

# Architectural Plasticity in Spiking Neural Networks

Katarzyna Kozdon, Peter J Bentley

University College London, London, United Kingdom  
k.kozdon@cs.ucl.ac.uk

## Abstract

We can talk about learning optimisation in terms of three biological processes: evolution, development and learning. It has been argued that all three are necessary for intelligence to emerge. Together, they shape the brain through regressive and progressive plasticity.

In this paper, we explored the effects of structural plasticity on learning in spiking neural networks with spike-timing-dependent plasticity: first, we systematically analysed three synapse pruning approaches (random, weight-dependent and activity-dependent) and their effects on networks' weights, spiking activity and performance on a clustering task. Then, we examined the use of a minimalistic evolutionary approach to develop growth rules for spiking neural networks with or without pruning.

We found that pruning combined with a simple weight homeostasis mechanism can be used to reduce spiking neural networks' size without a performance loss; pruning of weak connections increases the learning rate. Evolution of developmental rules led to a rapid fitness increase of the rudimentary embryo networks; addition of pruning significantly improved the learning rate of the model, and synaptic homeostasis preserved stable spiking activity in the networks even during drastic growth.

## Introduction

Learning based on data gathered during an organism's lifetime is only one time-scale of brain's optimisation for data processing. We can talk about this optimisation in terms of three processes: evolution, development and life-long learning, also referred to as phylogenetic, ontogenetic and epigenetic (POE) (Sipper et al. 1997). They take place at different but interconnected time-scales. It has been argued that all three are necessary for intelligence to emerge (Downing 2015).

POE processes lead to changes in the brain architecture in one of two directions: progressive, e.g. through neurogenesis and synaptogenesis (Amrein et al. 2011; Sherwood and Gómez-Robles 2017) and regressive, e.g. through neuronal death and synaptic pruning (Vanderhaeghen and Cheng 2010; Low and Cheng 2006).

Evolutionary origins of the brain are not clear and our knowledge is largely limited to its volume and major structures (Park et al. 2018). Even less is known about neurogenesis, the emergence and evolution of learning and intelligence. The sparsity of the evolutionary data forces us to rely on the comparative analysis of modern species. However, no matter how many species we compare, they all emerged from the same evolutionary process; our sample size of evolutionary processes is equal to 1. Furthermore, brain evolution was restricted by various spatial and metabolic constraints.

Neurodevelopment is a term traditionally used to describe creation, growth, and patterning of the nervous system from embryogenesis until adulthood (Kandel and Schwartz 2013). As the term "adulthood" is usually used to describe a stage after achieving full body size or sexual maturity, it is not always possible to clearly identify this phase in an artificial organism. After adult neurogenesis and connection pruning were discovered, neurodevelopment is also used to refer to plastic processes in the adult brain (Sanes et al. 2012), making it harder to distinguish from what was traditionally referred to as learning.

Pruning (Rathi et al. 2019; Shi et al. 2019; Frazier and Martin 2014), neurogenesis (Tabata and Adachi 2008; Kasabov 2013) and synaptogenesis (Frazier and Martin 2014) have previously been implemented in spiking neural network (SNN) models. Their joint implementations in spiking POE systems are uncommon, but such models receive an increasing interest. In particular, evolved plastic artificial neural networks (EPANNs) (Soltoggio et al. 2018), recreate evolution *in silico* to automatically create and optimise learning agents. Unlike many other neuro-evolutionary approaches, EPANNs aim to create plastic networks capable of continuous learning rather than an optimised final product. This process can be further extended with an explicit developmental mechanism, akin to artificial embryology (Stanley et al. 2003; Miller et al. 2019, 2020).

The nervous system has to constantly adapt to the changes taking place during development and learning to preserve its functionality. A few known mechanisms may contribute to-

wards homeostatic synaptic plasticity (HSP), both at global (Turrigiano 2008) and local (Yu and Goda 2009) level. Synaptic scaling (SS) (Turrigiano et al. 1998) has been suggested to play an important role in maintaining the function of the brain during major architectural changes. Spiking neural network (SNN) models lacking homeostasis experience silencing and saturation, which impedes learning (Abbott and Nelson 2000; Oja 1982). Some of the SNN homeostasis implementations proposed stabilisation of the sum of weights (Liang et al. 2018; Kozdon and Bentley 2019).

In this paper, we explored the effects of progressive and regressive architectural plasticity on learning in SNNs with spike-timing-dependent plasticity (STDP) and weight homeostasis. The first goal of the paper was to systematically examine different synaptic pruning approaches to identify a method which at least preserves SNNs' performance on a task. Pruning lowers the computational cost and data required to train artificial neural networks (LeCun et al. 1990; Zhu and Gupta 2018). The second goal was to examine a minimalistic genetic algorithm which indirectly encodes networks' developmental rules. In both cases, we used a synaptic homeostasis mechanism (Kozdon and Bentley 2019) to test whether it stabilises networks with plastic architecture thus allowing us to work with them over many generations.

## Model description

### Networks with regressive plasticity

The model is an extension of a static-architecture model described in Kozdon and Bentley 2018 and consists of a population of SNNs with leaky integrate-and-fire (LIF) neuron model; 20% of the neurons are randomly selected to be inhibitory. STDP is used to update the weights:

$$\Delta w_{ij} = \begin{cases} w_{ij} \cdot inhLTP, & \text{if } t_j - t_i \geq \text{time\_step} \\ & \text{and } n_j \text{ is excitatory} \\ & \text{and } n_i \text{ is inhibitory} \\ w_{ij} \cdot LTP, & \text{if } t_j - t_i < \text{time\_step} \\ -w_{ij} \cdot LTD, & \text{if } t_j - t_i \geq \text{time\_step} \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

Where  $w_{ij}$  is the weight between the presynaptic neuron  $i$  and postsynaptic neuron  $j$  and  $t$  is spiking time of a given neuron. The parameters long-term potentiation ( $LTP$ ), long-term depression ( $LTD$ ) and inhibitory long-term potentiation ( $inhLTP$ ) (together with the fourth parameter  $discharge$ , which is used to uniformly scale weight values) were previously a subject of evolution; in this paper, they are constant.

