Title	Evolving neural networks through bio-inspired parent selection in dynamic environments
Author(s)	Sunagawa, Junya; Yamaguchi, Ryo; Nakaoka, Shinji
Citation	Biosystems, 218, 104686 https://doi.org/10.1016/j.biosystems.2022.104686
Issue Date	2022-08
Doc URL	http://hdl.handle.net/2115/90278
Rights	© 2022. This manuscript version is made available under the CC-BY-NC-ND 4.0 license https://creativecommons.org/licenses/by-nc-nd/4.0/
Rights(URL)	http://creativecommons.org/licenses/by-nc-nd/4.0/
Type	article (author version)
File Information	Biosystems_218_104686.pdf

Evolving Neural Networks through Bio-inspired Parent Selection in Dynamic Environments

Junya Sunagawa[1,*], **Ryo Yamaguchi**[2], **Shinji Nakaoka**[2]

[1]Department of Graduate School of Life Science, Hokkaido University, Hokkaido, Japan
[2]Department of Advanced Transdisciplinary Science, Hokkaido University, Hokkaido, Japan

*Corresponding author
E-mail address:
sunachi110106-7@eis.hokudai.ac.jp (J. Sunagawa)
ryamaguchi@sci.hokudai.ac.jp (R. Yamaguchi)
snakaoka@sci.hokudai.ac.jp (S. Nakaoka)

Abstract

Environmental variability often degrades the performance of algorithms designed to capture the global convergence of a given search space. Several approaches have been developed to challenge environmental uncertainty by incorporating biologically inspired notions, focusing on crossover, mutation, and selection. This study proposes a bio-inspired approach called NEAT-HD, which focuses on parent selection based on genetic similarity. The originality of the proposed approach rests on its use of a sigmoid function to accelerate species formation and contribute to population diversity. Experiments on two classic control tasks were performed to demonstrate the performance of the proposed method. The results show that NEAT-HD can dynamically adapt to its environment by forming hybrid individuals originating from genetically distinct parents. Additionally, an increase in diversity within the population was observed due to the formation of hybrids and novel individuals, which has never been observed before. Comparing two tasks, the characteristics of NEAT-HD were improved by appropriately setting the algorithm to include the distribution of genetic distance within the population. Our key finding is the inherent potential of newly formed individuals for robustness against dynamic environments.

Keywords

Dynamic environment, bio-inspired, evolutionary algorithm, genetic algorithms, crossover, neural network

1. Introduction

Neuroevolution (NE) is an area of machine learning that develops artificial neural networks (ANNs) using an evolutionary algorithm (EA) or a similar bio-inspired stochastic search/optimization algorithm (Floreano et al., 2008; Papavasileiou et al., 2021). ANNs originate from biological neural networks in the brain. They are generally composed of interconnected neurons and weights to determine their behavior. EAs evolve ANNs to find suitable network parameters and topology structures inspired by the principles of Darwinian evolution. The concept of EA is to maintain a population of solution candidates through the search space and process a stochastic search that has the advantage of sampling many points simultaneously by selection, crossover, and mutation. Thus, the population can avoid falling to the local optima, leading to global optima. The potential of NE has been widely recognized and applied in fields ranging from quantum physics (Piacentino, 2009) to video games (Stanley et al., 2005) and chemical reaction networks (Dinh et al., 2015); which are considered competitive with deep learning (Stanley et al., 2019).

One approach to gaining considerable attention in EA is the neuroevolution of augmenting topologies (NEAT) algorithm (Stanley and Miikkulainen, 2002). NEAT adopts a framework in which an ANN adapts to an environment by evolution, owing to unique properties represented by “speciation”. Speciation is an idea from biology that each ANN is regarded as an individual in terms of genetic relationships. Genetically close individuals (i.e., the same species) can cross over, while genetically distinct individuals (i.e., different species) cannot. Speciation prevents the population from losing innovative ANNs having near-global convergence in subsequent generations.

In methods such as Genetic Algorithm (GA), speciation often matches individuals to improve the performance of the algorithm. Various selection rules have already been proposed for parent selection, such as roulette wheel selection, tournament selection, and rank selection. These well-known methods find parents with better fitness (Thierens and Goldberg, 1994). On the other hand, in the field of natural ecology and evolution, various studies have argued that the hybridization process between distinct individuals can significantly affect the evolution of the population (Mallet, 2007). Empirical and theoretical studies in biology have shown that hybridization between ecologically divergent taxa results in a rise in genetic variation, allowing the hybrids to adapt to a novel environment, thus forming a new species (Grant and Grant, 2011; Grant and Grant, 2019; Yamaguchi and Otto, 2020). The introduction of genetic relationships as a novel parent selection process would improve the performance of GA.

In this work, we propose a new method called NEAT with biological Hybridization for a Dynamic environment (NEAT-HD), which extends NEAT by replacing the parent selection process of NEAT with a bio-inspired operation based on the genetic distance. NEAT-HD allows the population to crossover within a given species group and between genetically distinct species. This property results in the emergence of a hybrid individual, defined as offspring produced from parents of two different species. Despite the low probability of emergence, we demonstrate that a population including both interspecies and intraspecies crossover effectively searches a search space in dynamic environments where the solution changes over time. More specifically, we show that NEAT-HD can achieve the suppression of considerable fitness reduction and diversity preservation, referred to as fitness endurance contributed by genetic diversity when the environment changes dynamically.

The remainder of this paper is organized as follows. Section 2 discusses previous studies on parent selection as a critical attribute of the search for exploration and exploitation in GA and the NE methodology, which includes parent selection and variability for dynamic environments. In Section 3, we present and discuss the features of NEAT-HD. Section 4 explains our experimental methodology and our testing results and discussion for a simple task. In addition, we examined the performance of NEAT-HD against another more difficult task. Concluding remarks and suggestions for future research are summarized in Section 5.

2. Related Works

1.- Exploration and Exploitation in parent selection

To develop a successful algorithm, Eas, there is a need to address the exploration and exploitation of a search space (Črepinšek et al., 2013). The former aims to approach entirely new areas of search space to find better solutions. The latter seeks to visit a search space of previously visited points to sophisticate the solutions already found. As a rule of thumb, maintaining a good ratio between exploration and exploitation is required to establish a practical algorithm, that is, to control operators such as selection of individuals that enable offspring, mutation, and crossover. Parent selection is also among the significant components that influence the balance between exploration and exploitation.

1.1 – Parent selection in genetic algorithms

Fitness proportional-based methods such as roulette wheel selection, tournament selection, and rank selection (Thierens and Goldberg, 1994) are widely used for parent selection. All these operators are based on the common assumption that individuals with higher fitness are more likely to generate offspring than those with lower fitness. Parents are chosen by non-uniform sampling, which depends only on fitness.

1.2 – Parent selection and diversity preservation

Some algorithms are designed to trigger innovation in search space exploration by utilizing parent selection, allowing crossover between different types of relatively dissimilar solutions. For instance, Drezner and Marcoulides (2003) and Farias and De Magalhães (2018) proposed algorithms in which one parent was chosen randomly from the population. The second parent was selected from randomly chosen *K* individuals and chooses the most dissimilar one. This second parent is often neither the most similar nor the most dissimilar from the first parent. It has been reported to preserve the diversity of the whole population.

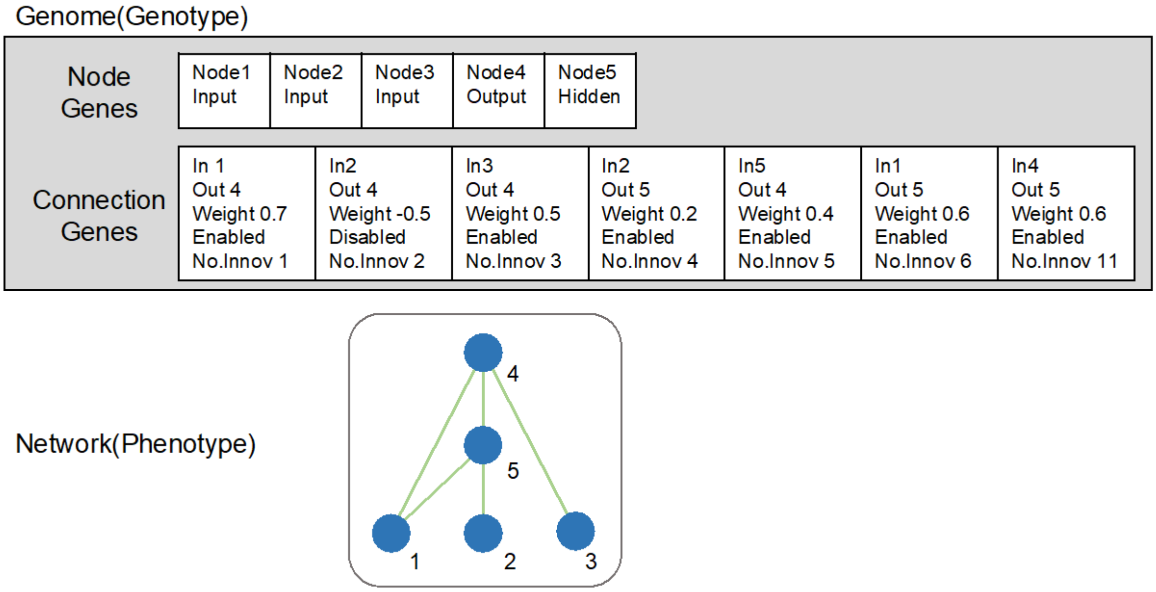
2.- NeuroEvolution of Augmenting Topologies (NEAT)

NEAT is a highly sophisticated algorithm that finds the optimal neural network structures that solve a wide range of tasks (Dinh et al., 2015; Piacentino, 2009; Stanley et al., 2005; Stanley et al., 2019). NEAT has gained attention in the area of NE because it does not rely on a fixed neural network structure. However, it allows the evolution of its topology structures that perform significantly well. Compared to previous works, NEAT has some efficiency, such as assigning an ID called historical markings that avoids the loss of information embedded in the topology and making a sub-group to prevent an innovative network from being excluded from the population, called speciation.

2.1.- Encoding

NEAT represents the phenotype directly from the genotype (Fig. 1). The genotype contained two lists of genes. A node gene in the genotype corresponds to a node in the network.

A connection gene corresponds to a connection between two nodes in the network. The historical markings assigned to each gene make them distinguishable. Mutation and crossover occur by adding a change to the list of genes.



Node1	Node2	Node3	Node4	Node5
Input	Input	Input	Output	Hidden

In 1	In2	In3	In2	In5	In1	In4
Out 4	Out 4	Out 4	Out 5	Out 4	Out 5	Out 5
Weight 0.7	Weight -0.5	Weight 0.5	Weight 0.2	Weight 0.4	Weight 0.6	Weight 0.6
Enabled	Disabled	Enabled	Enabled	Enabled	Enabled	Enabled
No.Innov 1	No.Innov 2	No.Innov 3	No.Innov 4	No.Innov 5	No.Innov 6	No.Innov 11

Fig 1. Example of mapping from genotype to phenotype. For one genotype, two genes were listed: node genes and connection genes. The numbers (1, 2, …) and the type (Input, Output, Hidden) are assigned in the node genes. In the connection nodes, the direction of the connection (e.g., In 1–4), and the weight are assigned. Additionally, the genes allocated as “Disabled” were not expressed in the phenotype. The historical markings are assigned to each gene to identify each gene (e.g., No.innov 1).

2.2.- Crossover

When comparing two individuals, the use of historical markings helps to distinguish an identical gene (Fig. 2). Two lists of genes are lined up, and identical genes are regarded as matching genes. Otherwise, the non-identical genes are regarded as disjoint or excessive, producing a difference between the two individuals. The matching genes are always inherited from the parents. Disjoint and excess genes are inherited randomly. In this way, an offspring is generated that avoids the competing convention problem. The competing convention problem is that the two networks have an identical topology structure, although the genotype representation is different, making them incompatible for crossover.

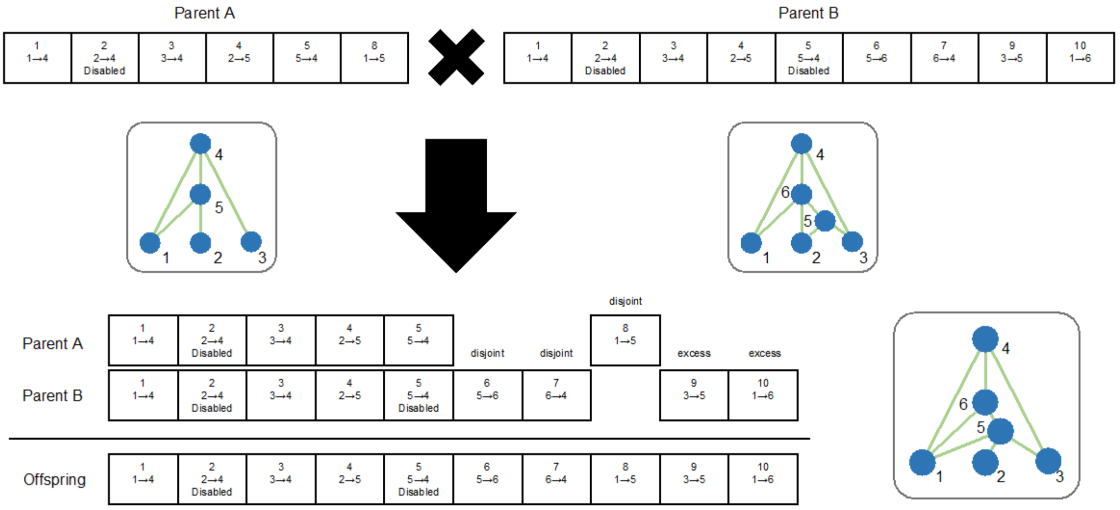


Fig 2. An example of the crossover between two individuals (parent A and parent B). The top number of the genes is the historical markings. Although the structures of the individuals are different, each gene is aligned. Thanks to historical markings, the identical genes, called matching genes, are aligned together. Matching genes are directly inherited to offspring. For the non-identical genes such as disjoint and excess are randomly inherited to offspring. The offspring illustrated is an example when all genes are inherited.

2.3.- Mutation

Two types of mutations occur in the NEAT (Fig. 3). In the add node mutation, a new node splits an existing old connection and re-assigns new historical markings to new split connections (Fig. 3, top). The old connection was disabled. A new connection connects the two existing old nodes in the add connection mutation. It assigns a historical marking to the new connection (Fig. 3, bottom). These mutations make the individual innovative, which has never been seen previously. Although innovative individuals are likely to die early in subsequent generations, speciation saves innovation by dividing the whole population.

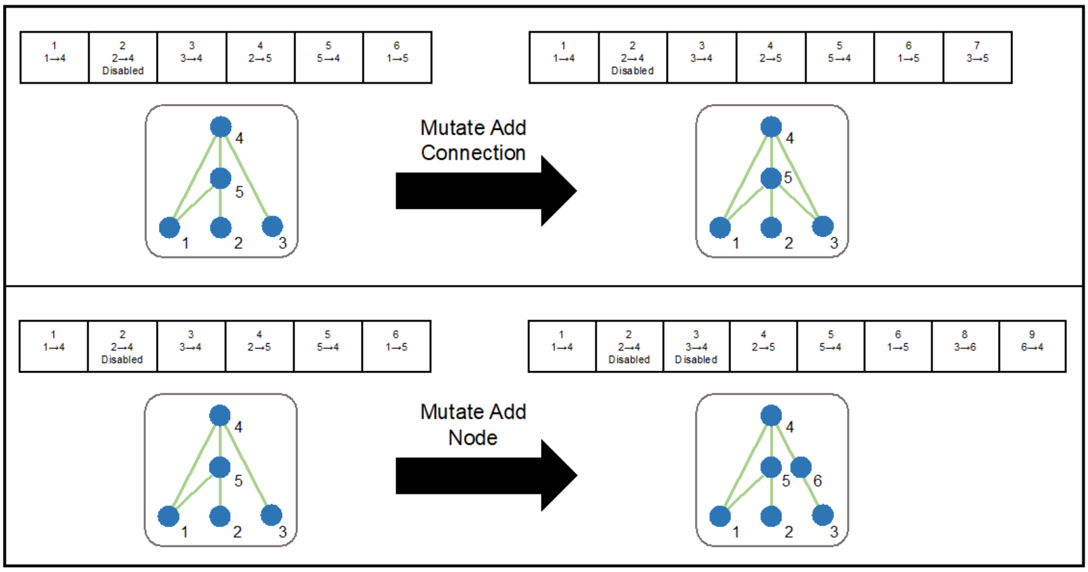


Fig 3. Two types of mutations, add connection mutation (top) and add node mutation (bottom). For the add connection mutation, a new connection is added between selected two nodes, and a new historical marking is assigned to the new connection. For the add node mutation, a new node splits the already-existing connection into two new connections. The new two connections are assigned new historical markings, and the old connection is re-defined as "Disabled".

2.4.- Speciation

Some new connections made by a mutation are added to an individual that might become innovative. This individual is often likely to die in subsequent generations because the weight of the newly added structure is not maturely optimized for the network. A niching scheme, called speciation, was introduced to protect these networks and expects survival. Speciation divides the population into non-overlapping sets of similar individuals based on their shared evolutionary history.

To divide the population into species such that similar structures are in the same species, speciation uses a compatibility function to determine whether two individuals should be in the same species or not. Owing to the historical marking, each gene is comparable to the difference between two individuals. When comparing the genotypes of two selected individuals, identical or non-identical genes were detected. Some genes are in excess or disjoint. These genes are counted as their difference, converted to the compatibility distance to express the relative similarity of the two selected networks. The distance δ between two individuals is a simple linear combination of the number of excess *E* and disjoint *D* genes and, as well as the average weight differences of matching genes *W*, including disabled genes.

$$\delta = \frac{c_1 E}{N} + \frac{c_2 D}{N} + c_3 \cdot \overline{W} \tag{1}$$

where c_1, c_2, and c_3 are the adjusted coefficients, and *N* is the number of genes in the larger genome. The distance measure δ allows us to use a compatibility threshold to divide the group into species. In each generation, the individuals were sequentially placed into the species.

Suppose the distance between the focal individual and the representative individual of the species is under the compatibility threshold. In that case, the focal individual is assigned as a member of the species. Otherwise, if the distance between the focal one and the representative individual of every species is greater than the compatibility threshold, a new species originates with the focal individual as its representative.

Having formed a new species, stagnation promotes species extinction in which fitness is not updated within the species for some generations. Apart from stagnation, species can also be extinct when none of the individuals are assigned to that species because of the compatibility threshold.

3.- Speciation and diversity in EC

In EC, speciation restricts access to other species, thus maintaining diversity. As in the case of NEAT, the notion of preserving diversity is helpful because it prevents pre-optimized individuals from being excluded, which might be a novel solution later. Speciation strategies have been developed in algorithms derived from NEAT, such as grouping the population-based hidden neurons (Hadjiivanov and Blair, 2016) or clades sharing a common ancestor (Knapp and Peterson, 2019). To develop diversity, Novelty Search (NS) performed well (Lehman and Stanley, 2011). Its primary strategy is not to set a fitness function related to the domain field but rather to set it related to the individual’s dissimilarity from the population. NS has enabled the development of new quality diversity algorithms (QDs) to generate large collections of diverse high-performance solutions (Cully et al., 2015; Pugh et al., 2016). Additionally, these developments appear to be related to open-endedness, which attempts to understand processes that continue to generate exciting artifacts indefinitely in artificial life (Alife) (Packard et al., 2019; Stanley et al., 2017; Taylor et al., 2016). One recent study incorporates the idea of coevolution, in which both individuals and environments evolve interactively and permanently (Brant and Stanley, 2019; Brant and Stanley, 2020). While NS or QD can actively produce novel individuals, our work aims to incorporate an idea from a biological context. Although some controversy exists that novel algorithms based on metaphors of natural processes lead the area away from scientific rigor (Sörensen, 2015), still, special attention has been paid to natural computations that copy the inherent advantages of different behaviors observed in biological systems. Bio-inspired computation has been explored to solve complex EC tasks (Del Ser et al., 2019).

4.- Challenge to Environmental variability

Many real-world optimization problems are subject to changing environments. They have become one of the most active areas in the EC field for decades (Jin and Branke, 2005; Nguyen et al., 2012). That is, an algorithm not only finds the optimum in the vast search space but also tracks it over time. Various techniques have been developed to adapt to a dynamic environment, such as memory-based (Goh and Tan, 2008; Jiang and Yang, 2016; Wang and Li, 2010), prediction-based (Jiang et al., 2017; Muruganantham et al., 2016; Zhou et al., 2014), and diversity-based (Deb and Karthik, 2007; Li and Yang, 2012). Memory-based methods retrieve archived historical information aimed at convergence. Prediction-based methods collect archives from the past that the population passes along and integrate them into various predictions of a changing optimum. Diversity-based methods enhance population variations for some individuals to address environmental changes. From the perspective of diversity-based methods, one example is that an extension algorithm of NEAT (Krčah, P., 2012) which is capable of maintaining species size based not only on the previous generation but also the arbitral past generations, has shown significant performance over NEAT. The strategy used by the authors involved maintaining the number of species and allowing long-life species to form when the fitness of individuals dramatically changes between generations. As a diversity-based method, we challenge environmental variability by maintaining diversity by choosing parents according to their similarities.

