



Title	Complexity of bird song caused by adversarial imitation learning
Author(s)	Yamazaki, Seiya; Iizuka, Hiroyuki; Yamamoto, Masahito
Citation	Artificial life and robotics, 25(1), 124-132 https://doi.org/10.1007/s10015-019-00559-5
Issue Date	2020-02
Doc URL	http://hdl.handle.net/2115/80329
Rights	This is a post-peer-review, pre-copyedit version of an article published in Artificial Life and Robotics. The final authenticated version is available online at: http://dx.doi.org/10.1007/s10015-019-00559-5 .
Type	article (author version)
File Information	AROB_journal_final_yamazaki.pdf



[Instructions for use](#)

Complexity of Bird Song Caused by Adversarial Imitation Learning

Seiya Yamazaki · Hiroyuki Iizuka · Masahito Yamamoto

Abstract Biological evolution produces complexity through genetic variations based on randomness. In conventional communication or language simulation models, genetic variations based on randomness and fitness function rewarding task achievements play an important role in evolving communication signals to complex ones. However, it is known that not only genetic variations evolve communication but also imitative learning during developmental processes contribute to the evolution of communication. What we investigated here was to find a different principle of generating complexity which does not rely on the randomness or external environmental complexity but only on the learning processes in communication. Our hypothesis is that the contradictory learning mechanism we call the adversarial imitation learning can work to increase the complexity without relying on the random processes. To investigate our hypothesis, we implemented the adversarial imitation learning on a simulation where two agents interact and imitate with each other. Our results showed that the adversarial imitation learning causes chaotic dynamics and by investigating the learning results in different types of interaction between the two, it was clarified that the adversarial imitation learning is necessary for the emergence of the chaotic time series.

Keywords Adversarial imitation learning, Complexity, Chaos

S. Yamazaki · H. Iizuka · M. Yamamoto
Graduate School of Information Science and Technology,
Hokkaido University, Kita 14, Nishi 9, Kita-ku, Sapporo,
Hokkaido, 060-0814, Japan
E-mail: yamazaki@complex.ist.hokudai.ac.jp
H. Iizuka
E-mail: iizuka@complex.ist.hokudai.ac.jp
M. Yamamoto
E-mail: masahito@complex.ist.hokudai.ac.jp

1 Introduction

Evolution produces complexities through genetic variations based on randomness. The produced variations are selected by screening them to find more suitable variations for the environment. The selected variations produce the new variations in the next generations, which are then selected again. The successive processes of the selections and reproductions with variations generate sophisticated and complex, but not random, structures and behaviors of individuals that survive. There is no explicit fixed fitness function as used in the genetic algorithm [1], but only survivability works as natural selections. The survivability produces the selection pressure towards not only individual units but also groups of individuals as a unit within or among species [2, 3]. The mechanism and open-endedness of evolution produces complexity everywhere [4, 5]. This kind of complexity always depends on the genetic variations brought by randomness intrinsically.

On the other hand, the complexity of bird songs or language can also evolve through communication with others in a community by teaching or imitations [6–8]. The chirping of birds is neither simple nor random but has grammatical structures and sounds like songs. Ten Cate and Okanoya show that the grammars of complex bird songs can be represented as finite-state automaton [9]. It is also shown that birds that generate more complex songs have the advantage for territorial claims [10] and courtship of females [9]. Those birds change their own song by imitating each other.

There are roughly two kinds of communication or language models in terms of different causes that evolve the communication signals to a complex one. One is that the models have rich environments and the agents that handle communication signals need to describe a variety of objects or situations to achieve given tasks. For example, the naming game model by Steels investigates how the vocabularies converge as shared common names for the multiple objects [11]. Marocco and Nolfi show that communication signals

can be self-organized and used to express their situations to achieve a task [12]. In these models, complexity is given explicitly as the external stimuli and the communication signals evolve as complexity of the external stimuli.

Another is the communication-based models where there are no explicit external stimuli but the ways of interaction make communication complex. Sasahara and Ikegami show that the bird song evolves to complex ones and the grammatical structure emerges by modeling birds' interjection communication [13]. In this study, the ethological aspect that female birds select male birds' songs according to their favorite is modeled as interjection communication. Depending on the mating scores, the grammatical structure evolves with genetic operations. Hashimoto and Ikegami show that the grammar of the agents evolved in complexity by modeling the dilemma in which the agent accepts the others' signal while generating its own original signal not to be accepted [14]. Suzuki and Kaneko showed that the bird song time series becomes more complex by modeling the imitating and not-to-be-imitated processes using the logistic map and evolutionary techniques [15]. Evolution of a bird song is directed to the edge of chaos in their model.

