Augmented Neat Algorithm For Enhanced Cognitive Interaction (NEAT-X)

1st Nihal Anil
Student, Dept of Computer Science Engineering
Universal Engineering College
Thrissur,Kerala
dicor577@gmail.com

2nd Ms. Nighila Abhish
Assistant Professor, Dept of Computer Science Engineering
Universal Engineering College
Thrissur,Kerala
nighila@uec.ac.in

3rd Jesila Joy
Student, Dept of Computer Science Engineering
Universal Engineering College
Thrissur,Kerala
jesilajoy@gmail.com

4th Noora Sajil
Student, Dept of Computer Science Engineering
Universal Engineering College
Thrissur,Kerala
noorasajill@gmail.com

5th Vishnuraj P R
Student, Dept of Computer Science Engineering
Universal Engineering College
Thrissur,Kerala
vraj08352@gmail.com

***Abstract*—Artificial neural networks (ANNs) are utilized in a variety of practical applications, from pattern recognition to controlling robots. Neuroevolution (NE), which involves the artificial evolution of neural networks through the use of genetic algorithms, has demonstrated significant potential in tackling complicated reinforcement learning tasks. This paper provides a comprehensive overview of the leading methods for evolving artificial neural networks (ANNs), called NeuroEvolution of Augmenting Topologies(NEAT). NEAT excels in evolving neural networks with diverse structures but faces scalability challenges, especially with extensive networks or high-dimensional input spaces. As the complexity of the problem increases, the search space expands exponentially, hindering NEAT's exploration effectiveness. After performing mutation, we identify the best mutations, and similar substructures are discovered and added to the mutation list. The improved version of NEAT algorithm requires less computational resources and will give optimized solution. After adding it to the mutation list with some minor modifications, it is demonstrated that the performance of NEAT can be improved.**

Keywords—NeuroEvolution of Augmenting Topologies,NVIDIA Isaac Sim,Artificial neural networks,Neuroevolution.

I. INTRODUCTION

Over the past five decades, scholars across various disciplines have employed models of biological neural networks not only to gain deeper insights into the functioning of biological nervous systems but also to develop robust tools for engineering purposes. Artificial neural networks (ANNs) are computational models, implemented either in software or specialized hardware, designed to emulate the behavioral and adaptive characteristics of biological nervous systems. Typically, ANNs consist of interconnected processing units, often referred to as "neurons," which can receive multiple inputs and produce outputs. Conceptually, an ANN can be represented as a directed graph, where each node corresponds to a neuron model. At its simplest, a neuron model involves calculating a weighted sum of incoming signals, which is then transformed by a non-linear transfer function. However, more advanced neuron models may incorporate discrete-time or continuous-time dynamics. The connections between neurons, represented as edges in the graph, are characterized by synaptic weights. Neurons that interact directly with the external environment are commonly known as input or output neurons [Fig. 2]. The architecture, or topology, of a neural network is defined by the arrangement of neurons and the possible connections between them.

A groundbreaking NeuroEvolution technique known as NeuroEvolution of Augmenting Topologies (NEAT) has been developed with the aim of enhancing architecture optimization by reducing the dimensionality of the search space for connection weights. NEAT achieves this by gradually expanding network topologies from their minimal forms, ultimately identifying the optimal topology with the smallest possible dimensional space. In NEAT, each neural network's genome represents a potential solution to the problem at hand. As the size of neural networks grows, the number of potential network structures and parameter combinations increases exponentially. This vast expansion of the search space makes it challenging for NEAT to effectively explore all possible solutions, resulting in longer training times and reduced chances of finding optimal solutions.As the size of neural networks increases, the search space expands exponentially, requiring more computational resources to explore all potential solutions. This leads to longer training times, higher memory usage, and increased computational costs, making it impractical or unfeasible to exhaustively search the entire solution space. After performing mutation, we identify the best mutations and discover similar substructures within the best mutations using graph coloring algorithms.Then, the similar substructures are compressed and added to the mutation list one by one.By compressing substructures, we can reduce the overall structural complexity of the network.For evaluating the performance of the improved NEAT, We utilize the NVIDIA Isaac Sim.

