



OPEN ACCESS

EDITED BY

Jonathan Mapelli,
University of Modena and Reggio Emilia, Italy

REVIEWED BY

Maryam Parsa,
George Mason University, United States
Zdenka Kuncic,
The University of Sydney, Australia
Qi Xu,
Dalian University of Technology, China

*CORRESPONDENCE

Jiulin Du

✉ forestdu@ion.ac.cn

Tielin Zhang

✉ tielin.zhang@ia.ac.cn

Bo Xu

✉ xubo@ia.ac.cn

†These authors have contributed equally to this work

RECEIVED 20 October 2023

ACCEPTED 27 March 2024

PUBLISHED 16 April 2024

CITATION

Wang Y, Wang Y, Zhang X, Du J, Zhang T and Xu B (2024) Brain topology improved spiking neural network for efficient reinforcement learning of continuous control.
Front. Neurosci. 18:1325062.
doi: 10.3389/fnins.2024.1325062

COPYRIGHT

© 2024 Wang, Wang, Zhang, Du, Zhang and Xu. This is an open-access article distributed under the terms of the [Creative Commons Attribution License \(CC BY\)](https://creativecommons.org/licenses/by/4.0/). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

Brain topology improved spiking neural network for efficient reinforcement learning of continuous control

Yongjian Wang^{1,2†}, Yansong Wang^{3,4†}, Xinhe Zhang¹, Jiulin Du^{3,5*}, Tielin Zhang^{1,2,3*} and Bo Xu^{1,2,3,4*}

¹Institute of Automation, Chinese Academy of Sciences, Beijing, China, ²School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China, ³Institute of Neuroscience, State Key Laboratory of Neuroscience, Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences, Shanghai, China, ⁴School of Future Technology, University of Chinese Academy of Sciences, Beijing, China, ⁵School of Life Science and Technology, ShanghaiTech University, Shanghai, China

The brain topology highly reflects the complex cognitive functions of the biological brain after million-years of evolution. Learning from these biological topologies is a smarter and easier way to achieve brain-like intelligence with features of efficiency, robustness, and flexibility. Here we proposed a brain topology-improved spiking neural network (BT-SNN) for efficient reinforcement learning. First, hundreds of biological topologies are generated and selected as subsets of the Allen mouse brain topology with the help of the Tanimoto hierarchical clustering algorithm, which has been widely used in analyzing key features of the brain connectome. Second, a few biological constraints are used to filter out three key topology candidates, including but not limited to the proportion of node functions (e.g., sensation, memory, and motor types) and network sparsity. Third, the network topology is integrated with the hybrid numerical solver-improved leaky-integrated and fire neurons. Fourth, the algorithm is then tuned with an evolutionary algorithm named adaptive random search instead of backpropagation to guide synaptic modifications without affecting raw key features of the topology. Fifth, under the test of four animal-survival-like RL tasks (i.e., dynamic controlling in Mujoco), the BT-SNN can achieve higher scores than not only counterpart SNN using random topology but also some classical ANNs (i.e., long-short-term memory and multi-layer perception). This result indicates that the research effort of incorporating biological topology and evolutionary learning rules has much in store for the future.

KEYWORDS

spiking neural network, brain topology, hierarchical clustering, reinforcement learning, neuromorphic computing

1 Introduction

The mammalian brains, ranging from the simpler mouse brain to the more complex monkey and human brains, share some key functional circuits or brain regions to support different cognitive functions, including but not limited to sensation, memory, and decision-making. The brain network has been widely discussed in recent decades for its complexity (Luo, 2021). For example, the mouse brain network connectome at various scales has been largely examined, including the neuron-scale imaging of a cubic millimeter

of mouse cortex (Yin et al., 2020), the mesoscale connectome of the entire mouse brain (Oh et al., 2014), and the macroscale network-motif topology analysis (Zhang et al., 2017). Many key topologies related to cognitive functions have been identified with the help of new optical or even electron microscopy, along with the well-designed experimental paradigms (Luo, 2021).

The mouse brain contains at least 213 brain regions, and the sparseness of the entire brain is <36% (Oh et al., 2014), which makes it a good network reference to guide the design of spiking neural networks (SNNs) in especially neuromorphic computing manners (Maass, 1997). Until now, many key biological features have been incorporated into SNNs, including but not limited to neuronal heterogeneity, feed-forward or recurrent connections, and multiscale plasticity (Izhikevich, 2004; Zenke and Gerstner, 2017).

Different from artificial neural networks (ANNs), whereby single artificial backpropagation (BP) is used for network learning (Lillicrap et al., 2020), the learning algorithms in SNNs are various, such as plasticity-based algorithms [e.g., the spike-timing-dependent plasticity (Dan and Poo, 2004), short-term plasticity (Zhang et al., 2018), self-backpropagation (Zhang et al., 2021a)], gradient-based algorithms [e.g., reward propagation (Zhang et al., 2021b), surrogate gradient (Cramer et al., 2022)], and the evolutionary algorithms (Bäck and Schwefel, 1993).

However, there is a serious conflict between biological topology and corresponding learning rules since a predefined topology will usually be revised or destroyed by gradient or plasticity-based algorithms (Bellec et al., 2020). Here we run further by considering some evolutionary algorithms, which have also been verified efficient in tuning SNNs for their simplicity and efficiency, and what's most important, resolving this conflict problem by selectively pruning some trivial branches in network topology during learning.

In this paper, the main goal is to incorporate some subsets of brain topology (BT) into SNNs, and then train them using an evolutionary algorithm during reinforcement learning (RL) tasks. The detailed process and contribution of this paper can be concluded in the following parts:

- Some important subsets of network topology are filtered out from the source brain topology by considering some biological constraints. As a result, three key BTs have been generated from the mesoscale connectome of Allen mouse brain atlas (Oh et al., 2014) by the Tanimoto hierarchical clustering algorithm. Different types of BTs are further analyzed by the distribution of the three-node network motif to answer why the topology might work from the perspective of intuitive biological analysis.
- The BT-improved SNNs (BT-SNNs) are designed by incorporating different types of BTs and SNNs using numerical solver-improved leaky integrate-and-fire neurons, whereby an evolutionary-type learning algorithm is used to efficiently guide the synaptic modification without affecting key network topology.
- Four benchmark RL tasks in OpenAI Mujoco environment (Brockman et al., 2016), also with some key features of animal survival, are used to test the performance of the

proposed algorithm, including MountainCar-v2 (a car learns to stop at a mountain), Half-Cheetah-v2 (a dog that learns to run), Humanoid-v2 (a human that learns to run), and HumanoidStandup-v2 (a human that learns to stand up). The BT-SNNs have reached a higher average reward than their counterpart algorithms, including SNNs using random topologies and classical ANNs, such as long-short-term memory (LSTM) and multi-layer perception (MLP).