3. Approach: distance-based parent selection

We propose NEAT-HD to employ a parent selection operator, allowing genetically distinct individuals to be chosen as parents (Fig.4(a)). From a biological perspective, genetically close ones (e.g., creatures belonging to the same species) are likely to crossover and leave offspring. Conversely, the more distinct the biological relationship between two individuals, the less likely they are to be parents. Our approach generates a small proportion of hybrid offspring. We demonstrate that NEAT-HD preserves innovation through the speciation scheme and also makes the population robust to dynamic environmental changes by hybrid offspring. Algorithm 1 represents the pseudocode of the novel distance-based parent selection (Fig.4(b)).

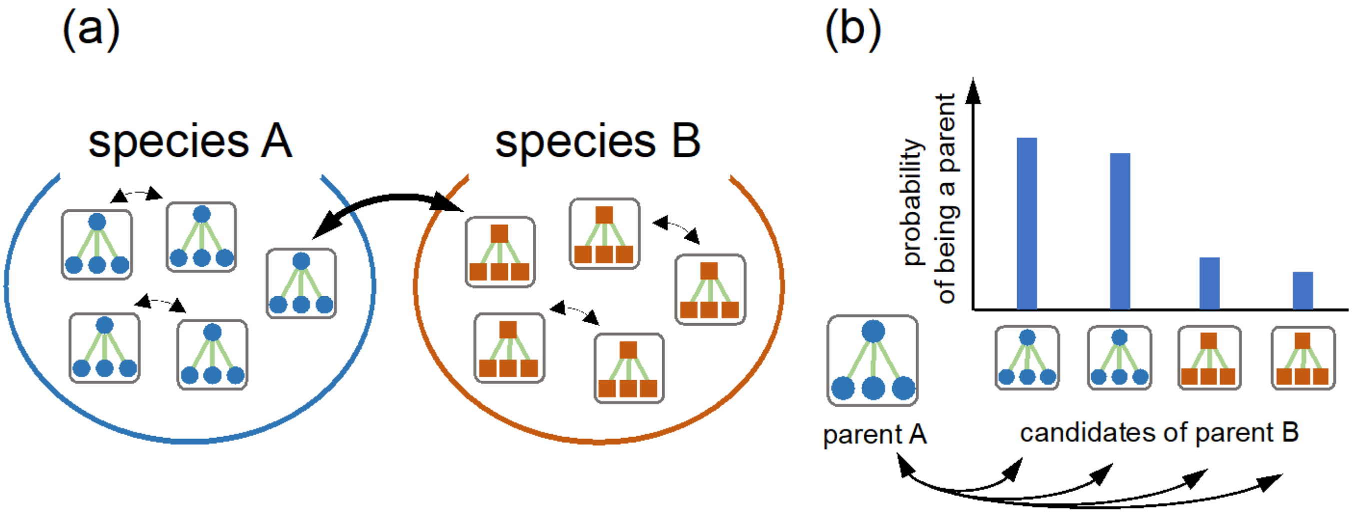


Fig 4. (a) The concept of biological hybridization. In most major cases, crossover happens within the same species (e.g., only species A or species B, represented by small black arrows). However, in a rare case, crossover happens between different species (e.g., species A and B, represented by a large black arrow) that are genetically distinct and generate hybrid offspring. (b) The process of selecting a pair to generate offspring. Select one parent (parent A), and calculate the mating probability through all possible pairs (represented by black arrows) from the whole population. Then choose the other parent (parent B) by non-equal sampling. Though the most possibilities are that individuals within the same species get parents, some rare cases happen so that parents from different species generate offspring.

Algorithm 1 Pseudocode for distance-based parent selection

```
Require: $P = \langle P_1, ..., P_j \rangle$
for $i = 1...n$ do
    $Parent_A \leftarrow Selection(Species(i))$
    for $j = 1...PopulationLength$ do
        $ParentCandidate_B \leftarrow Population(j)$
        $D_j \leftarrow GeneticDistance(Parent_A, ParentCandidate_B)$
        $I_j \leftarrow GeneticIncompatibility(D_j)$
        $P_j \leftarrow MatingProbability(I_j)$
    end for
    $Parent_B \leftarrow SelectParent_B(P)$
end for
```

The algorithm selects each species, excluding low-ranked individuals from the population (Step 2 in Algorithm 1). At each iteration, a set of first parent is generated within the species. For each parent, the candidate of the second parent is selected through the population (Step 4). In Step 5, genetic distances D between individual i and j are calculated (Gaier and Ha, 2019) as follows:

$$D(i,j) = c_1 \frac{G}{N' + 1} + c_2 \bar{W}, \tag{2}$$

where G is the number of non-matching genes of the two individuals, and $\bar{W}$ is the average weight difference of matching genes between individual i and j. The normalization constant N' is the number of added genes to the minimal network of two individuals. c_1 and c_2 are the gene and weight coefficients, respectively. In Step 6, GeneticIncompatibility I is calculated using the genetic distance using (2). Genetic incompatibility refers to the difficulties faced by the offspring by mating two genetically distinct individuals. When genetic incompatibility increases, the probability of creating offspring decreases. Herein, we used a sigmoidal function to associate the genetic distance with genetic incompatibility, which reports a high likelihood of speciation even while allowing hybridization (Yamaguchi and Iwasa, 2017).

$$I(i,j) = \frac{1}{1 + exp\big(-a(-b + D)\big)}, \tag{3}$$

where a and b are parameters that determine the steepness and coordination of incompatibilities of 0.5, respectively. Finally, in Step 7, the probability of being chosen as a parent is given by MatingProbability P:

$$P(i,j) = 1 - I. \tag{4}$$

Using these metrics from (2) to (4), the mating pair is chosen from the population based on the above mating probability.

We expect that utilizing this function will make hybrid offspring more likely to maintain diversity, stability, and reactivity within the population in changing environments.

The NEAT algorithm adopts speciation such that the genetic distance between the representative of a species and each member of the species is under the compatibility threshold defined by an optimizer. This indicates that the genetic distance between two individuals within a species might be greater than the compatibility threshold. Unlike this previous approach, such as implicitly assuming the possibility of mating between genetically distinct individuals, we test an explicit approach that calculates the genetic distance between individuals. Therefore, it might have a chance of crossover between different species. NEAT-HD is tested in two classic control problems: single-cart pole swing-up and lunar lander experiments in Sections 4.

4. Experiments and Results

We conducted two experiments, the cart-pole swing-up, and the lunar lander tasks, using the same parameters (Table 1). NEAT was executed for 50 runs of 300 generations, where 128 agents were the total population using the code from (Gaier and Ha. 2019). NEAT-HD used the parameters $a = 0.2$, $b = 1.3$ in (3). Source code for the experiment is available at https://github.com/j-s9vcp/NEAT-HD.

Table.1

The parameters used in this study.

Parameters	Value
generation	300
population size	128
generations to trigger stagnation	64
number of repetition when evaluating individuals	2
type of speciation (“neat” or “none”)	“neat”
percentage chance of applying second objective when using Multi Objective Optimization	0.0
activation function for hidden nodes	5
probability of adding connection	0.15
probability of adding node	0.1
probability of crossover	0.8
probability to enable disabled connection	0.01
probability of changing node activation function	0.0
probability of mutating connection weight	0.8
probability of enabling each initial connection	1.0
precent of individuals to remove from parent pool before selection	0.1
precent of individuals to pass on to next generation unchanged	0.1
how to scale individual rank (“exponential” or “linear”)	“exp”
number of competitors in each tournament	2
species threshold	1.3
weighting of non-matching genes in compatibility distance calculation	1
weighting of weight differences in compatibility distance calculation	0.5

4.1. Cart-Pole Swing-Up

Choosing a domain with simple input and a relatively small network architecture is desirable to see how the population includes individuals from the same species and different species. We tested the cart-pole swing-up task as our first examination to validate the effectiveness of NEAT-HD, one of the most classic and straightforward control problems.

The goal of the cart-pole swing-up task is to balance the pole upright by moving the cart left and right (Fig.5(a)). The trained ANN has to add nodes by mutation because the cart encounters nonlinear phases such as swinging up the pole or balancing the pole. The cart has six inputs, including the information of the cart and the pole; two of them are the position and velocity of the cart, three are the sine and cosine of the angle and angular velocity of the pole, and the other is the bias term. This setup is identical to that employed by (Gaier and Ha, 2019).

Fitness is evaluated by how long the pole is straight up from the cart and how close the cart is to the optimal position in a fixed time. The fitness is expressed as follows:

$$F = F_{pole} \times F_{cart}. \tag{5}$$

$$F_{pole} = \frac{\cos\theta + 1}{2}. \tag{6}$$

$$F_{cart} = \frac{x - (optimal\ position)}{2.4} \times \frac{\pi}{2}, \tag{7}$$

where θ is the angle from the perpendicular line of the ground to the pole, x is the coordinate of the cart. Over 300 generations, the dynamic change is set at the pre-specified 100th and 200th generations. The $optimal\ position$ at $x = 0$ remains until the first change to $x = -1.5$, and finally, at the second change transition to $x = 1.5$ (Fig.5(b)). We intend to change the magnitude of the transition to observe the differences. The second change was more severe than the first change because the magnitude was more significant. Thus, the second change seems to greatly influence the population, which is close to convergence significantly. Note that the magnitudes of the first and second changes were comparable. The first change has little effect on the second change due to the long-generation pass after the first change, in which the population can re-organize for convergence.

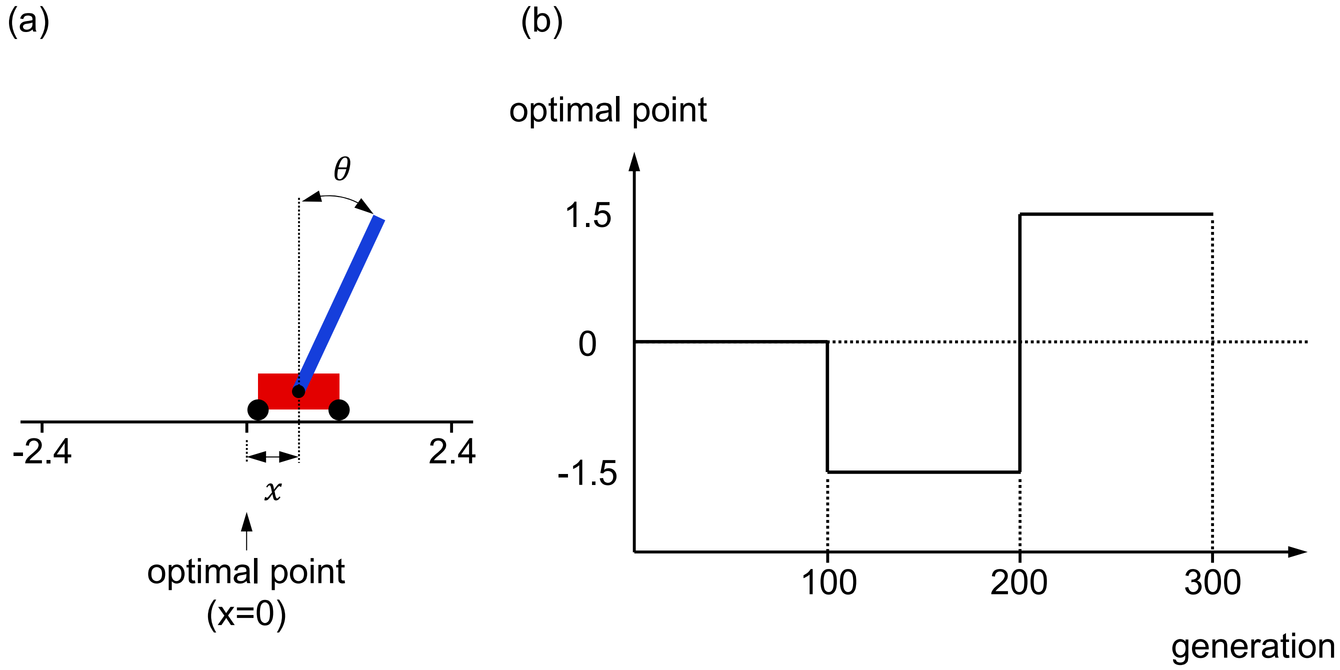


Fig 5. Overview of the cart-pole swing-up task. (a) The goal is to balance the pole upright by moving the cart left and right. (b) the trajectory of the optimal point.

Parents are regarded as the same or different species by genetic distance, proceeding with the algorithm. Here, utilizing a threshold can determine the species identity, called the compatibility threshold, which was initially introduced by (Stanley and Miikkulainen, 2002). When the genetic distance between a parent is under the compatibility threshold, the two individuals are regarded as the same species. As the compatibility threshold directly affects the parent sameness, we changed the value of the compatibility threshold to some extent and explored the difference within the range.

The following three performance metrics were employed to check how the algorithm could endure a dynamic environment. We utilized and modified the performance measures developed by (Weicker, 2002) to this specific study for that purpose.

-Accuracy indicates how the population can suppress dynamic changes that occur suddenly. This metric can be measured by recording the best-fitness observed from the population at generation *t* when the optimal position of the cart changes. The accuracy at generation *t* is defined as:

$$Accuracy_t = \frac{Best_t - Min_t}{Max_t - Min_t}. \tag{8}$$

where $Best_t$ is the best-fitness at generation *t*, Max_t is the possible max-fitness value at generation *t* in the task, Min_t is the possible worst-fitness value at generation *t* in the task. The accuracy ranges from 0 to 1, where 1 is the best possible value.

-Stability is the ratio of the number of iterations that successfully recovered after the dynamic change at generation *t*. It is counted as “recovered” when the mean value of the accuracy after the dynamic change is greater than or equal to the mean value of the accuracy before the dynamic change. The stability at generation *t* is defined as:

$$Stability_t = \frac{\sum_k^{(total\ number\ of\ iterations)} (stability_{iteration=k})}{(total\ number\ of\ iterations)}, \tag{9}$$

$$stability_{iteration=k} = \begin{cases} 1, & if\ \dfrac{\underset{t\leq i\leq t+99}{Best}(accuracy_i)}{\underset{t-100\leq i\leq t-1}{Mean}(accuracy_i)} \geq 1 \\ 0, & otherwise \end{cases}. \tag{10}$$

where $\underset{t-100\leq i\leq t-1}{Mean}(accuracy_i)$ is the mean value of the accuracy between generations $t-100$ and $t-1$. This value corresponds to the accuracy criterion before the dynamic change. $\underset{t\leq i\leq t+99}{Best}(accuracy_i)$ is the best accuracy value between generation t and $t+99$. This value corresponds to the best value that an algorithm can increase from dynamic change. For both $\underset{t-100\leq i\leq t-1}{Mean}(accuracy_i)$ and $\underset{t\leq i\leq t+99}{Best}(accuracy_i)$, 100 generations before and after the change are considered because of the duration of 100 generations between two changes at the 100th and 200th generations.

-Reactivity represents the speed at which the population adjusts their strategies and recovers their fitness from the environmental change at generation *t*. An assumption in this metric is that the iteration is regarded as recovered in the stability metric. Measurements are applied, such as recording how many generations take to catch up with the fitness before the environmental change, the mean value of the accuracy. The reactivity at generation *t* is defined as

$$Reactivity_t = Min\left\{(t'-t) \middle| \frac{accuracy_{t'}}{\underset{t-100\leq i\leq t-1}{Mean}(accuracy_i)} \geq 1\right\} \cup \{maxgen - t\}. \tag{11}$$

where $t, t' \in \mathbb{N}$ and $t < t' \leq maxgen$, with $maxgen$ referring to the total number of generations. Lower values indicate a better and faster reaction to changes.

4.2. Application Results of NEAT-HD on Cart-Pole Swing-Up task

First, it is necessary to verify that hybrid individuals are born, as shown in Figure 6. We can confirm that hybrids were formed more often in NEAT-HD than in NEAT in the low region of the compatibility threshold. This allows mating between genetically distinct parents. Figure 7 shows an overview of the best-fit values along each compatibility threshold. We confirmed that the fitness dropped in both algorithms (NEAT and NEAT-HD) at the 100th and 200th generations, where the optimal position of the cart-pole changes. In addition, it is observed that the drop at the 200th generation is more significant than that at the 100th generation because of the significant dynamic change at the 200th generation. When the compatibility threshold is 0.5, and 0.9, the best fitness of NEAT-HD is slightly below that of NEAT throughout the generation. As the compatibility threshold increased, both algorithms exhibited fewer differences. In the high region of the compatibility threshold, the essential difference between the two algorithms decreases, thus resulting in similar trajectories.

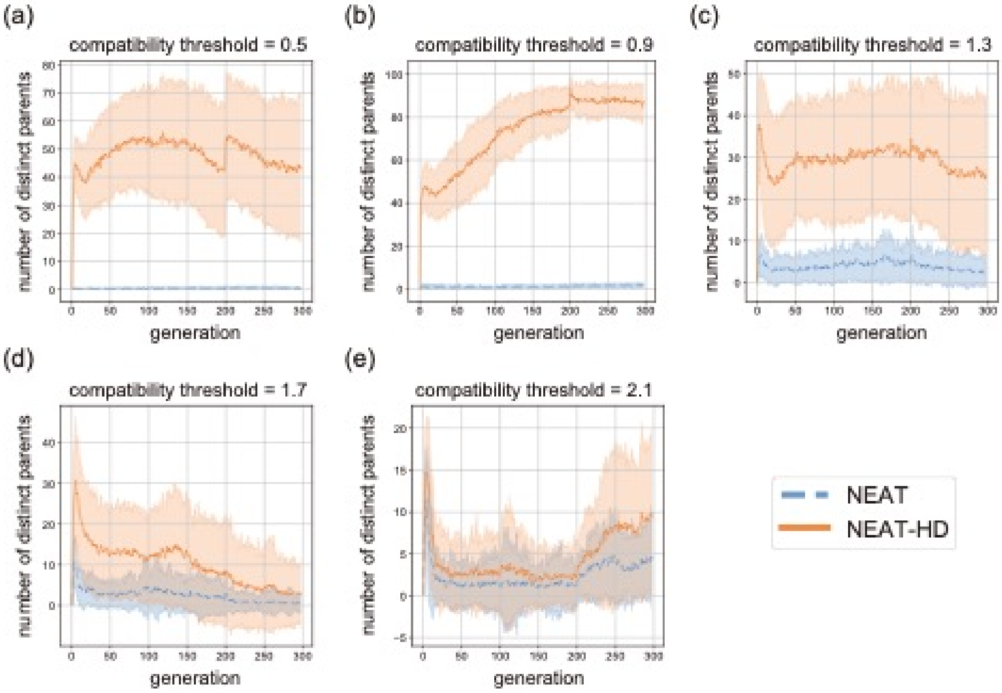


Fig 6. The number of hybrid offspring through generation at each compatibility threshold. The lines represent NEAT-HD, and dashed lines represent NEAT. The shaded regions indicate the standard deviation from the mean.

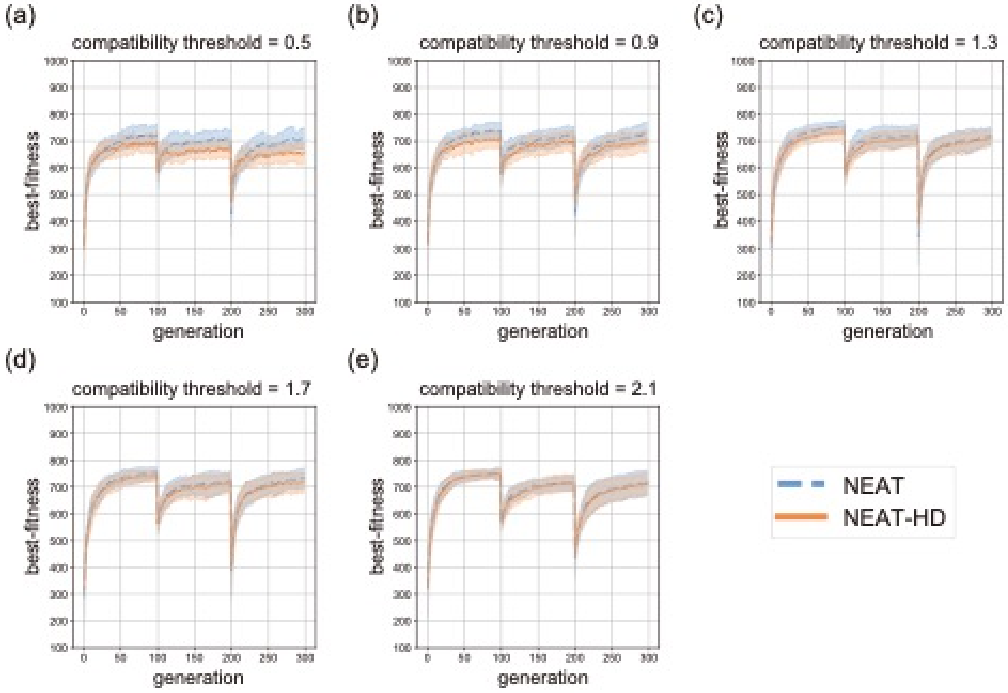


Fig 7. Best-fitness through generation at each compatibility threshold in the cart-pole task. The lines represent NEAT-HD, and dashed lines represent NEAT. The shaded regions indicate the standard deviation from the mean.

We also investigated the three metrics to test the robustness of the proposed algorithm to dynamic environmental changes (Fig. 8 and Table.2). The accuracies at the 100th and 200th generations are illustrated in Figure 8(a) and 8(b), respectively. When a small change occurred at the 100th generation, no significant difference was observed between both algorithms. At the 200th generation, on the other hand, some significance was found at compatibility thresholds of 0.5, 0.9, and 1.3 ($p < 0.05$; Welch's t-test). The difference between the two algorithms appears in the lower region of the compatibility threshold, demonstrating that NEAT-HD endures the dynamic change. These results indicate a trade-off between the best-fitness trajectory and endurance against the change. In the case of stability, it is determined whether the best-fitness recovers or remains after the dynamic change (Table.2). Both algorithms have similar stability because the best-fitness recovers after the dynamic change. More specifically, at the lower region of the compatibility threshold, NEAT-HD showed higher stability. However, NEAT was higher at the high region of compatibility threshold as well. In the case of reactivity, it detected the number of generations required to recover; the shorter the generations that need to recover, the higher their reactivity (Figure 8(c) and 8(d)). At both the 100th and 200th generations, in the low region of the compatibility threshold, significant differences are found so that NEAT-HD only requires a short duration to recover from dynamic change.

