# Characterizing individual behaviors by using Recurrent Neural Networks

Timothe Anne[1]
Supervisors: Anja Philippsen[2] and Yukie Nagai[2]

[1] École Normale Suprieure de Rennes (ENS Rennes)
Bruz, France
timothee.anne@ens-rennes.fr
[2] Center for Information and Neural Networks (CiNet),
National Institute of Information and Communication Technology (NICT)
Osaka, Japan
{anja,yukie}@nict.go.jp

**Abstract.** Individuals with autism display a wide range of atypical behaviors, but little is know about the internal mechanisms causing them. We used a recurrent neural network model in drawing tasks to study the internal mechanisms that can cause different behaviors. We first studied the impact of the learning order on the internal representation. Secondly, we suggest that people with autism have problems with prediction. In a previous study by Saito et al., chimpanzees, who lack prediction, and children presented different behaviors. We investigated an internal mechanism of the network allowing it to predict and to replicate some of those different behaviors.

**Keywords:** autism · recurrent neural networks · internal representation · predictive coding.

**Date**: From 14 May 2018 to 10 August 2018

**Location**: Center for Information and Neural Networks (CiNet),
National Institute of Information and Communication Technology (NICT)
Osaka, Japan

## 1 Introduction

Autism spectrum disorder (ASD) is a developmental disorder which touches more and more children. In the United-States the prevalence of ASD among 8 years-old children went from 1 in 150 children in 2000 to 1 in 59 children in 2014 [2]. It is characterized by difficulties in social interaction and atypical perception, but little is known about the internal mechanisms causing it.

Pellicano et al. [3] recently suggested that such mechanisms could be the inability of the brain to imitate or, more generally, to predict. For a typical developed person, their brain integrates top-down knowledge (such as prior knowledge, expectations or contextual information) with bottom-up sensory perceptions (such as what we see, hear or feel). Lawson et al. [4] suggested that an imbalance between these two processing pathways could be a potential cause for ASD. People with ASD might not use prediction and pay much more attention to the raw sensory stimuli, causing phenomena such as hypersensitivity to changes in brightness or environmental noise [3].

Directly studying the internal mechanisms of the brain is difficult, it is easier to use computational models which mimic them. By replicating some behaviors and understanding the internal mechanisms of the model which caused them, we can discover hints to some equivalent mechanisms which might exists in the brain. Computational models such as recurrent neural network have been used by previous studies to demonstrate how certain network parameters can be changed to replicate typical behaviors observed in individuals with autism or schizophrenia [5], [6].

In this paper we present two separate studies relying on the same neural network model, the Stochastic Continuous Time Recurrent Neural Network (SCTRNN). In the first study we investigated how the learning order of different trajectories influences the speed of the learning but also the quality of the internal representation. We suggest that a fast and successful learning does not always lead to the best internal representation.

The second study consisted of using our model to replicate several behaviors linked to the use of top-down predication. Saito et al. [1] showed that chimpanzees and typical developed children present different behaviors when facing to a drawing task. The children old enough would complete missing parts whereas chimpanzees never displayed this behavior. We suggest that chimpanzees are an extreme case of lack of top-down prediction. By being able to replicate those behaviors, we showed that our model is able to replicate a wide range of behaviors and can be reused in future ASD's studies.

In Section 2 we present the related work and our motivation to realize these two studies. In Section 3 we present our model. In Section 4 we present the learning order influence and in Section 5 we present our completion task.

## 2 Related work and Motivation

### 2.1 Computational modeling of individual behavior

Recurrent neural network models have a good capacity for learning to realize complex tasks and use mechanisms inspired by the way our brain learn. This is why they have been employed as computational models for replicating the behavior of people with developmental disorders.

Murata et al. [7] proposed a novel type of recurrent neural network model (SCTRNN) which can not only learn the mean but also the variance of time series. They showed that a humanoid robot using the proposed network can learn to reproduce latent stochastic structures hidden in fluctuating training trajectories. We choose this model because of this capacity to learn the variance.

Using this model, Idei et al. [6] could replicate repetitive behavior patterns of ASD subjects in a recurrent neural network model by adjusting the sensitivity of the network to the estimated variance of the external signal. As suggested by Lawson et al. [4], such an altered confidence could increase or decrease the prediction error and impair predictive learning.

Yamashita and Tani [5] employed a different recurrent neural network with two layers working on different timescales. They demonstrated that disconnectivity between these layers cause symptoms in the model which can be related to symptoms of schizophrenia.

Those studies demonstrate that this network model can replicate individual differences in terms of behavior. In our two studies, we investigated the influence that some parameters can have on the model's behaviors.

