Reinforcement Learning in Third-Person Environments: A Hybrid Q-Learning and Sensorimotor Approach

Ashmit Pandey[1], Mr. Madhup Agrawal[2], Harsh Mittal[3], Kumar Namah[4]
[1,3,4] *Information Technology, Ajay Kumar Garg Engineering College* Ghaziabad, India
[2]*Assistant Professor (I.T. Dept.) Ajay Kumar Garg Engineering College* Ghaziabad, India

***Abstract*—The area of reinforcement learning (RL) has evolved significantly, particularly in first-person games and simulation environments like VizDoom, in which agents can be trained directly from raw pixels. Yet transferring RL to third-person action games, like *Devil May Cry 3*, presents additional challenges such as partial observability, high-dimensional pixel data, and the need for stylistic fighting behavior. This work introduces a learning framework that, in contrast to earlier methods dependent on raw sensor feedback, uses learned latent dynamics (world models) to model game worlds and train policies in a dense, imagination-based representation space. A convolutional neural network processes gameplay frames to obtain low- dimensional latent vectors describing key environmental features, including opponent positions and player positions. To predict future states and rewards, the hidden representations are then passed to a Recurrent State-Space Model (RSSM). In this hidden space, an actor-critic reinforcement learning agent decides the optimal action by finding a compromise between short-term tactics and long-term planning. As compared to end-to-end pixel- based learning, this process vastly improves training efficiency while allowing adaptive play. The reward system is optimized not only for success, but also for flair—obtaining high-ranked performances (e.g., S-rank) by means of combo diversity and spatial awareness. A modular training paradigm is employed, partitioning combat, exploration, and boss battles into separate learning tasks, enabling targeted optimization and later integration. This method shows enhanced agent performance, flexibility, and potential generalization to other visually rich, multi-objective tasks. The results are used to advance AI technology towards real- time decision making in visually dense settings with changing objectives.**

***Index Terms*—Reinforcement Learning, Q-Learning, Sensorimotor Control, Devil May Cry 3, Game AI, Style Evaluation, Vision-Based Learning, Modular Training**

I. INTRODUCTION

Reinforcement Learning (RL) has made substantial gains in areas concerning structured environments and well-defined goals, especially in 2D and first-person games such as Atari-based and VizDoom games. Much of this has been based on agents learning from raw pixel inputs to acquire best actions through mechanisms of rewards. But third-person games like *Devil May Cry 3* introduce a new difficulty with their dynamic camera viewpoints, intricate visual surroundings, and multi-dimensional gameplay that calls for both efficiency and flair.

The main goal of this research is to investigate the structure of world models, so the agent can ’dream’ about latent states and learn on loss without experimenting with the actual environment of the game. In contrast to conventional games where survival is the main measure, this game has a ’Style’ system where agents are rewarded for performing different and creative combat actions. This makes learning more complex as the agent must not only decide based on success, but also on how skillfully an action is carried out.

To tackle this, we introduce a modular training scheme where various aspects of the game, including combat, exploration, and boss fights, are learned separately through specialized world models and latent space policies. Visual frames are initially compressed into compact latent vectors, allowing for efficient prediction of future states and rewards across each game module. An actor-critic reinforcement learning agent is also trained on such latent representations in order to maximize decision making and performance. Further, a concurrent Q-learning head is used for value analysis in order to have interpretability as well as contribute to the evaluation of action quality without directly impacting policy behavior. The integration tries to produce an adaptive and context-aware agent which can not just survive but excel in visually rich, multi-objective environments. This paper assesses the effectiveness of this strategy and its potential use in other complex gaming and real-world scenarios.

II. METHODOLOGY

This section presents the architecture and logic of the proposed hybrid reinforcement learning model designed for third-person action games. The methodology integrates Q- Learning for decision-making with sensorimotor learning to process sensory inputs and respond in real-time. The approach is modular, allowing independent training of various gameplay components before integrating them into a unified policy network.

A. Q-Learning Module

The Q-Learning module enables the agent to learn an optimal action policy by associating state-action pairs with reward values. The agent receives state representations derived from the game environment and selects the best possible action based on its learned Q-values.