II. THEORY AND RELATED WORKS

A. *NeuroEvolution of Augmenting Topologies (NEAT)*

NEAT presents several advances in the evolution of neural networks. Through historical markings, NEAT offers a solution to the problem of competing conventions in a population with diverse topologies. This approach uniquely identifies genes and tracks them through generations, allowing NEAT to maintain consistency and comparability among different network structures. This way, NEAT can efficiently evolve neural networks with varying architectures while avoiding issues arising from topological differences between individuals in the population.In [1] it present a method which outperforms the best fixed-topology method on a challenging benchmark reinforcement learning task.NEAT's genetic encoding aims to align genes during mating, where genomes contain connection and node genes [Fig 3]. Connection genes link nodes and carry information like weight and status. Crossover combines beneficial traits from different individuals, promoting genetic diversity. Mutation alters connection weights and network structures, gradually increasing genome size.In [2] Lowering crossover rates helps maintain valuable genetic patterns during mating, enhancing NEAT's performance. Adjusting crossover parameters can improve NEAT's learning process, allowing gradual evolution while preserving information, leading to better outcomes compared to traditional methods. In [3], realization of feature selection through NEAT is investigated which aims to pick a subset of features that are relevant to the target concept. Two major goals in machine learning are discovery and improvement of solutions to complex problems.[4] Evolutionary computations, encompassing genetic algorithms and evolutionary programming, are population-based search methods that have proven effective in numerous complex tasks.

B. *Robotics*

NEAT has advanced robotics with adaptive control.In [5] Nvidia introduces an AI agent combining GPT-4's language skills with reinforcement learning for complex robot training. Challenges persist in reinforcement learning, including reward design.In [6], the paper proposes that a Robo-Advisor (RA) is a financial service that offers automated wealth-management advice based on algorithms, eliminating the need for a human planner.The paper [7] proposed the impact of chat-bot humanization, specifically focusing on the human-like characteristics of avatars,on customers’ purchase intentions and their willingness to reuse chatbots A novel evolutionary neural network controller guides quadrotors through autonomous flight, optimizing trajectory with precision. In [8] the study presents genetic algorithm-based optimal trajectory design for drones in post-disaster communication scenarios.In [9] the simulation of an inverted pendulum as a control task, the aim is to learn how to balance the pendulum without any prior understanding of its dynamics. Unlike other uses of neural networks for this task, performance feedback is only provided as a failure signal when the pendulum falls or hits the limits of a horizontal track, rather than after each step.

C. *Brain Computer Interface*

NEAT facilitates the development of Brain-Computer Interfaces (BCIs), translating neural signals into actionable commands. By evolving neural network architectures, NEAT-based BCIs offer intuitive interfaces for individuals to interact with the external world using their thoughts or intentions. Its applications extend beyond healthcare to entertainment, gaming, education, self-control, and marketing.In [10], presents a prototype brain-computer interface (BCI) integrated with a video game for upper limb rehabilitation. Using electroencephalographic signals processed through specific techniques, the system predicts users' motor intentions, allowing them to control a character's movement in a virtual environment created in Unity3D. In [11] the integration of brain science into AI is explored, highlighting opportunities and challenges. Understanding the brain's mechanisms can advance AI research significantly, bridging the gap between AI and human intelligence. This integration promises insights into intelligence and enhancements in AI systems, building upon breakthroughs like deep learning.The paper [12] aims for accessibility, explaining complex concepts in everyday language, catering to a broad audience seeking to grasp BCI’s implications and challenges in various spheres of human life.In [13], this paper introduces the concept of mapping of Artificially Intelligent(AI) computational systems. The concept of homunculus from human neurophysiology is extended to AI systems.the paper [14] proposes the intersection of neuroscience and artificial intelligence (AI), tracing the historical roots of AI research back to Alan turing’s question on machine thinking. It highlights the success of the deep network (deep net) architecture.

D. *Gaming*

NEAT revolutionizes game development, enabling creators to design adaptive, immersive experiences across platforms. With NEAT, developers innovate gameplay, content, and AI features, captivating players and setting new standards.In [15], NEAT evolves agents with diverse playstyles, though concerns arise about limited human interaction and unpredictable behavior.In [16] NEAT also optimizes artificial neural networks for games like Flappy Bird, enhancing NPCs' realism and adaptability. This leads to dynamic gameplay enriched with lifelike NPC interactions.The [17], focuses on agents playing computer games. They used a novel method combining NEAT and clustering to create diverse playstyles without predefined rules.In [18], it explores the integration of robotics and artificial agents, emphasizing the use of simulated environments for testing pathfinding and search space optimization algorithms.Importantly,[19] NEAT networks are able to learn with a reward function rather than back-propagation, meaning that you don’t need a giant dataset to train it on.The paper [20] demonstrates that NEAT can effectively evolve neural networks to control the movement and attack tactics (micro) for groups of ranged units in real-time strategy games, leading to skilled kiting behavior that mimics the tactics used by professional players.

