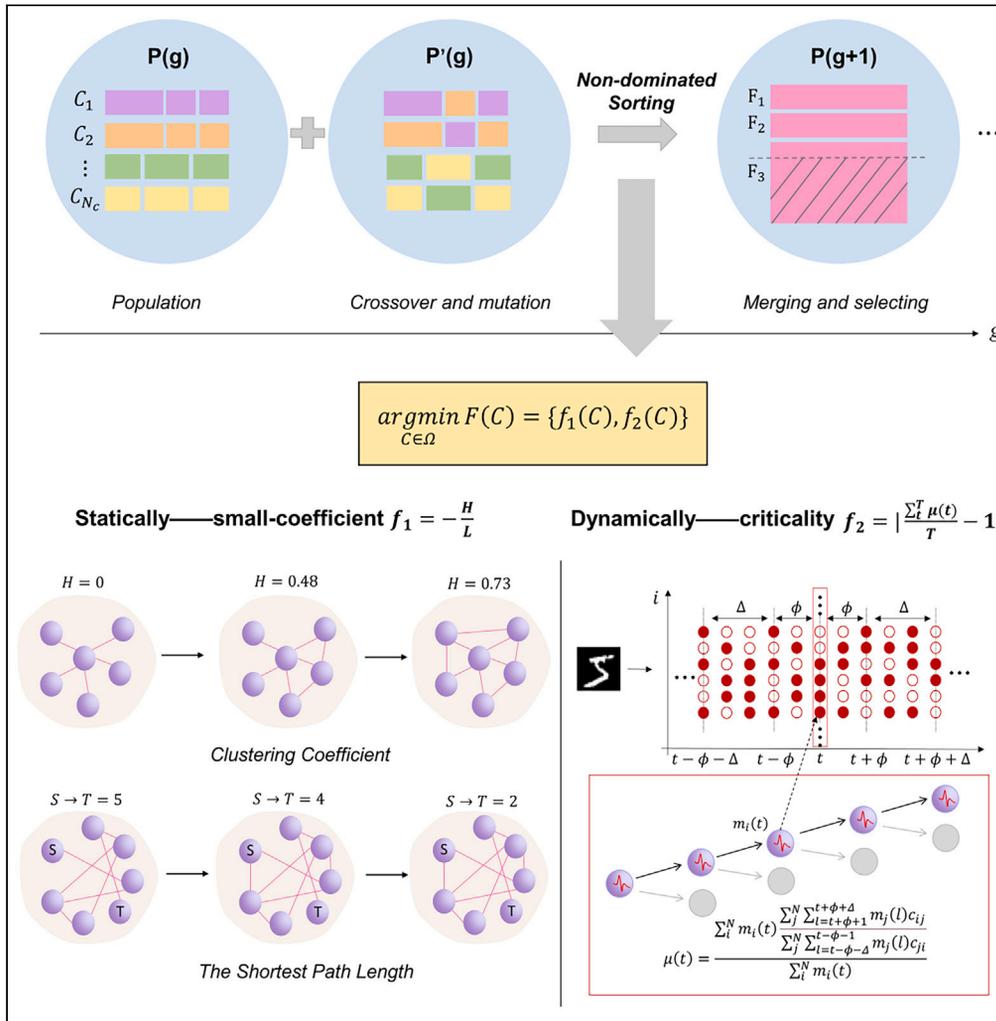


Article

Emergence of brain-inspired small-world spiking neural network through neuroevolution



Wenxuan Pan, Feifei Zhao, Bing Han, Yiting Dong, Yi Zeng

yi.zeng@ia.ac.cn

Highlights

Adaptive evolution driven by brain-inspired small-world topology and criticality

Recurrent SNN emerges with efficient and streamlined brain-inspired structures

Brain-inspired evolution enhances energy-saving and versatility in multiple tasks



Article

Emergence of brain-inspired small-world spiking neural network through neuroevolution

Wenxuan Pan,^{1,2,5} Feifei Zhao,^{1,5} Bing Han,^{1,2} Yiting Dong,^{1,3} and Yi Zeng^{1,2,3,4,6,*}

SUMMARY

Studies suggest that the brain's high efficiency and low energy consumption may be closely related to its small-world topology and critical dynamics. However, existing efforts on the performance-oriented structural evolution of spiking neural networks (SNNs) are time-consuming and ignore the core structural properties of the brain. Here, we introduce a multi-objective Evolutionary Liquid State Machine (ELSM), which blends the small-world coefficient and criticality to evolve models and guide the emergence of brain-inspired, efficient structures. Experiments reveal ELSM's consistent and comparable performance, achieving 97.23% on MNIST and outperforming LSM models on MNIST and Fashion-MNIST with 98.12% and 88.81% accuracies, respectively. Further analysis shows its versatility and spontaneous evolution of topologies such as hub nodes, short paths, long-tailed degree distributions, and numerous communities. This study evolves recurrent spiking neural networks into brain-inspired energy-efficient structures, showcasing versatility in multiple tasks and potential for adaptive general artificial intelligence.

INTRODUCTION

How can the human brain perform many intricate advanced cognitive functions yet run on less power than a light bulb? Numerous studies extend to unravelling the complex wiring rules and firing patterns that govern its operation. There is a notable consistency in brain anatomy across the human species, with different regions frequently associated with specific cognitive functions.¹ Notably, densely connected community structures and hub nodes within these specific regions enhance the efficiency of information processing and integration in the brain.^{2,3}

From a static topological perspective, the mammalian cortex, including the human brain, has been shown to be a complex network that balances randomness and regularity,^{4–7} exhibiting small-world characteristics with densely clustered local connections and short path lengths.^{4,8,9} Dynamically, in the face of complex and fluctuating environments, the human brain demonstrates optimal computational and information processing capabilities near the critical state where network activity oscillates between order and disorder, as well as synchrony and asynchrony.^{10–13}

Developing efficient transmission topologies and optimal dynamic structures is not the result of artificial design but rather a product of natural evolution. While existing human-crafted network structures may enhance performance, they often remain constrained by inherent paradigms.¹⁴ To empower models to autonomously discover optimal network architectures, the field of Neural Architecture Search (NAS) has emerged.^{15–20} Although most NAS work aligns with the deep learning trend, focusing on searching for deep network structures,^{21–25} to the best of our knowledge, none of these algorithms consider the biologically economical small-world topology and critical dynamics characteristic of the brain.

The spiking neural network (SNN), categorized as the third-generation neural network, not only emulates the discrete communication of biological neurons but also integrates multiple biological plasticity learning rules, which is more in line with the information processing mechanism in brain.²⁶ In this work, we utilize a large-scale, recurrently connected SNN known as Liquid State Machines (LSM),²⁷ a kind of reservoir network, which consists of three parts: the input information is processed by a liquid layer containing randomly fixed connections, and then abstracted by the readout neurons into the final output as shown in Figure 1. Owing to its intricate liquid structure and low training cost, LSM serves as a suitable model for investigating brain-inspired connectivity.^{27–33}

Most existing evolutionary LSMs predominantly concentrate on parameter optimization, such as liquid density^{34,35} and liquid size,³⁶ often resulting in inefficiency. An evolutionary framework of a three-step search is introduced,³⁵ including architectural parameters such as multiple-liquid architecture, liquid density, excitatory neuron ratio, and so forth. The impact of dividing the internal architecture of a large liquid into

¹Brain-inspired Cognitive Intelligence Lab, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China

²School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 101408, China

³School of Future Technology, University of Chinese Academy of Sciences, Beijing 101408, China

⁴Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences, Shanghai 200031, China

⁵These authors contributed equally

⁶Lead contact

*Correspondence: yi.zeng@ia.ac.cn

<https://doi.org/10.1016/j.isci.2024.108845>



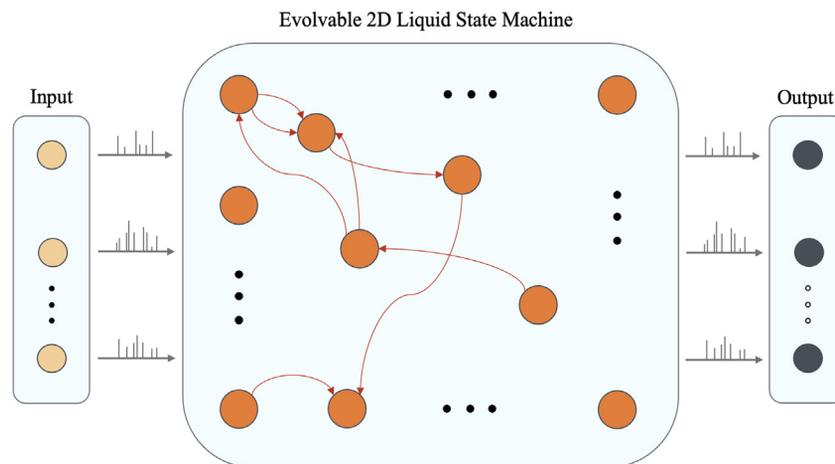


Figure 1. The architecture of LSM

In the traditional definition, randomly connected spiking neurons simultaneously receive time-varying signals from external inputs and other neurons. The recursive connectivity enables input signals to be converted to liquid layer dynamics, which are then abstracted by the readout layer.

multiple small liquids on network efficiency is analyzed.³⁷ Some studies respectively apply covariance matrix adaptive evolutionary strategy (CMA-ES) and differential evolution algorithm (DE) to optimize the topology and parameters of the reservoir.^{34,38} Other NAS-based SNN models aim to maximize classification accuracy with limited computing resources. For instance, energy-efficient SNN architectures³⁹ are evolved for both classification accuracy and the number of spikes. However, directly using performance as a fitness function is prohibitive and time-consuming. A NAS scheme is explicitly designed for SNNs,⁴⁰ with an indirect evaluation method as the evolution goal, which saves the time of directly using the trained classification accuracy and is more in line with the computing mechanism of biological neurons. But architecturally, the search space for just one backward connection is not large enough. In summary, while these studies contribute valuable insights and methodologies to the field, they either fall short on evolutionary innovation, exhibit inefficiency or do not fully leverage the unique topological characteristics of the brain, ultimately constraining their learning capabilities and efficiency.