At the beginning of evolution, a population of networks is created, and their weight and gene values are initialised with random values within the permitted range. As described previously (Kozdon and Bentley 2018), networks are trained

and tested using 20 x 25 pixel binary bitmaps containing one of four geometric patterns: cross, ellipse, grid and rectangle (fig. 1). Fitness was defined as the ability to cluster the inputs by movement direction. Clustering inputs by shape was also tested to assess performance changes on a fitness-unrelated task during evolution and the possibility of transfer learning (Borthakur and Cleland 2017; Iqbal et al. 2017).

In this paper, the model is extended by the addition of regressive plasticity in the form of synaptic pruning during training, according to one of three approaches:

- **Random:** a constant proportion of connections is removed at random during training. This approach was used to test network disruption.
- **Synaptic strength-dependent:** connections whose weight values are below a set threshold are removed during training, with the caveat that the number of connections pruned per iteration is capped to avoid a rapid network destabilisation and allow the homeostatic mechanism to gradually adapt the network. It has been suggested that synaptogenesis followed by pruning of weak connections improves network performance (Chechik et al. 1998).
- **Synaptic activity-dependent:** connection is removed if a pair of connected neurons do not fire within a given time-window. This approach was inspired by the reports of coordinated firing protecting connections from pruning (Cohen-Cory 2002; Helias et al. 2008).

Additionally, connections are pruned only if the neuron would have at least one input and output left to allow a clearer interpretation of how the removal of synapses affects the networks. The effects of neuron removal would be especially impactful in our model as the sum of weights is a function of the number of neurons.

After training, networks are stabilised using a homeostatic mechanism where the sums of excitatory and inhibitory weights are normalised and are functions of the number of neurons in the network (and thus constant in networks with a constant number of neurons) (Kozdon and Bentley 2019).

$$\begin{aligned} w_{excit} &= n_{excit\_neuron} \cdot 100 \\ w_{inhib} &= n_{inhib\_neuron} \cdot 100 \end{aligned} \quad (2)$$

where  $n_{excit\_neuron}$  and  $n_{inhib\_neuron}$  is the number of excitatory and inhibitory neurons, respectively.

Next generation is created based on the fitness of the SNNs from the previous generation. The fittest third of the networks is used to re-populate next generation by creating a child-clone, a child which undergoes another round of training, and a child with a mutation in one of the learning hyperparameters which then undergoes training (Kozdon and Bentley 2018).

## Evolution of developmental rules

The second part of this paper describes a model using an indirect genetic encoding of developmental rules. Indirect encoding was reported to be beneficial because a small genome can encode even a large network, and because the solution space which evolution has to search is smaller (Downing 2015). The chromosome contains four genes:

- probability of neurogenesis;
- probability of feed-forward synaptogenesis;
- probability of recurrent synaptogenesis;
- probability of horizontal synaptogenesis;

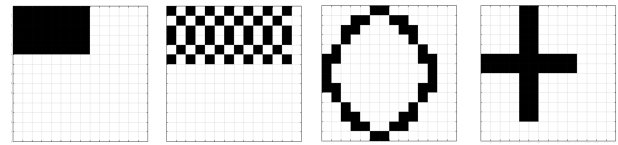
As probabilities, these values are constrained between 0 and 1. A single occurrence of neurogenesis adds a neuron with one input and one output, both feed-forward. Neurogenesis is restricted to the hidden layer; growing the input layer would change the amount of input data available to the networks, whereas changing the number of the output neurons would make it harder to reliably compare the performance of the networks (which is based on clustering of the output vectors) within and between generations. Extending neurogenesis to all layers would be a valid and interesting approach, but here we decided to focus on the role of internal architecture.

A single occurrence of synaptogenesis adds one synapse targeting a random neuron in the next layer (feed-forward), in the previous layer (recurrent) or in the same layer (horizontal). The presynaptic neuron is selected randomly from a qualifying pool of neurons, e.g. the pool of neurons in the input and hidden layers for a feed-forward connection.

Training and testing is performed as previously, then the top fittest half of the networks are used to populate the next generation as follows:

1. Parent A is selected from the fittest quarter of the networks and parent B from the second quarter. The unchanged parents are passed to the next generation.
2. Parent A gives rise to a child-clone. In the child, one randomly selected gene is mutated by adding or subtracting a value between 0.001 and 1.0. The obtained new gene value is constrained to the 0 to 1 range.
3. Parents A and B give rise to a child who inherits a combination of their genes. The gene combination is created using a randomly selected cross-over point; the beginning of the chromosome up to the cross-over point comes from one of the parents (A or B at random), the rest from the other parent.

This approach was based on our preliminary experiments, and selected for its suitability to both preserve best phenotypes and provide a source of novelty throughout the evolution. If the fitness of parent A is lower than 40% (lower than



**Figure 1:** Examples of bitmaps used as input data. A) The data set consisted of a square, grid, ellipse and cross. Each shape was composed of 40 black pixels, it was placed at a random location within the visual field (20 x 25 pixels, one for each input neuron) and moved up, down, left or right with the speed of 1 pixel per frame.

an average randomly-generated organism), population slots assigned to the parent A, parent B (which by default has an even lower fitness) and their children are re-initialised.

The proposed encoding of growth rules is indirect and minimalistic. If only the chromosomes are inherited, evolution will select growth rules which most reliably create fit networks but do not necessarily create similar architectures.