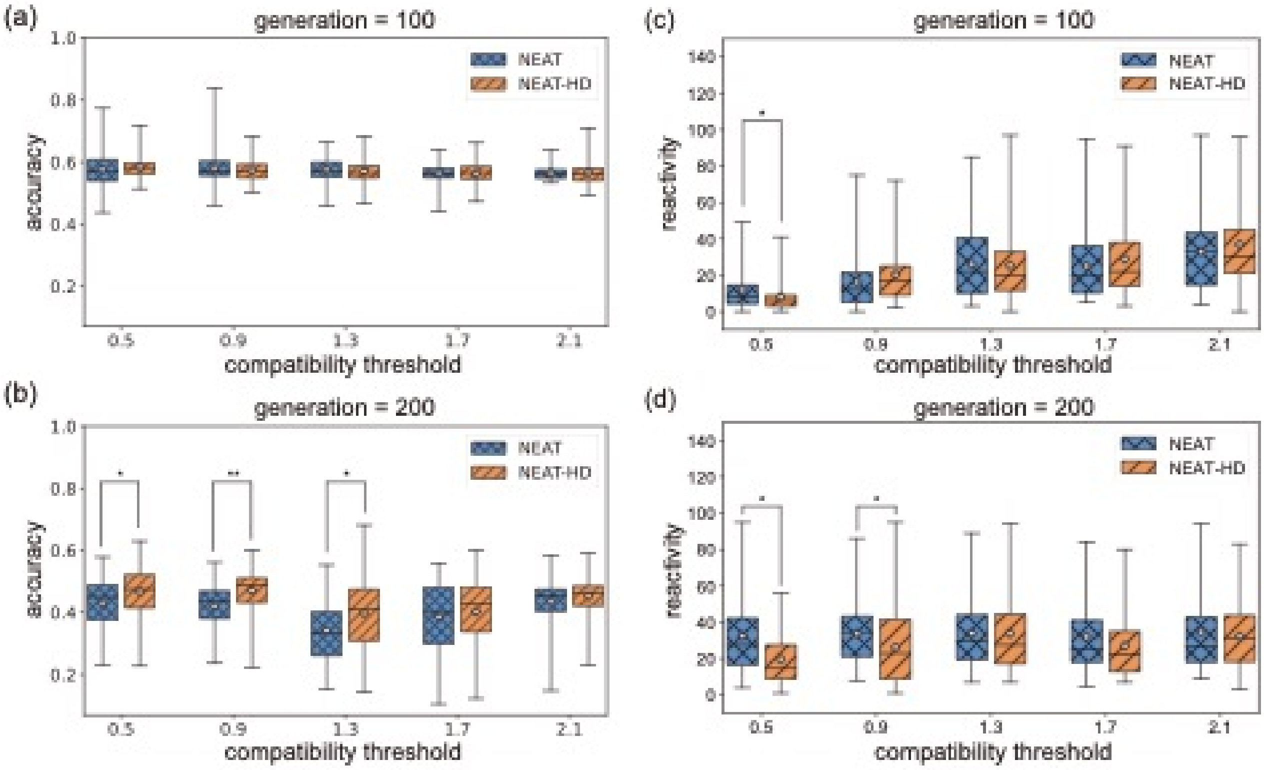


Fig 8. Left side are accuracies at the 100th (a) and 200th (b) generation at each compatibility threshold. Right side are reactivity at the 100th (c) and 200th (d) generation at each compatibility threshold. The white cross in the box represents the mean value.

Table.2

The stabilities of both algorithms (NEAT and NEAT-HD) at each compatibility threshold in cart-pole swing-up task.

compatibility threshold	0.5		0.9		1.3		1.7		2.1	
type	NEAT	NEAT-HD	NEAT	NEAT-HD	NEAT	NEAT-HD	NEAT	NEAT-HD	NEAT	NEAT-HD
generation=100	1.0	1.0	0.94	1.0	0.88	0.94	0.92	0.90	0.86	0.84
generation=200	0.96	1.0	0.96	0.96	0.80	0.84	0.92	0.98	0.86	0.84

To determine the diversity of the population through simulations, the number of species was calculated, as shown in Figure 9. At a compatibility threshold of 0.5, the species number is significantly affected when dynamic changes occur, as the species number falls suddenly. When the threshold was increased, the effect was relaxed. At the compatibility thresholds of 1.3 and 1.7, the decrease in species number stopped after some generations in response to the change, such as the 160th and 260th generations. This is identical to the stagnation duration of approximately 60 generations (see Table.1 for the parameters used) that species hold their individuals when fitness has not been updated. In some cases (e.g., compatibility threshold of 1.3, 1.7), NEAT-HD contributed to maintaining the diversity compared to NEAT, as NEAT-HD holds more species in terms of the mean value. In the other region, it is difficult to say that the contribution of NEAT-HD is greater than that of NEAT.

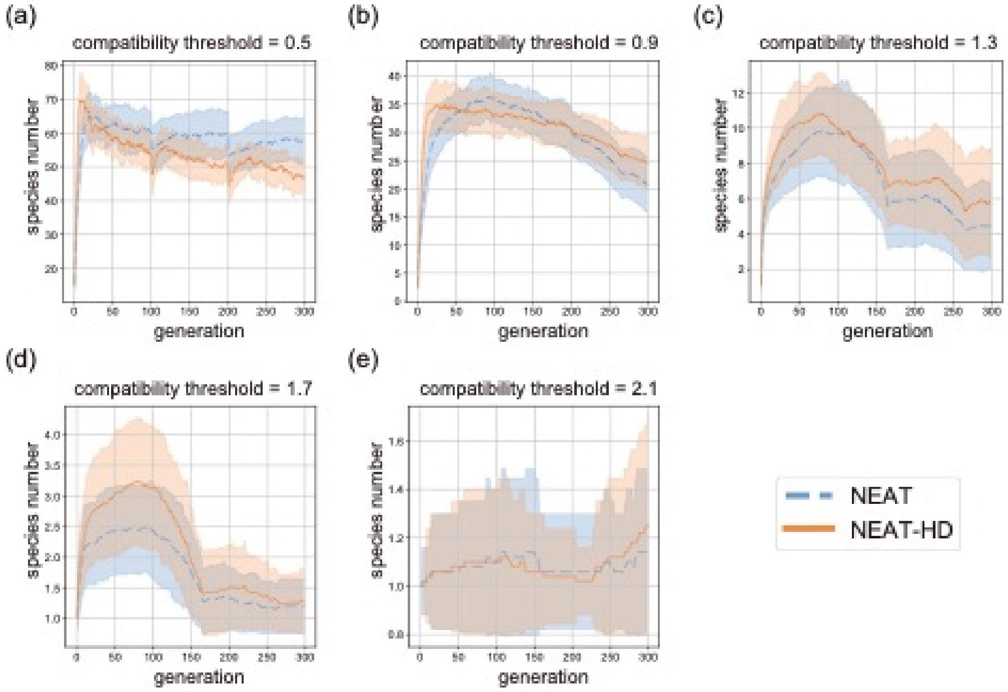


Fig 9. The species number through generation at each compatibility threshold. The lines represent NEAT-HD, and dashed lines represent NEAT. The shaded regions indicate the standard deviation from the mean.

The numbers of newly formed and extinct species are shown in Figure 10. In the lower region of the compatibility threshold, the number of newly formed and extinct species was higher in NEAT-HD than in NEAT. As the threshold increases, the difference decreases toward disappearance. Figure 11 shows another consequence for observing a significant difference in the species' duration between NEAT-HD and NEAT at the compatibility threshold of 0.5, 0.9, 1.3, and 1.7 ($p < 0.001$; Welch's t-test). This result implies that the cycles of the current model from formation to extinction are shorter due to the high rate of species formation and extinction. Figures 8(c), 8(d), 10, and 11 show the sigmoid effect proposed in (Yamaguchi and Iwasa, 2017). More specifically, as a new species is generated, the species becomes extinct shortly after that, and another new species is formed. This short cycle constantly generates new species during the speciation process of NEAT-HD. As species formation is rapid in NEAT-HD, species extinction is fast, resulting in a decrease in the species number in the compatibility threshold of 0.5 (Figure 9 (a)).

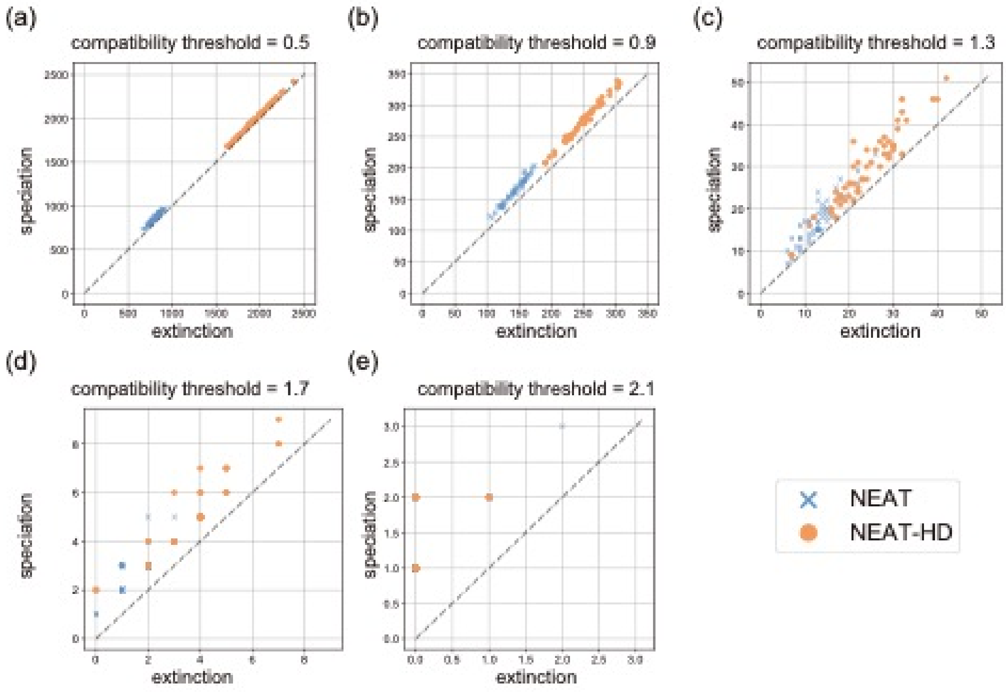


Fig 10. The relationship between the total number of the species and the number of species extinction counted through the run at each compatibility threshold. The orange circle represents NEAT-HD, and the blue cross represents NEAT. The baseline drawn by dashed line indicates that all species formed through the generation were entirely extinct until the last generation.

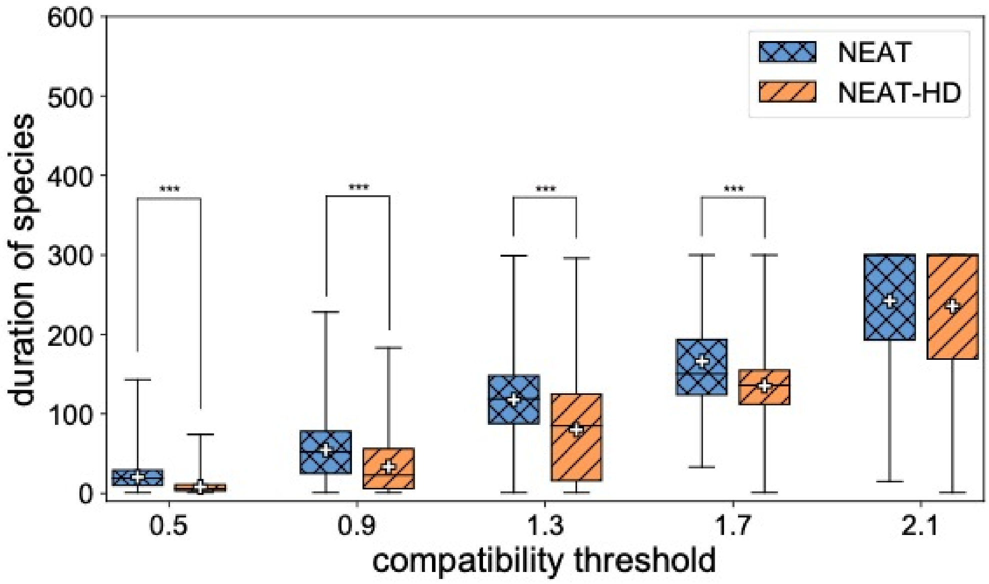


Fig 11. Duration of species at each compatibility threshold. The white cross in the box represents the mean value.

Figure 12 depicts the number of novel networks observed through the generations, which were not observed in previous generations. We defined "novel" as to whether the topology of the focal network was ever seen before in the whole prior generations. The weights on the edges between nodes are neglected to determine the pure difference in the topological architecture. Over generations, the number of novel networks increased in both models. In the last 300th generation, NEAT-HD was significantly higher than NEAT at the compatibility threshold of 0.5, 0.9, 1.3, and 1.7 ($p < 0.05$; Welch's t-test), indicating the effect of forming hybrid individuals from different species.

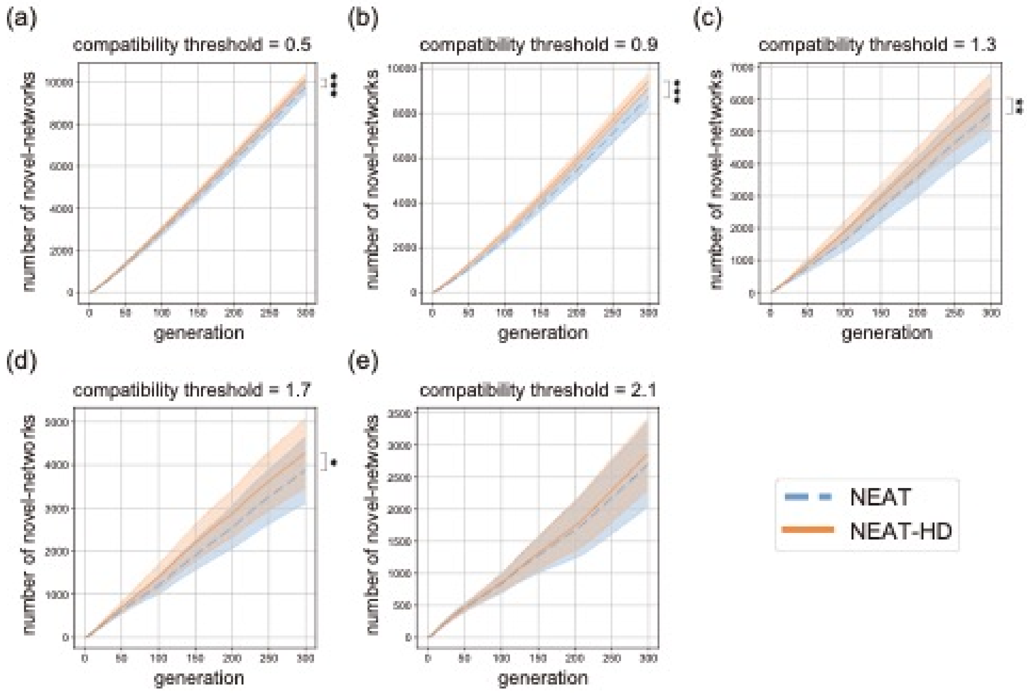


Fig 12. The number of novel networks through generation at each compatibility threshold. The lines represent NEAT-HD, and dashed lines represent NEAT. The shaded regions indicate the standard deviation from the mean.

From the overall results, as the compatibility threshold decreases, the properties of NEAT-HD have been increasingly highlighted, such as the formation of hybrid offspring, the acceleration the emergence of new species, and the extinction with short species duration, all of which influence the endurance in terms of accuracy. We note one disadvantage of NEAT-HD, which affects the best-fitness to decrease to some extent, as shown in Figure 7.

4.3. Lunar-Lander

Difficult domains include more inputs that are given more information about the environment. Another concern is that the domains that can be dynamic are desired in this study, where the optimal position varies across generations. We adopted a Lunar-Lander task as a challenge, a control task often used in machine learning, and a toolkit from Open AI (Brockman et al., 2016). In this task, a lander aiming to land on the moon's surface in a realistic physics simulation was controlled by ANNs (Fig.13(a)).

The task is as follows. The lander is set at a certain height pre-specified and starts to land smoothly using three engines. The engine can adjust its position above the surface and stability of the rotation, with one throttle pointing downward and the other two pointing in the left and right directions. The lander has nine inputs, including the information of the lander and whether the lander is attached to the surface; four of them are the x- and y-axes of the position and velocity, two of which are the angle and angular velocity of the lander, two are for attachment to the surface, and the other is the bias term. The possible output actions are to fire the left orientation engine, fire the main orientation engine, fire the right orientation engine, and do nothing. The task ends if the lander crushes, receiving a penalty reward or rest for a while receiving an additional reward.

Fitness is given as the sum of the rewards using an addition and subtraction method. As mentioned above, if the lander is crushed, the minus reward is assigned, and if the lander rests on the pad for a while, the plus reward is assigned. Additionally, some elements give the lander better rewards through the landing, such as maintaining a stable posture and using less engine fuel. Around the landing pad, the presence of an attachment on the ground provides a reward. We set this task to be dynamic as well as the cart-pole task by changing the optimal position for some generations. By default, the optimal position, the center of the landing pad, is at coordinates (0, 0). We changed the position to (-5, 0) at the 100th generation and (5, 0) at the 200th generation, respectively, over 300 generations (Fig.13(b)). Similar to the cart-pole task, the second change was set to be severe. We tested this task, which was more difficult because of the increased input data; thus, the algorithm must find the optimal solution in the expanded search space compared to the cart-pole task.

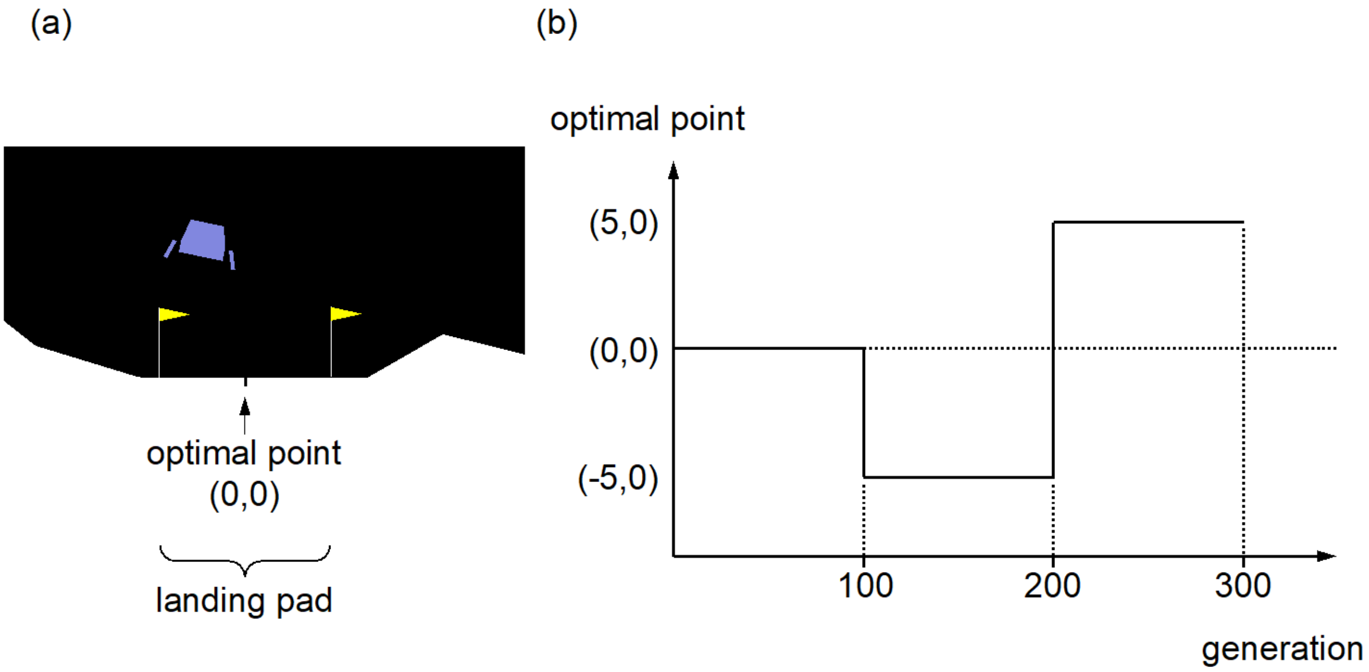


Fig 13. Overview of the lunar lander task. (a) The goal for the lander is to land safely on the ground with less use of engine. The area covered by two flags represents the landing pad. The optimal point is the center of the landing pad. (b) The trajectory of the optimal point.

4.4. Application results of NEAT-HD on the Lunar-Lander task

The number of hybrid individuals is shown in Figure 14. Similar to the cart-pole task, we confirmed that hybrids were formed more often in NEAT-HD than in NEAT in the low region of the compatibility threshold.