Inspired by the topology of the biological nervous system, this article introduces a multi-objective evolutionary SNN model that not only incorporates various brain-inspired topological properties but also consistently enhances general performance across multiple tasks. The main highlights can be summarized in the following three points.

- (1) We evolve the architecture of a recurrent spiking neural network to exhibit biologically plausible small-world topological properties (densely local-connected hub nodes, large number of communities, and long-tail degree distribution) and a dynamic critical steady state. Brain-inspired evolutionary goals simultaneously bring about an improvement in classification accuracy.
- (2) The proposed multi-objective evolutionary algorithm considers static small-world coefficients, encompassing shortest path length and clustering coefficients, and dynamic criticality, serving as fitness functions to guide the emergence of brain-inspired efficient structures.
- (3) Our model achieves classification accuracy of 98.12%, 97.23%, and 88.81% on MNIST, NMNIST, and Fashion-MNIST, respectively, superior among models of similar complexity. Experimental results demonstrate that adaptively evolved LSMs improve performance with biologically plausible structures and firing patterns with lower complexity and energy consumption. In reinforcement learning tasks, the decision-making ability that outperforms other algorithms demonstrates the versatility of ELSM across multiple tasks. The degree distribution of the evolved network nodes exhibits the characteristics of a long-tailed distribution, similar to that found in biological brains.

RESULTS

Image classification task

Experimental settings

We validate our model on MNIST,⁴¹ NMNIST,⁴² and Fashion-MNIST⁴³ datasets to prove the effectiveness. The evolved model (whose liquid connection pattern is denoted as C_{opt}) is trained for 5000 epochs. Multiple evolutions are conducted with varying random seeds, and Figure 1 presents the best outcome among these trials.

We construct four ablation models: 1) baseline LSM model with randomly generated liquid layers (marked as RLSM), 2) evolved small world topologies LSM model after 1000 generations (marked as ESLSM), 3) evolved criticality LSM model after 1000 generations (marked as ECLSM), 4) evolved multi-objective LSM model after 1000 generations (the proposed model, marked as ELSM) for the ablation studies of fitness functions on MNIST, NMNIST, and Fashion-MNIST.

Table 1. Detailed parameter settings of ELSM

Parameters	Value	Description
ρ_{init}	0.01	initial connection density
ρ_1	0.001	minimum density allowed by evolution
ρ_2	0.03	maximum density allowed by evolution
N	8000	number of liquid neurons
G_{th}	1000	maximum number of generations
N_c	60	population size
N_{offs}	80	mating pool size
n_m	5	number of mutated genes
T	20	number of time steps
k	2	number of crossover points
m_{rate}	0.5	mutation probability

To ensure the fairness of the comparison experiment, except for the liquid connectivity (the best individual C_{opt}), the other settings of the comparison model during training are the same, including the weight of the liquid layer, the connection between the input and the liquid layer, and the connection between the liquid layer and the readout layer.

The settings for the evolutionary algorithm used in the experiments are detailed in Table 1. All network weights, including the input-liquid weight matrix, liquid-readout weight matrix, and weights inside the liquid layer (the value of the weight rather than the connectivity which is obtained by neuroevolution) are initialized randomly. We set the batch size for all image datasets to 100. The weights between the liquid layer and the readout layer are updated using the AdamW optimizer, with a learning rate of 0.001 and a weight decay applied every 50 epochs at a rate of 0.0001.

Comparative result

The comparison between the proposed ELSM and other models on MNIST, NMNIST, and Fashion-MNIST is shown in Table 2. ELSM achieves 98.12%, 88.81%, and 97.23% accuracy on MNIST, Fashion-MNIST, and NMNIST, respectively. ELSM outperforms the best LSM models reported so far NALSM⁴⁴ by 0.51% and 2.97% on MNIST and Fashion-MNIST, and is superior to another NAS-LSM³⁵ on NMNIST by 4.73% as shown in Table 2. On MNIST, ELSM surpasses other evolved-architecture LSM, LSM-SHADE,³⁸ and Multi-liquid LSM³⁷ by 3.62% and 2.62%, respectively. Overall, ELSM exhibits better performance than all other LSM models on MNIST and Fashion-MNIST.

We also list the SOTA performance achieved by deep models: for example, on NMNIST and MNIST, BP-STA achieves accuracies of 99.71% and 99.67%, respectively, outperforming ELSM by 2.48% and 1.55%. However, its number of parameters is 34 and 649 times larger than ELSM's. When the complexity is similar, ELSM outperforms many hierarchical models as shown in Table 2.

The neuroevolution process we have designed adopts a multi-objective approach, guiding the evolutionary algorithm toward efficiency from the perspectives of both physical topology and network dynamics in the brain, rather than directly pursuing classification accuracy as the sole criterion for evaluating fitness. Surprisingly, as the evolution progresses, individuals not only exhibit brain-inspired properties of small-world and critical state but also demonstrate a significant improvement in classification accuracy, particularly when compared to other LSM models.

Ablation study

Static and dynamic evolution goals. To investigate the impact of various evolutionary goals on the performance of individuals, we conduct ablation studies across each dataset using four models: RLSM, ECLSM, ESLSM, and ELSM—detailing their classification accuracies on each dataset in Table 3. Randomly generated LSMs have low accuracy and large variance. ECLSM performs $97.51 \pm 0.06\%$, $96.7 \pm 0.24\%$, and $88.54 \pm 0.1\%$ on MNIST, NMNIST, and Fashion-MNIST, which is significantly better than all RLSMs by 1.62%, 6.6% and 3.96%. ESLSM achieves $97.88 \pm 0.12\%$, $96.65 \pm 0.23\%$, and $88.24 \pm 0.13\%$ on MNIST, NMNIST, and FMNIST, outperforming RLSMs by 1.99%, 6.55% and 3.66% respectively. The evolutionary models outperform RLSM, with ELSM, which incorporates both small-world properties and criticality as evolutionary goals, showcasing the best performance across various datasets. It can be seen from Table 3 that ELSM has better accuracy and smaller variance than ESLSM and ECLSM, reaching $98.02 \pm 0.03\%$, $97 \pm 0.23\%$, and $88.78 \pm 0.04\%$ performance on MNIST, NMNIST, and Fashion-MNIST respectively.

The comparison of ESLSM, ECLSM, and ELSM across all datasets is presented in Figure 2. Regarding criticality, different ECLSM and ELSM have evolved multiple times (different random seeds) for different datasets. The small-world property is data-independent, meaning that the result of a single evolutionary run can apply to all datasets. However, the chosen optimal individual C_{opt} might vary for each dataset, as it is determined based on the training accuracy after 100 epochs.

