Paper Summary Report - Go with the Flow: Reinforcement Learning in Turn-based Battle Video Games

Joseph Nikhil Reddy Maramreddy
CS 5325.001
Department of Computer Science
Texas State University
yue15@txstate.edu

I. Introduction

The paper investigates whether a Reinforcement Learning (RL) agent can be used as an opponent in a simple turn-based battle game to provide Dynamic Difficulty Adjustment (DDA) that keeps players in a state of flow (neither bored nor frustrated). The authors implement a SARSA agent that chooses actions (rather than simply changing stats) and modify rewards online to induce difficulty shifts when a player is on long winning/losing streaks. They evaluate agent behavior (action distributions) and player experience (Game Experience Questionnaire components) with human participants.

A. *Key claims and results*

- The RL agent learns patterns of play (e.g., preferential use of healing early) and can produce battles that are often "neck-and-neck" rather than one-sided.
- Among tested exploration/exploitation settings, the (30% exploration / 70% exploitation) configuration showed the best tradeoff (quicker learning, good challenge).
- Player ratings (GEQ components) suggest the RL opponents tend to produce more balanced gameplay and sometimes higher flow/engagement scores than static easy/hard opponents, although statistical significance is mixed.

II. Background & Motivation

The paper positions itself in DDA and game flow literature: flow is the balance between perceived challenge and perceived skill [Csikszentmihalyi]. DDA techniques alter game properties at runtime to keep players within the flow channel. The authors argue that, for turn-based battles, modifying opponent behavior via RL (choosing different moves) is an under-explored avenue compared to changing opponent stats or environment.

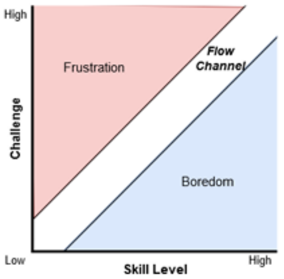


Figure 1: Csikszentmihalyi's flow model. The player's gameplay experience is optimal if it is maintained in the flow channel, i.e., it never becomes too much or not enough challenging, respect to the player's skill level.

III. Game Design & Mechanics

Genre: A simple 2-player turn-based battle: player vs. opponent; both start at full HP (max 75). Each turn the actor selects one of four moves. The battle ends when one side's HP ≤ 0.

Moves available (both sides):

- Hit: weak, low damage (≈2–5).
- Charge: moderate, consistent damage (≈10–15).
- Combo: variable multi-hit (2–5 hits of 2–5 each; up to 25).
- Heal: restores 10 HP.
- (Player additionally has a Flee action to quit.)

Damage calculation: Damage combines the attack of the move and unit attack/defense stats in a formula (not deeply specified in closed form in the paper), but the damage ranges above are empirically given.

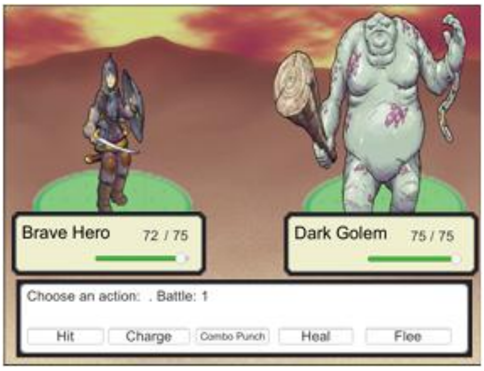

Choose an action: . Battle: 1

Figure 2: screenshot of the game's battle scene. The player's character is depicted on the left, while the opponent's character is on the right. Move selection for the player is shown in the bottom box.

IV. Reinforcement Learning Agent Design

A. *Algorithm*

They use SARSA (on-policy temporal difference method), chosen because SARSA tends to produce safer behavior during learning (performance during learning matters here). The SARSA update used is the canonical one:

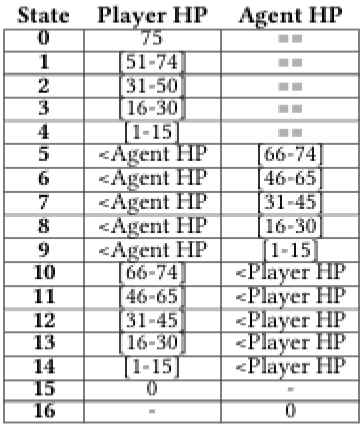
$$Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma Q(s', a') - Q(s, a)]$$

with standard definitions: state s, action a, reward r, next state s', next action a', learning rate α, discount γ.