III. PROPOSED SYSTEM

The proposed system aims to leverage NEAT in conjunction with the NVIDIA Isaac Simulator for the generation and simulation of quadcopters. NEAT, a powerful evolutionary algorithm, facilitates the development of neural networks by dynamically evolving their architectures to solve complex tasks efficiently. The NVIDIA Isaac Simulator provides a realistic virtual environment for testing and training these neural network controllers, offering sophisticated physics simulation and high-fidelity graphics. By integrating NEAT with the Isaac Simulator, the system enables the autonomous evolution and evaluation of quadcopter control policies in various simulated scenarios.The system utilizes the Ullman algorithm to reduce structure size by identifying similarities. This algorithm efficiently finds structural similarities between objects, helping to identify redundant components.

Integrating it with NEAT and the NVIDIA Isaac Simulator allows recognition of comparable elements in quadcopter designs or simulations, aiding in consolidating or eliminating redundancy. This enhances computational efficiency and reduces resource consumption, optimizing both NEAT's evolutionary process and the Isaac Simulator.By identifying comparable elements, such as structural components or behavioral patterns, this approach facilitates the streamlining of designs and simulations, effectively reducing redundancy and enhancing computational efficiency.The original NEAT algorithm is available in Python but does not function directly in Unity, a popular game development platform. To enable its use in Unity for simulations, the NEAT algorithm has been converted to C#.

After the mutation operation, we select the best mutations from the set. These structural adjustments induced by mutations facilitate the exploration of new solutions within the neural network architecture, promoting innovation and diversity in the population. By integrating Ullmann's algorithm for subgraph matching, NEAT can effectively identify and integrate beneficial alterations, improving its ability to navigate complex problem spaces. This dynamic adaptation process not only enhances robustness to environmental changes but also enables the network to continuously refine its performance, ultimately maximizing its overall effectiveness and versatility.

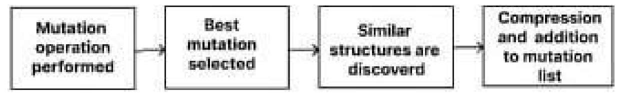

Fig. 1. Proposed Block Diagram.

A. NVIDIA Isaac Sim

NVIDIA Isaac Sim is a flexible robotics simulation platform designed to offer a quicker and more efficient approach to designing, testing, and training AI-driven robots. It’s powered by Omniverse to deliver scalable, photorealistic, and physically accurate virtual environments for building high-fidelity simulations.NEAT enables the automatic optimization of neural network architectures and parameters through evolutionary processes. By integrating NEAT with Isaac Sim, you can evolve neural networks that are specifically tailored to perform well in simulated robotic environments, leading to more efficient and effective robotic behaviors. This platform is valuable for accelerating the development process and improving the performance of robotic algorithms before deploying them in the real world.

IV. FIGURES AND TABLES

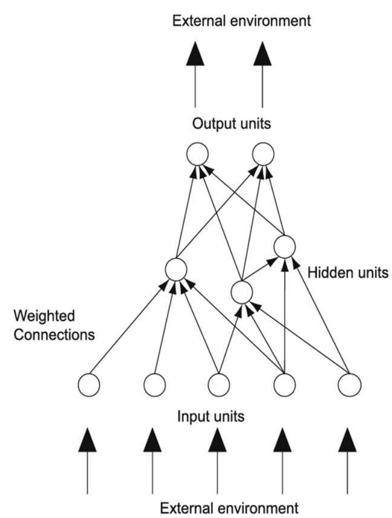

Fig. 2. A generic neural network architecture.

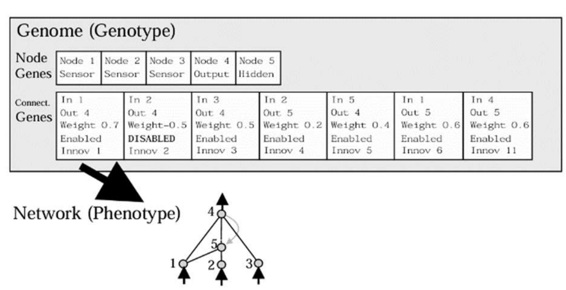

Fig. 3. Neat Encoding.

V. RESULTS AND DISCUSSION

In this study, we trained a quadcopter agent in NVIDIA Isaac Sim using the NEAT algorithm, aiming to achieve autonomous aerial navigation. The training process involved simulating the quadcopter's flight dynamics and optimizing its control policies through reinforcement learning. The integration of NEAT with NVIDIA Isaac Sim provided a robust platform for training and evaluating the quadcopter agent's performance, offering insights into its adaptive control strategies and emergent behaviors. To evaluate the training efficiency of the NEAT algorithm, we measured the computational resources required for training the quadcopter agent in NVIDIA Isaac Sim. The training process was conducted on a system equipped with an NVIDIA GeForce RTX 3090 GPU and an Intel Core i9 processor. The algorithm achieved convergence within approximately 2 hours of training time, demonstrating efficient utilization of computational resources. One notable improvement was the enhanced convergence speed of the evolutionary process, allowing for more rapid generation of effective quadcopter control policies. The Ullman algorithm effectively identified and compressed redundant substructures within quadcopter designs, leading to more streamlined neural network architectures and improved computational efficiency.