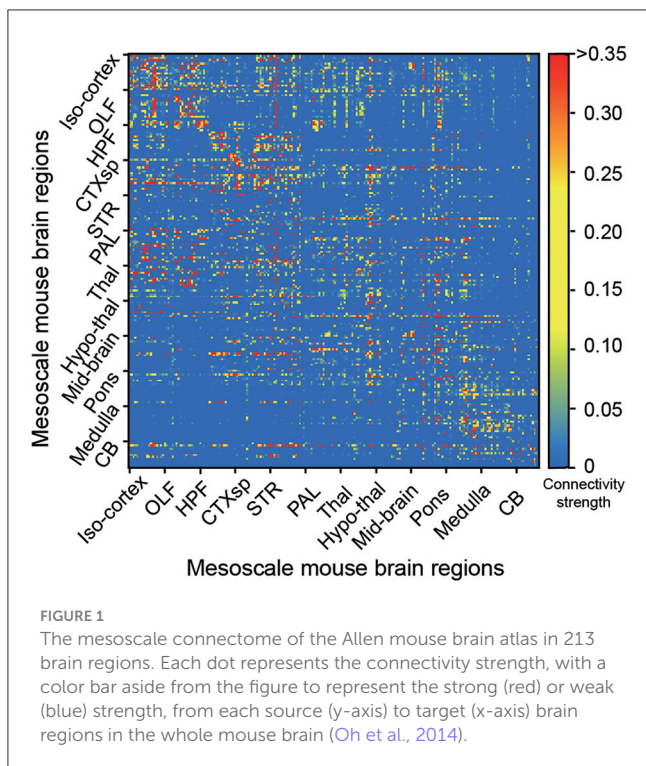
2 Related works

Borrowing key topology knowledge from different animal brains is challenging, caused by raw data analysis and topology-informed computation. For the network topology, a copy-and-paste approach, i.e., copying the structural synaptic connectivity map of a mammalian brain and pasting it to a three-dimensional network in solid-state memories of neuromorphic engineering, has been proposed with the spirit of reverse-engineering the brain (Ham et al., 2021). Some distilling algorithms try to make an abstraction of a teacher network to a much smaller student network, but with less computational cost and comparable performance (Han et al., 2015; Hinton et al., 2015). The biological topology-focused algorithm by using a sub-graph sparse network to replace a previous global dense one, named as lottery ticket hypothesis, has been proposed to achieve comparable or even higher performance (Frankle and Carbin, 2018). Some researchers believe that the network topology and synaptic weights are two independent dimensions. Hence, they focus more on learning synaptic weights and leave the topology fixed with feed-forward, recurrent, or some scale of sparseness topology. A new study focuses on these two aspects both by learning weights and topology simultaneously toward a much more efficient algorithm (Han et al., 2015). Similar to it, a biological network using *C. elegans* topology has also been proposed to achieve higher scores in RL paradigms than those using random topology, which to some extent, indicates the efficiency of the biological topology in network learning (Hasani et al., 2020).

SNNs frequently underperform relative to ANNs in handling complex tasks (Deng et al., 2020). There are studies that apply deep learning, gradient descent, and backpropagation to biologically reasonable SNNs (Eshraghian et al., 2023). There are also studies using neural pruning methods to implement adaptive sparse learning SNN (Li et al., 2024). Some studies using knowledge distillation and connection pruning methods to dynamically optimize synaptic connections in SNN (Xu et al., 2023).

Some studies have instantiated Biological Neuronal Networks (BNNs) into Recurrent Neural Networks (RNNs) for network structure exploration (Goulas et al., 2021). Some people also combine the feature learning ability of CNN with the cognitive ability of SNN to improve the robustness of SNN (Xu et al., 2018), and some other works have emulated the brain's synaptic connections and dynamic behaviors through Nanowire Networks (NWNs) to facilitate learning and memory functions (Loeffler et al., 2023).

For the learning algorithms under RL tasks, a multiscale dynamic coding algorithm has been proposed to improve an



SNN on OpenAI Mujoco tasks (Zhang et al., 2022). Besides, a traditional continuous-time differential learning algorithm has been proposed for RL tasks containing continuous dynamics (Doya, 2000). A hybrid learning framework, incorporating SNNs for energy-efficient mapless navigation, has been proposed and applied on the neuromorphic hardware (Tang et al., 2020).

However, most of these proposed algorithms overlook the importance of network topology in learning, especially the exploration of inter-cluster topological relationships within brain regions, and neglect some key features by following gradient-based or plasticity-based algorithms. The further incorporation of network topology, especially those related to cognitive functions of sensation, motor, and reward learning, can exhibit more power on animal-survival-like RL tasks. It is becoming an important consensus that the topology is at least as important as synaptic weights to the network performance. Here we employ a hierarchical clustering algorithm to generate some network topology from the Allen mouse brain atlas first and then incorporate a standard evolutionary algorithm to guide the synaptic modification without using traditional gradient and plasticity-based rules.

3 Methods

3.1 The raw brain topology in Allen mouse brain atlas

Analyzing a set of biological topologies is usually the first step to support the following network-topology simulation in neural networks. Here we select the mesoscale Allen mouse brain atlas provided by the Allen Institute for Brain Science (Oh et al., 2014). It contains publicly available resources on brain region

morphology (e.g., the common coordinate framework, CCF) and mesoscale network topology at sub-brain region scale which covers bidirectional topology in 213 brain regions.

A 3D model containing at least 213 brain regions is first constructed based on the mouse brain CCF for visualization, analysis, and functional simulation (see Section 4 for more details). The 213 brain regions are separated into three subgroups: the sensation group, including but not limited to the primary somatosensory area, primary visual area, primary auditory area, and accessory olfactory bulb; the motor group, including but not limited to the primary motor area, dentate nucleus, and motor nucleus of trigeminal; the left brain regions except the previously mentioned two groups but related to some key cognitive functions, including but not limited to the hippocampus for memory, basal ganglia for reward learning.

The bidirectional connectivity of the whole Allen mouse brain is shown in Figure 1, containing the mapping connectivity from a source brain region to a target region in the total 213 brain regions (Oh et al., 2014). It is easier to find that the connectivity matrix is much sparser, which is considered the key feature of biological structures compared to those in recurrent neural networks.

3.2 The Tanimoto hierarchical clustering

The connectivity matrix of the mouse brain atlas (213×213 size) is clustered into sub-clusters for an easier simulation. The Tanimoto clustering algorithm is selected as the main method to group all connections (Ahn et al., 2010; Kalinka and Tomancak, 2011), which could be concluded as the following Equation 1, where $S(e_{i,k}, e_{j,k})$ represents the similarity between links $e_{i,k}$ and $e_{j,k}$ that share a node k :

$$S(e_{i,k}, e_{j,k}) = \frac{\mathbf{a}_i \cdot \mathbf{a}_j}{|\mathbf{a}_i|^2 + |\mathbf{a}_j|^2 - \mathbf{a}_i \cdot \mathbf{a}_j}, \quad (1)$$

where the vector $\mathbf{a}_i = (\tilde{A}_{i1}, \dots, \tilde{A}_{iN})$ describes the connectivity strength between the node i and its first-order neighborhoods, and the \tilde{A}_{ij} is set as the following Equation 2:

$$\tilde{A}_{ij} = \frac{1}{k_i} \sum_{i' \in n(i)} w_{i,i'} \delta_{i,j} + w_{i,j}, \quad (2)$$

where w_{ij} is the connectivity strength for edge e_{ij} , $n(i)$ is a neighborhood set defined as $\{j | w_{ij} > 0\}$, $k_i = |n(i)|$, and $\delta_{ij} = 1$ when $i = j$ or else $\delta_{ij} = 0$. Then the dendrogram can be cut at a large partition density height to get link and node clusters. The detailed Tanimoto hierarchical clustering algorithm can be found at Algorithm 1.