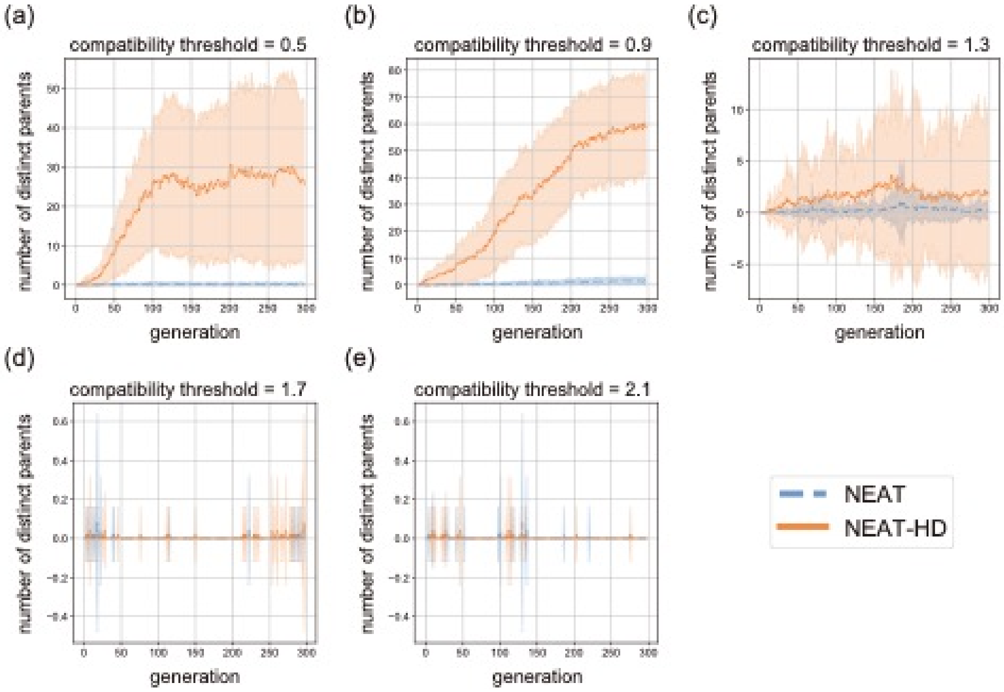


Fig 14. The number of hybrid offspring through generation at each compatibility threshold. The lines represent NEAT-HD, and dashed lines represent NEAT. The shaded regions indicate the standard deviation from the mean.

Over 50 iterations, the trajectory of the best-fitness is plotted in Figure 15. The mean of the best-fitness seemed to show a similar result, as shown in Figure 7. In the lower region of the compatibility threshold, NEAT performs negligibly better than NEAT-HD. As the threshold increased, the difference decreased.

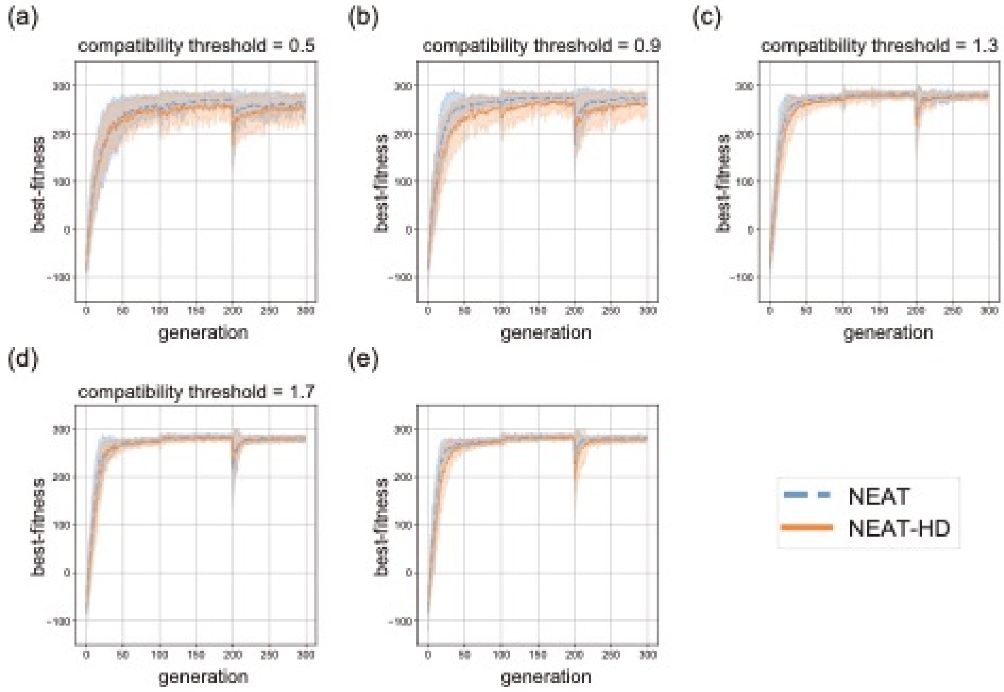


Fig 15. Best-fitness through generation at each compatibility threshold in the lunar lander task. The lines and dashed lines are utilized to separate the algorithm. The shaded regions indicate the standard deviation from the mean.

The accuracy shown in Figure 16(a) and 16(b) indicates that some significant differences were observed at the 100th generation (compatibility threshold of 0.9, and 1.7, $p < 0.05$; Welch's t-test). In addition, at the 200th generation, significant differences were observed (compatibility threshold of 2.1, $p < 0.05$; Welch's t-test). In all cases where differences were significant, the mean value of best-fitness in NEAT was higher than in NEAT-HD. A trade-off observed in the cart-pole task was not apparent here. The stabilities were all 1.0 for each compatibility threshold and the 100th and 200th generations (Table.3). Figure 16(c) and 16(d) illustrates the reactivity at the 100th and 200th generations. At the 200th generation, NEAT-HD showed high reactivity, which was significant compared to NEAT at the compatibility threshold of 0.5, 0.9, and 1.7 ($p < 0.05$; Welch's t-test). These results indicate that NEAT-HD exhibited low endurance in terms of accuracy when dynamic changes occurred, although it recovered quickly in reactivity.

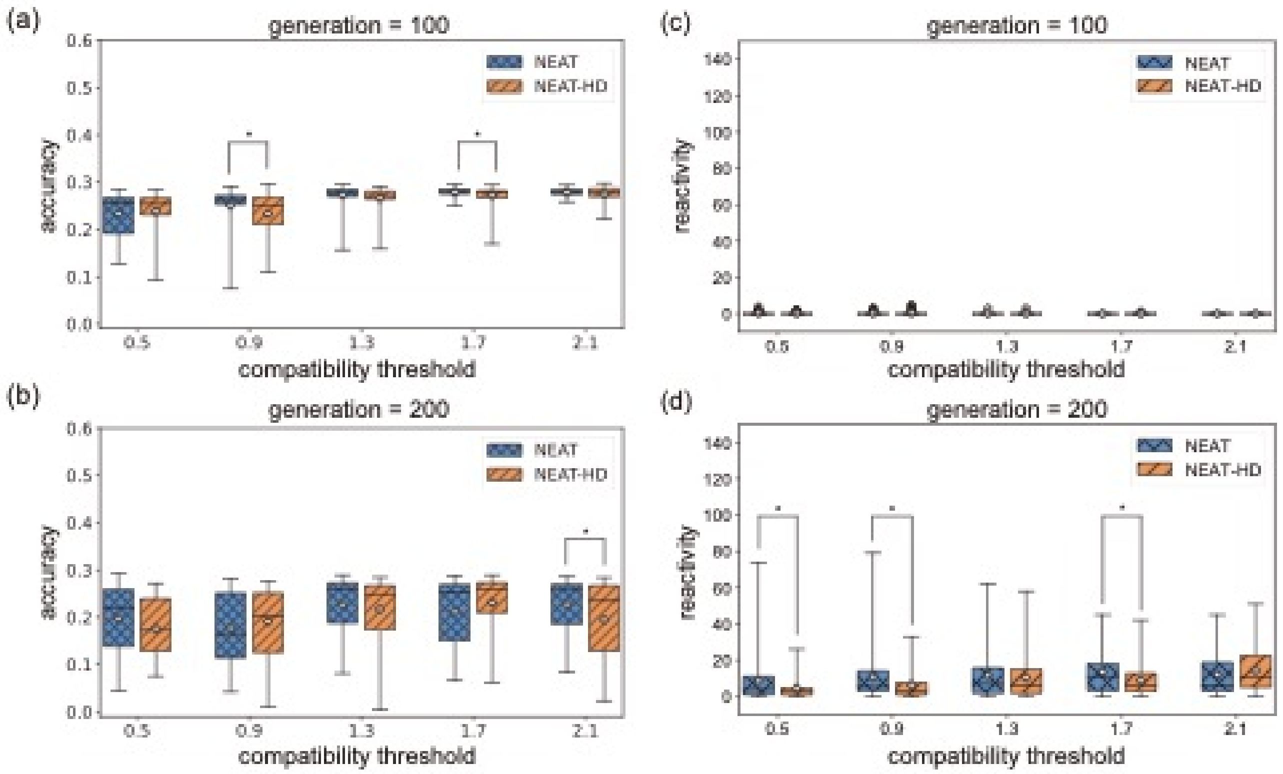


Fig 16. The left side is accuracies at the 100th (a) and 200th (b) generation at each compatibility threshold. The right side shows reactivities at the 100th (c) and 200th (d) generation at each compatibility threshold. The white cross in the box represents the mean value.

Table.3

The stabilities of both algorithms (NEAT and NEAT-HD) at each compatibility threshold in lunar lander task.

compatibility threshold	0.5		0.9		1.3		1.7		2.1	
type	NEAT	NEAT-HD	NEAT	NEAT-HD	NEAT	NEAT-HD	NEAT	NEAT-HD	NEAT	NEAT-HD
generation=100	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
generation=200	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

The species number in Figure 17 as a diversity metric shows similar results for the two tasks compared with Figure 9. At the compatibility threshold of 0.5, as shown in Figure 17(a), the number decreases soon after a sudden rise, similar to Figure 9(a). Figure 18 illustrates the total number of species and the number of extinctions. Figure 18 implies that NEAT-HD generates more species and extinctions, especially in the low region of the compatibility threshold. As the threshold increased, both algorithms overlapped, showing minor differences. Figure 19 shows the duration of each species. At the low region of compatibility threshold (e.g., 0.5, and 0.9), significant differences were found in the sense that species generated by NEAT-HD have a shorter duration until extinction than NEAT ($p < 0.001$; Welch's t-test). Similar to the cart-pole task, a short cycle from the formation of new species to extinction is observed. In addition, at a compatibility threshold of 0.5 (Figure 17(a)), the decrease in the species number can be attributed to the rapid formation of species and extinction.

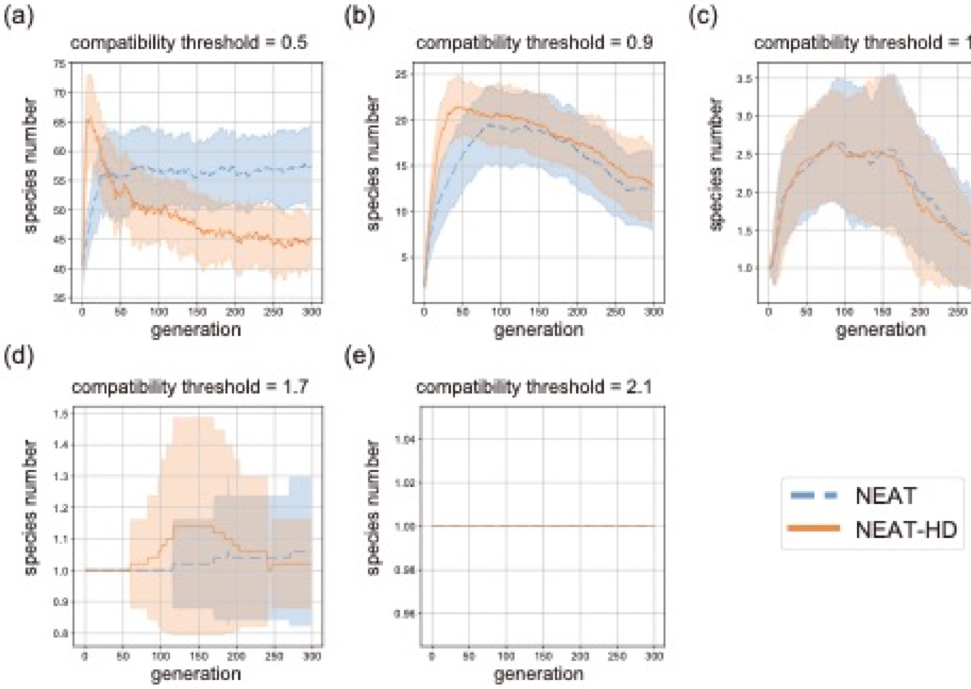


Fig 17. The species number through generation at each compatibility threshold. The lines represent NEAT-HD, and dashed lines represent NEAT. The shaded regions indicate the standard deviation from the mean. At (e), the species number was 1 through the generation and all simulations, expressing only as a line.

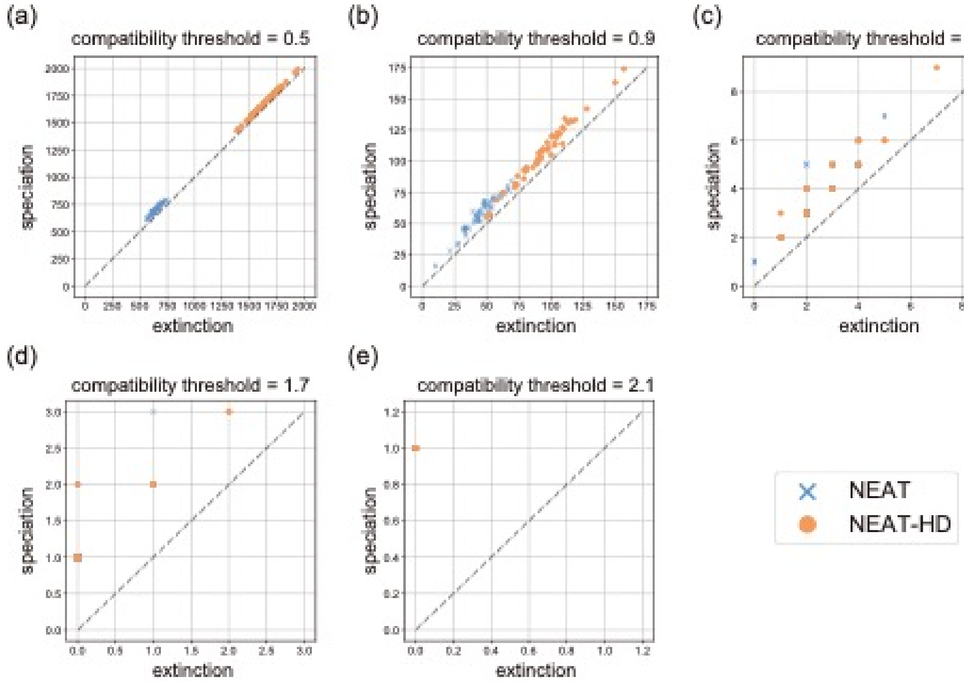


Fig 18. The relationship between the total number of the species and the number of species extinction counted through the run at each compatibility threshold. The orange circle represents NEAT-HD, and the blue cross represents NEAT. Dashed lines are the baseline, which means that all species formed through the generation were entirely extinct until the last generation.

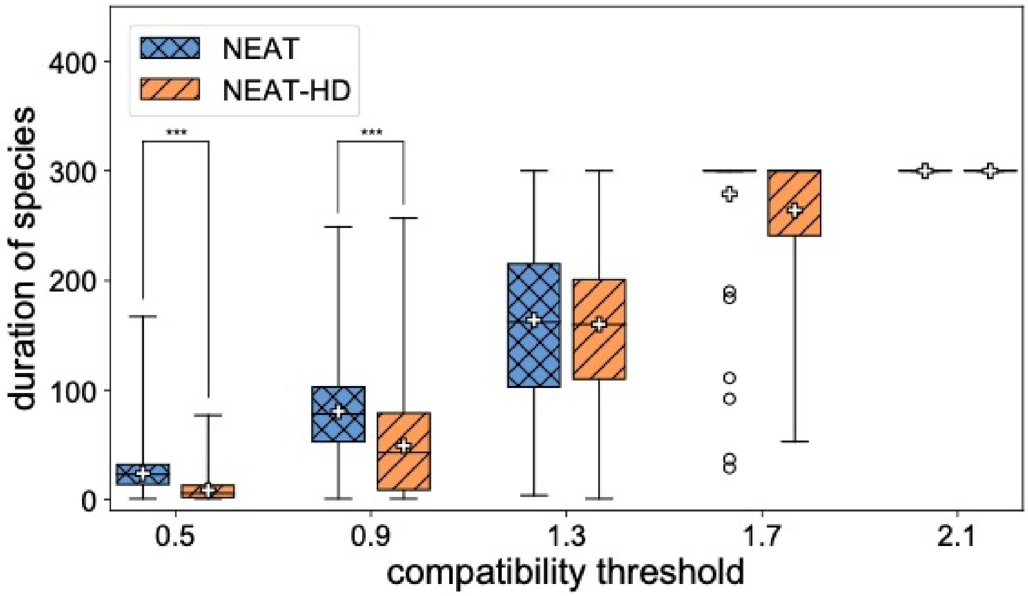


Fig 19. Duration of species at each compatibility threshold. The white cross in the box represents the mean value.

The novel networks that emerged over generations are summarized in Figure 20. Comparing the last 300 generations, hybrid offspring are generated significantly more often at the low region of compatibility threshold of 0.5, 0.9, and 1.7 ($p < 0.05$; Welch's t-test).

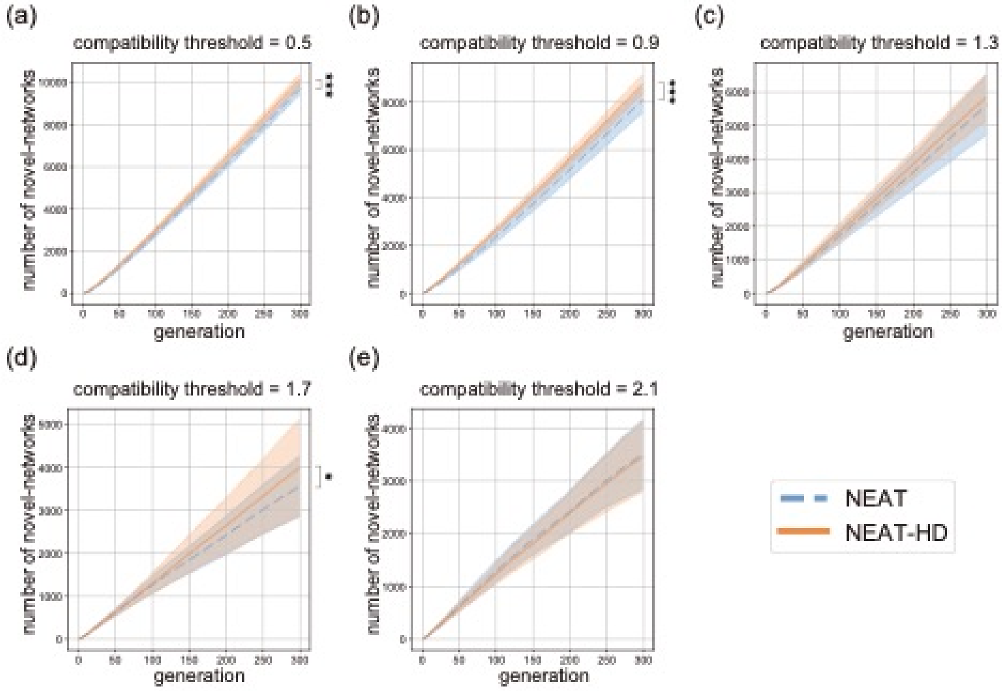


Fig 20. The number of novel networks through generation at each compatibility threshold. The lines represent NEAT-HD, and dashed lines represent NEAT. The shaded regions indicate the standard deviation from the mean.

In this task, from the perspective of species-cycle, the short duration from species formation to extinction affects the diversity, thus expecting to help the endurance against dynamic change. However, the accuracy metrics show the opposite results. We need to determine the distribution of the population diversity that crucially affects the fitness evaluation.

5. Conclusion

This article proposes NEAT-HD, an algorithm that adapts a dynamic environment with a population including offspring from different species. This method attempts to preserve genetic/species diversity by forming hybrid offspring between genetically distinct parents to adapt to environmental changes. An essential advantage of NEAT-HD is that it employs the sigmoid function to determine the probability of mating, as supported by a study in which speciation was accelerated (Yamaguchi and Iwasa, 2017). We argue that some offspring from distinct parents can help endure environmental changes.

We compared NEAT-HD with NEAT for two control tasks: (1) a single cart-pole swing-up task and (2) a lunar lander task. We tested how NEAT-HD behaves in these tasks under various compatibility thresholds. Our results showed two different results. One showed endurance in terms of accuracy when a dynamic change occurred. In addition, the contribution to the diversity was observed in the sense that the production of new hybrid individuals drastically accelerated the formation of species. However, one of the major pitfalls of this study is that the best-fitness is relatively low in NEAT-HD compared with NEAT at the complementary expense for endurance in terms of accuracy. Moreover, in some cases, diversity did not influence endurance in terms of accuracy. Our continued efforts will be made to implement and evaluate new approaches derived from this study.

