Proceedings of the 4th International Conference on Software Engineering and Machine Learning
DOI: 10.54254/2755-2721/2026.34212

Dynamic Difficulty Adjustment in Games: From Rules to Reinforcement Learning

Luran Wang

Faculty of Engineering, The University of New South Wales, Sydney, Australia
luran.wang@student.unsw.edu.au

Abstract. Static difficulty settings fail to account for variations in the abilities of players as well as their preferences over time, which can lead to either excessive difficulty or boredom, thus affecting the gameplay experience. Dynamic Difficulty Adjustment (DDA) aims to keep a relative equilibrium between player skill and difficulty level of tasks by adjusting the difficulty level in real-time. Over the past few years, DDA research has successively progressed from rule-based static difficulty adjustments to data-driven player modeling, optimization techniques, and reinforcement learning for decisions. But there are many differences between studies in terms of observation signals, player models, control targets, optimisation goals and evaluation processes, which makes it difficult to establish a unified foundation for comparing research results. This paper, from the perspective of review, builds a unified framework of "perception-modeling-decision-execution" to systematically explore the development of DDA technologies from rule-based approaches, dynamic scripting to reinforcement learning. It empowers a discussion of the features and drawbacks of different methods in regard to objective setting, system closed-loop control, evaluation criteria, and practice feasibility. This paper also discusses that the current bottlenecks in DDA research are not only the performance of algorithms, but also the inconsistency in objective settings, lack of evaluation criteria, and reproducibility of benchmark tests. This review could provide a guide for future DDA research on framework design, method comparison and application.

Keywords: Dynamic difficulty adjustment, Game AI, Player modeling, Reinforcement learning

1. Introduction

Gaming is very personal and static difficulty settings may not adapt to the individual's skill level. Too much difficulty results in frustration, while too little can easily lead to boredom. DDA provides a middle ground in which the game automatically adapts the difficulty level in response to player feedback, making it more likely that players will stay in a state where the difficulty level is relatively matched to their skills, thus increasing their sustained interest and enjoyment of the game [1-3]. However, DDA also comes with design considerations. If it is obvious that adjustments are made, the system may cause a feeling that players are being "manipulated", or even lead to a sense of

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.34212

unfairness. As a result, DDA needs to achieve a good trade-off between adaptability, imperceptibility and acceptability [1].

Over the past years, research on DDA has progressed from initial rule-based difficulty adjustment to a variety of approaches that combine player modeling, data mining and reinforcement learning. But a key structural problem in this field remains: the lack of direct comparability among different research results [3-5]. DDA systems may use different observation methods (such as telemetry or multimodal data), different modeling methods (such as skill estimation or type classification), different adaptation mechanisms (such as rule-based or reinforcement learning) and different targets and metrics for validation. Current systematic reviews suggest that few true closed-loop systems are available that integrate perception, modeling and adaptation. Knowledge-based methods remain dominant because of their interpretability and efficiency, whereas learning-based methods are limited by the need for large amounts of data, lack of transparency, and engineering challenges [4, 5].

This paper adopts the four dimensions of "perception-modeling-decision-making-execution" as a unified framework. Specifically, perception determines the upper limit of information, modeling determines the controllability of the system state, decision-making determines the control strategy, and execution determines the actual control mechanism and associated risks. Building on this framework, this paper further identifies key bottlenecks in current research, focusing on issues such as inconsistencies between objectives and evaluation protocols, as well as a lack of reproducible benchmarks.

2. Concepts and scope

In this paper, DDA refers to a system that dynamically adjusts difficulty-related mechanisms or content during gameplay based on player-related signals, thereby bringing the experience closer to the target range. This paper does not limit DDA solely to "parameter scaling," but allows its scope to extend to opponent strategies and the content layer (e.g., experience-driven content adjustment), as content-layer adjustments have been systematically discussed in research on experience-driven content generation and are often inseparable from "difficulty" in practice [6, 7].

This paper strives to "minimize enumeration and maximize alignment," prioritizing original/classic works (such as DDA design constraints and dynamic scripting) alongside recent authoritative reviews and framework works (such as modularization, multi-strategy integration, and challenges in closed-loop implementation) to support structural conclusions.

