Integrated Information in Genetically Evolved Braitenberg Vehicles

Hongju Pae[1(✉)] and Jeffrey L. Krichmar[1,2]

[1] Department of Cognitive Sciences, University of California, Irvine, Irvine, CA 92697, USA
hjpae@uci.edu, jkrichma@uci.edu
[2] Department of Computer Science, University of California, Irvine, Irvine, CA 92697, USA

Abstract. Integrated information, denoted as Φ, quantifies the intrinsic information within causal systems. Despite its profound theoretical implications, applications of Φ have mostly taken place in simulations of arbitrary systems, particularly in terms of biological realism. This study applies Φ calculations to biologically inspired robotic agents that adapt to environmental conditions, thus providing a novel context for observing changes in information integration. The agents' neural network is evolved to demonstrate behavior similar to Braitenberg's Vehicles. The neuromechanical design of these evolved agents are then suitable for Φ analysis. Interestingly, early generations had higher Φ values. In later generations the diversity of connection weights and the Φ values decreased, leading to simpler and more reactive neural activations.

Keywords: Integrated information · Neurorobotics · Artificial neurobiological systems · Genetic evolutionary algorithm

1 Introduction

Integrated information, Φ, quantifies the intrinsic causal information of a network system, by measuring the information above and beyond the sum of its parts [1,13,14]. Inspired by subjective consciousness, Φ aims to measure information intrinsic to the system. Despite its intriguing motivation, practical applications in neurobiological systems are rare due to the complexity of calculation. Previous simulation studies primarily focused on verifying its theoretical aspects, using logic gates to represent causal relationships rather than actual neuronal wiring [2,5,11]. However, the emergence of conscious processes that Integrated Information Theory aims to explain through Φ fundamentally occurs within biological systems. Thus, our study shifts the focus towards biological plausibility by calculating Φ in biologically inspired systems. We aim to compare information integration across behavioral scenarios using robotic agents that show adaptive behaviors.

O. Brock and J. Krichmar (Eds.): SAB 2024, LNAI 14993, pp. 236–247, 2025.
https://doi.org/10.1007/978-3-031-71533-4_18

In this study, the neurobiological robots are modeled on the Braitenberg vehicle, a biological agent introduced by Valentino Braitenberg in his book *Vehicles: Experiments in Synthetic Psychology* [3]. Although these vehicles feature a relatively simple structure with two light sensors connected to two motor wheels, they exhibit distinct behavioral patterns based on the neuron's firing and wiring mechanisms, which aligns well with the study's objectives. The complexity within these vehicles is sufficient for Φ calculations, making them suitable for simulations that examine the relationship between biological functionality and Φ.

Evolutionary computation will be used to optimize the behavioral patterns of these robots by altering the connection weights. Based on the genetic representation and fitness function, starting from randomly generated genes and selection towards higher fitness, this process is also can be considered as supporting the biological basis of the simulation [9]. This study will utilize artificial neural networks to build robots, using the connectivity weights of neural networks as the genetic representation to simulate the evolution toward the behavior of Braitenberg vehicles. This lets us explore the comparison of connectivity weights and integrated information in a neural network system.

The key contributions of this study are as follows: 1) Identify the relationship among the functional behavior of biologically inspired agents and Φ. 2) Identify the relationship between the connectivity weights selected by the evolutionary algorithm and the Φ values.

2 Methods

2.1 Design of the Neural Network

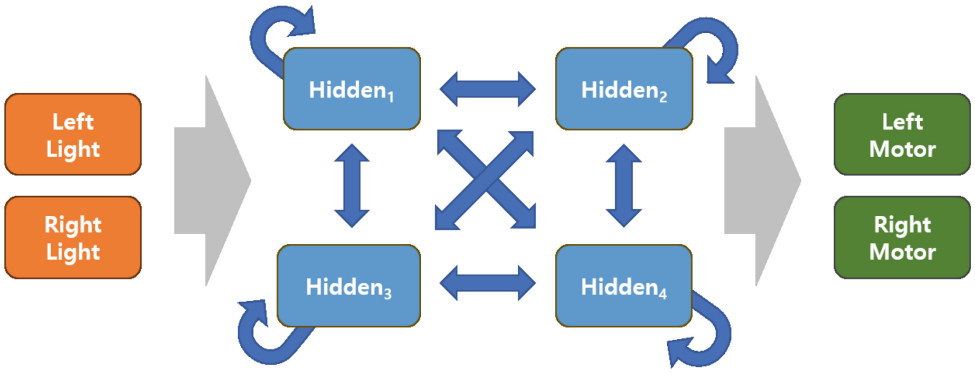

Fig. 1. An overview of the neural network architecture of simulated robots.

