Joint Learning Neuronal Skeleton and Brain Circuit Topology with Permutation Invariant Encoders for Neuron Classification

Minghui Liao\textsuperscript{1, 2, 3}, Guojia Wan\textsuperscript{1, 2, 3 *}, Bo Du\textsuperscript{1, 2, 3 *}

\textsuperscript{1}National Engineering Research Center for Multimedia Software, School of Computer Science, Wuhan University, China
\textsuperscript{2}Institute of Artificial Intelligence, Wuhan University, China
\textsuperscript{3}Hubei Key Laboratory of Multimedia and Network Communication Engineering, Wuhan University, China
minghui@whu.edu.cn, guojia@whu.edu.cn, dubo@whu.edu.cn

Abstract

Determining the types of neurons within a nervous system plays a significant role in the analysis of brain connectomics and the investigation of neurological diseases. However, the efficiency of utilizing anatomical, physiological, or molecular characteristics of neurons is relatively low and costly. With the advancements in electron microscopy imaging and analysis techniques for brain tissue, we are able to obtain whole-brain connectome consisting neuronal high-resolution morphology and connectivity information. However, few models are built based on such data for automated neuron classification. In this paper, we propose NeuNet, a framework that combines morphological information of neurons obtained from skeleton and topological information between neurons obtained from neural circuit. Specifically, NeuNet consists of three components, namely Skeleton Encoder, Connectome Encoder, and Readout Layer. Skeleton Encoder integrates the local information of neurons in a bottom-up manner, with a one-dimensional convolution in neural skeleton’s point data; Connectome Encoder uses a graph neural network to capture the topological information of neural circuit; finally, Readout Layer fuses the above two information and outputs classification results. We reprocess and release two new datasets for neuron classification task from volume electron microscopy (VEM) images of human brain cortex and Drosophila brain. Experiments on these two datasets demonstrated the effectiveness of our model with accuracy of 0.9169 and 0.9363, respectively. Code and data are available at: https://github.com/WHUminghui/NeuNet

Introduction

Identifying various neuron types in a nervous system is a fundamental task in numerous neuroscience studies. Assigning neuron type is a necessary step to understand the anatomical and functional properties of a brain circuit (Armañanzas and Ascoli 2015). By identifying and classifying various types of neurons, researchers can determine which neurons are most susceptible to causing some neurodegenerative diseases, such as Alzheimer’s (Buckner et al. 2005) and Amyotrophic Lateral Sclerosis (ALS). However, due to the vast number and minuscule size of neurons, anatomical-based neuron classification methods are exceedingly time-consuming and labor-intensive.

Recently, volume electron microscopy (VEM) imaging and analysis techniques (Januszewski et al. 2018) contribute the availability of whole-brain three-dimensional (3D) reconstruction with cellular resolution, providing a new opportunity to understand the neuronal properties and functions. Specifically, the 3D images of brain tissue generated by VEM contain a wealth of information about neuron size, location, morphology, and connectivity. We show a neuron in Figure 1 from Drosophila connectome (Scheffer et al. 2020). We can observe that the morphology of the complex nerve fibers (including axons and dendrites) that spans multiple brain regions. The red centerline within the fiber voxel is referred as the skeleton of the neuron, which can be depicted as a set filling with 3D points in XYZ coordinates, $P = \{p_q | q = 1, \ldots, m\}$. A large number of such neurons are interconnected, forming the connectome of Drosophila brain (Scheffer et al. 2020). The connectome can be depicted by a complex network, which is formalized as a graph $G = (\mathcal{V}, \mathcal{E})$. In this network, nodes represent the neurons and edges represent the synaptic connections between them. Apparently, both of these two data exhibit permutation invariance. That is, permuting or rearranging the order of the ele-
ments in these data does not change the results or the information conveyed by the data.

