GANS: Genetic Algorithm and Neural Network Integration for Optimal Brain Selection in Snake Game

Bambang Pudjoatmodjo[1], Mugi Praseptiawan[2*], Ulka Chandini Pendit[3], Rusnida Romli[4]

[1]Multimedia Engineering Technology, School of Applied Science, Telkom University (Indonesia)
[2]Informatics Engineering, Faculty of Industrial Technology, Institut Teknologi Sumatera, (Indonesia)
[3]Department of Computing, Sheffield Hallam University, Sheffield S1 1WB (United Kingdom)
[4]Computer and Communication Engineering, Universiti Malaysia Perlis (Malaysia)

* Corresponding author: bpudjoatmodjo@telkomuniversity.ac.id

International Jornal of Informatics and Computing

Received 26 September 2025 | Accepted 23 October 2025 | Early Access 30 November 2025

Abstract

Snake games have emerged as an engaging subject in artificial intelligence and optimization research due to the growing interest in developing autonomous agents capable of controlling the snake intelligently. This study presents a hybrid approach by integrating a Genetic Algorithm (GA) with a Neural Network (NN) to enhance the snake game's performance, effectively forming an adaptive and intelligent control system or "brain." In this framework, the Snake game is modeled as an optimization problem, where the GA is employed to optimize the parameters of the NN to improve the decision-making process of the snake. The GA operates by evolving a population of individuals each representing a set of strategies through selection, crossover, and mutation. These operations are iteratively applied to discover optimal solutions within the vast parameter space. The integrated neural network enables the snake to make real-time decisions based on environmental stimuli, enhancing its survival and goal-seeking behavior. Fitness evaluation is performed based on everyone's gameplay performance, where the most successful individuals contribute to the next generation. Experimental results demonstrate that the combination of GA and NN significantly improves snake gameplay performance. The fitness score acts as a performance indicator, showing that higher-generation populations tend to yield better results. For instance, snakes trained over 100 generations achieved scores around 8, while those trained over 500 generations exceeded scores of 15. This confirms the effectiveness of evolutionary optimization in training neural networks for game-based AI tasks.

Keywords

Snake Game,
Genetic Algorithm,
Neural Network,
Optimization,
Artificial Intelligence

I. Introduction

In recent years, the integration of artificial intelligence algorithms into game environments has gained significant attention as an experimental platform for testing optimization and adaptive learning techniques [1]. Among classical benchmark problems, the Snake Game represents a sequential decision-making task within a discrete and dynamically expanding state space [2]. The progressive growth of the snake introduces increasing environmental complexity, making the game suitable for evaluating evolutionary and neural-based learning models [3].

Genetic Algorithms (GA) have been widely adopted to solve complex optimization problems characterized by non-linear and high-dimensional search spaces [4]. Inspired by natural selection principles, GA iteratively improves candidate solutions through selection, crossover, and mutation operators [5]. Within the snake game context, GA evolves movement strategies and behavioural parameters by encoding them as chromosomes and evaluating their fitness based on performance scores [6]. Prior studies have demonstrated that GA based controllers outperform heuristic methods in terms of consistency and adaptability [7].

Despite its strengths, standalone GA suffers from limitations in parameter initialization, which is typically random and lacks structural learning. Pure random initialization may enhance exploration but often results in slower convergence and unstable early-generation performance. Consequently, integrating Neural Networks (NN) into the evolutionary framework has emerged as a promising approach to enhance parameter representation before evolutionary selection [8].

Neural Networks function as universal function approximators capable of modelling non-linear relationships between environmental inputs and action outputs [9]. In hybrid approaches, GA operates as an evolutionary optimizer of NN weights, while NN serves as the decision-making module [10]. Such integration has demonstrated significant improvements across domains, including software effort

Please cite this article as: Pudjoatmodjo, P., Praseptiawan, M., Chandini Pendit, P., & Rusnida, R. (2025). GANS: Genetic Algorithm and Neural Network Integration for Optimal Brain Selection in Snake Game. International Journal of Informatics and Computing, 1 (2), 82-89.

estimation with accuracy improvements of up to 91.3% [11]. By introducing NN as a structured parameter generator prior to selection, the solution space becomes more organized compared to purely random initialization.

Although GA–NN hybridization has been explored in the Snake Game domain, several research gaps remain. First, previous studies often limit evolutionary cycles to fewer than 500 generations due to convergence computational constraints [12]. Second, many implementations focus solely on fitness values without providing graphical user interface (GUI)-based behavioural visualization, which is essential for qualitative performance evaluation. Third, the impact of large-scale generational expansion (>500 generations) on convergence stability and performance scaling has not been systematically analysed.

To address these gaps, this research proposes a hybrid framework integrating a Genetic Algorithm with a Double Hidden Layer Neural Network (DHLNN) as the snake's brain architecture. The main contributions are summarized as follows: (1) Integration of NN as a structured parameter filtering mechanism prior to GA-based evolution. (2) Experimental evaluation of generational scaling up to 1000 generations to analyse performance trends and convergence behaviour. (3) Implementation of an interactive GUI to enable real-time behavioural observation alongside quantitative fitness analysis.