The brain's architecture is indirectly encoded in the genome. Despite its indirectness, the genome reliably encodes core architectural features and lowers the requirement for experience-based learning during organism's life-time. In our model, we can approximate this inheritance of brain structures without increasing the complexity of our genome by making the children inherit architecture of one of their parents.

This model also includes regressive plasticity in the form of synaptic strength-dependent pruning described above.

## Experimental design

### Regressive architectural plasticity

**Experiment 1: Random pruning** In Kozdon and Bentley 2019, we described homeostatic weight normalisation methods for the stabilisation of the activity of SNNs and in result allowing longer training. The best performance was achieved by keeping the sums of excitatory and inhibitory weights constant. Here, the first goal was to test whether this weight homeostasis enables us to work with SNNs with changing connectomes, in particular with networks destabilised with synaptic pruning.

A population of 200 feed-forward networks with 500 input, 50 hidden and 10 output neurons was initialised and observed for several training and testing cycles. A cycle was defined as follows:

1. prune a proportion (1% or 5%) of synapses at random;
2. train the networks;
3. normalise the sums of excitatory and inhibitory connections;
4. test the networks.

This cycle was repeated 20 times, meaning that the networks underwent the same number of training cycles they would during 20 evolutionary generations. To better understand the relationship between pruning and normalisation, we did not use the evolutionary approach: survival of the fittest would alleviate the effects of pruning. Pruning in each cycle ultimately lead to a loss of 19% and 66% of synapses in networks with 1% and 5% pruning rate, respectively.

Learning hyperparameters were set to values evolved by the fittest networks with neuron type-specific weight homeostasis in Kozdon and Bentley 2018, and were as follows:  $discharge = 4.571$ ,  $LTP = 0.052$ ,  $inhLTP = 0.005$ ,  $LTD = 0.033$ .

Time step was = 0.02 s.

Ten independent repeats were performed.

**Experiment 2: Conditional pruning** We hypothesised that pruning of connections in SNNs with weight homeostasis would create smaller networks without the loss of precision. The initial network architecture was as in the previous experiment. We examined two pruning approaches:

- synaptic strength-dependent pruning: synapses were pruned if they were weaker than 0.01;
- synaptic activity-dependent pruning: synapses were pruned if the pair of neurons did not fire together during a whole training session (200 frames).

No more than 1% of weights were pruned per training example (10 input frames) to create a gradual pruning process. Random pruning was not tested here as results of experiment 1 indicated that it was not significantly better than the control.

Details of control network training, testing and evolution were as described in Kozdon and Bentley 2018, except for the number of evolutionary generations which was increased to 200 to allow time for network growth. Networks with synaptic pruning used one of two starting architecture sources:

- the first generation of evolved control networks;
- the last generation of evolved control networks;

to compare the role of a blank slate vs pre-evolved architecture.

Networks were a subject to the survival of the fittest, with the fittest third of the organisms used to repopulate next generation; networks were not re-initialised to avoid repeated replacement of pruned architectures with fresh not-pruned ones.

We tested how pruning affects the performance of the networks, their weight properties and spiking patterns. Significance was calculated using the Kruskal-Wallis test with a *post hoc* pairwise comparisons. The goal was to at least maintain the performance of the networks while decreasing

their size, and to understand how pruning connections in this complex system affects the weight values.

Ten independent repeats were performed.

## Evolution of developmental rules

**Experiment 3: Development of networks with architectural inheritance** We hypothesised that an evolutionary approach can be used to develop growth rules for SNNs with hereditary architectures, and that the ability to prune some of the inherited connections improves learning. In the brain, pruning plays an essential role during development (Low and Cheng 2006) and malfunctions of pruning lead to severe aberrations in the brain connectome and can be lethal (Ceconi et al. 1998; Kuida et al. 1998).

Two populations of 40 network were evolved over 200 generations: one population with progressive architectural plasticity, and one with the addition regressive plasticity in the form of synaptic strength-dependent pruning.

The networks were initialised with only one hidden neuron connected to 50% of input and output neurons at random. The initial architecture was thus feed-forward but could be made recurrent during synaptogenesis. The number of input and output neurons was as previously (500 and 10, respectively).

Learning hyperparameters were set to the values used in experiment 1.

We analysed networks' genotypes and phenotypes (the evolved architectures, activity and performance on the clustering tasks).

Ten independent repeats were performed.

## Results

### Results 1: Random pruning

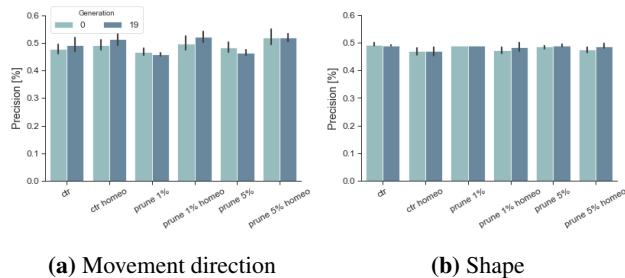
Weight homeostasis significantly improved performance of all networks ( $p \leq 0.02$  for control,  $p \leq 0.001$  for pruned)(fig. 2a). Pruning in the absence of homeostasis led to a gradual fitness decrease, with a significant difference between control and networks with 1% pruning rate; gradual fitness increase was observed in all other cases, but these results did not reach a statistical significance.

Precision changes on the shape-clustering task were less pronounced (fig. 2b).

Overall, the above results suggest that even simple weight homeostasis method based on maintaining a constant sum of neuron type-specific weights achieve promising results in networks with plastic architectures. Homeostasis improved fitness of all networks, and rescued the negative fitness trend in the pruned networks.