Table 2. Comparative performance of different LSM and SNN models on MNIST, NMNIST, and Fashion-MNIST datasets

Dataset	Model	Structure	Layers	Accuracy (%)
MNIST	Unsupervised-SNN ⁵³	Hierarchical SNN	2	95
	LIF-BA ⁵⁴	Hierarchical SNN	3	97.09
	Temporal SNN ⁵⁵	Hierarchical SNN	2	97.2
	STiDi-BP ⁵⁶	Hierarchical SNN	2	97.4
	SN ⁵⁷	Hierarchical SNN	3	97.93
	STDBP ⁵⁸	Hierarchical SNN	7	99.4
	BP-STA ⁵⁹	Hierarchical SNN	8	99.67
	CMA-ES-LSM ³⁴	LSM	2	92.6
	LSM-SHADE ³⁸	LSM	2	94.5
	Multi-liquid LSM ³⁷	LSM	2	95.5
	NALSM ⁴⁴	LSM	2	97.61
ELSM	LSM	2	98.12	
NMNIST	DECOLLE ⁶⁰	Hierarchical SNN	2	96
	AER-SNN ⁶¹	Hierarchical SNN	2	96.3
	STBP NeuNorm ⁶²	Hierarchical SNN	8	99.53
	BP-STA ⁵⁹	Hierarchical SNN	10	99.71
	IonicLSM ⁶³	LSM	2	91.48
	NAS-LSM ³⁵	LSM	2	92.5
	ELSM	LSM	2	97.23
	NALSM ⁴⁴	LSM	2	97.51
Fashion-MNIST	SL-SNN ⁶⁴	Hierarchical SNN	3	85.3
	Unsupervised-SNN ⁵³	Hierarchical SNN	2	85.31
	BS4NN ⁶⁵	Hierarchical SNN	2	87.3
	STDBP ⁵⁸	Hierarchical SNN	7	90.1
	BackEISNN ⁶⁶	Hierarchical SNN	6	93.45
	NALSM ⁴⁴	LSM	2	85.84
	ELSM	LSM	2	88.81

For Figures 2A–2C, the smaller the value on the x axis, the stronger the criticality. For Figures 2D–2F, the larger the value on the x axis, the more obvious the small-world characteristics. From the results of polynomial fitting (shown in blue), it can be seen that as the evolution proceeds, the fitness of C_{opt} of each generation increases continuously and the classification accuracy also increases. It shows a certain degree of positive correlation between the indirect time-saving evolution goal of ELSM and the classification accuracy. In Figures 2A–2F, red marks consistently appear at the top, indicating that ELSM outperforms better performance than single-objective evolution while ensuring maximized two evolutionary objectives.

Liquid size analysis. By adjusting the number of liquid neurons, denoted as N , we conduct a comparative analysis of the classification performance of reservoirs of different sizes on the MNIST and Fashion-MNIST. After several runs, the average results are shown in Table 4.

The results show that the 8000-size reservoir performs best (98.02 ± 0.03) compared to the 5000 (97.65 ± 0.31), 6000 (97.67 ± 0.26), 7000 (97.72 ± 0.23) and 9000 (97.87 ± 0.18) sizes. LSM marked in Table 4 represents the highest accuracy among other LSMs achieved by NALSM.⁴⁴ It can be seen that changing the size of ELSM does not affect its superiority over all LSMs. Regardless of size changes, ELSM consistently outperforms all LSMs on different datasets, which demonstrates the proposed brain-inspired evolution has strong adaptability across diverse network sizes and tasks.

Energy consumption analysis. The energy consumption of spiking neural networks is often qualified by the number of spikes generated. We count the spikes of C_{opt} evolved in the 0th generation, the 500th generation, and the 1000th generation, which are labeled ELSM-0, ELSM-500, and ELSM respectively. The variations before and after evolution are tracked as depicted in Table 5.

Observations indicate a substantial reduction in spiking activity for the initial models (ELSM-0) following 1000 generations of evolution. To establish a benchmark, the timestep for NALSM⁴⁴ is consistent with the configuration of ELSM (20). The spiking activity and corresponding classification accuracy of NALSM in Table 5, showcasing the superiority of ELSM in efficiency: on MNIST, ELSM achieves a 3.55% higher

Table 3. Final performance of models with different evolution goals on all datasets

Dataset	Model	Accuracy (%)
MNIST	RLSM	95.89 ± 0.97
	ESLSM	97.88 ± 0.12
	ECLSM	97.51 ± 0.06
	ELSM	98.02 ± 0.03
NMNIST	RLSM	90.1 ± 29.94
	ESLSM	96.65 ± 0.23
	ECLSM	96.7 ± 0.24
	ELSM	97 ± 0.23
Fashion-MNIST	RLSM	84.58 ± 1.48
	ESLSM	88.24 ± 0.13
	ECLSM	88.54 ± 0.1
	ELSM	88.78 ± 0.04

accuracy than NALSM, while saving approximately 50% of energy. On NMNIST, ELSM enhances accuracy by 1.53% and concurrently reduces energy consumption by roughly 60%.

This decline in energy consumption is accompanied by a notable enhancement in classification accuracy, underscoring the efficiency of the evolutionary process on different datasets: on MNIST, compared to the unevolved model, ELSM-500 achieves approximately a 30% reduction in energy and a 2.86% increase in accuracy, whereas ELSM reduces energy by approximately 60% and enhances accuracy by 3.02%. On

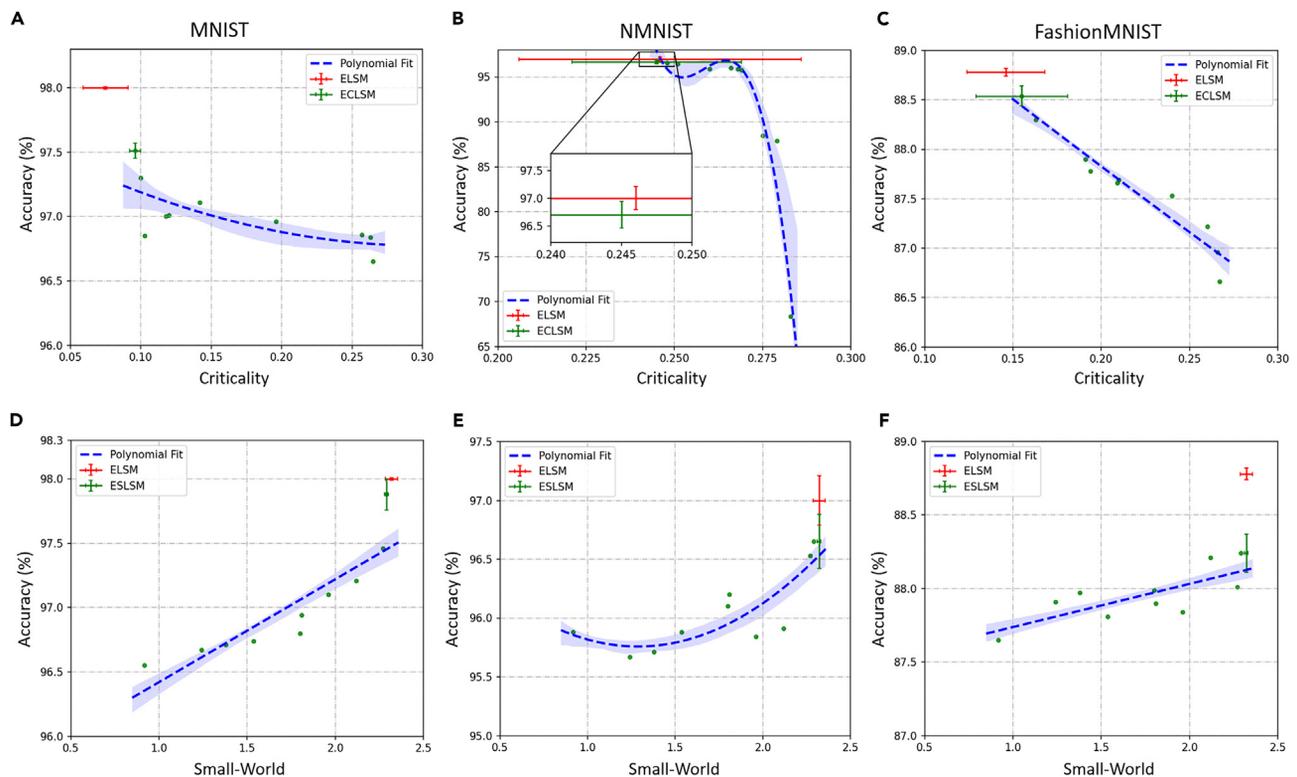


Figure 2. Comparison of ELSM, ECLSM and ESLSM on different datasets

(A–C) Results on evolving criticality on MNIST, NMNIST and Fashion-MNIST. The x axis represents the distance between the criticality of the individual and 1, as shown in Equation 15.

(D–F) Results on evolving small-world properties on MNIST, NMNIST and Fashion-MNIST. The x axis represents the small-world coefficient of the individual as shown in Equation 8. The green dot represents the result of 5000 epochs training of C_{opt} selected in every 100 generations and is fitted by a polynomial (blue line). The green marks with variance indicate the final single-object evolution results of ECLSM and ESLSM, and the red mark indicates the multi-object evolution result of ELSM.

Table 4. Average performance of different-sized LSM on MNIST and Fashion-MNIST

Dataset	Model	N	Accuracy (%)
MNIST	LSM	–	97.61
	ELSM	5000	97.65±0.31
	ELSM	6000	97.67±0.26
	ELSM	7000	97.72±0.23
	ELSM	8000	98.02±0.03
	ELSM	9000	97.87±0.18
Fashion-MNIST	LSM	–	85.84
	ELSM	5000	87.52 ± 0.24
	ELSM	6000	87.81±0.19
	ELSM	7000	88.43±0.12
	ELSM	8000	88.78±0.04
	ELSM	9000	88.22±0.35

NMNIST, ELSM-500 achieves approximately a 50% reduction in energy and a 14.32% increase in accuracy, whereas ELSM reduces energy by approximately 60% and enhances accuracy by 14.67%.