### 2.2 The first study: Learning Order

The idea of studying the learning order came from an old idea presented by Vygotskii [8]: the Zone of Proximal Development. This suggests that to improve the learning of a child it is better to sort tasks in increasing complexity in order to have each task slightly more complex than the previous one but still feasible with help of a teacher. We wanted to see if such a methodology has an impact in recurrent neural network training. From an engineer's point of view, it can be better to give all the training data from the start than to give the training data in increasing complexity phases. We investigated whether this is the case by looking at the prediction error and also at the quality of the internal representation.

### 2.3 The second study: Completion Task

Saito et al. [1] examined the evolutional origin of representational drawing. They performed an experiment consisting in directly comparing the drawing behavior of human children and chimpanzees. During free drawing on incomplete facial stimuli, they revealed the remarkable difference between the two

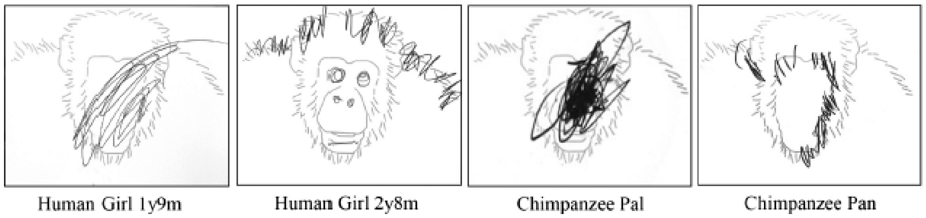



Fig. 1: Some examples of children and chimpanzees behaviors taken from Saito et al. [1].

species. Old enough humans ($N = 57$, 6-38 months) tend to complete the missing parts even with immature motor control, whereas chimpanzees never completed the missing parts and instead marked the existing parts or traced the outlines, some examples of behaviors are presented in Figure 1.

Numerous researchers depicted some similarity in terms of behavior between chimpanzees and people with ASD (even though you cannot find it clearly written in a paper). Both present difficulties to use top-down prediction, people with ASD behaviors being between the two extremes behaviors: those of chimpanzees (totally lack of prediction) and typical developed people (typical use of prediction).

For our second experiment we wanted to use our model to replicate some of the behaviors presented in their study. To investigate the use of top-down prediction as internal mechanism we used a method of prediction presented by Murata et al. [9]. They showed that the trained SCTRNN can recognize trajectories by inferring the initial states that can reproduce the trajectory through the same maximum likelihood estimation scheme as that used for network training. We demonstrate that by recognizing an incomplete trajectory, the model can successfully predict what should be the complete trajectory to draw, and draw it as do typical developed people.

## 3 The SCTRNN model

Stochastic Continuous Time Recurrent Neural Network (SCTRNN) is a recurrent neural network model that can learn time series by learning to map an input signal to an output signal. The *Stochastic* adjective refer to its capacity to not only estimate the output signal's mean but also to estimate the output signal's variance. In this paper we are using this network to draw different trajectories in a 2D-environment.

The weight update via backpropagation is not only based on the prediction error, but is also inversely weighted by the predicted variance. In case of high variance estimation, the influence of the prediction error is reduced, which induce a more stable learning in case of noisy training signal. It was introduced by Murata et al. [7].

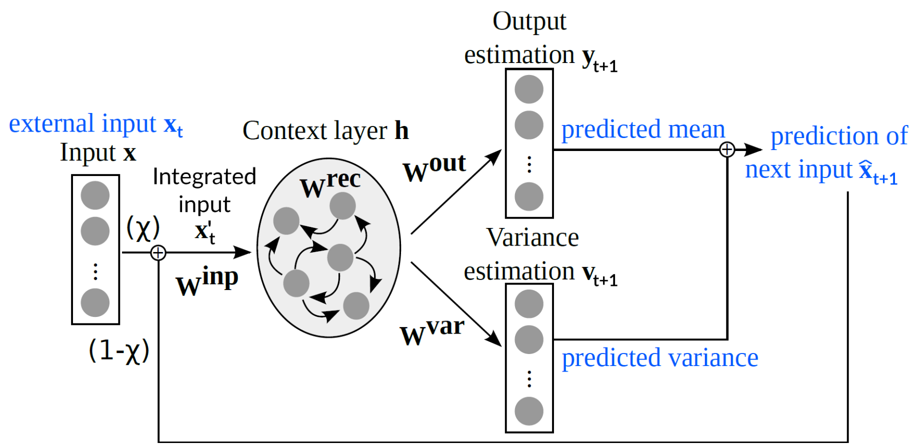



Fig. 2: Schematic view of a SCTRNN: A context layer of neurons processes the external input and predicts the next time step input signal's mean and variance.