II. Related Work

Genetic Algorithms (GA) have been extensively applied to game-based optimization problems due to their robustness in navigating complex and non-convex search spaces [13]. In the context of the Snake Game, GA has been utilized to optimize movement heuristics, speed adaptation, and food-search strategies through fitness-based evolutionary selection [14]. These approaches encode behavioral parameters as chromosomes and evaluate individuals based on scoring functions such as smoothness, spatial efficiency, and food acquisition rate [15]. Several studies report that GA-based agents demonstrate higher consistency compared to rule-based or heuristic controllers, particularly in stochastic environments. The effectiveness of GA strongly depends on crossover and mutation rates, which directly influence exploration exploitation balance [16]. However, parameter initialization remains largely random, which may lead to slow convergence and instability in early generations.

Neural Networks (NN) have been widely adopted in game intelligence due to their capacity to approximate non-linear decision boundaries [17]. In Snake Game implementations, NN typically functions as a controller that maps environmental sensory inputs (food direction, obstacle proximity, and self-collision distance) to discrete movement actions [18]. Research has demonstrated that deeper architectures, such as Double Hidden Layer Recurrent Neural Networks (DHLRNN), can improve convergence speed and generalization capability compared to single hidden-layer models [19]. The inclusion of feedback mechanisms enables associative memory and dynamic state modeling, which are beneficial for sequential decision-making tasks. However, pure NN-based approaches often require extensive training iterations and may suffer from local minima or overfitting when training data diversity is limited. Furthermore, gradient-based learning may become inefficient in highly dynamic and non-stationary environments such as Snake Game, where the state space continuously expands.

To address the individual limitations of GA and NN, hybrid evolutionary neural frameworks have been proposed. In such models, GA typically serves as an optimizer for NN weights, while NN operates as the decision-making core [20]. This hybridization enhances exploration through evolutionary mechanisms while preserving structured representation via neural modeling [21]. Hybrid GA–NN models have shown promising results across various domains. For instance, [11] reported accuracy improvements of 8.9% in software test effort estimation using GA-optimized neural networks. Similarly, demonstrated that NN-generated parameters, when evolved via GA, yield improved stability compared to purely random initialization. In Snake Game research, GA–NN integration has been implemented to evolve movement strategies. However, prior studies exhibit several constraints. First, generational depth is often restricted (<500 generations) due to convergence instability or computational cost. Second, evaluation metrics are frequently limited to fitness values without behavioral visualization, reducing interpretability. Third, systematic investigation of large-scale generational expansion and its effect on performance scalability remains underexplored.

III. Methodology

Figure 1 below illustrates the schematic of the method employed for snake game analysis in this research.

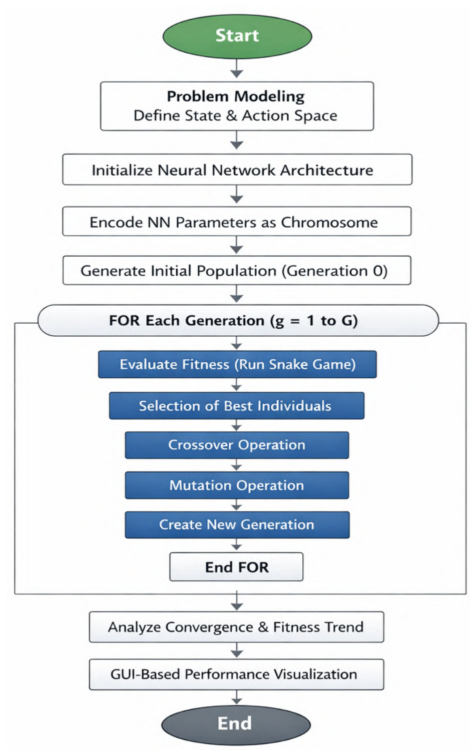

Fig. 1. Research Methodology

A. Problem Modeling

At each time step t, the environment state is represented as:

$$S_t = \{x_h, y_h, x_f, y_f, B_t\} \quad (1)$$

Where:

(x_h, y_h)= snake head position,

(x_f, y_f)= food location,

B_t = body and obstacle configuration.

The action space is defined as:

$$a_t \in = \{up, down, left, right\} \quad (2)$$

The objective is to maximize the fitness function:

$$\frac{max}{\theta} F(\theta) \quad (3)$$

Where θ represents NN parameters optimized by GA.

B. Neural Network Architecture

The brain of each snake agent is modeled using a Double Hidden Layer Neural Network:

$$h_1 = \sigma(W_1 S_t + b_1) \quad (4)$$

$$h_2 = \sigma(W_2 S_1 + b_2) \quad (5)$$

$$o_t = W_3 h_2 + b_3) \quad (6)$$

The action is selected as:

$$a_t = \arg max\,(o_t) \quad (7)$$

The full parameter set $\theta = W_1, W_2, W_3, b_1, b_2, b_3$ is encoded as a chromosome in the GA population.