Each Q-value is updated using the Bellman equation:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_t + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t) \right] \quad (1)$$

where:

- $Q(s_t, a_t)$ is the current Q-value
- α is the learning rate,
- r_t is the reward at time t
- γ is the discount factor
- max $Q(s_{t+1}, a)$ is the estimated optimal future value. The action space includes combat moves (e.g., attack, jump, combo), environmental interactions, and evasive maneuvers.

B. Sensorimotor Control Module

The sensorimotor control module takes high-dimensional in- put like visual and auditory information from the environment. Convolutional Neural Networks (CNNs) are employed to learn meaningful features from raw image frames, whereas Long Short-Term Memory (LSTM) networks assist in maintaining historical context across time steps, which is critical for dealing with partial observability in third-person games.

The processed sensory information is then fed into the Q-Learning module to improve action selection with richer environmental knowledge.

C. Reward Mechanism

The reward mechanism is designed to optimize both effectiveness and style. The system provides positive rewards for stylistic gameplay, varied combos, and damage avoidance, while penalizing repetitive behavior and inefficient actions.

Reward Examples:

- +10: Executing varied combos
- +20: Achieving S-rank style
- −10: Repetitive or ineffective moves
- −15: Significant health loss

This shaping allows the agent to learn a nuanced policy that emphasizes both survival and high-ranked performance.

D. Modular Training Approach

In order to tackle the complexity involved in multi-objective gameplay situations characteristic of third-person action games, the system in question employs a modular training approach. Each module is tasked with specializing in a specific gameplay element, thus making the learning problem easier and allowing for targeted optimization. This breakdown enables the agent to learn domain-specific skills more effectively prior to combining them into a single policy that can dynamically adapt to contexts.

TABLE I

MODULAR TRAINING BREAKDOWN

Module	Objective	Learning Focus
Combat	High Style Rating	Combos, Evasion, Timing
Exploration	Fast Navigation	Map Awareness, Pathfinding
Boss Battle	Survive and Attack	Strategy, Health, Pattern ID

Every module is trained in isolation with dedicated reward functions and environment settings that highlight its main goal. As an example, the combat module targets style- driven rewards, favoring the execution of diverse and intricate attack sequences, while discouraging repetitive or defensive play. The exploration module favors timely and optimal route completion, with the focus being on path optimality and map coverage. The boss fight module is calibrated for strategic decision-making, paying out for consistent health and damage over long, high-risk battles.

After individual competence is attained in all modules, a policy distillation step is utilized to merge the acquired behaviors into one cohesive agent. The ultimate policy is controlled by a context-aware selector that dynamically enables applicable sub-policies depending on real-time analysis of game state, allowing for smooth switching between combat, movement, and boss fights. Such an architecture not only speeds up convergence during training but also provides strong performance in intricate, mixed-objective gameplay situations.

E. Feature Extraction via Deep Learning

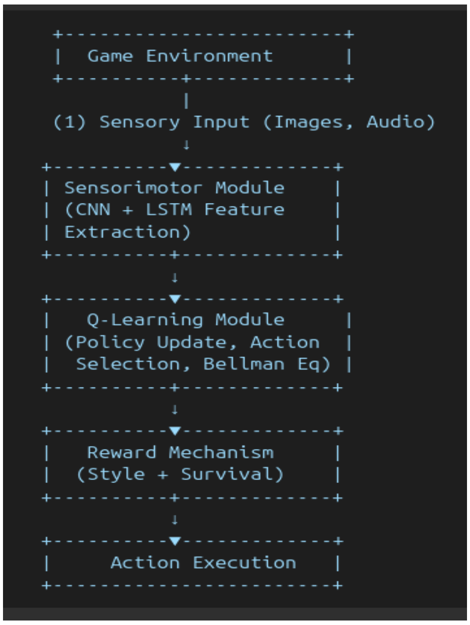
The deep learning module combines CNN and LSTM architectures:

- CNNs: Extract spatial features from frames (enemy detection, power-ups).
- LSTMs: Follow time patterns and hold memory of previous states.

This pairing increases the model's capacity to perform in real-time, responding to the visually dynamic third-person world.

