

# Multi-Phase Deep Q-Learning for Human-Like NPC Behavior in Fighting Games

Ujjwal Rastogi, MSc. Big Data Analytics and Artificial Intelligence

**Abstract**— Reinforcement Learning (RL) has demonstrated significant success in developing intelligent agents capable of autonomous decision-making in complex environments (Mnih et al., 2015). However, achieving human-like strategic behavior in fast-paced games remains a challenge. This study proposes a multi-phase Deep Q-Learning framework to train a fighting game agent that learns progressively, beginning with fundamental movement and spatial awareness, advancing to combat tactics, and culminating in adaptive behavior against evolving opponents. Each training phase refines specific skill sets through tailored reward functions and dynamic difficulty adjustment. The proposed approach builds upon prior work in hierarchical and curriculum-based RL (Bengio et al., 2009; Florensa et al., 2017), aiming to produce more realistic, strategic non-player characters (NPCs) in game environments. Preliminary results suggest that phase-wise training stabilizes learning and enhances long-term strategy retention, offering a step toward more human-like AI opponents in interactive simulations.

**Index Terms**— Reinforcement Learning(RL), Deep Learning(DL), Machine Learning(ML), Artificial Intelligence(AI), Deep Q-Network(DQN)

## I. BACKGROUND

Reinforcement Learning (RL) enables agents to learn optimal behavior through trial and error, receiving feedback as rewards or penalties (Sutton and Barto, 2018). Deep Q-Networks (DQNs) combined Q-learning with neural networks to handle high-dimensional state spaces, achieving human-level performance in tasks like Atari games (Mnih et al., 2015).

However, DQN-based methods often struggle in complex, dynamic domains such as fighting games, which require nuanced strategies like defense, timing, and counter-attacks. Single-phase training can lead to repetitive or unrealistic behavior (Baker et al., 2019).

To address this, hierarchical and curriculum-based learning decomposes tasks into simpler sub-goals, enabling progressive skill acquisition and better generalisation (Bengio et al., 2009; Florensa et al., 2017). Multi-phase training applies this idea by structuring learning into stages, e.g., focusing first on movement and positioning, then on combat tactics and adaptive decision-making.

In games, realistic non-player characters (NPCs) are crucial for player engagement (Yannakakis and Togelius, 2018). A multi-phase RL framework, with staged learning, dynamic difficulty adjustment, and evolving rewards, can produce adaptive, human-like NPC behavior, mimicking the natural progression of player skill.

## II. METHODOLOGY

### A. Environment Setup

The 2D fighting game environment (Coding With Russ, 2024) was adapted to enable RL training, including agent interactions, state observation, and reward tracking. Episodes

consisted of fixed time steps, and opponents included snapshots of previously trained agents to support dynamic difficulty adjustment (Baker et al., 2019; Silver et al., 2017).

### B. State Space

The agent's state is a 6-dimensional vector capturing relative positions, health, vertical velocity, and attack cooldown.

### C. Action Space

The agent has a discrete action space of 5 actions: Move left, Move Right, Jump, Attack 1, and Attack 2 (the same effect as Attack 1)

### D. Reward Design

Multi-phase rewards shape learning progressively:

- **Phase 1** Basic gameplay - small rewards for moving closer, landing hits, avoiding damage.
- **Phase 2** Adaptation - rewards for winning, strategic actions, and responding to varied opponents.
- **Phase 3** Stability - simplified rewards focusing on spacing and successful hits.

### E. Algorithm

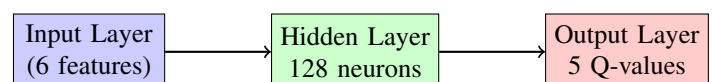


Fig. 1. Compact DQN architecture for the fighting game agent.

The agent was trained using Deep Q-Networks (DQN) with optional Double DQN extensions to reduce the overestimation of Q-values (Hasselt et al., 2016).