Overall, validation on multiple datasets shows that the evolutionary process endows ELSM with superior classification performance and energy consumption, significantly outpacing the most optimal LSM model⁴⁴ in terms of efficiency.

Decision-making task

In order to verify the versatility of ELSM across multiple tasks, we construct a 3x5 T-shaped maze as shown in Figure 3A, in which ELSM receives the current state of the agent as input, and outputs a probability distribution over actions: up (0), right (1), and down (2). The liquid size is set to 64, and other parameters and algorithm processes are consistent with the image recognition task. Each action taken by the agent is recorded as a step, and the total simulation is 500 timesteps.

Specifically, the state of the agent has three dimensions which are the observed conditions in the three directions (maybe wall, road, food, and poison, represented by 0, 1, 2, and 3, respectively). For example, the state of the agent shown in Figure 3A at the starting point is recorded as (0, 1, 0), that is, walls are seen to the left and right, and the road is seen ahead.

Parameters update

ELSM calculates the loss of each step based on the reward collected and updates the parameters of the readout layer based on the Policy Gradient algorithm. The change in the distance between the agent and food after taking the action is denoted as *dis*, and the reward function is:

$$\text{reward} = \begin{cases} 3, & \text{achieve food} \\ -3, & \text{achieve poison} \\ 1, & \text{dis} < 0 \\ -1, & \text{dis} \geq 0 \end{cases} \quad (\text{Equation 1})$$

Comparative results

To evaluate the decision-making ability of ELSM, we compare it with the classic reinforcement learning algorithm Q-learning⁴⁵ ($\gamma = 0.9$, $\alpha = 0.1$ in the taken Bellman equation Eq. 2), DQN⁴⁶ (one input layer, one hidden layer with 50 neurons and one output layer, $\gamma = 0.99$, $\alpha = 0.1$), and a two-layer LSTM⁴⁷ (one layer LSTM contains 64 neurons and one fully connected layer) also trained by the policy gradient algorithm. The

Table 5. Energy consumption comparison on different datasets

Model	MNIST		NMNIST	
	Accuracy (%)	Spikes	Accuracy (%)	Spikes
NALSM ⁴⁵	95.67	17419.02	95.7	22936.37
ELSM-0	95.1	20775.96	82.56	21772.15
ELSM-500	97.97	14497.91	96.92	10776.75
ELSM	98.12	8709.58	97.23	8976.97

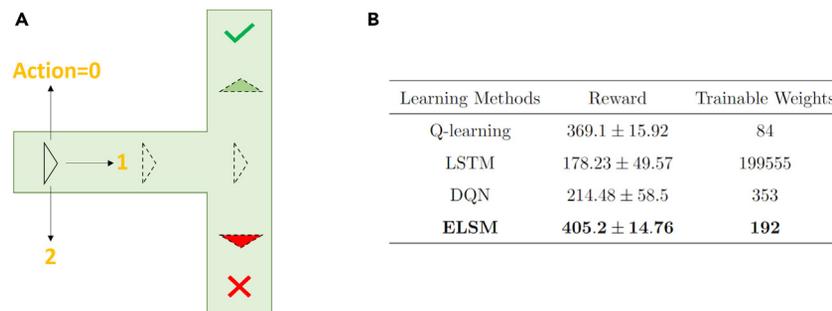


Figure 3. T-maze task

(A) The T-maze environment where triangles represent agents, and green and red endpoints represent food and poison, respectively. The horizontal width of the maze is 3, and the vertical length is 5.

(B) Performance and complexity of different models on T-maze.

sum of rewards obtained by models within $T = 500$ steps is calculated as the performance, which is recorded as *Reward*, as shown in Equation 3.

$$Q(s, a) = Q(s, a) + \alpha \left[\text{reward}(s, a) + \gamma \max_{a'} Q'(s', a') - Q(s, a) \right] \quad (\text{Equation 2})$$

$$\text{Reward} = \sum_t^T \text{reward}_t \quad (\text{Equation 3})$$

The complexity of each model and the average performance over multiple runs are shown in Figure 3B. The performance of LSTM is the worst, probably because such model is not suitable for this task, resulting in overfitting. The performance of DQN is worse than Q-learning, probably because the state and action space of the T-maze task is small, while Q-learning can quickly find the optimal strategy and DQN uses a deep network to approximate the Q-value adding a lot of unnecessary computing cost. The stability of Q-learning is better than DQN and LSTM but lags behind the proposed ELSM. It can be seen that ELSM surpasses other algorithms in both performance and stability with low complexity, owing to the advantages of its brain-inspired static and dynamic topology internally.

DISCUSSION

A long time ago, when the Liquid State Machine (LSM) was first proposed as a tool for exploring brain function,²⁷ many studies^{33,48,49} adopted it in computational models as a general supervised learning technique.⁵⁰ These studies have replicated some biologically plausible findings using LSM, although limitations in network scale and algorithmic mechanisms have constrained their application in computational modeling. Nevertheless, these pioneering efforts have demonstrated the potential of LSMs to investigate the mechanisms underlying brain-inspired structures and functions.

Recent work on the performance-oriented evolution of SNN architectures is often time-consuming and does not adequately capture the topological characteristics associated with brain function. In this work, we introduce ELSM, an evolutionary recurrent Spiking Neural Network model, which uniquely adopts both static small-world topological characteristics and dynamic criticality inspired by brain function—as indirect evolutionary goals. ELSM achieves classification accuracies of 98.12%, 97.23%, and 88.81% on MNIST, NMNIST, and Fashion-MNIST respectively, and outperforms the best LSM models reported so far by 0.44% and 2.97% on MNIST and Fashion-MNIST, surpassing many deep SNN models with an equivalent number of parameters. On MNIST and NMNIST, ELSM achieves performance comparable to deep SNNs, while maintaining extremely low model complexity. Evolution not only enhances the performance of ELSM but also significantly reduces energy consumption. The ablation experiments confirmed that the above two evolutionary goals have a certain degree of positive correlation with the classification accuracy, and the performance of the evolved model far exceeds that of the random LSM. At the same time, the multi-objective evolution model (the proposed ELSM) performs better than the single-objective model. In addition to image classification tasks, ELSM also shows good performance in decision-making tasks compared to other common reinforcement learning models, demonstrating the versatility of ELSM.

We further analyze more structural brain-inspired topological features as follows.

Hourglass structure and sparse coding in *Drosophila* mushroom body

Some studies have found an hourglass-like mapping relationship in the mushroom body module of the *Drosophila* brain: the nervous system converges from the ultra-high-dimensional signals provided by sensory cells to a small number of projection neurons (PNs), and conducts sparse encoding through a large number of Kenyon cells (KCs). Subsequently, lower-dimensional signals are extracted to characterize the real world. This process bears a resemblance to the information processing methodology of LSM, as illustrated in Figure 4. The dotted

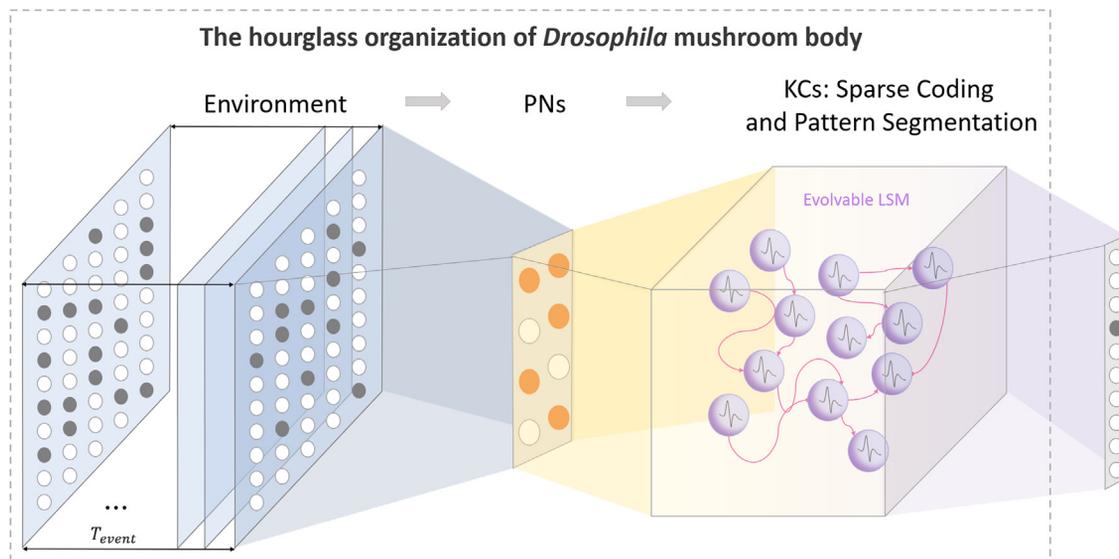


Figure 4. The ELSM inspired by the *Drosophila* mushroom body

The dashed box marks the hourglass structure found in the *Drosophila* mushroom body, consistent with LSM.

box highlights the hourglass structure similar to that in the mushroom body of *Drosophila*, marking a distinct difference between LSM and hierarchical neural networks.