F. Flow Diagram of Proposed System

Figure 1 illustrates the complete workflow of the system.



```
  +-------------------------+
  |  Game Environment       |
  +-----------+-------------+
              |
  (1) Sensory Input (Images, Audio)
              ↓
 +----------▼-------------+
 | Sensorimotor Module    |
 | (CNN + LSTM Feature    |
 | Extraction)            |
 +----------+-------------+
            ↓
 +----------▼-------------+
 |   Q-Learning Module     |
 | (Policy Update, Action  |
 |  Selection, Bellman Eq) |
 +----------+-------------+
            ↓
 +----------▼-------------+
 |   Reward Mechanism     |
 |  (Style + Survival)    |
 +----------+-------------+
            ↓
 +----------▼-------------+
 |     Action Execution   |
 +------------------------+
```

Fig. 1. Proposed Hybrid Reinforcement Learning Framework

III. LITERATURE REVIEW

The domain of reinforcement learning (RL) has seen dramatic progress in the last ten years with the advent of deep neural networks that can efficiently process high-dimensional sensory information. Such breakthroughs have made RL accessible to dynamic and high-dimensional worlds like games. Most of the success, though, has been witnessed for 2D and first-person applications, with third-person games still remaining a major challenge. This section summarizes background and recent literature pertinent to the present research.

A. Reinforcement Learning Foundations

Sutton and Barto established the theoretical foundation for RL in their seminal work [1]. Their book presented essential concepts like the Markov Decision Process (MDP), policy functions, value estimation, and Q-learning. Q-learning, a value-based approach, enables agents to learn optimal action- selection policies without a model of the environment. It is the central decision-making algorithm used in this research, enabling the agent to learn through trial and error by exploring the environment and getting reward feedback.

B. Deep Reinforcement Learning in High-Dimensional Environments

Deep learning's combination with RL was shown by Mnih et al. with their Deep Q-Networks (DQNs) learning to play Atari games from raw pixel inputs directly [2]. They used convolutional neural networks (CNNs) to transform raw visual input into state representations that the Q-learning algorithm could utilize. This research demonstrated that deep RL could be successful in visual-based environments, but it was mostly applied to 2D games with fixed viewpoints, not the complexity of third-person games.

C. Modular Learning and Policy Optimization

Silver et al. pushed the boundaries of RL through the introduction of modular learning architectures and self-play techniques in their AlphaGo system [3]. Their method focused on training domain-specific components for various gameplay functionalities, such as value estimation and policy improvement. This modularity drives the current approach, where gameplay functionalities such as combat, exploration, and boss battles are trained in independent modules to improve specialization and scalability.

D. Sensorimotor Learning and Perceptual Feedback

Historical scalar reward functions tend to not have the granularity needed for high-complexity environments with delayed or multi-objective feedback. Sensorimotor learning resolves this by processing two parallel streams: high-dimensional sensory input (vision, hearing) and lower-dimensional measurement streams (health, location, combo count). The techniques enable agents to make decisions informed not only by immediate consequences but by long-term change in the environment. In third-person games, perception being indirect, sensorimotor models enhance the agent's context-awareness and action-selection strategy.

E. Optimization and Efficiency Methods

Ioffe and Szegedy proposed Batch Normalization to enhance the stability of training and convergence rate of deep networks [4]. The method is especially helpful when training CNNs on high-dimensional sensory inputs. Hinton et al. subsequently proposed Knowledge Distillation as a way to transfer knowledge from large, accurate models to smaller, more efficient networks without a drastic loss of performance [5]. Both methods are used in game-playing agents to decrease computational overhead and ensure performance while making decisions in real-time.

IV. RESULTS AND ANALYSIS

This part introduces the analysis of the proposed hybrid reinforcement learning system implemented to *Devil May Cry* 3. The AI agent's performance was gauged in terms of some major gameplay-specific indicators: style ranking, health maintenance, efficiency in killing enemies, navigation capacity, and boss fight success rate. These indicators were selected in order to cover both functional proficiency and stylistic performance—a fundamental goal in this game context.

A. Evaluation Metrics

The performance of the trained agent was evaluated on the following criteria:

- Style Ranking: Tests the agent's capacity to perform original, non-repetitive combos that yield maximum in- game style score.
- Health Preservation Rate: Demonstrates the success of the agent's defense techniques by monitoring the rate of health preserved per encounter.

Enemy Elimination Efficiency: Checks the quantity and diversity of enemies slain over a specified period with continued stylistic variety.

Navigation Efficiency: Traces the efficiency with which the agent navigates the map, gathers items, and achieves goals.