In all of these simulation models, the generators of communication signals evolve with random-based genetic operators and fitness evaluations calculated on the basis of task achievements or communication. As the long history of evolution has shown, accumulated randomness makes the world more complex, but even only the learning process works to make the world complex. What we investigated here is to find a different principle for generating complexity which does not rely on the randomness or external environmental complexity, but only on the learning processes in communication. Our hypothesis is that the contradictory learning mechanism used in the imitation game model by Suzuki and Kaneko can work to increase the complexity without relying on the random processes. In order to investigate this, we constructed a simulation model where two agents interact with each other and modify their behaviors through the learning principles that are on the loss of objective functions, which models the contradictory learning mechanism which we call the adversarial imitation learning. As a result, we show that the adversarial imitation learning leads the communication signals to the complex one, which is chaotic.

2 Simulation model

2.1 Adversarial imitation learning

Adversarial imitation learning involves agents imitating others' songs while their songs should not be imitated by the others. To imitate the others' song, the generation models of the songs need to be similar, but if these models are too similar, it implies that the others can imitate as well. We

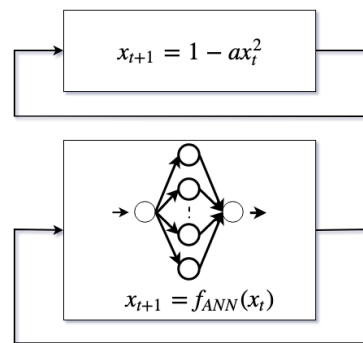


Fig. 1 Top: Generation model of the logistic map used in the conventional model. Bottom: Generation model with an artificial neural network in our model.

postulate that this kind of contradictory learning pressure produces complex songs.

Our simulation model consists of a time series generation phase as its own original bird songs and imitation phase to imitate the other bird songs [16]. The time series generation is performed by a feedforward neural network. The artificial neural network is updated based on the bias of the adversarial imitation learning. Our model is explained using a case that involves two agents interacting with and imitating each other, nonetheless, it can be easily extended to a scenario with a greater number of agents.

2.2 Generation of their own bird songs

Suzuki and Kaneko used the logistic map [15]. The bird song time series was generated by the following equation,

$$x_{t+1} = 1 - ax_t^2, \quad (1)$$

where x_t is a single phoneme at time t and a is a nonlinearity parameter. The generators evolve by changing parameter a with the evolutionary technique. The song dynamics from a simple to a complex one can be represented by the parameter but the dynamical system depends on the logistic map.

In order for generators to represent any functions, we use a feedforward neural network instead of the logistic map as shown in Fig. 1. The network consists of three layers, which are the input, hidden, and output layers. There are no recurrent connections. The output of the neural network is calculated as follows:

$$x_{t+1} = f_{ANN}(x_t), \quad (2)$$

$$= g\left(\sum_{i=1}^N w_i(g(v_i x_t + b_i^1)) + b^2\right), \quad (3)$$

where w_i and v_i are weight, b_i^1 and b^2 are bias, N indicates the number of hidden neurons, and $g(x)$ is the activation function. The tanh function is used here.

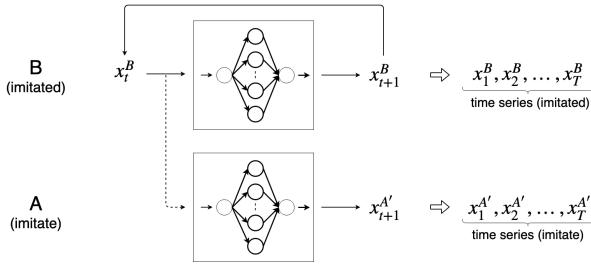


Fig. 2 Agent B generates its own time series while listening to its own song, but agent A generates its own time series while listening to B's songs.

2.3 Imitation of the other songs

When a bird agent imitates the other bird's songs, it generates the bird song time series while listening to the others' songs. Fig. 2 represents the overview of this imitation process.

