fire

MDPI

Article

Reinforced NEAT Algorithms for Autonomous Rover Navigation in Multi-Room Dynamic Scenario

Dhadkan Shrestha [1] and Damian Valles [2,*]

[1] Department of Computer Science, Texas State University, 601 University Dr, San Marcos, TX 78666, USA; gsu7@txstate.edu
[2] Ingram School of Engineering, Texas State University, 601 University Dr, San Marcos, TX 78666, USA
* Correspondence: d_v173@txstate.edu

Abstract: This paper demonstrates the performance of autonomous rovers utilizing NeuroEvolution of Augmenting Topologies (NEAT) in multi-room scenarios and explores their potential applications in wildfire management and search and rescue missions. Simulations in three- and four-room scenarios were conducted over 100 to 10,000 generations, comparing standard learning with transfer learning from a pre-trained single-room model. The task required rovers to visit all rooms before returning to the starting point. Performance metrics included fitness score, successful room visits, and return rates. The results revealed significant improvements in rover performance across generations for both scenarios, with transfer learning providing substantial advantages, particularly in early generations. Transfer learning achieved 32 successful returns after 10,000 generations for the three-room scenario compared to 34 with standard learning. In the four-room scenario, transfer learning achieved 32 successful returns. Heatmap analyses highlighted efficient navigation strategies, particularly around starting points and target zones. This study highlights NEAT's adaptability to complex navigation problems, showcasing the utility of transfer learning. Additionally, it proposes the integration of NEAT with UAV systems and collaborative robotic frameworks for fire suppression, fuel characterization, and dynamic fire boundary detection, further strengthening its role in real-world emergency management.

Keywords: autonomous navigation; NEAT (NeuroEvolution of Augmenting Topologies); transfer learning; multi-room environments; rover simulation; artificial neural networks; evolutionary algorithms; firefighting robotics; path planning; reinforcement learning; wildfire management; UAV collaboration; fuel characterization; fire boundary detection

check for updates

Academic Editors: Washington Rocha, António Vieira and Marcos Francos

Received: 7 November 2024
Revised: 9 January 2025
Accepted: 10 January 2025
Published: 23 January 2025

Citation: Shrestha, D.; Valles, D. Reinforced NEAT Algorithms for Autonomous Rover Navigation in Multi-Room Dynamic Scenario. *Fire* **2025**, *8*, 41. https://doi.org/10.3390/fire8020041

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Autonomous navigation is a field of study that focuses on developing systems capable of navigating and maneuvering without human intervention. These systems rely on various technologies, including sensors, computer vision, and decision-making algorithms, to perceive their environment, plan paths, and execute movements. Autonomous navigation has applications in robotics, self-driving vehicles, unmanned aerial vehicles (UAVs), and other domains where autonomous operation is desirable [1]. One of the critical challenges in autonomous navigation is the ability to perceive and interpret the surrounding environment accurately. This often involves data fusion from multiple sensors, such as cameras, lidars, and radars, to comprehensively understand the environment [2]. Additionally, robust path planning and decision-making algorithms are required to determine the optimal route while avoiding obstacles and adhering to navigation rules [3]. There are many sectors

where implementing autonomous navigation systems would be highly beneficial. One critical field where these systems could have a significant impact is firefighting.

Autonomous navigation systems can provide critical assistance to firefighters in hazardous environments. These systems, equipped with sensors like thermal cameras and gas detectors, can create real-time maps of structures and identify potential hazards [4]. Path planning algorithms allow the systems to determine optimal routes through smoke and debris while avoiding obstacles [5]. Some systems utilize robust unmanned ground vehicles (UGVs) that can autonomously navigate inside buildings and relay video/data to incident commanders outside [6]. Aerial drones with autonomous navigation capabilities can also provide situational awareness from above [7]. By reducing risks to firefighters and enhancing situational awareness, autonomous navigation technology has the potential to save lives during fire events such as wildfires and residential, chemical, and industrial cases.

Machine learning and deep learning offer significant potential for addressing critical challenges in wildfire management, including fire suppression, fuel load characterization, and risk assessment [8,9]. Among the adaptive algorithms, NeuroEvolution of Augmenting Topologies (NEAT) is a prominent method for evolving neural networks to handle dynamic and complex tasks [10]. NEAT's ability to optimize performance in decision making based on real-time environmental data makes it a compelling tool for advancing wildfire science. For example, NEAT can be utilized to develop predictive models for fire spread, optimize resource allocation strategies, and generate detailed risk maps to support proactive evacuation planning. Additionally, implementing NEAT-enhanced autonomous systems, such as rovers, is crucial for hazardous terrains, enabling real-time monitoring and post-fire recovery analysis. The work presented in this paper explores NEAT's capabilities in simulating environment management and highlights its applications in multi-room navigation tasks as an emulation of real-world scenarios.

In the past, similar research has been conducted in different parts of the country. One notable project was the deployment of teleoperated and autonomous robots during the Fukushima Daiichi nuclear disaster in 2011. Researchers from several organizations, including Chiba University, Tohoku University, and the University of Tokyo, sent in ground robots like Quince and PackBot to survey the damage and radiation levels inside the reactor buildings. These robots used autonomous navigation capabilities to map the interiors and identify hazards while being operated remotely by human controllers [4]. University of California San Diego researchers developed autonomous navigation algorithms for fire incidents. Their Wildfire Integrated Sensor Firebase Exploratory Robots (WISERs) could navigate through smoke and rubble to provide real-time 3D mapping and situational awareness. Field tests showed that ground robots could effectively explore structures after fires [11]. Researchers from České Budějovice and Czech Technical University developed autonomous aerial and ground robots for urban search and rescue scenarios like fires. Their unmanned aerial vehicles (UAVs) used vision-based navigation to identify survivors and guide the deployment of ground rescue robots [12]. Hence, using autonomous systems to assist firefighters has always been a significant vision to achieve.

The autonomous exploration of unknown environments poses numerous challenges researchers are actively addressing. A common issue is regional legacy, where exploration methods leave smaller, unexplored areas that reduce overall efficiency in later stages [13]. Additionally, achieving an optimal balance between exploration paths and efficiency is difficult due to the lack of a precise global optimization function that accounts for future planning, often leading to suboptimal navigation [13]. Environmental complexity further complicates exploration, as unstructured terrains present diverse shapes, fuzzy boundaries, and ambiguous semantic categories [14]. Limited sensory coverage in unmanned ground vehicles (UGVs) exacerbates inefficiency, resulting in frequent stop-and-go movements

and backtracking [15]. Non-holonomic constraints of wheeled vehicles introduce specific path limitations, making applying many existing exploration methods directly challenging. Moreover, updating target points during exploration is critical, as previously chosen frontiers can become suboptimal, requiring dynamic adjustments [15]. In scenarios like disaster zones or military operations, the inability to pre-collect data hinders the effectiveness of map-building systems [14]. Signal vulnerabilities, such as obstructions in forests or canyons, also impair positioning accuracy, while rapidly changing natural terrains necessitate frequent map updates to maintain reliability [14]. These challenges highlight the complexity of autonomous exploration and underscore the need for advanced algorithms and robust strategies to enhance performance in such environments.

Gaussian Process Regression (GPR) and Bayesian Optimization (BO) are techniques extensively studied for enhancing autonomous exploration. GPR provides a probabilistic framework for modeling unknown environments using observed data, making it a valuable tool for predicting spatial features and estimating uncertainties in unexplored regions [16,17]. Its incremental nature supports efficient online map building, and its ability to model complex, non-linear relationships enables its application to diverse terrains [18]. GPR has been utilized in exploration to model local perception, evaluate global exploitation quality, identify navigable free space, and construct metric–topological maps; it facilitates the quantification of uncertainties, which is essential for strategic exploration decision-making [18].

BO, often used with GPR, leverages the uncertainty estimates provided by GPR to optimize the exploration process. It guides the selection of sampling points, balancing the trade-off between exploring uncertain areas and focusing on promising regions [17]. BO outperforms traditional frontier-based methods by maximizing mutual information gain, maintaining safety through uncertainty-aware decision making, and enabling adaptive strategies that evolve. When integrated, GPR and BO enable active exploration by targeting high-uncertainty regions, iteratively refining exploration models, and embedding these capabilities into model predictive control (MPC) frameworks to balance exploration with safety and efficiency objectives [17]. Unlike GPR and BO, which primarily focus on uncertainty quantification and sampling optimization, reinforcement learning excels in scenarios where exploration strategies must evolve continuously in response to environmental feedback.

The primary objective of this research is to evaluate the performance of the developed AI model and autonomous rover in multi-room environments, precisely scenarios involving three or four rooms. The aim is to observe and analyze how the rovers navigate and explore these environments. The key performance metrics to be investigated include the order in which the rooms are visited, the time taken to discover the first room, the subsequent rooms, and the eventual return to the initial starting position. A fundamental goal is to assess the capability of the rovers to systematically traverse all available rooms within the environment and ultimately return to their original deployment location. This research focus is motivated by the potential real-world application of deploying autonomous rovers in enclosed spaces, such as apartments or buildings, where the rovers would be expected to explore the entire area and return to their initial position, enabling data collection and potential reusability.

Investigating multi-room scenarios is crucial as it emulates practical situations where autonomous systems must navigate complex, segmented environments effectively. This research study aims to validate the robustness, efficiency, and reliability of the developed AI model and rover system in addressing real-world exploration and mapping tasks by evaluating the order of room visitation, time-to-discovery metrics, and the ability to return to the starting point. The successful demonstration of these capabilities would pave the way

for deploying such autonomous systems in various applications, including surveillance, inspection, search and rescue operations, and environmental monitoring within enclosed structures [19].

2. Materials and Methods

The simulation of this research was performed on a 2D map created using Pygame. The programming language, tools, and libraries used for this research are Python 3.10.9, Pygame 2.5.1, pandas 1.5.3, neat-Python 0.92, sckit-learn 1.2.1, tensorboard 2.14.1, matplotlib 3.6.3, and gym 0.26.2. Pygame is a cross-platform set of Python modules designed for writing video games. It includes computer graphics and sound libraries, allowing developers to create fully featured games and multimedia programs. Pygame capabilities enable the simulation to visualize the rover's path, manage real-time interactions, and provide a platform for testing and improving AI algorithms for efficient and safe navigation in firefighting operations. The simulation map we have created is 1500 pixels in width and 800 pixels in height. The grid size of 150 $\times$ 150 is used to divide the map into a grid, where each cell represents an area of 150 $\times$ 150 pixels. This grid structure helps the rovers track the exploration and visitation of different zones within the simulation. The map is created according to our needs and saved as a PNG file, which will later be imported for the simulation process. On the map, we have various colors/regions. Figures 1 and 2 depict the simulated map for the rover navigation. The white areas represent obstacles the rovers must maneuver around, the black regions denote navigable spaces or rooms, and the green zones indicate target areas the rovers must cover as part of their navigation task. The transitions between black regions represent entry points connecting the different rooms. These components collectively define the environment's challenges and objectives for the rovers. We used the rover image as an agent that will move throughout the environment to learn things, for which we used the NEAT (NeuroEvolution of Augmenting Topologies) algorithm.

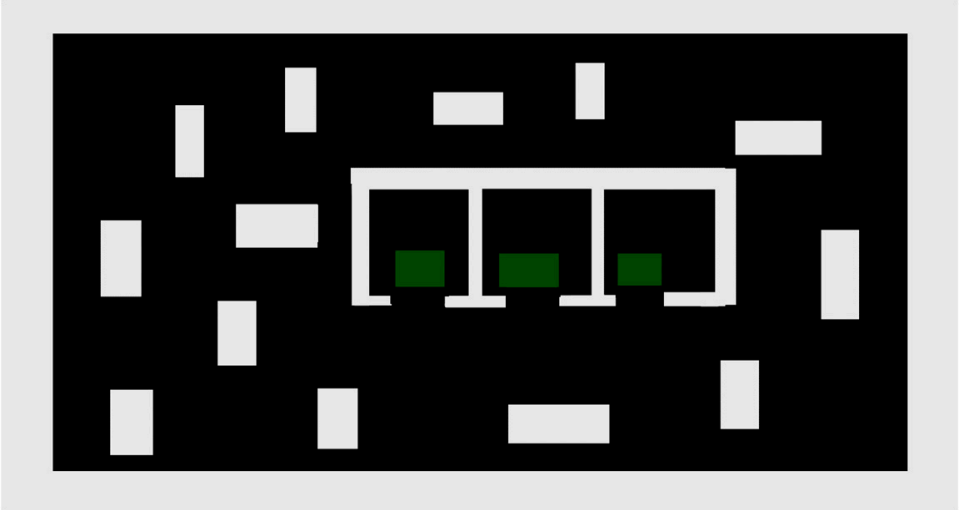


Figure 1. Map with three-room scenarios with obstacles.

NEAT is an evolutionary algorithm that generates artificial neural networks (ANNs). It evolves both the topology and weights of the networks, allowing them to grow more complex over generations by adding new nodes and connections. NEAT is particularly effective for tasks where the optimal network structure is not known in advance and can vary in complexity. The NEAT algorithm is particularly well suited for our research because of its distinctive method of evolving neural networks. Firefighting environments

are unpredictable and dangerous. NEAT's capability to evolve and adapt neural networks in response to these environments ensures that autonomous systems or decision-support tools developed using this approach can adjust to changing conditions, such as sudden fire outbreaks or structural collapses. This adaptability enhances their effectiveness in real-world firefighting operations. The algorithm's evolving nature fine-tunes decision-making processes, making the AI-driven tools and autonomous systems used by firefighters more efficient and accurate over time, potentially saving lives and resources [20].

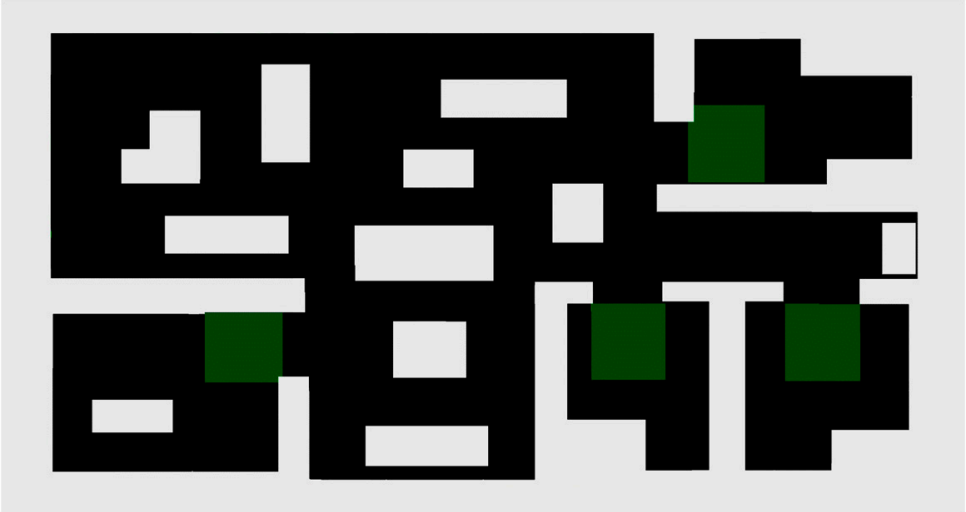


Figure 2. Map with the four-room scenarios with obstacles.

Figure 3 shows a basic understanding of how NEAT works. It starts with basic and simple neural networks, which gradually interact with the environment and evolve into complex or advanced neural networks. The arrow in the figure represents the input signal or data flowing into the neural networks. In contrast, the shades of blue represent the nodes or neurons in the neural network. The complexity of the network evolves as more iterations of the simulation occur, providing robustness in its capacity to navigate the design goals. NEAT creates a diverse population of neural networks, each with distinct architectures and parameter sets. It then "evolves" these networks over successive generations through processes like mutation and crossover, selecting the most successful networks to advance to the next generation [21].

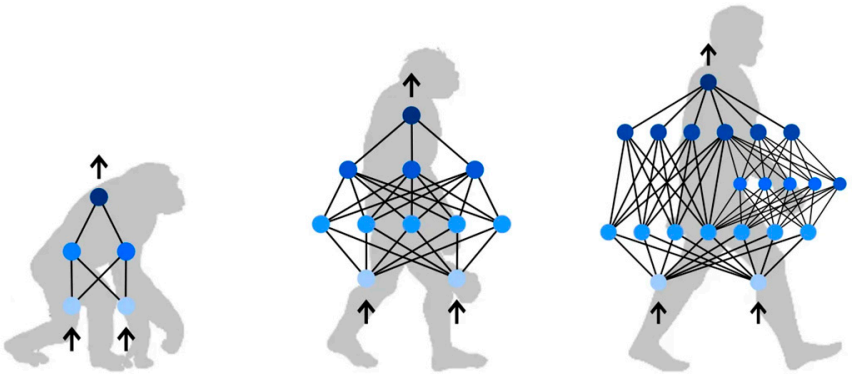


Figure 3. Evolution of how NEAT functions through simulation generations [15].