The morphology, or shape, of a neuron can provide significant insights into its function within the nervous system. Traditional methods based on feature statistics (Gillette and Ascoli 2015; Kanari et al. 2019; Zhu et al. 2022) usually conduct analysis on the optical data. However, the resolution of neuronal morphological data obtained from optical microscopes is relatively low, which cannot provide neuronal connectivity information. Importantly, the connectivity pattern (Fulcher and Fornito 2016) of neuron can also reveal the properties about its function.

To date, jointly learning the morphology and connectivity remains unexplored. In this paper, we propose a unified, efficient framework called NeuNet, a neuron classification framework that incorporates morphological information of neuron skeleton and topological information of connectome. NeuNet consists of a skeleton-level feature extractor and a connectome-level feature extractor, which well respect permutation invariance of points and topology connectivity in the input. After extracting the morphological features from neuron skeleton data and the topological information from the connectome, a fusion module combines the above two features to finally obtain the representation of the neuron. The main contributions of NeuNet are as follows:

- NeuNet can learn neuronal representations of entire brain circuit in a high-throughput and faster manner and use them for downstream tasks.
- Our model incorporates the skeleton’s morphological information and the connectome’s topological information of neurons, both of which are in a permutation-invariant manner, and it is in line with the actual characteristics of neuron VEM data.
- We reprocessed and released two 3D brain reconstruction data. We introduced baseline models from other tasks, and upon comparative evaluation, our model manifested superior classification performance, successfully capturing the latent feature representations of neurons.

Related Work

Neurons exhibit diverse properties in terms of connectivity, morphology, and functionality, and the classification of neurons is the cornerstone for exploring the structure and functional mechanisms of brain circuit (Wagner, Regev, and Yosef 2016). Since the functions of neuron are closely related to its morphology, several methods have been proposed for comparing and classifying neurons from light-microscopy (Markram et al. 2015; Costa et al. 2016a; Peng et al. 2015; Simbül et al. 2014).

With the development of electron microscope imaging speed (Eberle et al. 2015), electron microscopy can provide higher resolution and more informative photographs of brain tissue sections. Taking advantage of these innovations, the brain circuit reconstruction have been made to map the connectivity in *Drosophila* optic and antennal lobes (Berck et al. 2016; Takemura et al. 2013), the mouse retina, thalamus, and cortex (Greene, Kim, and Seung 2016; Lee et al. 2016; Morgan et al. 2016). However, there are relatively few works (Costa et al. 2016b; Schubert et al. 2019; Zhu et al. 2022) that use neural networks for feature extraction and classification of the above-described neuron data, and none of which take into account the topological information of brain circuit.

Methodology

Notations and Problem Definition

**Neuron Data.** We designed NeuNet to integrate morphological features about the skeleton and topological features about connectome. A skeleton-level neuron data is represented as a set of points \( P = \{ p \mid p = (x, y, z) \} \), where each point \( p \) is a coordinate vector.

For modeling connectivity, we regard each neuron as a node. Let synapse connections from pre-synaptic cells to post-synaptic cells be represented as edges, where the number of synapses between two neurons is as connection strength. We then constructed a graph to represent the connectome. We denote the graph with \( n \) nodes (neurons) by \( G = (V, E) \), where \( V \) represents its node set and \( E \) represents the set of edges. The adjacency matrix of \( G \) is denoted by \( A \in R^{n \times n} \) with its \((i, j)-th\) entry \( A[i, j] = e \) indicating the strength of the connection between node \( i \) and node \( j \), and \( e \) is numerically equal to the number of synapses connected between neuron \( i \) and neuron \( j \).

**Permutation Invariance.** Our input include a point set and a graph. The point set is inherently unordered. That is, permuting the input order of points within this point set does not affect the skeleton morphology of the neuron. As for the input graph, it is a 2-tuple \((V, E)\) that consists of two sets: a node set \( V \) and an edge set \( E \). Likewise, it should follow permutation invariance as well.