Let us explain the imitation phase in the case where agent A imitates B. Agent B generates its own bird song time series $(x_1^B, x_2^B, \dots, x_T^B)$ with its own neural network f_{ANN}^B while listening to its own songs as follows,

$$x_{t+1}^B = f_{ANN}^B(x_t^B). \quad (4)$$

Agent A generates the bird song time series $(x_1^{A'}, x_2^{A'}, \dots, x_T^{A'})$ with its own neural network f_{ANN}^A while listening to the songs of agent B, that is, receiving the time series of agent B as follows:

$$x_{t+1}^{A'} = f_{ANN}^A(x_t^B). \quad (5)$$

On the other hand, when agent B imitates A, the roles are exchanged. B generates the bird song time series while listening to the songs generated by A. These interactions are expressed as follows:

$$x_{t+1}^A = f_{ANN}^A(x_t^A), \quad (6)$$

$$x_{t+1}^{B'} = f_{ANN}^B(x_t^A). \quad (7)$$

Each bird agent has a single initial value to start their own and imitate songs. As well as the weights and the biases of the network, the initial value is also updated by learning. The initial value x_0 is obtained through the \tanh function to restrict it in $[-1, 1]$ as follows:

$$x_0 = \tanh(x_0'). \quad (8)$$

The same initial value is used to generate their songs for both the generation and imitation phases.

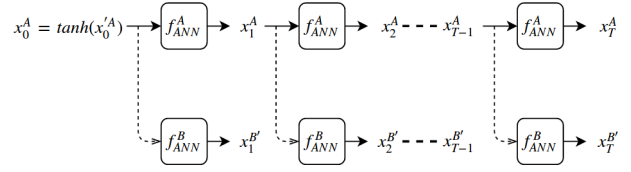


Fig. 3 Learning of the initial values of agents' bird songs using back-propagation through time (BPTT). The errors are propagated to the initial values along this flow.

2.4 Objective function

The time series generated in the generation and imitation phases are evaluated in terms of the adversarial imitation learning in order to train the agent's network. Two types of errors are calculated. One is the imitating-loss and another is the not-imitated-loss. The imitating-loss is calculated to evaluate how well the agent could have imitated the others' songs. The not-imitated-loss evaluates how much the generated songs could have not been imitated by the others.

The imitating-loss of A indicates how close the time series generated in the imitation phase of A is to B's original song, and is calculated as follows:

$$E_{imitating}^A = \frac{1}{T} \sum_{t=1}^T (x_t^{A'} - x_t^B)^2. \quad (9)$$

The not-imitated-loss of A indicates how far the original song of A is from the time series generated in the imitation phase of B, and is calculated as follows:

$$E_{not-imitated}^A = \frac{1}{T} \sum_{t=1}^T (2 - |x_t^A - x_t^{B'}|)^2. \quad (10)$$

The imitating-loss and the not-imitated-loss of B are also calculated in the same manner, as shown in eq (11) and (12).

$$E_{imitating}^B = \frac{1}{T} \sum_{t=1}^T (x_t^{B'} - x_t^A)^2, \quad (11)$$

$$E_{not-imitated}^B = \frac{1}{T} \sum_{t=1}^T (2 - |x_t^B - x_t^{A'}|)^2. \quad (12)$$

The not-imitated-loss is also used to learn the initial value x_0' . The errors are back-propagated to the initial value by regarding the generating process of songs as the recurrent neural network like $x_t = f_{ANN}(f_{ANN}(\dots(f_{ANN}(\tanh(x_0')))))$, as shown in Fig 3. Therefore, the initial value can be updated by BPTT [17] such that each value of the two-time series are more distant.

Those errors are integrated into the total errors of the agent, which can be expressed as follows:

