Self-Supervised Contrastive Graph Views for Learning Neuron-level Circuit Network

Junchi Li[1,2,3], Guojia Wan*[1,2,3], Minghui Liao[1,2,3], Fei Liao[4], and Bo Du*[1,2,3]

[1] National Engineering Research Center for Multimedia Software, School of Computer Science, Wuhan University, China
[2] Institute of Artificial Intelligence, Wuhan University, China
[3] Hubei Key Laboratory of Multimedia and Network Communication Engineering, Wuhan University, China
[4] Department of Gastroenterology, Renmin Hospital of Wuhan University
{junchili, guojiawan, minghui, feiliao, dubo}@whu.edu.cn

Abstract. Learning Neuron-level Circuit Network can be used on automatic neuron classification and connection prediction, both of which are fundamental tasks for connectome reconstruction and deciphering brain functions. Traditional approaches to this learning process have relied on extensive neuron typing and labor-intensive proofread. In this paper, we introduce FlyGCL, a self-supervised learning approach designed to automatically learn neuron-level circuit networks, enabling the capture of the connectome's topological feature. Specifically, we leverage graph augmentation methods to generate various contrastive graph views. The proposed method differentiates between positive and negative samples in these views, allowing it to encode the structural representation of neurons as adaptable latent features that can be used for downstream tasks such as neuron classification and connection prediction. To evaluate our method, we construct two new Neuron-level Circuit Network datasets, named HemiBrain-C and Manc-C, derived from the FlyEM project. Experimental results show that FlyGCL attains neuron classification accuracies of 73.8% and 57.4%, respectively, with >0.95 AUC in connection prediction tasks. Our code and data are available at GitHub Repositoryhttps://github.com/mxz12119/FlyGCL.

Keywords: Neuron-level Circuit Network · Connectome · Neuron classification · Neuronal Connection Prediction

1 Introduction

Recently, with the rapid development of deep learning [11,26,14,10] and volumetric microscopy (VEM) technology, we have been available to access synaptic-level connectivity of the whole brain in small model animals such as fruit fly[4], zebrafish[20], etc. This has brought new hope and opportunities for deciphering nervous systems.

A connectome can be viewed as a complex network of more than tens of millions or more neurons and supporting cells that work together to carry out its

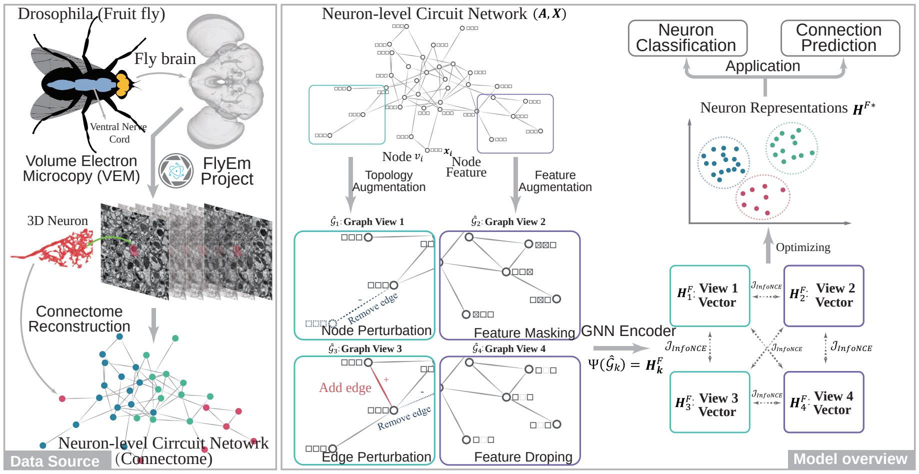
Drosophila (Fruit fly)
Fly brain
Ventral Nerve Cord
Volume Electron Microcopy (VEM)
FlyEm Project
3D Neuron
Connectome Reconstruction
Neuron-level Cirrcuit Netowrk (Connectome)
Data Source
Neuron-level Circuit Network $(\boldsymbol{A},\boldsymbol{X})$
Node v_i
$\boldsymbol{x}_i$
Node Feature
Topology Augmentation
Feature Augmentation
$\hat{\mathcal{G}}_1$: Graph View 1
$\hat{\mathcal{G}}_2$: Graph View 2
$\hat{\mathcal{G}}_3$: Graph View 3
$\hat{\mathcal{G}}_4$: Graph View 4
Remove edge
Node Perturbation
Feature Masking
Add edge
Remove edge
Edge Perturbation
Feature Droping
GNN Encoder
$\Psi(\hat{\mathcal{G}}_k) = \boldsymbol{H}^F_k$
$\boldsymbol{H}^F_1$: View 1 Vector
$\boldsymbol{H}^F_2$: View 2 Vector
$\boldsymbol{H}^F_3$: View 3 Vector
$\boldsymbol{H}^F_4$: View 4 Vector
$\mathcal{J}_{InfoNCE}$
Optimizing
Neuron Representations $\boldsymbol{H}^{F*}$
Application
Neuron Classification
Connection Prediction
Model overview