After the Tanimoto clustering, a community of sub-connectivity matrices in 213 brain regions can be hierarchically separated at a desired partition density. The partition density D can be calculated as the following Equation 3.

$$D = \frac{2}{M} \sum_i \frac{m_i + 1 - n_i}{(n_i - 2)(n_i - 1)}, \quad (3)$$

```

Input: All connections and Tanimoto coefficient  $S$ ;
Assign each connection to its own cluster;
Sort  $S$  from large to small as  $S_{(L_1, L_1)}, \dots, S_{(L_N, L_N)}$ ;
 $i =$  the number of clusters,  $j = 1$ ;
for  $i > 1$  and  $j < N+1$  do
  Merge the clusters which contain connection  $L_j$  and
   $L_j$ ;
  Encode  $S_{(L_j, L_j)}$  as the height;
  Store the process in the dendrogram;
   $j = j + 1$ ,  $i =$  the number of clusters;
end for
Output: The dendrogram.

```

Algorithm 1. Tanimoto hierarchical clustering.

where m_i is defined as the number of connections giving a specific cluster i ; n_i is defined as the number of nodes in the same cluster i , and M is the number of connections for the whole network which contains all clusters. The D indicates the density of connections, with its value adjusted relative to the theoretical maximum and minimum connection scenarios within the network. This adjustment allows for a standardized comparison of connection densities across different network configurations.

The community connectedness of cluster i , as defined by Equation 4, quantifies the degree of connection between cluster i and other clusters, reflecting a comparison of external connections to other clusters relative to internal connections within the cluster itself. For clusters containing a large number of brain regions, this value tends to be lower (e.g., 10–20), indicating a higher proportion of internal connections. If the value is too low (<10), there may be artifacts that interfere with the value of statistical research.

$$C_i = \frac{n_i(n_i - 1)e_b(i)}{2e_w(i)n_i\hat{d}}, \quad (4)$$

where n_i is defined as the number of nodes within the cluster i ; $e_b(i)$ is defined as the number of connections between cluster i and its neighborhood clusters; $e_w(i)$ is defined as the number of connections within the cluster i . \hat{d} is defined as the whole-network average degree.

3.3 The brain topology

The brain-region clusters are generated from the 213 brain regions of the Allen mouse brain atlas by the Tanimoto clustering algorithms first, and then biological experts make a selection by considering some biological constraints. The 71 sub-clusters after clustering are concentrated in three intervals, <10 nodes, 30 to 60 nodes, and greater than 100 nodes. Taking into account the clustering principle and the artifacts present in the experiment, the interval of 30 to 60 nodes is the most preferred for brain topology experiments. Considering the subsequent tasks such as Mujoco, the key clusters we study need to have sensory, memory and motor functions. The detailed procedure of brain-region clustering contains five steps:

- The Tanimoto clustering algorithm is used to make a hierarchical clustering of these 213 brain regions. Different brain regions can be generated at different clustering height levels, as shown in Figure 2A.
- The selection of clustering height is inducted by biological experts. A smaller or bigger clustering height will cause the partition density to be too small or too big, representing allocating all brain regions into the same cluster or an independently different one, respectively. Then the clustering height (sparseness) is set as 0.8 and get 71 clusters, as shown in Figure 2B and Equation 3.
- The proper density is verified by visualizing the participation of each brain region in each cluster generated in the previous step, as shown in Figure 2C.
- As can be seen from Figure 2C, according to the sparsity designed in Figure 2B, the design of our 71 cluster factor Tanimoto clustering method shows connectivity at different scales. There are two conditions for sub-clusters to be selected for further processing: first, the connectivity is in the appropriate range, that is, the community connectedness is 10–40 in Figure 2D; second, it is biologically reasonable, that is, the cluster includes brain areas with sensory, memory, and motor functions.
- Some key clusters (i.e., three ones after analysis) with a different number of brain regions (i.e., the cluster with the index of 31, 46, and 49) are generated and named the NET-31, NET-46, and NET-49.

During these five steps, the brain-region clusters can be automatically generated as candidate clusters, which are efficient without a time-consuming manual summary, which is important for the efficient network topology generation at the whole mouse-brain scale.

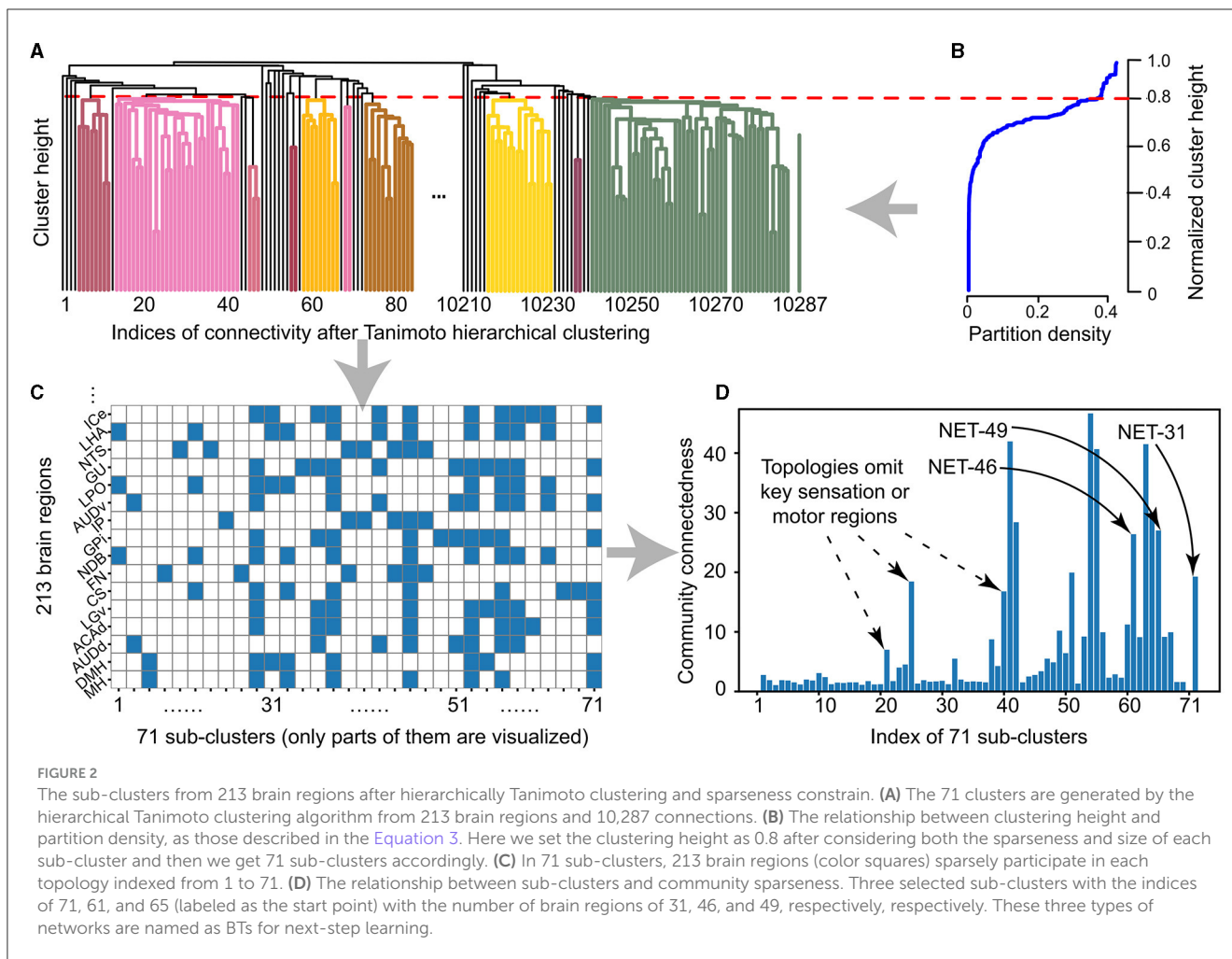
3.4 The biologically-plausible SNNs

Both the leaky-integration neuron (LI) (Hasani et al., 2020) and leaky integrate-and-fire (LIF) neuron (Liu and Wang, 2001) with excitatory and inhibitory types are used as the basic neuron model for the next-step simulation of SNNs at the network scale. The design of the LI model is represented as the following Equation 5.