Due to its permutation invariance, any learners working with such data should be invariant under transformations. For example, changing the input order of points, nodes and edges should not modify its neuron classification results.

**Neuron Classification.** In the task of neuron classification, we are given skeleton-level point set \( P \), the connectome level graph \( G = (V, E) \), and the neuron labels denoted by one-hot vector \( Y \). Each labeled node \( i \in V \) is associated with a one-hot vector \( y_i \subset \{0, 1\}^C \) which encodes its ground-truth class, where \( C \) is the number of predefined classes. We needed to learn a function \( f_\theta(i) (G, P) \) parameterized by \( \theta \) which can predict the correct class for a given unlabeled node \( i \).

Network Architecture

NeuNet consists of three sub-modules (Figure 2): Skeleton Encoder, Connectome Encoder, and Readout Layer. Skeleton Encoder is used to extract the morphological information of neuron; Connectome Encoder is used to extract the topological information of the neural circuit; Readout Layer fuses the skeleton-level and connectome-level representation, and output a \( 1 \times C \) classification score vector with a Multilayer Perceptron(MLP).
Neural circuit

Neuron skeleton

Figure 2: Architecture of proposed NeuNet. On the one hand, the skeleton data of neurons are input into Skeleton Encoder, and Sampling operation samples the points in the skeleton’s point set using FPS. Grouping operation divides the attached points into groups with the sampled points as the center, and ConvBlock extracts the features on these group’s points and fuses them using max pooling to obtain the group’s global features. The above process is repeated until the skeleton’s global features are obtained. On the other hand, GNN-based Connectome Encoder mines the topological features of the neural circuit through the information interaction between nodes and neighboring nodes. Finally, Readout Layer fuses the above two features and outputs the classification results with an MLP.

**Skeleton Encoder**

Neurons possess intricate branching patterns, encompassing fundamental local structures such as axons, dendrites, and curvatures. These complex branching morphologies often provide insights into the classification and functional roles of neuron (Udvary et al. 2022). However, inferring the functions and classifications of neurons directly from their 3D structures remains a challenging task. Therefore, there arises a necessity to develop morphology feature extractor for neuron morphology, which is able to obtain global representation of the neuron skeleton concerning its local structure.

Considering that neuron skeleton data is an unordered point set, we designed Skeleton Encoder in bottom-up manner consisting of multiple SampleBlocks, which can extract consistent feature representations from varying input orders of skeleton data pertaining to the same neuron. The SampleBlock abstraction level is made of three key operations (Figure 2): Sampling, Grouping, and Conv1D.

**Sampling.** The key component of CNN is the convolution kernel, which extracts local features of an image by performing Hadamard product with the certain region (kernel size) of pixels, and acquires a larger range of semantic features by multiple convolutional operations. Drawing inspiration from this concept, we aim to construct local feature extractor working on different regions of a neuron skeleton. Initially, it is imperative to determine the regions with this feature extractor is intended to interact. Given input points \( \{p_1, p_2, \ldots, p_m\} \), we employ iterative farthest point sampling (FPS) (Li et al. 2022) to select a subset of points \( \{p_{i_1}, p_{i_2}, \ldots, p_{i_k}\} \) as the centroids of these regions. This selection ensures that the sampled points are the farthest apart (in European distance in this study). In contrast to random sampling, FPS can offer a wider coverage, thereby establishing a more comprehensive receptive field among the given points. This facilitates the acquisition of global morphological information from the input points.

**Grouping.** The subsequent task involves delineating the respective extent of these regions around the central points. For a given set of points \( P = \{p_1, p_2, \ldots, p_m\} \) and points \( S = \{p_{i_1}, p_{i_2}, \ldots, p_{i_k}\} \) selected by Sampling, we take \( S \) as the centroids and find \( k \) points in \( P \) that are most adjacent to the centroid respectively to form \( S \) group. Considering that the density and distribution of points in \( P \) are not uniform, we use ball query (Qi et al. 2017) to find \( k \) points that are within a radius of the centroid point to guarantee a fixed regional scale. The input to the Grouping operator is a point set of size \( m \), and the output are groups of point sets of size \( s \cdot k \), where each group corresponds to a local region and \( k \) is the number of points in the neighborhood of centroid points.