- **Network Architecture:**

- Input layer: 6 neurons (state dimensions)
- Hidden layer: 128 neurons, fully connected
- Output layer: 5 neurons (action Q-values)

- **Exploration:** Epsilon-greedy policy with linear annealing, phase-specific adjustments to epsilon decay.
- **Replay buffer:** Experience replay and target network updated periodically to improve stability (Mnih et al., 2015).
- **Multi-phase training:** Phase 2 used Phase 1 snapshots as opponents, Phase 3 used a mixture of Phase 1 and Phase 2 snapshots to promote generalisation (Baker et al., 2019; Silver et al., 2017).

The agent uses an epsilon-greedy policy for exploration, with experience replay and a target network to improve stability (Mnih et al., 2015). Multi-phase training allowed the agent to progressively adapt to complex behaviors, using opponent snapshots to expose it to diverse strategies (Baker et al., 2019; Silver et al., 2017).

### F. Hyperparameters

Key parameters include  $\gamma=0.99$ , batch size=64, replay buffer=100,000, train start=2000, target update=200 steps. Learning rate reduced in Phase 3 to  $5e-5$ ;  $\epsilon$  decayed from 1 to 0.05 over 800–1500 episodes depending on phase.

## III. RESULTS

### A. Phase 1 - Basic Learning

Phase 1 focused on teaching agents the core mechanics of the game, including movement, jumping, and attack. Both Player 1 and Player 2 were trained for 1000 episodes using a DQN architecture with a 6-dimensional state space and 5 discrete actions.

#### Learning Matrices:

- **Smoothed Average Reward per Episode:** Rewards of Player 1 and Player 2 form an alternating “DNA-like” pattern, as illustrated in Fig. 2, a reflection of the adversarial setup where one player’s success inherently means the other’s loss. This indicates the model effectively captured the competitive dynamics of the environment.
- **Smoothed Loss per Episode:** The training loss shows a gradual decline, as illustrated in Fig. 3, confirming that both agents were learning stable policies over time.

**Observation:** Phase 1 demonstrates successful learning of basic combat behavior. The alternating reward patterns validate that the reward system works as intended in a two-player adversarial context, with both agents improving their play mechanics progressively.

### B. Phase 2 – Adaptation & Dynamic Difficulty Adjustment

In Phase 2, the environment was modified to promote adaptability. Player 1 was frozen from Phase 1, while Player 2 continued training against multiple saved snapshots from

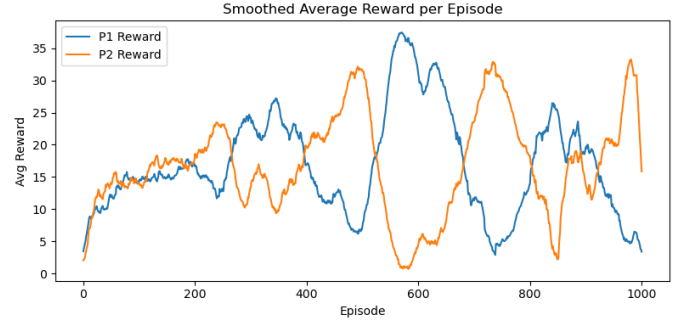


Fig. 2. Smoothed average reward per episode (P1 vs. P2).

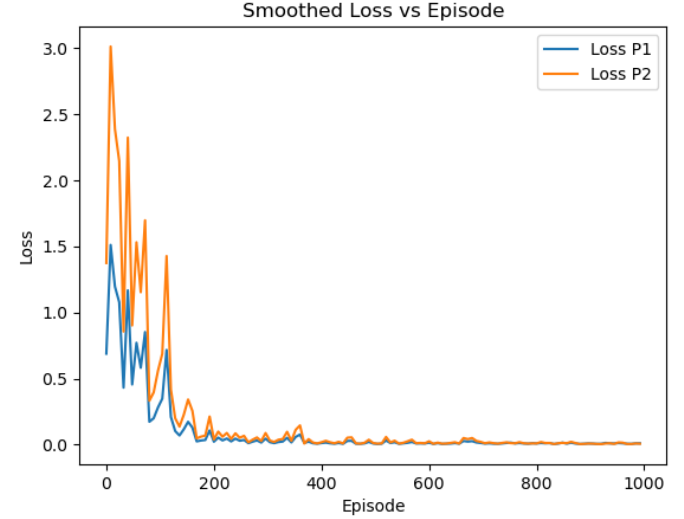


Fig. 3. Smoothed loss per episode.