### 3.1 Network Overview

A schematic view of the network is presented in Figure 2. The network is trained to predict time series $x = x_0, x_1, \ldots, x_t, \ldots, x_T$ by estimating at time $t$ the next input $x_{t+1}$ from the current input $x_t$ and from the history of computations that is captured by the recurrent context layer.

The internal state of each neuron in the recurrent context layer, $u^{rec}_{t,i}$, is updated by integrating the internal state of the previous time step with the current input, both scaled by the time scale parameter $\tau$:

$$u^{rec}_{t,i} = (1 - 1/\tau)u^{rec}_{t-1,i} + 1/\tau(\Sigma^{Input} + \Sigma^{Context}) \tag{1}$$

where $\Sigma^{Input} = \sum_{j=1}^{I} w^{inp}_{ij} x_{t,j}$ corresponds to the weighted input from the $I$ input neurons and $\Sigma^{Context} = \sum_{j=1}^{C} w^{rec}_{ij} h_{t-1,j}$ corresponds to the weighted context activation of the previous time step from the $C$ context neurons. The variables $w_{ij}$ denote the weight from the neuron $i$ to the neuron $j$ and $h_{t,j} = tanh(u^{rec}_{t,j})$ denotes the activation of the context neuron $j$ at time $t$.

The internal states of the neurons in the output layers are computed directly from the context layer and the corresponding output weights:

$$u^{out}_{t,i} = \sum_{j=1}^{C} w^{out}_{ij} h_{t,j}, \quad u^{var}_{t,i} = \sum_{j=1}^{C} w^{var}_{ij} h_{t,j}. \tag{2}$$

The two formulas for computing the output estimation $y_{t+1}$ and the variance estimation $v_{t+1}$ are:

$$y_{t+1,i} = tanh(u_{t,i}^{out}), \tag{3}$$
$$v_{t+1,i} = exp(u_{t,i}^{var}). \tag{4}$$

And at last, the weights $w_{ij}$ of the networks are updated in order to maximize the likelihood that the estimated mean and variance values describe the true time series $x$. In each training step, we minimize the negative log likelihood of the predicted training signal, defined as:

$$-\ln L = \sum_{t=1}^{T}\sum_{i=1}^{O}(\ln(2\pi v_{t+1,i}) + \frac{(x_{t+1,i} - y_{t+1,i})^2}{2v_{t+1,i}}), \tag{5}$$

where $O$ is the output dimensionality ($I = O$ in our study). We implemented the SCTRNN via the CHAINER framework [10].

### 3.2 Learning

At each step of the learning, called epoch, several training trajectories are given to the network, this corresponds to a training batch. Then we compute the likelihood and update the network's weights accordingly thanks to the first-order gradient-based backpropagation optimization [11]. When the training data corresponds to several trajectories, they are given in the same proportion and in a random order in each training batch.

To allow the network to learn several trajectories, different initial states of the context layer neurons are set, each one corresponding to a different trajectory. For each trajectory, the initial state of each neuron $i$ is initialized with $u_{0,i}^{rec} = 0$ and updated during training accordingly to the likelihood. If we use only this function to optimize the initial states they would constantly diverge to achieve a better separation, thus we use a second optimization function which keeps the variance of the initial states close to a predefined value (see the work of Murata et al. [7] for more details).

### 3.3 External contribution

One important parameter is the external contribution parameter $\chi$, which determines how much the network relies on the external signal and the network's own prediction to update its internal representation, see Figure 2.

The integrated signal $x'_{t+1}$ that corresponds to the input to the context layer, see Figure 2, is generated by combining the external signal $x_{t+1}$ with the estimated signal of the previous time step, $\widehat{x}_{t+1}$. To mimic the noise property of the original trajectory, the estimated signal is generated by adding Gaussian noise with variance $v_{t+1}$ to the averaged output $y_{t+1}$:

$$x'_{t+1} = \chi x_{t+1} + (1-\chi)\widehat{x}_{t+1}, \text{ where } \widehat{x}_{t+1} = \mathcal{N}(y_{t+1}, v_{t+1}). \tag{6}$$