Fig. 1: The left frame depicts the data source, which represents a connectome in the terms of neuron-level circuit network that is reconstructed from VEM images of a nervous system of a male adult *Drosophila*. The right frame illustrates our model architecture, where such neuron-level circuit networks, augmented through various views, are subjected to contrastive learning to obtain graph embeddings. These embeddings can subsequently be utilized for neuron classification and connection prediction.

functions [17]. Neurons create complex modules within neural circuits and connectomes. Revealing the pattern of neuronal types and connections throughout connectivity is an effective method for comprehending the brain's structural and functional organization. Recent studies in neuroscience [9,1] have underscored the critical necessity for new methods to conduct automatic neuron cell typing (Neuron classification [15,27]) and connection prediction as fundamental tasks for analyzing the connectome.

In recent years, there has been a significant acceleration in the development of Graph Neural Networks (GNNs)[13,24,6]. These methods provide a powerful framework for learning continuous vector representations enabling to analyze network data. However, GNNs concentrate on semi-supervised learning, necessitating a substantial quantity of labels. Recent advancements in Self-Supervised Learning (SSL)[3] have catalyzed an increase in research focused on acquiring graph representations [23] without relying on human annotations. Specifically, manual proofreading still plays a relatively important role in existing neuron cell typing and connection reconstruction [5].

In this paper, we introduce **FlyGCL**, a self-supervised learning approach applied to fruit fly connectome, leveraging augmented contrastive graph views. FlyGCL is designed to learn the topological representation of graphs, enabling effective neuron classification and connection prediction. By leveraging node perturbation, edge perturbation and feature mask, we create various contrastive graph views through random modifications of the initial input. It then acquires representations by differentiating between positive and negative samples. In this way, the model can learn the topological features of the network without needing to label the categories of neuron types. This method can effectively capture the features of nodes using only partial connection information. The learned node feature representations are versatile. We conducted extensive experiments to prove that these features can be effectively applied to tasks such as neuron classification, connection prediction, neuron clustering.

Our contributions:

- We propose FlyGCL, a self-supervised neuron-level circuit network learning model, filling the application gap between graph learning and brain wiring diagram study.
- We built Hemibrain-C and Manc-C datasets from FlyEM project [5]. Furthermore, we experimentally verified that FlyGCL is a powerful feature extractor for neuron classification and neuron connection prediction.

2 Method

Annotation We consider neuron-level circuit network in the brain as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ represents its node set and $\mathcal{E}$ represents the set of edges. Each edge $(\upsilon_i, \upsilon_j) \in \mathcal{E}$ denotes a connection from the neuron υ_i to the neuron υ_j. The adjacency matrix of $\mathcal{G}$ is denoted by $A \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$, where $A_{ij} = 1$ if (υ_i, υ_j) is connected, otherwise $A_{ij} = 0$.

2.1 Overview

FlyGCL is schematically presented in Figure 1.

- **Data source** Thanks to the great contribution from FlyEM project[5], we can access two connectomes, Hemibrain and Manc. These connectomes have rich information about neuron morphology, synaptic connections, mitochondria distribution, and cell category. We fetched these connectomes from the publicly available database NeuPrint[16]. Specifically, Hemibrain covers half of the central fly brain with 2500 neurons and 20 million connections[18] [6].