C. Genetic Algorithm Optimization

Employed to optimize the neural network parameters representing the snake agent's brain. Everyone in the population encodes the full set of network weights and biases as a chromosome. The optimization process begins with random population initialization, followed by fitness evaluation based on game performance. Individuals with higher fitness are selected probabilistically to reproduce. New offspring are generated through crossover, which combines parameters from two parents, and mutation, which introduces small random perturbations to maintain diversity. This evolutionary cycle is repeated across generations, progressively improving the agent's decision-making performance and maximizing the fitness function.

D. Evolutionary Cycle

The evolutionary process is iteratively executed for G generations. In each generation, individuals are evaluated, selected, and reproduced through crossover and mutation to form a new population. This cycle continues until the maximum generation G is reached, progressively improving the fitness performance of the agent.

E. GUI-based Performance Visualization

A real-time graphical display of the Snake agent while it plays, used to observe behavior (movement path, food seeking, obstacle avoidance) and performance indicators (score/fitness, generation ID, step count). It complements numerical fitness logs by making strategy quality and failure patterns directly interpretable. Additionally, the GUI facilitates behavioral comparison across generations, supports debugging of evolutionary instability, and enables qualitative validation of convergence trends beyond purely statistical evaluation metrics.

IV. Result and Analysis

A. Experimental Overview

The experimental evaluation was conducted to analyze the performance scalability of the proposed hybrid GA–NN framework under varying generational depths. Four generational configurations were tested: 10, 100, 500, and more than 500 generations (up to 600–1000). Performance was evaluated using maximum fitness score, convergence behavior, stability, and computational time. All experiments were implemented in Python using a custom evolutionary training loop integrated with a real-time GUI visualization module.

B. Performance Across Generations

At 10 generations, the system demonstrated early-stage adaptive behavior. The highest recorded score was 6, indicating that the agent began developing basic food-seeking and collision-avoidance strategies. However, movement patterns remained unstable and suboptimal, reflecting limited evolutionary exploration. This stage represents initial exploration with insufficient evolutionary pressure for stable convergence.

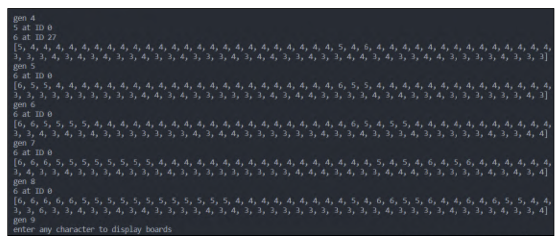

Fig. 2. The highest score is 6

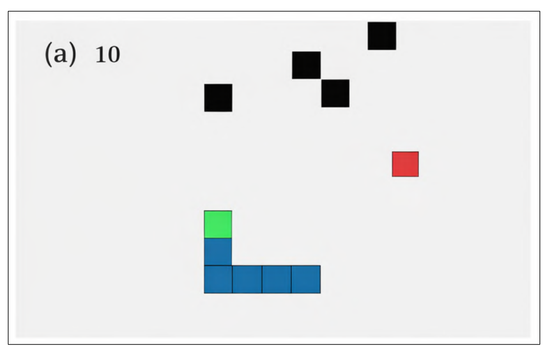

Fig. 3. GUI of snake game with 10 generations

From an evolutionary optimization perspective, insufficient iteration restricts both exploitation of promising solutions and adequate mutation-driven diversity. Consequently, movement trajectories remain inconsistent and locally reactive rather than strategically optimized. The instability observed at this stage aligns with theoretical expectations of early-generation dynamics, where exploration dominates over convergence, and fitness variance across individuals remains relatively high.

Therefore, the results at 10 generations represent an initial adaptive response rather than a stable convergence state, confirming that greater generational depth is required to achieve robust evolutionary refinement.

When extended to 100 generations, the maximum score increased to 8. Behavioral observation via GUI indicated improved navigation consistency and reduced premature collision frequency. The increase suggests that evolutionary selection and mutation began reinforcing beneficial parameter combinations. Nevertheless, convergence remained moderate, as strategy refinement was still evolving.