B. *State space*

States are discrete and derived from the relative HP bands of the two agents. Through empirical tuning they chose 17 states (states 0 – 16).

- States 0–4: both parties have equal HP but in decreasing bands (75; 51–74; 31–50; 16–30; 1–15).
- States 5–9: agent has greater HP (bands).
- States 10–14: player has greater HP (bands).
- State 15: player HP = 0 (agent wins).
- State 16: agent HP = 0 (agent loses).



State	Player HP	Agent HP
0	75	==
1	[51-74]	==
2	[31-50]	==
3	[16-30]	==
4	[1-15]	==
5	<Agent HP	[66-74]
6	<Agent HP	[46-65]
7	<Agent HP	[31-45]
8	<Agent HP	[16-30]
9	<Agent HP	[1-15]
10	[66-74]	<Player HP
11	[46-65]	<Player HP
12	[31-45]	<Player HP
13	[16-30]	<Player HP
14	[1-15]	<Player HP
15	0	-
16	-	0

Table 1: the states for our SARSA RL agent.

C. *Action Space*

The agent's action set is the same move set available in the game (Hit, Charge, Combo, Heal). The agent chooses among these four actions each turn.

D. *Reward design & DDA mechanism*

1) Terminal rewards: a positive reward for agent victory and negative reward for agent defeat. The authors experiment with two magnitudes: ±100 and ±50 (i.e., reward/penalty pairs 100/-100 and 50/-50).

2) Reward swapping for DDA: the system tracks streaks of player wins or losses; after five consecutive wins/losses, the rewards are swapped so that the agent is encouraged to lose (if the player has been losing frequently) or encouraged to win (if the player has been winning frequently). This is their primary DDA lever: switching win/lose terminal rewards to bias the agent toward easing or increasing difficulty.

E. *Exploration and policy*

Epsilon-greedy policy is used with pre-configured exploration/exploitation rates: (50%/50%), (30%/70%), (70%/30%). They also implement a Tabu Search Exploration variant to temporarily forbid recently used actions for some steps to encourage exploring alternatives. The paper highlights the impact of these configurations on learning dynamics.

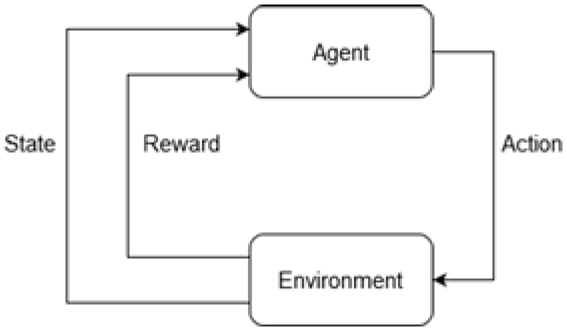


Figure 3: diagram of the SARSA algorithm.

V. Evaluation: Experimental Setup & Results

A. *Studies Performed*

Two studies were performed, one for agent behavior and actions, another for the gameplay experience.

1) Study 1 - Agent Behaviour & Actions

- 10 games (A–J), each game = 30 battles, total 300 battles. Four human players (2 male, 2 female; ages 20–40) with a mix of skill (2 experienced, 1 intermediate, 1 beginner). Games vary reward magnitudes and exploration/exploitation ratios. They analyze which moves players use and which moves the RL agent learns to use over time.

2) Study 2 - Gameplay Experience:

- Players played against different AI types: fixed easy, fixed hard, and RL-based agents with different exploration settings. After sessions they filled the Game Experience Questionnaire (components: success, challenge, skillful, effort - per IJsselsteijn et al.). Authors compare mean component scores and run paired t-tests (with Bonferroni correction) across gameplay types.