**Conv1D.** For the points of the groups generated by Grouping operation, they retain an inherent unordered nature. In order to extract consistent features from groups’ points irrespective of the order of input points, we employ one-dimensional convolution for capturing local morphology features within the groups. The input skeleton feature \( X_{in} \in R^{m \times d_{in}} \) is fed into Conv1D and is transformed into the next latent skeleton feature \( X_{out} \in R^{m \times d_{out}} \) as \( X_{out} = \text{ReLU}(W \ast X_{in} + b) \), where \( W \in R^{s \times d_{in} \times d_{out}} \) is the kernel and \( k \) is the kernel size. \( b \) denotes bias. \( \ast \) is the valid matrix multiplication. When the initial input is the point set \( P \), the corresponding skeleton feature is \( X_{g_{yz}} \in R^{m \times 3} \). Otherwise, \( d_{in} \) and \( d_{out} \) are intermediate latent feature dimensions.
Table 1: Overall classification accuracy on the HemiBrain dataset and within-class accuracy for 10 of its classes.

<table>
<thead>
<tr>
<th>Method</th>
<th>Acc</th>
<th>LC</th>
<th>KC</th>
<th>SMP</th>
<th>AVLP</th>
<th>SLP</th>
<th>CL</th>
<th>PS</th>
<th>LHAV</th>
<th>PLP</th>
<th>LHPV</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLP</td>
<td>0.1796</td>
<td>0.1763</td>
<td>0.1652</td>
<td>0.1798</td>
<td>0.2032</td>
<td>0.1863</td>
<td>0.1765</td>
<td>0.1956</td>
<td>0.1652</td>
<td>0.1956</td>
<td>0.1698</td>
</tr>
<tr>
<td>PointNet++ (Qi et al. 2017)</td>
<td>0.8627</td>
<td>0.8821</td>
<td>0.8962</td>
<td>0.7584</td>
<td>0.8546</td>
<td>0.8654</td>
<td>0.8756</td>
<td>0.8745</td>
<td>0.8522</td>
<td>0.8921</td>
<td>0.8541</td>
</tr>
<tr>
<td>GCNII (Chen et al. 2020)</td>
<td>0.5266</td>
<td>0.4956</td>
<td>0.5361</td>
<td>0.5411</td>
<td>0.5323</td>
<td>0.5362</td>
<td>0.5412</td>
<td>0.5212</td>
<td>0.5421</td>
<td>0.4812</td>
<td>0.4932</td>
</tr>
<tr>
<td>DGCNN (Wang et al. 2019)</td>
<td>0.8560</td>
<td>0.8425</td>
<td>0.8632</td>
<td>0.8410</td>
<td>0.8651</td>
<td>0.8510</td>
<td>0.8710</td>
<td>0.8632</td>
<td>0.8365</td>
<td>0.8683</td>
<td>0.8360</td>
</tr>
<tr>
<td>AGNN (Thekumparampil et al. 2018)</td>
<td>0.5327</td>
<td>0.5621</td>
<td>0.5214</td>
<td>0.5412</td>
<td>0.5630</td>
<td>0.5489</td>
<td>0.5321</td>
<td>0.5423</td>
<td>0.5621</td>
<td>0.5421</td>
<td>0.5213</td>
</tr>
<tr>
<td>Point-Transformer (Zhao et al. 2021)</td>
<td>0.8473</td>
<td>0.8516</td>
<td>0.8326</td>
<td>0.8426</td>
<td>0.8321</td>
<td>0.8561</td>
<td>0.8502</td>
<td>0.8632</td>
<td>0.8412</td>
<td>0.8564</td>
<td>0.8214</td>
</tr>
<tr>
<td>NeuNet(ours)</td>
<td>0.9169</td>
<td>0.9721</td>
<td>1.0000</td>
<td>0.9291</td>
<td>0.9259</td>
<td>0.9369</td>
<td>0.9200</td>
<td>0.9000</td>
<td>0.8929</td>
<td>0.9245</td>
<td>0.8813</td>
</tr>
</tbody>
</table>