Phase 1. The idea was to introduce varied opponents, simulating different play styles and encouraging dynamic difficulty adjustment (DDA). Training again lasted for 2000 episodes.

#### Learning Metrics:

- **Smoothed Average Reward per Episode:** The rewards show strong turbulence during early episodes, illustrated in Fig. 4, reflecting the agent’s difficulty in adapting to diverse opponents. Over time, the rewards gradually stabilise, suggesting partial adaptation and improved decision-making.
- **Smoothed Loss per Episode:** Player 1’s loss remains constant, as learning was disabled for it, while Player 2’s loss initially fluctuates before levelling out, as illustrated in Fig. 5, consistent with exposure to a non-stationary training environment.

**Observation:** Phase 2 introduces complexity by exposing the active agent to a range of previously learned opponents. The resulting turbulence and gradual stabilisation align with findings in multi-agent reinforcement learning, where opponent diversity can destabilise early learning but ultimately improve robustness and adaptability (Bengio et al., 2009; Baker et al., 2019).

### C. Phase 3 – Stability and Reward Simplification

Phase 3 builds on Phase 2 by simplifying the reward function to promote more stable learning while training against

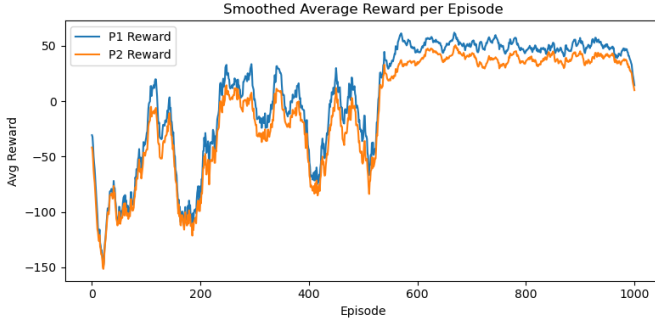


Fig. 4. Smoothed average reward per episode (P1 vs. P2).

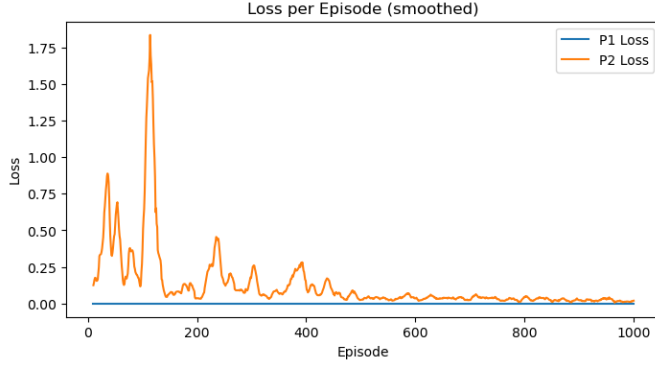


Fig. 5. Smoothed loss per episode.

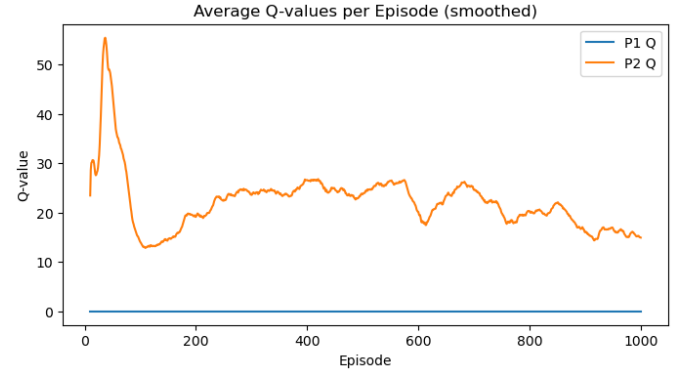


Fig. 6. Smoothed average Q-values per episode.