B. *Key quantitative observations*

- **Player move distribution:** Players heavily favor Charge and Combo (highest damage). Heal and Hit are used less by players; however, the RL agent learns to favor Heal initially (sometimes over-using it).
- **Agent behavior evolution:** The RL agent starts by exploring and can over-use Heal (e.g., in one game the agent used Heal 25 times in a battle), which dragged out battles and often produced losses for the agent in early phases. Over time, the agent reduces Heal usage and shifts to more attacking moves.

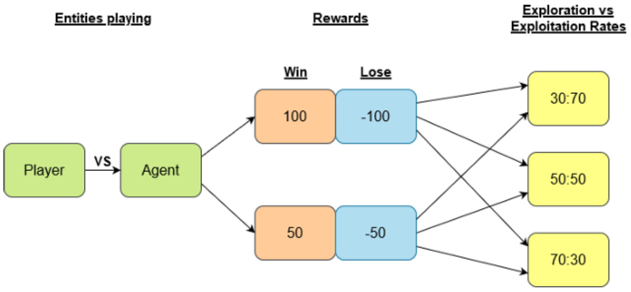


Figure 4: tree diagram with all possible parameter choices.

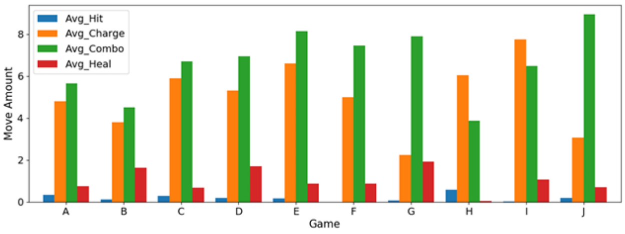


Figure 5: player's average moves over 10 games (A-J).

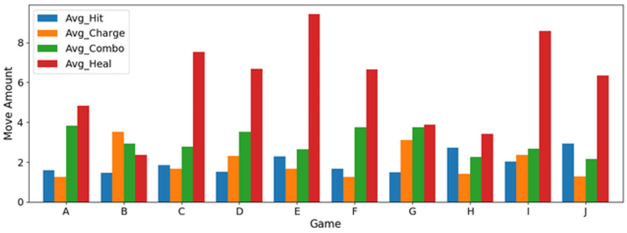


Figure 6: RL agent's average moves over 10 games (A-J).

Figures 6 - 8 illustrate these dynamics (action counts and HP traces).

- **Exploration configuration:** The (30,70) exploration/exploitation setting produced better results in terms of learning quicker and producing challenging but fair battles; authors highlight this configuration as their best empirical performer.

- **Player experience / GEQ:** Mean component scores for RL agents tend to fall in a middle/"balanced" range for many components (interpreted as consistent with flow). Paired t-tests indicate some significant differences between easy/hard and other types for challenge and effort, but many comparisons between RL variants did not reach significance after correction. Authors report that RL variants (especially 30-70) showed promising GEQ means. See Tables 2 - 3 and Figures 7 - 9 for details.

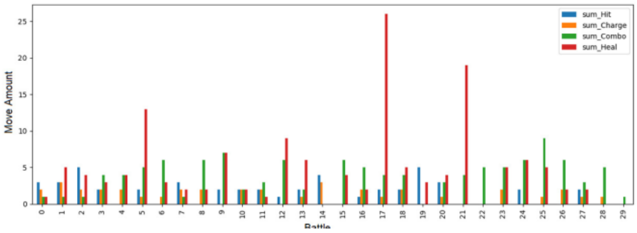


Figure 7: RL agent's moves sum in the 30 battles of game A.

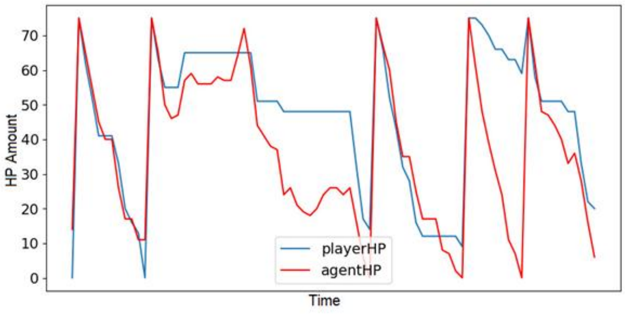


Figure 8: Player's and agent's HPs from battles [16 – 20] of game A.