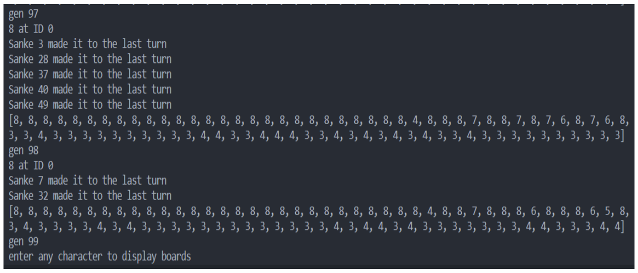

Fig. 4. The highest score is 8

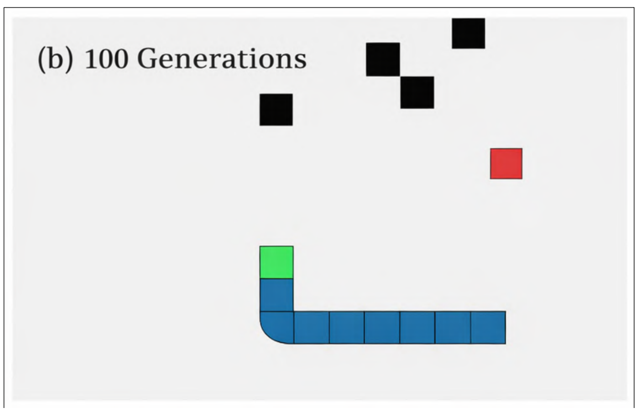

Fig. 5. GUI of snake game with 100 generations

From an evolutionary dynamic's standpoint, the interaction between crossover and mutation operators appears to have strengthened exploitation of high-performing individuals while maintaining sufficient diversity for continued exploration. However, the convergence pattern at this stage remains moderate. Fitness growth, although consistent, does not yet exhibit full stabilization, indicating that the search process is still navigating the parameter landscape and refining strategic decision-making structures. Thus, the 100-generation configuration represents a transitional phase between exploratory adaptation and structured convergence, where evolutionary mechanisms begin to produce coherent behavioral optimization but have not yet reached peak performance efficiency.

At 500 generations, a significant improvement was observed, with the maximum score reaching 17. The agent demonstrated: (1) More strategic food targeting, (2) Efficient space utilization, (3) Improved avoidance of self-collision. The convergence curve showed a steeper upward trend during early generations followed by gradual stabilization. This indicates that the hybrid GA–NN framework effectively balances exploration (mutation) and exploitation (selection).

The convergence trajectory exhibits a rapid fitness acceleration during earlier generations, followed by a gradual stabilization trend. Such a pattern is theoretically consistent with evolutionary optimization dynamics, where early exploration identifies promising regions of the search space, and subsequent exploitation intensifies refinement within those regions. The stabilization phase suggests reduced variance among top-performing individuals, indicating increasing population homogeneity around near-optimal solutions.

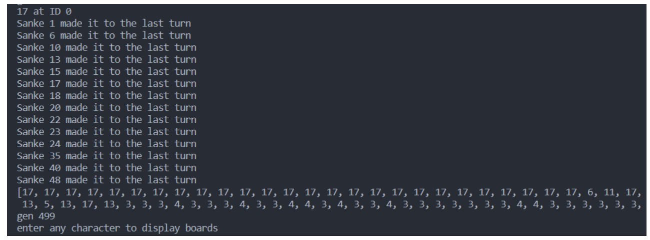

Fig. 6. The highest score is 17

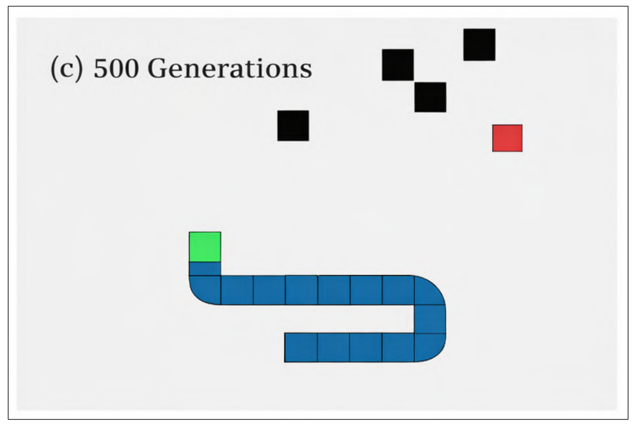

Fig. 7. GUI of snake game with 500 generations

The observed behavior demonstrates that the hybrid GA–NN framework effectively balances mutation driven exploration with selection-driven exploitation. Neural parameter encoding provides structured decision representation, while evolutionary operators progressively optimize weight configurations. At this generational depth, the system transitions from reactive movement patterns to adaptive, strategy-oriented navigation, marking a strong convergence phase within the evolutionary learning process.

Further expansion to 600 generations produced a maximum score of 20, demonstrating continued performance improvement. However, marginal gain per additional generation began to decrease, indicating the onset of diminishing returns. Beyond approximately 1000 generations, no further objective improvement was observed, suggesting convergence saturation and increased computational redundancy.

