

Supplementary Materials for

Deep Clustering Representation for Spatially Resolved Transcriptomics Data via Multi-view Variational Graph Auto-Encoders with Consensus Clustering

Supplementary Notes

1. Comparison with other spatial domain identification methods parameter settings

We quantitatively compared STMVGAE with other methods on different datasets, including the non-spatial method SCANPY [1], and the spatial methods stlearn [2], SEDR [3], SpaGCN [4], DeepST [5], STAGATE [6] and STAMaker [7]. The parameter settings of these methods are as follows:

- **SCANPY:** First, we used the same data preprocessing method as STMVGAE to preprocess gene expression (log-transformed, normalized and selecting the top 3,000 HVGs). PCA dimensionality reduction was then used to reduce the gene expression data to 30 PCs. Finally, we used the `scipy.pp.neighbors()` function default parameters provided by the SCNAPE package [1] to calculate neighbors, and the `scipy.tl.louvain()` function is used to allocate spots. Additionally, the resolution parameter was tuned manually to ensure the number of clustering is equal to the ground truth.
- **stlearn:** We chose default parameters for stlearn on the DLPFC dataset. Specifically, the `stLearn.SME_normalized()` function was performed on the raw gene expression of all genes with the parameter `use_data="raw"` and `weights="physical_distance"`. Then the first 30 PCs of the SME normalized matrix were used for clustering. We did not use stlearn for training on the melanoma dataset because it does not support training without histology images.
- **SEDR:** SEDR can be trained on all datasets, and we retain all its default parameters except for empirically selecting the number of neighbors on different datasets to ensure reasonable results. We perform the same strategy on each dataset, looking for the number of neighbors that gives the best results between 6 and 12 neighbors. We set `n` in the `SEDR.graph_construction()` function to 10 on the DLPFC dataset and to 12 on all other datasets.
- **SpaGCN:** We use its recommended parameters for SpaGCN in all datasets.
- **DeepST:** We retain all the default parameters of DeepST and set `k` in the `deepen._get_graph()` function to 12. Additional, We tested the results on the melanoma dataset with DeepST set up without using histological images.
- **STAGATE:** STAGATE builds the graph by looking for neighbors within a radius, so the parameter `r` in the `STAGATE.Cal_Spatial_Net()` function changes in each dataset. We used the same rules as SEDR to select `r`. In DLPFC, we used the recommended parameter `r` set to 150, `r` in the BCDC data set to 350, `r` in the melanoma data set to 2, and `r` in the BRCA data set to 300.
- **STAMaker:** Recommended parameters are used in STAMaker, and neighbor selection is consistent with STAGATE. We set `n` to randomly initialize the model in STAMaker to 5.

2. Evaluation metrics of clustering

ARI. The adjusted Rand index (ARI) [8] is a measure of the similarity between two clusterings, and it is an external evaluation index. We introduce ARI to calculate the similarity between the results obtained by STMVGAE spot assignment and manual annotation. The calculation of ARI must first calculate the values of the contingency table. The contingency table contains the following four parts: *TF* is the count of spot pairs classified into the same cluster in both the true and predicted clustering. *TN* is the count of spot pairs classified into different clusters in both the true and predicted clustering. *FN* is the count of spot pairs classified into the same cluster in the true clustering but into different clusters in the predicted clustering. *FP* is the count of spot pairs classified into different clusters in the true clustering but into the same cluster in the predicted clustering. The value range of ARI is between [-1,1]. Generally, the closer the ARI value is to 1, the better the result. The closer the ARI value is to 0, the clustering result is the same as the random clustering result. The calculation method of ARI is based on paired samples. It considers the combination of samples of the same category in different clusters in two clustering results and compares it with random situations. ARI is computed as:

$$ARI = \frac{TP + TN - E}{TP + TN + FP + FN - E} \quad (1)$$

The expected value of the index, denoted as *E*, represents the value that would be obtained if the clustering were entirely random. It is calculated as follows:

$$E = \frac{(TP + FP) \times (TP + FN) + (FN + TN) \times (FP + TN)}{TP + TN + FP + FN} \quad (2)$$

NMI. Normalized Mutual Information (NMI) is an indicator used to evaluate the performance of clustering algorithms. It measures the similarity between two clustering results. The NMI value ranges between [0,1]. The closer the value is to 1, the more similar the two clustering results are, while the closer the value is to 0, the less similar they are. *P* represents the spatial domain clustering result and *T* represents the ground truth clustering labels. Their entropies are denoted as *H(P)* and *H(T)* respectively. NMI has been widely used to evaluate the

performance of spatial domain identification in spatial transcriptomic data analysis [9]. The calculation formula for NMI is as follows:

$$NMI = \frac{MI(P, T)}{\sqrt{H(P)H(T)}} \quad (3)$$

HS. In unsupervised clustering, Homogeneity Score (HS) is a metric used to evaluate clustering results, which measures whether the samples in each cluster belong to the same category [6]. The value of HS ranges from 0 to 1. The closer the value is to 1, the better the clustering result is, that is, each cluster contains samples of the same category. $H(C)$ is the entropy of the true class, which represents the uncertainty of the class distribution of the samples in the data set; $H(C|K)$ is the conditional category entropy of a given clustering result, which represents the uncertainty of the category distribution of the sample when the clustering result is known. The calculation formula for HS is as follows:

$$HS = 1 - \frac{H(C|K)}{H(C)} \quad (4)$$

Purity. In unsupervised clustering, Purity is a metric used to evaluate clustering results, which measures whether the samples contained in each cluster belong to the same category. The value range of Purity is between 0 and 1. The closer the value is to 1, the better the clustering result is, that is, each cluster contains samples of the same category. N is the total number of samples in the dataset, k represents the index of the cluster, j represents the index of the real category, c_k represents the sample set in cluster k , and t_j represents the sample set in real category j . The $|c_k \cap t_j|$ in the formula represents the size of the intersection of samples in cluster k and samples in true category j . The calculation formula for Purity is as follows:

$$Purity = \frac{1}{N} \sum_k \max_j |c_k \cap t_j| \quad (5)$$

3. Downstream analysis of DLPFC dataset

STMVGAE proved effective for both Umap visualization and PAGA trajectory analysis. We visualized the low-dimensional embeddings and presented the spatial trajectories (Figure 2E and Figure S2). Taking the Umap visualization analysis of 151508 slices as an example, stlearn did not make full use of the spatial coordinate information, and the Umap visualization could not assign each class to a reasonable spatial location. There was no obvious boundary between different layers of STAGATE and SpaGCN, different spatial domains of STAGATE were almost squeezed together and SpaGCN formed a ring structure. DeepST and SEDR achieved relatively good results, but the PAGA trajectory of DeepST was disordered, and the layer_2 and layer_3 of SEDR were mixed. In contrast, STMVGAE presented spots that clearly organized the different layers and accurately reflected the developmental sequence of the cortical layer [10], not only did STMVGAE clearly organize the boundaries between each layer, but the spots of the different layers were not located in the other layers.