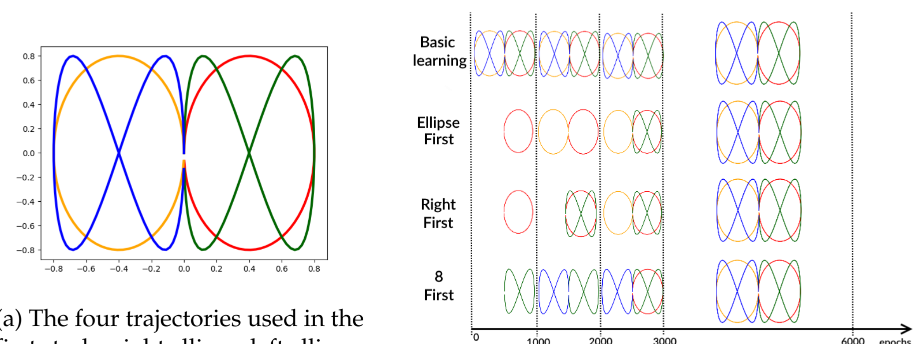
During generation after learning, this parameter can be varied to create different modes of execution. If $\chi = 1$, the network relies only on the external input, it thus reactively follows the external signal without regard to its previous prediction. If smaller values of $\chi$ are used, the network merges the external input with its own estimation. If $\chi = 0$, the network relies only on its own prediction and employs a proactive behavior. Networks with enough training can proactively generate a learned trajectory if the corresponding initial state is selected at the start of the generation.

## 4 First Study: Learning order

### 4.1 Hypothesis

The idea was to study the influence of the learning order of several trajectories. It is believed [8] that children learn more easily if they start by learning the easiest tasks first. We study how it influenced the speed of the learning of our model but also the quality of the internal representation achieved at the end of the learning. We suggest that starting with the simplest shape first leads to a better learning of the task in terms of the quality of the internal representation. We suggest that the internal representation is linked to how the model can react to new inputs and thus generalize its learning.

### 4.2 Method




(a) The four trajectories used in the first study: right ellipse, left ellipse, right 8 shape and left 8 shape

(b) The 4 different order used in the first study.

Fig. 3

The task consisted in drawing four trajectories, see Figure 3a: two shapes, an ellipse and a 8 shape, at two positions, right and left. To represent errors in the perception mechanism and improve stability we added some Gaussian noise to the trajectories. The basic learning of this four trajectories consisted in giving them to the network in the same proportion from the start of the learning. At each epoch, the network will update its weights and initial states accordingly to the four trajectories. With this setting the network was able to learn to draw the four trajectories in less than 2000 epochs.

For the other learnings conditions, in order to see the effect of some ordering we separated the learning in successive phases:

1. the first trajectory for 1000 epochs;
2. the first two trajectories for 1000 epochs;
3. the first three trajectories for 1000 epochs;
4. the four trajectories for 3000 epochs.

Thus, the total length of the learning was fixed 6000 epochs for the four following methods:

- the basic learning: all four trajectories from the start;
- the ellipse (simplest shape) first: Right Ellipse $\rightarrow$ Left Ellipse $\rightarrow$ Right 8 shape $\rightarrow$ Left 8 shape;
- the position right first: Right Ellipse $\rightarrow$ Right 8 shape $\rightarrow$ Left Ellipse $\rightarrow$ Left 8 shape;
- the 8 shape (hardest shape) first: Right 8 shape$\rightarrow$ Left 8 shape $\rightarrow$ Right Ellipse $\rightarrow$ Left Ellipse.

As the conditions: the position left first and the position right first are symmetrical, we did not use the position left first order. Figure 3b gives a schematic view of the four orders we used. We trained 5 networks of 70 neurons for each learning order.

We used two measures: the first one consists of computing the error between the proactive generation of the four trajectories and the training trajectories every 200 epochs during the learning, even when the networks had not seen yet the four trajectories. The second one consists of measuring the quality of the internal representation at the end of the learning.

The internal representation of a recurrent neural network corresponds to the dynamics of its context layer which can be studied by observing the time course of context neuron activations during the drawing. We evaluated the time course activations in the trained networks (after 6000 epochs) during the proactive drawing of the four trajectories. By applying principal component analysis (PCA) [12] on the network's time course activations during trajectory generation, the 70-dimensional time course (corresponding to the 70 neurons activations through the generation) can be projected to a time course in a lower dimensional representation.

Figure 4a depicts an example of the first two principal components of the activation time course. We can see that the trajectories in the same position had time courses in a similar position in the PCA space and that the trajectories of the same shape have similar time courses in terms of shape.