In the NeuroEvolution of Augmenting Topologies (NEAT) algorithm, the crossover process plays a critical role in evolving neural networks by combining the genetic material of two-parent networks to produce an offspring network [22]. As illustrated in Figure 4, the

gene structures of the parent networks are aligned based on their innovation numbers, with matching genes being inherited from either parent and non-matching genes categorized as disjoint or excess. The offspring inherits these genes following specific rules: matching genes can come from either parent, disjoint and excess genes are inherited from the more fit parent or randomly if parents have equal fitness, and disabled genes remain disabled if they are in either parent [23]. This process ensures genetic diversity and enables the evolution of complex neural network structures over successive generations, promoting innovation and adaptation within the population [10]. There are a few problems with using NeuroEvolution; one is the competing conventions problem.

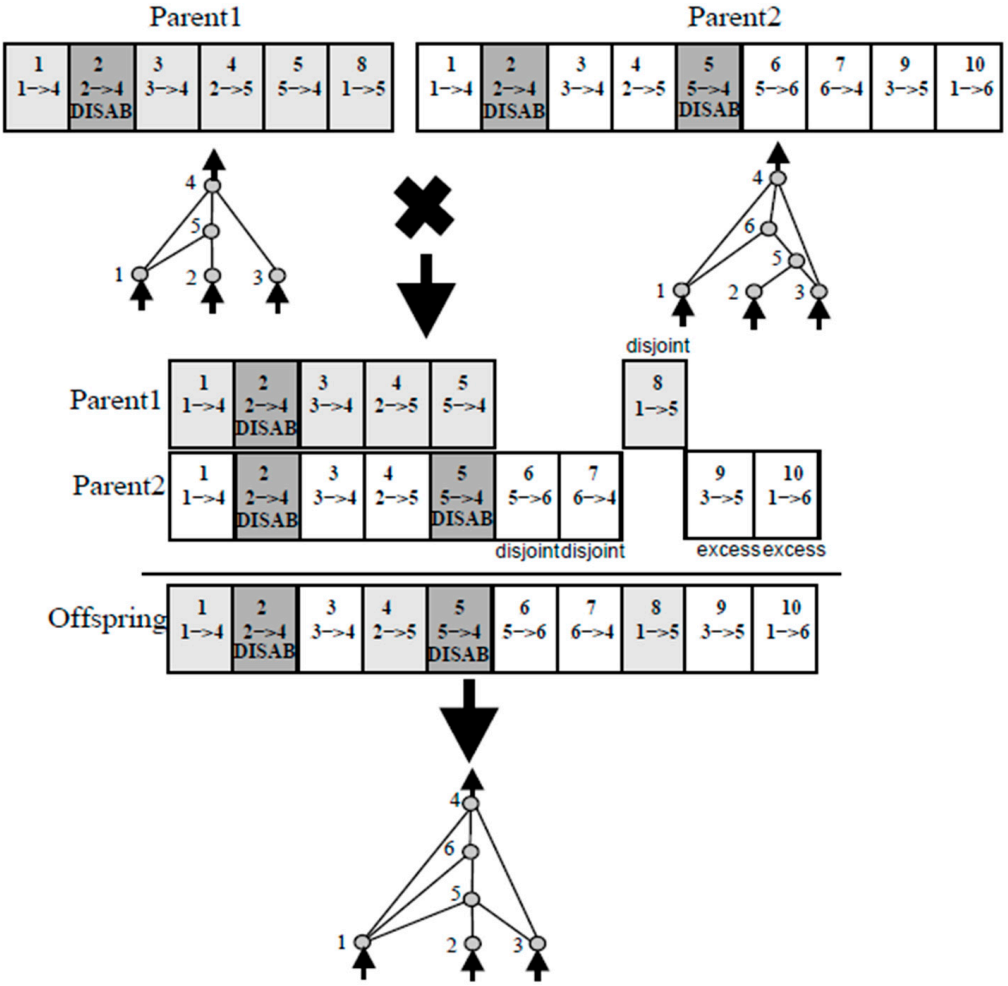


Figure 4. Crossover in NeuroEvolution of Augmenting Topologies (NEAT) [20].

Figure 5 illustrates the competing conventions problem in NeuroEvolution, specifically during the crossover of neural networks with different topologies. Two neural networks are shown here with the nodes A, B, and C. Both networks have the same connection, but the only difference is the ordering of hidden nodes. The network on the left has the order of A, B, and C, while the network on the right has the order of C, B, and A. The arrow in different nodes represents the input signals to the network. Genes representing the same function might be misaligned during crossover due to different node ordering. While the network is crossed over, the resulting offspring have incorrect connections. In such scenarios, NEAT solves this by using historical marking to track the origin of each gene, ensuring the proper alignment and avoiding the competing conventions problem [24].

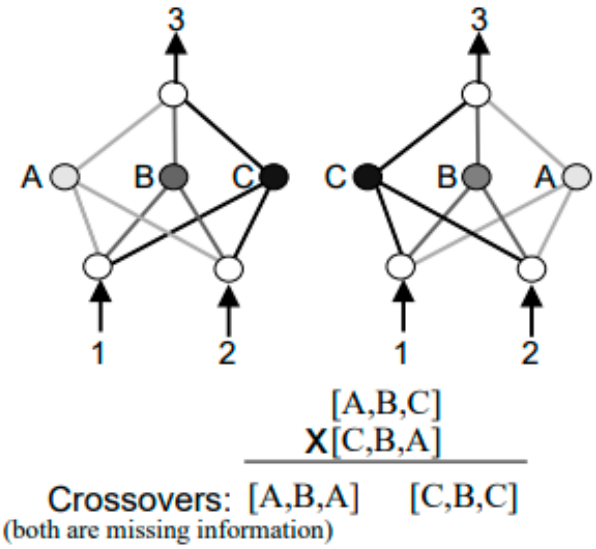


Figure 5. Competing conventions problem [20].

On top of this, we also implement the concept of reinforcement learning, particularly in how the neural networks are evaluated and evolved using NEAT. It incorporates reinforcement learning principles by rewarding certain behaviors and using those rewards to guide the evolution process. Some reinforcement learning aspects that we have implemented at present are as follows:

- Fitness Rewards: We implement fitness scores for each rover based on performance. For instance, rovers receive rewards for visiting new green zones, reaching the starting point after visiting zones, and avoiding obstacles. These fitness scores guide the NEAT algorithm in selecting and evolving the neural networks.
- Exploration and Exploitation: Through the evolutionary process, rovers explore different strategies to maximize their fitness. The best-performing strategies (exploitation) are carried forward and mutated to explore new strategies (exploration).
- Learning from Experience: While not traditional RL, the neural networks evolve based on their performance over multiple generations. This can be seen as a form of learning from experience, as networks that perform well are more likely to be preserved and optimized.

In our simulation, we utilized a population of 50 rover agents to explore the environment and learn the specified objectives. The NEAT configuration file was set to use a feedforward neural network, as this architecture outperformed the recurrent neural network in our previous research. The config file contains all the parameters, such as population, hidden networks, mutate rate, bias rate, etc., which can be tweaked according to project needs [25]. All parameters were kept consistent with those used in the feedforward network configuration. The parameters used in the NEAT algorithm were chosen based on preliminary experiments and prior research to balance exploration and exploitation effectively. The parameter selection process involved a series of iterative experiments to identify configurations that maximize fitness scores and task completion rates. We explored methods such as the following:

- GridSearch: evaluated multiple combinations of mutation and crossover rates.
- Adaptive parameter tuning: Adjusted parameters dynamically based on performance in earlier generations. Future work could leverage advanced optimization techniques, such as Bayesian Optimization or Genetic Algorithms, to further refine parameter selection.

Key parameters include the following:

- Population size: Set to 50, ensuring sufficient genetic diversity while maintaining computational efficiency. A larger population size could enhance exploration but would increase computational costs.
- Fitness criterion: Set to max. Using max ensures that the highest-performing individual in each generation drives the evolution process.
- Activation function: Set to Tanh. The Tanh function was chosen for its ability to handle both positive and negative inputs, making it suitable for complex decision-making tasks.
- Connection mutation parameter: Set to 0.5. These rates prevent the networks from becoming overly complex or too simplistic, maintaining adaptability across generations
- Node mutation parameter: Set to 0.2. Lower probabilities for adding and deleting nodes compared to connections help stabilize the network structure while allowing incremental complexity growth. These parameters ensure that the networks evolve complexity gradually, avoiding premature overfitting or convergence.
- Elitism: Set to 3. Preserving the top three networks from each generation ensures that the best-performing solutions are retained for further refinement.

Besides these parameters, we used different mathematical formulas in our project for various purposes, such as the following:

- To calculate the distance between two points like the distance moved by the rover over time to detect if the rover is stuck, the distance between the car's current position and its previous positions to detect revisitation or distance from the rover to its starting point when checking if the rover has returned after visiting green zones, we used

$$distance = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{1}$$

- To extend the radar beam from the rover's center to the given direction and check for obstacles, we used

$$x = car_center_x + \cos(angle) \ * \ length \tag{2}$$

$$y = car_center_y \ + \sin(angle) \ * \ length \tag{3}$$

- To rotate the rover and point around the center when the rover changes its direction, we used

$$rotated_x = \cos(angle) \ * \ original_x - \sin(angle) \ * \ original_y \tag{4}$$

$$rotated_y = \sin(angle) \ * \ original_x + \cos(angle) \ * \ original_y \tag{5}$$

The simulation monitors various aspects of rover performance, such as distance traveled, time spent, collisions, green zone entries, time to visit any green zone, the order of green zone visitation, and the return of the rover to the starting point. These data are collected during the simulation's update phase and stored for later analysis. Performance metrics, including rewards and penalties, are used to assess the fitness of each rover agent. The fitness function considers penalties for proximity to walls, revisiting the same areas, and rewards for entering new grids or reaching green zones. The value of rewards and penalties varies from accomplishment. If the rover visits the green zone for the first time, it will receive a reward of +10, if it returns to the starting point after visiting green zones, it will receive a reward of +100, for visiting each grid, it will be rewarded +1, for striking in the wall rovers, it will receive a penalty of -2, for revisiting the same location multiple times within the short period, it will receive rover penalty of -50, and if rovers do not move a sufficient distance over a certain number of time steps, they are penalized by being

deactivated. The rewards and penalties in this NEAT-based rover simulation are designed to promote exploration, efficient navigation, and goal achievement (visiting green zones and returning to the starting point) [26]. Rewards encourage behaviors that align with these goals, while penalties discourage inefficient or undesirable behaviors, such as collisions, the revisitation of the same location, and becoming stuck.

Additionally, the simulation generates a heatmap to visualize the frequency of rover positions, which can be crucial for strategic planning and analysis in real-life scenarios [27]. Utilizing augmented reality interfaces enables operators to visualize robot trajectories and operational zones in real-time, facilitating the identification of critical areas and the assessment of operational strategies overlayed with heatmaps, thereby supporting informed decision-making in future deployments [28]. The heatmap is created by tracking the rover's position over time, reducing the resolution of heatmap points, and applying a Gaussian blur for smoother visualization. The final heatmap is then blended with the original map image to highlight areas with higher rover concentrations [29].

Along with the heatmap, the simulation outputs three text files and three image files. Each text file records the rover's data during the simulation. The car_data.txt file records minute details of each rover, such as green zone visited time, order, which green zone was visited at which generation and at which time, etc. Another text file named fitness_statistics.txt records the highest and average fitness of each generation. The final text file, generation_reward.txt, records how many rovers have traveled to all the green zones on the map and how many of those rovers went to the starting position after visiting all green zones.

Compared to our previous work, we made several key changes and improvements in the techniques which will be beneficial for real-life implementation. Some of the key differences in their impact on real-life implementation are as follows:

1. Grid-based exploration
   We implemented a grid-based system for exploring the environment, replacing the dynamic clustering from the last time. This change encourages more thorough exploration. In firefighting scenarios, this could lead to more comprehensive searches of buildings or areas, ensuring no spots are missed.
2. Improved collision detection
   We implemented a more sophisticated collision detection system, including checks for the front and back of the car. This could translate to better obstacle avoidance in complex environments, crucial for navigating debris-filled areas during firefighting operations.
3. Stuck detection
   It now includes a mechanism to detect if a rover is stuck and deactivate it if necessary. This could prevent rovers from wasting time in unproductive areas, allowing for more efficient resource allocation in time-critical situations.
4. Enhanced green zone (target area) interaction
   It tracks unique green zone visits and implements a more complex reward system for visiting these areas. This could be analogous to identifying and thoroughly investigating potential fire hotspots or areas of concern in a firefighting scenario.
5. Return-to-start behavior
   It includes logic for rovers to return to their starting point after completing objectives. This could be crucial for firefighting rovers to return to a safe zone or recharging station after completing their tasks.
6. More detailed performance tracking
   The new script tracks various performance metrics, including the number of rovers that visit all zones and return successfully. This detailed tracking could provide valu-

able insights into improving rover strategies and identifying areas for improvement in real-world applications.

7. Adaptive exploration
   The new script implements a system to detect when exploration has stagnated and potentially adjust strategies. This could allow firefighting rovers to adapt their search patterns based on the effectiveness of the current plan, potentially leading to more efficient operations.
8. Improved data logging and analysis
   It has comprehensive data logging and visualization capabilities. This could give firefighters and operators better insights into rover performance and environmental conditions, aiding decision making and strategy refinement.
9. Modular reward system
   It now has a more flexible reward system that can be easily modified to encourage different behaviors. This modularity could allow for quick adjustments to rover behavior based on specific firefighting scenarios or changing priorities during an operation.

We ran the simulation multiple times with varying numbers of generations. The simulation started with 100 generations and was later increased to 500, 1000, 5000, and 10,000 generations. Simulations were conducted for both three-room and four-room scenarios. Output files for each simulation were saved in TXT and JPEG formats to facilitate data analysis.

Additionally, we performed a transfer learning process to evaluate its impact on the results. This involved using pre-trained population information from a previous one-room scenario, trained for 200 generations. The goal was to determine whether rovers with prior knowledge of objectives perform better. Our initial research observed that rovers began learning vaguely after 200 generations, making this a suitable baseline for transfer learning.

3. Results

We used the Learning, Exploration, Analysis, and Processing (LEAP2) next-generation cluster for the simulation process at Texas State University. We started with three-room simulations and then four-room simulations. After completing these standard simulations, we carried out the simulation again for the three-room and four-room simulations, but this time, we used transfer learning to see the results. In the end, we compared the result of the standard simulations with the transfer learning simulations to see whether it was better to obtain vague knowledge about the room.

3.1. Standard Simulations

In this simulation approach, we explore a scenario involving three rooms and four distinct situations. We focus on a model where the rover begins each simulation as a blank slate devoid of prior knowledge or experience. At the outset of every run, the rover must understand its environment, objectives, or methods to achieve its goals. It must interact with its surroundings in real time to discover its purpose and learn how to accomplish its tasks. This approach allows us to observe how the rover adapts and develops strategies from scratch in each new situation across the various rooms and scenarios, providing insights into its learning process without the benefit of prior experience.

3.1.1. Three-Room Simulation Results

1. Simulation for 100 Generations

Figure 6 shows progression in fitness scores throughout 100 generations. The average fitness improved significantly, reaching approximately 2693.88 by Generation 100,

compared to much lower values in the earlier generations. The maximum fitness saw considerable fluctuations, with peaks reaching as high as 38,803 around Generation 60.

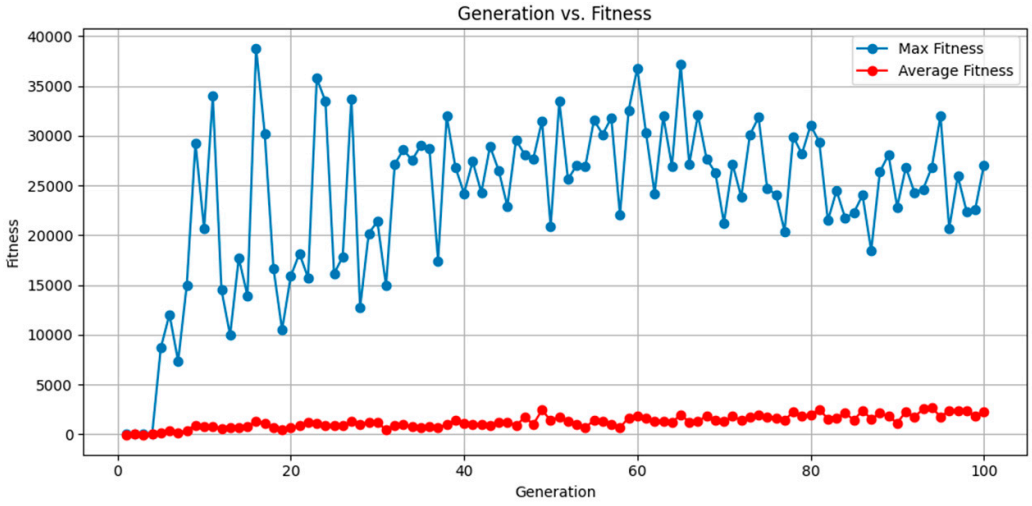


Figure 6. MaxAvg fitness plot for 100 generations.

The heatmap in Figure 7 indicates a high concentration of rover activity around the starting area, suggesting a focus or difficulty in navigating further. The statistics reveal that very few rovers return to the starting point after visiting all zones.