$$E^A = E_{imitating}^A + E_{not-imitated}^A, \quad (13)$$

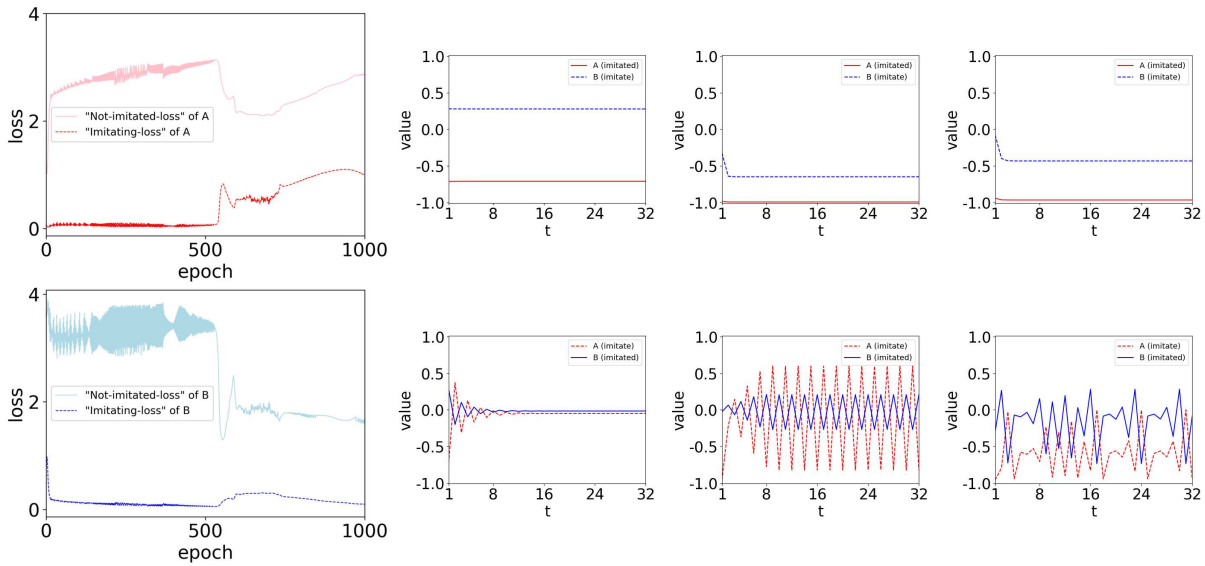


Fig. 4 Leftmost column: Changes of imitating-loss and not-imitated-loss. The remaining graphs show examples of bird song time series obtained at 1, 550 and 700 epoch from the left. The upper row is the case where A is imitated by B (B imitates A) and lower is *vice versa*.

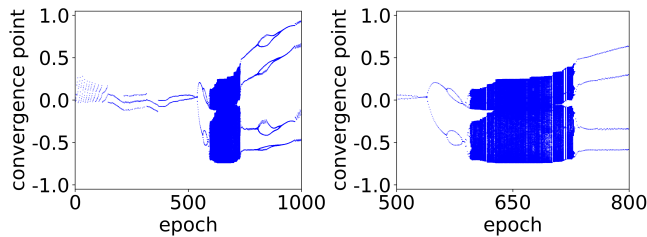


Fig. 5 The bifurcation diagrams of agent B. The right graph shows an enlarged view of the left.

$$E^B = E_{imitating}^B + E_{not-imitated}^B. \quad (14)$$

The weights, biases, and initial values of both agents are updated based on these errors, using back propagation. The parameters of the network are updated by the stochastic gradient descent algorithm. In order to avoid the exploding gradient problem, the gradient clipping method is used [18]. The artificial neural networks of A and B are updated at the same time after the generation and imitation phases.

3 Experiments

3.1 Experimental setting

We performed the experiment in the following experimental setup. The length of the generated time series T is 32, and the number of learning epochs is 1,000. The number of nodes N in the hidden layer of the neural network is 32.

3.2 Adversarial imitation learning between two agents

The simulation results of both agents performing the adversarial imitation learning are explained. The losses over epochs and the examples of the generated time series are shown in Fig. 4. The imitating-loss is smaller than the not-imitated-loss in both two agents. At the beginning of the learning, both agents generate a simple time series which fall into a fixed-point. At 550 epoch, period-2 time series are generated by agent B and A's imitating time series becomes period-2, too. On the other hand, the original time series of A remains a fixed-point. At 700 epoch, B's original time series shows more complex dynamics like chaos.

To analyze the changes of the song dynamics through the adversarial imitation learning, we make the bifurcation diagram as Fig. 5. The vertical axis shows the convergence points of the time series that is generated when agent B generates its own song by itself until 2,000 time steps. The time series until 200 time steps are removed and the remains are plotted. If it is converged to a single point, the convergence points are plotted on a single point at the corresponding epoch. The graph shows when and how the dynamics changes. The results show that B's song is very simple at the beginning, but the number of periodicity increase. Then, the periodicity disappears and it becomes a chaotic-like dynamics. The adversarial imitation learning appears to transform the songs into chaos.

3.3 Evaluation by Lyapunov exponent

From the bifurcation diagram shown in Fig. 5, we can see that the adversarial imitation learning makes the songs com-

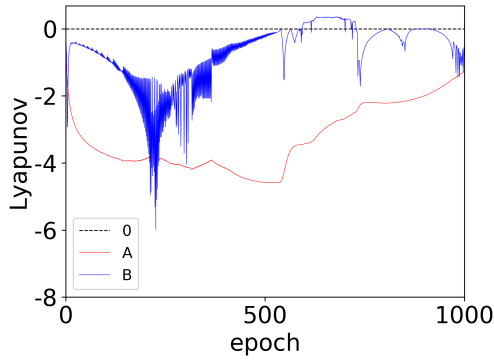


Fig. 6 Transitions of the Lyapunov exponents of two agents.

plex. In order to evaluate the obtained bird songs objectively, we introduce the Lyapunov exponent λ [19].