To further explore downstream tasks, we used the same procedure as SpaGCN [4] to identify SVGs (Figure 2F). We detected a total of 35 SVGs on the 151508 slice, which were dispersed across different domains. These included 4 SVGs in domain 0, 8 SVGs in domain 2, 1 SVG in domain 3, 21 SVGs in domain 4, and 1 SVG in domain 6. We utilized different colors to represent the relative expression levels of related genes, and the different domain SVGs identified by STMVGAE in Figure 2F matched the artificially annotated cortical layer structure in Figure 2A.

Since some neuronal layers were difficult to label with a single gene, we constructed metagenes to label specific domains (Figure 2G). Due to the fewer number of spots in Layer_2, it was difficult to detect genes enriched in this domain. Therefore, we significantly enhanced the expression pattern by increasing genes such as *CXCL14*, *HPCAL1*, *MBP*, etc.

4. Downstream analysis of BRCA dataset

Figure 3D showed the results of Umap visualization of the low-dimensional embeddings of the six methods. There was a mess in stlearn. In the visualizations of SpaGCN and DeepST, there was a discontinuity of spatial domains, and some spatial domains existed separately from the whole. There were no very clear boundaries between the SEDR space domains. STAGATE appeared to have a slight blend of spots between different domains. The Umap visualization of STMVGAE revealed that most of the spots were well organized and had clear boundaries between different domains. With the identified domains, we further identified the SVGs in different spatial domains. We detected a total of 468 SVGs on the BRCA dataset, which were dispersed across different domains. As shown in Figure 3E, in the task of identifying SVGs, STMVGAE accurately identified the *CXCL14* in domain 17, which had been proven to have prognostic significance in breast cancer [11].

5. Downstream analysis of BCDC dataset

To further explore the spatial domains we generated, we performed SVG identification on the BCDC dataset. A total of 1364 SVGs were identified on the BCDC dataset, including 78 SVGs in non-tumor region 1 and 1286 SVGs in tumor region 2. We speculated that there were only two types of BCDC datasets, so there were more SVGs that could be identified by STMVGAE on BCDC dataset (Figure 4E).

6. STMVGAE can accurately identify spatial domains on the Melanoma dataset

We evaluated the performance of STMVGAE on melanoma cancer from Thrane *et al.* [12]. There were three distinct areas in this data: melanoma, stroma, and lymphoid tissue, with an additional unannotated area [13]. We also used four evaluation indicators to measure the performance of STMVGAE in this dataset. The simplified version of STMVGAE performed better than several other comparison methods on this dataset, and STMVGAE achieved significant improvements. STMVGAE attained the highest ARI value of 0.48 among all competing methods, and it was the only one among several comparison methods with a Purity value of more than 0.8 (Table 2).

7. STMVGAE is capable of analyzing high-resolution Olfactory dataset

To further test STMVGAE spatial domain identification, we also tested it on a mouse olfactory bulb cell dataset from the high-resolution Stereo-seq [14] platform. We chose two methods for comparison, SEDR and STAGATE. The performance of these two methods was second only to STMVGAE on the DLPFC dataset, as shown in [Figure S3B](#). STMVGAE could well identify the laminar flow structure of mouse olfactory bulb cell data, and it was consistent with the artificially annotated laminar flow structure [3] ([Figure S3A](#)). The SEDR method was not accurate enough in identifying the rostral migratory stream (RMS) spatial domain. Not only did it not correctly identify the range of this spatial domain, but it also had no clear boundary with the granule cell layer (GCL). The STAGATE method was also not accurate enough in identifying the rostral migratory stream (RMS) spatial domain. It did not identify the spatial domain as a continuous area, and STAGATE was not accurate enough in the outer spatial domain internal plexiform layer (IPL), granule cell layer (MCL), and glomerular layer (GL) spot allocation, which was confusing. In contrast, STMVGAE accurately identified the rostral migratory stream (RMS), and for the granule cell layer (GCL), internal plexiform layer (IPL), granule cell layer (MCL), glomerular layer (GL), and the olfactory nerve layer (ONL) was more accurate ([Figure S3C](#)). We verified our results by detecting marker genes in each layer, and the results showed that Dbi and Fam155a were strongly expressed on the rostral migratory stream (RMS) and the granule cell layer (GCL). Our experimental results were consistent with some previous studies [15, 16]. The above experimental results showed that STMVGAE could process ST data at different spatial resolutions.

8. STMVGAE corrects for batch effects

We compared the performance of STMVGAE, SCANPY, and SEDR in batch data integration processing using the deep method SEDR versus the non-deep method scanpy ([Figure S4A](#) and [Figure S5E](#)). For batch integration, we used Harmony technology [17], which demonstrated superior performance in scRNA-seq. We selected the first four slices in the DLPFC dataset for integration (151507, 151508, 151509, 151510), and the visualization obtained by scanpy was heavily mixed with speckles between different domains. For STMVGAE and SEDR embedding, cells in different cortical layers exhibited a distinct order of separation and development, while SEDR did not identify small areas as clearly as STMVGAE. We also obtained the manual annotation of the joint batch data by integrating the manual annotation of each slice. We calculated the results obtained by three different methods and the manual annotation of four indicators, the NMI values obtained by STMVGAE and SEDR were comparable, and the other three indicators were the highest values obtained by STMVGAE, and the ARI value of STMVGAE was 0.49. Considering that the data in the DLPFC dataset presented different data patterns, we divided the DLPFC dataset according to the data schema for joint batch analysis. In addition, we also performed a joint batch analysis of two other slices with different data patterns on the DLPFC dataset using STMVGAE, as shown in [Figure S4B](#), C, we used STMVGAE to perform a combined batch analysis of the middle four slices (151669, 151670, 151671, 151672) and the last four slices (151673, 151674, 151675, 151676). In the data analysis results of these two joint batches, it could be found that the Umap visualization results obtained by STMVGAE could well isolate cortical layer cells with a clear developmental sequence.

The ensemble results suggested that different methods embedding the contained spatial information might produce different results. The embedding obtained by STMVGAE could not only perform batch integration tasks on datasets with different expression patterns but also achieve the best performance compared to other methods. In conclusion, STMVGAE could effectively realize batch integration of spatial transcriptomics data.