The robot's neural network architecture comprises 8 neurons: 2 neurons in the input layer, 4 neurons in the hidden layer, and 2 neurons in the output layer. Each input neuron is connected to all 4 neurons in the hidden layer, and each output neuron is also connected to these 4 hidden layer neurons. The hidden

layer neurons are fully-connected, including self-connections. The input neurons function as sensors detecting light, with their activation directly proportional to the intensity of light detected from the left side and right side. The output neurons control the motors of the robot, corresponding to the motors on either side of a Braitenberg vehicle. The neurons in the hidden layer receive activation signals from the input layer and transmit them to the output layer. Here, the hyperbolic tangent (tanh) activation function is used for all neurons. All connections are weighted connections that are free to evolve to excitatory (positive) or inhibitory (negative) values. These weights correspond to the genome in the evolving process, and are determined through the following evolutionary computation process to emulate the functions of the Braitenberg vehicle (Fig. 1).

2.2 Evolving the Vehicles

The robots were evolved to mimic Braitenberg vehicles for fear and aggression (2A and 2B) and for lover and explorer (3A and 3B) [3]. A trial consisted of 200 movements in the Webots environment [16] during which the robot's position was collected. A trial began with the robot 0.75 m from the light source and from a position either 0.2 m to the left or right of the arena's center. Figure 2 shows the starting position of the trial on the Webots environment.

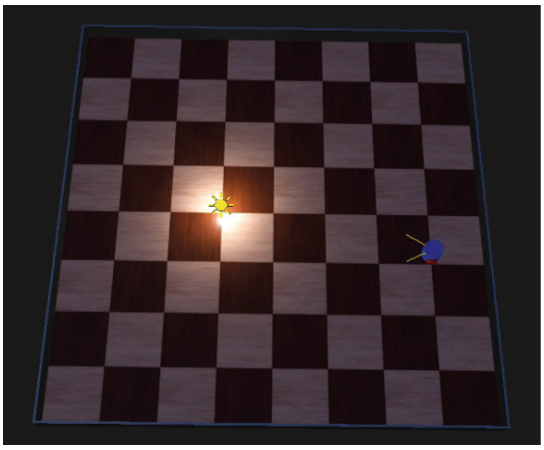

Fig. 2. Rendering of the simulated vehicle and arena under Webots environment.

Evolutionary computation was used to shape vehicle behavior [9]. The *genome* consisted of the neural network connection weights and was initialized with a uniform random distribution between -1 to 1. A population contained 100 individuals. After all individuals completed the two trials, the *fitness* was calculated for each individual. Fitness was judged by how closely the trajectory of an evolved agent matched an ideal agent by calculating the overall mean squared error (MSE) between trajectories.

The ideal agent was based on the vehicle simulations described in Chap. 1 of [6]. It has the input layer (light sensor) directly connected to the output layer (motor) without any hidden layers. The core part of the underlying code is as follows.

```
# Vehicle 2A
leftSpeed = leftLight
rightSpeed = rightLight
```

```
# Vehicle 2B
leftSpeed = rightLight
rightSpeed = leftLight
```

```
# Vehicle 3A
leftSpeed = maxLight - leftLight
rightSpeed = maxLight - rightLight
```

```
# Vehicle 3B
leftSpeed = maxLight - rightLight
rightSpeed = maxLight - leftLight
```

From the given pseudocode, `Speed` refers to the motor's speed, and `Light` refers to the activation of the light sensor. Vehicle 2 had excitatory connections and Vehicle 3 had inhibitory connections from the light sensors to the robot's motors. As a result, 2A and 3B will trace a path heading away from the light source, and 2B and 3A will trace a path heading toward the light source. [7] further provides simulation code of Braitenberg vehicles.

If an individual was successful, that is, its fitness had a lower MSE than the previous generation, it survived into the next generation. Otherwise, it was deleted and replaced with the prior individual. A full evolutionary run or *set* consisted of 1000 *generations* at which point asymptotic population performance was observed. The parameters for the evolutionary algorithm were chosen based on a series of pilot runs and guidance from [9].

After a generation, agents were subjected to *crossover* and *mutation*. Multipoint crossover was carried out by selecting the top 25% as parents, which would then generate 10 children from the remaining 75% of the population. Both parents and children were drawn from a uniform random distribution. A child contained 50% of each parent's genome as drawn from a uniform random distribution. After the crossover operation, mutations were carried out on 10% of the genes for all individuals in the population. The gene to be mutated was drawn from a random distribution. A mutation consists of taking the existing gene, a connection weight, and altering it by drawing the new gene from a normal distribution with a mean of the old gene and a standard deviation. To promote exploration, the standard deviation varied. On the first generation and every 200 generations, the standard deviation was set to 1. For each subsequent generation, the standard deviation was either increased by 0.01 if the number of successful individuals was greater than 20 (5% of the population), or decreased otherwise. This had the effect of initially exploring the evolutionary landscape and later refining when the population as close to a solution [9]. The standard deviation was kept between 0.1 and 1.0.

A total of 5 sets for each vehicle type (2A, 2B, 3A, 3B) were simulated. For each set, individuals of the top 25% in a generation, that is, 25 individuals in order of high fitness, were selected as the *representative individuals* for that generation.

2.3 Calculating Φ

The calculations and descriptions of Φ in this study are based on IIT v3.0. Φ is derived from the informational difference between a system's transition probabilities when a given connection exists versus when it does not [13,14]. Since a network's connection shows whether a particular node would contribute to another node's state change, the information from each connection can be retrieved by selecting only a subset of nodes out of the system and dropping out the rest. This results in *partitioning* the network, cutting the connection between selected subsets and dropped nodes.

In IIT, the nodes of interest are called *mechanism*, and the subset of the system as *purview*. Among all possible purviews, the smallest informational distance is defined as the irreducible intrinsic information of the mechanism.

$$intrinsic\_information = min(D(p(system_{unpartitioned})||p(system_{partitioned})\ ))$$

Here, iteration of the same calculation over every possible subset selection inside the mechanism and purview is performed (i.e. partitioning the mechanism and purview), and the specific partitioning that results in the smallest informational distance is the Minimum Information Partition (MIP). Then, the largest informational distance between the purview under MIP and the mechanism is defined as ϕ (small phi) of the specific mechanism.

$$MIP_{mechanism} \sim min(D(p(mechanism_{partitioned})||p(purview_{partitioned})))$$

$$\phi_{mechanism} := max(D(p(mechanism_{unpartitioned})||p(purview_{MIP})))$$

Now the same iteration is done for the original system. The partitioning over the original system, not just over mechanisms and purviews, is performed to discover the MIP of the system level. The difference is that the sum of $\phi_{mechanism}$ contributes to calculating informational distance among system-level connection cuts. Finally, Φ of the system is defined by the largest informational distance among candidate mechanisms.

$$\Phi_{system} := max(D(p(system_{mechanism})||p(system_{MIP})))$$

To optimize iteration, this study limited the hidden layer to four neurons. Computation of Φ is achieved with the Python library `PyPhi` [12]. The theoretical basis of the calculation of integrated information is further detailed in [12] and [14]. This section will mostly focus on detailing how the activation data was specifically transformed in order to calculate Φ.

The activation of the neural network over time captures the state transitions for the robot vehicles during behavior. *Testing runs* were conducted using genes from representative individuals to capture the activity of the neural network. Each testing run was performed under the same conditions as during vehicle evolution. While testing runs, the neural network activity and the trajectory of

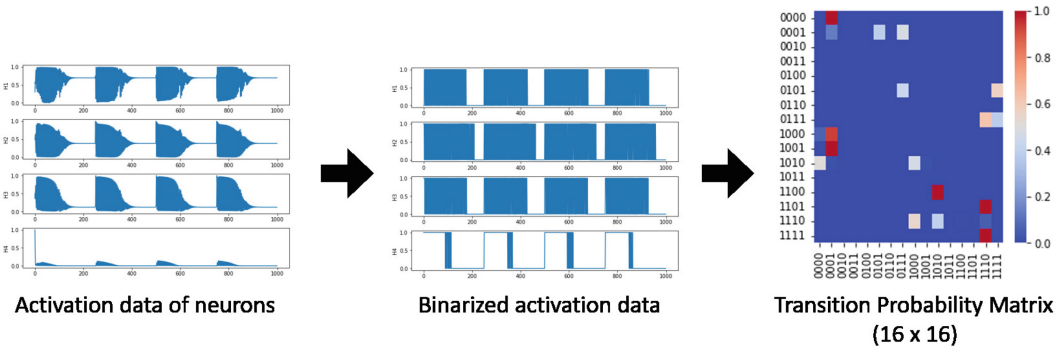

Fig. 3. Schematic process of obtaining TPM from activation data.

individuals are recorded over 1000 timepoints. Hence, the functional behavior of each individual robot and its corresponding Φ during those movements are measured.

The informational difference of state changing from one state to another can be captured as the probabilistic difference of such transition. Thus, the *transition probability matrix (TPM)* plays the key role in the computation of Φ [14]. From the calculation of Φ, the dropped-out nodes do not affect state transitions, thus calculating informational distance over different mechanisms and purviews is done by reducing the dimension of the TPM, by adding up the transition probabilities related to those nodes.

To obtain the TPM of an individual neural network, the system's activity must be expressed as either 0 or 1. Therefore, each neuron's activation data is normalized and then binarized using the median as the threshold to determine if it is considered active. The transition probability for each state was then extracted by counting how many times a transition from a particular state to a particular state has occurred over the 1000 timepoints, and dividing this count by 1000. Only the activity of the hidden layer from the neural network was used in the calculation of Φ, and since the hidden layer contains 4 nodes, the total number of possible states for the system is $4^2 = 16$. Specifically, at timepoint t_i, the state of the neural network could be one of 0000, 0001, 0010, 0011, ..., 1111, and at the next timepoint t_j, it will change to one of these states. The state transition probability from t_i to t_j is thus represented by a 16-by-16 matrix. Each entry (i, j) in this matrix gives the probability that the current state i will transition into the next state j. Following the TPM convention in `PyPhi`, this is referred to as a state-by-state TPM [12]. This process is illustrated in Fig. 3.

Importantly, the interpretation concerns cases where a state transition never occurs within the provided activity data, leading to the sum of the probability not equal to 1. For a state-by-state TPM, the sum of each row of any column must always equal 1. Since integrated information fundamentally represents causal information, it is impossible to calculate Φ in nondeterministic systems. Hence, the system must be assumed deterministic if Φ is to be calculated, which in turn, any row of a state-by-state TPM that does not sum to 1 would indicate that

the activity data lacks sufficient information to fully represent transitions for that case. That is, unobserved transitions in the data should be deemed due to transitions to states external to the dataset. Under such interpretation, if the sum of a specific row in the TPM is 0, it indicates insufficient observations of the current state; if the sum is between 0 and 1, it indicates insufficient observations of the next state. In this study, instances occurred where the sum of certain rows in the TPM is 0. To interpret these cases, a *context node* is introduced. This context node is assumed to activate under such unobserved transitions, and it is marginalized during the calculation process to obtain only the information derived from the activation of the actual hidden layer neurons.

The Φ values are non-negative, with a lower bound of zero and no defined upper bound [1,14]. Since Φ is not an absolute metric, it is unsuitable for comparing information integration across systems under different structures directly. It should be used to explore relative differences under different conditions within the same-structured system.

2.4 Analysis of Results

To examine changes in Φ across evolution, testing runs were conducted for the representative individuals of the 0th, 100th, 200th, 500th, and 1000th generations. To determine whether the differences in Φ values between generations are statistically significant, repeated measures ANOVA is used. Additionally, differences across vehicle types are assessed using the Kruskal-Wallis test.

To explore the relationship between Φ and the connectivity weights (genes), t-distributed stochastic neighbor embedding (t-SNE) analysis [10] was conducted on evolved weights from both early (100th) and late (1000th) generations. Vehicle type, set, and the magnitude of Φ were labeled for t-SNE clustering and visualized accordingly. For Φ values, agglomerative hierarchical clustering, which is less affected by outliers [8], was used to categorize them into three clusters: high, medium, and low based on their magnitude.

3 Results

3.1 Evolution of Vehicles

For all four vehicle types (2A, 2B, 3A, 3B), fitness approached an asymptote around the 500th generation. As shown in Fig. 4, the best-so-far fitness individual in the final generation exhibited functional behavior characteristics of the expected Braitenberg vehicle.

Furthermore, Φ from each vehicle type appeared to be statistically different according to the Kruskal-Wallis test. As in Table 1a, the probability that Φ calculated from each vehicle type would share the same distribution decreases significantly with increasing generation. Therefore, it can be safely concluded that the population of integrated information would differ depending on the vehicle type. Based on this result, each vehicle type was considered independent for the remaining analyses.

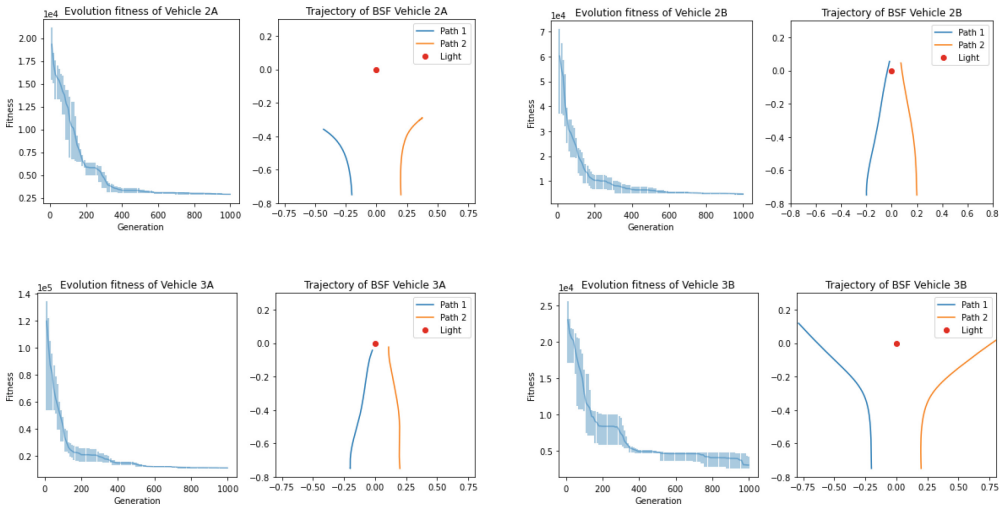

Fig. 4. Fitness change over generation and movement trajectory of evolved best-so-far (BSF) individuals of each vehicle type.

Table 1. Statistical test results.

(a) Test statistic and corresponding p-value of Kruskal-Wallis test of Φ over different vehicle types from each generation.

Generation	test statistic	p-value
0	5.3053	0.1508
100	25.4399	$< .0001$
200	57.9434	$< .0001$
500	59.1832	$< .0001$
1000	110.0884	$< .0001$

(b) F-statistic and corresponding p-value of repeated measures ANOVA of Φ over generation from each vehicle type.

Vehicle type	F-statistic	p-value
2A	18.7849	$< .0001$
2B	22.9872	$< .0001$
3A	7.8776	$< .0001$
3B	24.0576	$< .0001$

3.2 Changes in Φ over Generation

For all four vehicle types, Φ decreased as generations progressed. The trend and its statistics can be found in Fig. 5. Additionally, the difference in Φ over generations was found to be statistically significant for each vehicle type, as confirmed through repeated measures ANOVA conducted for each type in Table 1b.

The average value of Φ mostly falls below 1. While Φ cannot be directly compared in a strict sense across different studies due to its non-absolute scale, however, other studies with systems of similar node numbers also report Φ values mostly ranging from 0 to 2, suggesting there is no significant anomaly in the case of our calculation [2,4,5,11].

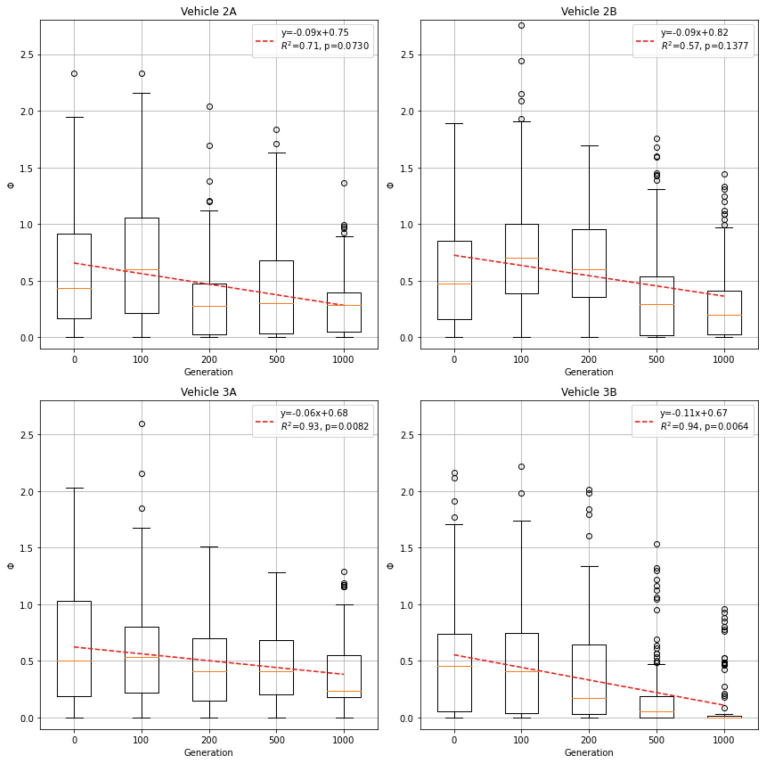

Fig. 5. Changes of Φ under increasing generation among four different types of vehicles. Regression line illustrated as dotted red line. (Color figure online)

One notable point is that, although the average value of Φ generally decreases as generations progress, there are still a considerable number of outliers exceeding the average throughout all generations. Consequently, in the subsequent t-SNE analysis, vehicle type, set, and the magnitude of Φ were labeled to facilitate tracking of individuals with high Φ values through visualization.

3.3 Changes in Connection Weights over Generation

The t-SNE analysis in Fig. 6 demonstrates how evolution has influenced connection weights. In this simulation setting, evolution appears to have led individuals to converge on specific genes (weights). Interestingly, this convergence of weights resulted in decreased Φ values. The bottom right plot in Fig. 6 shows that only one cluster maintains a high Φ after the evolutionary process, predominantly distributed among the set 4 group that evolved the functional behavior of vehicle 2B. In contrast, most late-generation genes measured lower Φ values, especially when compared to the highly dispersed early-generation genes, suggesting that earlier generations could be more prone to maintain higher Φ values due to more integration between hidden layer nodes. Interestingly, as behavior becomes more stereotyped and accurate as Φ decreases. This transition has been observed in animals as they progress from planned behavior to habits when learning a task [15].

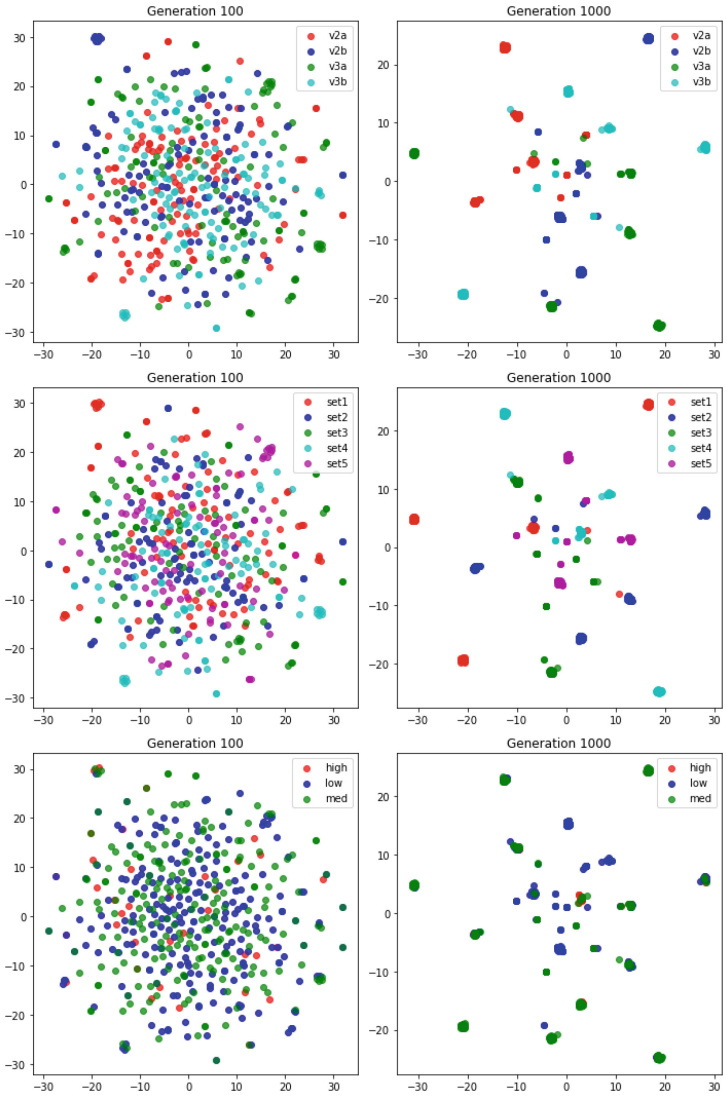

Fig. 6. t-SNE analysis. The first column shows the weights at the 100th generation, and the second column shows the 1000th generation. The first row is labeled with vehicle types, the second row is labeled with different runs (set1 - set5), and the third row is labeled with the Φ magnitude.

3.4 Tracking the Sample Individuals

In Fig. 7, individuals with distinct characteristics are selected to specifically examine their evolved behavior, connection weights, and Φ. It is notable that individuals of earlier generations, despite less accurate trajectories, can have higher Φ values. Note that the weights should not be similar since all individuals are selected from different t-SNE clusters. As seen in Fig. 6, there are multiple types of weights the individual can converge even for the same behavior. Figure 7 also shows that the TPM also can be different for the same behavior, though the TPM alone cannot directly tell the range of Φ value.

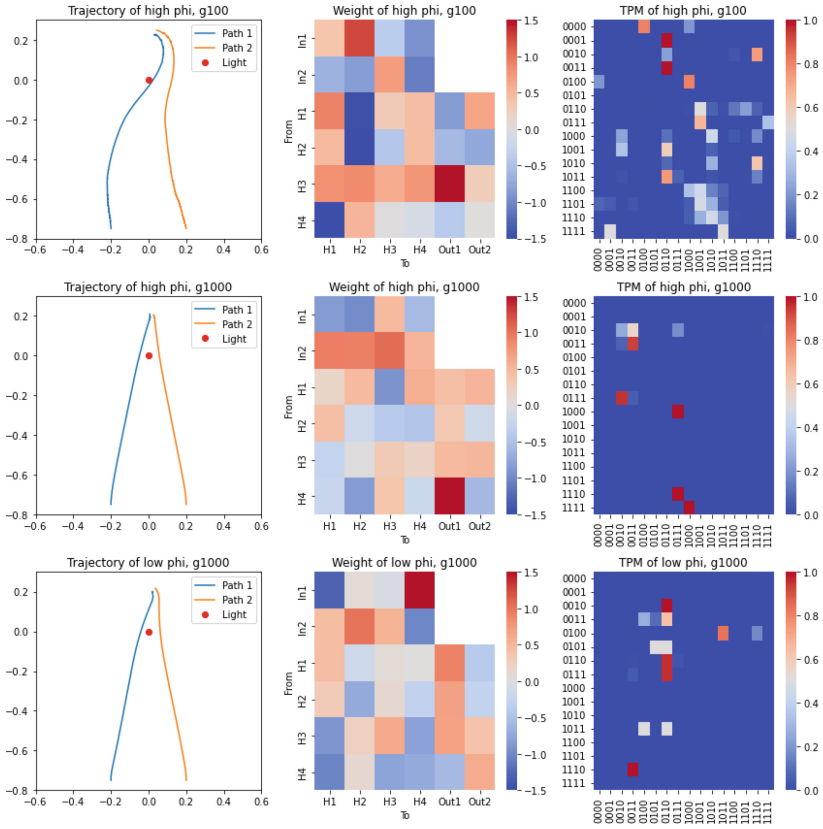

Fig. 7. Trajectory, neural network weight, and TPM of selected individuals. All individuals perform vehicle 2B behavior. **First row.** Early-generation high-Φ individual ($\Phi = 2.08529$). **Second row.** Late-generation high-Φ individual ($\Phi = 1.4463$). **Third row.** Late-generation low-Φ individual ($\Phi = 0$).