λ is the degree to which a slight difference in the dynamic system is enlarged exponentially and is used to tell whether it is chaotic. It is defined as follows:

$$\lambda = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=0}^{n-1} \ln |f'(x_i)|, \quad (15)$$

where $f(x)$ is the dynamical system, and x_i is the value of the time series. When $\lambda > 0$, it is a chaotic time series, but not a chaotic time series otherwise. In this study, n is 1,000, and because our song model is expressed as eq (3), we calculate the Lyapunov exponent λ with $f(x)$ as the artificial neural network that each agent has.

The changes of the Lyapunov exponents are shown in Fig. 6. In the beginning, the Lyapunov exponent becomes negative, which means that the dynamics is stable. The time series becomes periodic as shown in Fig. 5. After that, the stable dynamics is destabilized gradually and the Lyapunov exponent becomes zero or greater than zero, which means that the dynamics is chaotic. This chaos is sustained for a while. After that, the value stays around $\lambda = 0$. The reason why the chaotic dynamics come back to the simple dynamics is not clear, but we think that this is because there is no pressure in keeping the chaotic dynamics in our current model. Even when agent B's own song is chaotic, B must imitate A's songs that remain a simple time series. The contradiction of imitating and not-being-imitated can also work for simplicity after the chaos emerges. Therefore, B's songs become periodic and the Lyapunov exponent stays around $\lambda = 0$. This might be the edge of chaos. It is observed that the chaotic and simple dynamics appear alternately when we perform the simulations for longer periods.

The analysis of the Lyapunov exponents clearly shows that the time series, after training with the adversarial imitation learning, become chaotic.

3.4 Different types of interactions

In the experiment of the adversarial imitation learning between two agents, the generated time series gradually becomes complex and the chaos dynamics emerge. To investigate the conditions for emerging the chaotic time series, various types of interactions between the two agents were tested.

The ratios of emerging chaotic time series in agent B's song during learning, under different types of interactions, are shown in Fig. 7. Once the Lyapunov exponent of agent B's network, calculated at each epoch, becomes positive until 2,000 epochs, it is counted as the emergence of the chaotic time series and the ratio of over 100 trials is shown. In the figure, the arrow from agent A to B shows that agent A imitates agent B's song, which means that $E_{imitating}$ is included in the objective functions. If the arrow is blocked with the line at the destination, the destination agent is trained not to be imitated by the origin agent. It means that $E_{not-imitated}$ is included in the objective function. The dotted arrow with the blocked line shows that the origin agent is not trained to imitate the destination agent's song but the destination agent is trained not to be imitated by the origin agent. In other words, the objective function of the origin does not have $E_{imitating}$ and the destination includes $E_{not-imitated}$ against the time series generated by the origin. The adversarial imitation learning of the agent can be shown with the blocked line for solid or dotted arrow from the opponent and the solid arrow to the opponent. There are three patterns of the agents, i.e., trained with only imitating, only not to be imitated, or both (the adversarial imitation learning). In order to investigate whether the mutual learning interactions were required for the emergence of chaotic time series, the two additional agents were prepared. One was the fixed agent whose network parameters were never updated since the initial values were set at the first epoch. Another was the random agent whose network parameters were randomly set at the beginning of each epoch.

From the results, the chaotic time series does not appear at all when agent B has been trained with only imitating or only imitated, regardless of the types of agent A. Only when agent B performs the adversarial imitation learning, is the chaotic time series observed. The adversarial imitation learning of agent B is required to generate the chaotic time series in agent B. However, when agent A has been trained with only imitating or when agent A is the fixed or random network, the chaotic time series does not appear, or appears very rarely. It means that the emergence of the chaotic time series in agent B requires agent A's training not to be imitated.

