

Personalized Dynamic Difficulty Adjustment – Imitation Learning Meets Reinforcement Learning

Ronja Fuchs, Robin Gieseke, and Alexander Dockhorn

Faculty of EECS

Leibniz University Hannover

Hannover, Germany

dockhorn@tnt.uni-hannover.de

Abstract—Balancing game difficulty in video games is a key task to create interesting gaming experiences for players. Mismatching the game difficulty and a player’s skill or commitment results in frustration or boredom on the player’s side, and hence reduces time spent playing the game. In this work, we explore balancing game difficulty using machine learning-based agents to challenge players based on their current behavior. This is achieved by a combination of two agents, in which one learns to imitate the player, while the second is trained to beat the first. In our demo, we investigate the proposed framework for personalized dynamic difficulty adjustment of AI agents in the context of the fighting game AI competition.

Index Terms—Imitation Learning, Reinforcement Learning, Dynamic Difficulty Adjustment, Fighting Game AI

I. INTRODUCTION

Engaging players with an appropriate challenge is fundamental to a rewarding gaming experience. A study by Hagelback and Johansson [1] has shown that static difficulty settings often fail to adapt to individual skill levels, leading to frustration for beginners or boredom for veterans. At the same time, players reported games against an opponent that adapts to their performance to be more enjoyable. Dynamic Difficulty Adjustment (DDA) techniques [2] address this issue by adjusting difficulty based on a player’s performance, therefore, aiming to keep them in a constant state of flow [3].

This paper proposes a novel DDA technique, which aims to train a personalized opponent based on the player’s current behavior. While deep reinforcement learning (DRL) has been shown to result in human-compatible performance in a variety of gaming tasks [4], it is known to converge slowly, previously making it unsuitable for a real-time scenario. To overcome this issue, we implement a two-step process, in which one agent is asked to imitate the player’s behavior, and a second agent is trained to compete against the first. At specific intervals, the player’s current opponent is replaced with the second agent, presenting the player with a customized challenge. The replacement occurs after a specified number of observation steps. For this initial study, we swapped the agents after each single step in order to be suitable for short play sessions.

In the following, we shortly present related work on DDA (Section II) before introducing our proposed agent model (Section III). A preliminary evaluation (Section IV) has shown promising results in terms of reported player satisfaction.

Section V shortly concludes our work. Our related demo aims to present the proposed approach to the conference participants, allowing us to openly discuss the model while collecting more data on user experiences.

II. BACKGROUND AND RELATED WORK ON DDA

To achieve a state of flow [3], the developers’ objective is to provide players with sufficiently challenging activities that result in interesting and meaningful interactions [5]. DDA approaches often trade the designers’ control of the experience with the game’s freedom in creating new and interesting challenges on the fly.

In this work, we expand on previous approaches, which heavily focus on generating personalized levels (e.g. [6]). We not only parameterize an existing AI agent but propose a method to learn an AI opponent’s behavior from scratch. Designers are freed from creating flexible AI opponents but must still identify player performance variables and game mechanics affecting difficulty, as outlined by Hendrix et al. [7].

III. PERSONALIZED DYNAMIC DIFFICULTY ADJUSTMENT

To allow for a personalized dynamic difficulty adjustment (PDDA) experience that adapts to the player’s skill, we propose a framework combining imitation learning and reinforcement learning agents. Seamless integration of our PDDA machine learning-based opponent is achieved through a combination of three agents:

- **Opponent Agent:** The agent the player is currently playing against, which is to be replaced seamlessly while playing the game.
- **Imitation Learning Agent:** An agent that observes the player’s behavior and learns to replicate its actions.
- **Reinforcement Learning Agent:** An agent trained to win against the imitation learning agent.

Starting with a simple rule-based **opponent agent**, we collect observations on the player’s actions. The opponent agent is replaced with the current reinforcement learning agent at fixed intervals. Longer intervals will result in less frequent changes in the opponent, giving the training process more time. This can result in stronger agents to compete against but comes at the cost of less dynamic interactions.

Once enough observations have been stored, the **imitation learning agent** is trained to reproduce the player’s actions. For

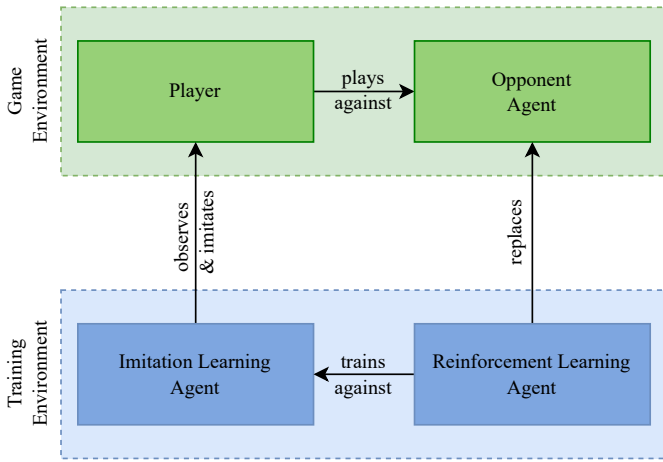


Fig. 1: Proposed agent architecture for personalized dynamic difficulty adjustment.

the development of our prototype, we used the Python-based library *River*¹ due to its implementation of an adaptive random forest classifier [8] that ensures fast learning every time the data set is updated. At the same time, it automatically detects drifts in the training data distribution, allowing the model to adapt to changes in player behavior.