We thus suggested that the internal representation is of good quality if the above two conditions are present:

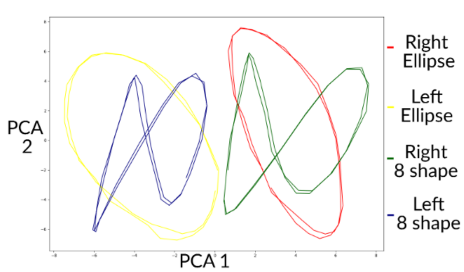
- the activation time courses of the trajectories in the same position (right or left) are close in the activation space and reciprocally, the activation time courses of the trajectories in different positions are far away, we refer to this as *position quality*;

- the activation time courses of the trajectories with the same shape (ellipse or 8 shape) have a similar shape and reciprocally, the activation time courses of the trajectories with different shapes are different, we refer to this as *shape quality*.

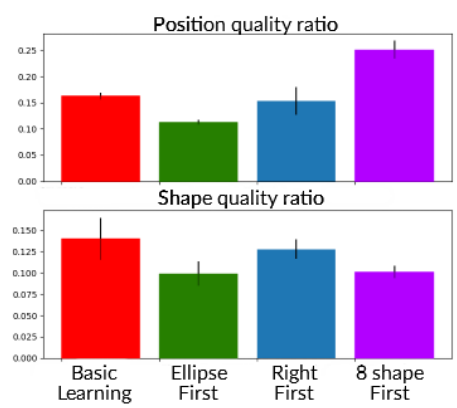
To measure similarity between two activation time courses we used dynamic time warping (DTW) [13] which measures similarity between two temporal sequences which may vary in speed. To measure the position similarity we used the raw activation time course and to measure the shape similarity we used the relative activation time course which consists of the relative movement from one point to the next which corresponds to the derivative.

To obtain two single real values representing the *position quality* and the *shape quality*, we computed the ratio between the measure which should be low: DTW on the raw time course between the same position (DTW on the relative time course between the same shape) and the measure which should be high: DTW on the raw time course between different positions (DTW on the relative time course between different shapes). We finally obtained two ratios, the *position quality* ratio and the *shape quality* ratio, which are low if the internal representation are of good quality.

### 4.3 Result




(a) Example of the first two components of the activation patterns during the drawing of the four trajectories. We can see that the activation time course of the same shape (ellipse or 8 shape) have similar shapes and the activation time courses of the trajectories at the same position are close to each other.




(b) The quality ratios, top: the *position quality* ratio and bottom the *shape quality* ratio. A low ratio means a good quality.

Fig. 4: Analysis of internal representation

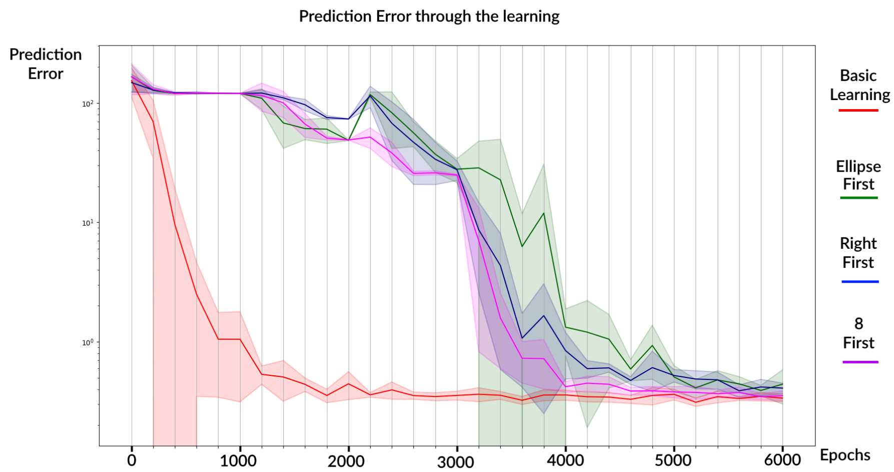



Fig. 5: The prediction error through the learning for the four methods.

From the prediction error depicted in Figure 5, we can see that the basic learning was the quickest to learn to draw the four trajectories, it became asymptotic after 2000 epochs. Starting with the more complex shape first allowed a faster learning than starting with the simplest shape first. And when a new trajectory was added (epochs 1000, 2000 and 3000) there were some following increases in the prediction error. We, thus, looked in more details the prediction error for each trajectory, and we saw that when we added a new trajectory, the old ones were deteriorated afterwards which resulted in an increase of the global prediction error.

From the qualities measures depicted in Figure 4b, we can see that the basic learning resulted in the worst *shape quality* whereas starting with similar shapes first resulted to the best *shape quality*. Starting with the more complex shape first resulted in the worst *position quality* whereas starting with the simplest shapes first resulted in both the best *position quality* and best *shape quality*.

### 4.4 Discussion

Starting the learning with the simplest shape first, allowed the network to build a better internal representation. This observation fits well with results from psychological studies [8] which recommend to have a gradual learning in terms of difficulty for children.