### Results 2: Conditional pruning

We tested five set-ups (table 1). Pruned networks initialised with control generation 0 architecture became significantly fitter during evolution (prune\_weak  $p = 0.01$ ; prune\_unused



**Figure 2:** Experiment 1: Clustering precision of inputs by movement direction and shape by networks with synaptic pruning with and without normalisation of the sum of weights. Pruning was performed per round, leading to the final loss of 19% and 66% of synapses in networks with 1% and 5% pruning rate, respectively. Clustering inputs by movement direction was the measure of evolutionary fitness. Bars show best score of the population and the error bars show standard deviation (SD). Population size = 200, independent repeats = 10.

$p = 0.009$ ), and performed better than pruned networks initialised with legacy architecture (control generation 199) suggesting potential benefits of working with blank slate, unevolved networks. Fitness of all networks improved with time, although the trend was more positive in pruned networks (fig. 3a, and table 1). Performance on the shape clustering task decreased for networks (fig. 3b, and table 1). In particular, `prune_weak` (which had the highest performance on the fitness-related task) had the highest precision in generation 0, but the lowest in generation 199 (8.5% decrease). These results suggest that SNNs developed task-specific activity under evolutionary pressure and that performance on the fitness-related task came at the expense of their ability to learn other tasks; no transfer learning was observed.

To better understand how pruning affects the properties of SNNs, we examined networks’ weights. Pruning levels in SNNs with synaptic strength-dependent pruning differed depending on the starting architecture reaching 70% and 54% for blank slate and legacy architecture, respectively (fig. 4 a, and table 1). The number of synapses in SNNs with activity-

dependent pruning decreased only by  $\sim 27\%$ , irrespective of the starting architecture, but analysis of the last ten generations showed a continuation of a significant pruning trend ( $p = 0.05$  and  $p = 0.04$  for `prune_silent` and `prune_silent_legacy`, respectively). This data shows that (for our selected hyperparameters) pruning levels were significantly higher in weight-dependent pruning than in activity-dependent pruning. There was no correlation between pruning level and fitness.

At the end of evolution, proportion of silent weights (weight  $\leq 0.001$ ) was significantly different for all comparisons ( $p < 0.001$ ) (fig. 4 b), but there was no correlation with either fitness or number of synapses.

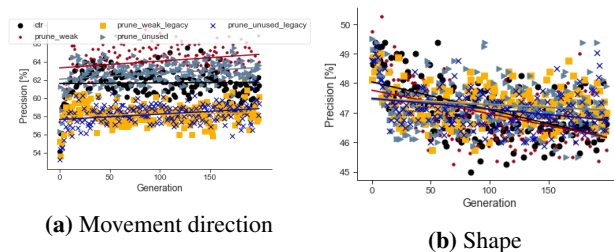
To understand effect of pruning on connectivity, we analysed how many neurons had only one input/output left at the end of evolution (pruning was not permitted to remove the last input/output of a neuron). SNNs with weight-dependent pruning had higher number of these neurons (fig. 4 d), but they also had higher pruning levels. Overall, the proportion of neurons with a single input/output was small ( $< 2\%$ ), showing that pruning was distributed and did not target individual neurons.

We also analysed how pruning affected balance between excitation and inhibition. All networks kept the sums of excitatory and inhibitory weights constant, but the numbers of these connections were allowed to change with pruning. We looked at the sums of weights belonging to four categories based on the identity of the presynaptic and postsynaptic neuron: excitatory-excitatory, excitatory-inhibitory, inhibitory-excitatory and inhibitory-inhibitory. The ratio of synapses of all types changed significantly ( $p < 0.05$ ) in all cases except `prune_weak_legacy` (fig. 4 e-h). The direction of the change was not consistent between different cases, indicating that the choice of a pruning method may indirectly affect balance between excitation and inhibition.

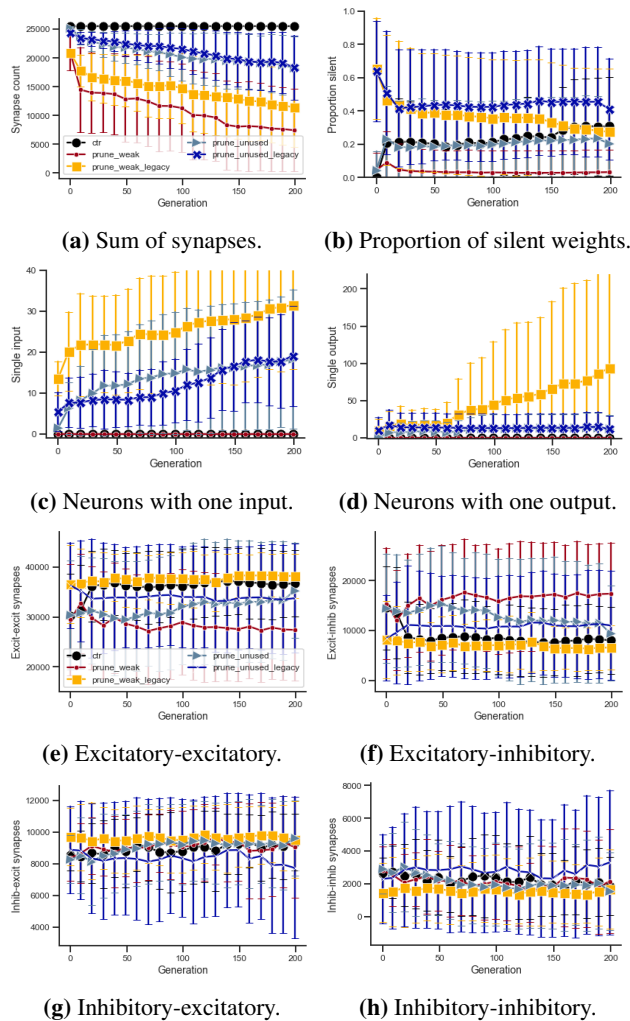
During evolution, spiking in the hidden layer changed significantly in all cases except `prune_unused_legacy` (fig. 5 a), and spiking in the output layer changed significantly in all cases ( $p < 0.001$ ) (fig. 5 b). However, in all cases, spiking