For the implementation of the **reinforcement learning agent**, our prototype makes use of the Advantage Actor Critic algorithm [9] and its synchronous and deterministic implementation by Stable Baselines 3 [10]. We train the agent in a separate training environment running in the background.

IV. PRELIMINARY EVALUATION OF PDDA

This study serves as an initial exploration into the feasibility of our approach. To evaluate it, we implemented the proposed PDDA model in the context of the FightingICE framework [11].

Throughout our study, the imitation learning agent received the relative position of both characters and the actions they currently perform, as well as the player’s input. When predicting player actions, we achieved an accuracy of 82-87 % on the training set with the imitation learning agent (we report a range of values due to the stream mining setup).

For the reinforcement learning agent’s input, we chose to use the current screen encoded as a 96×64 grayscale image. Here, we augmented the image by encoding the characters’ hit points and energy levels in the top left and top right corners of the screen using the grayscale value of two pixels each. For the sake of simplicity, we used the implementation’s default convolutional neural network (CNN) policy². During training, we gave the agent a positive reward for decreasing the other character’s hit points and a penalty for losing its own.

It’s important to note that due to time constraints, our study was conducted with a sample size of 5 participants, who each played three games against the MCTS agent provided by

the framework and the proposed agent model. After every game, they had to rate their experience on a scale from 1-10. Players on average reported higher ratings while playing against the proposed agent model (7.0 ± 1.09) when compared to the MCTS agent (6.6 ± 1.01). We are aware that the system’s expressiveness is limited and requires further testing, i.e. to evaluate the impact of each model’s accuracy on the players’ experience.

V. CONCLUSION AND FUTURE WORK

The framework for personalized dynamic difficulty adjustment presented in this work provides the opportunity to challenge players according to their individual skill levels. The combination of an imitation learning agent and a reinforcement learning agent results in a personalized system for DDA that requires minimal setup by the designers. In the future, we aim to expand on the evaluation of the proposed PDDA by increasing the number of study participants and analyzing the system’s impact on their perceived difficulty and player satisfaction for different opponent agents. Furthermore, the implementation of more complex imitation learning agents could provide a better approximation of the player’s behavior and may result in stronger opponents to match more experienced players. Extending this work to high-difficulty games, as well as exploring the accuracy of the agent to meaningfully generalize to the human player is a future consideration.

REFERENCES

- [1] Johan Hagelback and Stefan J Johansson. Measuring player experience on runtime dynamic difficulty scaling in an rts game. In *2009 IEEE Symposium on Computational Intelligence and Games*, pages 46–52. IEEE, 2009.
- [2] Mohammad Zohaib et al. Dynamic difficulty adjustment (dda) in computer games: A review. *Advances in Human-Computer Interaction*, 2018, 2018.
- [3] Mihaly Csikszentmihalyi and Mihaly Csikszentmihaly. *Flow: The psychology of optimal experience*, volume 1990. Harper & Row New York, 1990.
- [4] Kun Shao, Zhentao Tang, Yuanheng Zhu, Nannan Li, and Dongbin Zhao. A survey of deep reinforcement learning in video games. *arXiv preprint arXiv:1912.10944*, 2019.
- [5] Robin Hunicke. The case for dynamic difficulty adjustment in games. In *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology*, pages 429–433, 2005.
- [6] Georgios N Yannakakis and Julian Togelius. Experience-driven procedural content generation. *IEEE Transactions on Affective Computing*, 2(3):147–161, 2011.
- [7] Maurice Hendrix, Tyrone Bellamy-Wood, Sam McKay, Victoria Bloom, and Ian Dunwell. Implementing adaptive game difficulty balancing in serious games. *IEEE Transactions on Games*, 11(4):320–327, 2018.
- [8] Heitor M Gomes, Albert Bifet, Jesse Read, Jean Paul Barddal, Fabrício Enembreck, Bernhard Pfahringer, Geoff Holmes, and Talel Abdesslem. Adaptive random forests for evolving data stream classification. *Machine Learning*, 106:1469–1495, 2017.
- [9] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937. PMLR, 2016.
- [10] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021.
- [11] Feiyu Lu, Kaito Yamamoto, Luis H Nomura, Syunsuke Mizuno, Young-Min Lee, and Ruck Thawonmas. Fighting game artificial intelligence competition platform. In *2013 IEEE 2nd Global Conference on Consumer Electronics (GCCE)*, pages 320–323. IEEE, 2013.

¹<https://pypi.org/project/river/>

²https://stable-baselines3.readthedocs.io/en/master/_modules/stable_baselines3/common/policies.html