The ratio becomes the highest when agent A and B perform not-imitated and adversarial imitation learning, respectively. Under this interaction, the time series of agent A never

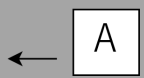
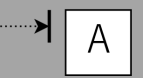
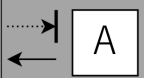


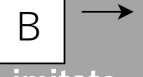
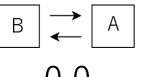
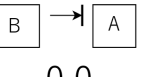
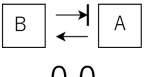
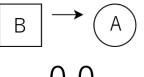
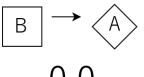
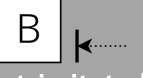
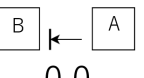
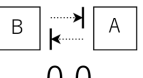
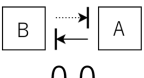


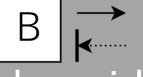
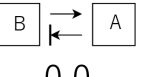
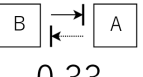
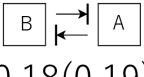

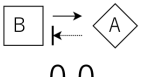
	 imitate	 not-imitated	 adversarial	 fix	 random
 imitate	 0.0	 0.0	 0.0	 0.0	 0.0
 not-imitated	 0.0	 0.0	 0.0	 0.0	 0.0
 adversarial	 0.0	 0.33	 0.18(0.19)	 0.04	 0.0

Fig. 7 Ratios that agent B generates chaotic time series in the various types of interaction. The number between brackets indicates the ratio of A. This is calculated over 100 trials. See the main text for the learning types.

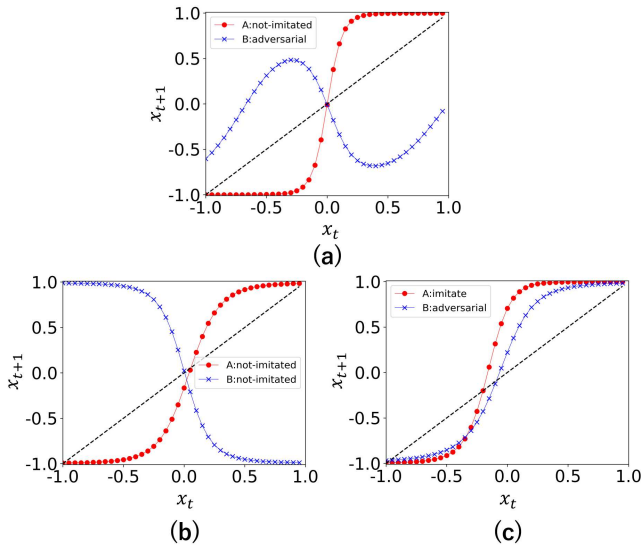


Fig. 8 Return maps obtained from the artificial neural networks of the agents which are trained in different interactions, i.e., (a): not-imitated and adversarial, (b): not-imitated and not-imitated, (c): imitating and adversarial. The artificial neural networks after trained for 2,000 epochs are analyzed.

becomes chaotic (it is shown in the opposite interaction, i.e., A: adversarial imitation learning and B: not-imitated learning). When both agents perform the adversarial imitation learning, the time series becomes chaotic with the lower ratio than the previous one. However, this is because there are possibilities of agents at both sides generating chaotic time series when both agents perform the adversarial imitation learning and the chaotic time series at one side, inhibits the agent at another side, becoming chaotic. In fact, there are no cases where both agents become chaotic in the experiments. From these results, we conclude that the adversarial imita-

tion learning causes an increase of complexity of the time series and makes it chaotic.

In order to analyze how the network has changed to chaotic by the adversarial imitation learning, we visualize the shapes of functions of two agents by the return maps. The example of simple and chaotic functions obtained in the different interactions are shown in Fig. 8. These return maps are made by giving inputs between -1 and 1 as x_t to the network and obtaining outputs x_{t+1} corresponding to inputs. Fig. 8 (a) shows the return maps obtained after 2,000 epoch training when agent A and B are trained with not-imitated and the adversarial imitation learning, respectively. (b) is for both trained with not-imitated learning and (c) for imitating and the adversarial imitation learning.

In (b), both agents try to sing completely different songs (not imitated with each other), the functions become opposite and converge. In (c), agent A tries only to imitate agent B without performing not-imitated learning. The return map of agent A becomes similar to agent B's, although agent B wants to stay away from agent A when it is imitated. (a) is the case where the chaotic time series are generated in agent B, most often. The return map must stay away from agent A but somehow needs to approach to it. Such contradiction bends the return map, which changes to the shape to generate the chaotic time series in the end.