[5] https://www.janelia.org/project-team/flyem

[6] Version:'hemibrain 1.2.1'

Dataset	$\|\mathcal{V}\|$	$\|\mathcal{E}\|$	Class	Average degree
HemiBrain-C	21739	3550403	25	326.64
Manc-C	23188	5243574	7	452.30

Table 1: Dataset information.

Manc is from the ventral nerve cord of a male fruit fly, including 2300 neurons and 84 million synaptic connections[21]. To study learning the topology nature of connectome, we only keep the connectivity information and manufactured Neuron-level Circuit Networks, resulting two practical datasets, Hemibrain-C and Manc-C. The detailed statistic information is included in 1.

- **Brief introduction of FlyGCL** Firstly, Various graph views are generated by topology augmentation and feature augmentation (Section 2.2). Then we use GNN encoder to learn the graph view representations (Section 2.3). Next, we use InfoNCE loss to optimize the architecture and subsequently obtain neuron representations. Finally, these neuron representations can be used for downstream neuron classification and connection prediction (Section 2.4).

2.2 Generating Graph View

To perform contrastive learning, We generate various graph views by graph augmentation. The augmented graph $\hat{\mathcal{G}}$ is formulated as $\hat{\mathcal{G}} = \psi(\mathcal{G})$, where ψ is an augmentation function. We augment the given graph to obtain two correlated views $\hat{\mathcal{G}}_i$, $\hat{\mathcal{G}}_j$, as a positive pair, where $\hat{\mathcal{G}}_i = \psi_i(\mathcal{G})$, $\hat{\mathcal{G}}_j = \psi_j(\mathcal{G})$ respectively. These augmentations include Node Perturbation (NP), Edge Perturbation (EP), Feature Masking (FM) and Feature Dropping (FD). All of them are schematically shown in Figure 1(right).

- **Node Perturbation** Node Perturbation primarily considers removing node. For each node, there is a probability p of being randomly dropped.
- **Edge Perturbation** Randomly adds or removes a subset of edges within a graph. Formally, we can write it as $M_{i,j} \sim \mathcal{B}(p)$, where $\mathcal{B}$ denotes a Bernoulli distribution with parameter p. The resulting adjacency matrix $\hat{\mathbf{A}}$ can be computed as $\hat{\mathbf{A}} = \mathbf{A} \bigodot \mathbf{M}$, where $\bigodot$ denotes a bit-wise operator. Frequently used function Edge Removing (ER) and Edge Adding (EA) are implemented through logical add and logical subtract operations, respectively.
- **Feature Masking(FM)** Randomly masks features of nodes with a Bernoulli distribution with probability p. Formally, that means $\mathbf{x}'_i = \mathbf{x}_i \bigodot \mathbf{M}$, where $\mathbf{x}_i$ is the feature vector of υ_i. $\mathcal{M} \sim \mathcal{B}(p) \in \{0, 1\}^d$.
- **Feature Dropping(FD)** Applies column-wise dropout to node feature matrix $\mathbf{X}$, where each column has a probability p of being randomly masked to zero.

2.3 Learning Contrastive Graph Views

GNN Encoder To learn the latent representations of various generated graph views, we use GNN encoder to aggregate neighbour information iteratively. The input data is the feature matrix $\mathbf{X} \in \mathbb{R}^{N \times d}$. Firstly, a linear layer is employed to project $\mathbf{X}$ into a continuous feature matrix $\mathbf{H}^{(0)} = \mathbf{X} \cdot \mathbf{T}$, where $\mathbf{T} \in \mathbb{R}^{m \times d^{(0)}}$ denotes type embedding, d is the feature dimension. And then we will feed the graph adjacency matrix $\mathbf{A}$ and the feature matrix $\mathbf{H}^{(0)}$ into GNN layer to learn the next latent features:

$$\mathbf{H}^{(l+1)} = ReLU(GNN(\mathbf{A}, \mathbf{H}^{(l)}) + \mathbf{H}^{(l)}), \mathbf{H}^{F} = MLP(\mathbf{H}^{(l+1)}), \qquad (1)$$

where $GNN(\cdot)$ denotes a conventional GNN encoder, such as GCN [13], GIN[24] etc. Here, we also use skip connection and another two layer MLP as a projection head to improve the learning capability. Subsequently, we can obtain $\mathbf{H}^{F}$ as the total graph view representations locating in the implicit feature space.