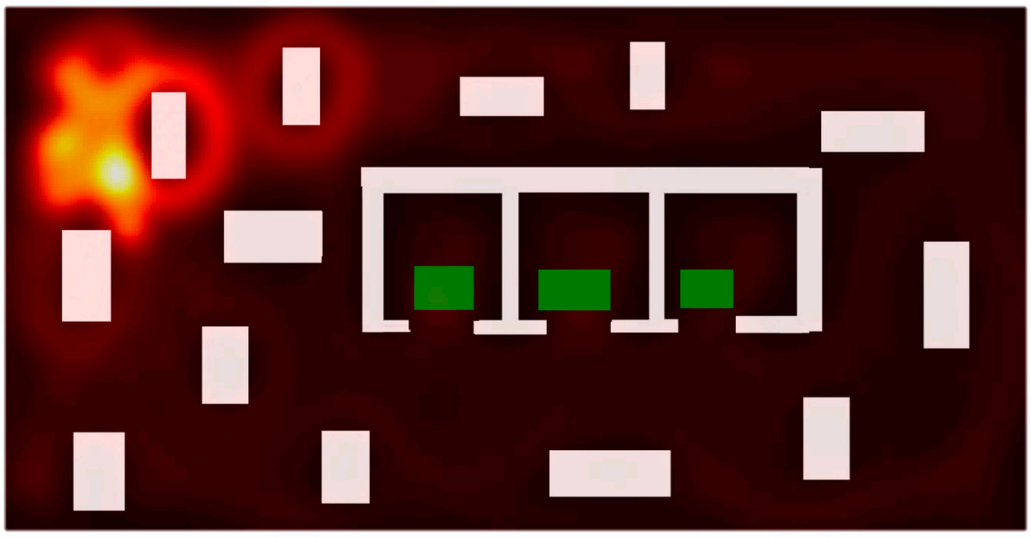


Figure 7. Heatmap for 100 generations.

2. Simulation for 500 Generations

The simulation results of the 500 generations in Figure 8 demonstrate a steady improvement in average and maximum fitness scores over time. The maximum fitness also significantly increases, with a peak value of 37,695 in Generation 67. The simulation results significantly improve average and maximum fitness scores over 500 generations. The highest recorded average fitness was approximately 8644 in Generation 450, while the maximum fitness achieved was around 37,695 in Generation 67. The fitness scores depict a general upward trend, indicating progressive learning and optimization by the NEAT algorithm.

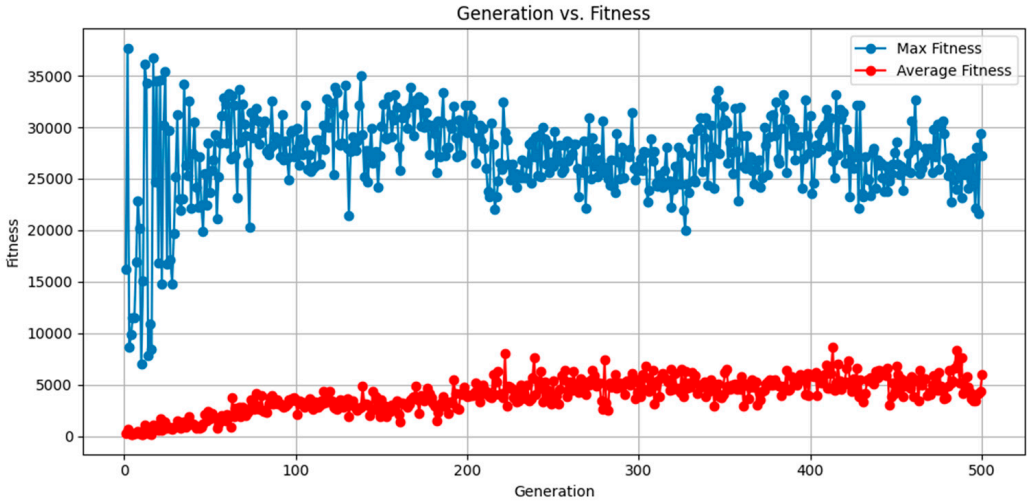


Figure 8. MaxAvg fitness plot for 500 generations.

The heatmap from Figure 9 indicates a strong focus of activity near the starting area, suggesting that the rovers have difficulty moving away from the start or frequently return to it. However, as generations progress, more rovers successfully visit all three green zones and return to the starting point, with the highest number of successful completions in a single generation being fifteen rovers in Generation 280.

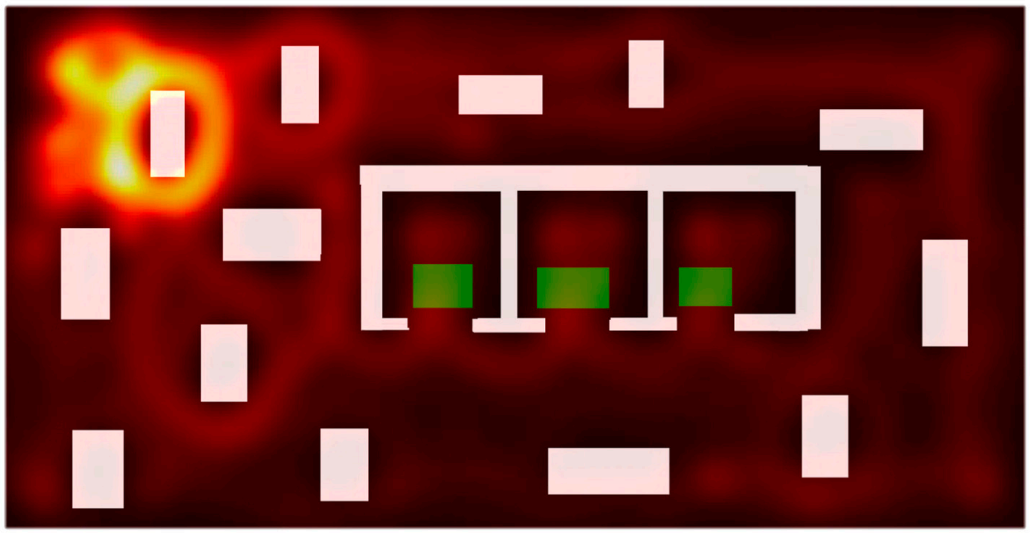


Figure 9. Heatmap for 500 generations.

3. Simulation for 1000 Generations

The fitness plot results of the 1000-generation simulation, as in Figure 10, indicate a clear trend in improvement in both average and maximum fitness over time. The average fitness increased steadily, reaching a peak of approximately 9086.44 in Generation 968. The maximum fitness also showed significant variability, with the highest value recorded at around 38,394 in Generation 7.

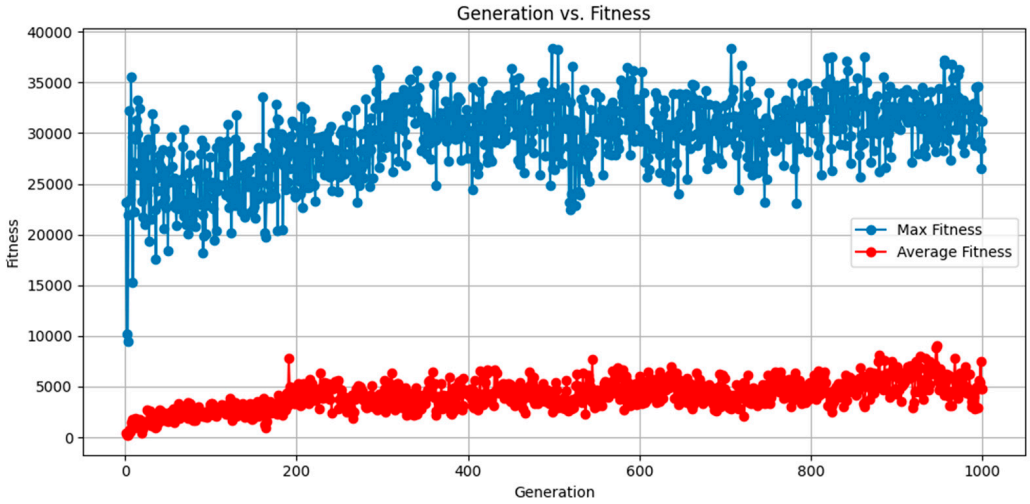


Figure 10. MaxAvg fitness plot for 1000 generations.

The heatmap from Figure 11 suggests that the rovers tend to concentrate their activity around the starting area, which might indicate challenges in navigating away from the start or a strategic focus on returning to this region. Throughout the generations, more rovers completed visiting all three green zones and returning to the starting point, with the highest number of successful completions being eighteen rovers in Generation 946.

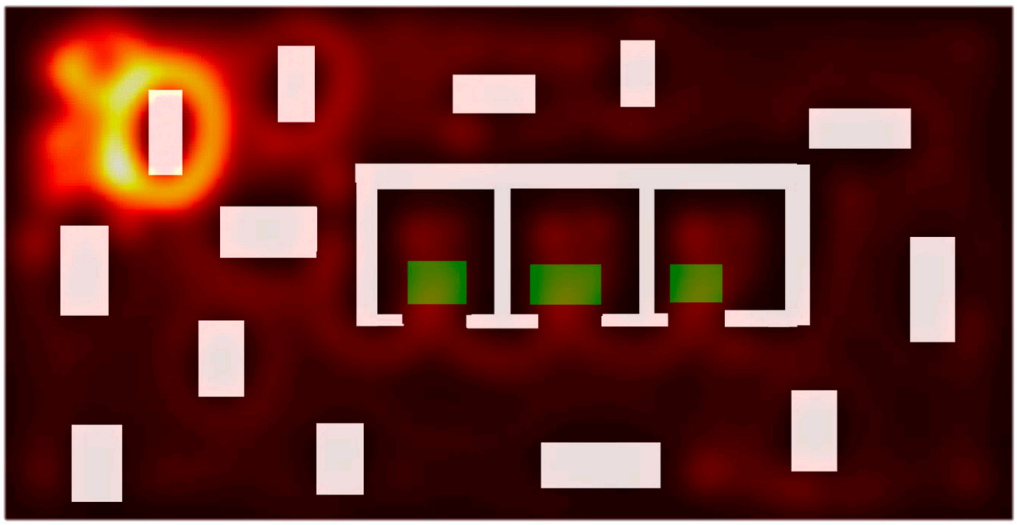


Figure 11. Heatmap for 1000 Generations.

4. Simulation for 5000 Generations

The simulation results indicate that throughout 5000 generations, the average fitness of the evolving population shows an overall increasing trend, though with significant fluctuations—the average fitness peaks at approximately 12,653.59 in some generations. Figure 12 shows that the maximum fitness consistently stays between 20,000 and 40,000, with a peak of 39,678, indicating that specific individuals perform significantly better than the average.

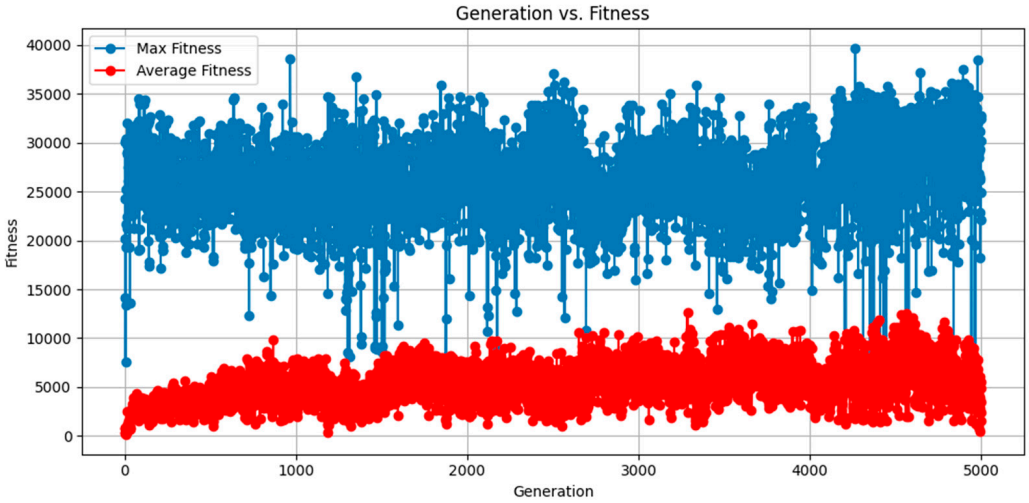


Figure 12. MaxAvg fitness plot for 5000 generations.

In terms of behavior, the population shows progress in navigating the environment and achieving objectives. By the later stages, multiple rovers consistently visited all three green zones and successfully returned, with a peak performance where up to 20 rovers completed this task in Generation 1764. From Figure 13, we can conclude that a significant number of rovers achieve their objectives while many other rovers partially complete their task of finding the green zones.

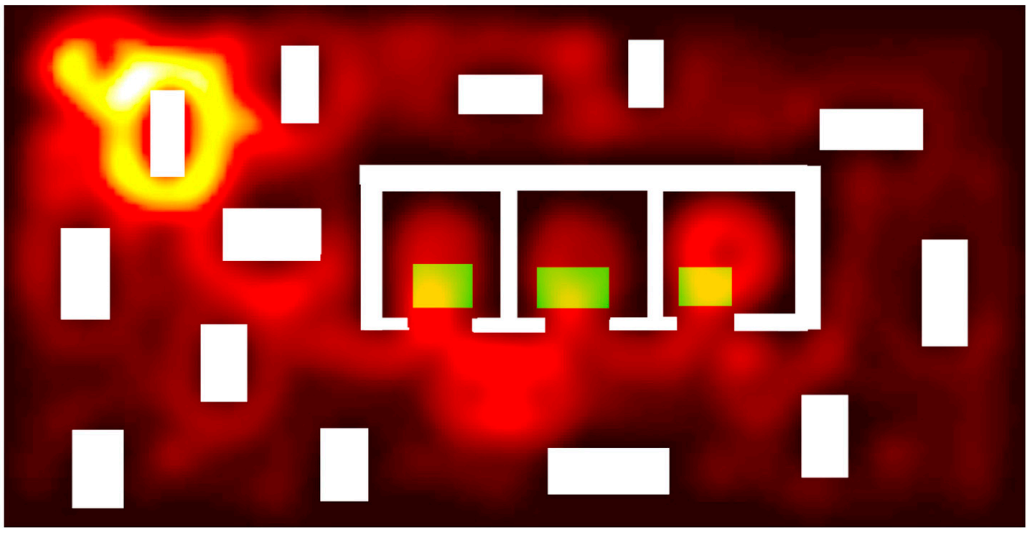


Figure 13. Heatmap for 5000 generations.

5. Simulation for 10,000 Generations

The 10,000-generation simulation experiment significantly improved average and maximum fitness scores over 10,000 generations. Figure 14 shows that the peak average fitness reached approximately 14,256.36 in the later generations, while the maximum fitness score reaches as high as 37,685 during the experiment. The highest number of rovers that completed the task and returned to the starting position was 34.

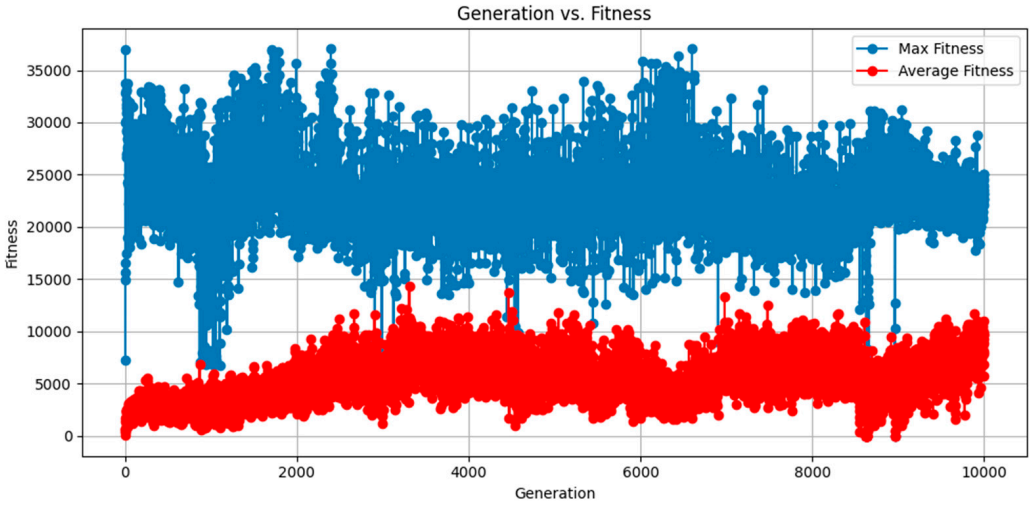


Figure 14. MaxAvg fitness plot for 10,000 generations.

The heatmap from Figure 15 reveals the highest activity concentration near the starting zone, indicating frequent car returns. The green zones show significant but less intense activity, suggesting successful car navigation. Overall, the rovers effectively reached critical areas on the map, highlighting the success of the neural network training.