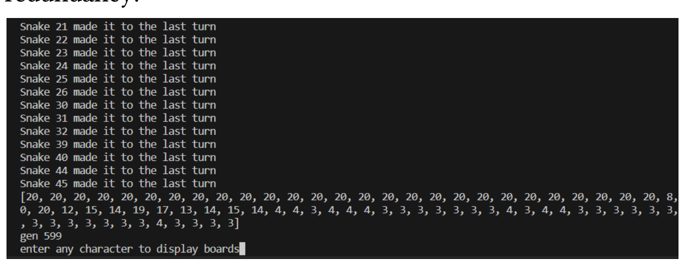

Fig. 8. The highest score is 20

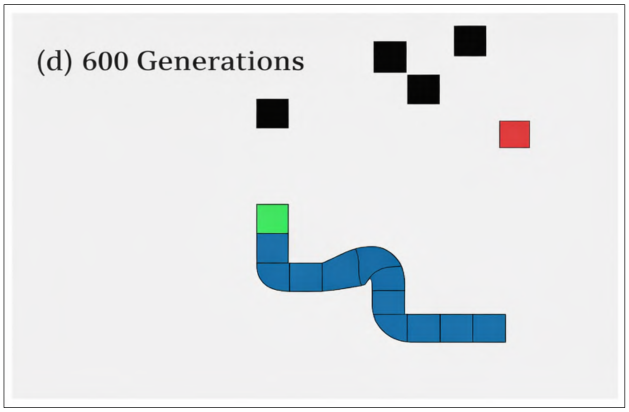

Fig. 9. GUI of snake game with 600 generations

Beyond approximately 1000 generations, no observable improvement in objective fitness was detected. This plateau suggests convergence saturation, where the evolutionary process reaches a stable local optimum. At this point, continued iterations primarily increase computational cost without yielding proportional performance benefits, leading to computational redundancy. The observed saturation aligns with theoretical convergence behavior in evolutionary algorithms, where fitness variance stabilizes and improvement rates asymptotically approach zero. These findings highlight an important trade-off between generational depth and computational efficiency, emphasizing the need for adaptive stopping criteria or dynamic mutation strategies in large-scale evolutionary neural frameworks.

C. Convergence Analysis

The evolutionary fitness trajectory observed in this study exhibits a characteristic three-phase convergence pattern, consistent with theoretical models in evolutionary computation and adaptive search dynamics.

Exploration Phase (0–100 Generations), the fitness curve demonstrates rapid fluctuations and high variance across individuals. This instability is expected, as the population begins from randomly initialized neural parameters. At this stage, mutation plays a dominant role, generating broad diversity within the search space. Selection pressure is still insufficient to consistently propagate high-performing genetic structures. From a search-theoretic perspective, the algorithm is primarily engaged in global exploration, sampling diverse regions of the parameter landscape. Because beneficial neural weight configurations are sparse in high-dimensional space, early convergence remains unstable and reactive. Behavioral patterns observed through GUI confirm this dynamic, where movement strategies appear locally adaptive but lack long-term planning consistency. However, unlike purely random GA initialization, the incorporation of structured parameter encoding via Neural Network architecture constrains the search within functionally meaningful decision boundaries. Even though weights are randomly initialized, the network structure itself enforces representational coherence, reducing chaotic search behavior. This structural prior contributes to relatively faster emergence of basic adaptive behavior compared to flat, non-structured chromosome encoding.

Acceleration Phase (100–500 Generations), between 100 and 500 generations, the convergence curve exhibits a steeper upward slope, indicating significant performance gains. This phase represents a transition from exploration-dominant dynamics toward balanced exploration–exploitation interaction. Selection pressure becomes more effective in amplifying advantageous neural weight combinations. Crossover recombines high-fitness structures, while mutation introduces controlled perturbations that refine decision boundaries. As a result, fitness variance across the population begins to decrease, and high-performing individuals become increasingly dominant. In optimization theory, this phase corresponds to intensification within promising regions of the solution space. The search is no longer uniformly random; instead, it concentrates around subspaces with high reward density. Neural parameter refinement leads to Improved directional prediction accuracy, reduced collision frequency, enhanced spatial efficiency. The acceleration phase demonstrates that the hybrid GA–NN framework successfully transforms random search into structured adaptive learning.

Stabilization Phase (>500 Generations), beyond 500 generations, the fitness curve begins to plateau. Although incremental improvements still occur, the slope decreases significantly. This behavior reflects convergence toward a near-optimal equilibrium. At this stage, population diversity diminishes as high-performing individuals dominate selection. Mutation introduces variations, but most new configurations fall within a narrow fitness band. The probability of discovering substantially superior parameter configurations decreases asymptotically. Importantly, the plateau does not imply algorithmic failure, but rather convergence saturation. The search space has been sufficiently exploited, and further iteration primarily increases computational cost without proportional performance gain.

D. Computational Trade-Off

The computational complexity of the proposed hybrid GA–NN framework grows proportionally with generational depth and can be approximated as $O(G\ x\ n\ x\ T)$.

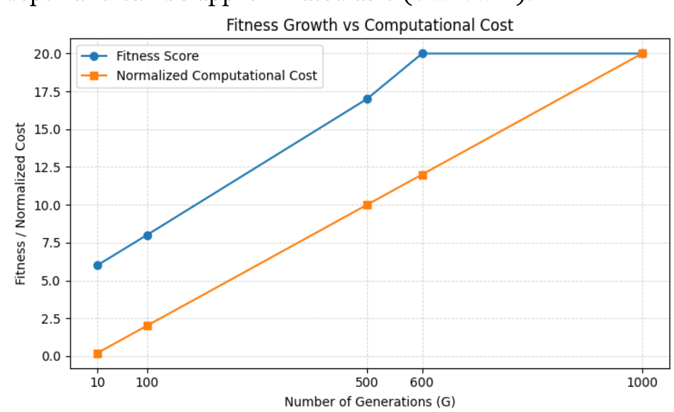

Fig. 10. Fitness Growth vs Computational Cost

The Figure 10 illustrates the relationship between fitness improvement and computational cost across increasing generations. Fitness rises significantly from 10 to 600 generations, showing effective evolutionary learning. However, after approximately 600 generations, the fitness curve plateaus, indicating convergence saturation. In contrast, computational cost continues to increase linearly with generational depth. This demonstrates a diminishing return effect, where additional computational effort yields minimal performance gain beyond the stabilization phase.