3. Unified analytical framework: perception-modeling-decision-making-execution closed-loop

Within the framework shown in Figure 1, DDA can be viewed as a closed-loop control system with objectives and constraints: the system acquires observations at the perception layer, forms state estimates at the modeling layer, selects control actions at the decision layer, and modifies the controlled object at the execution layer, thereby influencing subsequent experiences. Recent research emphasizes incorporating adaptive systems engineering knowledge (such as modular closed-loop systems, pluggable strategies, and executable "loop skeletons") into DDA, which supports the integration of multiple strategies and reduces engineering coupling [4, 5].

It is important to emphasize that rule-based, optimization, and reinforcement learning are not linear substitutes where one replaces the other, but rather implementation choices at the decision layer under different constraints. When online safety, interpretability, and rollback capability are hard constraints, rule-based or rule-learning hybrid approaches remain advantageous; when the goal

DOI: 10.54254/2755-2721/2026.34212

is long-term personalization and training/validation costs are acceptable, reinforcement learning holds greater potential.

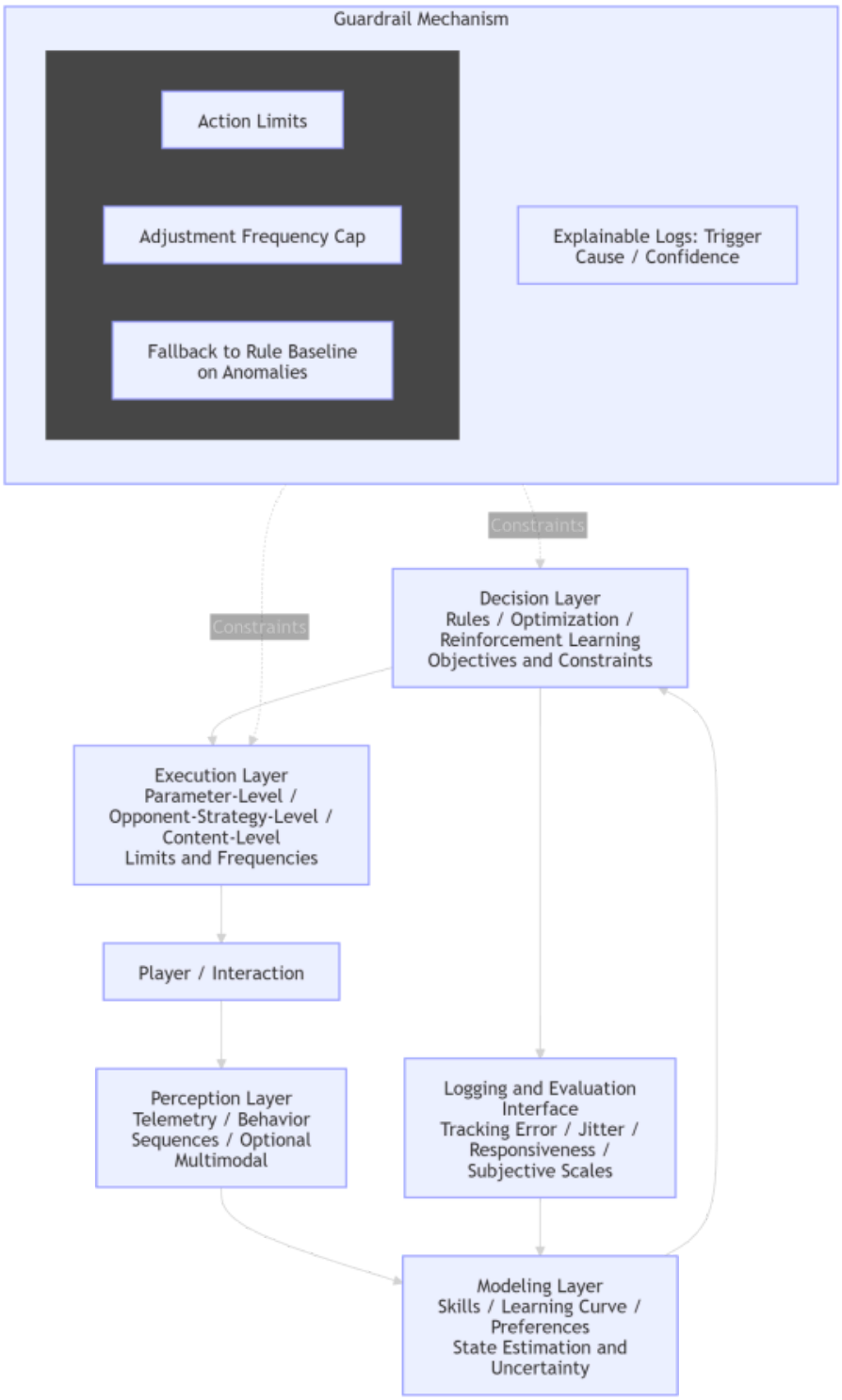


Figure 1. Analytical framework (picture credit: original)

DOI: 10.54254/2755-2721/2026.34212

3.1. Perception layer: player monitoring and data sources

Systematic reviews indicate that in-game telemetry (success rates, number of failures, scores, completion times, injuries/resource consumption, etc.) remains the most commonly used metric due to its low deployment costs, relatively manageable privacy risks, and better reproducibility; while richer non-intrusive sensing (such as facial expressions and peripheral interaction rhythms) may support the inference of emotions and stress, it remains relatively underdeveloped in fully closed-loop systems [4, 5, 8].

Therefore, the perception layer should adhere to a three-way trade-off between "information gain, cost, and compliance": clearly define the sampling frequency, noise sources, and compliance considerations for each type of signal, and ask whether it truly improves "decision controllability" rather than merely increasing offline prediction accuracy [4, 5].

3.2. Model layer: player state representation and online updates

