Appendix

A Evaluation metric

Macro-F_1: Macro-F_1 is a macro-average F_1 score, calculated by averaging F_1 scores of individual classes. This approach is beneficial for imbalanced datasets, as it equally weighs each class's performance. Precision is the ratio of true positives (TP) to false positives (FP), while recall is the ratio of true positives to false negatives (FN). The $F1$ score is the harmonic mean of precision and recall, calculated using the formula:

$$F_1 = 2 \times \frac{Presicion \times Recall}{Presicion + Recall} \tag{5}$$

After aggregating the F_1 scores for each class, the mean is taken to yield the Macro-F_1 score.
Micro-F_1: We also employed Micro-F_1 as one of the metrics for evaluating model performance. Micro-F_1 aggregates the predicted outcomes across all classes to compute the overall precision and recall, yielding a unified F_1 score. This approach is particularly applicable when equal importance is assigned to the predictive results of all classes. The computation of Micro-F_1 involves constructing a global confusion matrix that accumulates the true positives (TP), false positives (FP), and false negatives (FN) for all classes.

Connection prediction AUC:Connection prediction is a binary classification issue, so we adopts the Area Under the Curve (AUC) as a key metric for evaluating model performance. AUC is derived from the Receiver Operating Characteristic (ROC) curve, which illustrates the relationship between the True Positive Rate (TPR) and the False Positive Rate (FPR) across all possible classification thresholds. The AUC value ranges from 0 to 1, with values closer to 1 indicating superior classification performance.

B Experiment settings

Hardware All experiments are conducted on GPU as NVIDIA GeForce RTX 3080Ti (12 GB), CPU as Intel(R) Xeon(R) CPU E5-2620 2.10GHz.

C Supplementary Results

Table S2 is the another part of results on neuron classsfication.

Tool Name	Role	Version	Source
NeuPrint	Fetch connectome data	1.62.8	neuprint.janelia.org
Navis	Processing connectome data	1.5.0	navis.readthedocs.io
Neuroglancer	Visualize VEM data	-	neuroglancerhub.github.io
PyTorch	Build FlyGCL model	2.1	pytorch.org
PyTorch Geometric	Build FlyGCL model	2.0.5	pytorch-geometric.readthedocs.io
NetworkX	Implement AA, CN	3.2.1	networkx.org

Table S1: Used tools in our work.

	Method	10%	20%	30%	40%	50%
HemiBrain-C	FlyGCL	**52.64 ± 0.49**	**58.87 ± 0.64**	**60.28 ± 0.34**	**60.99 ± 0.62**	**61.48 ± 0.73**
	GCN	46.10 ± 0.41	50.18 ± 0.35	52.31 ± 0.64	53.65 ± 0.48	55.09 ± 0.59
	GIN	37.36 ± 0.25	41.84 ± 0.72	45.74 ± 0.62	47.21 ± 0.60	48.81 ± 0.37
	Node2Vec	32.65 ± 1.83	35.30 ± 1.70	34.31 ± 1.63	34.84 ± 2.31	35.95 ± 1.56
	GraphSAGE	44.91 ± 0.23	53.65 ± 0.43	54.90 ± 0.77	57.74 ± 0.63	60.08 ± 0.86
	JKNet	41.64 ± 0.78	46.90 ± 0.44	50.85 ± 0.41	52.74 ± 0.53	54.44 ± 0.47
	GATV2	32.76 ± 0.73	36.33 ± 0.88	38.56 ± 0.80	40.26 ± 1.10	41.77 ± 0.21
	UniMP	12.34 ± 1.42	10.43 ± 1.36	10.14 ± 1.27	10.13 ± 0.89	9.85 ± 0.97
Manc-C	FlyGCL	**22.43 ± 1.10**	**23.60 ± 0.89**	**23.73 ± 0.24**	**23.87 ± 0.96**	**24.26 ± 0.39**
	GCN	10.54 ± 0.25	13.61 ± 0.07	15.27 ± 0.19	16.23 ± 0.21	16.94 ± 0.70
	GIN	10.84 ± 0.16	11.81 ± 0.12	12.84 ± 0.27	13.34 ± 0.21	14.15 ± 0.16
	Node2Vec	9.54 ± 0.89	10.03 ± 0.96	10.14 ± 1.02	10.48 ± 1.16	10.92 ± 1.37
	GraphSAGE	7.90 ± 0.26	8.06 ± 0.38	8.45 ± 0.48	8.49 ± 0.45	8.61 ± 0.54
	JKNet	9.75 ± 0.25	9.59 ± 0.29	9.94 ± 0.32	10.07 ± 0.13	10.28 ± 0.05
	GATV2	9.41 ± 0.30	9.73 ± 0.31	10.48 ± 0.32	10.55 ± 0.06	10.85 ± 0.15
	UniMP	6.44 ± 0.41	5.93 ± 0.56	5.75 ± 0.52	5.76 ± 0.84	5.84 ± 0.75

Table S2: Neuron classification results on HemiBrain-C and Manc-C. Macro F1. Bold indicates the best performance in a column. Each number is the average performance for 10 random runs of the experiments. 10%-50% is training ratio.

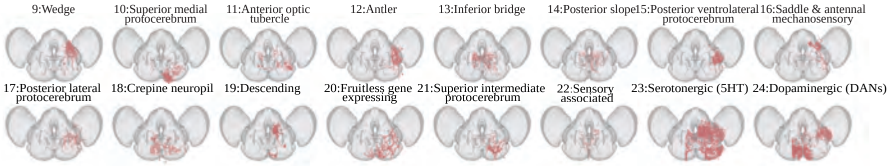

Fig. S1: The rest 16 classed of Hemibrain-C.