$$\begin{cases} \dot{V}_{i,t} = [I_L + \sum \hat{I}_{C,t} + \sum I_{C,t}] / C_m \\ I_{L,t} = \omega_L (E_L - V_{post,t}) \\ I_{C,t} = \omega_C (E_C - V_{post,t}) g_t \\ g_t = 1 / [1 + \exp(-\sigma (V_{pre,t} - \mu))] \end{cases}, \quad (5)$$

where C_m is the membrane capacitance of the neuron, $I_{C,t}$ and $I_{L,t}$ are the input currents of the chemical and leakage channels, respectively. E_C and E_L are the corresponding reversal potentials. $V_{post,t}$ and $V_{pre,t}$ are the membrane potentials of post-synapses and pre-synapses, respectively. g_t is the dynamic conductance of the membrane, defining whether a synapse is excitatory or inhibitory by E_C . ω_C and ω_L are the conductance in chemical and leakage channels, respectively.

The LI neuron can realize the adaptive calculation of the ordinary differential equation (ODE) and has a strong ability to



model the time series reaching a goal at any time step. Besides LI neurons which play key roles in the inner dynamics in the hidden layers of networks, we also introduce the sensory and motor neurons in the input and output layers, respectively, during the interaction with the environment.

$$\begin{cases} \dot{V}_{i,t} = (I_L + \sum \hat{I}_{C,t} + \sum I_{C,t}) (1 - S_{i,t}) / C_m \\ S_{i,t} = V_{i,t} > V_{th} \end{cases} \quad (6)$$

A hybrid numerical solver (Press et al., 2007) is used and combines with explicit Euler's discretization (Lechner et al., 2019), similar to that in Hasani et al. (2020), where a fixed-step solver is used to calculate ODE, and at each time step Δ_t , our approach complexity is around $O(|N_n| + |N_s|)$, where N_n is the number of neurons, and the N_s is the number of synapses, as shown in Equation 6.

After the membrane potential $V_{i,t}$ reaches the firing threshold V_{th} , the spiking flag $S_{i,t}$ is set as true, which will reset the update of the membrane potential $V_{i,t}$ by multiplying $1 - S_{i,t}$, with the spirit of biological leaky integrate-and-fire.

3.5 BT improved SNN

Biological experts group the 213 brain regions in the Allen mouse brain atlas into three subgroups. The first group is the input layer containing the sensation-related brain regions, e.g.,

the primary somatosensory and visual areas. The second group is the hidden layer containing the cognitive-function-related brain regions, e.g., the hippocampus and basal ganglia. The third group is the output layer containing the motor-related brain regions, e.g., the primary motor area and trigeminal motor nucleus. We also annotate the biological functions of the brain regions at each level of the clusters of interest (see Section 4 for more details), which directly link the biological regions to network layers.

Besides the topology with 213 brain regions (which can be considered the whole brain region, NET-213), different types of network topology with different numbers of brain regions are selected by biological experts for the next-step simulation. Using the configuration of the 0.8 sparseness during the hierarchical clustering level (Figure 2B), we select brain regions with the index of NET-31, NET-46, and NET-49 in all 71 sub-clusters (Figure 2C), where each number represents the number of brain regions in the selected topology. These clusters all cover sensation, cognitive function, and motor brain regions without omitting the key transfer region in a network (Figure 2D). The detailed brain regions in NET-46 will be further introduced in Section 4.

3.6 The evolutionary-based learning algorithm

The SNN with biological topology (i.e., connected to each node with the biological network NET-213, NET-31, NET-46, and

NET-49) can be tuned by many learning algorithms. Here we select the evolutionary-based algorithms for their topology-friendly advantages, i.e., the adaptive random search algorithm (ARS) (Hasani et al., 2020). We find it can also get around some serious problems in recurrent neural networks during reinforcement learning, including but not limited to gradient scaling problems and long-term dependence problems (Mania et al., 2018).

In this paper, we optimize the ARS algorithm and use it in RL tasks, whereby the agent learns to make decisions after observing the current state in an environment and then receives a timely or delayed reward. The fitness function is designed to collect these rewards and guide the direction of the random search. At the beginning of network learning, the agent makes random decisions for exploration, and a good decision for a lower fitness function will be kept by saving the current parameters and focusing more on the exploitation. The search-based algorithm ARS can train a network by repeating two training strategies until convergence. First, expected values are obtained by perturbation network parameters. Then the adaptive search algorithm calculates the distance between expected values and fitness function and uses it further to guide the search space for a smaller distance. Objectively, the ARS algorithm requires a certain amount of effort to identify and select potentially useful network structures, and the network learning convergence using ARS is slower than the standard gradient-based algorithms, where the desired gradient is calculated by re-sampling the dataset in a memory buffer. However, the memory buffer makes at least two serious problems: (1) the extremely high storage space; (2) the re-sampling of samples collected from the exploration is inefficient. Hence, the ARS can save computational costs without considering the storage space and re-sampling than the standard gradient-based algorithms, which indicates it is more suitable for online and neuromorphic computation.

3.7 The analysis of BTs using network motif

The NET-31, NET-46, and NET-49 contain many brain regions (with input, hidden, and output areas) and sparse connections. Here we use 2D and 3D visualization methods to highlight the main difference between these three network topologies.

For the 2D visualization, as shown in Figure 3A, the NET-31 contains six input regions, 23 hidden regions, and two motor regions. A total of 450 connections are plotted, containing 361 excitability and 89 inhibitory connections. As shown in Figure 3B, the NET-46 contains eight input regions, 36 hidden regions, and two motor regions. A total of 802 connections are plotted, containing 575 excitability and 227 inhibitory connections. As shown in Figure 3C, the NET-49 contains ten input regions, 37 hidden regions, and two motor regions. A total of 904 connections are plotted, containing 713 excitability connections and 191 inhibitory connections. The definition of the ratio of excitatory neurons is 70%, the same as that found in the brain cortex (Wildenberg et al., 2021).

For the 3D visualization of three networks (i.e., NET-31, NET-46, and NET-49), the connections of different brain regions from input, hidden, and motor areas are given under a background of the mouse-brain CCF. With the help of biological experts, the input

regions belong to the occipital lobe, the output regions belong to the parietal lobe, and the hidden regions are everywhere in the brain for the complex information processing, consistent with the biological functions, as shown in Figures 3D–F. For ease of visualization, the connections with connectivity strength lower than 0.05 in three networks are omitted. For example, only 207 excitatory and 60 inhibitory connections are visualized in NET-31.

The 3-node network motif (Milo et al., 2010) has been widely used to analyze the dynamic properties (Prill et al., 2005) and biological network features (Sporns and Kotter, 2004). Here we also use the 3-node network motif to analyze the connection distribution feature of the NET-31, NET-46, and NET-49. As shown in Figure 3, we use the “credible frequency” (the product of the occurrence frequency and $1 - P$) instead of the pure frequency to avoid the influence of some random features. Here P is the P -value of each motif in the selected network compared to the 1,000 randomly generated networks of the same size. Each generated network is sampled from a uniformly random distribution. The smaller P -value, the less likely a random network will have the same network features as a biological one.