Contrast Loss Function A contrastive loss function is defined to enforce the maximization of consistency between positive pairs (i.e., two enhanced views of the same graph) while minimizing the consistency between negative pairs (i.e., enhanced views of different graphs). The function used here is denoted as function Information Noise Contrastive Estimation (InfoNCE) [12]:

$$\mathcal{J}_{InfoNCE}(\upsilon_i) = -\frac{1}{N} \sum_{p_j \in \mathcal{P}(\upsilon_i)} \log \frac{e^{\theta(\upsilon_i, p_j)/\tau}}{e^{\theta(\upsilon_i, p_j)/\tau} + \sum_{q_j \in \mathcal{Q}(\upsilon_i)} e^{\theta(\upsilon_i, q_j)/\tau}}, \qquad (2)$$

where υ_i denotes the node embedding indexed from $\mathbf{H}^{F}$. $\mathcal{P}(\upsilon_i)$ denotes the positive set $\mathcal{P}(\upsilon_i) = \{p_i\}_{i=1}^{P}$, and $\mathcal{Q}(\upsilon_i)$ denotes the negative set $\mathcal{Q}(\upsilon_i) = \{q_i\}_{i=1}^{Q}$. $\theta(\cdot, \cdot)$ is cosine similarity measuring the similarity between two embeddings:

$$\theta(u, \upsilon) = \frac{u^{\top} \upsilon}{\parallel u \parallel \parallel \upsilon \parallel}.$$

2.4 Downstream Task

Neuron Classification After contrastive training, the optimized feature vector $\mathbf{H}^{F*}$ is obtained for downstream tasks. For neuron classification with C classes. We use a two fully-connected layers to re-trained on partial given labels. Then our method performs neuron classification the i-th neuron using sigmoid function σ as

$$\mathbf{y} = \sigma(FC(\mathbf{f}_i)), \mathbf{f}_i \in \mathbf{H}^{F*}, \mathbf{y} \in [0, 1]^{C}. \qquad (3)$$

Connection Prediction. In the task of connection prediction, we use (υ_i, υ_j) to denote a connection from υ_i to υ_j. Therefore, we concatenate $\mathbf{f}_i$ of υ_i and $\mathbf{f}_j$ of υ_j into a connection vector. We also re-train a two fully-connected layers to predict the plausibility whether (υ_i, υ_j) is connected:

$$\mathbb{P}(A_{i,j} = 1 | \theta) = \sigma(FC([\mathbf{f}_i; \mathbf{f}_j])), \mathbf{f}_i, \mathbf{f}_j \in \mathbf{H}^{F*}, \qquad (4)$$

where ; denotes vector concatenation.