Set-up	Movement clustering [%]		Shape clustering [%]		Pruned synapses [%]
	generation 0	generation 199	generation 0	generation 199	
ctr	54.1, SD = 4.8	61.4, SD = 12.7	48.9, SD = 1.2	46.6, SD = 3.3	0.0, SD = 0.0
<code>prune_weak</code>	57.4, SD = 3.7	64.0, SD = 8.1	49.8, SD = 2.4	41.3, SD = 2.9	70.2, SD = 28.7
<code>prune_weak_legacy</code>	55.3, SD = 10.2	60.1, SD = 13.7	48.1, SD = 2.0	47.0, SD = 3.4	54.2, SD = 27.7
<code>prune_unused</code>	56.4, SD = 4.2	63.6, SD = 7.7	49.5, SD = 1.7	47.9, SD = 4.4	26.9, SD = 22.0
<code>prune_unused_legacy</code>	53.3, SD = 9.0	59.4, SD = 12.1	48.8, SD = 0.0	46.7, SD = 2.0	26.6, SD = 22.2

**Table 1:** Experiment 2: comparison of network pruning methods: control, synaptic strength-dependent pruning initialised with control architecture from generation 0 and 199 (“`prune_weak`” and “`prune_weak_legacy`”, respectively), and synaptic activity-dependent pruning (“`prune_unused`” and “`prune_unused_legacy`”). For each set-up, the table lists precision of clustering by movement direction (fitness) and by shape, and percent of synapses pruned during evolution. The lowest and highest values in each column were highlighted blue and pink, respectively. SD is standard deviation. Population size = 30, independent repeats = 10.



**Figure 3:** Experiment 2: Precision of networks with different pruning mechanisms. Markers indicate best score of the population, bars show SD. Clustering inputs by movement direction was the measure of evolutionary fitness. Population size = 30, independent repeats = 10.



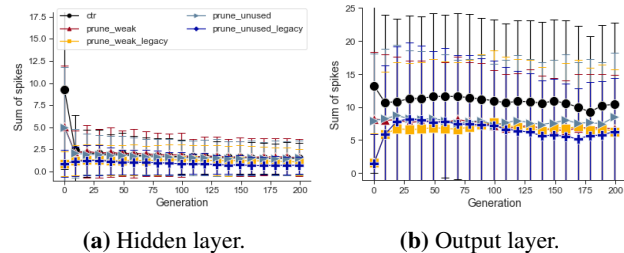
**Figure 4:** Experiment 2: Analysis of synaptic weights in networks with synaptic pruning. Median is marked on each box; error bars indicate SD. Population size = 30, independent repeats = 10.

levels plateaued and networks maintained a desirable level of sparse spiking without saturation and silencing.

Overall, fitness in the presence of pruning was not significantly different from the control, but the number of weights was significantly reduced.

### Results 3: Evolution of developmental rules

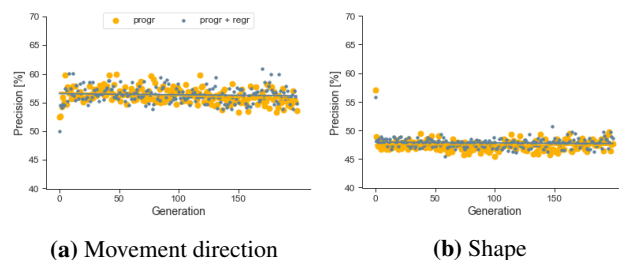
Two set-ups were tested: progressive plasticity, and a combination of progressive and regressive plasticity. Both set-ups showed a sharp initial increase in fitness followed by a gradual decrease (fig. 6a, table 2). The improvement was significant in SNNs with a combination of progressive and regressive plasticity ( $p = 0.02$ ). No significant improvement was recorded for progressive plasticity alone ( $p = 0.57$ ); its final fitness was also lower despite better initial results. Both set-ups showed a significant precision loss on the shape clustering task ( $p \leq 0.001$ )(fig. 6b). These results show that



**Figure 5:** Experiment 2: Analysis of spiking activity in networks with synaptic pruning. Error bars indicate SD. Population size = 30, independent repeats = 10.

Set-up	Movement clustering [%]		Shape clustering [%]	
	generation 0	generation 199	generation 0	generation 199
progr	52.5, SD = 5.7	53.6, SD = 4.7	57.0, SD = 2.4	47.6, SD = 2.8
progr + regr	50.0, SD = 3.6	54.8, SD = 3.9	55.8, SD = 4.5	48.6, SD = 4.1

**Table 2:** Experiment 3: Precision of networks with evolvable developmental rules (corresponds to fig. 6). The lowest and highest values in each column were highlighted blue and pink, respectively. SD is standard deviation.

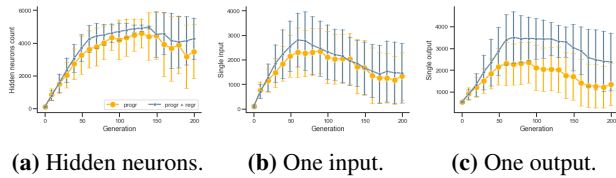


**Figure 6:** Experiment 3: Precision of networks with evolvable developmental rules. Two set-ups were tested: networks with evolvable progressive plasticity (“progr”) and networks with evolvable progressive plasticity and non-evolvable pruning of weak connections (“progr + regr”). Markers indicate best score of the population, bars show SD. Clustering inputs by movement direction was the measure of evolutionary fitness. Population size = 40, independent repeats = 10.

