [10]. The ML and AL agents both function as separate federates that exchange data and feedback. For communication between such federates and the simulator, a publish-subscribe model will be used, where each federate "publishes" its own data for other federates, and "subscribes" to the data that it requires. By leveraging the robust features of HLA, we can create a cohesive, adaptive, and scalable training system that can be easily implemented into current simulators.

7. Limitations and Future Work

In this study, we have explored the different ways adaptivity can be implemented in military training simulators. However, there were several gaps that could potentially be addressed in future works and research. Firstly, all discussion in this study has been theoretical in nature, and its viability for actual application in the field remains untested. Future work could explore the application of such models in the military, which would address concerns over model accuracy, computational demands and time needed to train models. A thorough evaluation of the system's performance taking on real world training scenarios would be crucial in gaining insights on aligning the possibilities of adaptivity to the operational needs of the military.

Another significant limitation lies in the specificity of the proposed models to certain types of training scenarios. The models discussed may not be universally applicable to all forms of military training, which could limit their effectiveness. Future research could look into developing more versatile models that can adapt to various types of training scenarios, ensuring broader applicability and utility. Furthermore, the technicalities of integrating each model, with each other as well as with the system and user interface, was not explored in this study and thus remains theoretical.

Additionally, the research has not addressed the potential security concerns associated with the implementation of AL systems in military training. The use of AI and data-driven models raises questions about data privacy, consent, and the potential for misuse. Future work should consider these security implications and develop frameworks to ensure that the use of AL systems is secure.

8. Conclusion

In conclusion, we have explored and evaluated the viability of applying AL to military training using simulators, and also proposed a possible method of application through the combination of a ML model and an AL model to deliver real-time insights and interventions to trainees. The integration of adaptive architecture in simulators signifies a significant step forward in the modernisation of military training and builds the foundation for further advancement of simulator-based training.

This study highlights the potential of AL systems to enhance the effectiveness and efficiency of military training. By providing personalised feedback and interventions, these systems can help trainees achieve higher levels of proficiency and preparedness in an increasingly complex and dynamic operational environment. However, the journey towards fully integrated adaptive military training simulators is still in its nascent stages. There is much to learn and refine, hence continued research and development in this field will be essential particularly in terms of practical application, model versatility and security considerations.

References

[1] Martin, F., Chen, Y., Moore, R.L. et al. Systematic review of adaptive learning research designs, context, strategies, and technologies from 2009 to 2018. Education Tech Research Dev 68, 1903–1929 (2020). https://doi.org/10.1007/s11423-020-09793-2

[2] Marraffino, M. D., Schroeder, B. L., Fraulini, N. W., Van Buskirk, W. L., & Johnson, C. I. (2021). Adapting training in real time: An empirical test of adaptive difficulty schedules. Military psychology : the official journal of the Division of Military Psychology, American Psychological Association, 33(3), 136–151. https://doi.org/10.1080/08995605.2021.1897451

[3] TENG, Teck-Hou; TAN, Ah-hwee; ONG, Wee-Sze; and LEE, Kien-Lip. Adaptive CGF for pilots training in air combat simulation. (2012). Proceedings of the 15th International Conference on Information Fusion, Singapore, Jul 9-12. 2263-2270. https://ink.library.smu.edu.sg/sis_research/6663

[4] Ustun, V., Kumar, R., Reilly, A., Sajjadi, S., & Miller, A. (2021). Adaptive synthetic characters for military training. arXiv.org. I/ITSEC A

[5] GeeksforGeeks. (2023). Types of Machine Learning. GeeksforGeeks. https://www.geeksforgeeks.org/types-of-machine-learning/

[6] Zhu, J., Wu, F., & Zhao, J. (2021). An overview of the action space for deep reinforcement learning. 2021 4th International Conference on Algorithms, Computing and Artificial Intelligence. An Overview of the Action Space for Deep Reinforcement Learning

[7] Engstrom, L., Ilyas, A., Santurkar, S., Tsipras, D., Janoos, F., Rudolph, L., & Madry, A. (n.d.). Implementation matters in deep RL: A case study on PPO and TRPO. OpenReview. Implementation Matters in Deep RL: A Case Study on PPO and TRPO | OpenReview

[8] GeeksforGeeks. (2024). Types of Reinforcement Learning. GeeksforGeeks. https://www.geeksforgeeks.org/types-of-reinforcement-learning/

[9] Şahin, Muhittin & Yurdugül, Halil. An intervention engine design and development based on learning analytics: the intelligent intervention system (In2S) (2019) (PDF) An intervention engine design and development based on learning analytics: the intelligent intervention system (In 2 S)

[10] Dr. Judith S. Dahmann. High Level Architecture for simulation. (1997). IEEE Conference Publication | IEEE Xplore. IEEE Xplore Full-Text PDF: