

The Role of Energy Constraints on the Evolution of Predictive Behavior

William Kang, Christopher Anand, and Yoonsuck Choe

Department of Computer Science and Engineering
Texas A&M University
College Station, TX 77843-3112, USA
{rkdvlfah1018,chrisanand,choe}@tamu.edu

Abstract. Prediction is an important foundation of cognitive and intelligent behavior. Recent advances in deep learning heavily depend on prediction, in the form of self-supervised learning based on prediction and reinforcement learning (reward prediction). However, how such predictive capabilities emerged from simple organisms has not been investigated fully. Prior works have shown the relationship between input delay and predictive function to compensate for such delay. In this paper, we investigate other key factors that may contribute to the emergence of predictive behavior in evolving neural networks. We set up a delayed reaching task with a two-segment articulated arm. The arm is controlled to reach a moving target, where the target's coordinate information is received with a delay. Following our previous work, we introduced a tool to extend the reach, when the target is beyond the arm's reach. In this task, without predicting the trajectory of the moving target, the controller cannot reach the target. For the controller, we used the NeuroEvolution of Augmenting Topologies (NEAT) algorithm. Our results indicate that an important fitness criterion for the emergence of predictive behavior is that of reduced energy usage (in the form of economy of motion). Further analysis shows that the number of recurrent loops correlates with target reaching performance, but more strongly so with the energy constraint. We expect our findings to lead to further investigations on the role of energy constraints on the evolution of predictive behavior.

Keywords: Prediction · Evolution · Neuroevolution · Fitness ¹

1 Introduction

Prediction forms an important foundation of cognitive and intelligent behavior [1, 18, 17]. Recent advances in deep learning heavily depend on prediction, in the form of self-supervised learning based on prediction and reinforcement learning (e.g., reward prediction) [13, 14].

¹ Kang, W., Anand, C., and Choe, Y. (2024, September). The Role of Energy Constraints on the Evolution of Predictive Behavior. In International Conference on Simulation of Adaptive Behavior (pp. 211-222).

However, how such predictive capabilities emerged from simple organisms through evolution has not been investigated sufficiently [7, 12]. Furthermore, what kind of external/internal factors and constraints could have influenced the development of prediction is unclear. Prior works have suggested the relationship between input delay and predictive function for its compensation [10, 11].

In this paper, we investigate other key factors that may contribute to the emergence of predictive behavior in evolving neural networks. We set up a reaching task with a two-segment articulated arm, with added input delay [9]. (Note that predicting an unknown true location like this is different from predicting the sensory consequence of action, which requires the true sensory input to compute the prediction error, as in [17].) The arm is controlled to reach a moving target, where the target’s coordinate information is received with a fixed delay. Following our previous work [9], we introduced a tool to extend the reach, when the target is beyond the arm’s reach. This task is not solvable without predicting the trajectory of the moving object, since reaching for the coordinate location based on the immediate input would lead to a location previously occupied by the moving object. For the controller, we used the NeuroEvolution of Augmenting Topologies (NEAT) algorithm [15] (cf. CTRNN [17] and FORCE [16, 6]).

Our results indicate that an important (auxiliary) fitness criterion for the emergence of predictive behavior is that of reduced energy usage (in the form of economy of motion). Further analysis shows that the number of recurrent loops correlate with target reaching performance, but more strongly so with the energy constraint. We expect our findings to lead to further investigations on the role of energy constraints on the evolution of predictive behavior.

2 Background

In this section, we will briefly review existing works relating to the evolution of prediction, and provide an overview of the NEAT algorithm, which we will use.

2.1 Evolution of Prediction

In our previous works, we show that predictive dynamics emerge in evolved neural network controllers when environmental conditions change [8, 21]. Aside from these, the few papers that discuss prediction in an evolutionary context include [7] and [12]. [7] mentioned the lack of computational studies on evolution of prediction, and proceed to propose how utility and predictive capabilities can co-evolve in a simulated agent. The work is based on the “Gap Theory” in evolutionary economics which exactly states the co-evolutionary nature of utility and prediction. [12], on the other hand, observed that curiosity (a form of intrinsic motivation) helps in the improvement of prediction, and these developmental structures can constrain evolution in return. Thus, the work is more about the emergent predictive ability shaping evolution, not about how evolution gives rise to predictive abilities.

2.2 NeuroEvolution of Augmenting Topologies (NEAT)

Topological neuroevolution methods evolve both topology and weights of neural networks. Our perspective is that natural evolution includes changes in the network topology in the brain, thus they mimic the natural evolution better than traditional weight-only neuroevolution methods. Moreover because the functionality of a neural network can be constrained by its topology, allowing the topology to evolve will set free the structural constraints and result in new capabilities such as recurrent dynamics. Amongst many variations of such an approach [19, 2, 20], we will use NeuroEvolution of Augmenting Topologies (NEAT) because of its advantages over other topological evolution methods [15]. Fig. 1 illustrates the main concepts of NEAT.

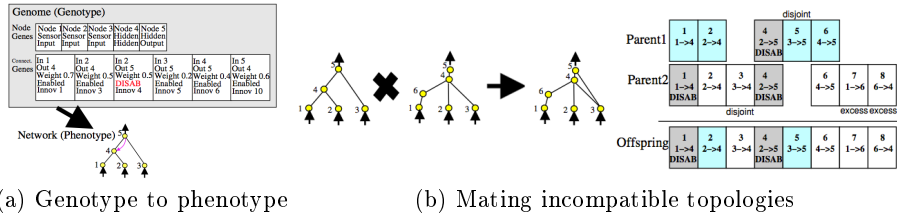


Fig. 1. NEAT. (a) Nodes and connections are separately encoded. (b) Connections with the same innovation number can be crossed over. Adapted from [15].

Historical marking is a major feature in the NEAT algorithm. By enumerating each innovation, NEAT solves the competing conventions problem, which is one of the main problems in neuroevolution [19, 15]. The crossover operation in NEAT happens between two genomes with identical historical marking (also called “innovation number”), regardless of their locations and size in the network (fig. 1). NEAT encodes the genome in two arrays, node genes and connection genes. Innovation number is assigned to each connection gene according to the order of its appearance throughout the evolutionary stages. The connections can also be enabled or disabled through mutation. Since connections can be generated arbitrarily between neurons, recurrent connections can also be generated. There are several other important facilities such as speciation, where a subpopulation of individuals are isolated from other subpopulations, forming a species.

3 Methods

In this section, we will discuss the details of the task and how NEAT is hooked up to the environment. We will also discuss the various fitness factors we used, and the performance metrics we employed for the evaluation.

3.1 Delayed Reaching Task

The delayed reaching task is illustrated in fig. 2. A two-segment arm with two joints can be controlled by its two joint angles (θ_1, θ_2) to reach a moving target (black square). To test the predictive capability, the task is modified so that the coordinate of the target object is fed to the controller with a delay (red square). If the controller tries to reach the delayed target, based on the delayed input, it will not be able to reach the true target. An additional obstacle is that the movement of the target can take it beyond the reach of the arm. The controller can decide to pick up a stick (green) to extend its reach. The stick automatically snaps onto the hand (circle) and extends the arm's reach once the stick handle is touched by the hand.

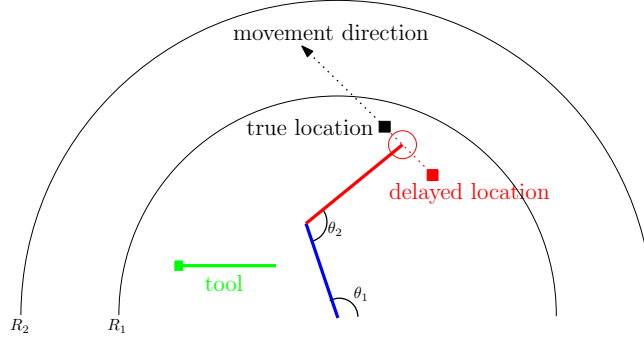


Fig. 2. Delayed Reaching Task. The task is to control the limb angles θ_1 and θ_2 to reach the moving target (black square). The location of the target is received by the controller with a delay (red square). If the target is beyond the arm's reach (R_1), the controller needs to pick up the stick (green) to extend the reach.

The target was initially placed randomly within one of four regions (upper left, lower left, lower right, upper right), and moved in one of four directions (0° , 45° , 135° , and 180° , respectively). The target was moved for 500 simulation steps in the beginning (which was the amount of delay), before the controller can start movement.

3.2 NEAT Controller

For control of the arm, we used NEAT. Fig. 3 shows the initial topology of the controller network. The 9 inputs and 2 outputs of the network were as follows.

Input: (1) θ_1 : joint angle 1, (2) θ_2 : joint angle 2, (3) *target_dist*: distance between hand and delayed target, (4) *target_angle*: angle from hand to delayed target, (5) *tool_dist*: distance between hand and tool handle (not delayed, since the tool is static), (6) *tool_angle*: angle from hand to tool handle, (7) θ_{1limit} : triggered when joint 1 limit is reached, (8) θ_{2limit} : triggered when joint 2 limit is

reached, (9) *target_touch*: tactile sensor triggered when hand touches the true target.

Output: (1) θ_1 : joint angle 1 adjustment, (2) θ_2 : joint angle 2 adjustment. The adjustments were limited to $-1.5^\circ \leq \theta_i \leq 1.5^\circ$.

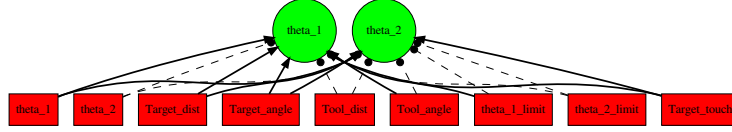


Fig. 3. Initial NEAT Topology. The initial NEAT controller topology is shown. Red: input, Green: output. Note that there are no hidden neurons, since this is the initial topology. See text for details.

3.3 Fitness

We tested a combination of different fitness factors, each scaled between 0.0 and 1.0, and the factors multiplied, as shown below (see the Appendix for details):

$$F = \alpha \times \prod_i F_i, \quad (1)$$

where α is a product of common factors used in all experimental conditions, and $\prod_i F_i$ a product of the fitness factors that are to be compared. The common factor α is defined as $\alpha = D \times N \times C$, where D is based on the mean square distance between the true target and the hand location (1 - this quantity), N is the average number of abrupt change of direction to discourage oscillating behavior (1 - this quantity), and C is the total hit count in each behavioral attempt. The main factors we compared were $F_i \in \{E, R, T\}$, where E is the energy factor (1 - normalized total hand travel distance), R the predictive reach factor (number of 5 or more consecutive hits), and T the tool factor (tool pick up). In our case, we tested these factor combinations: R, RT, ER , and ERT .

3.4 Performance Metrics

Other than the fitness, each individual was tested in multiple random behavioral attempts ($N_{max} = 100$) to measure the performance.

The first metric is the total number of “hits” (touching the target) in each behavioral attempt. (Note that all attempts were 5,000 steps. Also, the maximum value of total hits in each attempt depends on the relative position of the randomly initialized hand, tool, and target positions.) Although this is a good general metric, it does not measure the predictive capability.

$$total_hits = \sum_{n=1}^{N_{max}} \text{total number of all hits in attempt } n \quad (2)$$

Metric 1_hit measures the number of attempts in which the agent had reached the target at least once. The 1_hit is computed with equation 3. While this metric is standardized and can show the effectiveness of a network, it cannot distinguish between an accidental hit vs. predictive reach.

$$1_hit = \sum_{n=1}^{N_{max}} h_n, \text{ where } h_n = \begin{cases} 1 & \text{if target hit at least once in attempt } n \\ 0 & \text{else} \end{cases} \quad (3)$$

Hence, we also introduce the 5_hit metric (eq. 4) which counts the number of attempts in which the agent had reached the target at least 5 consecutive time steps (a different value may also be used, e.g., 10). This is to ensure that the reach of a target was intentional and therefore is used as our main metric to measure the predictive capabilities of the agents.

$$5_hit = \sum_{n=1}^{N_{max}} h_n, \text{ where } h_n = \begin{cases} 1 & \text{if target hit 5+ consec. steps in attempt } n \\ 0 & \text{else} \end{cases} \quad (4)$$

All the metrics mentioned above apart from the fitness scores, act as a standard metric to compare the successes of different agents. They share common extremes ($[0, 500,000]$ for the number of reaches and $[0, 100]$ for 1_hit or 5_hit) and are computed using the same equations resulting in standardized scores.

4 Experiments and Results

Each of the four fitness types (R, RT, ER, ERT) was evolved for 150 generations.

4.1 Population Average over Generations

Fig. 4 shows the population average of the fitness and the number of reaches over the generations. We find that the ER factor combination significantly outperforms the others (note that in this case, the evolutionary “trial” ended early, since it exceeded the preset fitness threshold).

4.2 Performance of Best Individuals

Next, we tested the best individuals for each fitness type, using the 1_hit and 5_hit metric (note that E is not a good metric due to trivial cases: e.g., when the agent did not move). For each individual tested, we ran the individual in a task 100 times (“attempts”), and counted the number of times it was able to hit the

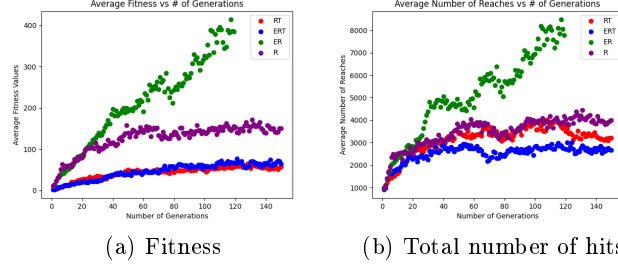


Fig. 4. Population Average of Fitness and Total Number of Hits.

target once or five consecutive times, respectively. We repeated n ($= 31$) such “runs” to measure the performance (fig. 5). Note that 1_hit and 5_hit can be at the most 100 ($N_{max} = 100$). For 1_hit , the results are mixed (fig 5(a)). ERT shows the best performance, followed by R , ER , then RT . Thus, there is no clear difference between (R, RT) vs. (ER, ERT) . However, fig. 5(b) shows that for 5_hit , (ER, ERT) clearly outperforms (R, RT) . Furthermore, fig. 5(c) shows that most of the 1_hit events are also 5_hit events for (ER, ERT) , showing that most reaching behavior is prediction based. However, this is not the case for the (R, RT) condition, where most reaching behavior is blind waving.

Considering that 5_hit is an indicator of predictive behavior (continuously and intentionally tracking the true target location, rather than randomly waving to hit the target by luck), we can conclude that the fitness types that include the Energy factor facilitates the emergence of predictive behavior.

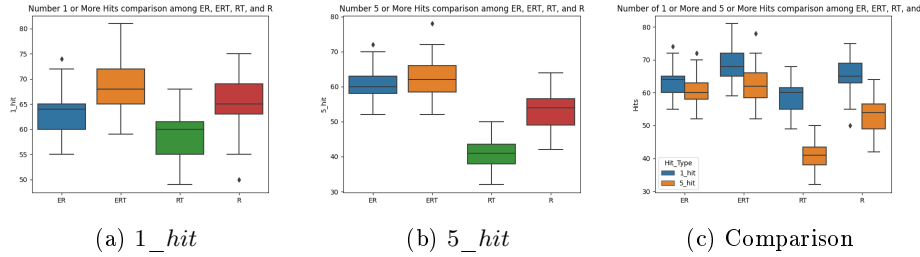


Fig. 5. Performance of Best Individuals. The (a) 1_hit and (b) 5_hit results are shown for the four fitness types. In (b): Mann-Whitney test: $ERT > RT$ [$n = 31, p = 1.36e - 11$], $ER > R$ [$n = 31, p = 9.71e - 07$]. (c) Compares (a) and (b) in a single plot.

4.3 Behavior

Observing the behavior can provide some insights on the relevance of our 1_hit and 5_hit metrics. Fig. 6 show some typical behaviors by fitness type. We plotted

the behavior in time lapse (vivid color = most recent frame). We can also see the target’s moving direction (black = true target, red = delayed target): For example, in fig. 6(b), the target is moving from lower left to upper right. For each fitness type, representative behavior that exhibit predictive property (a through d) and those that do not (e through h) are shown. For the top row (predictive), we see that the hand dwells close to the true target location (black square), ahead of the delayed target (red square). This kind of behavior may not be possible without some form of prediction, and may score high on the 5_hit criterion. For the bottom row (non-predictive), the hand makes broad sweeping gestures. This could lead to a high 1_hit score, but a low 5_hit score. With this, we can view the results in fig 5(c) in a new light: The fitness types that involve the Energy factor may be exhibiting predictive behavior, while those that do not are merely successful in reaching the target through undirected broad sweeping behavior.

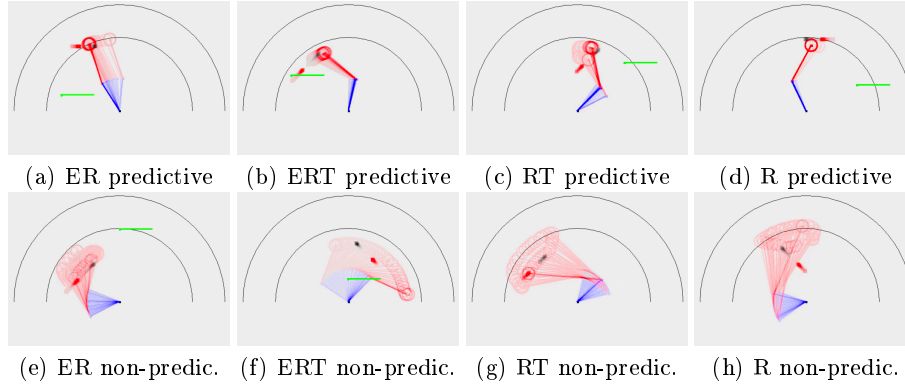


Fig. 6. Representative Behaviors. Time lapse of representative behaviors are shown (vivid color = most recent frame). Black square: true target. Red square: delayed input. Top row (a~d): predictive behavior. Bottom row (e~h): non-predictive behavior. Note that in (g) and (h), the tool is picked up, so the reach is extended. See text for details.

4.4 Network Topology

The evolved network topology also gave us further insights on how different fitness types shaped the controller’s behavior (fig. 7). At first glance, there are no distinguishable differences in appearance, other than some having more hidden neurons than others. Further analysis reveals an interesting property. Our previous work on analyzing evolved network topology showed a positive correlation between the number of loops and the performance [9]. We conducted the same kind of analysis, by counting the number of simple cycles in the evolved network (we used NetworkX for this [3]). The results are shown in fig. 8. Interestingly, for the fitness types that include the Energy factor (ER , ERT), the number of

loops are strongly correlated with the success rate ($r = 0.59$ and 0.51 , respectively). However, for those without the Energy factor (RT, R), the correlation is weak ($r = 0.17$ and 0.38 , respectively). These results suggest that the recurrent loops in the networks evolved with the energy constraints may be supporting predictive function better than those without. How these loops contribute to prediction need to be investigated (for initial attempts, see lesion studies in [5]).

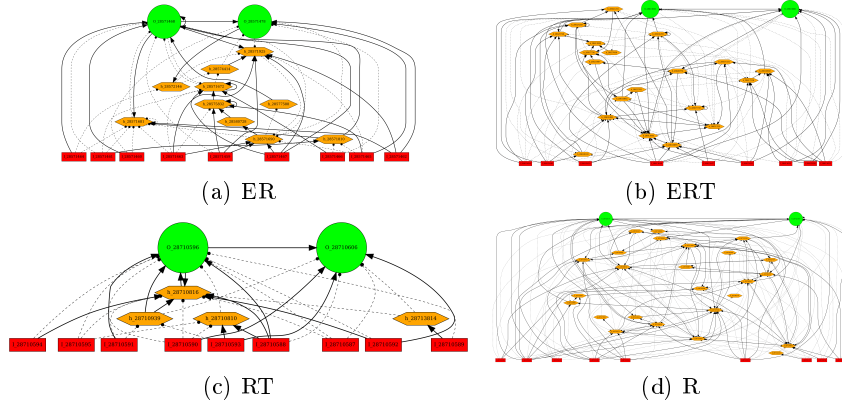


Fig. 7. Typical Evolved Topologies. Typical evolved topologies are shown for the different fitness types. Neurons: Red = input, Orange = hidden, Green = output. Connections: Solid lines with arrows = excitatory, Dashed lines with discs = inhibitory.

5 Discussion

The main contribution of this paper is the identification of constraint on energy as a major factor in the emergence of predictive behavior in the context of evolution. With naive performance metrics such as the number of times the target has been reached, evolution found equally effective strategies to reach the target such as broad sweeping, but the behavior was not indicative of prediction. It is also interesting to note that even with the addition of a fitness factor that explicitly rewards predictive behavior (the R factor, which basically uses 5_hit), that alone was not able to give rise to predictive behavior, i.e., E was needed.

What could be some other factors that contributed to prediction? As mentioned in the introduction, delay in the input could be one such factor [10, 11]. Due to delay in the sensory input, the need to compensate for this arises, which may be resolved by prediction: prediction of the present from the past. We have also found another curious factor that may be involved, which is environmental change, as briefly mentioned in the background. In a simple 2D pole balancing task, we found that individuals with predictive internal dynamics can be more robust in a changing environment (in this case, the change in the initial pole angle from $\theta < 5^\circ$ to $\theta < 10^\circ$) [8, 21]. With our research reported here, we can now

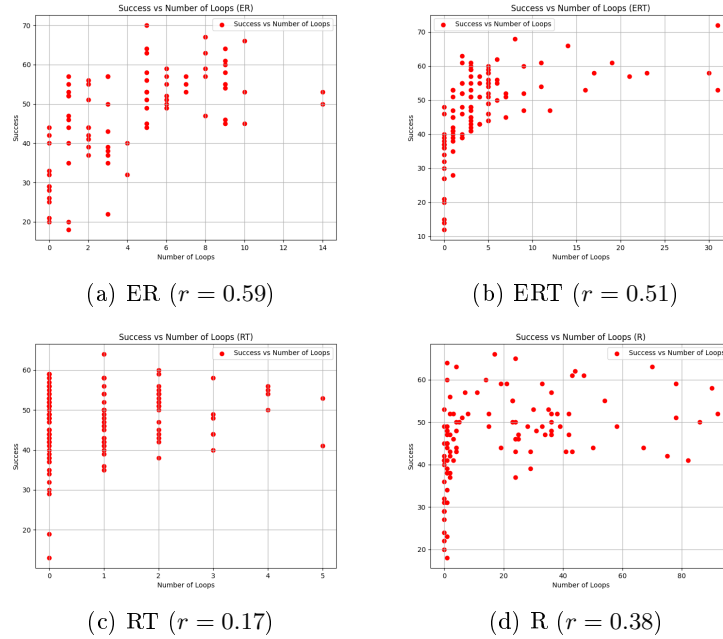


Fig. 8. Number of Loops vs. Success Rate. The number of loops (simple cycles in graph theory) in an individual vs. its success rate is plotted. The r values are the correlation coefficients. Each point corresponds to one individual in the population. We can see that the correlation is higher for the fitness types that includes the Energy factor (ER and ERT), compared to those that do not include this factor (RT and R).

summarize the seemingly unrelated factors that contribute to the emergence of prediction as (1) energy constraint, (2) delay, (3) environmental change.

6 Conclusion

In this paper, we investigated factors contributing to the emergence of predictive behavior in an evolutionary context. We used a delayed reaching task, where reaching the true target requires some form of prediction based on the delayed input from the past. We found that the energy factor plays an important role, allowing the evolved controllers to be more successful in the reaching task, and to exhibit more focused predictive reaching behavior. The controllers evolved without the energy factor were moderately successful, but the strategy were mostly based on undirected, systematic sweeping, not indicative of any prediction of the target’s trajectory. Furthermore, we found that structural innovations like recurrent loops in the evolved controllers play a more cohesive role in support of the predictive function, when the energy factor was included. These results suggest a key role of energy constraints in the evolution of predictive behavior.

Acknowledgments: Based on the undergraduate honor thesis [5]. Environment simulation and NEAT based on Qinbo Li’s code [9], using ANJI [4]. We would like to thank the anonymous reviewers for helping us clarify our points.

Appendix

The fitness factors were computed as follows. All factors were computed from N_{max} behavioral attempts (= 100) using the same individual chromosome, with each attempt running for a maximum of M_{max} simulation steps. W is the width of the arena (= 512 pixels), and L is the initial lead up steps which corresponds to the delay (= 500 steps). The factors were D : distance, N : turn, C : total hit count, E : energy, R : predictive reaches, T : tool.

$$\begin{aligned} D &= 1 - \sum_{n=1}^{N_{max}} \sum_{m=1}^{M_{max}} \frac{d_{nm}^2}{(WN_{max}M_{max})^2} & N &= 1 - \sum_{n=1}^{N_{max}} \frac{n_n}{N_{max}M_{max}} \\ C &= \sum_{n=1}^{N_{max}} \frac{c_n}{N_{max}M_{max}} & E &= 1 - \sum_{n=1}^{N_{max}} \frac{e_n}{N_{max}(M_{max}-L)} \\ R &= \sum_{n=1}^{N_{max}} \frac{r_n}{N_{max}} & T &= \sum_{n=1}^{N_{max}} \frac{t_n}{N_{max}} \end{aligned}$$

where, in each attempt, d_{nm} =distance between hand and true target in attempt n at time step m , n_n = number of sharp turns (= reversal of hand movement direction $> 90^\circ$), c_n = total number of hits, e_n = consumed energy (= cumulated hand travel distance), $r_n = 5_hit$, $t_n = 1$ if tool is held and 0 otherwise.

Table 1. NEAT (ANJI) Hyper-parameters

Evolution	
num.generations = 150	popul.size = 100
topology.mutation.classic = false	add.connection.mutation.rate = 0.02
remove.connection.mutation.rate = 0.01	remove.connection.max.weight = 100
add.neuron.mutation.rate = 0.01	prune.mutation.rate = 1.0
weight.mutation.rate = 0.75	weight.mutation.std.dev = 1.5
weight.max = 10.0	weight.min = -10.0
survival.rate = 0.2	selector.elitism = true
selector.roulette = false	selector.elitism.min.specie.size = 1
Speciation	
chrom.compat.excess.coeff = 1.0	chrom.compat.disjoint.coeff = 1.0
chrom.compat.common.coeff = 0.4	speciation.threshold = 0.25
Fitness Function	
stimulus.size = 9	response.size = 2
fitness.func.adjust.for.netw.size.factor = 0	fitness.threshold = 1000
fitness.target = 1200	
Activation Function	
initial.topology.activation = sigmoid	
Network Architecture	
initial.topology.fully.connected = true	init.topology.num.hidden.neurons = 0
initial.topology.activation.input = linear	recurrent = best_guess
recurrent.cycles = 1	ann.type = anji