Emergence of structural properties exist in the brain

To study the effect of evolution on the brain-inspired topology of ELSM, we count the changes in structural properties such as clustering coefficient, community, and criticality, as shown in Table 6. The clustering coefficient is used to measure the degree of node aggregation. Communities count the number of communities with a size of 5 in the network that can communicate through 4 common nodes.

While the clustering coefficient increases, the shortest path length slightly decreases (from 0.196 to 0.195), and notably, the total number of connections remains constant. Specifically, the connection density is about 1% in the random network, compared to a marginally lower 0.8% in the evolved network. This demonstrates that evolution achieves rapid and efficient information transfer not by adding more connections, but rather by optimizing existing connections, thereby reducing cost (in terms of the number of connections). The evolved network experiences a substantial increase in its clustering coefficient due to a rise in hub nodes, along with the emergence of more overlapping communities. This suggests a more intricate and highly interconnected set of relationships between subnetworks. Under limited connectivity, larger clustering coefficients, closer community connections, shorter shortest paths, and a state closer to criticality prove that our proposed multi-objective evolutionary algorithm can optimize LSM from both static and dynamic perspectives, and the evolved network architecture is more in line with the core structural characteristics found in the human brain.

The change of the network degree distribution before and after evolution is shown in Figure 5. Figure 5A shows the degree distribution of random LSM, following a normal distribution and the degrees of all nodes are concentrated between 100 and 200. After 1000 generations of evolution, the small-world properties of the network become obvious, and the degree shows an obvious long-tail distribution, as shown in Figure 5B. The evolutionary process results in the emergence of a small number of hub nodes. Among the 8000 neurons, the degrees of most nodes are concentrated between 0 and 100, with fewer nodes as the degree increases. The highest node degree observed is 1525.

Overall, ELSM demonstrates its versatility across multiple tasks with its biologically more plausible small-world coefficient and criticality, surpasses the best LSM models reported so far on MNIST and Fashion-MNIST datasets while achieving comparable performance to many deep SNN models with considerably less complexity. Upon analysis, it is evident that the evolved model possesses numerous topological structures that align with brain networks, including the presence of hub nodes, community clusters, and short path lengths.

Table 6. Changes in topology properties of ELSM before and after neuroevolution

Model	Clustering coefficient (H)	Communities (k = 4)	Criticality ($ \mu - 1 $)	Density (ρ)
Random	285.15	2	0.265	1%
Evolved	319.65	1255	0.096	0.8%

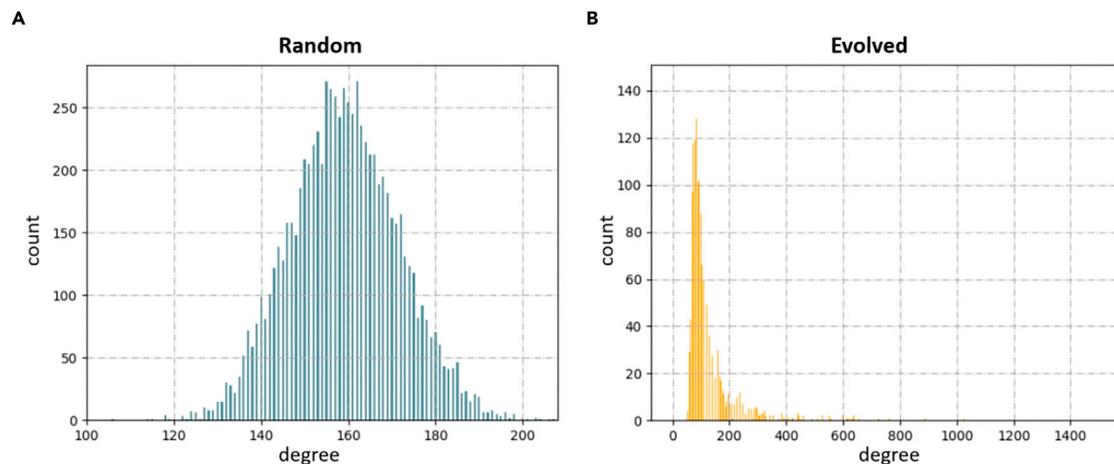


Figure 5. Degree distribution comparison

(A) Degree distributions of liquid layers of random structures.

(B) Degree distribution of the evolved individual.

Limitations of the study

The replication of the cerebellum's connectivity patterns using a low-power microcontroller,⁵¹ confirms that the evolution of neural networks drives the tuning of synapses to reduce systemic free energy. From the perspective of energy consumption, the characteristics of the brain's energy-saving operation mechanism under the immense pressure of information processing, including optimal input and noise, excitation/inhibition balance, the size of neurons and neuron clusters, have been analyzed.⁵² These studies have broadened the scope of potential objectives for brain-inspired evolution, drawing from diverse mechanisms of the human brain. However, their practical applicability in computational modelling remains to be validated. In the future, as neuroscience progresses, we will further explore these feasibilities in the field of artificial intelligence and more operation properties found in brain networks, hoping to identify more effective and energy-efficient brain-inspired functional features on SNNs that can be leveraged to guide efficient evolutionary processes.

In terms of applications, SNNs with various architectures (not just LSM or other deep SNNs) can be configured to form multiple brain regions through global self-organization and co-evolution. This enables the realization of numerous advanced cognitive functions and facilitates research on transfer learning, lifelong learning, and other related topics.

STAR★METHODS

Detailed methods are provided in the online version of this paper and include the following:

- [KEY RESOURCES TABLE](#)
- [RESOURCE AVAILABILITY](#)
 - Lead contact
 - Materials availability
 - Data and code availability
- [METHOD DETAILS](#)
 - Spiking neural network foundation
 - Neuroevolution algorithm
 - Initialization
 - Evaluation
 - Selection
 - Mutation
 - Next generation
 - Static and neuromorphic image processing
 - Training procedure and techniques
- [QUANTIFICATION AND STATISTICAL ANALYSIS](#)

ACKNOWLEDGMENTS

This work is supported by the National Key Research and Development Program (Grant No. 2020AAA0107800), the National Natural Science Foundation of China (Grant No. 62106261).

AUTHOR CONTRIBUTIONS

W.Pan, F.Zhao, B.Han, and Y.Dong designed the study under the supervision of Y.Zeng. W.Pan, F.Zhao, B.Han Y.Zeng, and Y.Dong performed the experiments and the analyses. W.Pan, F.Zhao, and Y.Zeng wrote the article.

DECLARATION OF INTERESTS

The authors declare that they have no competing interests.