E. Behavioral Validation via GUI

While quantitative fitness values provide objective performance indicators, GUI-based visualization enables deeper behavioral validation of the evolved agent. Across generational scaling, visual inspection revealed a clear transformation in movement dynamics and spatial decision-making patterns. At lower generations, snake trajectories appeared reactive and locally driven, often responding only to immediate food direction without anticipating spatial constraints. Movement discontinuity and frequent self-collision reflected incomplete decision boundary formation within the neural parameter space. As generational depth increased, observable behaviors evolved from reactive to strategic. At 500+ generations, the snake demonstrated: (a) Smooth trajectory curvature instead of abrupt directional shifts, (b) Proactive obstacle anticipation, (c) Body alignment efficiency minimizing self-trapping probability, (d) Spatial path planning beyond immediate food direction.

This qualitative improvement indicates that neural weight optimization has successfully encoded structured decision representations rather than memorized reactive responses. Importantly, GUI validation provides evidence that fitness gains are not merely numerical artifacts but correspond to genuine behavioral sophistication. The smoother trajectories and longer survival durations confirm that parameter refinement improves environmental awareness and long-term reward optimization. Thus, GUI observation functions as a qualitative cross-validation layer, reinforcing statistical findings and strengthening the internal validity of the evolutionary learning outcome.

F. Comparative Interpretation

The cumulative experimental evidence supports several theoretical interpretations. The hybrid framework consistently demonstrates superior performance as generational depth increases. Early-stage GA–NN configurations show limited adaptation; however, structured neural encoding enables progressive refinement that pure random GA encoding would struggle to achieve within the same iteration depth. This confirms that the integration of representational learning (NN) with global search (GA) produces synergistic optimization dynamics. Fitness growth is not linear but follows a characteristic convergence trajectory. Increasing generations enhances: (a) Parameter refinement depth, (b) Stability of high-performing individuals, (c) Reduction in fitness variance. This suggests that evolutionary pressure requires sufficient iteration cycles to effectively exploit promising regions in high-dimensional parameter space. The use of neural network architecture constrains the search within meaningful functional mappings. This structural constraint reduces chaotic exploration and accelerates emergence of coherent behavior, especially during early generations (0–100). Unlike flat chromosome encoding, NN representation embeds inductive bias that guides evolutionary search toward behaviorally interpretable solutions. Beyond ~600 generations, incremental fitness gain diminishes despite continued computational investment. This indicates population homogenization, reduced genetic diversity, local optimum stabilization. This saturation does not indicate failure but rather algorithmic maturity. However, it introduces computational trade-offs, emphasizing the need for adaptive stopping criteria or dynamic mutation scheduling, summary finding as shown in Table I.

TABLE I. Comparative Analysis

Generations	Max Score	Performance Trend
10	6	Early exploration
100	8	Moderate improvement
500	17	Strong convergence
600	20	Diminishing return

TABLE II. Convergence and Performance Summary

Generational Range	Max Score	Convergence Phase	Interpretation
0–10	6	Exploration	Early adaptive behaviour with unstable performance
10–100	8	Early Acceleration	Selective pressure begins improving quality
100–500	17	Strong Acceleration	Effective refinement and rapid fitness growth
500–600	20	Stabilization	Near-optimal convergence with diminishing gain
>600–1000	~20	Saturation	Performance plateau and equilibrium state

V. Conclusion

This research proposed a hybrid evolutionary–neural framework integrating a Genetic Algorithm (GA) with a Double Hidden Layer Neural Network (DHLNN) for optimizing decision-making behavior in the Snake Game environment. The framework models the problem as a dynamic optimization task in which neural parameters evolved through iterative genetic operations to maximize cumulative fitness. Experimental results demonstrate that generational depth plays a critical role in convergence quality and behavioral sophistication. The convergence trajectory follows a three-phase pattern: (1) exploration phase characterized by unstable adaptive behavior, (2) acceleration phase marked by rapid fitness improvement through effective parameter refinement, and (3) stabilization phase in which marginal gains diminish as the system approaches convergence equilibrium. The highest performance was achieved at approximately 600 generations, after which further iteration yielded negligible improvement while computational cost continued to increase linearly. Behavioral validation through GUI visualization confirmed that fitness improvement corresponds to genuine strategic enhancement rather than numerical artifacts. Higher generations produced smoother trajectories, improved spatial awareness, and reduced collision frequency, indicating successful neural parameter optimization. From a computational standpoint, the framework exhibits time complexity. While scalability improves solution quality, convergence saturation introduces a computational trade-off. Therefore, an optimal generational threshold exists approximately 500–600 generations in this research where performance gain and computational efficiency are balanced. The findings confirm that structured neural encoding enhances early-stage convergence stability compared to purely random GA initialization. The hybrid GA–