4 Conclusion

The present paper applied IIT, which is used to investigate conscious systems, to a classic thought experiment from synthetic psychology. Rather than applying supervised or reinforcement learning methods that would limit the neural architecture, we used an evolutionary approach that explored different weight configurations. As the evolutionary process proceeded, there was a reduction in weight diversity, and at the same time, a decrease in the Φ values. In the early stages of evolution where the robots were "exploring" to find the optimal path, the weights were diverse, but as weights became relatively consistent later on, the neurons' activation was also optimized, similar to behavior observed in rats [15]. Although the results were unexpected, they are consistent with the initial transition from random behavior to stereotyped, goal-driven behavior. However, to determine whether high Φ can be consistently observed in various instances of random behavior and low Φ in goal-driven behavior beyond the scope of this simulation, further studies in more complex systems including real-world experiments are necessary.

References

1. Albantakis, L., et al.: Integrated information theory (iit) 4.0: formulating the properties of phenomenal existence in physical terms. PLOS Comput. Biol. **19**(10), 1–45 (2023). https://doi.org/10.1371/journal.pcbi.1011465
2. Albantakis, L., Hintze, A., Koch, C., Adami, C., Tononi, G.: Evolution of integrated causal structures in animats exposed to environments of increasing complexity. PLOS Comput. Biol. **10**(12), 1–19 (2014). https://doi.org/10.1371/journal.pcbi.1003966
3. Braitenberg, V.: Vehicles: Experiments in Synthetic Psychology. MIT Press, Cambridge (1986)
4. Gomez, J.D., Mayner, W.G.P., Beheler-Amass, M., Tononi, G., Albantakis, L.: Computing integrated information (ϕ) in discrete dynamical systems with multivalued elements. Entropy **23**(1), 6 (2021). https://doi.org/10.3390/e23010006
5. Hoel, E., Albantakis, L., Marshall, W., Tononi, G.: Can the macro beat the micro? integrated information across spatiotemporal scales. Neurosci. Consciousness (2016). https://doi.org/10.1093/nc/niw012
6. Hwu, T., Krichmar, J.: Neurorobotics: Connecting the Brain, Body and Environment. MIT Press, Cambridge (2022)
7. Hwu, T., Krichmar, J.: Neurorobotics: Connecting the brain, body and environment. (2022). https://github.com/jkrichma/NeurorobotExamples/tree/main
8. Jain, A., Dubes, R.: Algorithms for Clustering Data. Prentice Hall advanced reference series, Prentice Hall, Upper Saddle (1988). https://books.google.co.kr/books?id=7eBQAAAAMAAJ
9. Jong, K.A.D.: Evolutionary Computation: A Unified Approach. MIT Press, Cambridge (2016)
10. van der Maaten, L., Hinton, G.: Visualizing data using t-sne. J. Mach. Learn. Res. **9**(86), 2579–2605 (2008). http://jmlr.org/papers/v9/vandermaaten08a.html
11. Marshall, W., Albantakis, L., Tononi, G.: Black-boxing and cause-effect power. PLOS Comput. Biol. **14**(4), 1–21 (2018). https://doi.org/10.1371/journal.pcbi.1006114
12. Mayner, W.G.P., Marshall, W., Albantakis, L., Findlay, G., Marchman, R., Tononi, G.: Pyphi: a toolbox for integrated information theory. PLOS Comput. Biol. **14**(7), 1–21 (2018). https://doi.org/10.1371/journal.pcbi.1006343
13. Moon, K., Pae, H.: Making sense of consciousness as integrated information: evolution and issues of integrated information theory. J. Cogn. Sci. **20**(1), 1–52 (2019). https://doi.org/10.17791/jcs.2019.20.1.1
14. Oizumi, M., Albantakis, L., Tononi, G.: From the phenomenology to the mechanisms of consciousness: integrated information theory 3.0. PLOS Comput. Biol. **10**(5), e1003588 (2014). https://doi.org/10.1371/journal.pcbi.1003588
15. Redish, A.D.: Vicarious trial and error. Nat. Rev. Neurosci. **17**(3), 147–159 (2016). https://doi.org/10.1038/nrn.2015.30
16. Webots: http://www.cyberbotics.com. Open-source Mobile Robot Simulation Software