In future work, it will be necessary to combine NEAT-HD with other diversity-preserving methods, especially in dynamic environments. More specifically, other methods, such as NS or QD, are subject to testing within the same task. For example, the method proposed by (Brant and Stanley, 2019) and (Brant and Stanley, 2020) uses a mutation-only NS that spreads behavioral diversity. Our distance-based scheme can also be applied. Thus, diversity measurements must be unified in the study to compare them appropriately. Furthermore, it is interesting to employ the idea of genetic incompatibility with HyperNEAT (Stanley et al., 2009), an extended version of NEAT. HyperNEAT can treat very large neural network topologies. Thus the impacts on distance-based crossover are expected to be significant. Another concern is the use of our proposed algorithm in the area of multimodal optimization problems. In multimodal optimization problems, approaches that utilize species such as species conservation (Li et al., 2002) are thought of an effective way to capture many global optimums. Our approach also has an expectation of improving the population diversity so that the population can handle multiple solutions within a single run. For future improvement of our proposed method, we must consider the criterion of the algorithm. In this study, genetic distance represents the difference between the two ANNs. Other metrics, such as behavioral representation, might be an alternative option that contributes to the population diversity used in the field of evolutionary robotics (Mouret and Doncieux, 2012).

Acknowledgments

This study was supported in part by Grants-in-Aid for JSPS Scientific Research (KAKENHI) 21K19813, JST MIRAI JPMJMI19B1, and JST Moonshot RD – MILLENNIA Program Grant Number JPMJMS2024-9 (to S.N.).

References

Brant, J.C., Stanley, K.O., 2019. Benchmarking open-endedness in minimal criterion coevolution. Proceedings of the Genetic and Evolutionary Computation Conference.

Brant, J.C., Stanley, K.O., 2020. Diversity preservation in minimal criterion coevolution through resource limitation. Proceedings of the 2020 Genetic and Evolutionary Computation Conference.

Brockman, G., et al. ‘Openai gym.’ arXiv Preprint ArXiv:1606.01540, 2016.

Črepinšek, M., Liu, S.-H., Mernik, M., 2013. Exploration and exploitation in evolutionary algorithms: A survey. ACM Comput. Surv. 45, 1–33. Doi:10.1145/2480741.2480752.

Cully, A., Clune, J., Tarapore, D., Mouret, J.B., 2015. Robots that can adapt like animals. Nature. 521, 503–507. Doi:10.1038/nature14422.

Deb, K., Karthik, S., 2007. Dynamic multi-objective optimization and decision-making using modified NSGA-II: A case study on hydro-thermal power scheduling, in: International conference on evolutionary multi-criterion optimization. Springer, Berlin, Heidelberg, pp. 803–817.

Del Ser, J. et al., 2019. Bio-inspired computation: Where we stand and what’s next. Swarm Evol. Comput. 48, 220–250. Doi:10.1016/j.swevo.2019.04.008.

Dinh, H.Q., Aubert, N., Noman, N., Fujii, T., Rondelez, Y., Iba, H., 2015. An effective method for evolving reaction networks in synthetic biochemical systems, in: IEEE Trans. Evol. Computat. 19, 374–386. Doi:10.1109/TEVC.2014.2326863.

Drezner, Z., Drezner, T.D., 2020. Biologically inspired parent selection in genetic algorithms. Ann. Oper. Res. 287, 161–183. Doi:10.1007/s10479-019-03343-7.

Drezner, Z., Marcoulides, G.A., 2003. ‘A distance-based selection of parents in genetic algorithms.’ Metaheuristics: Computer Decision-Making. Springer, Boston, Massachusetts, pp. 257–278.

Farias, R.G., De Magalhães, C.S., 2018. Parent selection strategies in niching genetic algorithms IEEE Congress on Evolutionary Computation (CEC), 2018, pp. 1–8. Doi: 10.1109/CEC.2018.8477847.

Floreano, D., Dürr, P., Mattiussi, C., 2008. Neuroevolution: From architectures to learning. Evol. Intel. 1, 47–62. Doi:10.1007/s12065-007-0002-4.

Gaier, A., Ha, D., 2019. Weight Agnostic Neural Networks. arXiv Preprint ArXiv:1906.04358.

Goh, C.K., Tan, K.C., 2008. A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization. IEEE Trans. Evol. Computat. 13, 103–127. doi: 10.1109/TEVC.2008.920671.

Grant, P.R., Grant, B.R., 2019. Hybridization increases population variation during adaptive radiation. Proc. Natl Acad. Sci. U. S. A. 116, 23216–23224. doi:10.1073/pnas.1913534116.

Grant, P.R., Grant, B.R., 2011. How and Why Species Multiply: The Radiation of Darwin's Finches. Princeton University Press, Princeton.

Gupta, D., Ghafir, S., 2012. An overview of methods maintaining diversity in genetic algorithms. Int. J. Emerg. Technol. Adv. Eng. 2, 56–60.

Hadjiivanov, A., Blair, A., 2016. Complexity-based speciation and genotype representation for neuroevolution, in: IEEE Congress on Evolutionary Computation (CEC), 2016. IEEE Publications, pp. 3092–3101. doi:10.1109/CEC.2016.7744180.

Jiang, M., Huang, Z., Qiu, L., Huang, W., Yen, G.G., 2017. Transfer learning-based dynamic multiobjective optimization algorithms. IEEE Trans. Evol. Computat. 22, 501–514. doi:10.1109/TEVC.2017.2771451.

Jiang, S., Yang, S., 2016. A steady-state and generational evolutionary algorithm for dynamic multiobjective optimization. IEEE Trans. Evol. Computat. 21, 65–82. doi：10.1109/TEVC.2016.2574621.

Jin, Y., Branke, J., 2005. Evolutionary optimization in uncertain environments—A survey. IEEE Trans. Evol. Computat. 9, 303–317. doi:10.1109/TEVC.2005.846356.

Knapp, J.S., Peterson, G.L., 2019. Natural evolution speciation for neat, in: IEEE Congress on Evolutionary Computation (CEC), 2019. IEEE Publications, pp. 1487–1493. doi:10.1109/CEC.2019.8790153.

Krčah, P., 2012. Effects of speciation on evolution of neural networks in highly dynamic environments. International Conference on Learning and Intelligent Optimization. Springer, Berlin, Heidelberg. doi:10.1007/978-3-642-34413-8_39.

Lehman, J., Stanley, K.O., 2011. Abandoning objectives: Evolution through the search for novelty alone. Evol. Comput. 19, 189–223. doi:10.1162/EVCO_a_00025.

Li, J. P., Balazs, M. E., Parks, G. T., Clarkson, P. J., 2002. A species conserving genetic algorithm for multimodal function optimization. Evol. Comput. 10, 207-34. doi:10.1162/106365602760234081.

Li, C., Yang, S., 2012. A general framework of multipopulation methods with clustering in undetectable dynamic environments. IEEE Trans. Evol. Computat. 16, 556–577. doi:10.1109/TEVC.2011.2169966.

Mallet, J., 2007. Hybrid speciation. Nature. 446, 279–283. doi:10.1038/nature05706.

Mouret, J.-B., Clune, J., 2015. Illuminating Search Spaces by Mapping Elites. arXiv Preprint ArXiv:1504.04909.

Mouret, J.B., Doncieux, S., 2012. Encouraging behavioral diversity in evolutionary robotics: An empirical study. Evol. Comput. 20, 91–133. doi:10.1162/EVCO_a_00048.

Muruganantham, A., Tan, K.C., Vadakkepat, P., 2016. Solving the ieee cec 2015 dynamic benchmark problems using kalman filter based dynamic multiobjective evolutionary algorithm, in:. Proceedings in Adaptation, Learning and Optimization. Springer, Cham, (239–252). doi:10.1007/978-3-319-27000-5_20.

Nguyen, T.T., Yang, S., Branke, J., 2012. Evolutionary dynamic optimization: A survey of the state of the art. Swarm Evol. Comput. 6, 1–24. doi:10.1016/j.swevo.2012.05.001.

Packard, N., Bedau, M.A., Channon, A., Ikegami, T., Rasmussen, S., Stanley, K.O., Taylor, T., 2019. An overview of open-ended evolution: Editorial introduction to the open-ended evolution II special issue. Artif. Life. 25, 93–103. doi:10.1162/artl_a_00291.

Papavasileiou, E., Cornelis, J., Jansen, B., 2021. A systematic literature review of the successors of “NeuroEvolution of augmenting topologies”. Evol. Comput. 29, 1–73. doi:10.1162/evco_a_00282.

Piacentino, G.M., 2009. First Observation of Electroweak Single Top Quark Production. doi:10.1103/PhysRevLett.103.092002

Pugh, J.K., Soros, L.B., Stanley, K.O., 2016. Quality diversity: A new frontier for evolutionary computation. Front. Robot. AI. 3, 40. doi:10.3389/frobt.2016.00040.

Sörensen, K., 2015. Metaheuristics—The metaphor exposed. Int. Trans. Op. Res. 22, 3–18. doi:10.1111/itor.12001.

Stanley, K.O., Bryant, B.D., Miikkulainen, R., 2005. Evolving neural network agents in the NERO video game. Proc. IEEE, 182–189.

Stanley, K.O., Clune, J., Lehman, J., Miikkulainen, R., 2019. Designing neural networks through neuroevolution. Nat. Mach. Intell. 1, 24–35. doi:10.1038/s42256-018-0006-z.

Stanley, K.O., D’Ambrosio, D.B., Gauci, J., 2009. A hypercube-based encoding for evolving large-scale neural networks. Artif. Life. 15, 185–212. doi:10.1162/artl.2009.15.2.15202.

Stanley, K.O., Lehman, J., Soros, L., 2017. Open-Endedness: The Last Grand Challenge You’ve Never Heard Of. While Open-Endedness Could Be a Force for Discovering Intelligence, It Could Also Be a Component of AI Itself.

Stanley, K.O., Miikkulainen, R., 2002. Evolving neural networks through augmenting topologies. Evol. Comput. 10, 99–127. doi:10.1162/106365602320169811.

Taylor, T. et al., 2016. Open-ended evolution: Perspectives from the OEE workshop in York. Artif. Life. 22, 408–423. doi:10.1162/ARTL_a_00210.

Thierens, D., Goldberg, D., 1994. Springer. Convergence models of genetic algorithm selection schemes International conference on parallel problem solving from nature. Berlin, Heidelberg.

Wang, Y., Li, B., 2010. Multi-strategy ensemble evolutionary algorithm for dynamic multi-objective optimization. Memetic Comp. 2, 3–24. doi:10.1007/s12293-009-0012-0.

Weicker, K., 2002. Performance measures for dynamic environments, in:. Lecture Notes in Computer Science. Springer, New York. 2439, 64–73. doi:10.1007/3-540-45712-7_7.

Yamaguchi, R., Iwasa, Y., 2017. A tipping point in parapatric speciation. J. Theor. Biol. 421, 81–92. doi:10.1016/j.jtbi.2017.03.018.

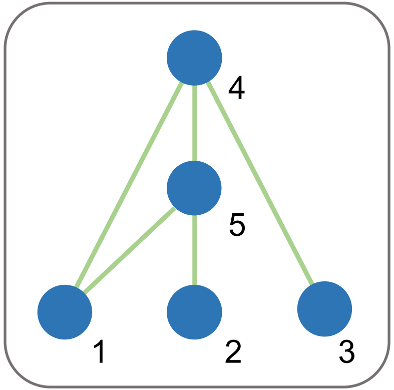
Yamaguchi, R., Otto, S.P., 2020. Insights from Fisher’s geometric model on the likelihood of speciation under different histories of environmental change. Evolution. 74, 1603–1619. doi:10.1111/evo.14032.

Zhou, A., Jin, Y., Zhang, Q., 2014. A population prediction strategy for evolutionary dynamic multiobjective optimization. IEEE Trans. Cybern. 44, 40–53. doi:10.1109/TCYB.2013.2245892.

Genome(Genotype)

Node Genes	Node1 Input	Node2 Input	Node3 Input	Node4 Output	Node5 Hidden		
Connection Genes	In 1 Out 4 Weight 0.7 Enabled No.Innov 1	In2 Out 4 Weight -0.5 Disabled No.Innov 2	In3 Out 4 Weight 0.5 Enabled No.Innov 3	In2 Out 5 Weight 0.2 Enabled No.Innov 4	In5 Out 4 Weight 0.4 Enabled No.Innov 5	In1 Out 5 Weight 0.6 Enabled No.Innov 6	In4 Out 5 Weight 0.6 Enabled No.Innov 11

Network(Phenotype)

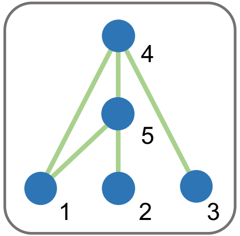


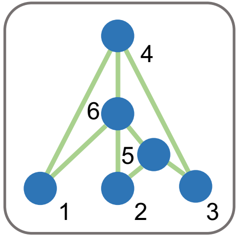
Parent A

1 1→4	2 2→4 Disabled	3 3→4	4 2→5	5 5→4	8 1→5

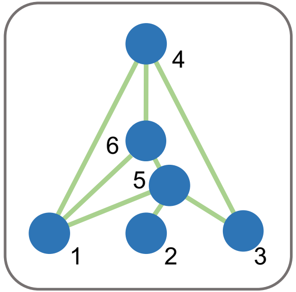
Parent B

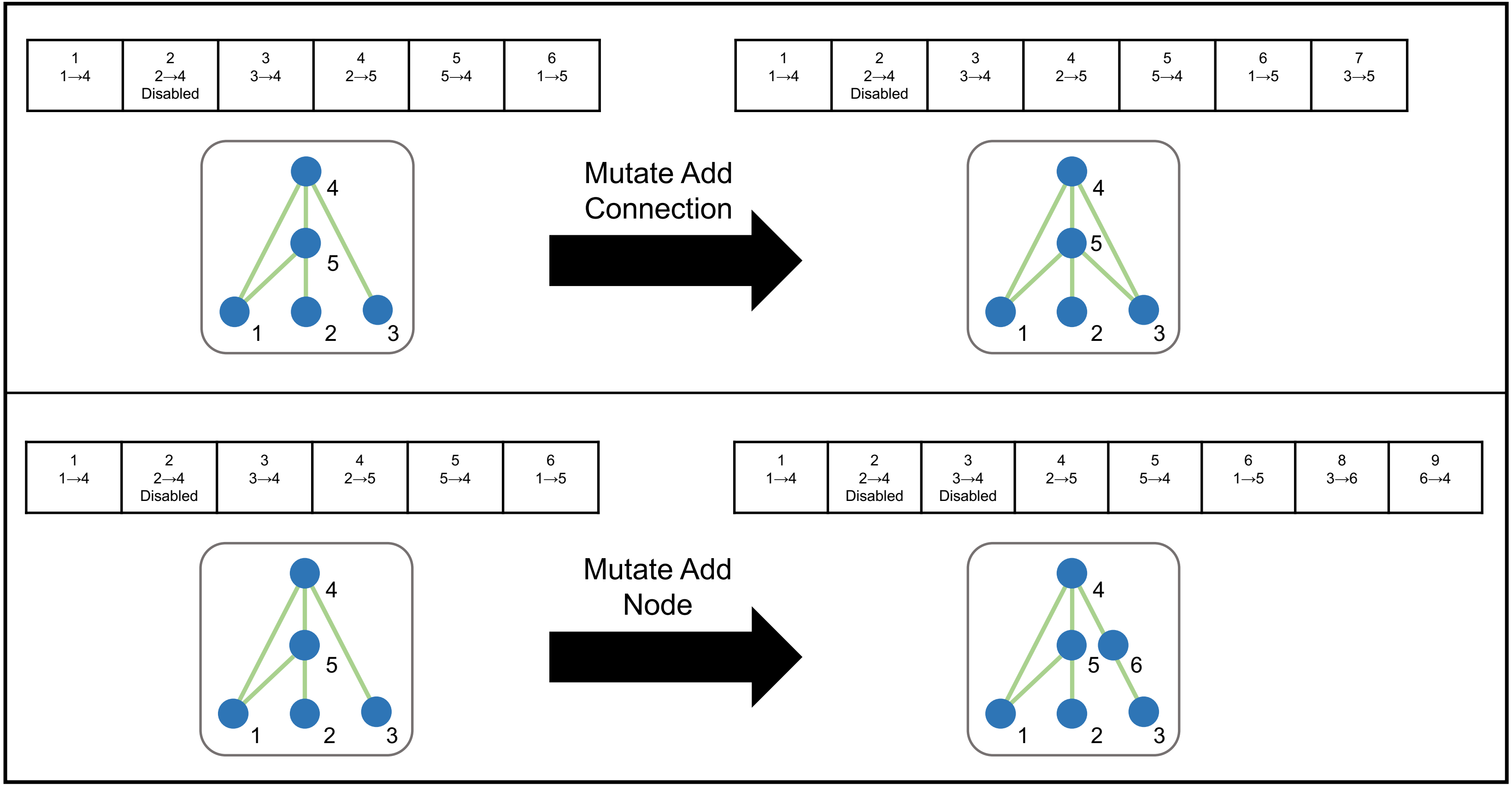
1 1→4	2 2→4 Disabled	3 3→4	4 2→5	5 5→4 Disabled	6 5→6	7 6→4	9 3→5	10 1→6