The goal of player modeling is not simply to achieve "more accurate predictions," but rather to provide decision-makers with interpretable, controllable state variables (such as skills, learning curves, typology/preferences, recent fluctuations, and uncertainty). A classic review proposes using clustering to identify player types and supervised models to quickly recognize them, thereby supporting intelligent difficulty adaptation [2, 3, 9].

Closed-loop-wise, there are at least three issues for the modeling layer. First, whether the state variables are equivalent to the "difficulty control objectives" (e.g., skill assessment and challenge adjustment); second, whether state updates are fast enough (short-term trajectory inference); Third, does it quantify uncertainty and support conservative judgement (minimizing the risk of misjudgment, which would result in a poor user experience). These issues of "closed-loop usability" have been frequently pointed out in recent reviews: many of the studies are at the "identification of the experience" or "modeling of the experience" level, but do not include an "identification → adaptation" closed loop [4, 5].

3.3. Decision-making: from rules to optimization and reinforcement learning

3.3.1. Rules and dynamic scripts: interpretable and controllable baselines

Initial DDA studies stressed that adaptations should not over-ride the experience as much as possible; otherwise, they might be perceived as "manipulations". Consequently, threshold, rule and script-based approaches have been the staples [1, 3]. The benefits of these approaches include explainability, auditability, easy definition of boundaries, and reversibility.

Dynamic scripting is an approach in which "rules can be learned online." Through online learning of the weights of the scripting rules, strategies for opponent or difficulty adapt to the player's strategies, thus achieving a trade-off between explainability and adaptability [10].

3.3.2. Reinforcement learning: DDA as a sequential decision making

Reinforcement learning defines the task of "difficulty adjustment" as a sequence of decisions: the system chooses an action in a certain state and receives a reward, thus learns a strategy to obtain the greatest long-term reward. The classic formulation uses a Markov decision process (MDP) and a value function and an objective for policy optimization [11].