3.5 The balance of the learning pressures

We show that adversarial imitation learning encourages complication of the bird song time series, and reveal the condition of the emerging chaotic time series. However, it is not clear how much the imitating and not-imitated learning contribute to the emergence of the chaotic time series and if there is the best-weighted combination of imitating and not-

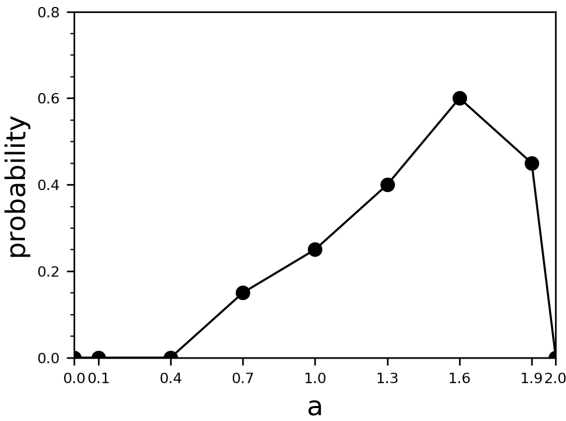


Fig. 9 Ratios of emerging chaotic time series in agent B's song depending on the weight parameter a of imitating and not-imitated learning.

imitated learning for the chaotic time series. In order to clarify this, we performed simulations with the objective functions with different a as follows,

$$E = aE_{\text{imitating}} + (2.0 - a)E_{\text{not-imitated}}, \quad (16)$$

where a is the learning weight parameter that decides the balance between imitating and not-imitating learning, and the bigger this parameter is, the more agent imitates the opponents' songs. This type of interaction is fixed to the case where agent A and B are trained with only not-imitated and adversarial imitation learning, respectively, because only agent B has a chance to generate the chaotic time series and agent A never, and we can focus on agent B's songs. For each parameter a , we performed the experiments 20 times and counted the number of cases where the Lyapunov exponent for agent B's network became positive.

The results are shown in Fig. 9. If the not-imitated learning is too strong, the chaotic time series does not emerge. When agent B only imitates the opponent ($a = 2.0$), the chaotic time series is not produced, either. However, if not-imitated learning is introduced even slightly into imitating learning, the chaotic time series can emerge. The ratio of emerging chaos varies greatly depending on the weights of imitating and not-imitated learning and becomes 0.6 at the best weights. Our results show that the adversarial imitation learning works for the emergence of the chaotic time series and small amounts of not-imitated learning lead to chaos.

4 Discussion

This paper suggests that the complication of time series is proceeded by the adversarial imitation learning, which is modeled as the learning that an agent must imitate others' songs while the agent must generate its own song that is not imitated by the others using a neural network. The generated

time series is a fixed-point at the beginning of the learning. However, gradually the time series become periodic, finally, the chaotic time series is generated.

Several studies have reported that the dilemma gives rise to complexity in the simulation. It is the case that the dilemma by the adversarial imitation learning make the dynamics generated by the neural network more complex. The dilemma is realized by two contradictory loss functions, which are the imitating-loss and the not-imitated-loss. This is actually similar to GAN (Generative Adversarial Network) which is a popular method to generate complex and realistic pictures with a deep neural network [20]. As with a GAN, the competing loss functions might help the network to develop complex structures.

Moran and Pollack present the coevolutionary dynamics of complexity growth in a variety of multi-species simulations modeled with finite state automata [21]. In their research, the complexity grows up the most in the contradictory situation where both cooperation and competition are required at the same time. As well as their results, in our simulation, the complication only progresses when the adversarial imitation learning is performed. The difference from these models is that they use finite-state automata as a basic structure, which is modulated by the random processes as genetic mutation. The automata have a discretized structure and it is rather easy to introduce the complex structure because it is modularized. In our model, we used an artificial neural network as a generator function, which only produced very simple fixed-point dynamics at the beginning. What we show is that even without the discretized or modularized structures nor abrupt changes given by random mutation, complication comes to be embedded into the network by the adversarial imitation learning.

Unlike the previous simulation model [15], in this paper, the complication and chaos emerged by only the contradictory learning processes. This means that the complication can be produced in a closed system without randomness and any external events as long as there is the only a contradictory situation.

In our current model, a feedforward neural network is used for generating time series, which means that the neural network cannot produce the time-dependent dynamics. If the adversarial imitation learning enhances the complication of the time series, generating chaotic dynamics is the only way to produce complex time series. What we want to show here is not that the bird songs are chaotic, but that the adversarial imitation learning enhances the complication of the time series without external randomness. If a recurrent neural network that can hold memories is used, the network would be able to produce complex time series with syntactic structures as the bird songs can be represented as finite-state automaton [9].

From the ethological point of view, the interaction like adversarial imitation learning has not been reported yet but it is known that birds generating complex songs are stronger for the defense of their territory [10] and those songs are obtained by imitation [9]. These findings would imply that the individuals that can imitate the others and are not imitated by the others would be able to obtain the highest benefits for survivability. This could be applied to not only the complication of bird songs but also behaviors or strategies to get foods or even body structures that can produce more complex behaviors to imitate the others and not to be imitated by the others. Our model might be able to explain wider phenomena of complication.