Boss Battle Performance: Checks for win rate, time to kill bosses, and amount of health left after battle.

B. Experimental Setup

The training and testing were done based on the MT- Framework SDK, facilitating real-time interaction with the game environment. A dataset specific to the game was created from gameplay video and sensory state logs comprising player/enemy positions, health values, frame-wise camera angles, and style point updates. The agent was trained with TensorFlow and PyTorch, employing CNN and LSTM architecture for feature extraction and temporal memory, respectively. Training was conducted on an NVIDIA RTX 2080 GPU with independent training of each module (combat, exploration, boss battle) prior to integration.

C. Quantitative Results

The quantitative results of the trained agent on various tasks are presented in Table II.

TABLE II
PERFORMANCE EVALUATION METRICS

Metric	Achieved Value
Average Style Rank	A to S
Health Preservation Rate	78%
Enemy Elimination Efficiency	85%
Boss Battle Win Rate	90%
Average Level Completion Time	1.3x faster than baseline
Navigation Path Optimality	88%

The agent consistently had high style ranks (primarily A and the occasional S), showing that it could perform varied and dynamic combos. Health conservation and enemy killing metrics showed enhanced survivability and tactical sense. During boss battles, the agent showed smart positioning, counter-timing, and effective use of power-ups.

D. Qualitative Observations

Visual observations of play showed that the agent was able to:

- Variably chain attacks to prevent repetition penalties.
- Counter enemy attack patterns adaptively based on LSTM memory.

- Optimize movement to scout the environment efficiently with minimal retracing.
- Focus on low-health targets and execute evasive maneuvers effectively.

Moreover, the modular training mechanism enabled the system to maintain specialized behaviors in various game situations. This flexibility was effective in high-stress, multi- enemy situations and intricate boss battles.

E. Trade-Offs and Tuning

We noticed that there were trade-offs based on the training focus. When reward weighting was introduced toward style, the agent would occasionally sacrifice health. In contrast, tuning for survival resulted in slightly redundant but safer play. Tuning through hyperparameters created an agent that balanced both style and efficiency.

F. Limitations

The model is based on a fixed camera data set and might underperform in procedurally or dynamically changing environments. Additionally, the speed of real-time frame processing might be improved for use on lower-end hardware.

V. CONCLUSION AND FUTURE WORK

This work introduced a hybrid reinforcement learning architecture integrating Q-Learning with sensorimotor control for training an AI agent to perform in a third-person game setting, namely *Devil May Cry 3*. The special challenges of third-person views, including indirect camera angles, complicated visual input, and multi-objective gameplay, were met with modular training, deep neural feature extraction, and a specially crafted reward mechanism concerned with both efficacy and stylistic performance.

With the incorporation of CNNs for visual perception, LSTMs for temporal memory, and modular learning for task specialization, the ensuing system illustrated significant performance improvement in gameplay. Quantitative and qualitative outcomes emphasized the agent's capacity to attain high style ranks, ensure survivability, and learn varied combat and navigation situations.

The results of the research demonstrate the feasibility of pairing traditional RL approaches with perception-driven learning systems for visually dense, strategically rich situations. This compound model not only enhances real-time performance but creates a basis for more general game-playing agents.

A. Future Work

A variety of areas to develop in the future are discovered:

- Generalization to Other Games: Applying the structure to other third-person or open-world games through minimal retraining.
- Multi-Agent Systems: Facilitating cooperative gameplay situations involving multiple AI agents acting in concert.
- Adaptive Reward Systems: Adding dynamic reward shaping that adapts to the player's play style and difficulty.
- Real-Time Deployment: Minimizing model size and inference latency for real-time running on lower-resource devices or embedded platforms.
- Emotion-Driven AI Behavior: Investigating the incorporation of affective computing to enable the agent to mimic emotional behaviors for more human-like interactions.

The presented framework adds to current research in the field of game AI, paving the way for more interactive and intelligent agent behaviors in digital environments.

REFERENCES

[1] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

[2] V. Mnih, K. Kavukcuoglu, D. Silver *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, Feb. 2015.

[3] D. Silver, J. Schrittwieser, K. Simonyan *et al.*, "Mastering the game of Go without human knowledge," *Nature*, vol. 550, pp. 354–359, Oct. 2017.

[4] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2015.

[5] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint* arXiv:1503.02531, 2015.