	Method	10%	20%	30%	40%	50%
Hemibrain-C	FlyGCL	**73.80 ± 0.45**	**78.03 ± 0.30**	**78.86 ± 0.21**	**79.40 ± 0.13**	**79.87 ± 0.18**
	GCN	67.96 ± 0.08	69.84 ± 0.14	70.98 ± 0.15	72.06 ± 0.24	73.14 ± 0.22
	GIN	61.63 ± 0.07	64.45 ± 0.12	66.53 ± 0.09	67.67 ± 0.42	68.64 ± 0.30
	Node2Vec	62.49 ± 0.38	64.52 ± 0.20	65.20 ± 0.35	66.27 ± 0.74	67.38 ± 0.35
	GraphSAGE	70.57 ± 0.07	74.20 ± 0.12	76.21 ± 0.23	77.62 ± 0.16	78.72 ± 0.11
	JKNet	68.15 ± 0.76	70.26 ± 0.15	72.14 ± 0.20	73.62 ± 0.27	74.66 ± 0.26
	GATV2	56.46 ± 0.73	60.53 ± 0.84	62.30 ± 0.94	63.63 ± 0.72	64.63 ± 0.12
	UniMP	34.03 ± 1.68	30.41 ± 3.53	29.76 ± 2.90	29.63 ± 2.71	29.05 ± 2.57
Manc-C	FlyGCL	**57.40 ± 0.90**	**58.90 ± 1.17**	**59.25 ± 0.26**	**60.96 ± 1.37**	**62.28 ± 1.51**
	GCN	41.64 ± 0.23	45.88 ± 0.09	47.90 ± 0.18	49.56 ± 0.65	50.27 ± 0.63
	GIN	44.54 ± 0.54	46.03 ± 0.12	46.33 ± 0.04	46.11 ± 0.73	46.88 ± 1.02
	Node2Vec	43.65 ± 2.15	43.74 ± 1.47	43.93 ± 1.73	45.20 ± 2.24	46.42 ± 1.98
	GraphSAGE	42.44 ± 1.76	43.03 ± 1.32	43.90 ± 1.41	44.76 ± 1.29	46.39 ± 0.21
	JKNet	38.08 ± 1.12	38.12 ± 0.65	40.44 ± 1.90	40.65 ± 1.84	44.09 ± 2.90
	GATV2	45.69 ± 0.46	46.76 ± 0.53	47.29 ± 0.47	46.81 ± 0.57	47.05 ± 1.22
	UniMP	52.45 ± 1.54	44.31 ± 7.98	41.95 ± 9.16	42.33 ± 9.47	43.32 ± 9.91

Table 2: Neuron classification F1(Micro) results on HemiBrain-C and Manc-C. Bold indicates the best performance in a column. Each number is the average performance for 10 random runs of the experiments. 10%-50% are the sizes of various training graph.

3 Experiments and Discussion

Neuron classification For comparison with other methods, we use the following representative approaches including Graph Embedding (Node2Vec [7]), GNN (GCN[13], GIN[24], GraphSAGE[8], JKNet [25]), and the recent SOTA methods (GATV2[2], UniMP[19]). We use F1(Micro) as metric to report neuron classification results with 25 classes on HemiBrain-C and 7 classes on Manc-C. As shown in Tables 2, it demonstrates that FlyGCL performs good results. We get a score of 78.03% on the HemiBrain-C dataset using a training ratio of 20 % edges, which is better than other baselines. Similarly, our method performs best on the Manc-C dataset. Figure 2 visually shows how our method performed, with a clear line indicating that it did well at identifying neuron types.

Connection prediction On HemiBrain-C and Manc-C, our method is compared with heuristic methods[7] (Adamic-Adar (AA), Common Neighbour (CN)), GCN[13], and Node2vec [7] for link prediction, with results presented in Figure 3. We report ROC and PR curves for evaluating the capability of predicting connections between two neurons. Accordingly, we also plot AUC bars from ROC curves. As depicted in Figure 3, FlyGCL exhibits the largest area enclosed by the PR and ROC curves, indicating its superior capability in connection prediction.

[7] Implemented by NetworkX package: https://networkx.org/.

The AUC values further substantiate this, with FlyGCL achieving the highest performance, reaching 0.971 and 0.959 across both datasets. Furthermore, to investigate the scaling performance, we report the training size of the circuit networks versus connection prediction performance in Figure 4.

Corresponding results show that our method achieves great performance even using <5% training edges, yielding a rapid performance increase in AUC as the train ratio increases, surpassing 0.9 at a train ratio of 0.03.

Latent Feature Visualization The learned latent feature H^F from 20% ratio on Hemibrain-C is reduced into two-dimensional coordinates by tSNE[22], resulting in a clustering visualization (Figure 5). Distinct clusters are evident, with crisp boundaries and increased inter-cluster distances. This indicates that our model effectively learned implicit features through the proposed contrastive framework on such circuit network.