Table 2: Overall classification accuracy on the H01 dataset and within-class accuracy on its 6 layers. L1, L2, ..., L6 are the 6 types of cells at different layers of the human cerebral cortex.

<table>
<thead>
<tr>
<th>Method</th>
<th>Acc</th>
<th>L1</th>
<th>L2</th>
<th>L3</th>
<th>L4</th>
<th>L5</th>
<th>L6</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLP</td>
<td>0.3239</td>
<td>0.3364</td>
<td>0.3562</td>
<td>0.3235</td>
<td>0.3102</td>
<td>0.3321</td>
<td>0.3014</td>
</tr>
<tr>
<td>PointNet++</td>
<td>0.6242</td>
<td>0.6025</td>
<td>0.6821</td>
<td>0.6214</td>
<td>0.6025</td>
<td>0.6412</td>
<td>0.6123</td>
</tr>
<tr>
<td>GCNII</td>
<td>0.6147</td>
<td>0.6231</td>
<td>0.6512</td>
<td>0.5831</td>
<td>0.6451</td>
<td>0.5821</td>
<td>0.6612</td>
</tr>
<tr>
<td>DGCNN</td>
<td>0.6104</td>
<td>0.6120</td>
<td>0.6512</td>
<td>0.6023</td>
<td>0.6325</td>
<td>0.6230</td>
<td>0.5984</td>
</tr>
<tr>
<td>AGNN</td>
<td>0.6334</td>
<td>0.6231</td>
<td>0.6213</td>
<td>0.6423</td>
<td>0.5921</td>
<td>0.6215</td>
<td>0.6432</td>
</tr>
<tr>
<td>Point-Transformer</td>
<td>0.5910</td>
<td>0.6012</td>
<td>0.5821</td>
<td>0.6123</td>
<td>0.6215</td>
<td>0.6023</td>
<td>0.5920</td>
</tr>
<tr>
<td>NeuNet</td>
<td>0.9363</td>
<td>0.8113</td>
<td>0.9729</td>
<td>0.9134</td>
<td>0.9306</td>
<td>0.9273</td>
<td>0.9247</td>
</tr>
</tbody>
</table>

Here, \( \alpha \) models the strength of the initial residual connection, while \( \beta \) models the strength of the identity mapping. The final feature \( \mathbf{X}_c \) obtained after \( N_1 \) layers of propagation encapsulates the topological information of the contextual surroundings in which the neuron resides. Then, the row vector \( \mathbf{x}_c \in \mathbf{X}_c \) represent a neuron’s topological low-dimensional feature representation. In our experiment, the values of \( \alpha, \beta \) and \( N_1 \), as mentioned above, were set to 0.1, 0.5, 64 respectively.

Since a graph is a topological structure, its data structure also possesses permutation invariance. This means that permuting the node orders of the graph will not change the learned representation vector of the nodes.

Readout Layer

After Skeleton Encoder generates feature representations \( \mathbf{x}_c \) about neuron morphology and Connectome Encoder generates feature representations \( \mathbf{x}_c \) about the topological information, we concatenate them to form the final feature representation \( \mathbf{x} = \mathbf{x}_s \otimes \mathbf{x}_c \). Then, we use an MLP as a classifier to predict neuron label: \( c = \text{MLP}(\mathbf{x}) \), where \( c \) denotes the prediction results.