However although the network developed a good internal representation, it took longer for the network to converge. The reason is that the network needed to see the complex shape more times than the simplest shape to learned it whatever is the order. Thus, starting to see this complex shape only at the 3000th epoch retarded the global convergence.

Under the assumption that good internal representation is an important feature for learning, we could demonstrate that only looking at the behaviors of the network does not suffice to study the quality of the learning. We can conclude that it is always good to look at the internal representation learned by the network.

To improve this experiment in a future study it would be better to give to each methods the same amount of training for each trajectory, to only have the order as parameters.

## 5 Second Study: Completion task

### 5.1 Hypothesis

We suggest that the main difference between the chimpanzees and the children is that chimpanzees do not use top-down predication whereas children developed this mechanism during their early years.

The goal of this second study was to replicate some of the behaviors presented in the above mentioned free drawing study [1]. To do so, we wanted to confront our model with a completion task similar to the one given to the chimpanzees and typical developed children. We tested a possible internal mechanism of our network which allows it to replicate those different behaviors. Which can be helpful for understanding the internal mechanisms causing the behaviors of typical developed people and people with ASD.

Our assumption is that the use of top-down prediction can correspond in our model to the inference of the initial state to use because the initial state corresponds to which trajectory the network wants to draw.

### 5.2 Method

We first trained 10 networks during 4000 epochs to draw two shapes, ellipse and 8 shape, at four positions, right, left, top and down. After learning, we gave to those networks the beginning of the trajectories for different lengths: 5, 10, 15, 20 and 25 over the total length 25, in order to see if they were able to complete them.

There were three phases during this completion task, the first phase consisted in inferring an initial state to start the generation. The basic condition is to use the mean of the initial states learned during the training.

The second phase consisted in using the beginning of the trajectories as input for the generation during the length of the given trajectory. For this phase, we tested the $\chi$ values of the generation: 0, 0.5 and 1, which corresponded to proactive, mixed and reactive generation. This allowed us to investigate the influence of the bottom-up sensory perception influence on the task.

The third phase consisted, when the length was inferior to 25, to complete the trajectory. There was no input so $\chi$ was fixed to 0 and the generation was only proactive.

We wanted to compare the behaviors when the network use prediction and when it does not. As we said earlier the only parameter allowing the model to learn to draw several trajectories is the initial state. Hence, predict a trajectory corresponds to infer an initial state which allows the network to draw the trajectory. Predict which trajectory to draw starts by recognizing the incomplete trajectory. We used two different recognition mechanisms for the first phase.

The first one consists of selecting one of the initial states learned during the learning. Given the beginning of a trajectory, we generated the trajectory for each initial state and we chose the one with the least error. We will refer to this inference as the *Best* method.

The second mechanism was introduced by Murrata et al. [9], it consists to use the same training process presented in Section 3 but this time the weights of the network are fixed, thus only the initial states are updated. In this way, it can infer an initial state suitable for generating the observed trajectory. We took the network which has learned to draw the eight trajectories and we reinitialized the initial states to the value of the mean of all eight initial states. And for training data we used the beginning of the trajectories for the different lengths. We then used backpropagation for 200 epochs and we obtained for each beginning trajectory an initial state. We will refer to this inference as the *Backpropagation* (BP) method.

To compare the completion task with no inference we still had to choose the initial states. We decided to use the basic condition, which is to use the mean initial state. This corresponds to the start of the backpropagation as for the backpropagation method, the initial states are reinitialized to this same mean. We will refer to this inference as the *Mean* method.

### 5.3 Result

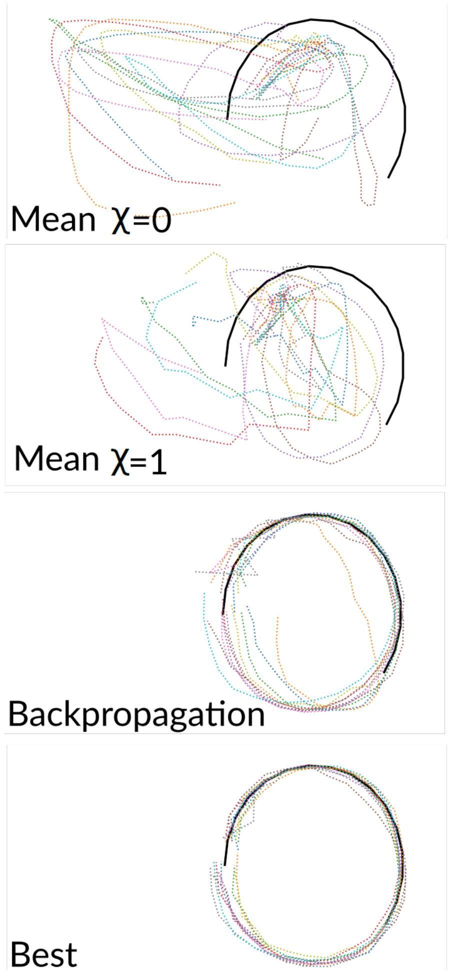
We present in Figures 6a-6b some examples of behaviors presented for the completion of one trajectory. First comparing the different methods for different values of $\chi$ and then for different length of the beginning trajectory for the *Backpropagation* method.