DOI: 10.54254/2755-2721/2026.34212

$$s_{t+1}\sim P(s_{t+1}\mid s_t,\ a_t),\qquad r_t=R(s_t,\ a_t,\ s_{t+1}) \tag{1}$$

$$\pi_\theta\left(a_t,\ \mathrm{s_t}\right)=Pr\left(a_t\mid \mathrm{s_t};\ \theta\right),\qquad J\left(\theta\right)=\mathbb{E}_{\pi\theta}\left[\sum_{t=0}^{T-1}\gamma^t r_t\right] \tag{2}$$

In DDA, it is suggested using the intuitive reward structure "target tracking error + smoothing penalty": Let mt be the challenge/performance agent metric (sliding success rate, time, etc.), m* be the target level, and λ be the jitter penalty weight, then

$$r_t=-\left(m_t-m^*\right)^2-\lambda\|\ a_t-a_{t-1}\ \|_1 \tag{3}$$

Value Functions and Action Value Functions:

$$V^\pi\left(s\right)=\mathbb{E}_\pi\left[\sum_{k=0}^{\infty}\gamma^t r_{t+k}\ \mid\ s_t=s\right],\qquad Q^\pi\left(s,\ a\right)=\mathbb{E}_\pi\left[\sum_{k=0}^{\infty}\gamma^k r_{t+k}\ \mid\ s_t=s,\ a_t=a\right] \tag{4}$$

Typical updates in Q-learning [12]:

$$Q\left(s_t,\ a_t\right)\leftarrow Q\left(s_t,\ a_t\right)+\alpha\left[r_t+\gamma\max_{a'}Q\left(s_{t+1},\ a'\right)-Q\left(s_t,\ a_t\right)\right] \tag{5}$$

The convergence conditions and classical theorems of the update rule form the foundation of the theory of reinforcement learning.

In policy optimization, PPO provides a family of practical and stable policy gradient methods, which focus on the usage of a "surrogate objective" and multiple passes of mini-batch training to strike a balance between efficiency and simplicity [13]. Table 1 shows the symbol list.

Table 1. List of symbols

Symbol	Meaning	DDA Semantic Mapping
t	Time Step/Window Index	One episode/one segment/one update cycle
s_t	Status	Player Skills/Stress/Preferences + Game Context
a_t	Action	Difficulty adjustment
P	Transfer distribution	Player Performance/Status Changes After Adjustments

DOI: 10.54254/2755-2721/2026.34212

Table 1. (continued)

r_t	Reward	Quantitative Agents for Experience Goals
m_t	Indicators	Success rate, time required, injuries, resource consumption, etc.
m^*	Objective	Desired level of challenge
γ	Discount factor	Long-term trade-offs
α	Learning rate	Value function update increment
θ	Strategy Parameters	Strategy parameters for learning-based DDA

Table 2. Comparison of key attributes between rule-driven/dynamic scripting and reinforcement learning

Dimension	Rule-driven/Dynamic scripting	Reinforcement Learning
The Nature of Decision-Making	Rules/Scripts; Dynamic scripts can be configured online	Learning Strategies to Maximize Long-Term Returns
Data Requirements	Low (relies on design experience; low deployment costs)	High (training samples/simulation/online data)
Interpretability	Strong (Traceable trigger conditions and actions)	Moderate to low (requires additional auditing and controls)
Long-term	Weak (difficult to observe the delay effect)	Strong (naturally expressed, long-term optimization)
Deployment Risks	Easy to set boundaries, easy to backtrack (safety)	Requires guardrails/limiting/verification/rollback
Versatility	The rules need to be rewritten as the game mechanics change	Transferable but highly dependent on state, rewards, and the target of execution

As Table 2 suggests, one of the reasons for the popularity of knowledge-driven methods, as revealed in the systematic review, is their interpretability and resource efficiency, while learning-based methods suffer from issues of data and interpretability [4, 5]. And, the DDA solutions in the study show the combination of AI and heuristics, which means that the "multi-strategy integration + modular framework" approach is more feasible than "single-algorithm replacement".

3.4. Execution layer: configurable objects and deployment policies

The execution layer refers to the "mechanisms for adjusting difficulty". Typical execution targets can be broadly classified into: the parameter level (enemy life/attack, resource multiplier or time limit); the opponent level (reaction speed, strategy depth); and the content level (level/zone structure, enemy composition, combinations of mission objectives) [3, 6, 14]. Content-level difficulty is often more effective at defining the "nature of the challenge", but it also requires more

DOI: 10.54254/2755-2721/2026.34212

experience modeling and evaluation protocols; recent experience-driven content generation studies offer a general framework that uses "experience goals" as a guide for content generation, and provides theoretical justification for content-level implementation.