Finally, we use cross entropy as a classification loss to train our model: \( \mathcal{L} = -\sum_{i=1}^{C} y_i \log c_i + (1 - y_i) \log (1 - c_i) \), where \( y_i \in \{0, 1\} \) is the ground truth for \( i \)-th type. \( |C| \) is the number of neuron types.

Experiments

Data Description

HemiBrain. HemiBrain (Scheffer et al. 2020) is a dense reconstruction of a portion of the central brain of Drosophila. We accessed the raw data through NeuPrint (Clements et al. 2020) and reprocessed it into 19,384 skeletons(point sets) of neurons which can be classified into 191 different classes. Regarding these neurons as nodes and
the synapses between neurons as connecting edges, these neurons are connected by 3,307,248 edges.

**H01.** H01 (Shapson-Coe et al. 2021) is from a human cerebral cortex through steps such as slicing, imaging, segmentation, and synaptic detection. We obtained the raw data and reprocessed it into 15,554 neurons connected by 28,246 edges, which are from 6 layers of the cortex. The neurons of HemiBrain and H01 datasets will be randomly partitioned into training, validation, and test dataset in an 8:1:1 ratio.

**Results**

**Neuron Classification.** On the HemiBrain and H01 dataset, we compare our method with MLP, PointNet++ (Qi et al. 2017), GCNII (Chen et al. 2020), DGCNN (Wang et al. 2019), AGNN (Thekumparampil et al. 2018) and PointTransformer (Zhao et al. 2021). The classification results are listed in Table 1 and Table 2. The numerical values reported in this study refer to either overall accuracy or within-class accuracy. Overall accuracy represents the proportion of correctly classified samples out of the total number of samples. While within-class accuracy pertains to the proportion of correctly classified samples within a specific category out of the total number of samples in that category. As Table 1 suggests, our model outperforms other methods on the HemiBrain dataset. And it is ahead of the second-place model PointNet++ by 6.46% and ahead of the last-place MLP by 411.41%. As shown in Table 2, our model also significantly outperforms the other models on the H01 dataset, with an overall accuracy of 48.70% higher than the second place.

We draw skeletons of neuron randomly selected from the HemiBrain dataset. As shown in Figure 3, NeuNet is able to learn the local and global structure of neurons’ skeletons. For example, NeuNet learns an 'L'-type structure with complex bifurcation structures at both ends in KC; NeuNet recognizes the complex dendritic structure followed by a long axonal structure as a key characteristic in LHA V.

**Representation Visualization.** We produce t-SNE (Van der Maaten and Hinton 2008) visualizations on x learned by various methods. As shown in Figure 4, our model can transform the complex neuronal data into clustered groups, demonstrating the strong classification ability of the learned representations. However, this ability is not observed in DGCNN and PointNet++. Fuzzy grouping is also observed in Point-Transformer, with the scatter within the groups being more dispersed and overlapping occurring between the groups. Additionally, we can find that the representation of sub-classes also tend to cluster together, even if the subclass labeling information is not used during the training phase.

**Neuron Retrieval.** The representational capacity of NeuNet for high-dimensional neuron data can be applied to neuron retrieval. On the HemiBrain dataset, We use x to compute the Euclidean distance between all neurons. From the test set, we randomly select four neuron representations and identify the nearest neighboring neurons. The experimental results are shown in Figure 5. It is noteworthy that the subclasses of the top10 neurons are consistent with the query neuron (subclass information of the retrieved neurons is not shown in the figure), despite the fact that subclass information was not utilized during the training phase.

From Figure 5, we can observe that the retrieved neurons exhibit a high degree of similarity to the query neuron, and
Figure 4: The visualizations using t-SNE (Van der Maaten and Hinton 2008) of neuron representation learned by various methods. Where each of color represents a specific class. Since there are up to 191 class of HemiBrain, different classes of point may be given the same color.