Comp.	Game type	μ	std		Comp.	Game type	μ	std
success	easy	3	1.33		challenge	easy	0	0
success	hard	**1.5**	0.85		challenge	hard	3.6	0.7
success	50-50	**2.5**	1.18		challenge	50-50	**2.2**	0.79
success	30-70	3.1	1.29		challenge	30-70	3	0.47
success	70-30	**2.2**	1.32		challenge	70-30	**2.5**	0.97
skillful	easy	2.8	1.23		effort	easy	0.1	0.32
skillful	hard	**2.3**	1.42		effort	hard	3.3	1.06
skillful	50-50	**2.5**	1.27		effort	50-50	**2.3**	1.06
skillful	30-70	2.8	1.4		effort	30-70	2.8	0.92
skillful	70-30	**2.5**	1.27		effort	70-30	**2.5**	1.08

Table 2: results of the gameplay experience evaluation study. For each component (successful, skillful, challenge, effort) and gametype(easy, hard, 50-50, 30-70, 70-30) we report the corresponding mean and standard deviation. Medium values (between 1.5 and 2.5), i.e., those corresponding to a "balanced" gameplay experience, are highlighted in bold.

Gameplay type pairs	success	skillful	challenge	effort
easy vs hard	**0.03**	1	**0.001**	**0.001**
easy vs 50-50	1	1	**0.001**	**0.005**
easy vs 70-30	0.107	1	**0.001**	**0.001**
easy vs 30-70	1	1	**0.001**	**0.001**
hard vs 50-50	0.085	1	0.067	0.848
hard vs 70-30	1	1	**0.011**	0.107
hard vs 30-70	**0.002**	0.522	0.239	0.957
50-50 vs 70-30	1	1	1	1
50-50 vs 30-70	0.239	0.811	0.368	1
70-30 vs 30-70	0.1	0.811	1	1

Table 3: results of paired t-tests computed between all the combinations of gameplay types. Significant p-values (p < .05) after Bonferroni correction are highlighted in bold.

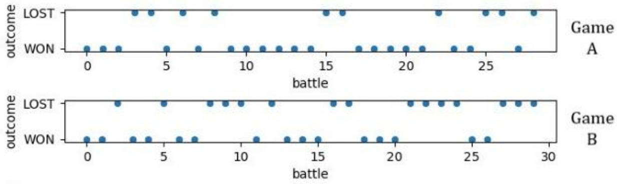


Figure 9: player's outcome for the battles in games A and B.

VI. Critical Analysis

A. *Primary Assumptions*

1. **Flow maps to midrange GEQ scores.** The analysis equates midrange component means (≈1.5–2.5 on their scale) with balanced flow, this is a practical heuristic rather than a validated continuous measure of flow for every context. The mapping is reasonable but indirect.

2. **Discrete HP bands capture state sufficiently.** The authors assume 17 discrete states (HP bands + terminal states) are an adequate summarization of game context for learning good policies. This discretization simplifies learning but hides finer dynamics (e.g., exact HP, move cooldowns, player tendencies).

3. **Terminal reward swapping provides adequate DDA.** The core DDA mechanism is swapping the sign of terminal rewards after five straight wins/losses. This assumes that changing terminal rewards is a robust and timely way to steer learning toward easier/harder outcomes for immediate DDA.