In all calculated network motifs, we want to highlight the motif-5 distribution (a type of cross-layer connection). The motif distributions for the three topologies share some common features, such as the motif-1, 2, 6, 7, 10, 11, and 13 are relatively higher than other motifs. The motif-5 and motif-9 are the main two differences that might be the main differences of functional circuits in these three topologies. Further analysis will be given in the performance comparison of these three networks.

4 Experiments

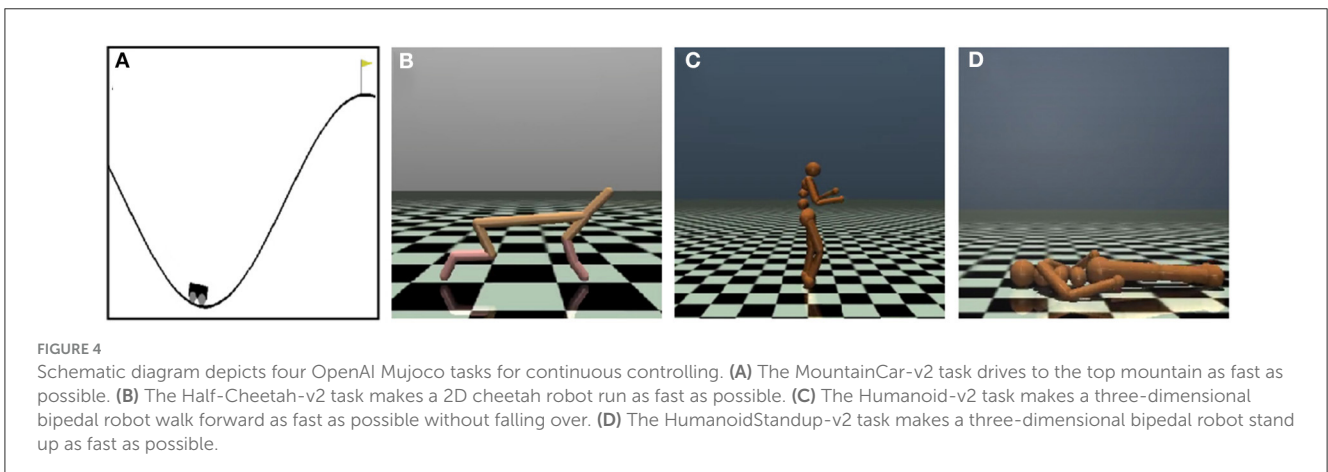
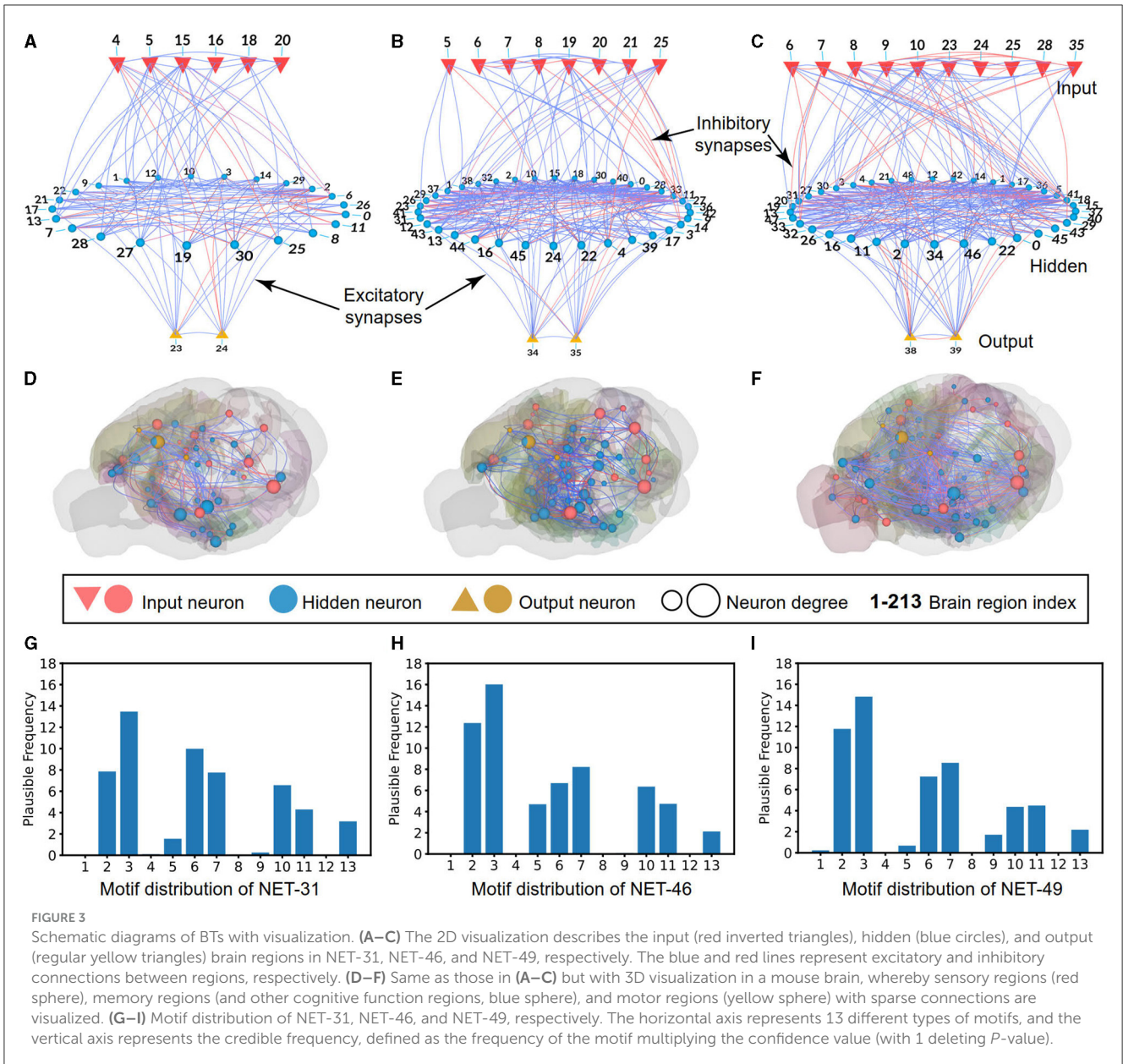
4.1 Introduction of tasks and implement details