Figure 5: Query neurons(red) and retrieved neurons(green) using NeuNet. The gray-shaded background represents the Drosophila brain, and the red shaded background indicates the region with the highest synaptic connectivity for the query neuron. Query neuron IDs and finer-grained categories are provided by (Scheffer et al. 2020). The subcategories of the top 10 retrieved neurons remain consistent with the query neurons(not shown in this figure), without the utilization of finer-grained subcategory information in NeuNet.

the retrieved neurons share common morphologies and location, indicating the potential presence of functional units within the corresponding regions.

Ablation Study. We replace the way of fusing features in Readout Layer from concatenation to max pooling and average pooling to construct NeuNet(maxPool) and NeuNet(avgPool), respectively; we discard the brain circuit information and use only the representation generated by Skeleton Encoder to design NeuNet-S; we remove skeleton information and employ solely the topological information generated by Connectome Encoder to design NeuNet-C. The experimental results of the aforementioned variants on the HemiBrain and H01 datasets are shown in Table 3 and Table 4, respectively. It can be seen that the ways of information fusion have little effect. When the NeuNet discards the skeleton information, its effectiveness decreases by 36.03% and 42.40% on the HemiBrain and H01 datasets, respectively. Similarly, when NeuNet discards the connectome information, the decreases on the HemiBrain and H01
Table 3: Classification accuracy of various variants of the NeuNet on the HemiBrain dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Acc</th>
<th>LC</th>
<th>KC</th>
<th>SMP</th>
<th>AVLP</th>
<th>SLP</th>
<th>CL</th>
<th>PS</th>
<th>LHAV</th>
<th>PLP</th>
<th>LHPV</th>
</tr>
</thead>
<tbody>
<tr>
<td>NeuNet</td>
<td>0.9169</td>
<td>0.9721</td>
<td>1.0000</td>
<td>0.9291</td>
<td>0.9259</td>
<td>0.9369</td>
<td>0.9200</td>
<td>0.9000</td>
<td>0.8929</td>
<td>0.9245</td>
<td>0.8813</td>
</tr>
<tr>
<td>NeuNet(maxPool)</td>
<td>0.9149</td>
<td>0.9814</td>
<td>1.0000</td>
<td>0.9291</td>
<td>0.9407</td>
<td>0.9279</td>
<td>0.9600</td>
<td>0.8714</td>
<td>0.9821</td>
<td>0.8929</td>
<td>0.9623</td>
</tr>
<tr>
<td>NeuNet(avgPool)</td>
<td>0.8612</td>
<td>0.8512</td>
<td>0.8325</td>
<td>0.8754</td>
<td>0.8695</td>
<td>0.8720</td>
<td>0.8426</td>
<td>0.8365</td>
<td>0.8714</td>
<td>0.8845</td>
<td>0.7965</td>
</tr>
<tr>
<td>NeuNet-S</td>
<td>0.5865</td>
<td>0.6021</td>
<td>0.6231</td>
<td>0.5841</td>
<td>0.5923</td>
<td>0.5814</td>
<td>0.5736</td>
<td>0.5596</td>
<td>0.6021</td>
<td>0.6254</td>
<td>0.6123</td>
</tr>
</tbody>
</table>