9. Ablation studies

We systematically evaluated STMVGAE using the DLPFC dataset. First, we selected five different graph convolutional layers (GCNConv, GATConv, SuperGATConv, SGConv, FeaStConv) to calculate the ARI values for each of the 12 slices on the DLPFC dataset ([Figure S5A](#)). GCN achieved good performance. Simultaneously, we tested the performance of different graph convolutional layers on the 151672 slices and obtained different hierarchical distributions under different network architectures. We found that STMVGAE achieved very good results on the 151672 tiles using different convolutional layers, and the ARI values were high. The self-supervised module indicated the goal regarding cluster optimization during training, which we initiated with or without the DEC self-supervised module. Secondly, the use of multi-modal data is a major feature of STMVGAE, and we tested the results with or without the integration of morphological image features. Finally, we tested whether the results obtained by integrating different adjacency matrices using consensus clustering (Table 2 and [Figure S5B](#)).

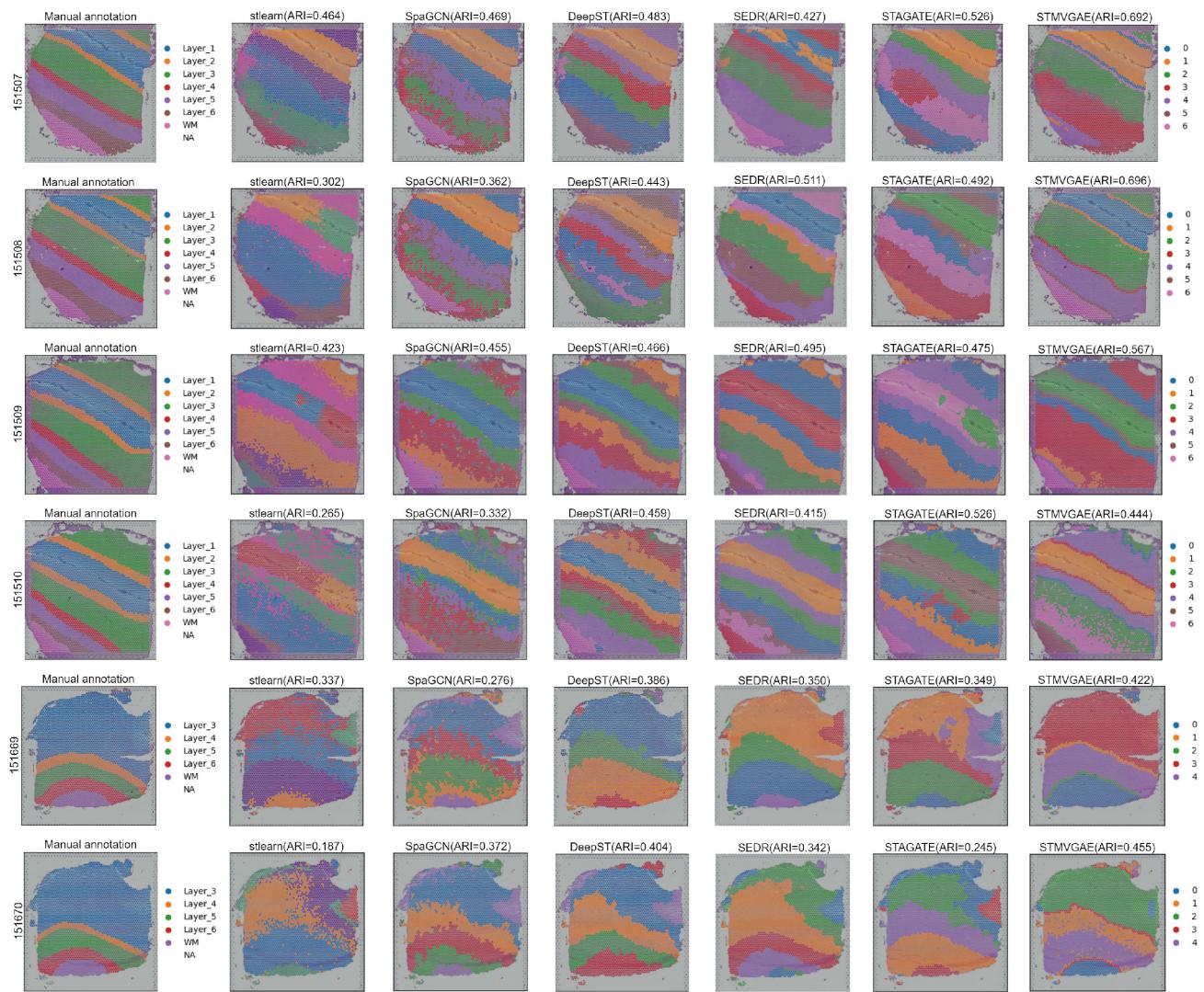
Additionally, it is a major feature of STMVGAE to use different adjacency matrix construction methods to learn different data patterns. Therefore, we not only used spatial coordinate information but also gene expression to construct adjacency matrices. We also evaluated the impact of multiple adjacency matrix construction methods on STMVGAE performance ([Figure S5C](#)). It was observed that the performance of constructing adjacency matrices using spatial coordinate location information significantly improved compared to that using gene expression, and the difference between the four adjacency matrices constructed using spatial coordinate location information was not significant. In order to find suitable model parameters, we conducted a grid search on the basic STMVGAE (without using multi-modal data, DEC self-supervised module, or consensus clustering). Our model used 3000 highly variable genes as input. Considering the retention of more information, the hyperparameters of the first linear layer were set to (1500,1000), the second linear layer to (700,600,500,400,300), and the third linear layer to (50,40,30,20). The hyperparameters of the graph convolutional layer were set to 64 and 8 ([Figure S5D](#)).

10. The results of adjacency matrix integration with different similarities

To comprehensively evaluate the adjacency matrices constructed by different similarity measures as input to train STMVGAE, and subsequently use the final results obtained by consensus clustering integration, we introduced four adjacency matrix constructions: Radius_balltree, Radius_kdtree, KNN_balltree, and KNN_kdtree, denoted as $A^{(1)}$, $A^{(2)}$, $A^{(3)}$, and $A^{(4)}$, respectively. The adjacency matrices constructed under four different similarity measures were used to train STMVGAE separately, and the results were plotted as box plots, showing their individual performance in [Figure S5C](#). It was observed that the adjacency matrices constructed with different similarity measures did not change significantly when used to train STMVGAE alone, indicating the robustness of STMVGAE.