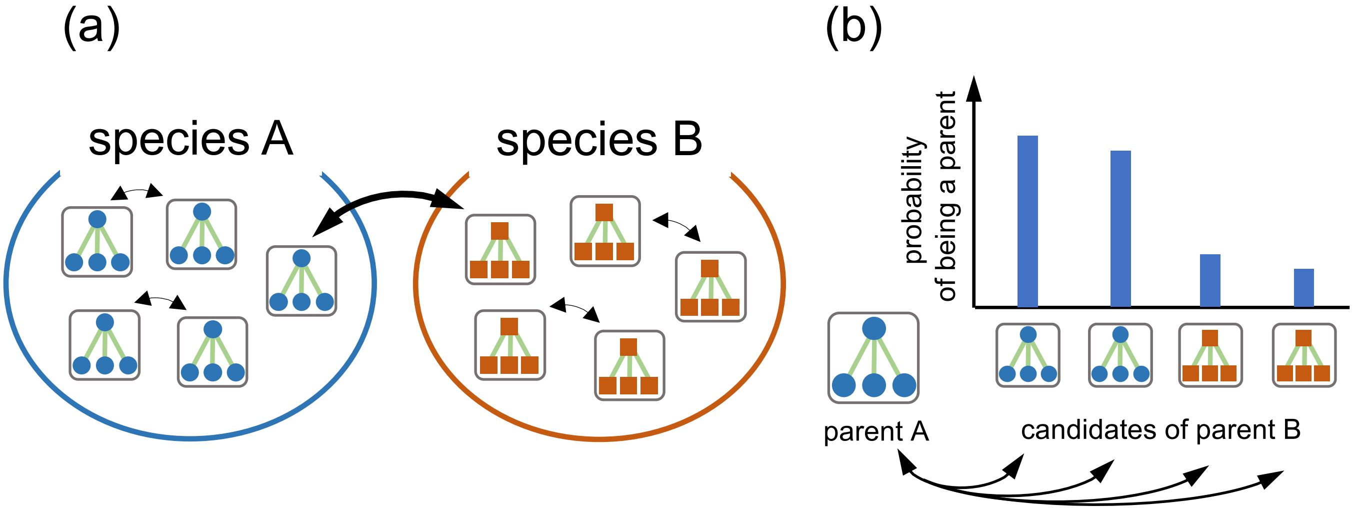


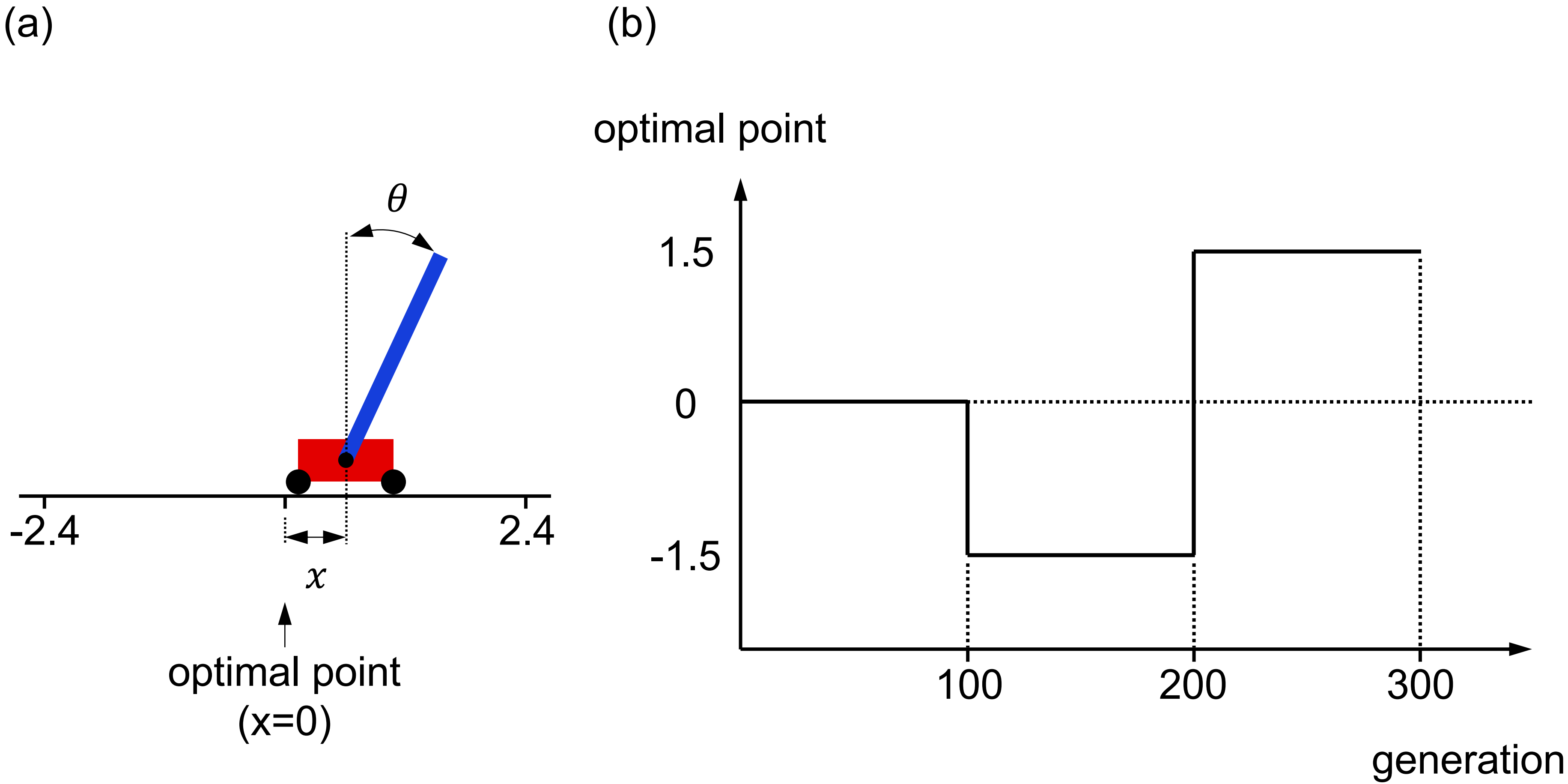


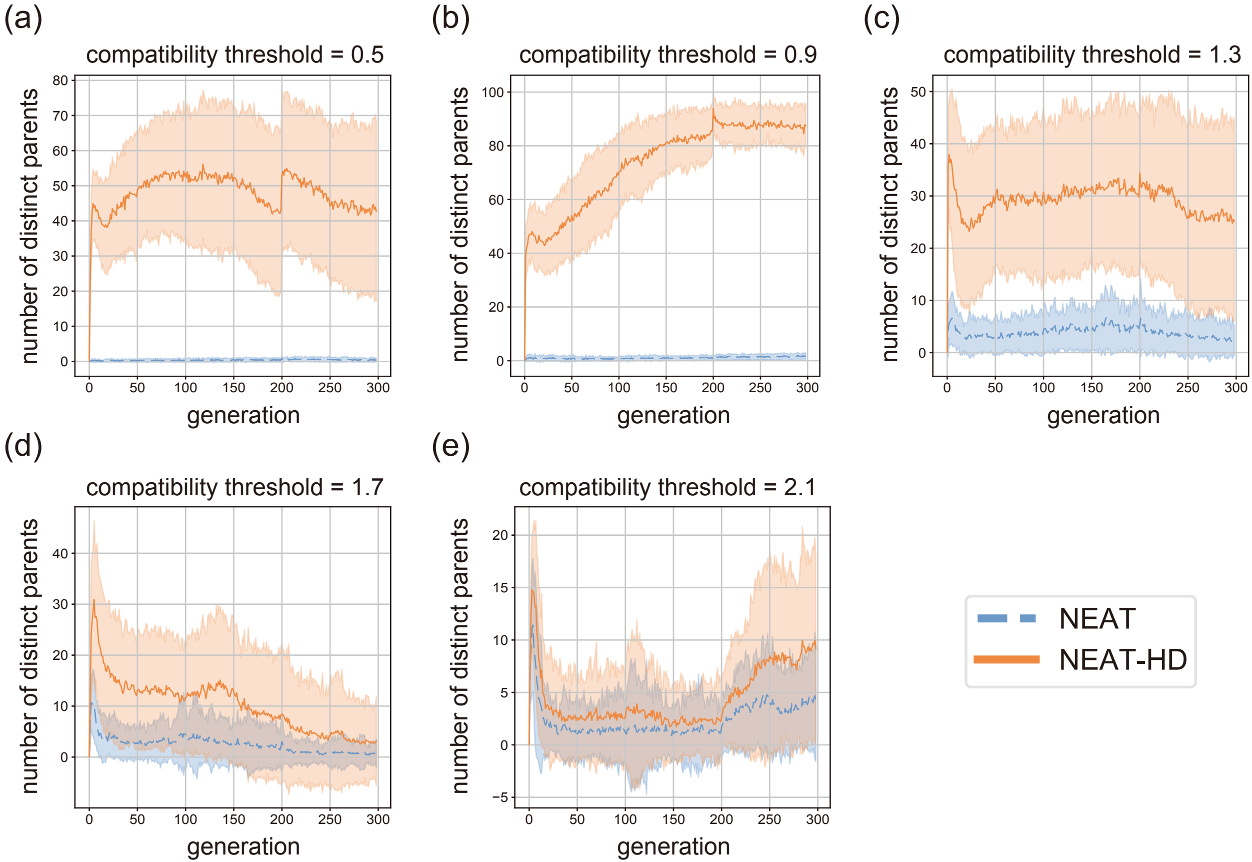
Parent A	1 1→4	2 2→4 Disabled	3 3→4	4 2→5	5 5→4	disjoint	disjoint	disjoint 8 1→5	excess	excess
Parent B	1 1→4	2 2→4 Disabled	3 3→4	4 2→5	5 5→4 Disabled	6 5→6	7 6→4		9 3→5	10 1→6
Offspring	1 1→4	2 2→4 Disabled	3 3→4	4 2→5	5 5→4 Disabled	6 5→6	7 6→4	8 1→5	9 3→5	10 1→6

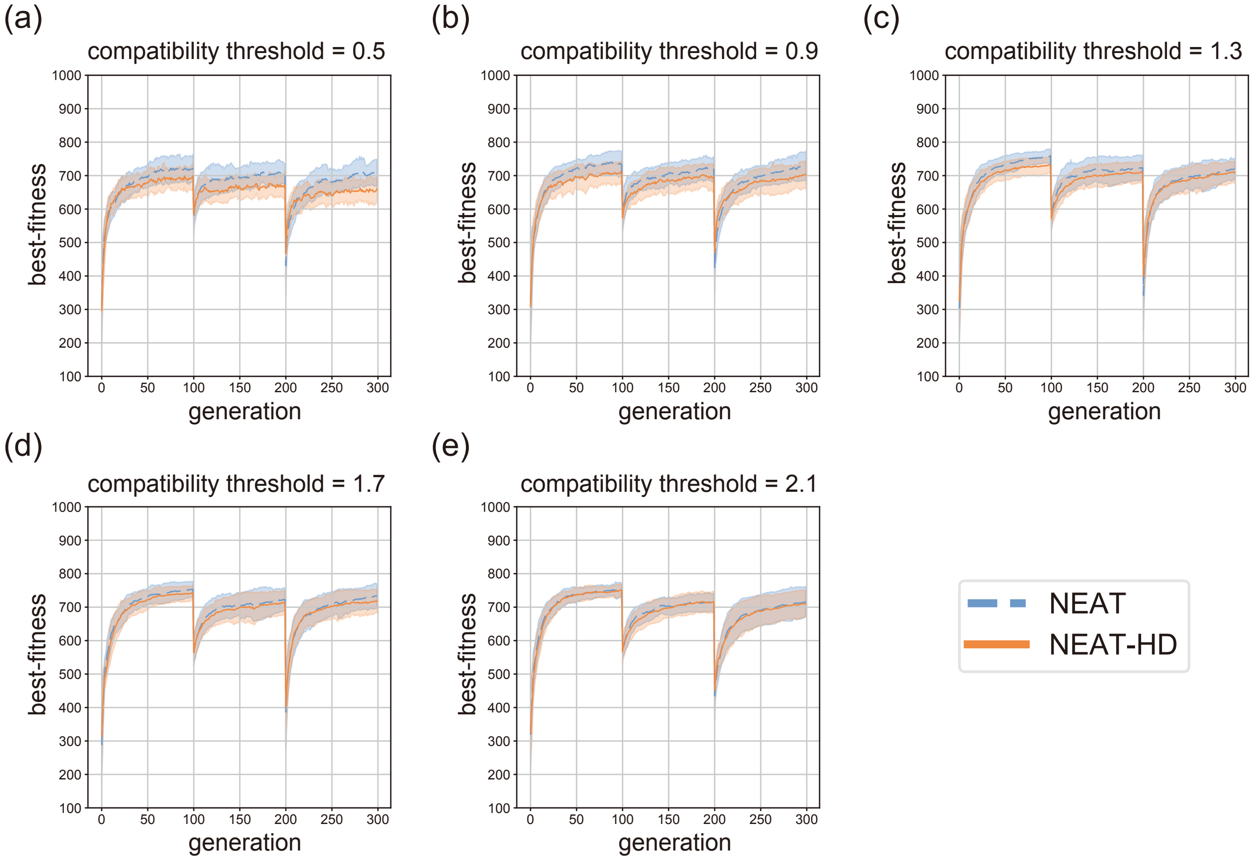


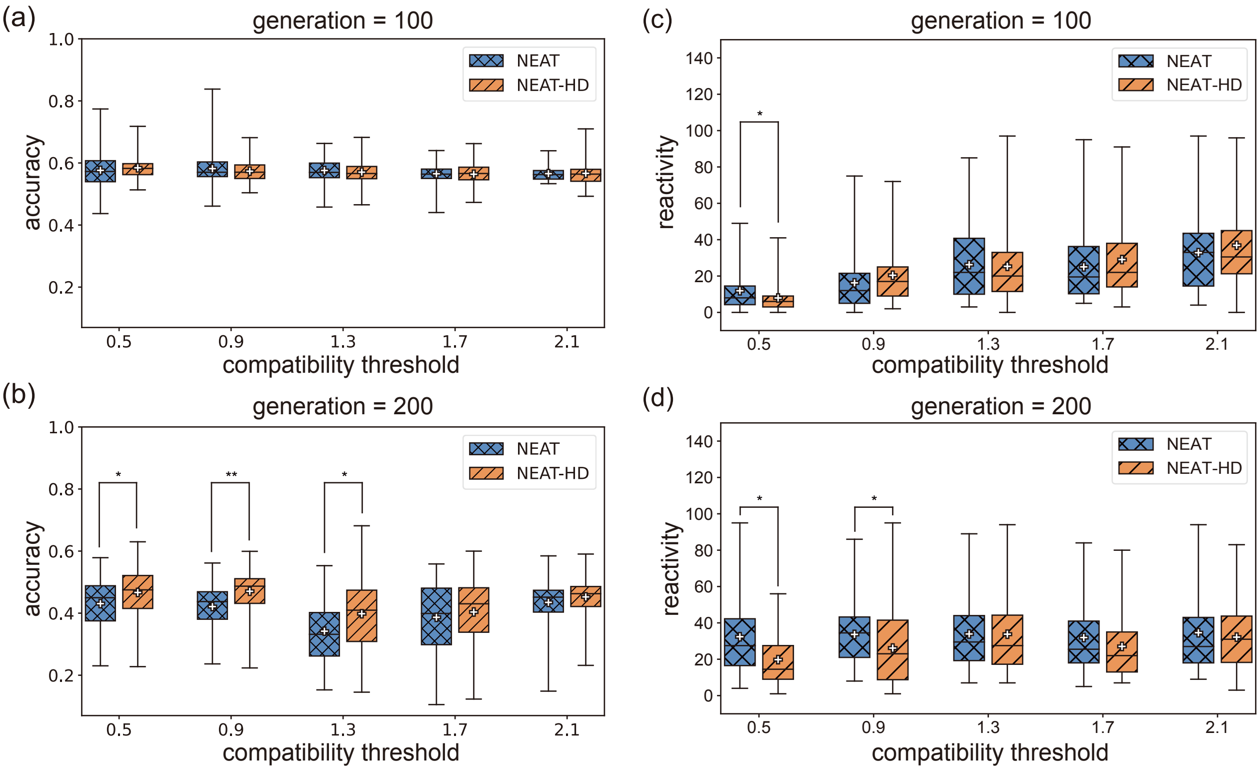

1 1→4
2 2→4 Disabled
3 3→4
4 2→5
5 5→4
6 1→5
4
5
1
2
3
Mutate Add Connection
1 1→4
2 2→4 Disabled
3 3→4
4 2→5
5 5→4
6 1→5
7 3→5
4
5
1
2
3
1 1→4
2 2→4 Disabled
3 3→4
4 2→5
5 5→4
6 1→5
4
5
1
2
3
Mutate Add Node
1 1→4
2 2→4 Disabled
3 3→4 Disabled
4 2→5
5 5→4
6 1→5
8 3→6
9 6→4
4
5
6
1
2
3

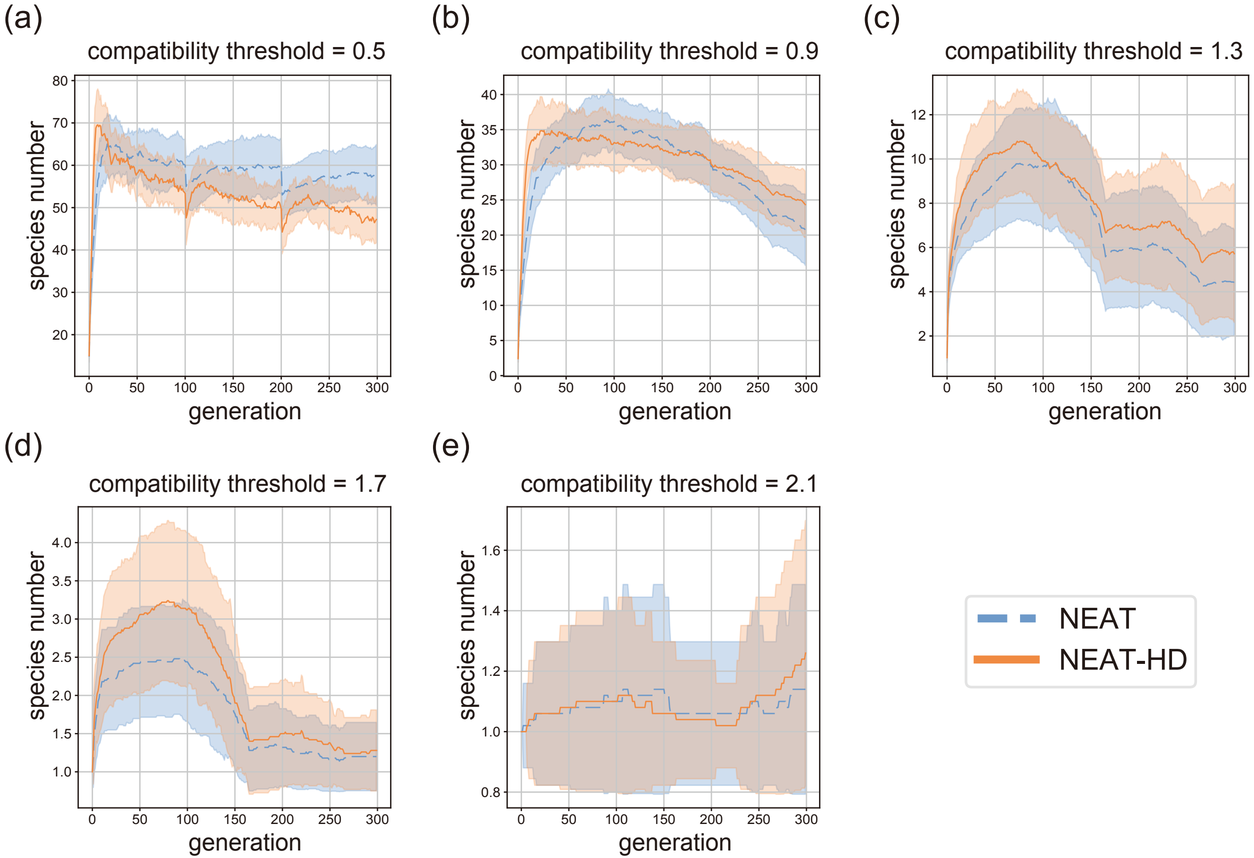

(a)
species A
species B
(b)
probability of being a parent
parent A
candidates of parent B

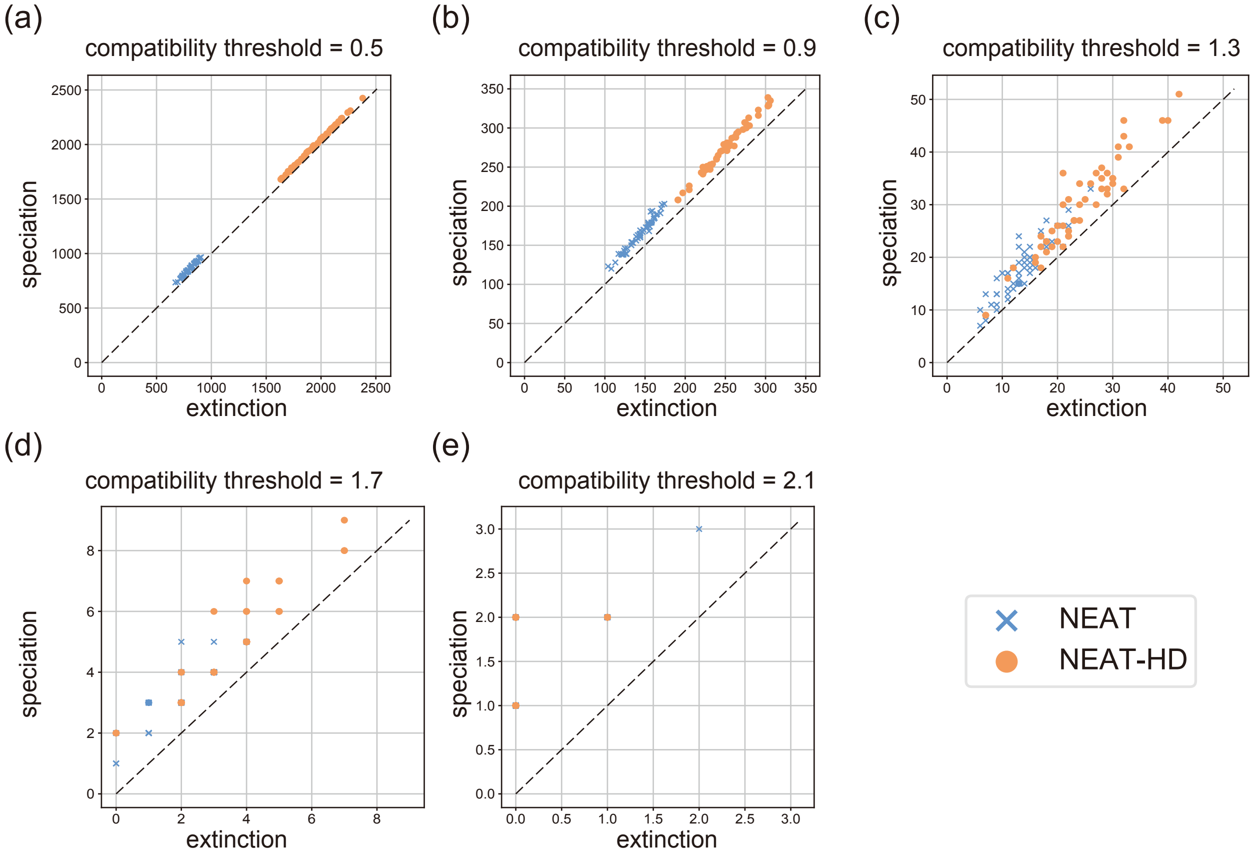

(a)
θ
-2.4
2.4
x
optimal point
(x=0)
(b)
optimal point
1.5
0
-1.5
100
200
300
generation