NN architecture effectively balances exploration and exploitation dynamics within a high-dimensional parameter space. Future research may explore adaptive mutation scheduling, dynamic stopping criteria, elitism strategies, or integration with reinforcement learning to further improve convergence efficiency and computational performance. Additionally, extending the framework to more complex environments may provide deeper insight into scalability and generalization capabilities.

References

[1] F. Naseer, M. N. Khan, A. Addas, Q. Awais, and N. Ayub, 'Game Mechanics and Artificial Intelligence Personalization: A Framework for Adaptive Learning Systems', *Education Sciences*, vol. 15, no. 3, p. 301, Feb. 2025. DOI: 10.3390/educsci15030301

[2] Md. R. R. Tushar and S. Siddique, 'A Memory Efficient Deep Reinforcement Learning Approach For Snake Game Autonomous Agents', in *2022 IEEE 16th International Conference on Application of Information and Communication Technologies (AICT)*, 2022, pp. 1–6. DOI: 10.1109/AICT55583.2022.10013603

[3] S. M. K. Sarkhi and H. Koyuncu, 'Optimization Strategies for Atari Game Environments: Integrating Snake Optimization Algorithm and Energy Valley Optimization in Reinforcement Learning Models', *AI*, vol. 5, no. 3, pp. 1172–1191, Jul. 2024. DOI: 10.3390/ai5030057

[4] D.-A. Popescu, 'An Enhanced Genetic Algorithm for Optimized Educational Assessment Test Generation Through Population Variation', *Big Data and Cognitive Computing*, vol. 9, no. 4, p. 98, Apr. 2025. DOI: 10.3390/bdcc9040098

[5] M. A. Albadr, S. Tiun, M. Ayob, and F. AL-Dhief, 'Genetic Algorithm Based on Natural Selection Theory for Optimization Problems', *Symmetry*, vol. 12, no. 11, p. 1758, Oct. 2020. DOI: 10.3390/sym12111758

[6] M. Kotyrba, E. Volna, H. Habiballa, and J. Czyz, 'The Influence of Genetic Algorithms on Learning Possibilities of Artificial Neural Networks', *Computers*, vol. 11, no. 5, p. 70, Apr. 2022. DOI: 10.3390/computers11050070

[7] K. R. Venkatesan and K. Ravi, 'An advanced metaheuristic algorithm-based control technique for frequency regulation of grid-connected PV system', *Results in Engineering*, vol. 29, p. 108899, Mar. 2026. DOI: 10.1016/j.rineng.2025.108899

[8] N. Xie, J. He, W. Wei, J. Ba, and T. Yang, 'Neural network-driven genetic algorithm for advanced fuel cell performance enhancement', *Results in Engineering*, vol. 29, p. 108426, Mar. 2026. DOI: 10.1016/j.rineng.2025.108426

[9] A. Pasini and S. Amendola, 'A Neural Modelling Tool for Non-Linear Influence Analyses and Perspectives of Applications in Medical Research', *Applied Sciences*, vol. 14, no. 5, p. 2148, Mar. 2024. DOI: 10.3390/app14052148

[10] M. Gavrilescu, S.-A. Floria, F. Leon, and S. Curteanu, 'A Hybrid Competitive Evolutionary Neural Network Optimization Algorithm for a Regression Problem in Chemical Engineering', *Mathematics*, vol. 10, no. 19, p. 3581, Sep. 2022. DOI: 10.3390/math10193581

[11] V. Chahar and P. K. Bhatia, 'Performance Analysis of Software Test Effort Estimation using Genetic Algorithm and Neural Network', *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 10, 2022. DOI: 10.14569/IJACSA.2022.0131045

[12] Y. Meraihi *et al.*, 'A Comprehensive Review of Archimedes Optimization Algorithm with its Theory, Variants, Hybridization, and Applications', *Archives of Computational Methods in Engineering*, Nov. 2025. DOI: 10.1007/s11831-025-10451-0

[13] V. S. Bisht, N. Joshi, G. S. Jethi, and A. S. Bhakuni, 'A Review on Genetic Algorithm and Its Application in Power System Engineering', 2021, pp. 107–130. DOI: 10.1007/978-981-15-7571-6_5

[14] Y. Yang, H. Chen, A. A. Heidari, and A. H. Gandomi, 'Hunger games search: Visions, conception, implementation, deep analysis, perspectives, and towards performance shifts', *Expert Systems with Applications*, vol. 177, p. 114864, Sep. 2021. DOI: 10.1016/j.eswa.2021.114864