Received: May 8, 2023

Revised: August 23, 2023

Accepted: January 3, 2024

Published: January 9, 2024

REFERENCES

- Sherrington, C. (1907). The integrative action of the nervous system. *J. Nerv. Ment. Dis.* 34, 801–802.
- Deco, G., Tononi, G., Boly, M., and Kringelbach, M.L. (2015). Rethinking segregation and integration: contributions of whole-brain modelling. *Nat. Rev. Neurosci.* 16, 430–439.
- Bassett, D.S., Greenfield, D.L., Meyer-Lindenberg, A., Weinberger, D.R., Moore, S.W., and Bullmore, E.T. (2010). Efficient physical embedding of topologically complex information processing networks in brains and computer circuits. *PLoS Comput. Biol.* 6, e1000748.
- Bassett, D.S., and Bullmore, E. (2006). Small-world brain networks. *Neuroscientist* 12, 512–523.
- Bullmore, E., and Sporns, O. (2012). The economy of brain network organization. *Nat. Rev. Neurosci.* 13, 336–349.
- Salvador, R., Suckling, J., Coleman, M.R., Pickard, J.D., Menon, D., and Bullmore, E. (2005). Neurophysiological architecture of functional magnetic resonance images of human brain. *Cerebr. Cortex* 15, 1332–1342.
- Vaessen, M.J., Hofman, P.A.M., Tijssen, H.N., Aldenkamp, A.P., Jansen, J.F.A., and Backes, W.H. (2010). The effect and reproducibility of different clinical dti gradient sets on small world brain connectivity measures. *Neuroimage* 51, 1106–1116.
- Hilgetag, C.C., and Kaiser, M. (2004). Clustered organization of cortical connectivity. *Neuroinformatics* 2, 353–360.
- Sporns, O., and Zwi, J.D. (2004). The small world of the cerebral cortex. *Neuroinformatics* 2, 145–162.
- Fosque, L.J., Williams-García, R.V., Beggs, J.M., and Ortiz, G. (2021). Evidence for quasicritical brain dynamics. *Phys. Rev. Lett.* 126, 098101.
- Priesemann, V., and Shriki, O. (2018). Can a time varying external drive give rise to apparent criticality in neural systems? *PLoS Comput. Biol.* 14, e1006081.
- Rocha, R.P., Koçillari, L., Suweis, S., De Filippo De Grazia, M., de Schotten, M.T., Zorzi, M., and Corbetta, M. (2022). Recovery of neural dynamics criticality in personalized whole-brain models of stroke. *Nat. Commun.* 13, 3683.
- Shew, W.L., Yang, H., Yu, S., Roy, R., and Plenz, D. (2011). Information capacity and transmission are maximized in balanced cortical networks with neuronal avalanches. *J. Neurosci.* 31, 55–63.
- Ren, P., Xiao, Y., Chang, X., Huang, P.-Y., Li, Z., Chen, X., and Wang, X. (2021). A comprehensive survey of neural architecture search: Challenges and solutions. *ACM Comput. Surv.* 54, 1–34.
- Pham, H., Guan, M., Zoph, B., Le, Q., and Dean, J. (2018). Efficient neural architecture search via parameters sharing. In *International conference on machine learning (PMLR)*, pp. 4095–4104.
- Brock, A., Lim, T., Ritchie, J.M., and Weston, N. (2017). Smash: one-shot model architecture search through hypernetworks. Preprint at arXiv. <https://doi.org/10.48550/arXiv.1708.05344>.
- Gong, X., Chang, S., Jiang, Y., and Wang, Z. (2019). Autogan: Neural architecture search for generative adversarial networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3224–3234.
- Pasunuru, R., and Bansal, M. (2019). Continual and multi-task architecture search. Preprint at arXiv. <https://doi.org/10.48550/arXiv.1906.05226>.
- Li, C., Peng, J., Yuan, L., Wang, G., Liang, X., Lin, L., and Chang, X. (2020). Block-wisely supervised neural architecture search with knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1989–1998.
- Zhang, M., Li, H., Pan, S., Chang, X., and Su, S. (2020). Overcoming multi-model forgetting in one-shot nas with diversity maximization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7809–7818.
- Zhang, T., Lei, C., Zhang, Z., Meng, X.-B., and Chen, C.L.P. (2021). As-nas: Adaptive scalable neural architecture search with reinforced evolutionary algorithm for deep learning. *IEEE Trans. Evol. Comput.* 25, 830–841.
- Patel, I., and Patel, S. (2020). An optimized deep learning model for flower classification using nas-fpn and faster r-cnn. *International Journal of Scientific & Technology Research* 9, 5308–5318.
- Liu, H., Simonyan, K., and Yang, Y. (2018). Darts: Differentiable architecture search. Preprint at arXiv. <https://doi.org/10.48550/arXiv.1806.09055>.
- Real, E., Aggarwal, A., Huang, Y., and Le, Q.V. (2019). Regularized evolution for image classifier architecture search. *Proc. AAAI Conf. Artif. Intell.* 33, 4780–4789.
- Chen, X., Xie, L., Wu, J., and Tian, Q. (2019). Progressive differentiable architecture search: Bridging the depth gap between search and evaluation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1294–1303.
- Maass, W. (1997). Networks of spiking neurons: the third generation of neural network models. *Neural Network*. 10, 1659–1671.
- Maass, W., Natschläger, T., and Markram, H. (2002). Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Comput.* 14, 2531–2560.
- Sussillo, D., and Abbott, L.F. (2009). Generating coherent patterns of activity from chaotic neural networks. *Neuron* 63, 544–557.
- Suárez, L.E., Richards, B.A., Lajoie, G., and Misis, B. (2021). Learning function from structure in neuromorphic networks. *Nat. Mach. Intell.* 3, 771–786.
- Damicelli, F., Hilgetag, C.C., and Goulas, A. (2022). Brain connectivity meets reservoir computing. *PLoS Comput. Biol.* 18, e1010639.
- Maes, A., Barahona, M., and Clopath, C. (2020). Learning spatiotemporal signals using a recurrent spiking network that discretizes time. *PLoS Comput. Biol.* 16, e1007606.
- Maass, W., and Markram, H. (2004). On the computational power of circuits of spiking neurons. *J. Comput. Syst. Sci.* 69, 593–616.
- Yamazaki, T., and Tanaka, S. (2007). The cerebellum as a liquid state machine. *Neural Network*. 20, 290–297.
- Zhou, Y., Jin, Y., and Ding, J. (2019). Evolutionary optimization of liquid state machines for robust learning. In *International Symposium on Neural Networks (Springer)*, pp. 389–398.
- Tian, S., Qu, L., Wang, L., Hu, K., Li, N., and Xu, W. (2021). A neural architecture search based framework for liquid state machine design. *Neurocomputing* 443, 174–182.
- Reynolds, J.J., Plank, J.S., and Schuman, C.D. (2019). Intelligent reservoir generation for liquid state machines using evolutionary optimization. In *2019 International Joint Conference on Neural Networks (IJCNN) (IEEE)*, pp. 1–8.
- Wijesinghe, P., Srinivasan, G., Panda, P., and Roy, K. (2019). Analysis of liquid ensembles for enhancing the performance and accuracy of liquid state machines. *Front. Neurosci.* 13, 504.
- Tang, C., Ji, J., Lin, Q., and Zhou, Y. (2022). Evolutionary neural architecture design of liquid state machine for image classification. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (IEEE)*, pp. 91–95.
- Na, B., Mok, J., Park, S., Lee, D., Choe, H., and Yoon, S. (2022). Autosnn: towards energy-efficient spiking neural networks. In