Table 4: Classification accuracy of various variants of NeuNet on the H01 dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Acc</th>
<th>L1</th>
<th>L2</th>
<th>L3</th>
<th>L4</th>
<th>L5</th>
<th>L6</th>
</tr>
</thead>
<tbody>
<tr>
<td>NeuNet</td>
<td>0.9363</td>
<td>0.8113</td>
<td>0.9729</td>
<td>0.9134</td>
<td>0.9306</td>
<td>0.9273</td>
<td>0.9247</td>
</tr>
<tr>
<td>NeuNet(maxPool)</td>
<td>0.9412</td>
<td>0.9436</td>
<td>0.9422</td>
<td>0.9449</td>
<td>0.9273</td>
<td>0.9462</td>
<td>0.8679</td>
</tr>
<tr>
<td>NeuNet(avgPool)</td>
<td>0.9419</td>
<td>0.9415</td>
<td>0.9624</td>
<td>0.9449</td>
<td>0.9273</td>
<td>0.9462</td>
<td>0.8679</td>
</tr>
<tr>
<td>NeuNet-S</td>
<td>0.6356</td>
<td>0.6541</td>
<td>0.6125</td>
<td>0.6023</td>
<td>0.6152</td>
<td>0.6201</td>
<td>0.6312</td>
</tr>
<tr>
<td>NeuNet-C</td>
<td>0.5123</td>
<td>0.5210</td>
<td>0.5023</td>
<td>0.5421</td>
<td>0.5011</td>
<td>0.5102</td>
<td>0.5321</td>
</tr>
</tbody>
</table>

Encoder-C is performed by message passing mechanism such that its time complexity is $O(|V| + |E|)$ (Chiang, Wei-Lin et.al., 2019 KDD)-linear in the number of network edges $|E|$ and nodes $|V|$, where we omit hyperparameter term. Encoder-S is $O(|P|)$-linear in the point amount of one neuron. Practical training times are shown in Table 5. The skeleton part has more data than connectome data such that Encoder-S dominates the run time. On HemiBrain, NeuNet is fast that one epoch takes about 77 s, and inferring a single sample takes about 0.0044 s. (tested on RTX 2080Ti).

Table 5: Run time (Seconds) under different scales of training scales.

<table>
<thead>
<tr>
<th>Scale</th>
<th>0.1</th>
<th>0.3</th>
<th>0.5</th>
<th>0.7</th>
<th>0.9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encoder-C</td>
<td>26.85</td>
<td>30.79</td>
<td>34.94</td>
<td>38.53</td>
<td>43.84</td>
</tr>
<tr>
<td>Encoder-S</td>
<td>329.35</td>
<td>617.90</td>
<td>916.42</td>
<td>1579.35</td>
<td>1513.44</td>
</tr>
<tr>
<td>NeuNet</td>
<td>356.88</td>
<td>649.78</td>
<td>952.43</td>
<td>1619.00</td>
<td>1558.36</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Scale</th>
<th>0.1</th>
<th>0.3</th>
<th>0.5</th>
<th>0.7</th>
<th>0.9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encoder-C</td>
<td>8.03</td>
<td>9.37</td>
<td>11.75</td>
<td>13.49</td>
<td>15.33</td>
</tr>
<tr>
<td>Encoder-S</td>
<td>271.48</td>
<td>503.12</td>
<td>723.85</td>
<td>946.36</td>
<td>1159.55</td>
</tr>
<tr>
<td>NeuNet</td>
<td>280.57</td>
<td>512.99</td>
<td>736.70</td>
<td>961.91</td>
<td>1176.94</td>
</tr>
</tbody>
</table>

Conclusion

In this paper, we propose a framework called NeuNet specifically for the neuron classification task, which incorporates both the morphological information of neuronal skeleton and the topological information of brain circuit with permutation invariance. NeuNet’s Skeleton Encoder mines the morphological information of neuronal skeleton in a bottom-up manner, while NeuNet’s Connectome Encoder employs a GNN to extract the topological information of brain circuit. The two information are fused and utilized by the Readout Layer for classification. Furthermore, we reprocess and release two datasets for neuron classification task. On these datasets, NeuNet demonstrate superior performance compared to other baseline models.
Acknowledgements

This work is partially supported by National Key RD Program of China (2023YFC2705700), the National Natural Science Foundation of China (62206202, 62225113), China Postdoctoral Science Foundation, China (2022M712461), Artificial Intelligence Innovation Project of Wuhan Science and Technology Bureau (No. 2022010702040070), The Fundamental Research Funds for the Central Universities, China (2042022kf1043).

References