References

1. Bubic, A., Von Cramon, D.Y., Schubotz, R.I.: Prediction, cognition and the brain. *Frontiers in human neuroscience* **4**, 1094 (2010)
2. Gruau, F., Whitley, D., Pyeatt, L.: A comparison between cellular encoding and direct encoding for genetic neural networks. In: Koza, J.R., Goldberg, D.E., Fogel, D.B., Riolo, R.L. (eds.) *Genetic Programming 1996: Proceedings of the First Annual Conference*. pp. 81–89. MIT Press, Cambridge, MA (1996)
3. Hagberg, A., Conway, D.: *Networkx: Network analysis with python*. URL: <https://networkx.github.io> (2020)
4. James, D., Tucker, P.: ANJI: Another NEAT Java implementation (2004), <https://anji.sourceforge.net/index.html>
5. Kang, W., Anand, C.: Emergence of Prediction in Delayed Reaching Task Through Neuroevolution. Engineering honors in computer science and engineering thesis, Texas A&M University (2023)
6. Kashyap, H.J., Detorakis, G., Dutt, N., Krichmar, J.L., Neftci, E.: A recurrent neural network based model of predictive smooth pursuit eye movement in primates. In: *2018 International Joint Conference on Neural Networks (IJCNN)*. pp. 1–8. IEEE (2018)
7. Korb, K.B., Brumley, L., Kopp, C.: An empirical study of the co-evolution of utility and predictive ability. In: *2016 IEEE Congress on Evolutionary Computation (CEC)*. pp. 703–710. IEEE (2016)
8. Kwon, J., Choe, Y.: Internal state predictability as an evolutionary precursor of self-awareness and agency. In: *Proceedings of the Seventh International Conference on Development and Learning*. pp. 109–114. IEEE (2008), <http://faculty.cs.tamu.edu/choe/ftp/publications/kwon.icdl08.pdf>
9. Li, Q., Yoo, J., Choe, Y.: Emergence of tool use in an articulated limb controlled by evolved neural circuits. In: *Proceedings of the International Joint Conference on Neural Networks* (2015), <http://faculty.cs.tamu.edu/choe/ftp/publications/li-ijcnn15.pdf>, DOI: 10.1109/IJCNN.2015.7280564
10. Lim, H., Choe, Y.: Compensating for neural transmission delay using extrapolatory neural activation in evolutionary neural networks. *Neural Information Processing—Letters and Reviews* **10**, 147–161 (2006), <http://faculty.cs.tamu.edu/choe/ftp/publications/lim.niplr06-reprint.pdf>
11. Nijhawan, R.: Motion extrapolation in catching. *Nature* **370**, 256–257 (1994)
12. Oudeyer, P.Y., Smith, L.B.: How evolution may work through curiosity-driven developmental process. *Topics in Cognitive Science* **8**(2), 492–502 (2016)
13. Shwartz Ziv, R., LeCun, Y.: To compress or not to compress—self-supervised learning and information theory: A review. *Entropy* **26**(3), 252 (2024)
14. Stachenfeld, K.L., Botvinick, M.M., Gershman, S.J.: The hippocampus as a predictive map. *Nature neuroscience* **20**(11), 1643–1653 (2017)
15. Stanley, K.O., Miikkulainen, R.: Evolving neural networks through augmenting topologies. *Evolutionary Computation* **10**, 99–127 (2002)
16. Sussillo, D., Abbott, L.F.: Generating coherent patterns of activity from chaotic neural networks. *Neuron* **63**(4), 544–557 (2009)
17. Tani, J.: *Exploring robotic minds: actions, symbols, and consciousness as self-organizing dynamic phenomena*. Oxford University Press (2016)
18. Tjøstheim, T.A., Stephens, A.: Intelligence as accurate prediction. *Review of Philosophy and Psychology* **13**(2), 475–499 (2022)

19. Whitley, D., Dominic, S., Das, R., Anderson, C.W.: Genetic reinforcement learning for neurocontrol problems. *Machine Learning* **13**, 259–284 (1993)
20. Yao, X.: Evolving artificial neural networks. *Proceedings of the IEEE* **87**(9), 1423–1447 (1999)
21. Yoo, J., Kwon, J., Choe, Y.: Predictable internal brain dynamics in EEG and its relation to conscious states. *Frontiers in Neurorobotics* **8**(00018) (2014), <http://journal.frontiersin.org/article/10.3389/fnbot.2014.00018/full>