- International Conference on Machine Learning (PMLR), pp. 16253–16269.
40. Kim, Y., Li, Y., Park, H., Venkatesha, Y., and Panda, P. (2022). Neural architecture search for spiking neural networks. Preprint at arXiv. <https://doi.org/10.48550/arXiv.2201.10355>.
 41. LeCun, Y. (1998). The Mnist Database of Handwritten Digits. <http://yann.lecun.com/exdb/mnist/>.
 42. Orchard, G., Jayawant, A., Cohen, G.K., and Thakor, N. (2015). Converting static image datasets to spiking neuromorphic datasets using saccades. *Front. Neurosci.* 9, 437.
 43. Xiao, H., Rasul, K., and Vollgraf, R. (2017). Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. Preprint at arXiv. <https://doi.org/10.48550/arXiv.1708.07747>.
 44. Ivanov, V., and Michmizos, K. (2021). Increasing liquid state machine performance with edge-of-chaos dynamics organized by astrocyte-modulated plasticity. *Adv. Neural Inf. Process. Syst.* 34, 25703–25719.
 45. Watkins, C.J., and Dayan, P. (1992). Q-learning. *Mach. Learn.* 8, 279–292.
 46. Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. (2013). Playing atari with deep reinforcement learning. Preprint at arXiv. <https://doi.org/10.48550/arXiv.1312.5602>.
 47. Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.* 9, 1735–1780.
 48. Yamazaki, T., and Tanaka, S. (2007). A spiking network model for passage-of-time representation in the cerebellum. *Eur. J. Neurosci.* 26, 2279–2292.
 49. Honda, T., Yamazaki, T., Tanaka, S., Nagao, S., and Nishino, T. (2011). Stimulus-dependent state transition between synchronized oscillation and randomly repetitive burst in a model cerebellar granular layer. *PLoS Comput. Biol.* 7, e1002087.
 50. Yamazaki, T., and Nagao, S. (2012). A computational mechanism for unified gain and timing control in the cerebellum. *PLoS One* 7, e33319.
 51. Gandolfi, D., Puglisi, F.M., Boiani, G.M., Pagnoni, G., Friston, K.J., D'Angelo, E., and Mapelli, J. (2022). Emergence of associative learning in a neuromorphic inference network. *J. Neural. Eng.* 19, 036022.
 52. Yu, L., and Yu, Y. (2017). Energy-efficient neural information processing in individual neurons and neuronal networks. *J. Neurosci. Res.* 95, 2253–2266.
 53. Diehl, P.U., and Cook, M. (2015). Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Front. Comput. Neurosci.* 9, 99.
 54. Samadi, A., Lillicrap, T.P., and Tweed, D.B. (2017). Deep learning with dynamic spiking neurons and fixed feedback weights. *Neural Comput.* 29, 578–602.
 55. Mostafa, H. (2018). Supervised learning based on temporal coding in spiking neural networks. *IEEE Transact. Neural Networks Learn. Syst.* 29, 3227–3235.
 56. Mirsadeghi, M., Shalchian, M., Kheradpisheh, S.R., and Masquelier, T. (2021). Stidi-bp: Spike time displacement based error backpropagation in multilayer spiking neural networks. *Neurocomputing* 427, 131–140.
 57. O'Connor, P., and Welling, M. (2016). Deep spiking networks. Preprint at arXiv. <https://doi.org/10.48550/arXiv.1602.08323>.
 58. Zhang, M., Wang, J., Wu, J., Belatreche, A., Amornpaisannon, B., Zhang, Z., Miriyala, V.P.K., Qu, H., Chua, Y., Carlson, T.E., and Li, H. (2022). Rectified linear postsynaptic potential function for backpropagation in deep spiking neural networks. *IEEE Transact. Neural Networks Learn. Syst.* 33, 1947–1958.
 59. Shen, G., Zhao, D., and Zeng, Y. (2022). Backpropagation with biologically plausible spatiotemporal adjustment for training deep spiking neural networks. *Patterns* 3, 100522.
 60. Kaiser, J., Mostafa, H., and Neftci, E. (2020). Synaptic plasticity dynamics for deep continuous local learning (decolle). *Front. Neurosci.* 14, 424.
 61. Liu, Q., Ruan, H., Xing, D., Tang, H., and Pan, G. (2020). Effective aer object classification using segmented probability-maximization learning in spiking neural networks. *Proc. AAAI Conf. Artif. Intell.* 34, 1308–1315.
 62. Wu, Y., Deng, L., Li, G., Zhu, J., Xie, Y., and Shi, L. (2019). Direct training for spiking neural networks: Faster, larger, better. *Proc. AAAI Conf. Artif. Intell.* 33, 1311–1318.
 63. Iranmehr, E., Shouraki, S.B., Faraji, M.M., Bagheri, N., and Linares-Barranco, B. (2019). Bio-inspired evolutionary model of spiking neural networks in ionic liquid space. *Front. Neurosci.* 13, 1085.
 64. Hao, Y., Huang, X., Dong, M., and Xu, B. (2020). A biologically plausible supervised learning method for spiking neural networks using the symmetric stdp rule. *Neural Network.* 121, 387–395.
 65. Kheradpisheh, S.R., Mirsadeghi, M., and Masquelier, T. (2022). Bs4nn: binarized spiking neural networks with temporal coding and learning. *Neural Process. Lett.* 54, 1255–1273.
 66. Zhao, D., Zeng, Y., and Li, Y. (2022). Backeinn: A deep spiking neural network with adaptive self-feedback and balanced excitatory-inhibitory neurons. *Neural Network.* 154, 68–77.
 67. Rumelhart, D.E., Hinton, G.E., and Williams, R.J. (1986). Learning representations by back-propagating errors. *nature* 323, 533–536.
 68. Zeng, Y., Zhao, D., Zhao, F., Shen, G., Dong, Y., Lu, E., Zhang, Q., Sun, Y., Liang, Q., Zhao, Y., et al. (2023). BrainCog: A spiking neural network based, brain-inspired cognitive intelligence engine for brain-inspired ai and brain simulation. *Patterns* 4, 100789.
 69. Watts, D.J., and Strogatz, S.H. (1998). Collective dynamics of 'small-world' networks. *nature* 393, 440–442.
 70. Telesford, Q.K., Joyce, K.E., Hayasaka, S., Burdette, J.H., and Laurienti, P.J. (2011). The ubiquity of small-world networks. *Brain Connect.* 1, 367–375.
 71. Harris, T.E. (1963). Theodore Edward. *The Theory of Branching Processes*, 6 (Springer).
 72. Stepp, N., Plenz, D., and Srinivasa, N. (2015). Synaptic plasticity enables adaptive self-tuning critical networks. *PLoS Comput. Biol.* 11, e1004043.
 73. Hesse, J., and Gross, T. (2014). Self-organized criticality as a fundamental property of neural systems. *Front. Syst. Neurosci.* 8, 166.
 74. Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Trans. Evol. Comput.* 6, 182–197.
 75. Zbigniew, M. (1996). Genetic algorithms+ data structures= evolution programs. In *Computational Statistics* (Springer-Verlag), pp. 372–373.
 76. Fang, W., Yu, Z., Chen, Y., Masquelier, T., Huang, T., and Tian, Y. (2021). Incorporating learnable membrane time constant to enhance learning of spiking neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2661–2671.
 77. LeCun, Y., Cortes, C., and Burges, C.J. (2010). Mnist Handwritten Digit Database. <http://yann.lecun.com/exdb/mnist>.

STAR★METHODS

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Software and algorithms		
NALSM ⁴⁴	GitHub	https://github.com/combra-lab/NALSM
ELSM	This paper	https://github.com/BrainCog-X/Brain-Cog/tree/main/examples/Structure_Evolution/ELSM
Other		
MNIST ⁷⁷	Yann LeCun et al. ⁷⁷	http://yann.lecun.com/exdb/mnist/
Fashion-MNIST ⁴³	GitHub	https://github.com/zalandoresearch/fashion-mnist
NMNIST ⁴²	Orchard, G et al. ⁴²	https://www.garrickorchard.com/datasets/n-mnist

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources and reagents should be directed to and will be fulfilled by the lead contact, Yi Zeng (yi.zeng@ia.ac.cn).

Materials availability

This study did not generate new unique reagents.

Data and code availability

This paper analyzes existing, publicly available data. These accession numbers for the datasets are listed in the [key resources table](#).

All original code has been deposited at https://github.com/BrainCog-X/Brain-Cog/tree/main/examples/Structure_Evolution/ELSM and is publicly available as of the date of publication. DOIs are listed in the [key resources table](#).

Any additional information required to reanalyze the data reported in this paper is available from the [lead contact](#) upon request.

METHOD DETAILS

Spiking neural network foundation

In this paper, we adopt leaky integrate-and-fire (LIF) neurons as the basic units of signal transmission. The formula for updating their membrane potential over time is as follows:

$$\delta = \frac{I(t) - V_m(t)}{\tau} \quad (\text{Equation 4})$$

$$V_m(t+1) = (V_m(t) + \delta)(1 - S(t)) + V_r S(t) \quad (\text{Equation 5})$$

$$S(t) = \begin{cases} 1, & V_m(t) \geq V_{th} \\ 0, & V_m(t) < V_{th} \end{cases} \quad (\text{Equation 6})$$

$V_m(t+1)$ and $V_m(t)$ are the membrane potential at time $t+1$ and t , respectively. As shown in [Equation 4](#), δ is determined by the membrane potential $V_m(t)$, the magnitude of the current $I(t)$ and the membrane potential time constant τ . When the membrane potential reaches the threshold V_{th} , the membrane potential is reset to V_r at the same time as the spike is transmitted (indicated by $S(t)$ as [Equation 6](#)). According to the membrane potential $V_m(t)$ and $S(t)$ at time t , the updating law of the membrane potential at the next time is shown in [Equation 5](#).

The standard LSM model is divided into three layers: an input layer, a liquid layer formed of sparsely connected neurons, and a readout layer. Weights of readout layer are optimized by backpropagation algorithm,⁶⁷ while weights in the liquid layer are randomly set. The LIF neuron model and backpropagation algorithm of the proposed evolutionary LSM are based on BrainCog framework.⁶⁸

Algorithm 1. The neuroevolution process of ELSM

```

Initialize: Population  $P(0) = \{C_1, C_2, \dots, C_{N_c}\}$ ;
Output: Evolved individual  $C_{opt}$ ;
for  $g = 0$  to  $G_{th}$  do
  if  $g = G_{th} - 1$  then
    Accuracy = Train ( $P(g)$ , 100)
     $C_{opt} = \text{Max}(P(g), \text{Accuracy})$ 
    return  $C_{opt}$ 
  end if.
   $obj[g,0] = \text{SmallWorld}(P(g))$ 
   $obj[g,1] = \text{Criticality}(P(g), \text{data})$ 
   $P'(g) = \text{CrossoverAndMutate}(\text{Select}(P(g), obj, N_{off}))$ 
   $P(g+1) = \text{Merge}(P(g), P'(g))$ 
   $P(g+1) = \text{Select}(P(g+1), obj, N_c)$ 
end for.

```

Neuroevolution algorithm

Randomly initialized liquid layer connectivity will be evolved to emerge more brain-inspired structures and dynamics. The whole neuroevolution process is presented as Algorithm 1.

Initialization

In a population of N_c individuals to be initialized, each individual represents the liquid connection pattern C of an N liquid neurons reservoir. We use a binary encoding method, each gene c_{ij} has two values of 0 or 1, indicating whether there is a synapse i connecting the presynaptic neuron and a post-synaptic neuron j ($0 < i, j < N$). Each individual is limited to be sparse and has only $N * N * \rho_{init}$ synaptic connections inside (the initial liquid density is recorded as ρ_{init}).