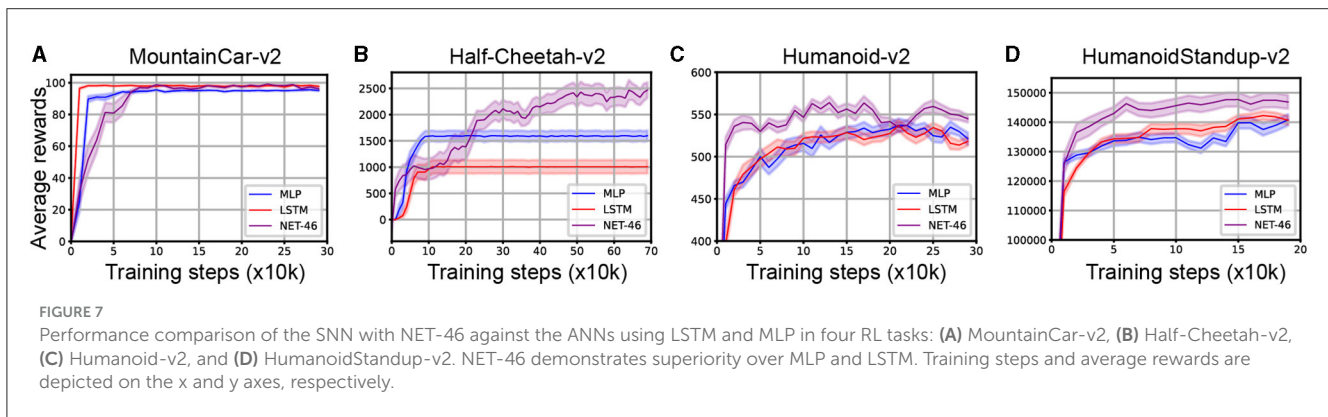
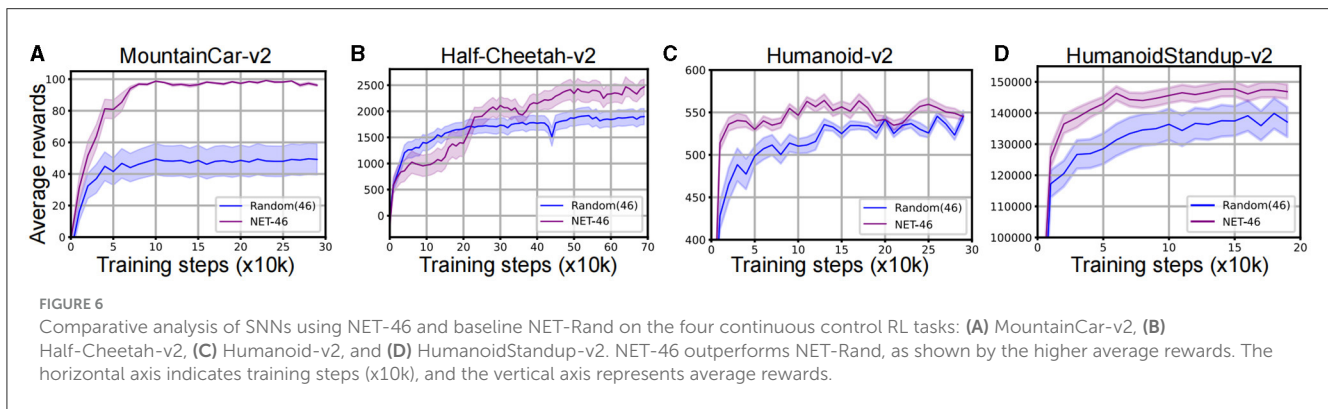
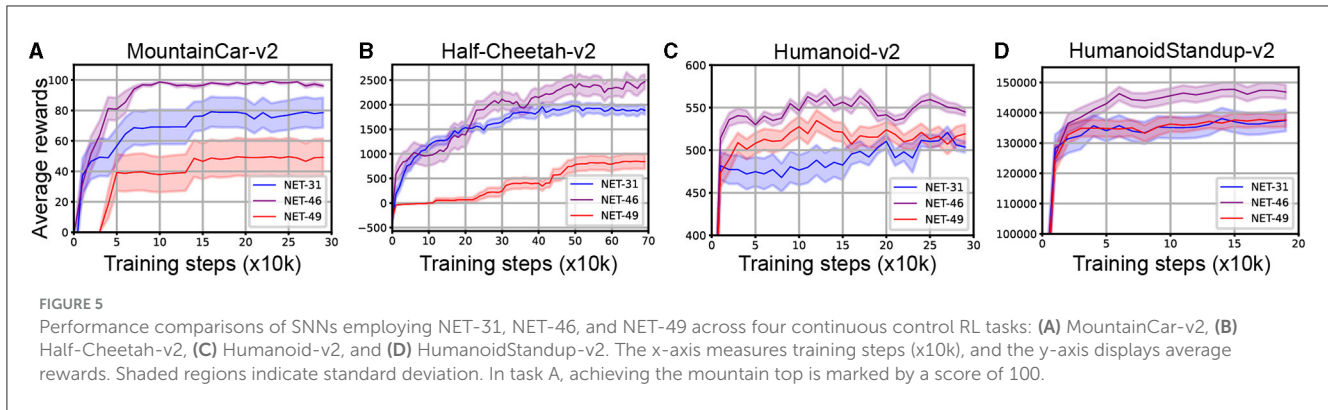
Four OpenAI gym games (Mujoco) were used to test the algorithms' performance, as shown in Figure 4. We select these Mujoco tasks instead of Atair 2000 games for their more dynamic features, especially animal-survival-like RL (Figures 4B–D).

In order to ensure the reproducibility of the proposed algorithms, we repeated each RL experiment ten times with different network initializations given different random seeds. Each RL task took 300 k (1k = 1,000) steps for learning and was evaluated every 10k step. At each evaluation time, we reported the average reward of over 10 episodes without giving any exploration noise, and each episode lasted for a maximum of 1 k execution steps. The MountainCar-v2, Half-Cheetah-v2, Humanoid-v2, and HumanoidStandup-v2 tasks are with state-action dimensions of [2, 1], [17, 6], [376, 17], and [376, 17], respectively. All these RL-related configurations are similar to those in the paper (Hasani et al., 2020), where a simpler network architecture borrowed from *C. elegans* is used.

We compared our algorithms to the benchmark LSTM and MLP networks. The experiments were built upon the open-source codebase from OpenAI Spinning Up.¹ The related algorithms, including NET-31, NET-46, NET-49, and

¹ <https://github.com/openai/spinningup>





Net-Rand, were all trained under the same standard ARS algorithm. We evaluated these algorithms on the four continuous control tasks under the same experimental configurations and compared their performance for further analysis. Unless for special statements, most algorithms use the same set of parameters.

4.2 Performance comparison of SNNs using different BTs

The performance of SNNs using three types of topology on four reinforcement learning tasks is shown in Figure 5. From the statistical results, the performance of SNNs using NET-46 is better

than those using NET-31 and NET-49, representing NET-46 could be the main best-topology candidate in the next experiments for comparing its performance with random networks and other state-of-the-art algorithms.

For the different distribution of network motifs in three BTs, it is obvious that motif-5 occupied a higher proportion in NET-46 than NET-31 and NET-49 (see Figures 3G–I for more details). It is impressive that the motif-5 contains a more cross-layer connection, making us speculate that the proper proportion of cross-layer connections plays a significant role in RL tasks. The motif-9 is another main difference between these three topologies. However, the influence of motif-9 is opposite to motif-5, where networks using more motif-9 exhibited poorer performance than other control algorithms.

TABLE 1 The performance comparisons in Mujoco RL.

Tasks	Architectures	Rules	RL Scores	Network sparsity
MountainCar-v2	LSTM ^a	BPTT	98.98 ± 0.59	0%
	MLP ^b	PPO	95.5 ± 1.5	0%
	Random	Search	49.59 ± 49.59	61%
	NET-46 (Ours)	Search	99.14 ± 0.12	61%
HalfCheetah-v2	LSTM ^a	BPTT	1009.51 ± 641.95	0%
	MLP ^b	PPO	1,601.05 ± 506.50	0%
	Random	Search	1,917.40 ± 819.39	61%
	NET-46 (Ours)	Search	2,468.18 ± 962.36	61%
Humanoid-v2	LSTM ^a	BPTT	537.19 ± 33.54	0%
	MLP ^b	PPO	537.36 ± 39.73	0%
	Random	Search	545.65 ± 32.79	61%
	NET-46 (Ours)	Search	564.13 ± 37.65	61%
HumanoidStandup-v2	LSTM ^a	BPTT	142255 ± 7502	0%
	MLP ^b	PPO	140,780 ± 9,676	0%
	Random	Search	139,863 ± 24,538	61%
	NET-46 (Ours)	Search	147,697 ± 11,931	61%

^aCitation for LSTM (Hochreiter and Schmidhuber, 1997). ^bCitation for MLP (Schulman et al., 2017). The metrics in bold are the corresponding metrics for our model, just for highlighting.