Ablation Analysis

- **Effectiveness of base encoder** To measure the contribution of different base encoders, we have modified the encoder layer in the collection of (GCN, GIN, GCN+Res). We report their accuracy for neuron classification on the HemiBrain-C and Manc-C datasets. As shown in Table 6, GCN+Res achieves 1% higher accuracy on the HemiBrain-C dataset compared to the other two methods, and a 3-5% increase on the Manc-C dataset. Therefore, we chose GCN+Res as the default encoder of FlyGCL.
- **Impact of different graph augmentations** We have also altered the augmentation methods to generate different views. The original functions were Edge Perturbation (EP) and Feature Masking (FM). We employed various combinations of these functions for graph augmentation. The experimental results, as depicted in Figure 5, indicate that EP significantly improves the outcome, particularly on the Manc-C dataset, with the highest accuracy achieved at edge removal probabilities of 0.5-0.6%. Node Perturbation(NP) enhances model performance when the drop probability is less than 0.5, after which the accuracy noticeably declines as the probability increases.

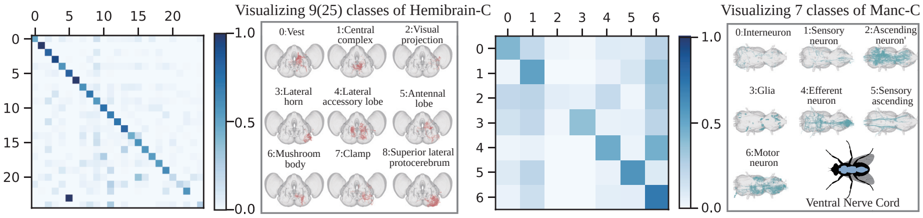

Fig. 2: Confusion matrix (HemiBrain-C (left), Manc-C(right)) on Neuron classification. The rest 16 classes of Hemibrain-C are listed in supplementary materials.

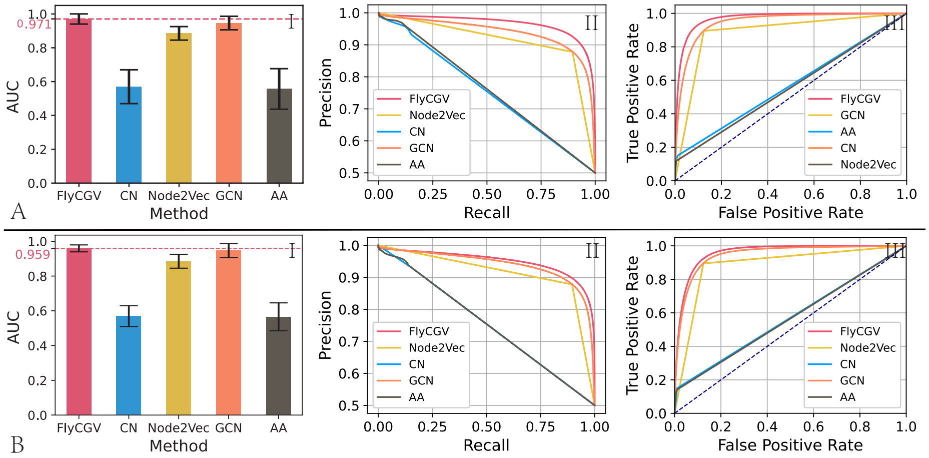

Fig. 3: Neuron connection prediction results. A(I,II,III):The AUC, ROC curves, and PR curves of HemiBrain-C. B(I,II,III):The AUC, ROC curves, and PR curves of Manc-C.

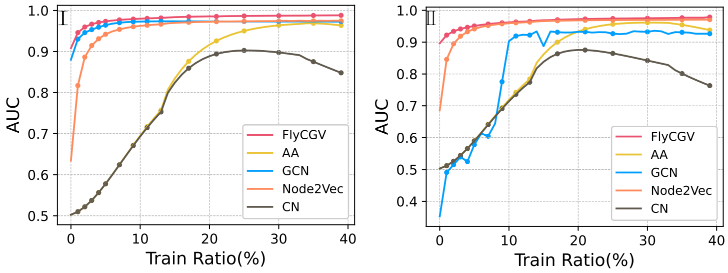

Fig. 4: Train ratio versus AUC performance:I) HemiBrain-C, II) Manc-C.