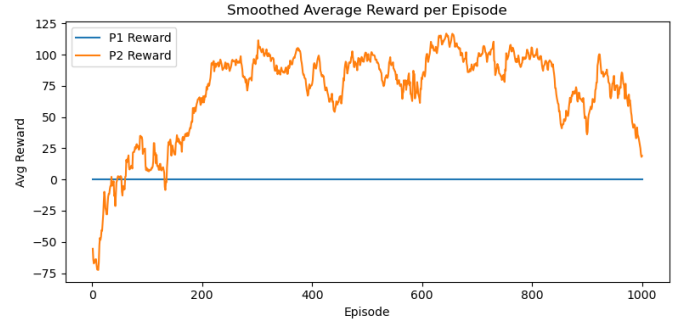


Fig. 7. Smoothed average reward per episode.

a mix of Phase 1 and Phase 2 snapshots. The learning rate was slightly reduced to  $5e-5$  to prevent large gradient updates, and  $\epsilon$ -annealing was extended to 1500 episodes for smoother exploration decay.

#### Learning Metrics:

- **Smoothed Average Q-Values per Episode:** Q-values show an initial sharp rise, illustrated in Fig. 6, indicative of temporary overestimation during early exploration, before gradually settling, consistent with known challenges in DQN’s temporal difference updates (Hasselt et al., 2016).
- **Smoothed Average Reward per Episode:** The reward curve remains turbulent and less structured than in Phase 1, illustrated in Fig. 7, implying that while basic competency is retained, stability and generalisation under varied conditions remain open challenges.

**Observation** Phase 3 shows partial recovery in stability after simplifying the reward structure, though fluctuations persist. The results highlight ongoing difficulties in balancing exploration, value estimation, and opponent diversity, motivating potential improvements such as Double DQN or adaptive reward scaling.

## IV. DISCUSSION

The multi-phase Double DQN training strategy allowed the agent to gradually acquire, adapt, and refine its behaviour in a competitive fighting environment. Each phase offered different insights into learning dynamics and the challenges of stability in adversarial reinforcement learning.

### A. Strengths

The staged approach proved useful for progressive skill acquisition.

- **Phase 1** successfully taught the agents fundamental mechanics such as movement, jumping, and attacking. The mirrored pattern in player rewards reflected the adversarial nature of the environment and confirmed that the reward structure was functioning as intended.
- **Phase 2** focused on adaptability. By training against multiple frozen opponents collected from earlier episodes, the agent learned to adjust to diverse play styles. This self-play-like approach aligns with curriculum learning principles (Bengio et al., 2009) and competitive reinforcement learning studies (Baker et al., 2019).
- **Phase 3** built upon this by simplifying the reward function to encourage smoother convergence. Despite some noise, the average Q-values stabilised over time, suggesting that the Double DQN formulation effectively reduced overestimation bias and improved the consistency of value updates.

### B. Challenges

Several limitations persisted across the phases.

- **Reward Fluctuations:** Rewards exhibited high variance, especially in later phases. This is expected in adversarial training, where one agent’s reward is often inversely related to its opponent’s performance, creating oscillations in cumulative returns.
- **Stability Issues:** Although Double DQN helped mitigate overestimation, temporary Q-value spikes and erratic

learning patterns were observed. These may have resulted from high opponent variability or insufficient replay buffer diversity. Methods such as Dueling DQN (Wang et al., 2016), Prioritised Experience Replay (Schaul et al., 2016), or entropy-based regularisation could help smooth learning.

- **Generalisation:** While the agent performed reasonably against familiar snapshots, it struggled against unseen behaviours, showing limited generalisation. Broader opponent sampling or domain randomisation might help the agent learn more transferable strategies.