[15] M. Dehvari, C.-K. Yang, and E. Armando, 'Mental Map-Preserving Visualization through a Genetic Algorithm', *Applied Sciences*, vol. 11, no. 10, p. 4336, May 2021. DOI: 10.3390/app11104336

[16] S. Katoch, S. S. Chauhan, and V. Kumar, 'A review on genetic algorithm: past, present, and future', *Multimedia Tools and Applications*, vol. 80, no. 5, pp. 8091–8126, Feb. 2021. DOI: 10.1007/s11042-020-10139-6

[17] R. Saleem, B. Yuan, F. Kurugollu, A. Anjum, and L. Liu, 'Explaining deep neural networks: A survey on the global interpretation methods', *Neurocomputing*, vol. 513, pp. 165–180, Nov. 2022. DOI: 10.1016/j.neucom.2022.09.129

[18] P. Yuhang, S. Yiqi, M. Qianli, G. Bowen, D. Junyi, and T. Zijun, 'Playing the Snake Game with Reinforcement Learning', *Cambridge Explorations in Arts and Sciences*, vol. 1, no. 1, Jul. 2023. DOI: 10.61603/ceas.v1i1.13

[19] I. D. Mienye and T. G. Swart, 'A Comprehensive Review of Deep Learning: Architectures, Recent Advances, and Applications', *Information*, vol. 15, no. 12, p. 755, Nov. 2024. DOI: 10.3390/info15120755

[20] M. Al-Aghbari and A. M. Gujarathi, 'Hybrid optimization approach using evolutionary neural network & genetic algorithm in a real-world waterflood development', *Journal of Petroleum Science and Engineering*, vol. 216, p. 110813, Sep. 2022. DOI: 10.1016/j.petrol.2022.110813

[21] H. A. Hurtado-Mora, L. A. Herrera-Barajas, L. J. González-del-Ángel, R. Pichardo-Ramírez, A. H. García-Ruiz, and K. E. Lira-García, 'Enhancing Neural Network Training Through Neuroevolutionary Models: A Hybrid Approach to Classification Optimization', *Mathematics*, vol. 13, no. 7, p. 1114, Mar. 2025. DOI: 10.3390/math13071114

Bambang Pudjoatmodjo

He has been working as a lecturer since 2002. At the beginning of his teaching career, he taught various subjects. In early 2011, he began studying film screenwriting. He then went on to study screenwriting for the development of digital games and interactive devices such as virtual reality and augmented reality. In 1998, the author earned a bachelor's degree in computational mathematics from the Bandung Institute of Technology. Then, in 2014, the author obtained a master's degree in informatics in the field of Software Reliability from Langlangbuana University. And in 2024, he obtained a doctorate from the Technical University of Malaysia Melaka (UTeM) in the field of Serious Games (Feedback-Based Evaluation Simulation). The author can be contacted via email: bpudjoatmodjo@telkomuniversity.ac.id and at the Center of Excellence for Human Centric Engineering, Institute of Sustainable Society, Telkom

University, Main Campus (Bandung Campus), Jl. Telekomunikasi no. 1, Bandung 40257, West Java, Indonesia.

Mugi Praseptiawan

He is a lecturer and researcher affiliated with the Institut Teknologi Sumatera (ITERA), Indonesia. His academic specialization lies in software quality engineering, information systems evaluation, and digital learning technologies. He is currently pursuing a doctoral degree with a research focus on developing a comprehensive software quality evaluation model for Massive Open Online Courses (MOOCs). His research interests include ISO/IEC 25010-based software quality modeling, structural equation modeling (SEM) for empirical validation, and the integration of the Information Systems Success Model (ISSM) and Expectation Confirmation Model (ECM) to examine user satisfaction and continuance intention in digital platforms. His work aims to design data-driven evaluation frameworks that improve system reliability, user experience, and the long-term sustainability of educational technologies.

Ulka Chandini Pendit

She is a lecturer and researcher at the Department of Computing, Faculty of Business, Technology and Engineering, Sheffield Hallam University, United Kingdom. She obtained her PhD (Multimedia) from UUM in 2016 and Bachelor of Multimedia (Hons) from UUM in 2011. Her research interests are in cultural heritage. Her work has won several medals in various international invention and innovation exhibitions. She is also interested in other arts fields, such as, novel writing, photography, and filmmaking.

Rusnida Romli

She is a senior lecturer at Universiti Malaysia Perlis (UniMAP) in the Faculty of Electronic Engineering & Technology (FKTEN), specializing in computer science disciplines. With extensive experience in teaching subjects like Computer Programming, Object-Oriented Programming, and Data Structures & Algorithms, Computer Graphics and Virtual Reality, she has a strong academic background in educating students across various levels. As a researcher, she has led numerous projects in areas such as Virtual Reality (VR), Augmented Reality (AR), and their applications in education, healthcare, and environmental awareness. Her research contributions are reflected in her publications, where she explores innovative technologies and their real-world implications. She has also been actively involved in supervising postgraduate and undergraduate research, contributing to the advancement of knowledge in her field.