4 Conclusion

In this paper, we introduced a self-supervised graph learning approach, **FlyGCL**, which captures neuronal features just through the topological structure of neuron-level circuit networks, circumventing the necessity for a substantial quantity of labels. The neuronal features acquired by our model are efficaciously applicable to tasks such as neuron connection prediction and neuron classification. Compared to existing baseline models, our model demonstrates superior performance, while concurrently requiring a more modest amount of training data. These applications underscore the potential of our methodology as a promising tool for connectome data analysis, facilitating more efficient and precise elucidation of the nervous system. This, in turn, contributes to a deeper comprehension of the brain's structural and functional organization.

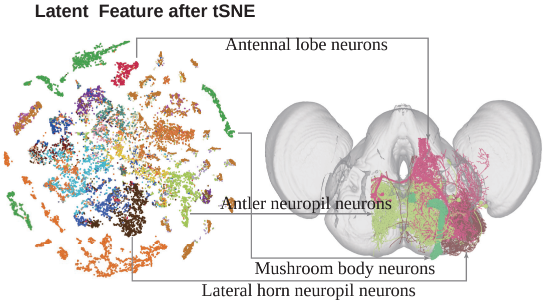

Fig. 5: tSNE visualization of the learned latent features $\mathbf{H}^{F*}$from FlyGCL on 20% Hemibrain-C.

Base Encoder	HemiBrain-C	Manc-C
GCNConv	73.76 ± 0.23	52.73 ± 0.49
GATConv	73.54 ± 0.14	55.38 ± 0.53
GCNConv+Res	74.23 ± 0.11	58.86 ± 0.68

Fig. 6: Performance of various GNN encoders on HemiBrain-C and Manc-C.

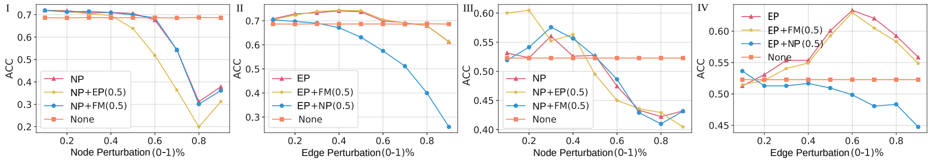

Fig. 7: Ablation study (HemiBrain-C (I, II), Manc-C(III, IV)), the impact of employing various graph view generation methods (NP, FM, EP, no-augmentation).

Acknowledgments. This work is partially supported by National Key RD Program of China (2023YF C2705700), NSFC (62206202, 62225113), the Innovative Research Group Project of Hubei Province under Grants 2024AFA017, AI Innovation Project of Wuhan Science and Technology Bureau (No. 2022010702040070).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Markus Axer and Katrin Amunts. Scale matters: the nested human connectome. *Science*, 378(6619):500–504, 2022.
2. Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? In *International Conference on Learning Representations*, 2021.
3. Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 22243–22255, 2020.
4. Sven Dorkenwald, Arie Matsliah, Amy R Sterling, Philipp Schlegel, Szi-Chieh Yu, Claire E McKellar, Albert Lin, Marta Costa, Katharina Eichler, Yijie Yin, et al. Neuronal wiring diagram of an adult brain. *bioRxiv*, 2023.
5. Sven Dorkenwald, Claire E McKellar, Thomas Macrina, Nico Kemnitz, Kisuk Lee, Ran Lu, Jingpeng Wu, Sergiy Popovych, Eric Mitchell, Barak Nehoran, et al. Fly-

wire: online community for whole-brain connectomics. *Nature Methods*, 19(1):119–128, 2022.

6. Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International Conference on Machine Learning*, pages 1263–1272, 2017.
7. Aditya Grover and Jure Leskovec. Node2vec: Scalable feature learning for networks. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 855–864, 2016.
8. Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in Neural Information Processing Systems*, 30, 2017.
9. Julie A Harris, Stefan Mihalas, Karla E Hirokawa, Jennifer D Whitesell, Hannah Choi, Amy Bernard, Phillip Bohn, Shiella Caldejon, Linzy Casal, Andrew Cho, et al. Hierarchical organization of cortical and thalamic connectivity. *Nature*, 575(7781):195–202, 2019.
10. Larissa Heinrich, Jan Funke, Constantin Pape, Juan Nunez-Iglesias, and Stephan Saalfeld. Synaptic cleft segmentation in non-isotropic volume electron microscopy of the complete drosophila brain. In *Medical Image Computing and Computer Assisted Intervention*, pages 317–325. Springer, 2018.
11. Michał Januszewski, Jörgen Kornfeld, Peter H Li, Art Pope, Tim Blakely, Larry Lindsey, Jeremy Maitin-Shepard, Mike Tyka, Winfried Denk, and Viren Jain. High-precision automated reconstruction of neurons with flood-filling networks. *Nature Methods*, 15(8):605–610, 2018.
12. Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 18661–18673, 2020.
13. Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.
14. Hanyu Li, Michał Januszewski, Viren Jain, and Peter H Li. Neuronal subcompartment classification and merge error correction. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 88–98. Springer, 2020.
15. Minghui Liao, Guojia Wan, and Bo Du. Joint learning neuronal skeleton and brain circuit topology with permutation invariant encoders for neuron classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024.
16. Stephen M Plaza, Jody Clements, Tom Dolafi, Lowell Umayam, Nicole N Neubarth, Louis K Scheffer, and Stuart Berg. neuprint: an open access tool for em connectomics. *Frontiers in Neuroinformatics*, 16, 2022.
17. Dale Purves, George J Augustine, David Fitzpatrick, William Hall, Anthony-Samuel LaMantia, and Leonard White. *Neurosciences*. De Boeck Supérieur, 2019.
18. Louis K Scheffer, C Shan Xu, Michal Januszewski, Zhiyuan Lu, Shin-ya Takemura, Kenneth J Hayworth, Gary B Huang, Kazunori Shinomiya, Jeremy Maitlin-Shepard, Stuart Berg, et al. A connectome and analysis of the adult drosophila central brain. *Elife*, 9:e57443, 2020.
19. Yunsheng Shi, Zhengjie Huang, Shikun Feng, Hui Zhong, Wenjin Wang, and Yu Sun. Masked label prediction: Unified message passing model for semi-supervised classification. In *International Joint Conference on Artificial Intelligence*, 2020.

20. Fabian Svara, Dominique Förster, Fumi Kubo, Michał Januszewski, Marco Dal Maschio, Philipp J Schubert, Jörgen Kornfeld, Adrian A Wanner, Eva Laurell, Winfried Denk, et al. Automated synapse-level reconstruction of neural circuits in the larval zebrafish brain. *Nature Methods*, 19(11):1357–1366, 2022.
21. Shin-ya Takemura, Kenneth J Hayworth, Gary B Huang, Michal Januszewski, Zhiyuan Lu, Elizabeth C Marin, Stephan Preibisch, C Shan Xu, John Bogovic, Andrew S Champion, et al. A connectome of the male drosophila ventral nerve cord. *bioRxiv*, pages 2023–06, 2023.
22. Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(11), 2008.
23. Dongkuan Xu, Wei Cheng, Dongsheng Luo, Haifeng Chen, and Xiang Zhang. Infogcl: Information-aware graph contrastive learning. In *Advances in Neural Information Processing Systems*, volume 34, pages 30414–30425, 2021.
24. Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2018.
25. Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. Representation learning on graphs with jumping knowledge networks. In *International Conference on Machine Learning*, pages 5453–5462, 2018.
26. Chi Zhang, Qihua Chen, and Xuejin Chen. Self-supervised learning of morphological representation for 3d em segments with cluster-instance correlations. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 99–108. Springer, 2022.
27. Tianfang Zhu, Gang Yao, Dongli Hu, Chuangchuang Xie, Pengcheng Li, Xiaoquan Yang, Hui Gong, Qingming Luo, and Anan Li. Data-driven morphological feature perception of single neuron with graph neural network. *IEEE Transactions on Medical Imaging*, 2023.