The engineering implementation is recommended to use a "layered fallback" strategy: parameter-level as "safe" channel and content-level as "enrichment" channel, falling back to the rule baseline when risk control indicators are triggered (corresponding to the guardrail component in Figure 1) [5, 14]. The latest framework-based research also highlights modularity and the integration of multiple strategies, to reduce the costs of engineering multiple DDA strategies into a game mechanism.

4. Challenges and prospects in research on dynamic difficulty adjustment

4.1. Current major bottlenecks

Current systematic reviews largely concur that the key unresolved problem in the area of DDA is not just the question of whether the algorithms are sufficiently robust, but that there is still no agreement on the definitions of goals and evaluation methods, which in turn means that it is hard to draw a consistent conclusion from different studies. While numerous studies have set their focus on dynamic difficulty adjustment, they vary greatly in terms of control goals, input signals, adjustment targets and evaluation metrics. As a result, the results of these studies are typically only applicable to the specific conditions of the experiment in each study, making it challenging to compare and learn from findings at a higher level. On one hand, many studies also simultaneously involve AI techniques, rule-based methods, and parameter adjustment, revealing that traditional rule-based methods are still a vital part of current DDA systems, rather than being completely replaced by novel techniques. At the same time, related reviews further suggest that future research should aim to provide more generalized, modular and flexible solutions, based on engineering experience in the area of adaptive systems to support the coordination of multiple strategies within a unified system [4, 5, 7].

Another review adds further support from the closed-loop perspective. From 2015 to 2024 in experience-driven adaptation studies, there are still few empirical systems that have implemented an integrated closed-loop "perception-modeling-adaptation" system. The majority of studies still use telemetry data for observation, and most remain knowledge-driven due to their high interpretability and low cost. At the same time, finer-grained experiential aspects like emotions and stresses are less considered in closed-loop systems. This suggests that the current DDA research bottlenecks are first of all the lack of consistent evaluation, closed-loop system engineering skills and system integration, and second the improvement of learning algorithms themselves [4-6].

4.2. Future directions

DDA research in the future should, built on a common set of design goals and evaluation protocols, further promote the engineering and modularization of closed-loop systems. First, the concepts of "appropriate difficulty" or "desired experience" should be defined as specifically as possible, and more uniform evaluation procedures should be developed that align with these research objectives to improve the reproducibility of research results. On the other hand, the collaborative design between perception, modeling, decision-making and execution should be enhanced, ensuring that DDA does not only focus on the optimization of individual modules, but also a whole system that is deployable, verifiable and rollback-friendly [5, 14].

DOI: 10.54254/2755-2721/2026.34212

At the research level, rule-based methods should not be compared with learning-based methods. Instead, more emphasis should be paid to their integration. By using modular structures and multi-strategy integration, more personalised adjustments should be provided while keeping interpretability and safety. Meanwhile, with the ongoing advancement of content-level regulation, long-term player modeling, and experience-oriented optimization, DDA should gradually develop from local adjustment to a holistic experience management approach [4-6]. Only through the continuous improvement of evaluation criteria, closed-loop implementation, and system integration can DDA research build a robust, consistent and practical development framework [4, 5, 14].

5. Conclusion

This paper adopts the "perception-modeling-decision-execution" closed-loop framework to illustrate the development of dynamic difficulty adjustment (DDA) from rule-based approaches and dynamic scripting to reinforcement learning. It describes the major factors driving this evolution, including improving the observability of the execution state, the increasing formalization in the setting of the goal, the ever-expanding target of execution, and the need for long-term personalised adjustment.

Meanwhile, it should be noted that there are still some critical challenges in the current DDA research, mainly reflected in the lack of consistency in the definition of goals and metrics, evaluation procedures, reproducible benchmarking, as well as the need for enhancing the engineering capacity and practical deployment of closed-loop systems.