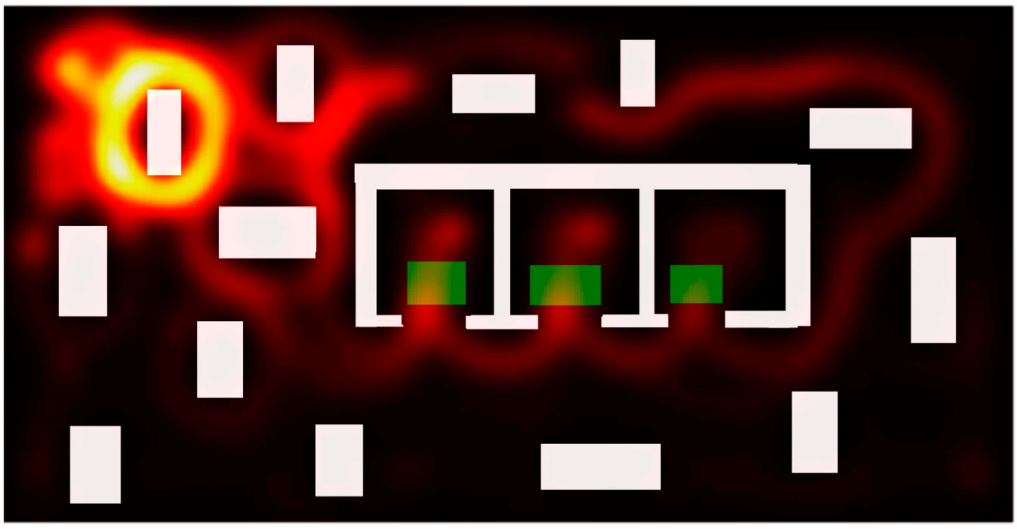


Figure 15. Heatmap for 10,000 generations.

3.1.2. Four-Room Simulations

1. Simulation for 100 Generations

Figure 16 simulation data results across 100 generations show a clear trend in fitness evolution over time. The maximum fitness peaked early in the simulations, with a value of 34,600 in Generation 50, indicating that the fittest individual in the population achieved a high level of optimization by this point. However, the average fitness of 9681.82 increased gradually, with fluctuations reflecting the population's ongoing adaptation and selection processes. In the rover data, specific green zones were visited at different times across generations, with the first complete visit to all zones occurring around Generation 52.

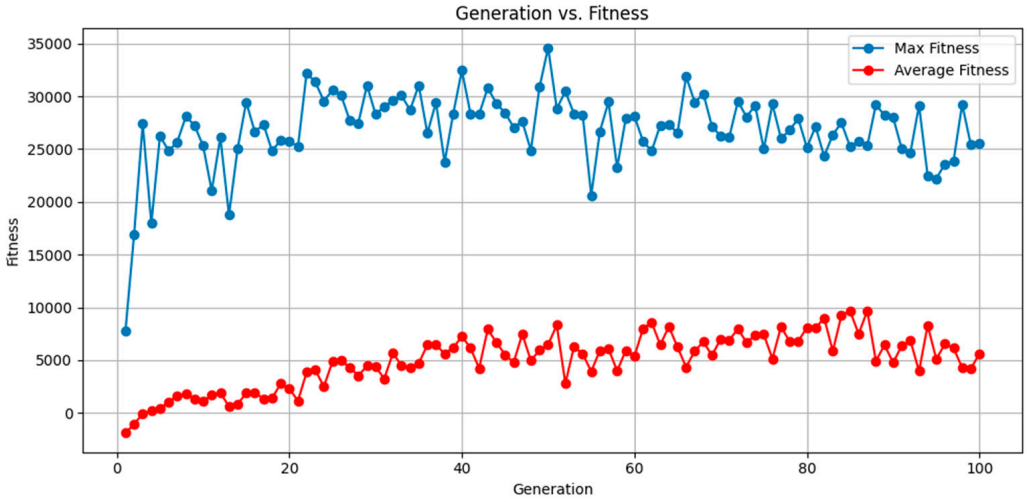


Figure 16. MaxAvg fitness plot for 100 generations.

The heatmap from Figure 17 of agent activity reveals that the most intense activity occurs near the starting position and along frequently traveled paths to the green zones, suggesting that while the agents effectively learn optimal paths, there remains potential for improving exploration in less-traveled areas.

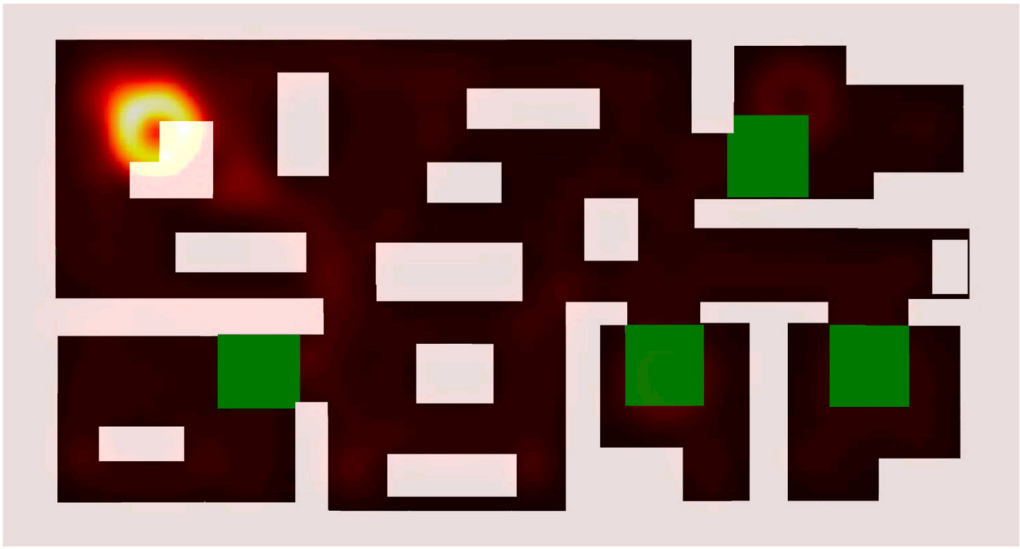


Figure 17. Heatmap for 100 generations.

2. Simulation for 500 Generations

The simulation result for the 500th generation in Figure 18 shows that the maximum fitness achieved is 35,563. The average fitness improves consistently over the generations, peaking at around 13,069.14 by the 500th generation. This trend is also reflected in the number of rovers successfully navigating through all four zones, which increases from 0 in the initial generations to a maximum of 13 in the 375th generation.

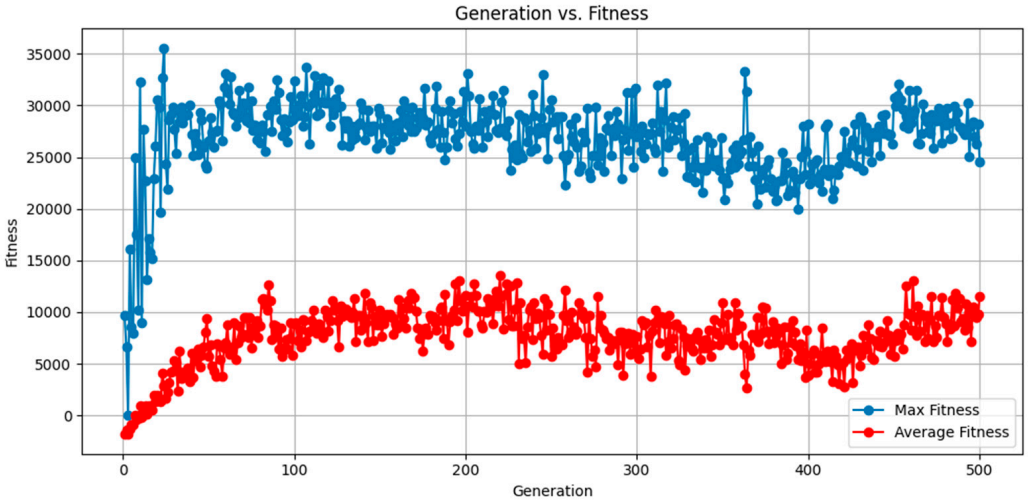


Figure 18. MaxAvg fitness plot for 500 generations.

As the evolutionary process progresses, the Figure 19 heatmap becomes more focused, with increased intensity around the optimal paths and target zones. This demonstrates that the rovers are learning to optimize their routes effectively.

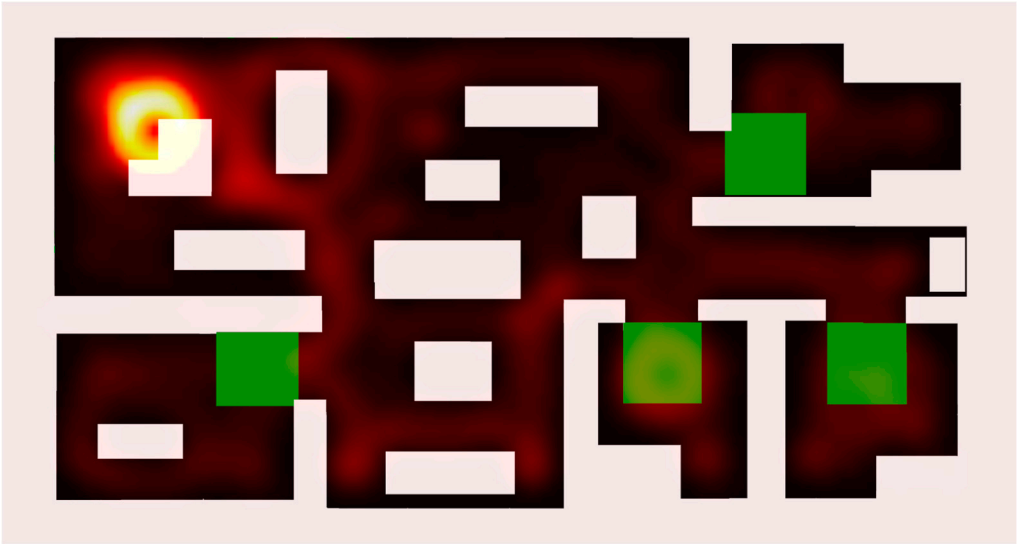


Figure 19. Heatmap for 500 generations.

3. Simulation for 1000 Generations

The fitness plot from Figure 20 of the 1000-generation simulation experiment shows significant fluctuations and improvements in average and maximum fitness scores over 1000 generations. The peak average fitness reaches approximately 15,804.82 in the later generations, while the maximum fitness score reaches as high as 35,844 during the experiment. The first generation's average fitness is notably low at −1944.88, which indicates the unknown objectives at the beginning of the simulation. The fitness values gradually increase throughout the simulation, which suggests that the rover is learning from the environment.

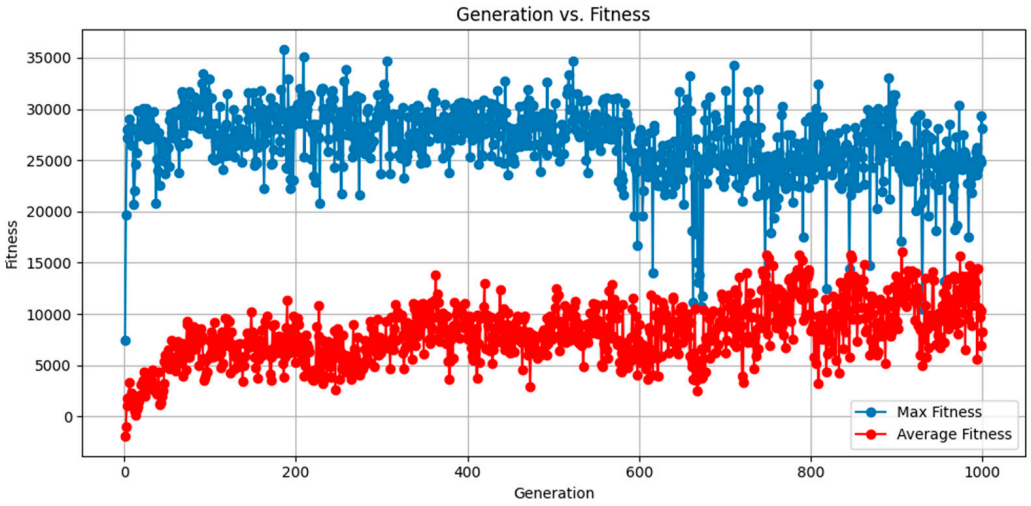


Figure 20. MaxAvg fitness plot for 1000 generations.

The Figure 21 heatmap data reveals concentrated activity near the starting point with some dispersion towards the green zones, indicating varying success in task completion. In some later generations, up to sixteen rovers complete all tasks.

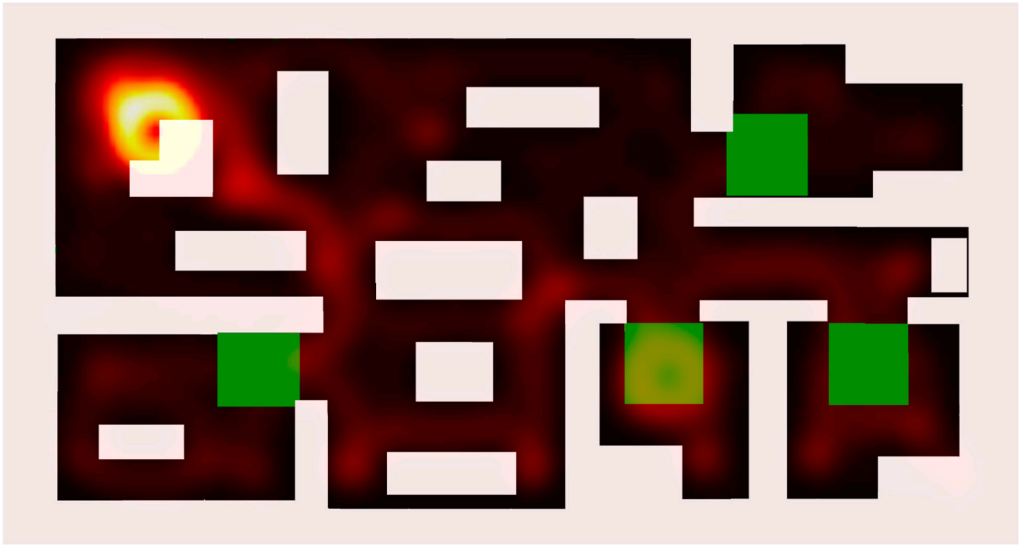


Figure 21. Heatmap for 1000 generations.

4. Simulation for 5000 Generations

Figure 22 shows the fitness plot of the 5000-generation simulation, which reveals significant fluctuations in average and maximum fitness levels, with a general trend in improvement over time. The average fitness fluctuates considerably, with peaks reaching approximately 15,214.27 and some points dipping below zero. Despite these fluctuations, the overall trend shows a gradual increase in fitness as the generations progress. The maximum fitness demonstrates similar variability, with peaks reaching around 35,894, indicating that specific individuals within the population can achieve high-performance levels.

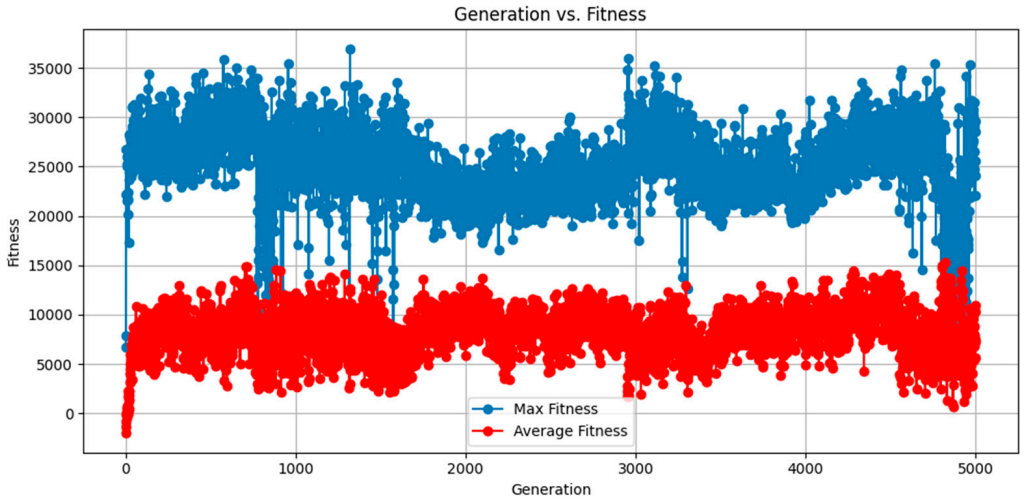


Figure 22. MaxAvg fitness plot for 5000 generations.

The heatmap analysis in Figure 23 highlights a concentration of activity in the top-left corner of the map, suggesting that this area might be a critical point for the cars, possibly serving as a starting location or a frequently visited area. The reward metrics show an increasing number of rovers successfully navigating through all four zones and returning to the starting point, with up to 23 rovers completing the task in some generations.