4.3 SNNs using NET-46 V.S. SNNs using random topology

The SNN using NET-46 exhibit a superior performance than SNNs using NET-31 and NET-49. However, we cannot claim the NET-46 is the best BT candidate without comparing it to an objective benchmark as the baseline. Hence, we select two types of benchmarks for verification: (1) the bottom baseline is defined by the SNN using a random network, given the name of NET-Rand; (2) ANNs define the top baseline using MLP or LSTM, which will be introduced extensively in the next section.

For the bottom baseline, we conducted a topology with the same number of brain regions and connections to the NET-46. The ratio of excitatory to inhibitory connections was 0.7 to 0.3. The SNN using NET-Rand was trained on the four RL tasks, and the inference performance comparison of it and NET-46 was shown in Figure 6 and Table 1. The experimental results showed that the performance of SNNs using the NET-rand was much lower than those using NET-46, which to some extent, indicated that the NET-46 contains some key topology advantage for the efficient RL. The performance on the MountainCar-V2 task was higher than other tasks, which the less complexity might cause.

Furthermore, the enhanced performance of NET-46 over NET-Rand in our experiments can be attributed to its biologically-informed structural properties, such as optimized connectivity patterns and modularity, which are absent in randomly generated networks. The SNN using NET-Rand could also be convergence but only with lower average rewards. Hence, now we can answer the hypothesis from the computation perspective that the evolutionary neural networks have stored some key prior knowledge in brain topology, which further contributes to the next-step network learning. The top baseline is then tested and showed in the following section.

4.4 SNN using NET-46 V.S. classical ANNs

We selected the MLP and LSTM and tested their performance on the four Mujoco continuous control RL tasks. The experimental results are shown in Figure 7. For the MountainCar-v2 task, our algorithm (i.e., SNN using NET-46) reached a comparable performance (99.14 ± 0.12) to the other two benchmark algorithms, including LSTM (98.98 ± 0.59 , $n = 10$, $P = 0.94$) and MLP (95.5 ± 1.5 , $n = 10$, $P < 0.01$). For the other three relatively more complex tasks, our algorithm performed much better and reached a higher performance than LSTM with [P value = 0.01, $P = 0.12$, and $P = 0.04$] and MLP with [$P = 0.01$, $P = 0.19$, and $P = 0.15$] for Half-cheetah-v2, Humanoid-v2, and HumanoidStandup-v2 RL tasks, respectively. See Table 1 for more details. It should be noted that although the brain-like topology algorithm represented by NET-46 in this article has better computational performance than MLP and LSTM algorithms, the SNN still has some room for further improvement in terms of computational cost.

5 Conclusion

Incorporating biological topology into SNNs can provide insights into the structural organization of neural networks. This study utilized the mesoscale connectome data from the Allen mouse brain atlas, involving 213 mouse brain regions, to explore how specific topological clusters (i.e., NET-31, NET-46, and NET-49) can be clustered, analyzed, filtered, and incorporated into SNNs. The focus was on examining the structural compatibility of these clusters with SNN architectures, aiming to understand their potential influence on network performance.

These three clusters' excitatory-inhibitory connection types and sparseness are consistent with the biological ones,

including sensory, hidden (for memory), and motor brain regions. The three BTs exhibited different performances during RL, and the NET-46 outperformed NET-31, NET-49, and the random network (NET-Rand). The detailed brain regions in NET-46 contain more auditory brain regions, more hidden brain regions for memory and multi-sensory integration, and more global neuromodulatory pathways, such as 5-HT projections from the CLI region to the nucleus and thalamus.

The experimental results showed that the mouse brain-like topology could improve SNNs from the perspective of accumulated rewards and network sparsity more than some ANNs, including the LSTM and MLP. We think more biological network-scale principles can further be incorporated into SNNs, and this integration of neuroscience and artificial intelligence has much in store for the future.

Data availability statement

The datasets presented in this study can be found in online repositories. The names of the repository/repositories and accession number(s) can be found in the article/supplementary material. The mouse brain topology dataset can be downloaded from <http://connectivity.brain-map.org>. The source code of this paper can be found at <https://github.com/thomasaimondy/BT-SNN>.

Author contributions

YoW: Formal analysis, Methodology, Resources, Software, Validation, Writing – original draft, Writing – review & editing. YaW: Data curation, Formal analysis, Methodology, Validation, Visualization, Writing – original draft. XZ: Formal analysis,

Methodology, Supervision, Writing – review & editing. JD: Supervision, Funding acquisition, Writing – review & editing. TZ: Formal analysis, Methodology, Resources, Writing – original draft, Writing – review & editing. BX: Supervision, Funding acquisition, Writing – review & editing.

Funding

The author(s) declare that financial support was received for the research, authorship, and/or publication of this article. This work was supported by the Strategic Priority Research Program of Chinese Academy of Sciences (No. XDA0370305), the Beijing Nova Program (Grant No. 20230484369), the Shanghai Municipal Science and Technology Major Project (Grant No. 2021SHZDZX), and the Youth Innovation Promotion Association of Chinese Academy of Sciences.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

- Ahn, Y.-Y., Bagrow, J. P., and Lehmann, S. (2010). Link communities reveal multiscale complexity in networks. *Nature* 466, 761–764. doi: 10.1038/nature09182
- Bäck, T., and Schwefel, H.-P. (1993). An overview of evolutionary algorithms for parameter optimization. *Evolut. Comput.* 1, 1–23. doi: 10.1162/evco.1993.1.1.1
- Bellec, G., Scherr, F., Subramoney, A., Hajek, E., Salaj, D., Legenstein, R., et al. (2020). A solution to the learning dilemma for recurrent networks of spiking neurons. *Nat. Commun.* 11:3625. doi: 10.1038/s41467-020-17236-y
- Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., et al. (2016). Openai gym. *arXiv preprint arXiv:1606.01540*.
- Cramer, B., Billaudelle, S., Kanya, S., Leibfried, A., Grbl, A., Karasenko, V., et al. (2022). Surrogate gradients for analog neuromorphic computing. *Proc. Natl. Acad. Sci. U S A*. 119:e2109194119. doi: 10.1073/pnas.2109194119
- Dan, Y., and Poo, M. M. (2004). Spike timing-dependent plasticity of neural circuits. *Neuron* 44, 23–30. doi: 10.1016/j.neuron.2004.09.007
- Deng, L., Wu, Y., Hu, X., Liang, L., Ding, Y., Li, G., et al. (2020). Rethinking the performance comparison between SNNs and ANNs. *Neural Netw.* 121, 294–307. doi: 10.1016/j.neunet.2019.09.005
- Doya, K. (2000). Reinforcement learning in continuous time and space. *Neur. Comput.* 12, 219–245. doi: 10.1162/089976600300015961
- Eshraghian, J. K., Ward, M., Neftci, E. O., Wang, X., Lenz, G., Dwivedi, G., et al. (2023). “Training spiking neural networks using lessons from deep learning.” in *Proceedings of the IEEE*. doi: 10.1109/PROC.2023.3308088
- Frankle, J., and Carbin, M. (2018). The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*.
- Goulas, A., Damicelli, F., and Hilgetag, C. C. (2021). Bio-instantiated recurrent neural networks: Integrating neurobiology-based network topology in artificial networks. *Neural Netw.* 142, 608–618. doi: 10.1016/j.neunet.2021.07.011
- Ham, D., Park, H., Hwang, S., and Kim, K. (2021). Neuromorphic electronics based on copying and pasting the brain. *Nat. Electr.* 4, 635–644. doi: 10.1038/s41928-021-00646-1
- Han, S., Pool, J., Tran, J., and Dally, W. J. (2015). Learning both weights and connections for efficient neural networks. *arXiv preprint arXiv:1506.02626*.
- Hasani, R., Lechner, M., Amini, A., Rus, D., and Grosu, R. (2020). “A natural lottery ticket winner: Reinforcement learning with ordinary neural circuits,” in *International Conference on Machine Learning*, 4082–4093.
- Hinton, G., Vinyals, O., and Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.
- Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. *Neur. Comput.* 9, 1735–1780. doi: 10.1162/neco.1997.9.8.1735
- Izhikevich, E. M. (2004). Which model to use for cortical spiking neurons? *IEEE Trans. Neural Netw.* 15, 1063–1070. doi: 10.1109/TNN.2004.832719
- Kalinka, A. T., and Tomancak, P. (2011). linkcomm: an r package for the generation, visualization, and analysis of link communities in networks of arbitrary size and type. *Bioinformatics* 27, 2011–2012. doi: 10.1093/bioinformatics/btr311