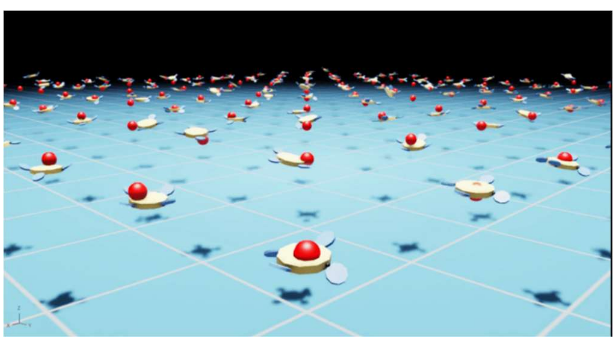

Fig. 4. Quadcopter Simulation.

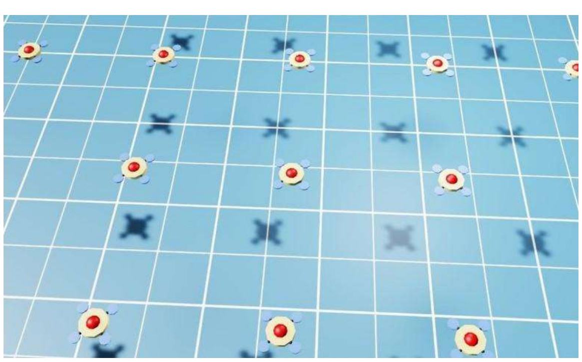

Fig. 5. Quadcopter Simulation Top View.

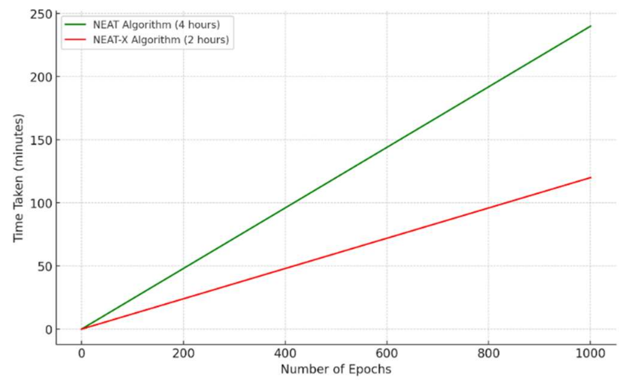

Fig. 6. Convergence graph.

This is the convergence graph comparing the time taken to train the model for the NEAT algorithm and the NEAT-X algorithm over the course of 1000 epochs: As seen in the graph, the NEAT-X algorithm converges at twice the speed of the NEAT algorithm, reducing the overall training time by half. This demonstrates the efficiency and effectiveness of the NEAT-X algorithm in achieving faster convergence and potentially better performance within a shorter training period.

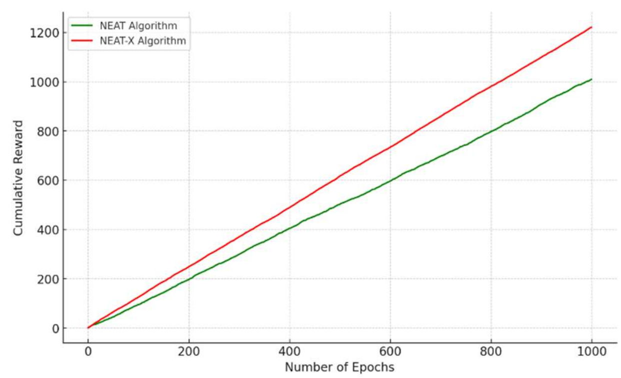

Fig. 7. Reward graph.

The Fig. 7 shows reward graph comparing the rewards achieved by the NEAT algorithm and the NEAT-X algorithm over the course of 1000 epochs.The graph demonstrates that both algorithms increase their rewards over time. However, the NEAT-X algorithm achieves higher rewards at a faster rate compared to the NEAT algorithm. This suggests that the NEAT-X algorithm is more efficient and effective in achieving better performance in a shorter time frame.The NEAT-X algorithm shows clear advantages in both training time and model performance, making it a promising approach for tasks involving neural network evolution and optimization.