A random matrix R is generated, and the values of all elements r_{ij} in R are between 0 and 1. The Boolean matrix C is obtained by calculating the result of $R < \rho_{init}$, as Equation 7:

$$c_{ij} = \begin{cases} 1, & \text{if } r_{ij} \leq \rho_{init} \\ 0, & \text{if } r_{ij} > \rho_{init} \end{cases} \quad (\text{Equation 7})$$

Evaluation

Identifying small-world topologies

Small-world networks are described as exhibiting two properties⁶⁹: a high clustering coefficient and a short average path length. Local short paths between most nodes with hubs induce highly connected sub-networks and a few long-distance connections, enabling efficient information transmission in the brain. To quantify these small-world characteristics, the calculation method we adopt,⁷⁰ referred to as the small-world coefficient, is as follows:

$$\lambda = \frac{H}{L} \quad (\text{Equation 8})$$

The clustering coefficient and the short path length between nodes are represented by H and L , respectively. The computation of the clustering coefficient for a single node is given by Equation 9, where o_i denotes the degree of neuron i . Here, a pair is defined as two edges that share neuron i as a common endpoint. Let e_i represent the number of all pairs of edges with i as the intermediate node. If two edges that both pass through neuron i are called a pair, e_i is the number of all pairs of edges with i as the intermediate node. The overall clustering coefficient is then calculated as the average clustering coefficient across all neurons, given that there are N neurons in the liquid layer.

$$h_i = \frac{2e_i}{o_i(o_i - 1)} \quad (\text{Equation 9})$$

$$H = \frac{\sum_i^N h_i}{N} \quad (\text{Equation 10})$$

The shortest path length of the network is calculated as Equation 11, where d_{st} represents the shortest path length between s neuron and t neuron ($d_{st} = 0$ if the path does not exist). V is the liquid neuron set.

$$L = \sum_{s,t \in V} \frac{d_{st}}{N(N-1)} \quad (\text{Equation 11})$$

Identifying criticality

A commonly used concept to measure the critical state of the nervous system is called branching ratio is derived from the branching process theory⁷¹ and describes the spatiotemporal cascade activity of the cerebral cortex in homeostasis. Specifically, the local branching ratio $\mu_i(t)$ at time t reflects the propensity of neuron i to either increase or decrease its activity in response to spikes transmitted within the liquid layer. This parameter is defined as follows.⁷²

$$\mu_i(t) = \frac{\sum_j^N \sum_{l=t+\phi+1}^{t+\phi+\Delta} m_j(l) c_{ij}}{\sum_j^N \sum_{l=t-\phi-\Delta}^{t-\phi-1} m_j(l) c_{ji}} \quad (\text{Equation 12})$$

$$\mu(t) = \frac{\sum_i^N m_i(t) \mu_i(t)}{\sum_i^N m_i(t)} \quad (\text{Equation 13})$$

$$\mu = \frac{\sum_t^T \mu(t)}{T} \quad (\text{Equation 14})$$

Consider $t = 1, 2, 3, \dots, T$ as discrete timesteps. c_{ij} denotes the presence or absence of a synapse connecting the presynaptic neuron i to the postsynaptic neuron j , taking a value of 1 (if a synapse is present) or 0 (if not). The term $m_i(t)$ represents the firing status of neuron i at time t , adopting a binary value: 0 for non-firing and 1 for firing. Therefore, Equation 14 is the ratio of the sum of the postsynaptic neuron spikes to the sum of the presynaptic neuron spikes during the simulated time T . Studies have shown that the closer the value of μ is to 1, the closer the dynamics of the network are to the critical state.^{44,72,73} Therefore, the quantitative criticality is calculated as follows:

$$\mu = |\mu - 1| \quad (\text{Equation 15})$$

Given more than one fitness function, we take into account both the physical topology characteristics and dynamic changes of the network when selecting superior individuals as the parents of the next generation. Referring to the fitness function proposed in Equation 8 and Equation 14, a larger small-world coefficient, λ , indicates a model structure more reflective of small-world characteristics. The smaller μ is, the closer the model dynamics are to the critical state. Therefore, this multi-objective optimization problem (MOP) can be described as $F : \Omega \rightarrow \mathbb{R}$:

$$\operatorname{argmin}_{C \in \Omega} F(C) = \{f_1(C), f_2(C)\} \quad (\text{Equation 16})$$

$$\text{s.t. } \rho_1 \leq \rho(C) \leq \rho_2$$

$\rho(C)$ is the density of the liquid layer, defined as the ratio of the number of liquid layer connections $N * N$. To keep the liquid density $\rho(C)$ stable during the evolution, the range of which is limited to ρ_1 and ρ_2 . The first evolutionary goal $f_1(C)$ is to maximize the small-world coefficient:

$$f_1(C) = \min(-\lambda(C)) \quad (\text{Equation 17})$$

where λ is calculated as Equation 8, measuring the static LSM topology properties.

The second evolutionary goal $f_2(C)$ is to minimize the criticality coefficient, which is formulated by:

$$f_2(C) = \min \mu(C) \quad (\text{Equation 18})$$

where μ is calculated as Equation 15, measuring the dynamic LSM topology properties.

Selection

The elitism approach and nondominated sorting strategy of the NSGA-II algorithm⁷⁴ are used here to generate mating pools of size N_{offs} and next-generation individuals. We employ the elitism approach and the nondominated sorting strategy from the NSGA-II algorithm⁷⁴ to create mating pools of size N_{offs} and to generate the subsequent generation of individuals.

Crossover

Consider two individuals, C_1 and C_2 , who are crossed to produce offspring C_3 and C_4 . We employ the k -point crossover operator r_c , selecting k genes as the crossover points:

$$C_{a_1 b_1}, C_{a_2 b_2}, \dots, C_{a_k b_k} \quad (\text{Equation 19})$$

where $0 < a_k, b_k < N$. Based on the chosen crossover points, each gene is divided into $k + 1$ segments:

$$\frac{(C_{0,0}, C_{a_1 b_1})}{S_1}, \frac{(C_{a_1 b_1+1}, C_{a_2 b_2})}{S_2}, \dots, \frac{(C_{a_k b_k+1}, C_{NN})}{S_{k+1}} \quad (\text{Equation 20})$$

We get the matrix E and D where $E_{s_1}, E_{s_3}, \dots, D_{s_2}, D_{s_4}, \dots$ are set to 1. The remaining elements in E and D are set to 0. Therefore:

$$C_3 = C_1 * E + C_2 * D \quad (\text{Equation 21})$$

$$C_4 = C_1 * D + C_2 * E \quad (\text{Equation 22})$$

Mutation

Assuming that C_1 is mutated into C_5 , flip bit mutation⁷⁵ is performed on n_m genes of C_1 . The mutation probability is set as m_{rate} : a random number m_{rand} is generated for each offspring after crossover, if $m_{rand} < m_{rate}$, the mutation is accepted. Each mutation is to select a gene c_{ij} in C_1 for inversion:

$$c_{ij} = \neg c_{ij} \quad (\text{Equation 23})$$

Parents $P(g)$ and population formed by crossover and mutation $P'(g)$ are merged, and the selection operator is applied to generate the next generation $P(g + 1)$.

Next generation

The above neuroevolution process of initialization, evaluation, selection, crossover, and mutation is repeated for G_{th} generations. Ultimately, every individual in the final generation undergoes training for 100 epochs, and the one with the highest classification accuracy, denoted as C_{opt} , is chosen as the outcome of the evolutionary process.

Static and neuromorphic image processing

MNIST

The handwritten dataset MNIST is one of the classic machine learning datasets, comprising 70,000 grayscale images of handwritten digits 0–9. It is divided into 60,000 training examples and 10,000 testing examples. The size of each image is $28 * 28$ pixels.

Fashion-MNIST

The Fashion-MNIST dataset consists of 70,000 grayscale images, with 60,000 for training and 10,000 for testing, distributed across ten categories. Each sample is a $28 * 28$ grayscale image.

NMNIST

NMNIST is a neuromorphic version of MNIST converted to MNIST images by an actuated pan-tilt camera platform. After 300 ms of signal acquisition, the dataset includes 60,000 training images and 10,000 test images, each of size $34 * 34$ pixels and with 2 channels.

For the static MNIST and Fashion-MNIST datasets, the inputs to the model are grayscale images with normalized pixel values ranging from 0 (black) to 1 (white). For the neuromorphic NMNIST, an extra preprocessing ensemble method⁷⁶ is adapted to convert the event stream into a frame stream, which is then fed into the model for classification. The two channels are combined into one by summing.

Training procedure and techniques

At each time step, a randomly generated weight matrix transforms the normalized grayscale image into the activity of a liquid layer consisting of N LIF neurons. This layer not only receives external input but also processes feedback from its previous firing pattern (0 indicates no firing, while 1 indicates firing) through a liquid-liquid weight matrix of size $N * N$. Subsequently, the firing pattern of the liquid layer is transmitted to a readout layer via a fully connected $N * 10$ weight matrix. After T time steps, the output of the model is the average value at each time step of the output layer's cumulative spikes. The backpropagation algorithm is employed to optimize the weights between the liquid and readout layers.

QUANTIFICATION AND STATISTICAL ANALYSIS

Use Python for statistical analysis. The results in Tables 3 and 4 are the average values of multiple runs and their error ranges.