- Lechner, M., Hasani, R., Zimmer, M., Henzinger, T. A., and Grosu, R. (2019). "Designing worm-inspired neural networks for interpretable robotic control," in *2019 International Conference on Robotics and Automation (ICRA)* (IEEE), 87–94. doi: 10.1109/ICRA.2019.8793840
- Li, Y., Fang, X., Gao, Y., Zhou, D., Shen, J., Liu, J. K., et al. (2024). "Efficient structure slimming for spiking neural networks," in *IEEE Transactions on Artificial Intelligence*. doi: 10.1109/TAI.2024.3352533
- Lillicrap, T. P., Santoro, A., Marris, L., Akerman, C. J., and Hinton, G. (2020). Backpropagation and the brain. *Nat. Rev. Neurosci.* 21, 335–346. doi: 10.1038/s41583-020-0277-3
- Liu, Y.-H., and Wang, X.-J. (2001). Spike-frequency adaptation of a generalized leaky integrate-and-fire model neuron. *J. Comput. Neurosci.* 10, 25–45. doi: 10.1023/A:1008916026143
- Loeffler, A., Diaz-Alvarez, A., Zhu, R., Ganesh, N., Shine, J. M., Nakayama, T., et al. (2023). Neuromorphic learning, working memory, and metaplasticity in nanowire networks. *Sci. Adv.* 9:eadg3289. doi: 10.1126/sciadv.adg3289
- Luo, L. (2021). Architectures of neuronal circuits. *Science* 373:eabg7285. doi: 10.1126/science.abg7285
- Maass, W. (1997). Networks of spiking neurons: the third generation of neural network models. *Neural Netw.* 10, 1659–1671. doi: 10.1016/S0893-6080(97)00011-7
- Mania, H., Guy, A., and Recht, B. (2018). Simple random search provides a competitive approach to reinforcement learning. *arXiv preprint arXiv:1803.07055*.
- Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., and Alon, U. (2010). Network motifs: Simple building blocks of complex networks. *Nat. Rev. Neurosci.* 11, 615–627. doi: 10.1126/science.298.5594.824
- Oh, S. W., Harris, J. A., Ng, L., Winslow, B., Cain, N., Mihalas, S., et al. (2014). A mesoscale connectome of the mouse brain. *Nature* 508, 207–214. doi: 10.1038/nature13186
- Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (2007). *Numerical Recipes with Source Code CD-ROM 3rd Edition: The Art of Scientific Computing*. Cambridge: Cambridge University Press.
- Prill, R. J., Iglesias, P. A., and Levchenko, A. (2005). Dynamic properties of network motifs contribute to biological network organization. *PLoS Biol.* 3:e343. doi: 10.1371/journal.pbio.0030343
- Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.
- Sporns, O., and Kotter, R. (2004). Motifs in brain networks. *PLoS Biol.* 2:e369. doi: 10.1371/journal.pbio.0020369
- Tang, G., Kumar, N., and Michmizos, K. P. (2020). "Reinforcement co-learning of deep and spiking neural networks for energy-efficient mapless navigation with neuromorphic hardware," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE), 6090–6097. doi: 10.1109/IROS45743.2020.9340948
- Wildenberg, G. A., Rosen, M. R., Lundell, J., Paukner, D., Freedman, D. J., and Kasthuri, N. (2021). Primate neuronal connections are sparse in cortex as compared to mouse. *Cell Rep.* 36:109709. doi: 10.1016/j.celrep.2021.109709
- Xu, Q., Li, Y., Fang, X., Shen, J., Liu, J. K., Tang, H., et al. (2023). Biologically inspired structure learning with reverse knowledge distillation for spiking neural networks. *arXiv preprint arXiv:2304.09500*.
- Xu, Q., Qi, Y., Yu, H., Shen, J., Tang, H., Pan, G., et al. (2018). "CSNN: an augmented spiking based framework with perceptron-inception," in *IJCAI*, 1646. doi: 10.24963/ijcai.2018/228
- Yin, W., Brittain, D., Borseth, J., Scott, M. E., Williams, D., Perkins, J., et al. (2020). A petascale automated imaging pipeline for mapping neuronal circuits with high-throughput transmission electron microscopy. *Nat. Commun.* 11:4949. doi: 10.1038/s41467-020-18659-3
- Zenke, F., and Gerstner, W. (2017). Hebbian plasticity requires compensatory processes on multiple timescales. *Philos. Trans. R. Soc. B* 372:20160259. doi: 10.1098/rstb.2016.0259
- Zhang, D., Zhang, T., Jia, S., and Xu, B. (2022). "Multiscale dynamic coding improved spiking actor network for reinforcement learning," in *Thirty-Sixth AAAI Conference on Artificial Intelligence*. doi: 10.1609/aaai.v36i1.19879
- Zhang, T., Cheng, X., Jia, S., Poo, M. M., Zeng, Y., and Xu, B. (2021a). Self-backpropagation of synaptic modifications elevates the efficiency of spiking and artificial neural networks. *Sci. Adv.* 7:eabh0146. doi: 10.1126/sciadv.abh0146
- Zhang, T., Jia, S., Cheng, X., and Xu, B. (2021b). Tuning convolutional spiking neural network with biologically plausible reward propagation. *IEEE Trans. Neural Netw. Learn. Syst.* 33, 7621–7631. doi: 10.1109/TNNLS.2021.3085966
- Zhang, T., Zeng, Y., and Xu, B. (2017). A computational approach towards the microscale mouse brain connectome from the mesoscale. *J. Integr. Neurosci.* 16, 291–306. doi: 10.3233/JIN-170019
- Zhang, T., Zeng, Y., Zhao, D., and Xu, B. (2018). "Brain-inspired balanced tuning for spiking neural networks," in *IJCAI*, 1653–1659. doi: 10.24963/ijcai.2018/229