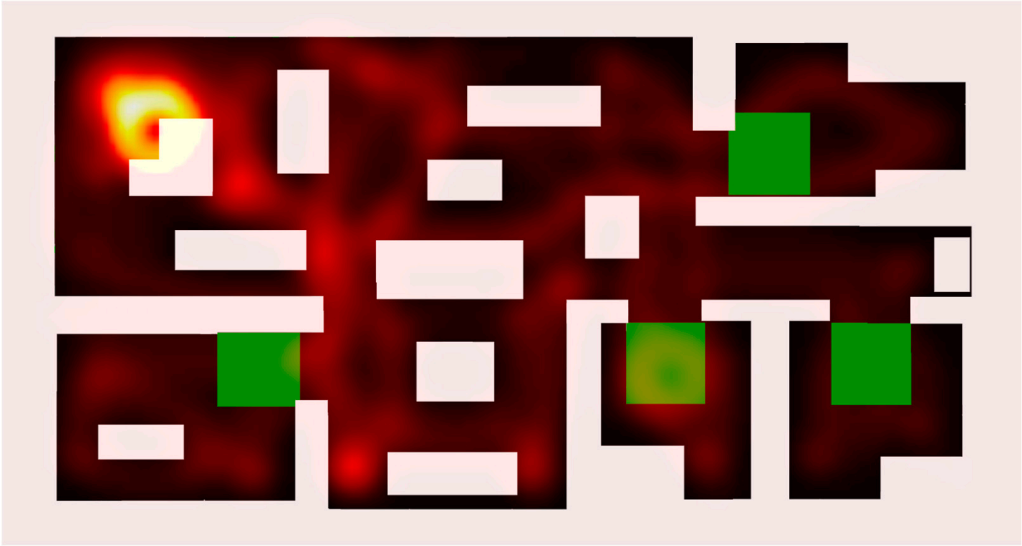


Figure 23. Heatmap for 5000 generations.

5. Simulation for 10,000 Generations

The 10,000-generation simulation experiment significantly improved average and maximum fitness scores over 10,000 generations. From the fitness plot, Figure 24 shows that the peak average fitness reaches approximately 18,315.3 in the later generations, while the maximum fitness score reaches as high as 37,605 during the experiment. The first generation's average fitness is notably low at −2036.12. The highest number of rovers that complete the task and return to the starting position amounts to 25.

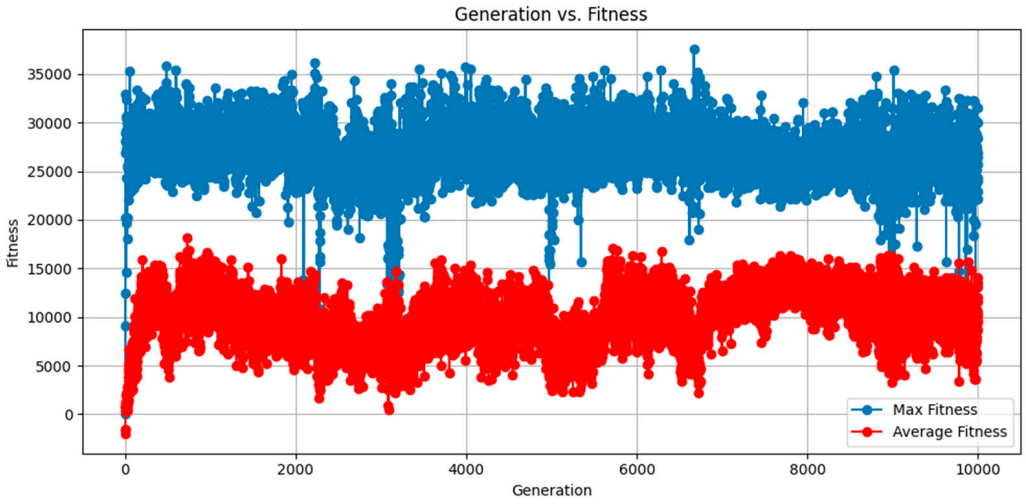


Figure 24. MaxAvg fitness plot for 10,000 generations.

The Figure 25 heatmap shows that the highest concentration of rover activity is near the starting zone, indicating successful return attempts by multiple cars. The green zones also show significant activity, reflecting successful navigation by the rovers to these zones. This pattern suggests that the evolved neural networks effectively guide the rovers toward the intended objectives on the map.

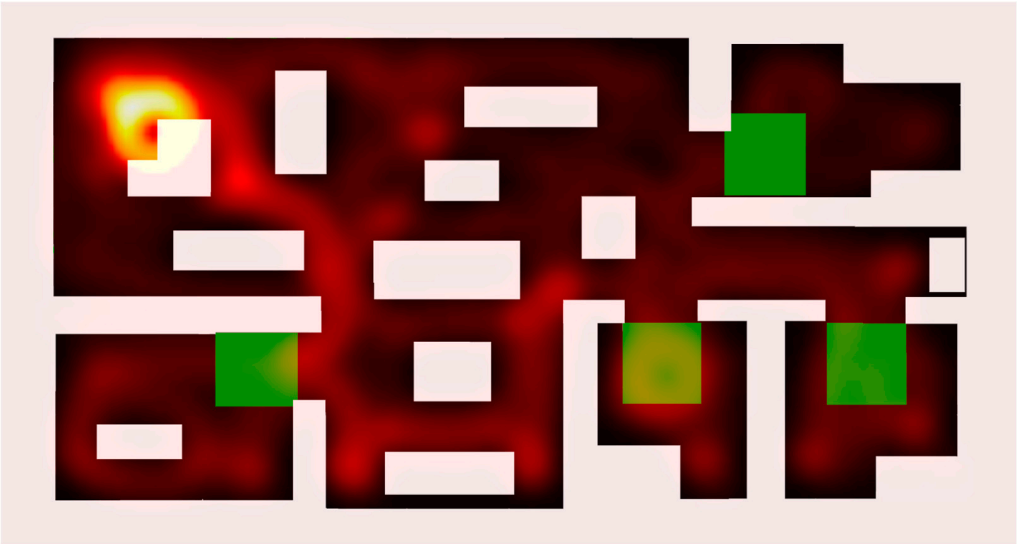


Figure 25. Heatmap for 10,000 generations.

3.2. Transfer Learning Simulations

We initiate a simulation that builds upon our previous work, focusing on three rooms and four scenarios, but with a critical difference: the implementation of transfer learning. Transfer learning involves applying knowledge from one task to a new, related task. In this case, our rover begins with pre-existing knowledge about its objectives and methods to achieve them. The foundation for this simulation is a pre-trained model developed in a single-room scenario over 200 generations. This pre-population of knowledge is the starting point for our current, more complex simulation [30].

By integrating comprehensive global and local obstacle data as transfer learning, the proposed path planning algorithm enables mobile robots to adapt their navigation strategies effectively, ensuring efficient and reliable performance in complex environments [31].

3.2.1. Three-Room Simulations

1. Simulation for 100 Generations

The transfer learning results show a progressive improvement in average and maximum fitness across generations, with notable peaks indicating successful adaptation by the models. In Figure 26, the average fitness fluctuated around 2000 but saw a significant upward trend after Generation 60, peaking at 4695.94 in Generation 74. The maximum fitness also demonstrated variability, reaching its highest value of 30,360 in Generation 86.

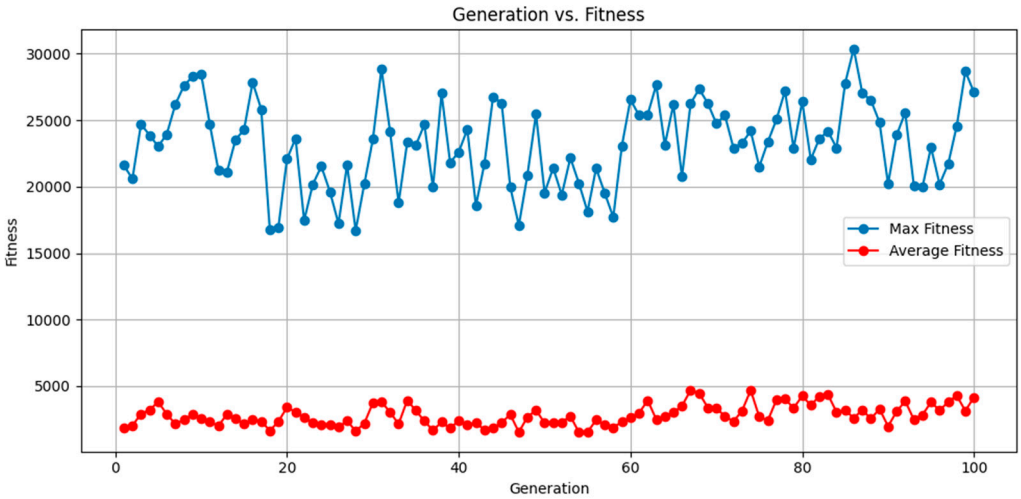


Figure 26. MaxAvg fitness plot for 100 generations.

The heatmap analysis in Figure 27 reveals a concentration of rover activity near the starting point and the first green zone, indicating that while the rovers frequently reach the initial zone, fewer successfully navigate to other zones, highlighting potential areas for further optimization. Over time, the number of rovers visiting all three green zones increases, with Generation 100 showing twelve rovers achieving this milestone and four returning to the starting point.

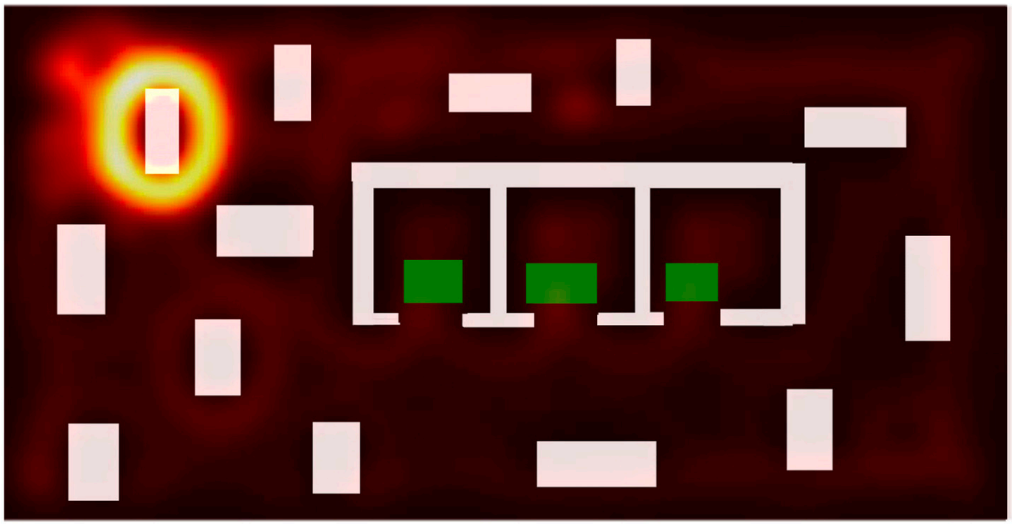


Figure 27. Heatmap for 100 generations.

2. Simulation for 500 Generations

The transfer learning experiment demonstrated a clear upward trend in average and maximum fitness across 500 generations, indicating significant learning and adaptation

over time. Figure 28 shows that the average fitness values, which start around 2000, gradually increase, peaking at approximately 10,756.16 by the final generation. This steady improvement suggests that the models become increasingly proficient at navigating the environment and completing the assigned tasks. The maximum fitness scores also grow substantially, reaching 33,915 in some generations.

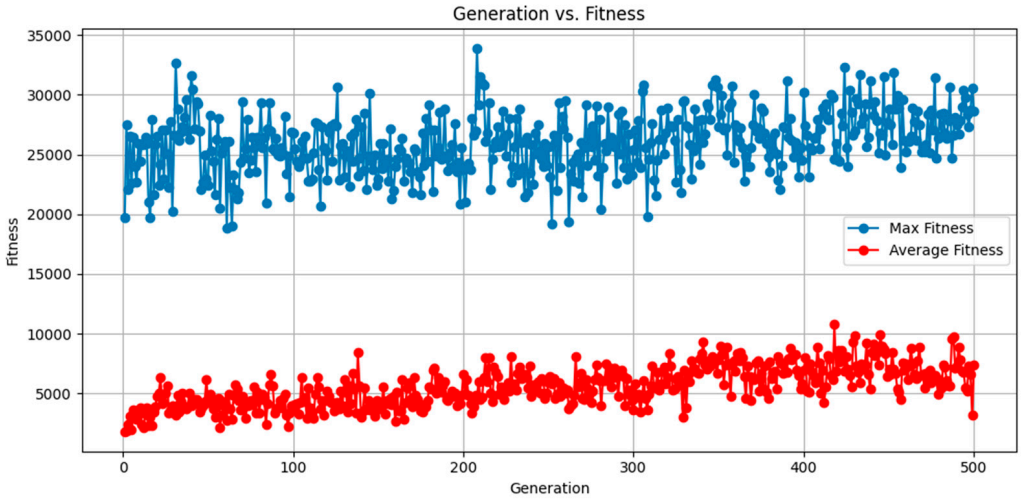


Figure 28. MaxAvg fitness plot for 500 generations.

The heatmap analysis further reveals that the models are highly active near the starting point, successfully navigating towards green zones, albeit with some variations in consistency. From Figure 29, we can observe that throughout the experiment, the number of rovers visiting and returning from all green zones progressively increases, with the highest performance observed in the later generations, where up to 22 rovers complete the task successfully.

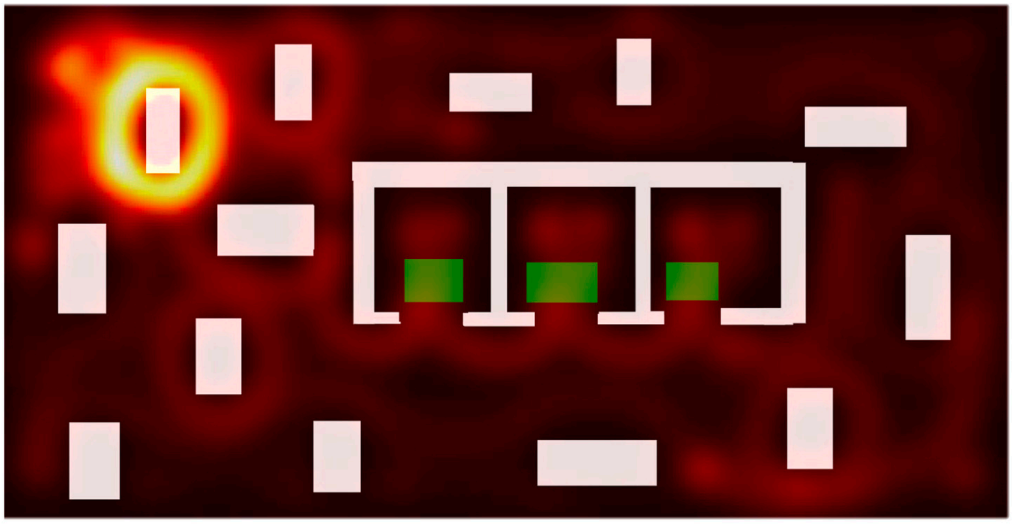


Figure 29. Heatmap for 500 generations.

3. Simulation for 1000 Generations

The simulation results indicate a progressive improvement in the rovers' performance throughout 1000 generations. From the fitness plot in Figure 30, we can conclude that the maximum fitness score reaches 34,192 in the final generation, with the average fitness score at 13,369.02.

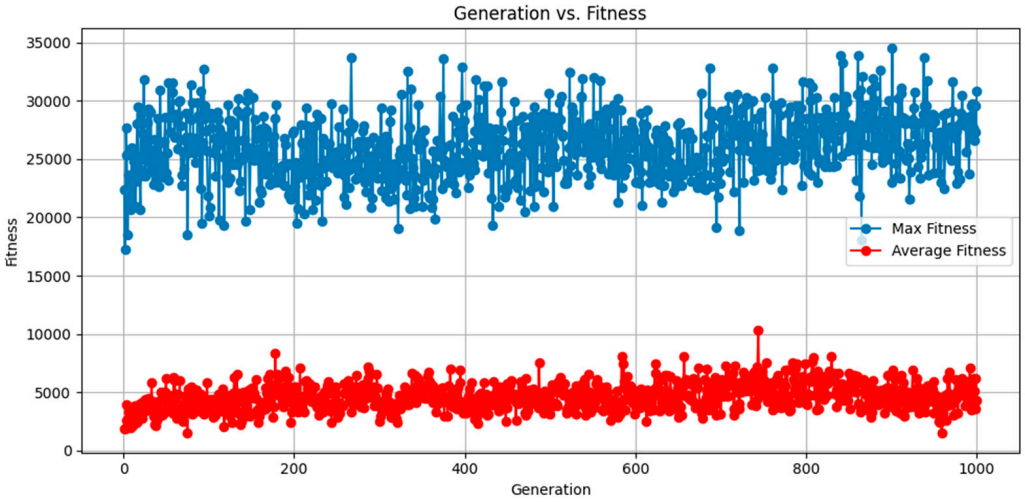


Figure 30. MaxAvg fitness plot for 1000 generations.

The visual heatmaps and fitness graphs reflect these improvements, highlighting areas of the map frequently visited by the rovers and illustrating the gradual enhancement in their navigation strategies. From Figure 31, we can see that in the heatmap throughout the simulation, there is a notable increase in rovers successfully visiting and returning from all three green zones.