We conducted experiments using any two of the four adjacency matrices as inputs to the model and calculated the ARI and NMI values on 12 slices of the DLPFC dataset. The results are shown in [Table S3](#) and [Table S4](#). Additionally, experiments were conducted to select any three of the four adjacency matrices as inputs, and all four adjacency matrices as inputs, followed by consensus clustering integration. The best results were achieved with $A^{(1)}$ and $A^{(3)}$, yielding an ARI of 0.5599. Moreover, using $A^{(1)}$, $A^{(3)}$, $A^{(4)}$ and $A^{(1)}$, $A^{(2)}$, $A^{(3)}$ and $A^{(4)}$ as inputs also yielded very good results, with ARIs of 0.5500 and 0.5420, respectively. Furthermore, regardless of how the four adjacency matrices were selected as inputs to the model, the worst results obtained were close to 0.5. This indicates that STMVGAE can effectively integrate the results obtained by adjacency matrices with different similarities, and the results are relatively stable.

Supplementary Figures



See next page

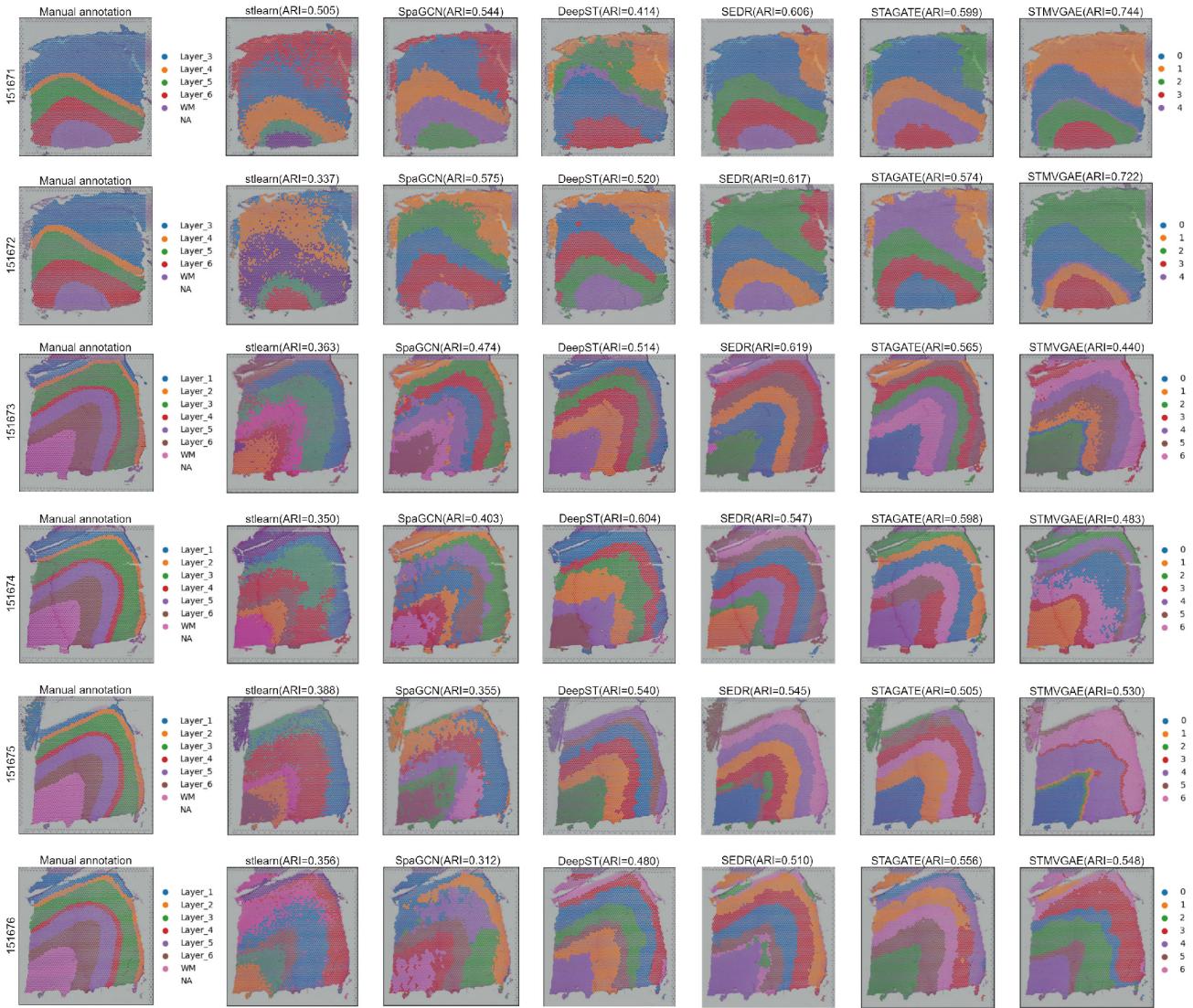
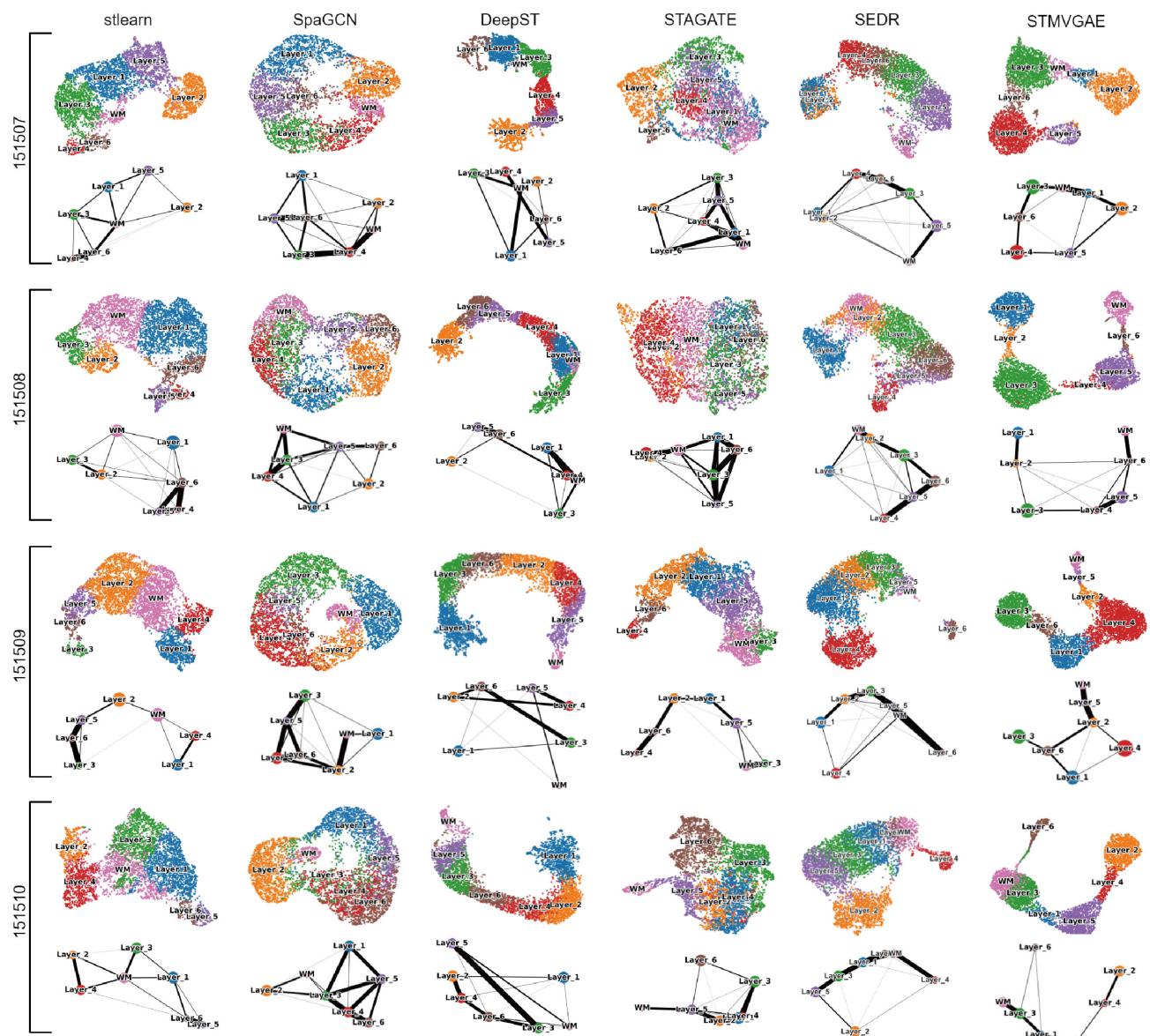
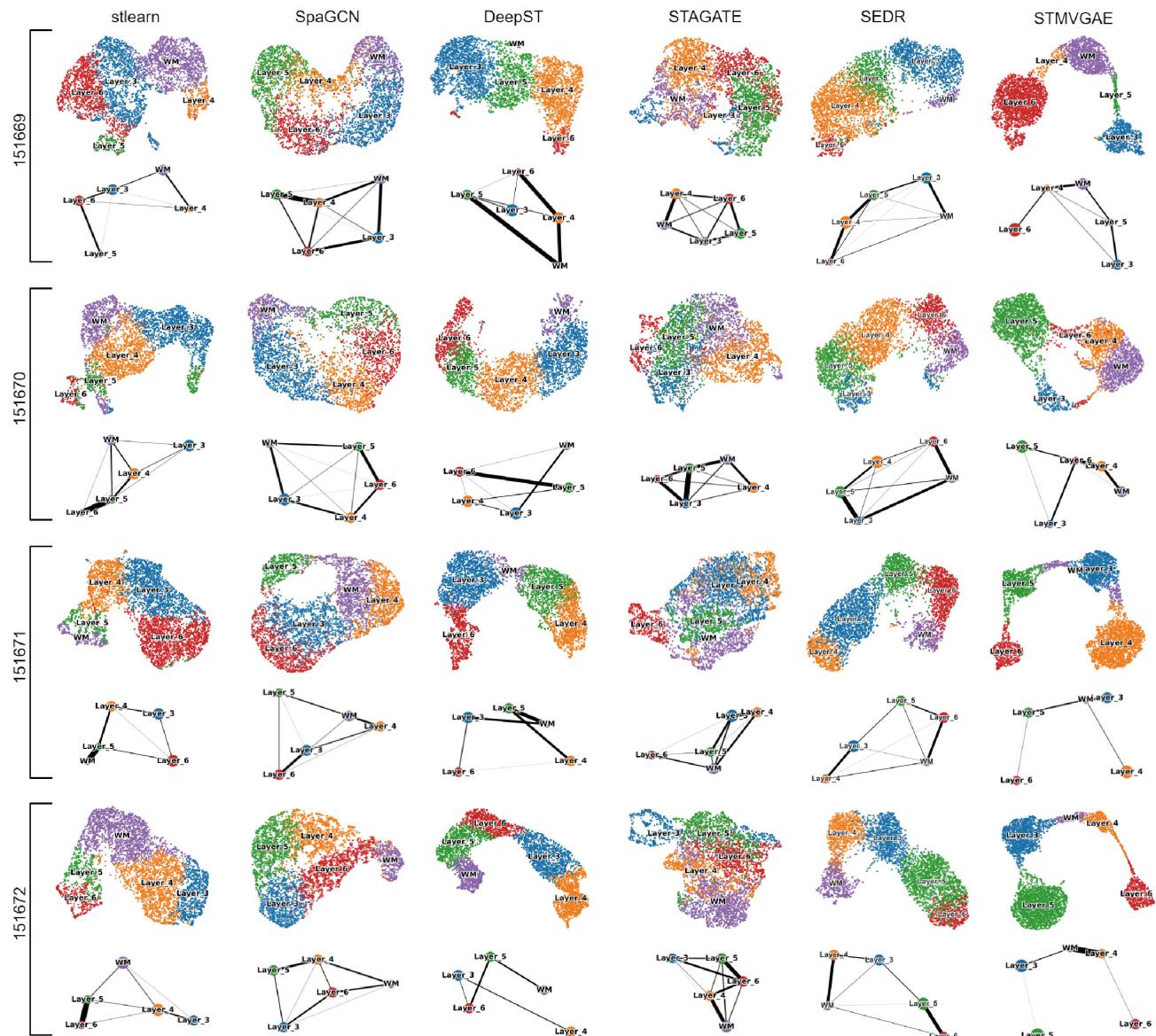