The modeling of objects and evaluation should be further refined. It should encourage the development of more complex learning-based DDA methods and content-level adaptation mechanisms with the assistance of guardrail and fallback mechanisms, and thus drive the advancement of DDA research that is more incremental, comparable and implementable.

References

[1] Hunicke, R. (2005) The Case for Dynamic Difficulty Adjustment in Games. Proceedings of the 2005 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology, 429-433. DOI: 10.1145/1178477.1178573.

[2] Yannakakis, G.N. and Hallam, J. (2009) Real-Time Game Adaptation for Optimizing Player Satisfaction. IEEE Transactions on Computational Intelligence and AI in Games, 1, 121-133. DOI: 10.1109/TCIAIG.2009.2024533.

[3] Zohaib, M. (2018) Dynamic Difficulty Adjustment (DDA) in Computer Games: A Review. Advances in Human-Computer Interaction, 2018, Article ID 5681652. DOI: 10.1155/2018/5681652.

[4] Mortazavi, F., Moradi, H. and Vahabie, A.H. (2024) Dynamic Difficulty Adjustment Approaches in Video Games: A Systematic Literature Review. Multimedia Tools and Applications, 83, 83227-83274. DOI: 10.1007/s11042-024-18768-x.

[5] Souza, C.H.R., Nascimento, D.F., Berretta, L.O. and Carvalho, S.T. (2025) Solutions for Dynamic Difficulty Adjustment in Digital Games: A Systematic Literature Review. Entertainment Computing, 55, 101041. DOI: 10.1016/j.entcom.2025.101041.

[6] Yannakakis, G.N. and Togelius, J. (2011) Experience-Driven Procedural Content Generation. IEEE Transactions on Affective Computing, 2, 147-161. DOI: 10.1109/T-AFFC.2011.6.

[7] Guo, Z., Thawonmas, R. and Ren, X. (2024) Rethinking Dynamic Difficulty Adjustment for Video Game Design. Entertainment Computing, 50, 100663. DOI: 10.1016/j.entcom.2024.100663.

[8] Huber, T., Mertes, S., Rangelova, S., Flutura, S. and André, E. (2021) Dynamic Difficulty Adjustment in Virtual Reality Exergames through Experience-Driven Procedural Content Generation. 2021 IEEE Symposium Series on Computational Intelligence, 1-8. DOI: 10.1109/SSCI50451.2021.9660086.

[9] Romero-Mendez, E.A., Santana-Mancilla, P.C., Garcia-Ruiz, M., Montesinos-López, O.A. and Anido-Rifón, L.E. (2023) The Use of Deep Learning to Improve Player Engagement in a Video Game through a Dynamic Difficulty Adjustment Based on Skills Classification. Applied Sciences, 13, 8249. DOI: 10.3390/app13148249.

DOI: 10.54254/2755-2721/2026.34212

[10] Spronck, P.H.M., Ponsen, M.J.V., Sprinkhuizen-Kuyper, I.G. and Postma, E.O. (2006) Adaptive Game AI with Dynamic Scripting. Machine Learning, 63, 217-248. DOI: 10.1007/s10994-006-6205-6.
[11] Sutton, R.S. and Barto, A.G. (2018) Reinforcement Learning: An Introduction. 2nd Edition, The MIT Press, Cambridge, Massachusetts.
[12] Watkins, C.J.C.H. and Dayan, P. (1992) Q-Learning. Machine Learning, 8, 279-292. DOI: 10.1007/BF00992698.
[13] Schulman, J., Wolski, F., Dhariwal, P., Radford, A. and Klimov, O. (2017) Proximal Policy Optimization Algorithms. arXiv: 1707.06347. DOI: 10.48550/arXiv.1707.06347.
[14] Climent, L., Longhi, A., Arbelaez, A. and Mancini, M. (2024) A Framework for Designing Reinforcement Learning Agents with Dynamic Difficulty Adjustment in Single-Player Action Video Games. Entertainment Computing, 50, 100686. DOI: 10.1016/j.entcom.2024.100686.