5 Conclusion

Our study showed that the adversarial imitation learning causes the chaotic dynamics and clarified that it is a necessary condition for the emergence of the chaotic time series by investigating the different types of interaction between the two. Because the feedforward neural network is used in our model, the complication is restricted into a spatial dimension of the bird songs. If the model is extended to the recurrent neural network, the complication can develop in a spatio-temporal dimensions. The grammatical structure of the bird songs as Okanoya shows in Bengalese finch's songs [9] could be seen in the temporal complexity in such an extended model.

Acknowledgements This work was partially supported by JSPS KAKENHI Grant Number JP18H05057.

References

1. John Holland. Adaptation in natural and artificial systems: an introductory analysis with application to biology. *Control and artificial intelligence*, 1975.
2. Charles Darwin and Alfred Wallace. On the tendency of species to form varieties; and on the perpetuation of varieties and species by natural means of selection. *Journal of the proceedings of the Linnean Society of London. Zoology*, Vol. 3, No. 9, pp. 45–62, 1858.
3. David Sloan Wilson. The group selection controversy: history and current status. *Annual review of ecology and systematics*, Vol. 14, No. 1, pp. 159–187, 1983.
4. Tom Froese, Nathaniel Virgo, and Takashi Ikegami. Life as a process of open-ended becoming: Analysis of a minimal model. In *ECAL*, pp. 250–257, 2011.
5. Thomas Miconi. Evosphere: evolutionary dynamics in a population of fighting virtual creatures. In *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pp. 3066–3073. IEEE, 2008.
6. Ofer Tchernichovski and Gary Marcus. Vocal learning beyond imitation: mechanisms of adaptive vocal development in songbirds and human infants. *Current opinion in neurobiology*, Vol. 28, pp. 42–47, 2014.
7. Ofer Tchernichovski and Fernando Nottebohm. Social inhibition of song imitation among sibling male zebra finches. *Proceedings of the National Academy of Sciences*, Vol. 95, No. 15, pp. 8951–8956, 1998.
8. Sébastien Derégnaucourt and Manfred Gahr. Horizontal transmission of the father's song in the zebra finch (*taeniopygia guttata*). *Biology letters*, Vol. 9, No. 4, p. 20130247, 2013.
9. Carel ten Cate and Kazuo Okanoya. Revisiting the syntactic abilities of non-human animals: natural vocalizations and artificial grammar learning. *Philosophical Transactions of the Royal Society B: Biological Sciences*, Vol. 367, No. 1598, pp. 1984–1994, 2012.
10. Clive K Catchpole. Vocal communication in birds. *Japanese Journal of Ornithology*, Vol. 30, No. 2-3, pp. 87–89, 1981.
11. Luc Steels and Paul Vogt. Grounding adaptive language games in robotic agents. In *Proceedings of the fourth european conference on artificial life*, Vol. 97, 1997.
12. Davide Marocco and Stefano Nolfi. Emergence of communication in embodied agents evolved for the ability to solve a collective navigation problem. *Connection Science*, Vol. 19, No. 1, pp. 53–74, 2007.
13. Kazutoshi Sasahara and Takashi Ikegami. Evolution of birdsong syntax by interjection communication. *Artificial Life*, Vol. 13, No. 3, pp. 259–277, 2007.
14. Takashi Hashimoto and Takashi Ikegami. Emergence of net-grammar in communicating agents. *BioSystems*, Vol. 38, No. 1, pp. 1–14, 1996.
15. Junji Suzuki and Kunihiko Kaneko. Imitation games. *Physica D: Nonlinear Phenomena*, Vol. 75, No. 1-3, pp. 328–342, 1994.
16. Seiya Yamazaki, Hiroyuki Iizuka, and Masahito Yamamoto. Emergence of chaotic time series by adversarial imitation learning. In *Artificial Life Conference Proceedings*, pp. 659–664. MIT Press, 2018.
17. AJ Robinson and Frank Fallside. *The utility driven dynamic error propagation network*. University of Cambridge Department of Engineering, 1987.
18. Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pp. 1310–1318, 2013.
19. Aleksandr Mikhailovich Lyapunov. The general problem of the stability of motion. *International journal of control*, Vol. 55, No. 3, pp. 531–534, 1992.
20. Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.
21. Nick Moran and JB Pollack. Effects of cooperative and competitive coevolution on complexity in a linguistic prediction game. In *Proceedings of the 14th European conference on artificial life*, pp. 298–205, 2017.