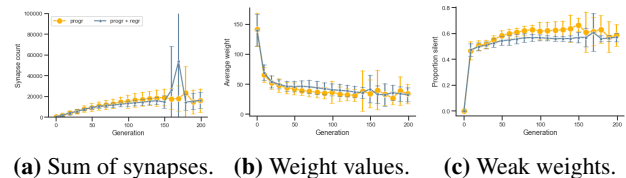
combining progressive and regressive plasticity improved the initial learning rate; however, both set-ups need further modification to avoid fitness loss with time. As before, increase in fitness happened at the expense of performance on the other task.

Next, we analysed networks’ genotypes and phenotypes. In the absence of evolutionary preference, the gene values were expected to oscillate around 0.5. Neurogenesis probability did not change significantly during evolution (fig. 8A). Conversely, networks with high probability of feed-forward synaptogenesis became significantly more abundant (fig. 8B)(progressive plasticity: 0.74, SD = 0.22,  $p = 0.002$ ; mixed plasticity: 0.78, SD = 0.26,  $p = 0.008$ ). Networks with mixed plasticity showed a significant evolutionary preference for higher probability of horizontal synaptogenesis (fig. 8C)(0.73, SD = 0.2,  $p = 0.02$ ). In case of both set-





**Figure 7:** Experiment 3: Architecture changes. A) Number of neurons in the hidden layer. Networks were initialised with a single hidden neuron connected at random to a half of input and output neurons. Probability of neurogenesis was genetically encoded. B) Number of single-input neurons. New neurons were initialised with a single input; in networks with regressive pruning, weak connections were removed providing there was at least one input left. C) Number of single-output neurons. New neurons were initialised with a single output; in networks with regressive pruning, weak connections were removed providing there was at least one output left. Markers show mean values, and error bars indicate SD. Population size = 40, independent repeats = 10.



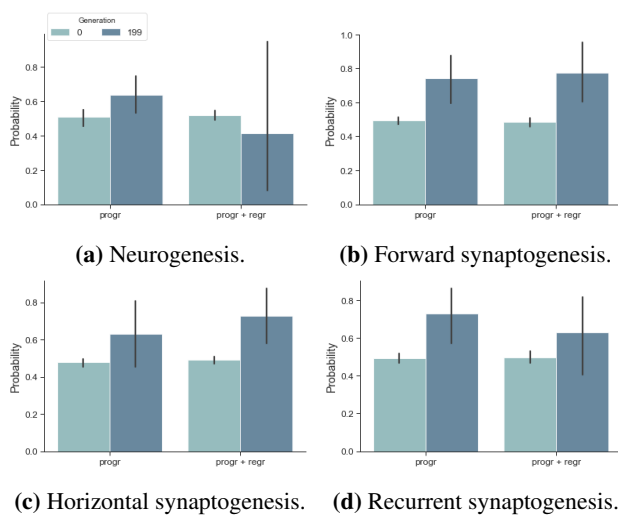
**Figure 9:** Experiment 3: Synaptic properties of networks with architectural plasticity. Markers show mean of the population, and error bars indicate SD. Population size = 40, independent repeats = 10.

at the end of evolution, the average number of hidden neurons was equal 3494 (SD = 1651) in networks with progressive plasticity and 4271 (SD = 1266) networks with mixed plasticity. The difference between the tested conditions was significant ( $p < 0.001$ ).

The number of neurons with one input increased during evolution (fig. 7B), but displayed a downward trend from generation 100 onward; this trend resembles changes in the number of hidden neurons. At the end of evolution, the number of such neurons was 1333 (SD = 960,  $p < 0.001$ ) in networks with progressive plasticity, and 1461 (SD = 1211,  $p < 0.001$ ) for networks with mixed plasticity. Differences between the two tested conditions were not significant ( $p > 0.6$ ), and pruning did not have a significant effect on the number of neurons with one input, which suggests that the presence of neurons with a single input was caused by neurogenesis rather than pruning.

The number of neurons with a single output followed the same trend as neurons with a single input (fig. 7C). At the end of evolution, networks with progressive plasticity had on average 1355 (SD = 955) single-output neurons, and networks with mixed plasticity had 2392 (SD 1314,  $p < 0.001$ ). These results suggest that regressive plasticity (pruning of weak synapses) had a significant effect on the number of outputs but not inputs. This is likely due to activity differences between the layers and subsequently the STDP patterns and weight values (the mapped data always activate a proportion of neurons in the input layer).

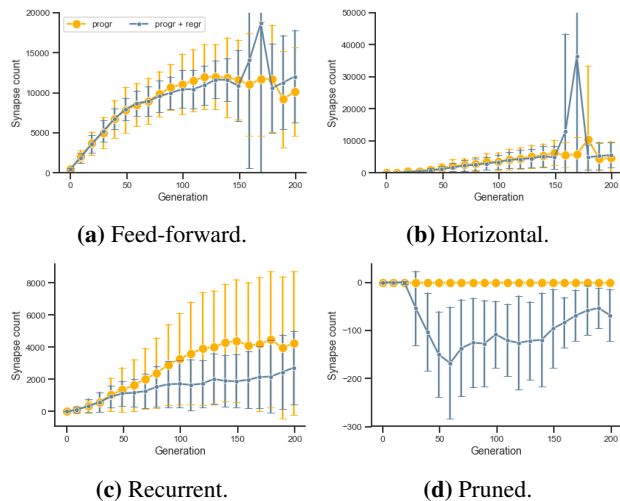
For both set-ups, the number of synapses significantly increased with time (fig. 9A) and reached 16104 (SD = 11159) in networks with progressive plasticity and 16451 (SD = 7617) for mixed plasticity. Pruning did not have a significant effect ( $p = 0.2$ ). Networks with mixed plasticity displayed a sharp transient increase in the number of synapses between generations 160 and 170. The increase was caused by two networks whose synapse count was two orders of magnitude higher than the average. They were subsequently removed from the population because of their low fitness: because of the high synaptic count, individual weight values were very low and 99.3% of the synapses were silent. Therefore, homeostasis of the sum of weights indirectly forced the networks to avoid excessive growth.