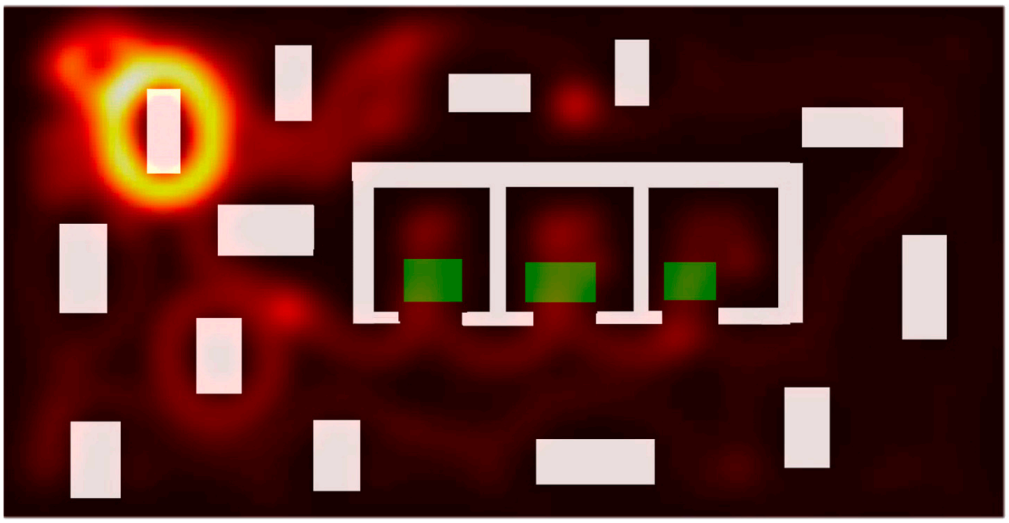


Figure 31. Heatmap for 1000 generations.

4. Simulation for 5000 Generations

The simulation for rover navigation over 5000 generations demonstrates notable improvements in average and maximum fitness. From Figure 32, we can observe that the average fitness values initially fluctuate widely. Still, as the generations progress, the fitness values begin to stabilize and increase, with the average fitness reaching around that in later generations. Similarly, the maximum fitness shows a steady upward trend, with values around 35,337. The number of rovers successfully visiting all three green zones and returning increases over time, with some generations achieving up to 32 successful returns.

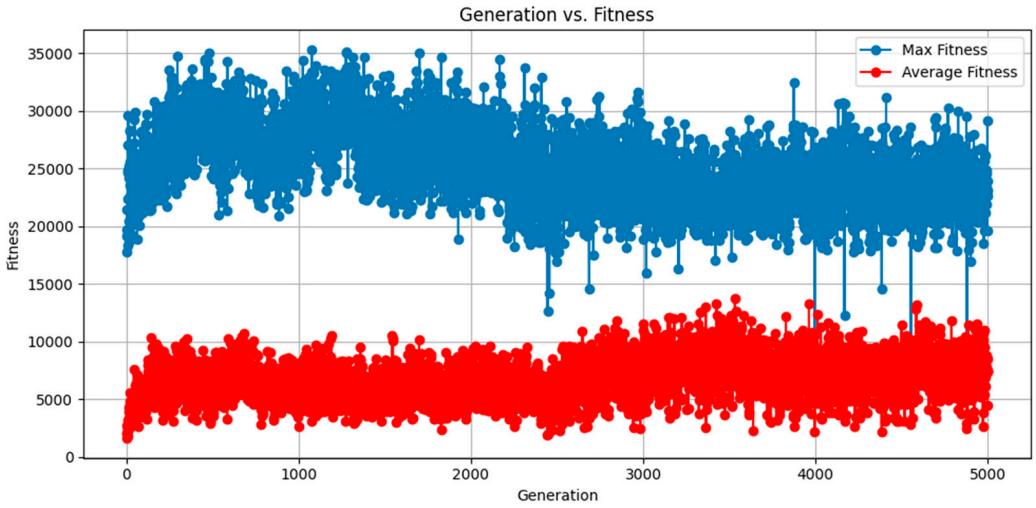


Figure 32. MaxAvg fitness plot for 5000 generations.

The heatmap analysis in Figure 33 highlights critical areas on the map where rovers frequently move, indicating zones of concentrated activity. These results demonstrate the efficacy of NEAT in evolving more competent networks over time, leading to better performance in the rover navigation task.

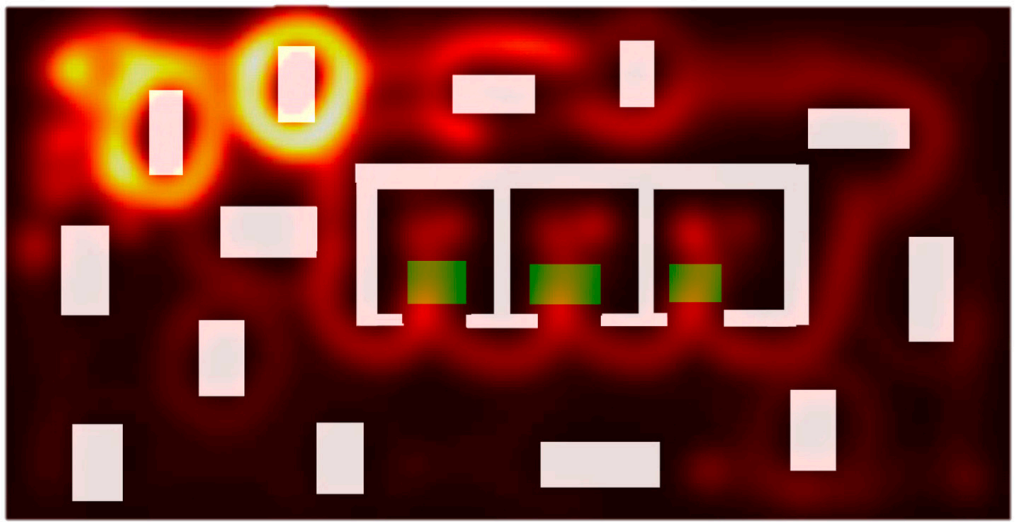


Figure 33. Heatmap for 5000 generations.

5. Simulation for 10,000 Generations

Figure 34's fitness plot shows the simulation results over 10,000 generations, demonstrating a significant evolution in rover performance. The average fitness of the rovers increases from around 2000 in the initial generations to peaks of approximately 14,000 by generation 2000, with a peak average fitness of 14,586.32. The maximum fitness shows consistent high performance, fluctuating between 25,000 and 35,000 throughout the simulation, with occasional peaks reaching as high as 37,503. The number of rovers completing the task of visiting all green zones and returning increases steadily, indicating an effective learning process.

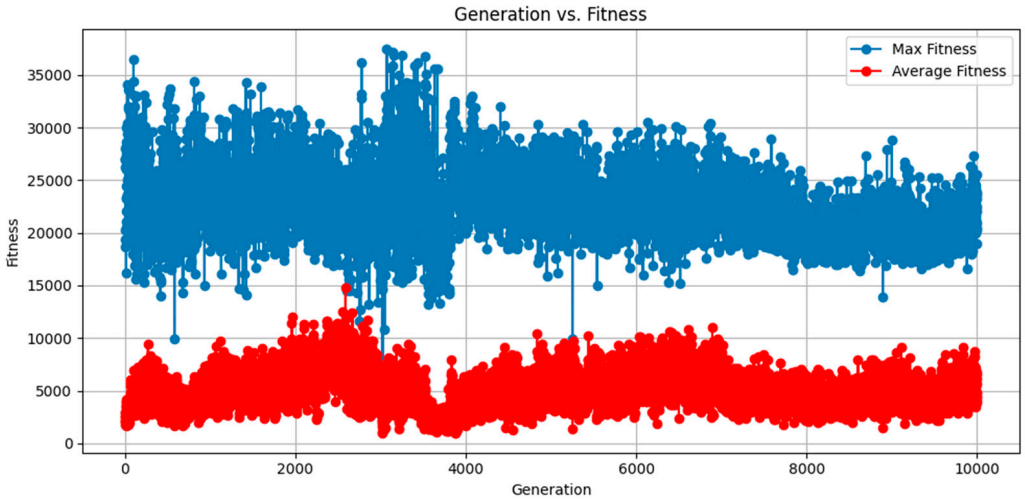


Figure 34. MaxAvg fitness plot for 10,000 generations.

The heatmap visualization in Figure 35 shows high activity concentration near the starting point, indicating that the rovers frequently navigate this area. Additionally, the heatmap reveals clear paths leading to the green zones, reflecting the cars' repeated successful attempts to reach these target areas throughout the simulation.

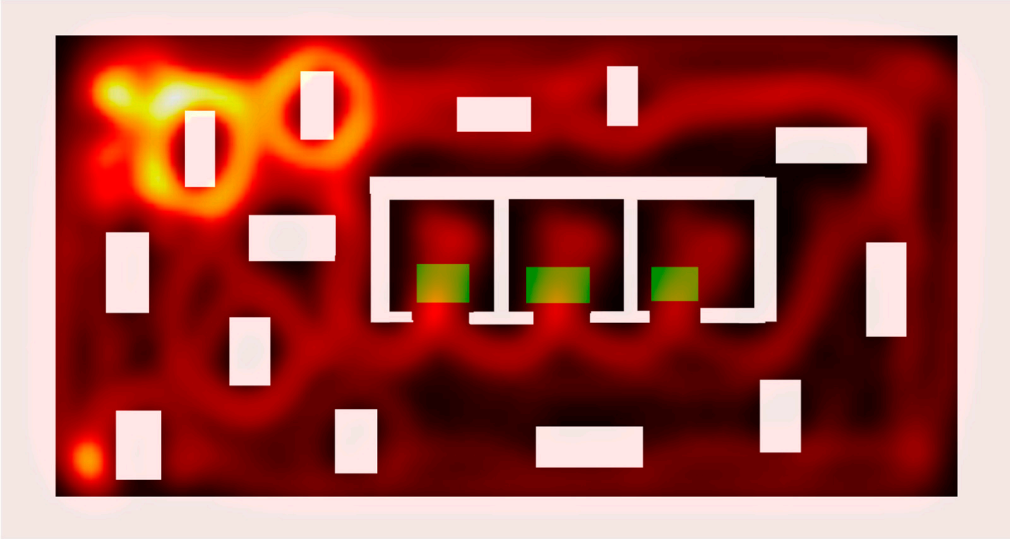


Figure 35. Heatmap for 10,000 generations.

3.2.2. Four-Room Simulations

1. Simulation for 100 Generations

The transfer learning experiment's result indicates significant improvements in fitness metrics and task completion over generations. From Figure 36, we can see that initially, average fitness shows considerable fluctuations but demonstrates a clear upward trend, particularly after Generation 40, peaking at 14,360.26 in Generation 70. Maximum fitness also displays variability, with notable peaks, reaching as high as 34,742 in Generation 94.

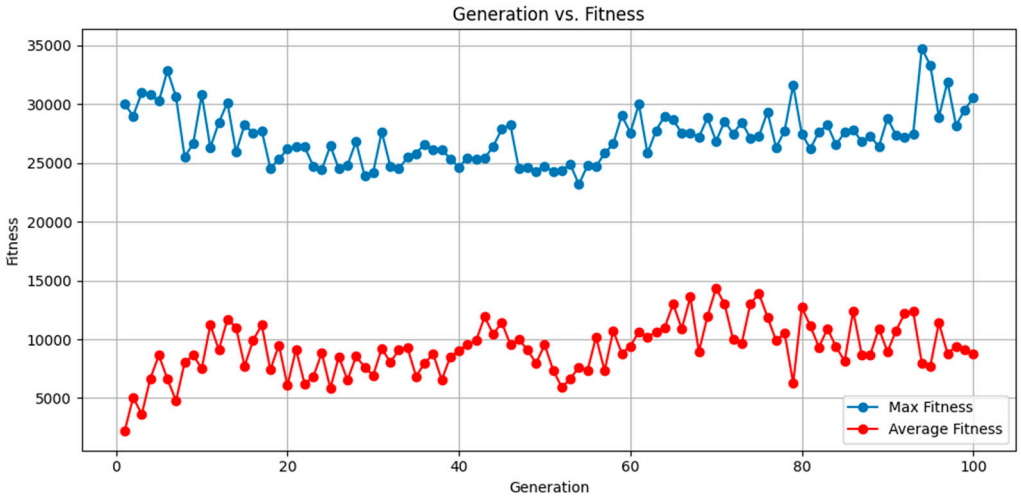


Figure 36. MaxAvg fitness plot for 100 generations.

The heatmap analysis in Figure 37 suggests a concentration of rover activity near the starting point, with successful navigation toward the green zones becoming more consistent over time. The data show that the number of rovers visiting all four green zones increases progressively, with eighteen rovers achieving this milestone in Generation 93, up from just one in early generations. Additionally, many rovers return to the starting point after visiting all zones, indicating improved pathfinding and task completion abilities.

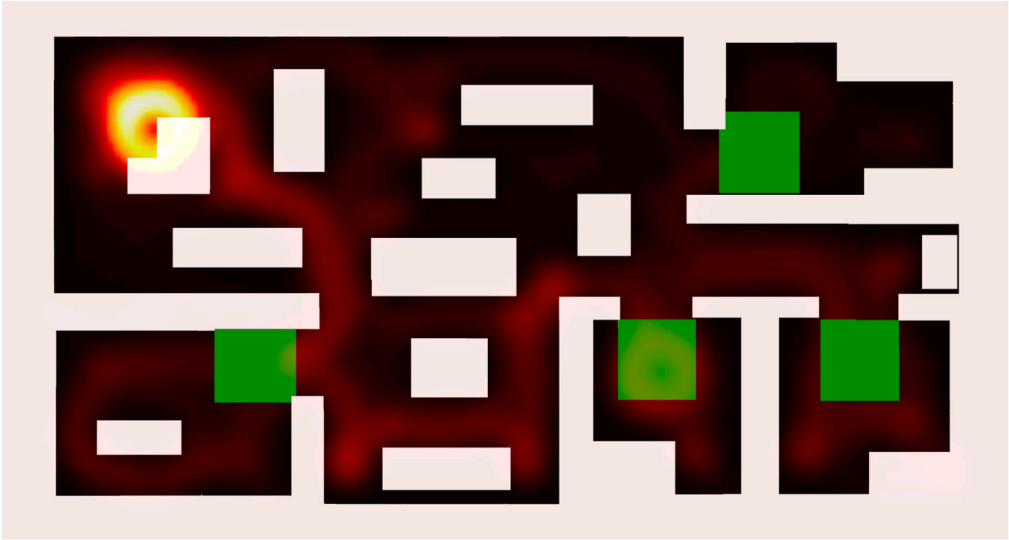


Figure 37. Heatmap for 100 generations.

2. Simulation for 500 Generations

The simulation here shows average and maximum fitness variability across the 500 generations. Figure 38 shows that the average fitness initially increases, peaking around Generation 140 at approximately 13,669.54 before experiencing fluctuations with a general decline towards the end. The maximum fitness displays peaks of up to 36,255 in the later generations.

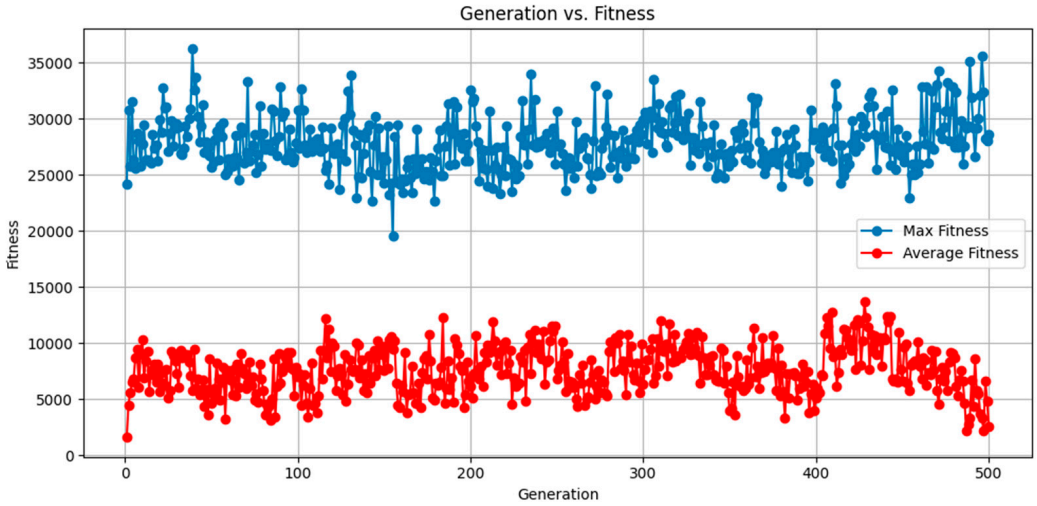


Figure 38. MaxAvg fitness plot for 500 generations.