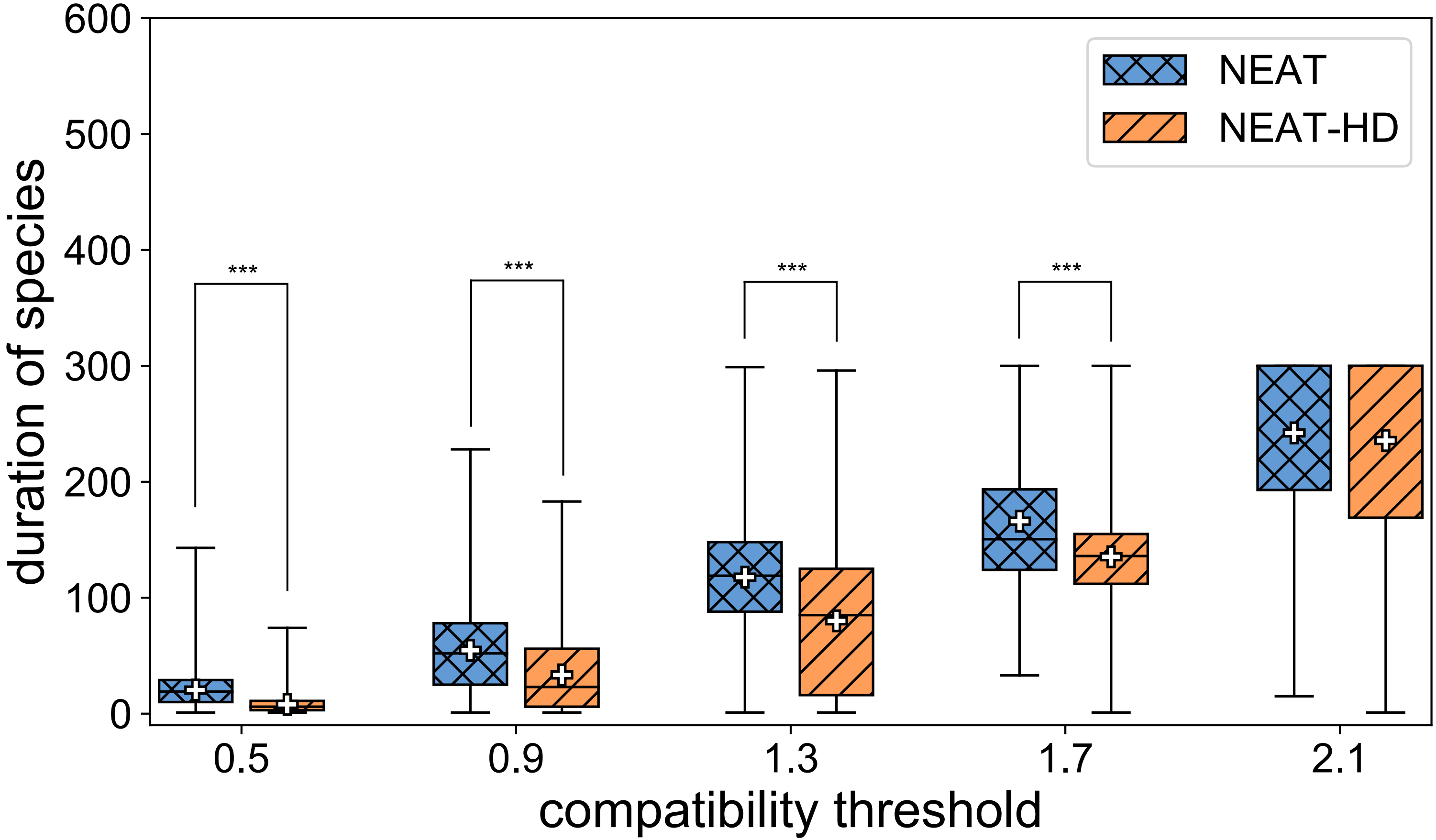

(a)
compatibility threshold = 0.5
number of distinct parents
0
10
20
30
40
50
60
70
80
generation
0
50
100
150
200
250
300
(b)
compatibility threshold = 0.9
number of distinct parents
0
20
40
60
80
100
generation
0
50
100
150
200
250
300
(c)
compatibility threshold = 1.3
number of distinct parents
0
10
20
30
40
50
generation
0
50
100
150
200
250
300
(d)
compatibility threshold = 1.7
number of distinct parents
0
10
20
30
40
generation
0
50
100
150
200
250
300
(e)
compatibility threshold = 2.1
number of distinct parents
−5
0
5
10
15
20
generation
0
50
100
150
200
250
300
NEAT
NEAT-HD

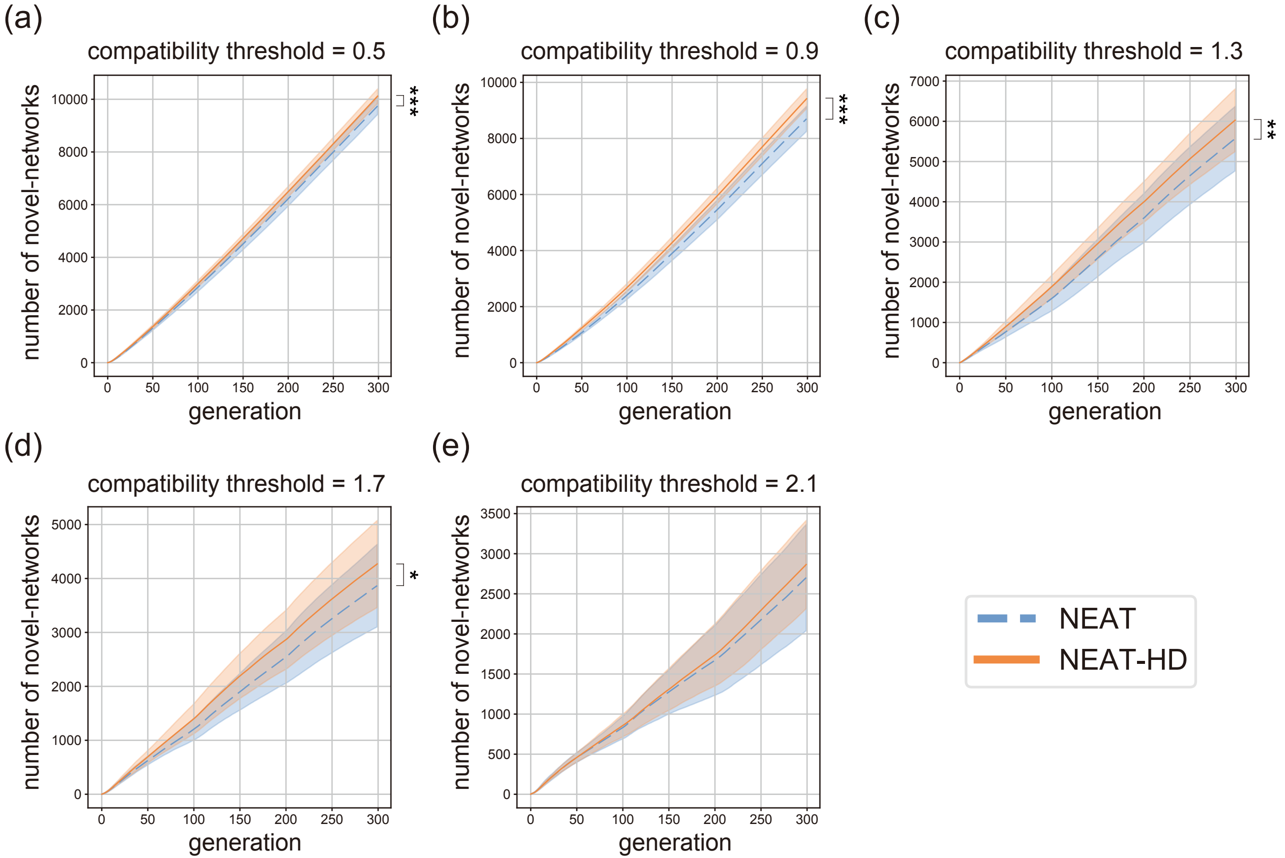

(a)
compatibility threshold = 0.5
(b)
compatibility threshold = 0.9
(c)
compatibility threshold = 1.3
(d)
compatibility threshold = 1.7
(e)
compatibility threshold = 2.1
best-fitness
1000
900
800
700
600
500
400
300
200
100
generation
0
50
100
150
200
250
300
NEAT
NEAT-HD

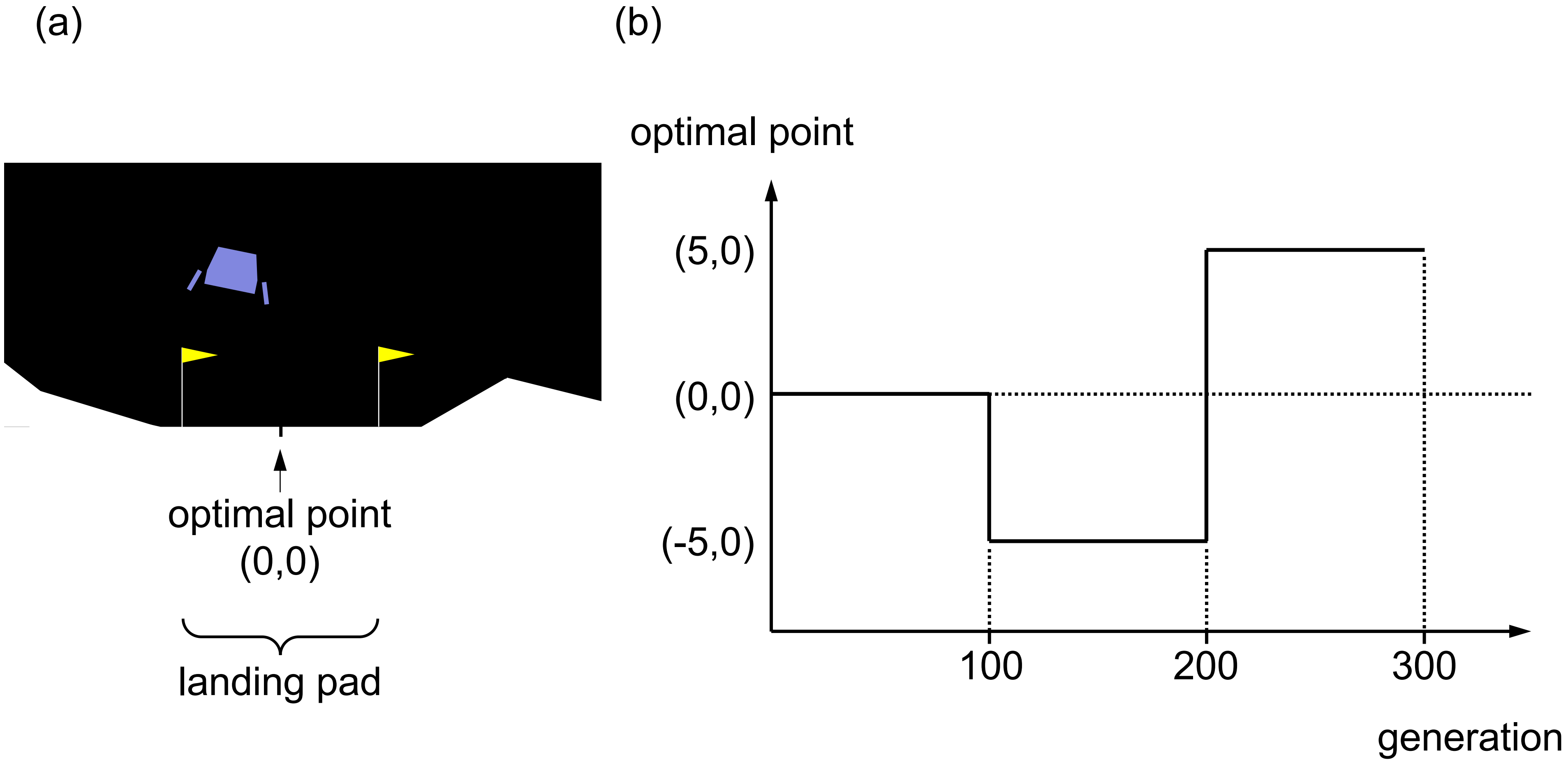

(a)
generation = 100
accuracy
1.0
0.8
0.6
0.4
0.2
0.5
0.9
1.3
1.7
2.1
compatibility threshold
NEAT
NEAT-HD
(b)
generation = 200
accuracy
1.0
0.8
0.6
0.4
0.2
*
**
*
0.5
0.9
1.3
1.7
2.1
compatibility threshold
NEAT
NEAT-HD
(c)
generation = 100
reactivity
140
120
100
80
60
40
20
0
*
0.5
0.9
1.3
1.7
2.1
compatibility threshold
NEAT
NEAT-HD
(d)
generation = 200
reactivity
140
120
100
80
60
40
20
0
*
*
0.5
0.9
1.3
1.7
2.1
compatibility threshold
NEAT
NEAT-HD

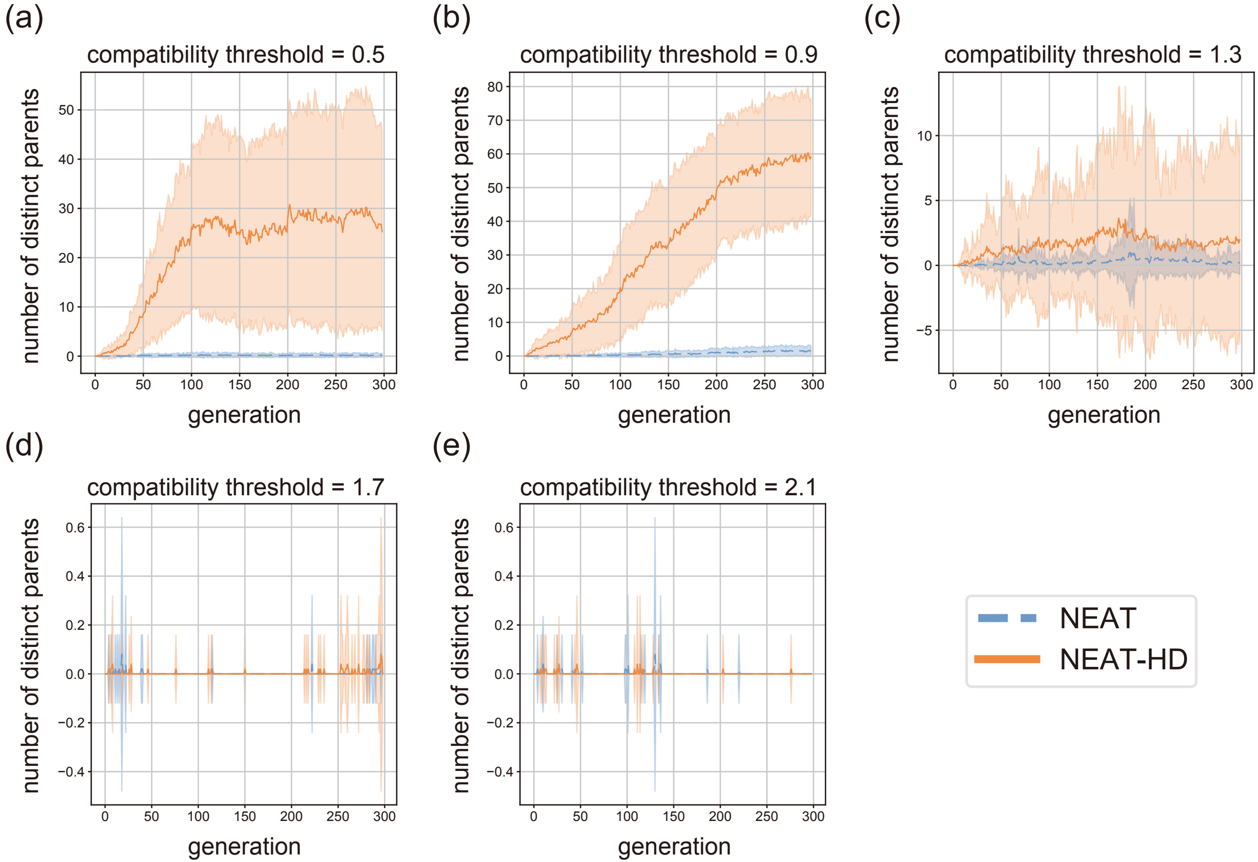

(a)
compatibility threshold = 0.5
species number
20
30
40
50
60
70
80
0
50
100
150
200
250
300
generation
(b)
compatibility threshold = 0.9
species number
0
5
10
15
20
25
30
35
40
0
50
100
150
200
250
300
generation
(c)
compatibility threshold = 1.3
species number
2
4
6
8
10
12
0
50
100
150
200
250
300
generation
(d)
compatibility threshold = 1.7
species number
1.0
1.5
2.0
2.5
3.0
3.5
4.0
0
50
100
150
200
250
300
generation
(e)
compatibility threshold = 2.1
species number
0.8
1.0
1.2
1.4
1.6
0
50
100
150
200
250
300
generation
NEAT
NEAT-HD

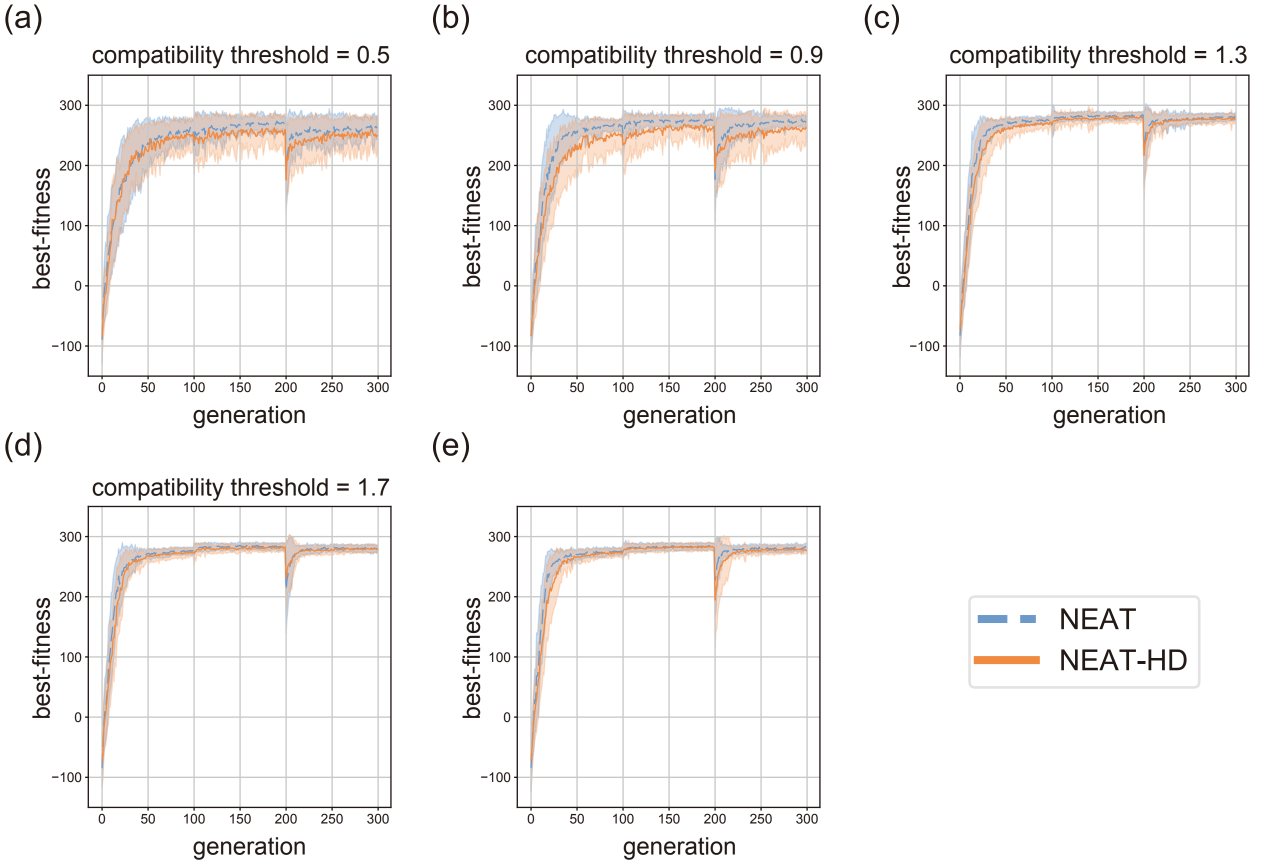

(a)
compatibility threshold = 0.5
speciation
0
500
1000
1500
2000
2500
extinction
0
500
1000
1500
2000
2500
(b)
compatibility threshold = 0.9
speciation
0
50
100
150
200
250
300
350
extinction
0
50
100
150
200
250
300
350
(c)
compatibility threshold = 1.3
speciation
0
10
20
30
40
50
extinction
0
10
20
30
40
50
(d)
compatibility threshold = 1.7
speciation
0
2
4
6
8
extinction
0
2
4
6
8
(e)
compatibility threshold = 2.1
speciation
0.0
0.5
1.0
1.5
2.0
2.5
3.0
extinction
0.0
0.5
1.0
1.5
2.0
2.5
3.0
× NEAT
● NEAT-HD

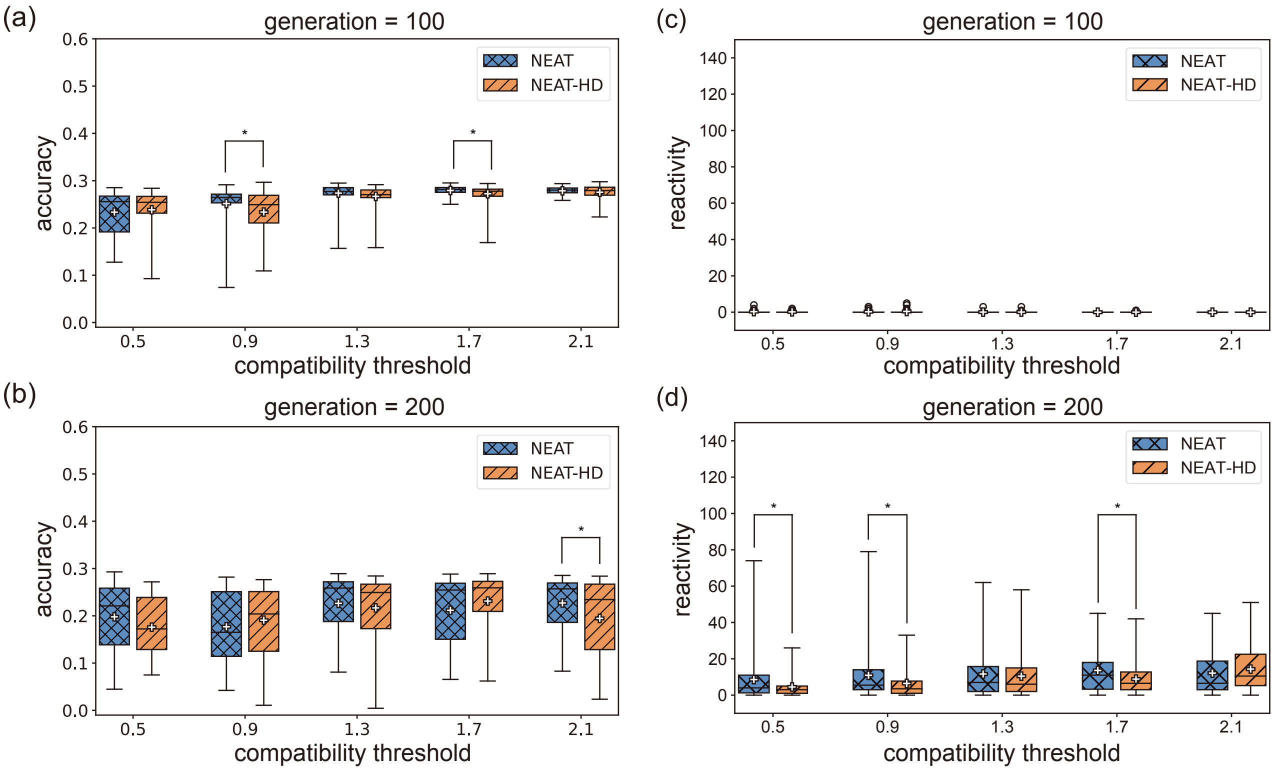

600
500
400
300
200
100
0
duration of species
0.5
0.9
1.3
1.7
2.1
compatibility threshold
NEAT
NEAT-HD
***
***
***
***