With the *Mean* and proactive generation ($\chi = 0$), the networks drew random scribbles in all the space, but with reactive generation the networks drew random scribbles closer to the position of the trajectory to complete. With *Backpropagation* and not enough length (5 and 10) the networks only realized a partial completion. With *Best* and *Backpropagation* with enough length (15, 20 and 25), the networks were able to complete the trajectories.

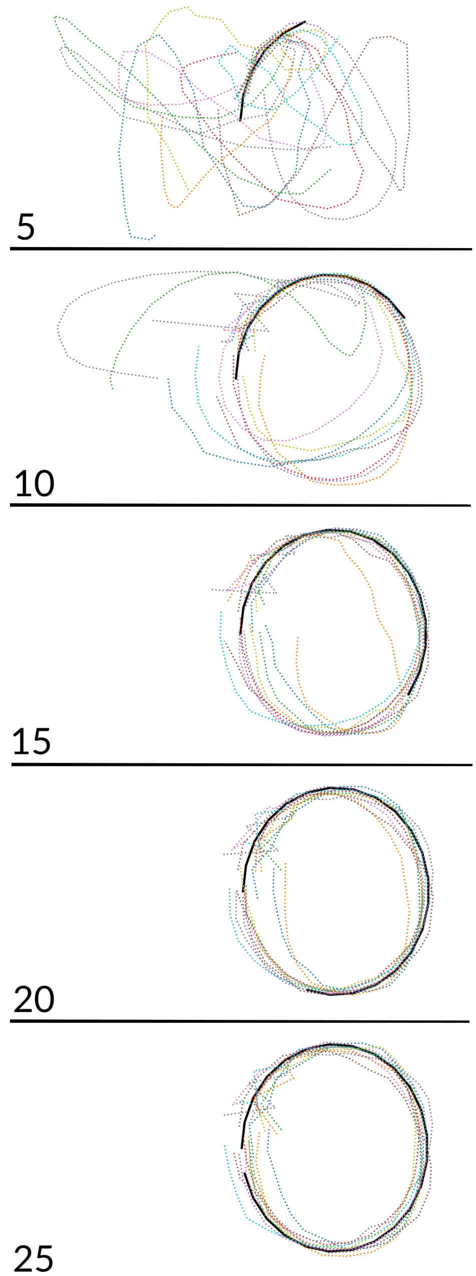
In all cases, we can see in Figure 7 that letting the network use bottom-up sensory perception ($\chi > 0$) decreased the error of the completion, which was expected.

### 5.4 Discussion

We were able to reproduce different behaviors by changing the way the model infers the initial states when trying to complete a trajectory. Using prediction




(a) Comparison of the completion for the length 15. For the Mean initial state inference, using $\chi = 1$ during generation seems to center the scribbles onto the ellipse to complete. For Best and Backpropagation, $\chi = 0$ as the networks were able to draw the trajectory on their own




(b) Completion task results for the Backpropagation initial states inference for different lengths.

Fig. 6: Some examples of the completion task results. The plain black trajectory corresponds to the beginning of the trajectory to complete. The others dashed trajectories correspond to the completion of the 10 networks.

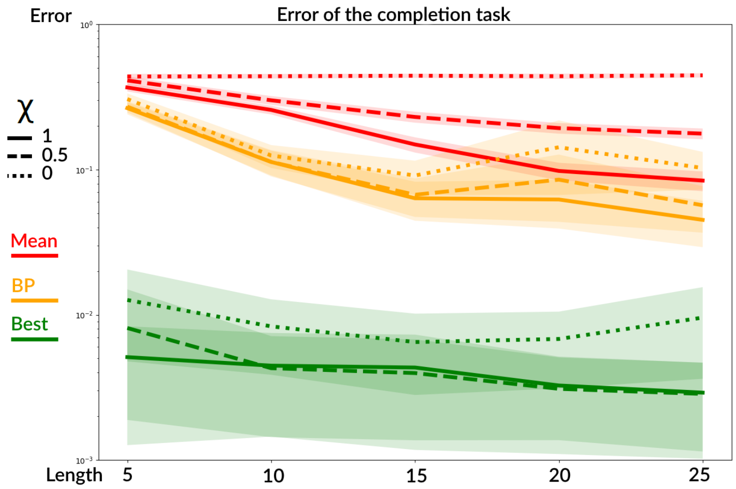



Fig. 7: The average error for all trajectories in function of the lengths for the different methods. Mean had the most error and Best had the least error.