Figure 39 is a heatmap whose analysis suggests that rover activity is concentrated near the starting point and the first green zone, with some dispersion towards other zones, indicating a learning process but with room for improvement in overall navigation. There is a gradual increase in the number of rovers visiting and returning from all four green zones. By the later generations, the number of rovers completing the task increases, with a notable performance in Generations 409 and 424, where up to sixteen rovers return after visiting all four zones.

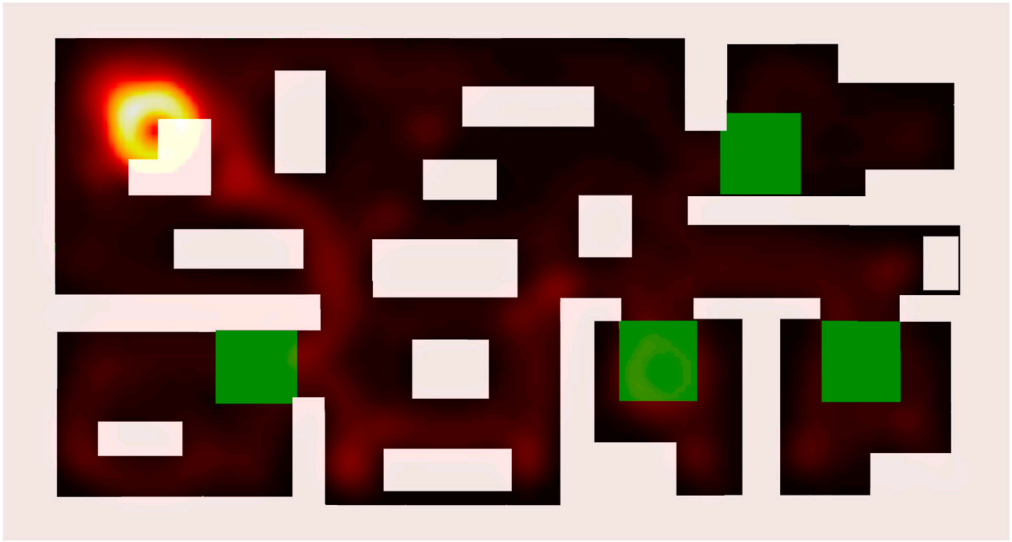


Figure 39. Heatmap for 500 generations.

3. Simulation for 1000 Generations

Figure 40 is a fitness plot of the simulation showing that the rovers' average fitness peaks at approximately 15,834.02 around Generations 300–400, although there are significant fluctuations throughout the 1000 generations. The maximum fitness remains relatively stable, peaking at 36,255, with an early upward trend that plateaus around Generation 300. The number of rovers completing all green zones is highest around Generations 100–200. Still, this performance declines in later generations, which might be caused by frequent rovers going round and round in the same area.

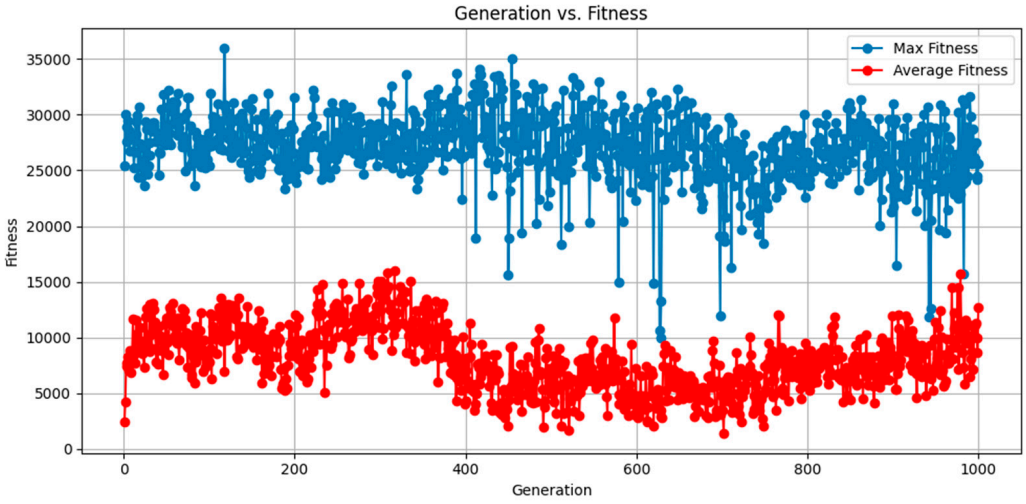


Figure 40. MaxAvg fitness plot for 1000 generations.

The heatmap analysis in Figure 41 reveals high activity concentration near the starting point, indicating that while the rovers explore the environment, they often struggle to consistently reach all areas, mirroring the mixed results in fitness and green zone completion in the latter part of the simulation.

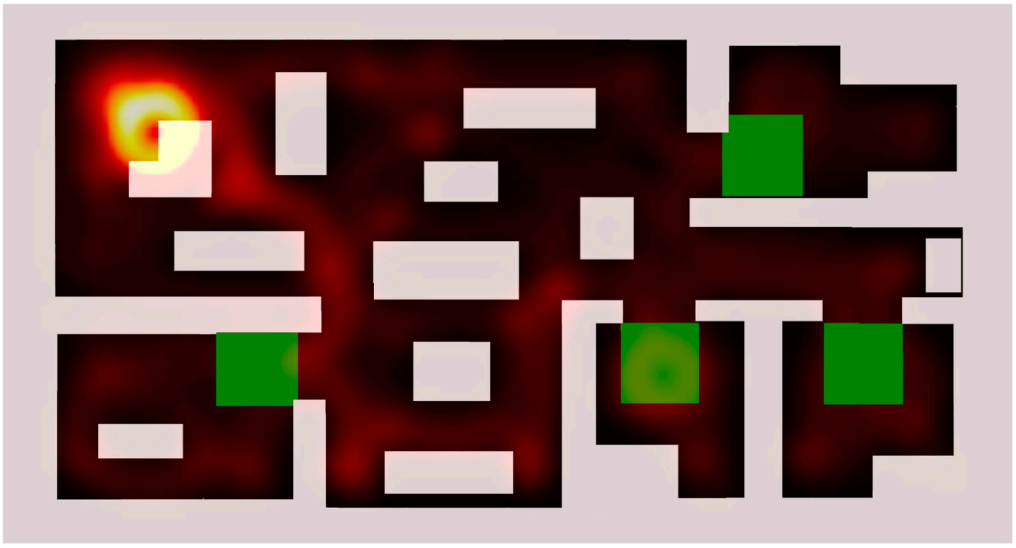


Figure 41. Heatmap for 1000 generations.

4. Simulation for 5000 Generations

The fitness plot from Figure 42 shows that the rovers' average fitness tends to fluctuate significantly over generations. Despite the fluctuations, there is a noticeable trend where the average fitness stabilizes in the latter half of the generations, though it still exhibits considerable variability. The maximum fitness also fluctuates but appears to achieve higher peaks more consistently, often reaching values of 36,848.

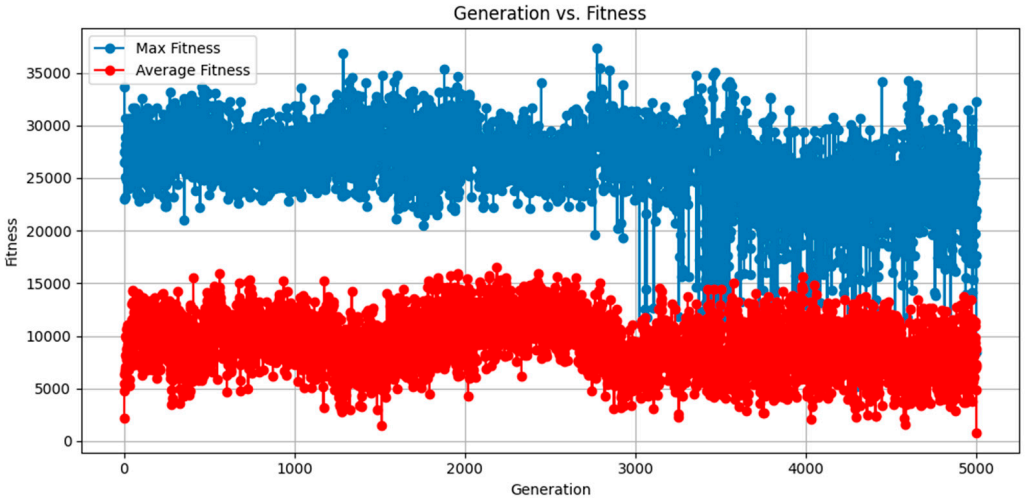


Figure 42. MaxAvg fitness plot for 5000 generations.

Figure 43's heatmap of the environment shows concentrated areas of activity, particularly near the starting zone and around the green zones, indicating that the rovers successfully navigate and interact with these critical areas of the map.

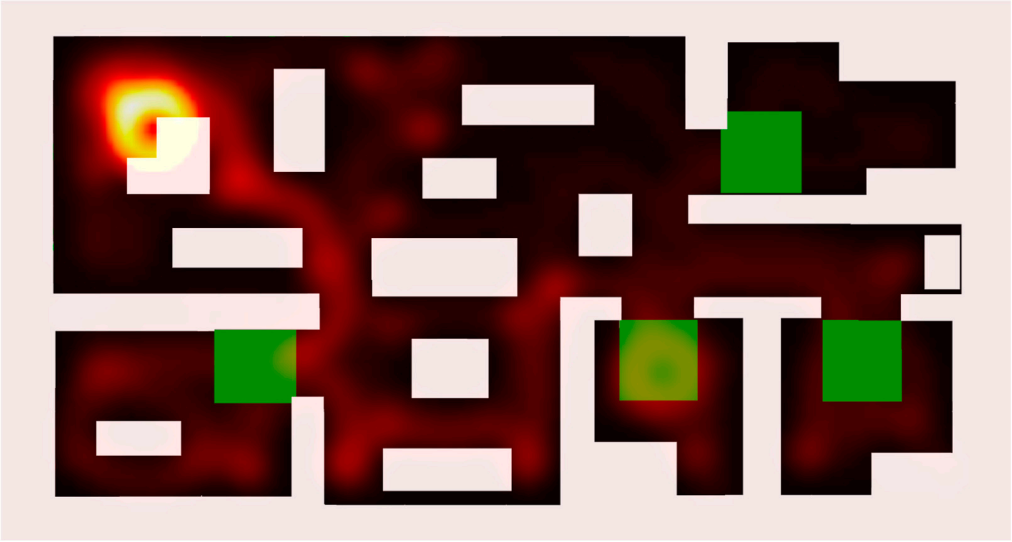


Figure 43. Heatmap for 5000 generation.

5. Simulation for 10,000 Generations

The simulation results indicate a steady progression in the fitness of the rovers across generations. The fitness plot in Figure 44 indicates that the maximum fitness reaches approximately 35,000 by the 10,000th generation, while the average fitness varies between 4000 and 10,000 across the generations. The data show that rovers increasingly visit and return after visiting all four zones, with notable peaks in successful attempts, as seen, for example, with 32 rovers. The heatmap analysis highlights that the regions near the zones and corridors show high activity, indicating frequent vehicle visits, which correlates with the higher fitness values observed in the later generations. This suggests that the evolved strategies effectively guide the rovers through the challenging map, improving their performance over time.

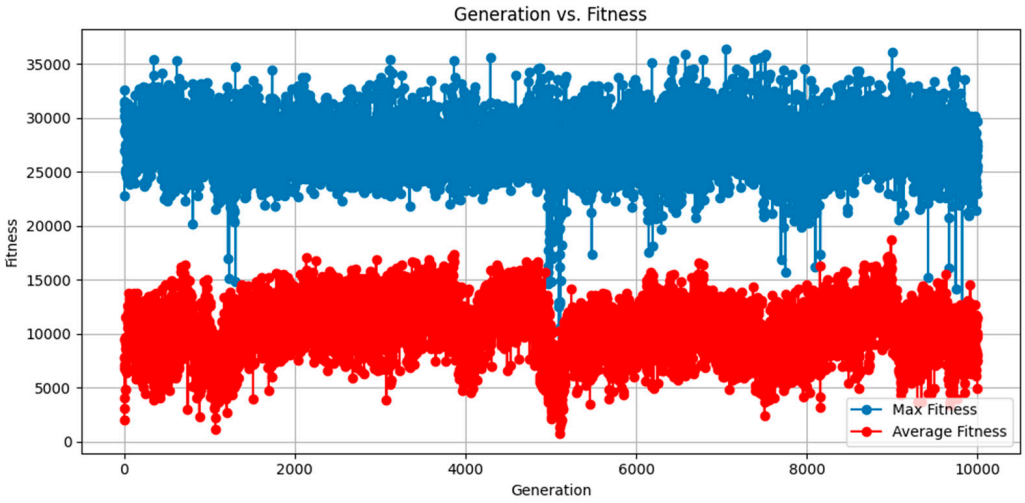


Figure 44. MaxAvg fitness plot for 10,000 generations.

The heatmap in Figure 45 reveals intense activity near the starting point and along the routes leading to the green zones, indicating that the rovers frequently navigate these areas. The bright spots suggest that the rovers spend significant time in these critical areas, optimizing their paths to complete their objectives.



Figure 45. Heatmap for 10,000 generations.

3.3. Summary of Results

Table 1 summarizes the simulation results of rovers evolving over 100, 500, 1000, 5000, and 10,000 generations in a three-room scenario with standard simulation. The data show that as generations progress, the number of rovers successfully visiting all three zones and returning to the starting point significantly increases, from three in Generation 100 to 34 in Generation 10,000. While the highest fitness scores fluctuate slightly, the average maximum fitness improves steadily, indicating that the best-performing rovers become more consistent in their tasks.

Table 1. Standard simulation results in three-room scenarios.

Generation	MaxFitness	Max Avg Fitness	Number of Rovers Inside All 3 Zones (Highest)	Number of Rovers Inside Returned After Visiting All 3 Zones (Highest)
100	38,803	2693.88	5	3
500	37,695	8644	18	15
1000	38,394	9086.44	20	18
5000	39,678	12,653.59	27	26
10,000	37,685	14,426.36	36	34

Table 2 summarizes the results of rovers evolving in a four-room scenario over 100, 500, 1000, 5000, and 10,000 generations using NEAT. As in the three-room scenario, the number of rovers successfully visiting all three zones and returning to the starting point increases with more generations, from eight in Generation 100 to 25 in Generation 10,000. The highest fitness scores gradually improve, and the average maximum fitness shows significant growth, indicating that the rovers become more adept at completing their tasks over time.

Table 2. Standard simulation results in four-room scenarios.

Generation	MaxFitness	Max Avg Fitness	Number of Rovers Inside All 3 Zones (Highest)	Number of Rovers Inside Returned After Visiting All 3 Zones (Highest)
100	34,600	9681.82	10	8
500	35,563	13,069.14	14	13
1000	35,844	15,804.82	18	16
5000	35,894	15,214.27	26	23
10,000	37,605	18,315.3	28	25

When comparing the four-room results to the three-room scenario, the four-room setup demonstrates a slower increase in performance metrics. This suggests that the additional complexity in the four-room environment challenges the rovers, leading to a more gradual improvement in their navigational abilities.

Table 3 shows the results of applying transfer learning in a three-room scenario over 100, 500, 1000, 5000, and 10,000 generations. The transfer learning approach shows marked improvement compared to the standard three-room scenario in Table 1. For instance, in the early generations (100 and 500), the number of rovers successfully visiting all three zones and returning after visiting them is significantly higher with transfer learning.

Table 3. Transfer learning results in the three-room scenario.

Generation	MaxFitness	Max Avg Fitness	Number of Rovers Inside All 3 Zone	Number of Rovers Inside Returned After Visiting All 3 Zones
100	30,360	4695.94	12	4
500	33,915	10,756.16	23	22
1000	34,192	13,369.02	24	18
5000	35,337	14,231.12	34	32
10,000	37,503	14,586.32	33	32

As generations progress, the benefits of transfer learning become more evident, with consistently higher values for both the number of rovers completing the task and the average fitness scores. This suggests that transfer learning enhances the rovers' ability to quickly adapt and perform complex tasks in the environment, resulting in more efficient learning and better overall performance than the standard approach.

Table 4 presents the results of applying transfer learning in a four-room scenario over 100, 500, 1000, 5000, and 10,000 generations. Compared to the standard four-room scenario, transfer learning shows clear advantages in improving rover performance.

Table 4. Transfer learning results in the four-room scenario.

Generation	MaxFitness	Max Avg Fitness	Number of Rovers Inside All 3 Zone	Number of Rovers Inside Returned After Visiting All 3 Zones
100	34,742	14,360.26	20	18
500	36,255	13,669.54	18	16
1000	36,255	15,834.02	25	23
5000	36,848	16,209.23	28	24
10,000	36,404	18,706.46	33	32

In the early generations, such as 100 and 500, the number of rovers successfully visiting all four zones and returning after visiting them is significantly higher with transfer learning. For example, in Generation 100, eighteen rovers returned after visiting all zones compared to just eight in the standard scenario. This trend continues across the generations, with transfer learning consistently achieving higher or comparable maximum fitness and average fitness scores.