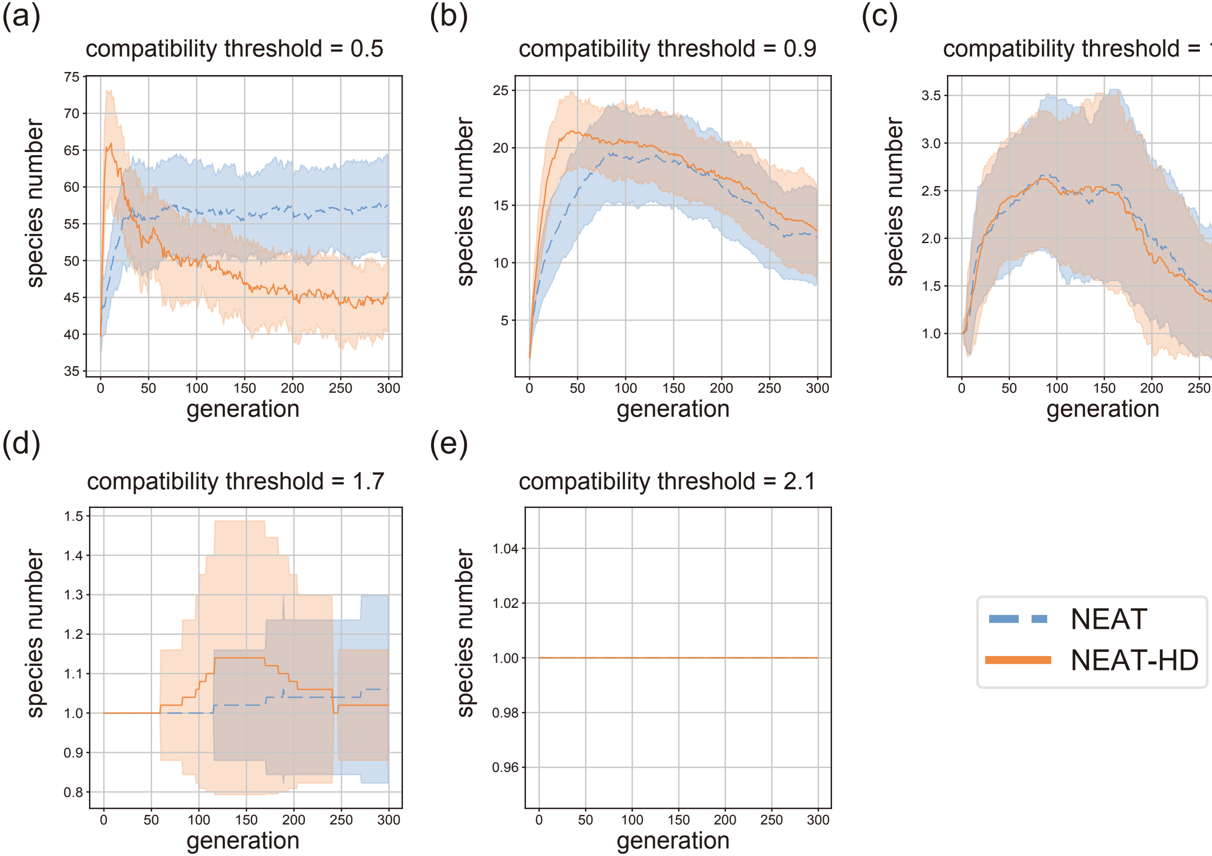

(a)
compatibility threshold = 0.5
number of novel-networks
0
2000
4000
6000
8000
10000
0
50
100
150
200
250
300
generation
***
(b)
compatibility threshold = 0.9
number of novel-networks
0
2000
4000
6000
8000
10000
0
50
100
150
200
250
300
generation
***
(c)
compatibility threshold = 1.3
number of novel-networks
0
1000
2000
3000
4000
5000
6000
7000
0
50
100
150
200
250
300
generation
**
(d)
compatibility threshold = 1.7
number of novel-networks
0
1000
2000
3000
4000
5000
0
50
100
150
200
250
300
generation
*
(e)
compatibility threshold = 2.1
number of novel-networks
0
500
1000
1500
2000
2500
3000
3500
0
50
100
150
200
250
300
generation
NEAT
NEAT-HD

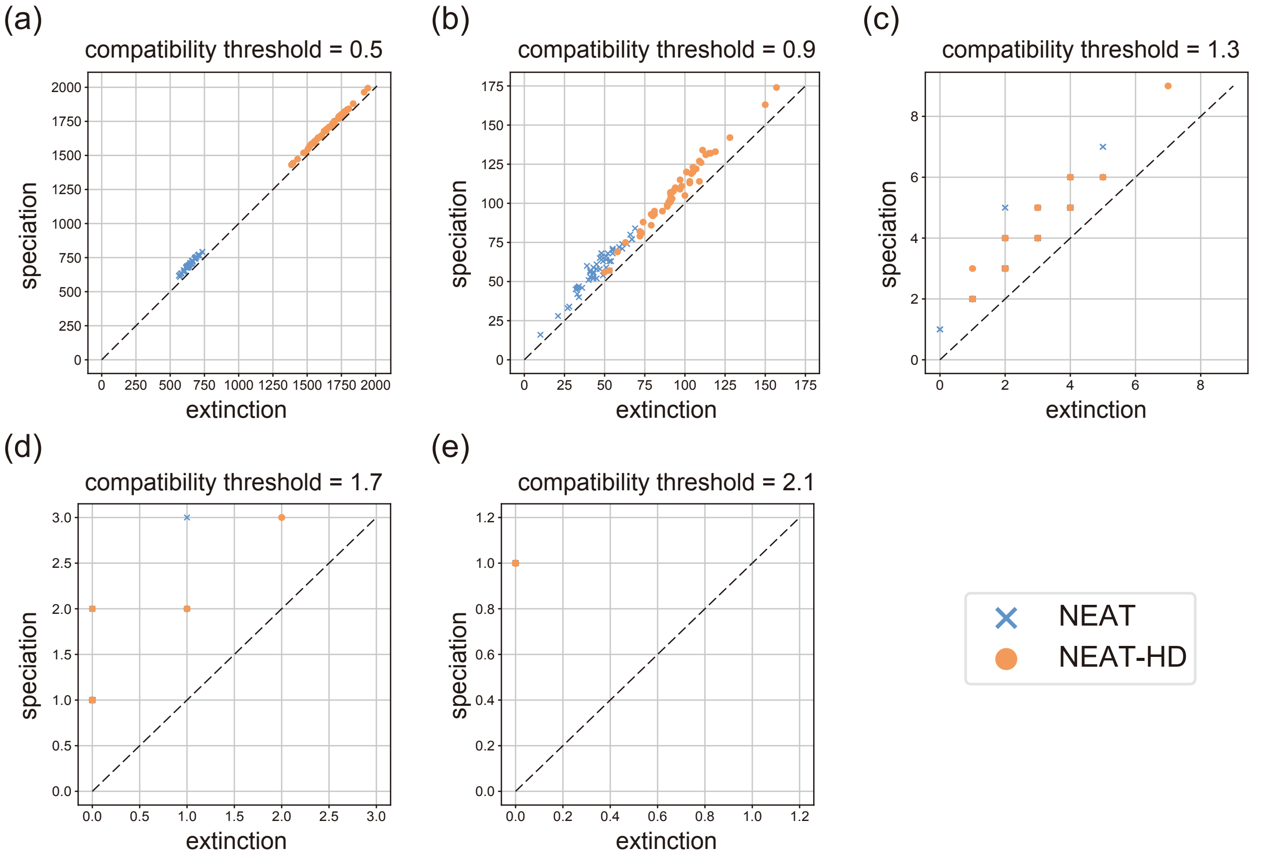

(a)
optimal point
(0,0)
landing pad
(b)
optimal point
(5,0)
(0,0)
(-5,0)
100
200
300
generation

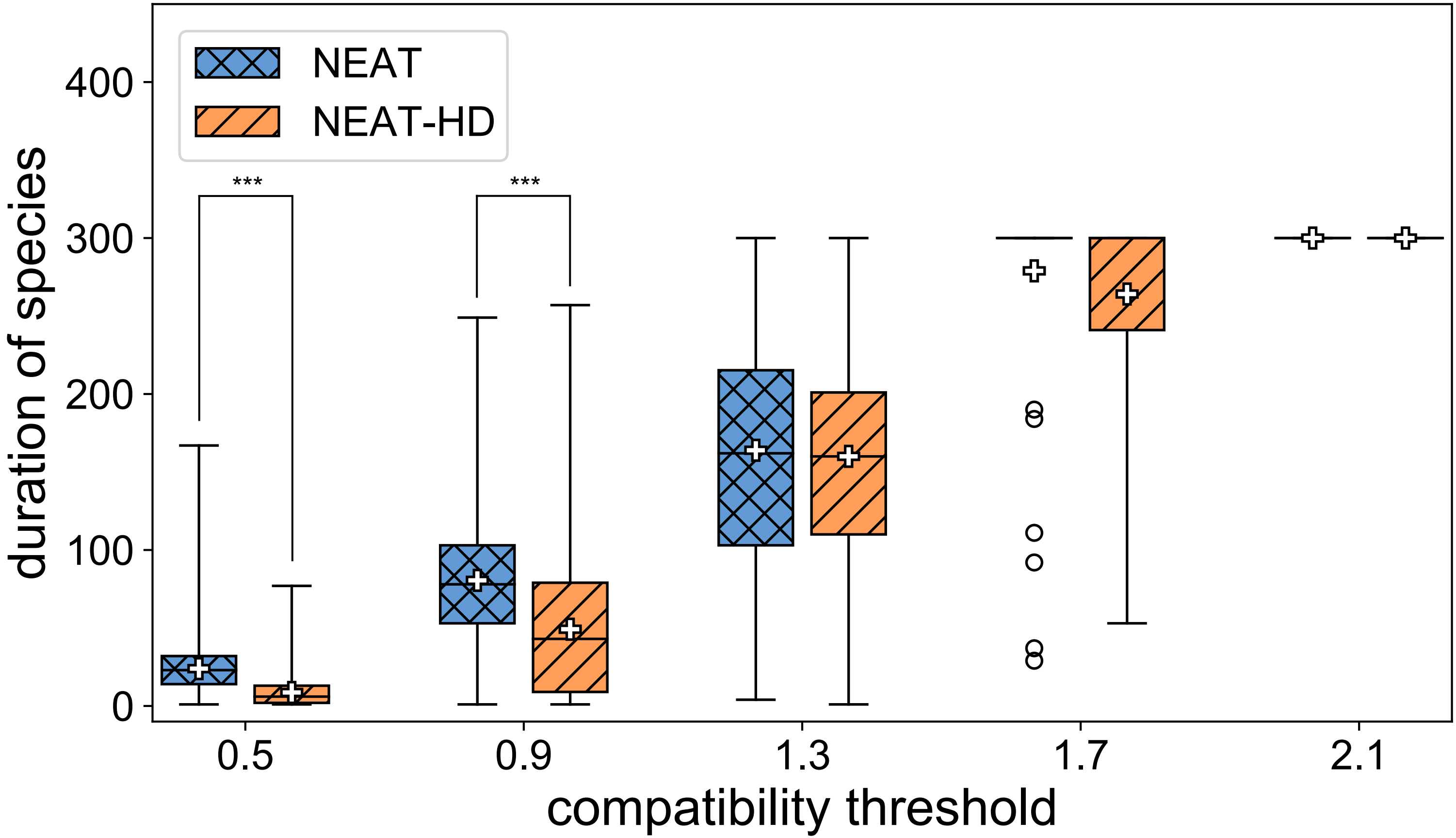

(a)
compatibility threshold = 0.5
number of distinct parents
0
10
20
30
40
50
generation
0
50
100
150
200
250
300
(b)
compatibility threshold = 0.9
number of distinct parents
0
10
20
30
40
50
60
70
80
generation
0
50
100
150
200
250
300
(c)
compatibility threshold = 1.3
number of distinct parents
−5
0
5
10
generation
0
50
100
150
200
250
300
(d)
compatibility threshold = 1.7
number of distinct parents
−0.4
−0.2
0.0
0.2
0.4
0.6
generation
0
50
100
150
200
250
300
(e)
compatibility threshold = 2.1
number of distinct parents
−0.4
−0.2
0.0
0.2
0.4
0.6
generation
0
50
100
150
200
250
300
NEAT
NEAT-HD


(a)
compatibility threshold = 0.5
(b)
compatibility threshold = 0.9
(c)
compatibility threshold = 1.3
(d)
compatibility threshold = 1.7
(e)
best-fitness
generation
300
200
100
0
−100
0
50
100
150
200
250
300
NEAT
NEAT-HD


(a)
generation = 100
accuracy
0.0
0.1
0.2
0.3
0.4
0.5
0.6
NEAT
NEAT-HD
*
*
0.5
0.9
1.3
1.7
2.1
compatibility threshold
(b)
generation = 200
accuracy
0.0
0.1
0.2
0.3
0.4
0.5
0.6
NEAT
NEAT-HD
*
0.5
0.9
1.3
1.7
2.1
compatibility threshold
(c)
generation = 100
reactivity
0
20
40
60
80
100
120
140
NEAT
NEAT-HD
0.5
0.9
1.3
1.7
2.1
compatibility threshold
(d)
generation = 200
reactivity
0
20
40
60
80
100
120
140
NEAT
NEAT-HD
*
*
*
0.5
0.9
1.3
1.7
2.1
compatibility threshold


(a)
compatibility threshold = 0.5
species number
generation
75
70
65
60
55
50
45
40
35
0
50
100
150
200
250
300
(b)
compatibility threshold = 0.9
species number
generation
25
20
15
10
5
0
50
100
150
200
250
300
(c)
compatibility threshold = 1.3
species number
generation
3.5
3.0
2.5
2.0
1.5
1.0
0
50
100
150
200
250
300
(d)
compatibility threshold = 1.7
species number
generation
1.5
1.4
1.3
1.2
1.1
1.0
0.9
0.8
0
50
100
150
200
250
300
(e)
compatibility threshold = 2.1
species number
generation
1.04
1.02
1.00
0.98
0.96
0
50
100
150
200
250
300
NEAT
NEAT-HD


(a)
compatibility threshold = 0.5
speciation
extinction
0
250
500
750
1000
1250
1500
1750
2000
(b)
compatibility threshold = 0.9
speciation
extinction
0
25
50
75
100
125
150
175
(c)
compatibility threshold = 1.3
speciation
extinction
0
2
4
6
8
(d)
compatibility threshold = 1.7
speciation
extinction
0.0
0.5
1.0
1.5
2.0
2.5
3.0
(e)
compatibility threshold = 2.1
speciation
extinction
0.0
0.2
0.4
0.6
0.8
1.0
1.2
NEAT
NEAT-HD


NEAT
NEAT-HD
***
***
duration of species
0
100
200
300
400
0.5
0.9
1.3
1.7
2.1
compatibility threshold

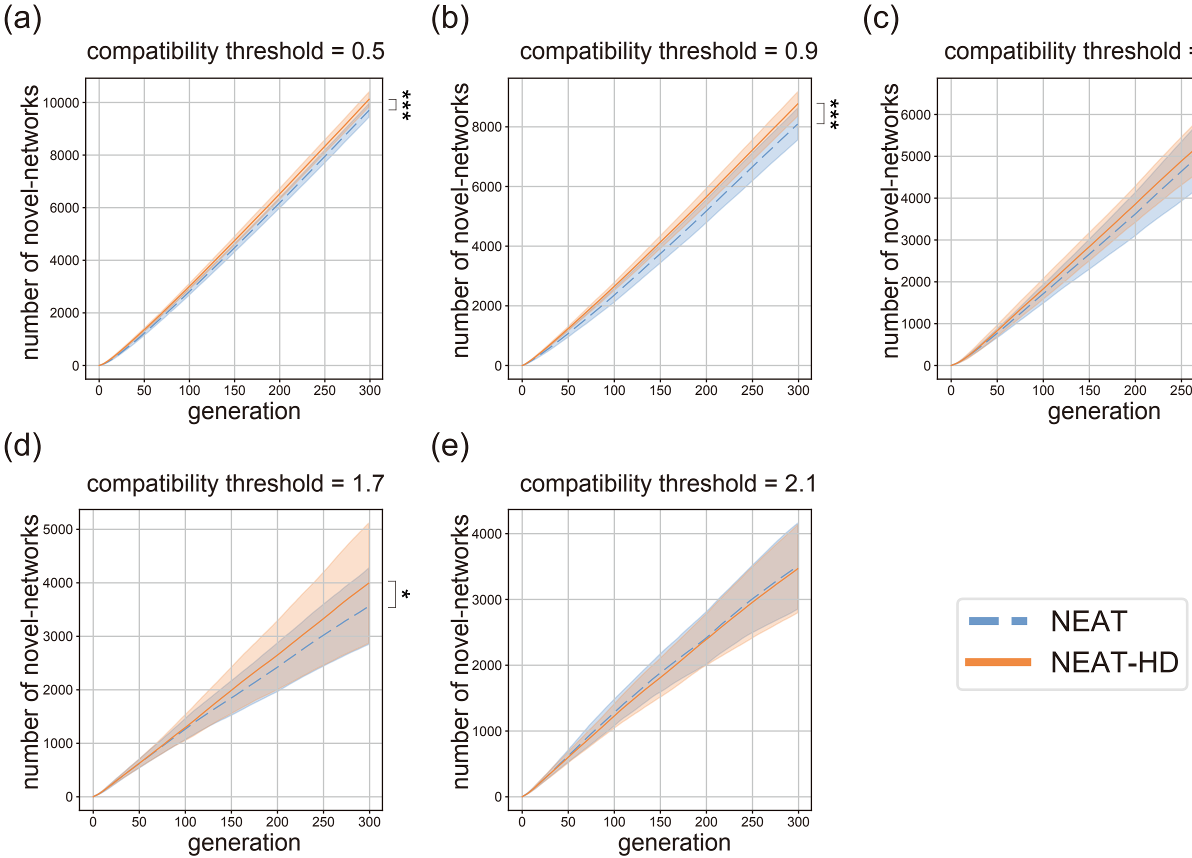

(a)
compatibility threshold = 0.5
number of novel-networks
0
2000
4000
6000
8000
10000
generation
0
50
100
150
200
250
300
***
(b)
compatibility threshold = 0.9
number of novel-networks
0
2000
4000
6000
8000
generation
0
50
100
150
200
250
300
***
(c)
compatibility threshold = 1.3
number of novel-networks
0
1000
2000
3000
4000
5000
6000
generation
0
50
100
150
200
250
300
(d)
compatibility threshold = 1.7
number of novel-networks
0
1000
2000
3000
4000
5000
generation
0
50
100
150
200
250
300
*
(e)
compatibility threshold = 2.1
number of novel-networks
0
1000
2000
3000
4000
generation
0
50
100
150
200
250
300
NEAT
NEAT-HD

Table.1

The parameters used in this study.

Parameters	Value
generation	300
population size	128
generations to trigger stagnation	64
number of repetition when evaluating individuals	2
type of speciation (“neat” or “none”)	“neat”
percentage chance of applying second objective when using Multi Objective Optimization	0.0
activation function for hidden nodes	5
probability of adding connection	0.15
probability of adding node	0.1
probability of crossover	0.8
probability to enable disabled connection	0.01
probability of changing node activation function	0.0
probability of mutating connection weight	0.8
probability of enabling each initial connection	1.0
precent of individuals to remove from parent pool before selection	0.1
precent of individuals to pass on to next generation unchanged	0.1
how to scale individual rank (“exponential” or “linear”)	“exp”
number of competitors in each tournament	2
species threshold	1.3
weighting of non-matching genes in compatibility distance calculation	1
weighting of weight differences in compatibility distance calculation	0.5

Table.2

The stabilities of both algorithms (NEAT and NEAT-HD) at each compatibility threshold in cart-pole swing-up task.

compatibility threshold	0.5		0.9		1.3		1.7		2.1	
type	NEAT	NEAT-HD	NEAT	NEAT-HD	NEAT	NEAT-HD	NEAT	NEAT-HD	NEAT	NEAT-HD
generation=100	1.0	1.0	0.94	1.0	0.88	0.94	0.92	0.90	0.86	0.84
generation=200	0.96	1.0	0.96	0.96	0.80	0.84	0.92	0.98	0.86	0.84

Table.3

The stabilities of both algorithms (NEAT and NEAT-HD) at each compatibility threshold in lunar lander task.

compatibility threshold	0.5		0.9		1.3		1.7		2.1	
type	NEAT	NEAT-HD	NEAT	NEAT-HD	NEAT	NEAT-HD	NEAT	NEAT-HD	NEAT	NEAT-HD
generation=100	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
generation=200	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0