VI. CONCLUSION

Combining NEAT with the NVIDIA Isaac Simulator offers a promising method for autonomously evolving and evaluating quadcopter control policies. By harnessing NEAT's dynamic architecture evolution alongside the realistic virtual environment provided by the Isaac Simulator, this system provides a robust platform for developing and testing complex neural network controllers.Furthermore, integrating the Ullman algorithm enhances computational efficiency and reduces resource consumption by identifying and eliminating redundant components in quadcopter designs or simulations. This optimization not only enhances the performance of NEAT's evolutionary process but also improves the fidelity of simulations within the Isaac Simulator.Moreover, converting the NEAT algorithm to C\# enables seamless integration with Unity, expanding its applicability for simulations in diverse domains, such as game development.Overall, this combined approach facilitates the adaptation of neural networks to changing environmental conditions or task requirements through mutations, ensuring continued effectiveness and adaptability in evolving scenarios.

Acknowledgment

The authors extend sincere gratitude to Mrs. Nighila Abhish for her Guidance and to project coordinator Mrs. Najila Nazar and Mrs. Gishma K M for mentorship. Their expertise played a vital role in shaping the direction and focus of this research.

References

[1] Stanley, K.O.; Miikkulainen, R. Evolving Neural Networks through Augmenting Topologies. Evol. Comput. 2002, 10, 99–127

[2] Heman Mohabeer and K. M. S. Soyjaudah, "Improving the Performance of NEAT Related Algorithms via Complexity Reduction in Search Space". Advances in Intelligent Systems and Computing ((AISC,volume 217))

[3] Soroosh Sohangir, Shahram Rahimi, Bidyut Gupta "Optimized feature selection using NeuroEvolution of Augmenting Topologies" 24-28 June 2013.

[4] P J Angeline 1, G M Saunders, J B Pollack "An evolutionary algorithm that constructs recurrent neural networks" PMID: 18267779

[5] Yecheng Jason Ma, William Lian, Guanzhi Wan, Anima Anandkum, and Dinesh Jayaraman, "EUREKA: HUMAN-LEVEL REWARD DESIGN VIA CODING LARGE LANGUAGE MODELS ". arXiv:2310.12931v1 [cs.RO] 19 Oct 2023.

[6] Ge R, Zheng Z, Tian X, Liao L. "Human–robot interaction:When investors adjust the usage of robo-advisors in peer-to peer lending." (2021)

[7] Schanke S, Burtch G, Ray G . "Estimating the impact of"humanizing" customer service chatbots." (2021)

[8] Tianrui Qiao,Yusuf A. Sambo, and Wasim Ahmad "Drone Trajectory Optimization using Genetic Algorithm with prioritized Base Stations". 2020 University of Glasgow, Scotland, United Kingdom.

[9] C.W. Anderson "Learning to control an inverted pendulum using neural networks" Volume: 9 Issue: 3.

[10] Jingtao Fan , Lu Fang , Jiamin Wu , Yuchen Guo and Qionghai Dai "From Brain Science to Artificial Intelligence". January 2020.

[11] Shiv Kumar Mudgal, Suresh K Sharma, Jitender Chaturvedi and Anil Sharma,"Brain computer interface advancement in neurosciences: Applications and issues" .Volume 20, June 2020, 100694

[12] Shiv Kumar, Anil Sharma, Suresh K Sharma, (2018) "Brain computer face advancement in neuroscience"

[13] Joghataie, "Extending concepts of mapping of human brain to artificial intelli-gence and neural network" (2020) in IEEE Transactions on Neural Networksand Learning Systems.

[14] Shimon Ullman. "Using neuroscience to develop artificial intelligence"(2019)

[15] V Asha,Arpana Prasad, C.R. Vishwanath, and K MadhavaRaj," Designing A Popular Game Framework Using Neat A Genetic Algorithms"(2023).

[16] Dr. T. Ramaswamy, Sindhu Lahari, and S. Sanjana, Teaching AI to Play Games using Neuroevolution of Augmenting Topologies (2022).

[17] Yu Iawasaki, Koji Hasebe,"Identifying Playstyles in Games with NEAT and Clustering"(2021) ,IEEE 25th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks

[18] Jerin Paul Selvan, Pravin S. Game,"Playing a 2D Game Indefinitely using NEAT and Reinforcement Learning". IEEE 2022, 5, 54–65..

[19] Robert MacWha "Evolving AIs using a NEAT algorithm" Mar 4, 2021

[20] Aavaas Gajurel, Sushil J Louis,Daniel J Méndez, Siming Liu "Neuroevolution for RTS Micro" 14 October 2018.

..