By Generation 10,000, the number of rovers returning after visiting all three zones reaches 32 with transfer learning, compared to 25 in the standard scenario. The increase in the maximum average fitness and the number of successful rover completions highlights the effectiveness of transfer learning in enhancing the rovers' ability to navigate and complete tasks in a more complex environment.

4. Discussion

The results of our study demonstrate the effectiveness of using NEAT for evolving autonomous navigation strategies in multi-room environments. The significant improvements in fitness scores and task completion rates across generations in three- and four-room scenarios highlight the algorithm's ability to adapt and optimize rover behavior over time.

The NEAT algorithm, combined with transfer learning, plays a critical role in addressing the challenges of autonomous navigation in dynamic, multi-room environments. Traditional rule-based systems struggle to adapt to unpredictable and complex scenarios where the environment and objectives change dynamically. In contrast, the NEAT algorithm evolves neural network architectures and weights, enabling the system to learn adaptive navigation strategies tailored to the environment. Transfer learning further enhances this adaptability by leveraging pre-trained models to reduce training time and improve performance in new environments.

The comparison between standard and transfer learning can be made more comprehensive by analyzing their learning processes, performance metrics, and contextual advantages. Standard learning starts from scratch, requiring significant time and computational resources for optimal performance. In contrast, leveraging pre-trained models, transfer learning demonstrates faster initial adaptation and higher efficiency, especially in early generations. The results show that transfer learning consistently achieves higher

fitness scores and more successful task completions in scenarios with environmental similarities to the pre-trained model, as highlighted by heatmap analyses indicating more focused and efficient navigation patterns. However, standard learning avoids the potential biases of pre-trained models in novel or highly distinct environments, allowing better generalization. In more complex setups, such as four-room scenarios, transfer learning remains efficient but may exhibit diminishing returns, suggesting the need for refinement or hybrid approaches. This discussion highlights that transfer learning excels in familiar environments and early adaptation. Standard learning is better suited to highly diverse or novel contexts, providing a clear understanding of when each method should be applied.

The performance difference between three- and four-room scenarios indicates that environmental complexity plays a crucial role in learning. The slower improvement rate in the four-room scenario underscores the challenges posed by increased spatial complexity and the need for more sophisticated navigation strategies. Heatmap analyses provide valuable insights into rover behavior, showing concentrated activity near starting points and green zones. This pattern suggests that the evolved strategies effectively prioritize critical areas of the environment, balancing exploration with task completion. Fitness score fluctuations arise from mutations' stochastic nature and NEAT crossover. These fluctuations indicate the exploration of diverse strategies essential for avoiding premature convergence. However, excessive variability may suggest suboptimal parameter settings, warranting further tuning. These variations might indicate the presence of local optima in the solution space or the need to fine-tune the NEAT parameters to achieve a more stable performance.

Although our simulation results are promising for both NEAT and transfer learning as approaches for developing autonomous navigation strategies to deal with complex multi-room environments, we realize that no simulation can capture all the complexities of the real world. Sensor noise, actuator imperfections, environmental variations, and unmodeled obstacles can easily make a well-working robot perform poorly in the physical world. The following steps are crucial for transferring simulation results to real-world scenarios:

- Sensor Integration: to replicate the environmental perception simulated in the study, real-world robots must be equipped with robust sensors (e.g., LIDAR, cameras, and IMUs).
- Algorithm Adaptation: the NEAT algorithm must handle real-time processing and adjust to physical constraints, such as battery life and motor precision.
- Physical Validation: Testing the algorithm on physical robots in controlled environments, followed by deployment in operational scenarios, will ensure its reliability and robustness. This transition is essential for validating the applicability of our findings to tasks such as firefighting and search and rescue missions.

The simulation results are estimated to be valid for approximately 60–70% of real-world scenarios because of their controlled nature. However, simulations may fail to capture all the dynamic and stochastic elements in natural environments. Therefore, the further testing of the proposed algorithms for effectiveness and robustness requires implementation on physical robotic platforms.

Integrating NEAT with unmanned aerial vehicle (UAV) systems offers transformative potential for wildfire mapping and management by enabling UAVs to navigate and adapt to dynamic fire conditions autonomously. NEAT's ability to evolve neural networks empowers UAVs to optimize flight paths in real time, avoiding hazards such as smoke and intense heat zones while ensuring maximum area coverage. This adaptability enhances wildfire map accuracy through continuous and reliable data collection, even in challenging environments. Equipped with NEAT-evolved decision-making algorithms, UAVs can

rapidly analyze extensive areas to characterize fuel loads with high precision, using data from thermal imaging, LiDAR, and multispectral sensors. This enables UAVs to identify high-risk regions, supporting targeted mitigation strategies and providing actionable insights for wildfire prevention.

NEAT-enabled UAV swarms can be deployed collaboratively in wildfire scenarios for effective management. These UAVs can monitor fire boundaries, providing real-time updates to command centers. At the same time, NEAT's adaptive algorithms determine optimal areas for retardant deployment based on current fire behavior and environmental conditions. Furthermore, NEAT-guided UAVs can navigate complex terrains, autonomously adjust flight parameters, and locate individuals in distress, even in regions with limited visibility or accessibility. This combination of adaptability, precision, and collaborative capabilities positions NEAT-integrated UAV systems as critical tools for wildfire response and mitigation.

The NEAT algorithm, while innovative, faces several challenges, including increasing computational demands as neural networks evolve, the risk of premature convergence to local optima requiring careful parameter tuning, difficulties in managing sensor noise and dynamic environments that simulations cannot fully replicate, and scalability limitations, as its effectiveness in more significant, more complex scenarios remains an area for further study.

In future work, we plan to implement our evolved neural networks on real robots with appropriate sensors and actuators. This includes adapting algorithms for real-time processing, hardware limitations, and safety in unstructured environments. Physical robot testing will provide valuable insights into the algorithms' practical applicability and help identify further adjustments necessary for performance improvement. By conducting physical experiments, we aim to bridge the gap between simulation and real-world application, enhancing the reliability of autonomous navigation systems for critical tasks such as firefighting and search and rescue missions [32].

5. Conclusions

This study demonstrates the potential of NEAT and transfer learning in developing autonomous navigation systems for complex, multi-room environments. The significant improvements in rover performance across generations, particularly with transfer learning, suggest that these approaches could be valuable in real-world applications such as firefighting and search and rescue operations.

The superior performance of transfer learning, especially in early generations, highlights its potential to accelerate the development of effective navigation strategies in new environments. This could be particularly beneficial in time-critical applications where rapid adaptation is crucial.

Integrating NEAT can revolutionize critical aspects of fire management, addressing key challenges with advanced predictive modeling, precise fuel load characterization, and highly detailed risk assessments. By enabling autonomous rovers to operate safely and efficiently in hazardous terrains, NEAT facilitates real-time monitoring and analytics, enhancing situational awareness and optimizing resource allocation during wildfire events. Future efforts will focus on validating these applications in realistic wildfire scenarios, bridging the gap between simulation and practical deployment to ensure robust and effective fire management solutions.

While the simulation results are promising, we acknowledge the necessity of validating our approach with physical robots. Implementing the algorithms on actual robotic systems will provide a more comprehensive understanding of their capabilities and limitations. This step is crucial for advancing towards real-world applications where reliability and robust-

ness are paramount. The rover consideration for the developed autonomous navigation system is being considered from the work in [33]. Future work should focus on enhancing the robustness of the evolved strategies, incorporating more realistic environmental factors, and testing the transferability of learned behaviors to physical robotic systems.

Similarly, integrating NEAT with UAV systems presents transformative opportunities for advancing wildfire management. This synergy holds promise for enhancing mapping accuracy, improving fuel load characterization, and enabling collaborative strategies in fire suppression and search-and-rescue operations. By leveraging the innovative potential of machine and deep learning techniques in emergency response, NEAT-driven UAV systems demonstrate significant applicability for addressing the complexities of wildfire scenarios. Future research will validate these capabilities through field experiments and expand the scope of NEAT-enabled UAV applications to address a broader range of wildfire contexts and challenges.

In conclusion, this research study contributes to the ongoing development of autonomous navigation systems, offering insights into using evolutionary algorithms and transfer learning for complex spatial reasoning tasks. The findings pave the way for more adaptive and efficient robotic systems capable of navigating challenging environments, with potential applications across various domains, including emergency response and exploration.

Author Contributions: Conceptualization, D.V.; methodology, D.V. and D.S.; software, D.S.; validation, D.V. and D.S.; formal analysis, D.V. and D.S.; investigation, D.S.; resources, D.V.; data curation, D.S.; writing—original draft preparation, D.S.; writing—review and editing, D.V.; visualization, D.S.; supervision, D.V.; project administration, D.V.; funding acquisition, D.V. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The authors will make the raw data supporting this article's conclusions available upon request.

Acknowledgments: The Texas State University's Translational Health Research Center contributed to this study's support.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Siegwart, R.; Nourbakhsh, I.R.; Scaramuzza, D. *Introduction to Autonomous Mobile Robots*, 2nd ed.; MIT Press: Cambridge, MA, USA, 2011.
2. Thrun, S.; Burgard, W.; Fox, D. *Probabilistic Robotics*; MIT Press: Cambridge, MA, USA, 2005.
3. LaValle, S.M. *Planning Algorithms*; Cambridge University Press: Cambridge, UK, 2006.
4. Nagatani, K.; Kiribayashi, S.; Okada, Y.; Otake, K.; Yoshida, K.; Yoshida, T.; Koyanagi, E.; Fukushima, M. Emergency response to the nuclear accident at the Fukushima Daiichi Nuclear Power Plants using mobile rescue robots. *J. Field Robot.* **2013**, *30*, 44–63. [CrossRef]
5. Jennings, J.S.; Whelan, G.A.; Evans, B.A. Cooperative robot navigation for multi-robot search and retrieval. *Electron. Lett.* **1997**, *33*, 99–101.
6. Jacoff, A.; Messina, E.; Evans, J.; Michalos, G.; Tadokoro, S.; Jacoff, A.; Messina, E. Emergency response robot evaluation exercise. In Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction, Boston, MA, USA, 5–8 March 2012; pp. 593–600.
7. Merino, L.; Caballero, F.; Martínez-de-Dios, J.R.; Ollero, A.; Viegas, D.X. An unmanned aircraft system for automatic forest fire monitoring and measurement. *J. Intell. Robot. Syst.* **2012**, *65*, 533–548. [CrossRef]

8. Marina, D.; Elia, M.; Giannico, V.; Spano, G.; Lafortezza, R.; Sanesi, G. Machine Learning Techniques for Fine Dead Fuel Load Estimation Using Multi-Source Remote Sensing Data. *Remote Sens.* **2021**, *13*, 1658. [CrossRef]
9. CMWF. "Machine Learning Ignites Wildfire Forecasting," Science Blog. 2024. Available online: https://www.ecmwf.int/en/about/media-centre/science-blog/2024/machine-learning-ignites-wildfire-forecasting (accessed on 21 December 2024).
10. Olson, E.; Stanley, J.; Rubinoff, R.; Chizeck, H.J.; Johnson, E.; Goldberg, K. Towards automated fire line mapping from fixed wings. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Las Vegas, NV, USA, 27–31 October 2003.
11. Feng, A.; Xie, Y.; Sun, Y.; Wang, X.; Jiang, B.; Xiao, J. Efficient Autonomous Exploration and Mapping in Unknown Environments. *Sensors* **2023**, *23*, 4766. [CrossRef] [PubMed]
12. Stanley, K.O.; D'Ambrosio, D.B.; Gauci, J. A hypercube-based encoding for evolving large-scale neural networks. *Artif. Life* **2009**, *15*, 185–212. [CrossRef] [PubMed]
13. Zhao, Y.; Yan, L.; Xie, H.; Dai, J.; Wei, P. Autonomous Exploration Method for Fast Unknown Environment Mapping by Using UAV Equipped With Limited FOV Sensor. *IEEE Trans. Ind. Electron.* **2024**, *71*, 4933–4943. [CrossRef]
14. Ersü, C.; Petlenkov, E.; Janson, K. A Systematic Review of Cutting-Edge Radar Technologies: Applications for Unmanned Ground Vehicles (UGVs). *Sensors* **2024**, *24*, 7807. [CrossRef] [PubMed]
15. Feng, X.; Cong, H.; Zhang, Y.; Qiu, M.; Hu, X. Autonomous Exploration Method of Unmanned Ground Vehicles Based on an Incremental B-Spline Probability Roadmap. *Sensors* **2024**, *24*, 3951. [CrossRef] [PubMed]
16. Ali, M.; Jardali, H.; Roy, N.; Liu, L. Autonomous Navigation, Mapping and Exploration with Gaussian Processes. In Proceedings of the Robotics: Science and Systems 2023, Daegu, Republic of Korea, 10–14 July 2023.
17. Benciolini, T.; Tang, C.; Leibold, M.; Weaver, C.; Tomizuka, M.; Zhan, W. Active Exploration in Iterative Gaussian Process Regression for Uncertainty Modeling in Autonomous Racing. *IEEE Trans. Control. Syst. Technol.* **2024**, 1–16. [CrossRef]
18. Jadidi, M.G.; Miro, J.V.; Dissanayake, G. Gaussian processes autonomous mapping and exploration for range-sensing mobile robots. *Auton. Robot.* **2016**, *42*, 273–290.
19. Saska, M.; Vonasek, V.; Krajnik, T.; Preucil, L. Autonomous deployable A-robot and UAV teams for situational modeling. In Proceedings of the Conference Towards Autonomous Robotic Systems (TAROS), Birmingham, UK, 1–3 September 2014; pp. 284–295.
20. Risi, S.; Stanley, K.O. Enhancing ES-HyperNEAT to Evolve More Complex Regular Neural Networks. *Artif. Life* **2012**, *18*, 331–365. [CrossRef] [PubMed]
21. Rasheed, M. Can NEAT Really Create Self-Learning Systems that Surpass Human Intelligence? Available online: https://www.linkedin.com/pulse/can-neat-really-create-self-learning-systems-surpass-rasheed-/ (accessed on 18 August 2024).
22. Wiegand, N.; Fairbank, M. Safe Crossover of Neural Networks through Neuron Alignment. *Neural Netw.* **2023**, *154*, 213–225. [CrossRef]
23. Mouret, J.-B.; Doncieux, S. Using Behavioral Exploration Objectives to Solve Deceptive Problems in Neuro-Evolution. In Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation, Montreal, QC, Canada, 8–12 July 2009; pp. 627–634. [CrossRef]
24. Stanley, K.O.; Miikkulainen, R. Evolving Neural Networks Through Augmenting Topologies. *Evol. Comput.* **2002**, *10*, 99–127. [CrossRef] [PubMed]
25. Gomes, J.; Mariano, P.; Christensen, A.L. Neural Networks as Controllers for Robot Swarms. In Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation, Amsterdam, The Netherlands, 6–10 July 2013; pp. 519–526. [CrossRef]
26. Mouret, J.-B.; Doncieux, S. Encouraging Behavioral Diversity in Evolutionary Robotics: An Empirical Study. *Evol. Comput.* **2012**, *20*, 91–133. [CrossRef] [PubMed]
27. Wilke, C.O. *Fundamentals of Data Visualization: A Primer on Making Informative and Compelling Figures*; O'Reilly Media: Sebastopol, CA, USA, 2019.
28. Ryo, S.; Adnan, K.; Tian, X.; Hooman, H.; Nicolai, M. Augmented Reality and Robotics: A Survey and Taxonomy for AR-enhanced Human-Robot Interaction and Robotic Interfaces. In Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems (CHI '22). Association for Computing Machinery, New York, NY, USA, 29 April 2022; Article 553, pp. 1–33. [CrossRef]
29. Shrestha, D.; Valles, D. Evolving Autonomous Navigation: A NEAT Approach for Firefighting Rover Operations in Dynamic Environments. In Proceedings of the 2024 IEEE International Conference on Electro Information Technology (eIT), Eau Claire, WI, USA, 30 May–1 June 2024; pp. 247–255. [CrossRef]
30. Taylor, M.E.; Stone, P. Transfer Learning for Reinforcement Learning Domains: A Survey. *J. Mach. Learn. Res.* **2009**, *10*, 1635–1685. Available online: https://www.jmlr.org/papers/volume10/taylor09a/taylor09a.pdf (accessed on 9 January 2025).
31. Zhang, Y.; Dong, B.; Zhang, L.; Liu, J.; Han, H. Path Planning Algorithm for Mobile Robots in Complex Obstacle Environments. *J. Field Robot.* **2019**, *36*, 972–987. [CrossRef]

32. Zhang, Y.; Liu, J.; Wang, Z.; Chen, H. Bridging the Gap Between Simulation and Real-World Applications in Robotic Systems: Challenges and Solutions. *Sensors* **2019**, *19*, 4956.
33. Rafiq, S.A.; Ellsworth, E.S.; Resendiz, O.; Varanasi, S.H.; Ishola, A.A.; Koldenhoven, R.; Farrell, J.W.; Li, Y.; Resendiz, M.D.; Aslan, S.; et al. Design of Autonomous Rover for Firefighter Rescue: Integrating Deep Learning with ROS2. In Proceedings of the 2024 IEEE World AI IoT Congress (AIIoT), Seattle, WA, USA, 29–31 May 2024; pp. 421–428. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.