Figure S1. Comparison of spatial domains identification by clustering assignments via STMVGAE, STAGATE, SEDR, DeepST, stlearn, and manual annotation in all 12 slices of the DLPFC dataset.



See next page



See next page

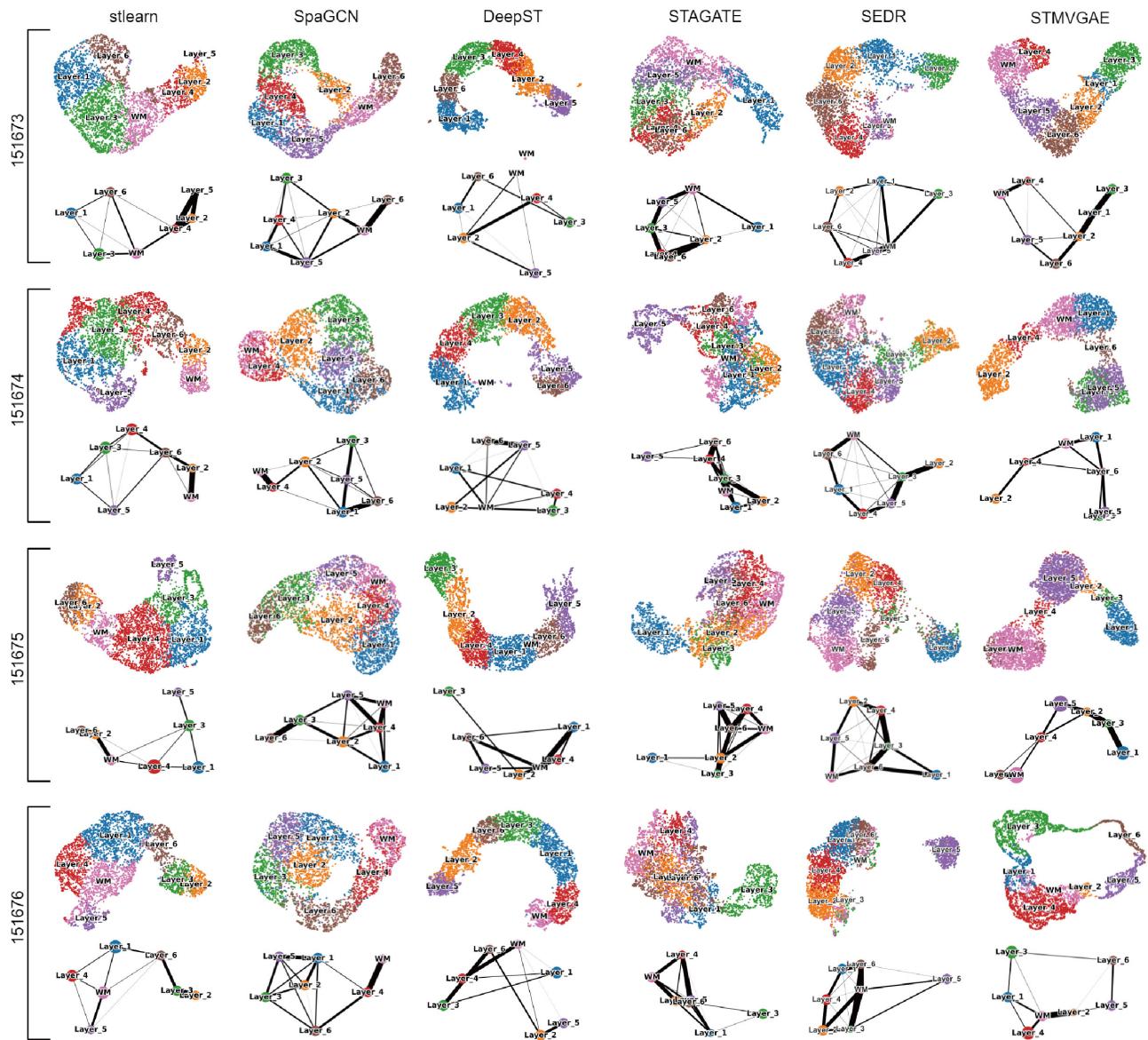


Figure S2. UMAP visualization and PAGA trajectory inference by STMVGAE, SEDR, STAGATE, DeepST, SpaGCN, and stlearn embeddings respectively.

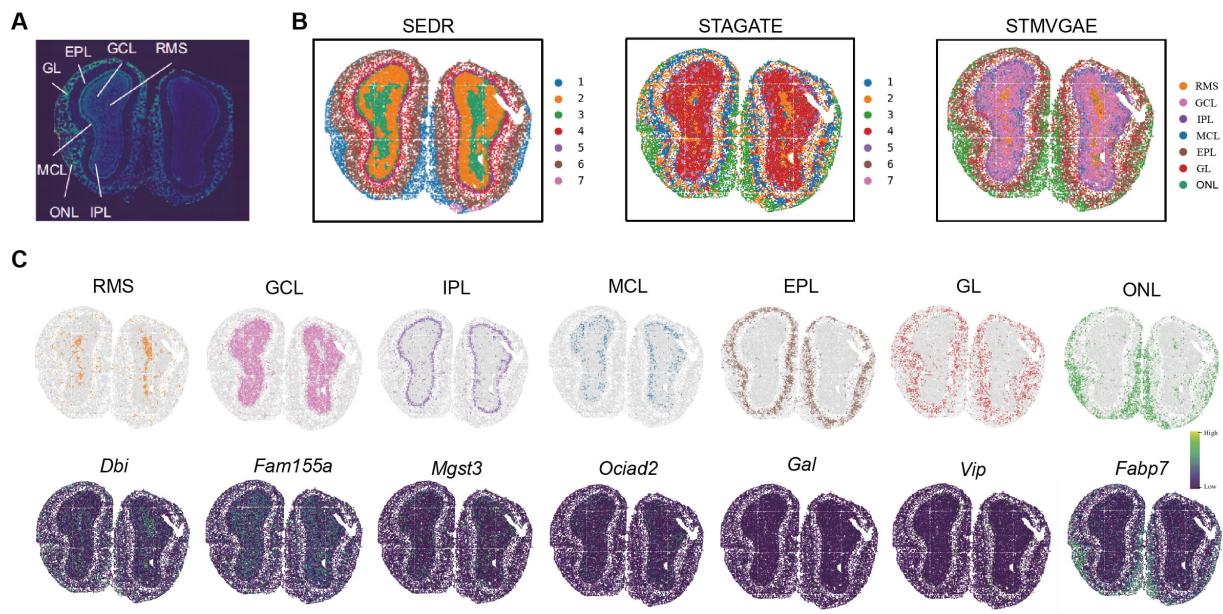


Figure S3. STMVGAE is able to accurately identify laminar structures in mouse olfactory bulb datasets. **(A)** Laminar organization of mouse olfactory bulb annotated in the DAPI-stained image generated by Stereo-seq. **(B)** Spatial domains generated by SEDR, STAGATE, and STMVGAE embeddings in the Stereo-seq mouse olfactory bulb tissue section. **(C)** Visualization of spatial domains identified by STMVGAE and the corresponding marker genes.