## V. LIMITATIONS AND FUTURE WORK

### A. Limitations:

- 1) **Reward Fluctuations and Learning Stability:** Although reward shaping was applied across phases, with different rewards tailored to each learning stage, some reward volatility remained, particularly in Phase 3. This limited the smoothness of policy convergence.
- 2) **Exploration-Exploitation Balance:** Despite using  $\epsilon$ -greedy exploration with decay, the agent sometimes converged to suboptimal but safe strategies, such as excessive jumping, which reduced exposure to varied combat scenarios.
- 3) **Limited Generalisation:** While multiple snapshots from previous phases and early DDA mechanisms improved adaptability, the agent still performed best against known opponent behaviours and showed limited performance against entirely new strategies.
- 4) **Training Efficiency:** Training required thousands of episodes to reach semi-stable behaviour. Computational demands and the need for large replay buffers made experimentation slow.

### B. Future Work:

- 1) **Advanced Curriculum and Self-Play:** The current snapshot-based opponent pool is a form of self-play. Future work could implement continuous self-play, where the agent constantly trains against evolving versions of itself. This may improve learning stability and adaptability.
- 2) **Population-Based Training:** Introducing a population of agents trained concurrently with diverse policies or hyperparameters could improve robustness. Agents could share successful strategies or replace weaker policies, leading to faster convergence and better generalisation.
- 3) **Expanded Opponent Diversity:** Increasing the number and type of opponents in the training pool, possibly including procedurally generated strategies, could improve generalisation and make the agent more resilient to unseen behaviours.
- 4) **Improved Stability Methods:** Although Double DQN mitigated overestimation, additional architectural improvements, such as Dueling DQN or Prioritised Experience Replay, could reduce residual instability and Q-value spikes.

## VI. CONCLUSION

This project demonstrated a multi-phase Double DQN framework for training a fighting game NPC with progressively complex behaviors. Phase 1 established basic skills, Phase 2 improved adaptability through training against diverse snapshots, and Phase 3 refined strategies with a simplified reward function.

Results show that Double DQN mitigated overestimation bias and enabled semi-human-like behaviors, though challenges like reward fluctuations, limited generalization, and residual instability remained. Future work with continuous self-play, population-based training, and enhanced exploration could improve stability and adaptability. Overall, structured multi-phase training is an effective approach for developing dynamic and responsive NPCs in competitive games.

## VII. REFERENCES

- 1) Florensa, C., Duan, Y., & Abbeel, P. (2017) ‘Reverse curriculum generation for reinforcement learning’, Proceedings of the Conference on Robot Learning (CoRL), pp. 482–495.
- 2) Baker, B., Kanitscheider, I., Markov, T., Wu, Y., Powell, G., McGrew, B., ... & Mordatch, I. (2019) ‘Emergent tool use from multi-agent autocurricula’, Proceedings of the International Conference on Learning Representations (ICLR).
- 3) Bengio, Y., Louradour, J., Collobert, R., & Weston, J. (2009) ‘Curriculum learning’, Proceedings of the 26th Annual International Conference on Machine Learning (ICML), pp. 41–48.
- 4) Hasselt, H. van, Guez, A., & Silver, D. (2016) ‘Deep Reinforcement Learning with Double Q-learning’, Proceedings of the AAAI Conference on Artificial Intelligence, 30(1), pp. 2094–2100.
- 5) OpenAI (2019) OpenAI Five: The Reinforcement Learning Agents that Play Dota 2 at a High Level. Available at: <https://openai.com/five/> (Accessed: 13 November 2025).
- 6) Russs123 (2019) brawler\_tut. GitHub. Available at: [https://github.com/russs123/brawler\\_tut](https://github.com/russs123/brawler_tut) (Accessed: 13 November 2025).
- 7) Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., Van Den Driessche, G., ... & Dieleman, S. (2016) ‘Mastering the game of Go with deep neural networks and tree search’, Nature, 529(7587), pp. 484–489.
- 8) Schaul, T., Quan, J., Antonoglou, I., & Silver, D. (2016) ‘Prioritized Experience Replay’, Proceedings of the International Conference on Learning Representations (ICLR).
- 9) Wang, Z., Schaul, T., Hessel, M., Hasselt, H. van, Silver, D., & de Freitas, N. (2016) ‘Dueling Network Architectures for Deep Reinforcement Learning’, Proceedings of the International Conference on Machine Learning (ICML), pp. 1995–2003.