B. Limitations & Weaknesses

1. **Simple environment - limited generalizability.** The game is intentionally small (4 moves, small state set). The paper acknowledges that scaling to richer turn-based systems (Pokémon, Dragon Quest) will increase action/state space and slow learning; results may not transfer directly.
2. **Small participant sample.** Study 1 used 4 players across 10 games (300 battles) and Study 2 appears to have similarly small N. The human sample size and diversity are limited, which reduces statistical power and generalizability of GEQ findings. The authors report that some comparisons are not statistically significant after correction.
3. **Reward shaping & DDA timing are ad hoc.** Swapping rewards after five consecutive wins/losses is an empirical choice; it may be too coarse or too abrupt, potentially destabilizing learning or producing unnatural behavior (e.g., spamming heals to exploit learning dynamic). The paper shows heal-spamming behavior by the agent when Heal has no usage restriction.
4. **Behavioral artifacts (heal spamming).** Because Heal has no usage restriction and the reward structure focuses on terminal outcomes, the agent sometimes learns to endlessly heal (dragging matches) rather than trying effective offensive play. The authors note this and suggest limiting repeated moves or penalizing spamming. This is an instance of reward-function misspecification / insufficient shaping.
5. **Limited state features.** The state uses only HP bands; it ignores move history, relative remaining potential damage (e.g., no modelling of combo probability variance), and player style. Without richer features or function approximation, the agent's policy expressiveness is limited.
6. **No comparison to alternative RL baselines.** The paper motivates SARSA over Q-learning conceptually but does not provide head-to-head performance comparisons (e.g., SARSA vs Q vs simple supervised bandit or model-based planners) to quantify the benefit. That leaves open whether SARSA is optimal for this DDA goal.
7. **Short training horizon.** Learning occurs online during human play; this keeps initial performance noisy. The paper's experiments show the agent learning over games, but an alternative (pretraining on simulated players, then fine-tuning with humans) is not explored. This could accelerate stable behavior.

C. How these limitations affect claims

The positive GEQ means and agent behavior patterns support proof of concept, but the small sample sizes + simple environment mean evidence for broad claims (that RL DDA will improve flow generally) is preliminary. The healing artifact exemplifies how reward design and action constraints can produce undesirable emergent behaviors, a risk for real games if not carefully constrained.

VII. Future Work

1. **Increase participant pool and diversity.** Run larger user studies (N >> 10) across skill strata and collect additional behavioral metrics (time per decision, physiological arousal, retention).
2. **Refine reward shaping.** Instead of swapping terminal rewards wholesale, consider:
   - shaping intermediate rewards to reflect closeness of HP (e.g., +f(|HPdiff|) that peaks when close),
   - penalizing repeated identical actions (anti-spam term),
   - using a moving average of player success to smoothly modulate difficulty. (The paper already experiments with reward magnitudes but not smooth schedulers.)
3. **Richer state features / function approximation.** Use additional features (recent move history, per-move cooldowns, success rates of moves) and consider function approximators (linear function approx. or small neural nets) for larger games. This would help generalize across similar state patterns.
4. **Compare RL algorithms.** Run controlled experiments comparing SARSA, Q-learning, and model-based or policy gradient methods to quantify learning speed, safety during learning, and final policy quality.
5. **Simulated pretraining + human fine-tuning.** Pretrain agents against simulated player policies, then adapt online with humans to accelerate convergence and reduce early absurd behavior (e.g., heal spam).
6. **Personalized agents.** Track per-player models (or per-cluster) to adapt difficulty to player style rather than using a single global RL policy + global reward swap. This would better support long-term engagement.

References

[1] Elinga Pagalyte, Maurizio Mancini, and Laura Climent. 2020. Go with the Flow: Reinforcement Learning in Turn-based Battle Video Games. In Proceedings of the 20th ACM International Conference on Intelligent Virtual Agents (IVA '20). Association for Computing Machinery, New York, NY, USA, Article 44, 1–8. https://doi.org/10.1145/3383652.3423868

[2] Sutton, R. S., Barto, A. G. (2018). Reinforcement Learning: An Introduction. The MIT Press.

[3] Norman, Kent. (2013). GEQ (Game Engagement/Experience Questionnaire): A Review of Two Papers. Interacting with Computers. 25. 278-283. 10.1093/iwc/iwt009.

[4] Sepulveda, Gabriel & Besoain, Felipe & Barriga, Nicolas A.. (2019). Exploring Dynamic Difficulty Adjustment in Videogames. 1-6. 10.1109/CHILECON47746.2019.898806

[5] Zohaib, Mohammad. (2018). Dynamic Difficulty Adjustment (DDA) in Computer Games: A Review. Advances in Human-Computer Interaction. 2018. 1-12. 10.1155/2018/5681652.

[6] Microsoft Research. (2023). Three new reinforcement learning methods aim to improve AI in gaming and beyond. Microsoft Research Blog. https://www.microsoft.com/en-us/research/blog/three-new-reinforcement-learning-methods-aim-to-improve-ai-in-gaming-and-beyond/