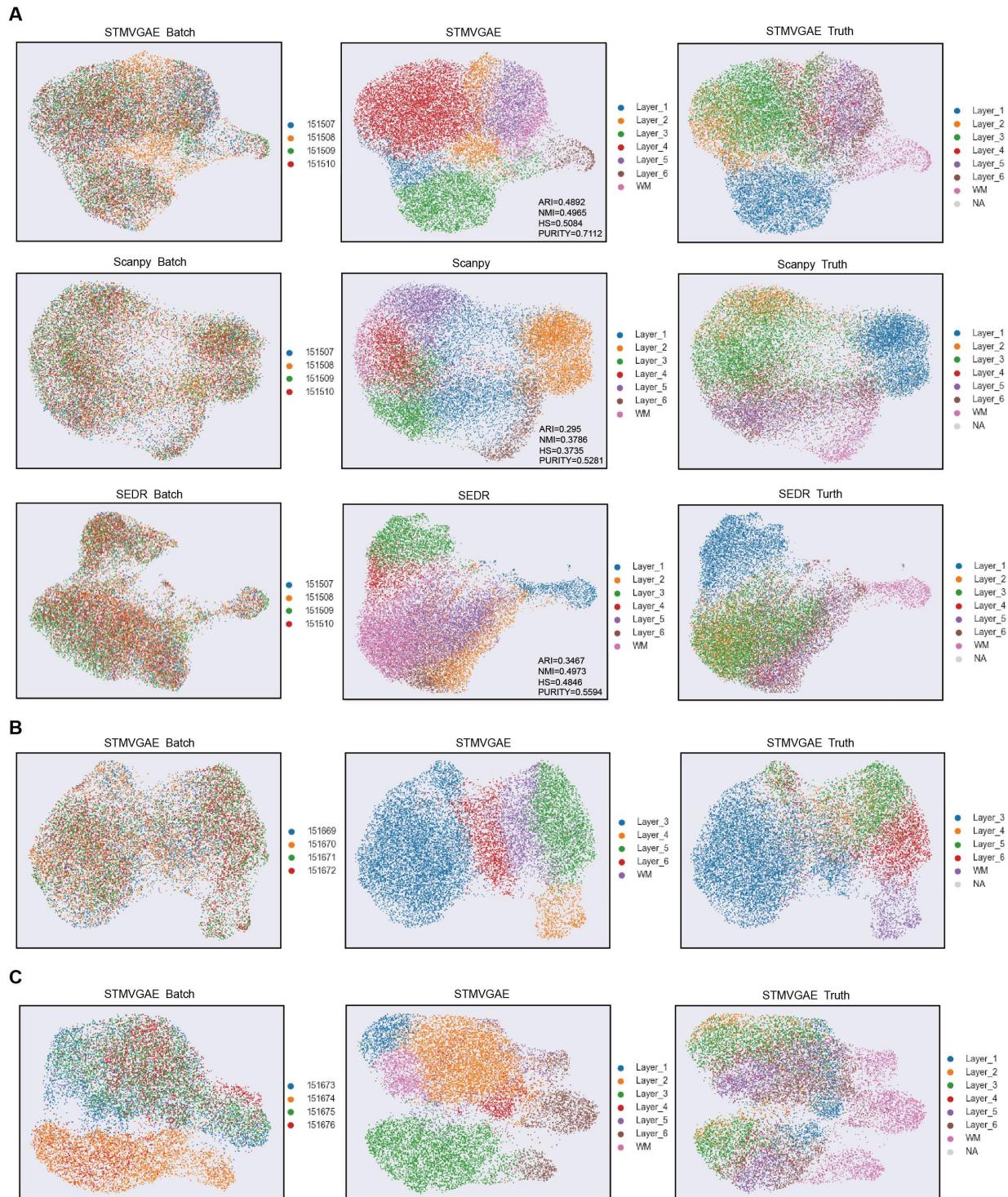


Figure S4. (A) UMAP visualization of batch integration algorithms on 151507, 151508, 151509, 151510 slices in DLPFC datasets. Each row represents the use of STMVGAE, Scipy, and SEDR methods for batch integration, and each column represents batches, identification spatial domains, and ground truth labels, respectively. (B) STMVGAE performs batch integration on 151669, 151670, 151671, 151672 slices in the DLPFC dataset. (C) STMVGAE performs batch integration on 151673, 151674, 151675, 151676 slices in the DLPFC dataset.