**Figure 8:** Experiment 3: Four genes were used to encode the probabilities of growth during each iteration. Error bars indicate SD. Population size = 40, independent repeats = 10.

ups, there also was an evolutionary preference for networks with a higher probability of recurrent synaptogenesis (fig. 8D)(progressive plasticity: 0.73, SD = 0.25,  $p = 0.08$ ; mixed plasticity: 0.63, SD = 0.30,  $p = 0.008$ ). These results show that networks with higher probabilities of synaptogenesis were associated with a higher fitness and were preferred during evolution. A wide range of neurogenesis values evolved in networks with mixed plasticity, indicating that there existed a preference for certain values, but it was not consistent between experimental repeats.

The average number of neurons in the hidden layer significantly increased with time (fig. 7A). However, after generation 150, there was a slight decrease in the number of neurons caused by the removal of unfit networks from the population. All networks were initialised with 1 hidden neuron;



**Figure 10:** Experiment 3: Connection types in networks with architectural plasticity. Markers show mean, and error bars indicate SD. Population size = 40, independent repeats = 10.

The increase in the number of synapses combined with the homeostasis of the sum of weights led to an decrease in the average weight values (fig. 9B). Direct analysis of the number of weak weights (fig. 9C) confirmed that the network size has increased, but its large proportion was inactive. Pruning of weak connections did not have a significant effect.

The number of all connection types significantly increased with time (fig. 10 A-C); networks with mixed plasticity evolved significantly more forward and horizontal connections whereas networks with progressive plasticity has more recurrent connections ( $p < 0.001$  for all), which echoed their genotype differences.

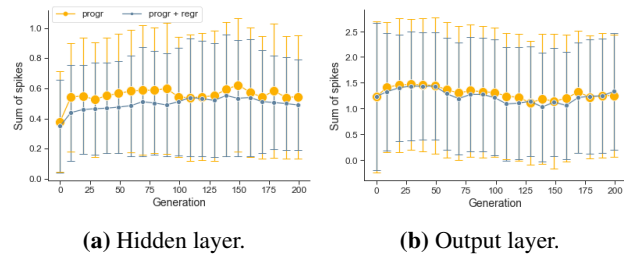
The number of connection pruned per generation fluctuated during evolution (fig. 10D, and ranged from 0 in generations 0 to 19, to 168 in generation 59. These results suggest that only a small proportion of total and weak connections were pruned, despite a high proportion of connections being silent. This is consistent with a high number of neurons having only one input or output, a condition which was protecting their connections from being pruned, and explains why addition of regressive plasticity had limited effects.

Spiking in the hidden and output layers remained stable during evolution, showing that synaptic homeostasis stabilised the networks despite the dramatic architectural changes, and can reliably be used to enable work with POE models of SNNs.

## Summary

We explored the neuroscience-inspired mechanism of structural plasticity (recessive and progressive) in the context of evolution, development and learning.

First, we demonstrated that weight type-specific synap-



**Figure 11:** Experiment 3: Analysis of spiking activity in networks with synaptic pruning. Markers show mean, and error bars indicate SD. Population size = 40, independent repeats = 10.

tic homeostasis reliably stabilises activity and improves fitness in SNNs, including SNNs with architectural plasticity. Secondly, we showed that synaptic strength-dependent and activity-dependent pruning mechanisms can be used to create smaller networks without the loss of precision; removal of weak weights improved the learning rate. Different pruning methods led to distinct activity levels but all tested cases retained desirable stable activity levels.

Finally, we created a POE system to explore the use of evolution to develop SNNs architecture indirectly through optimising developmental rules. Synaptic weights were tuned using STDP and homeostatic synaptic plasticity. Our model demonstrated that it is possible for SNNs with a rudimentary architecture to evolve developmental rules which improve their fitness; the combination of progressive and regressive plasticity was associated with a significant fitness increase during evolution. However, fitness decreased with time, likely due to excessive network growth: we observed that homeostasis of the sum of weights meant that a significant proportion of synapses in large networks was silent, and eventually the networks failed to provide meaningful spiking patterns. Additionally, evolution selected networks with high synaptogenesis probability. These genes were likely beneficial at the beginning of evolution but resulted in an excessive network size in later generations. Extending the evolution would not be an efficient way of correcting the early excessive growth in networks which became silent and need “reviving”. A possible solution to the excessive growth would be to introduce mechanisms which limit the size of the networks, e.g. by introducing a connection cost. Alternatively, the model could be refined by the inclusion of changing levels of plasticity during development, as it was reported in the brain (Sanes et al. 2012).

## Acknowledgements

This work was supported by Biotechnology and Biological Sciences Research Council (BBSRC) and the London Interdisciplinary Biosciences Consortium (LIDo).