such as the *Backpropagation* method proposed by Murrata et al [9] or the *Best* initial states learned, allowed the network to complete the trajectories, which corresponds to typical developed children's behaviors. Whereas using only the *Mean* initial states, which means no prediction, lead to scribbles similar to chimpanzees' behaviors. It shows that our model is suitable to replicate different behaviors present in individuals. In a future experiment, we want to demonstrate this effect also using noncontinuous shapes such as a face with eyes and mouth.

## 6 Conclusion

The autism spectrum disorder is a disorder with a wide range of behaviors. Today, there is no consensus to diagnose it and little is know about the internal causes. Computational model could provide help to a therapist, indicating which internal mechanisms might cause an observed behavior, suggesting how to support people with ASD, for example, by improving the environment.

In this paper we first study the impact of the learning order. We showed that when learning the easiest shapes first we get a worse prediction error, but a better internal representation. As we suggest that the quality of the internal representation is a key factor for generalization capability and thus a better learning, our findings relate with the Zone of Proximal Development developed by Vygotskii [8].

In the second study, we showed that the model completed trajectories only successfully if the initial states of the context activation layer was inferred before generation. If the network directly tried to generate the trajectory, starting from the mean initial state, the drawn trajectory did not follow the given trajectory. This corresponds to the drawings performed by chimpanzees who do not employ any prediction. As people with ASD are often assumed to perform less prediction than typically developed people, these network mechanism could in future studies be applied to also investigate differences in behavior of people with ASD.

## Acknowledgement

I would like to thank Anja Philippsen and Yukie Nagai for their help, guidance and supervision during my internship.

## References

1. Aya Saito, Misato Hayashi, Hideko Takeshita, and Tetsuro Matsuzawa. The origin of representational drawing: A comparison of human children and chimpanzees. 85, 11 2014.
2. Christensen DL et al. Baio J, Wiggins L. Prevalence of autism spectrum disorder among children aged 8 years autism and developmental disabilities monitoring network. *MMWR Surveill Summ 2018;67(No. SS-6):123.*, 11 Sites, United States, 2014.
3. Elizabeth Pellicano and David Burr. When the world becomes 'too real': a bayesian explanation of autistic perception. *Trends in cognitive sciences*, 16 10:504–10, 2012.
4. Rebecca P. Lawson, Geraint Rees, and Karl J. Friston. An aberrant precision account of autism. *Frontiers in Human Neuroscience*, 8:302, 2014.
5. Yuichi Yamashita and Jun Tani. Spontaneous prediction error generation in schizophrenia. *PLOS ONE*, 7(5):1–8, 05 2012.
6. H. Idei, S. Murata, Y. Chen, Y. Yamashita, J. Tani, and T. Ogata. Reduced behavioral flexibility by aberrant sensory precision in autism spectrum disorder: A neurorobotics experiment. In *2017 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, pages 271–276, Sept 2017.
7. Shingo Murata, Jun Namikawa, Hiroaki Arie, Shigeki Sugano, and Jun Tani. Learning to reproduce fluctuating time series by inferring their time-dependent stochastic properties: Application in robot learning via tutoring. *IEEE Transactions on Autonomous Mental Development*, 5:298–310, 2013.
8. L. S. Vygotskii and M. Cole. Mind in society: The development of higher psychological processes. *Cambridge: Harvard University Press.*, 1978.
9. Shingo Murata, Hiroaki Arie, Tetsuya Ogata, Shigeki Sugano, and Jun Tani. Learning to generate proactive and reactive behavior using a dynamic neural network model with time-varying variance prediction mechanism. *Advanced Robotics*, 28:1189–1203, 2014.
10. Seiya Tokui, Kenta Oono, Shohei Hido, and Justin Clayton. Chainer: a next-generation open source framework for deep learning. In *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Twenty-ninth Annual Conference on Neural Information Processing Systems (NIPS)*, 2015.
11. Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.
12. K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2:559–572, 1901.
13. Meinard Müller. *Information Retrieval for Music and Motion*. Springer-Verlag, Berlin, Heidelberg, 2007.