## References

- Abbott, L. F. and Nelson, S. B. (2000). Synaptic plasticity: Taming the beast. *Nature Neuroscience*, 3(11s):1178–1183.
- Amrein, I., Isler, K., and Lipp, H.-P. (2011). Comparing adult hippocampal neurogenesis in mammalian species and orders: influence of chronological age and life history stage. *European Journal of Neuroscience*, 34(6):978–987.
- Borthakur, A. and Cleland, T. A. (2017). A neuromorphic transfer learning algorithm for orthogonalizing highly overlapping sensor array responses. In *ISOEN 2017 - ISOCs/IEEE International Symposium on Olfaction and Electronic Nose, Proceedings*, pages 1–3. IEEE.
- Cecconi, F., Alvarez-Bolado, G., Meyer, B. I., Roth, K. A., and Gruss, P. (1998). Apaf1 (CED-4 homolog) regulates programmed cell death in mammalian development. *Cell*, 94(6):727–737.
- Chechik, G., Meilijson, I., and Ruppín, E. (1998). Synaptic Pruning in Development: A Computational Account. *Neural Computation*, 10(7):1759–1777.
- Cohen-Cory, S. (2002). The developing synapse: Construction and modulation of synaptic structures and circuits.
- Downing, K. (2015). *Intelligence Emerging: Adaptivity and Search in Evolving Neural Systems*. MIT Press.
- Frazier, M. and Martin, W. (2014). Effects Of Real-time Synaptic Plasticity Using Spiking Neural Network Architecture. (May):1–10.
- Heliás, M., Rotter, S., Gewaltig, M. O., and Diesmann, M. (2008). Structural plasticity controlled by calcium based correlation detection. *Frontiers in Computational Neuroscience*, 2(December):7.
- Iqbal, M., Xue, B., Al-Sahaf, H., and Zhang, M. (2017). Cross-Domain Reuse of Extracted Knowledge in Genetic Programming for Image Classification. *IEEE Transactions on Evolutionary Computation*, 21(4):569–587.
- Kandel, E. and Schwartz, J. (2013). *Principles of Neural Science, Fifth Edition*. McGraw Hill Professional.
- Kasabov, N. (2013). Evolving spiking neural networks for spatio-temporal pattern recognition. *Neural networks : the official journal of the International Neural Network Society*, 41:188–201.
- Kozdon, K. and Bentley, P. (2018). The Evolution of Training Parameters for Spiking Neural Networks with Hebbian Learning. *The 2018 Conference on Artificial Life*, pages 276–283.
- Kozdon, K. and Bentley, P. (2019). Normalisation of Weights and Firing Rates in Spiking Neural Networks with Spike-Timing-Dependent Plasticity. In *Developmental Neural Networks Workshop, the 2019 Conference on Artificial Life*, Newcastle, United Kingdom.
- Kuida, K., Haydar, T. F., Kuan, C. Y., Gu, Y., Taya, C., Karasuyama, H., Su, M. S., Rakic, P., and Flavell, R. A. (1998). Reduced apoptosis and cytochrome C-mediated caspase activation in mice lacking Caspase 9. *Cell*, 94(3):325–337.
- LeCun, Y., Denker, J. S., and Solla, S. A. (1990). Optimal Brain Damage.
- Liang, Z., Schwartz, D., Ditzler, G., and Koyluoglu, O. O. (2018). The impact of encoding–decoding schemes and weight normalization in spiking neural networks. *Neural Networks*, 108:365–378.
- Low, L. K. and Cheng, H. J. (2006). Axon pruning: An essential step underlying the developmental plasticity of neuronal connections.
- Miller, J. F., Wilson, D. G., and Cussat-Blanc, S. (2019). Evolving Developmental Programs That Build Neural Networks for Solving Multiple Problems. In *Genetic and Evolutionary Computation XVI*, pages 137–178. Springer, Zurich.
- Miller, J. F., Wilson, D. G., and Cussat-Blanc, S. (2020). Evolving Programs to Build Artificial Neural Networks. In Adamatzky, A. and Kendon, V., editors, *Emergence, Complexity and Computation Book 35*, pages 23–71. Springer.
- Oja, E. (1982). Simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, 15(3):267–273.
- Rathi, N., Panda, P., and Roy, K. (2019). STDP-Based Pruning of Connections and Weight Quantization in Spiking Neural Networks for Energy-Efficient Recognition. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 38(4):668–677.
- Sanes, D. H., Reh, T. A., and Harris, W. A. W. A. (2012). *Development of the nervous system*. Academic Press, Burlington.
- Sherwood, C. C. and Gómez-Robles, A. (2017). Brain Plasticity and Human Evolution. *Annual Review of Anthropology*, 46(1):399–419.
- Shi, Y., Nguyen, L., Oh, S., Liu, X., and Kuzum, D. (2019). A soft-pruning method applied during training of spiking neural networks for in-memory computing applications. *Frontiers in Neuroscience*, 13(APR):405.
- Sipper, M., Sanchez, E., Mange, D., Tomassini, M., Pérez-Uribe, A., and Stauffer, A. (1997). A phylogenetic, ontogenetic, and epigenetic view of bio-inspired hardware systems. *IEEE Transactions on Evolutionary Computation*, 1(1):83–97.
- Soltoggio, A., Stanley, K. O., and Risi, S. (2018). Born to learn: The inspiration, progress, and future of evolved plastic artificial neural networks.
- Stanley, K., Bryant, B., and Miikkulainen, R. (2003). Evolving adaptive neural networks with and without adaptive synapses. *The 2003 Congress on Evolutionary Computation, 2003. CEC '03.*, 4(2):2557–2564.
- Tabata, Y. and Adachi, M. (2008). A Spiking Network of Hippocampal Model Including Neurogenesis. In *Advances in Neuro-Information Processing*, pages 14–21, Auckland, New Zealand. Springer.
- Turrigiano, G. G. (2008). The Self-Tuning Neuron: Synaptic Scaling of Excitatory Synapses.
- Turrigiano, G. G., Leslie, K. R., Desai, N. S., Rutherford, L. C., and Nelson, S. B. (1998). Activity-dependent scaling of quantal amplitude in neocortical neurons. *Nature*, 391(6670):892–896.

- Vanderhaeghen, P. and Cheng, H. J. (2010). Guidance molecules in axon pruning and cell death.
- Yu, L. M. and Goda, Y. (2009). Dendritic signalling and homeostatic adaptation.
- Zhu, M. H. and Gupta, S. (2018). To prune, or not to prune: Exploring the efficacy of pruning for model compression. In *6th International Conference on Learning Representations, ICLR 2018 - Workshop Track Proceedings*.