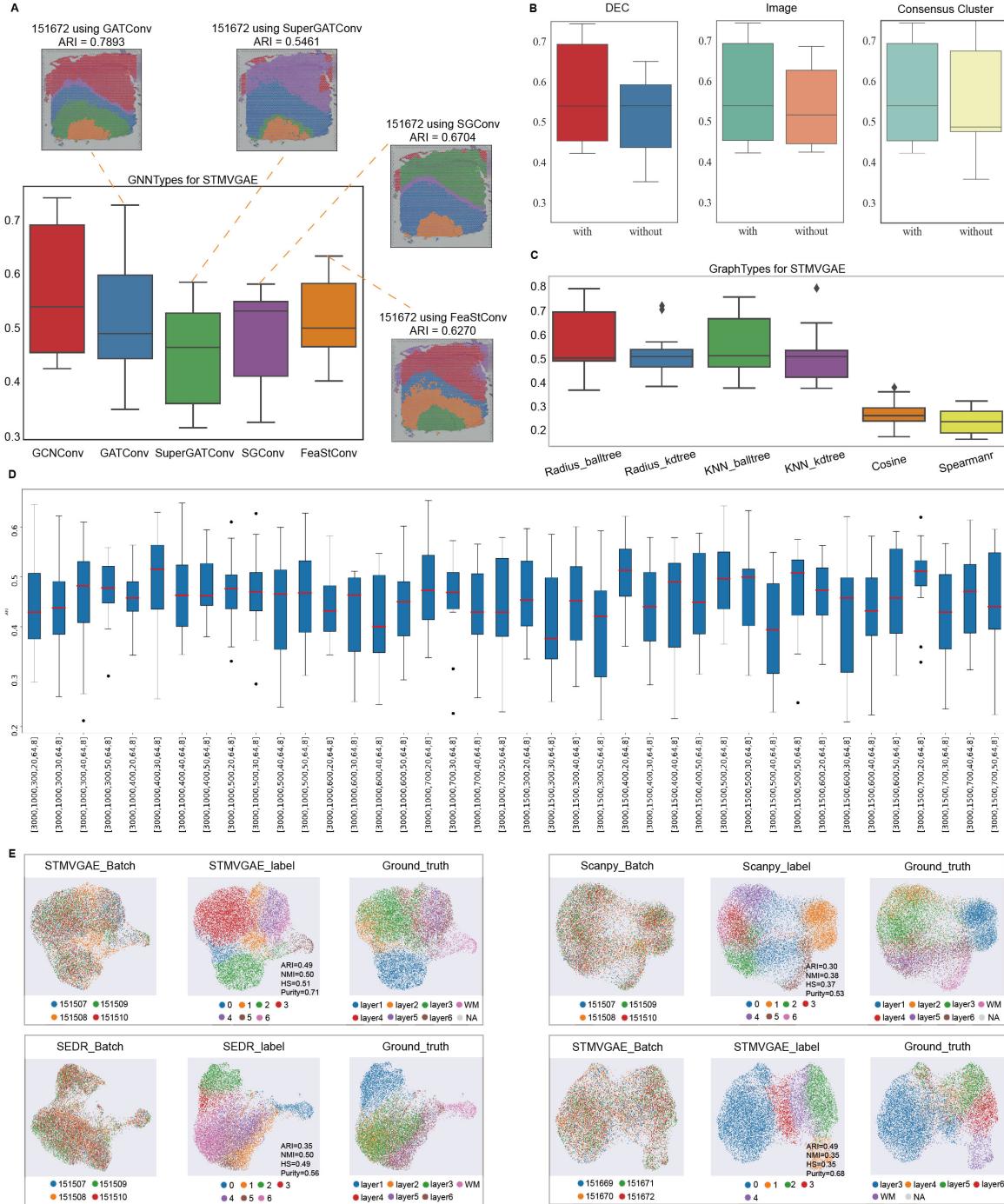


Figure S5. Ablation study and STMVGAE corrects for batch effects. **(A)** The ARI pirate graph of five GNN types, each of which was evaluated on 12 DLPFC slides, respectively. Spatial domain distributions of slides 151672 with various networks (GATConv, SuperGATConv, SGConv and FeaStConv) are displayed, respectively. **(B)** ARI box plots showing low-dimensional representation learning in STMVGAE with or without DEC self-supervised module, spatial data augmentation, and consensus clustering. **(C)** The ARI boxplot compares six methods for constructing adjacency matrices in STMVGAE. **(D)** The clustering accuracy of simple-STMVGAE with different hyperparameters in all 12 sections. The hyperparameters are selected by grid search of the first three linear layers and the last two graph convolutional layers of STMVGAE(basic). **(E)** UMAP plots of spatial integrated algorithms. They represent batches, recognition spatial domains, and ground truth labels, respectively.

Supplementary Tables

Table S1. Overview of comparative spatial domain identification methods.

Method	Methodology	Input Data	Downstream tasks	Link
SCANPY [1]	Non-spatial method	Gene expression data	Spatial domain identification Visualization Trajectory inference	https://scipy.readthedocs.io/
stlearn [2]	Deep neural network	Gene expression data Histology information	Spatial domain identification Visualization Trajectory inference	https://github.com/BiomedicalMachineLearning/stLearn
SEDR [3]	Variational graph autoencoders	Spatial location data Gene expression data	Spatial domain identification Visualization Trajectory inference Denosing Batch integration	https://github.com/HzFu/SEDR/
SpaGCN [4]	Graph convolutional networks	Spatial location data Gene expression data Histology information	Spatial domain identification Visualization Trajectory inference SVGs identification	https://github.com/jianhuupenn/SpaGCN/
DeepST [5]	Variational graph autoencoders	Spatial location data Gene expression data Histology information	Spatial domain identification Visualization Trajectory inference Batch integration	https://github.com/JiangBioLab/DeepST/
STAGATE [6]	Graph attention autoencoders	Spatial location data Gene expression data	Spatial domain identification Visualization Trajectory inference Denosing	https://github.com/zhanglabtools/STAGATE/

Table S2. Summary of the datasets in this study.

Platform	Tissue	Section	Number of domains	Spots	Genes
10X Visium	Human dorsolateral prefrontal cortex (DLPFC)[18]	151507	7	4226	
		151508	7	4384	
		151509	7	4789	
		151510	7	4634	
		151669	5	3661	
		151670	5	3498	33538
		151671	5	4110	
		151672	5	4015	
		151673	7	3639	
		151674	7	3673	
		151675	7	3592	
		151676	7	3460	
	Human breast cancer: ductal carcinoma in situ[13]	\	2	2518	17943
	Human breast[3] cancer	\	20	3798	36601
Spatialresearch	Melanoma cancer[12]	\	4	293	16148
Stereo-seq	Mouse olfactory bulb[14]	\	\	19109	27106

Table S3. STMVGAE performs graph combination test results on 12 slices of the DLPFC dataset. STMVGAE integrates the results of four different graphs in a free combination manner to calculate ARI, NMI, HS, and Pur (Purity) respectively. $A^{(1)}$, $A^{(2)}$, $A^{(3)}$, and $A^{(4)}$ represent Radius_balltree, Radius_kdtree, KNN_balltree, and KNN_kdtree respectively. The best result is underlined.

Slice	$A^{(1)} + A^{(2)}$				$A^{(1)} + A^{(3)}$				$A^{(1)} + A^{(4)}$				$A^{(2)} + A^{(3)}$				$A^{(2)} + A^{(4)}$				$A^{(3)} + A^{(4)}$			
	ARI	NMI	HS	Pur																				
151507	0.549	0.662	0.664	0.685	0.692	0.712	0.763	0.860	0.548	0.644	0.658	0.737	0.561	0.677	0.675	0.698	0.501	0.648	0.673	0.754	0.567	0.698	0.710	0.750
151508	0.594	0.657	0.681	0.813	0.696	0.703	0.724	0.821	0.666	0.689	0.739	0.841	0.582	0.620	0.640	0.801	0.503	0.604	0.606	0.691	0.573	0.664	0.654	0.686
151509	0.421	0.585	0.573	0.672	0.567	0.644	0.636	0.783	0.421	0.588	0.581	0.704	0.504	0.637	0.609	0.699	0.411	0.567	0.560	0.673	0.604	0.653	0.643	0.773
151510	0.557	0.651	0.610	0.737	0.444	0.562	0.532	0.660	0.403	0.559	0.530	0.653	0.548	0.651	0.607	0.719	0.496	0.648	0.613	0.734	0.410	0.560	0.544	0.648
151669	0.400	0.523	0.513	0.775	0.422	0.570	0.530	0.739	0.415	0.562	0.527	0.776	0.201	0.405	0.386	0.670	0.375	0.492	0.499	0.769	0.342	0.512	0.467	0.701
151670	0.386	0.509	0.468	0.722	0.455	0.559	0.527	0.758	0.337	0.475	0.433	0.693	0.324	0.486	0.433	0.650	0.314	0.455	0.412	0.688	0.246	0.414	0.376	0.601
151671	0.770	0.724	0.711	0.866	0.744	0.707	0.720	0.895	0.784	0.751	0.741	0.894	0.706	0.688	0.665	0.833	0.746	0.703	0.773	0.921	0.698	0.708	0.688	0.833
151672	0.670	0.697	0.814	0.925	0.722	0.724	0.740	0.851	0.640	0.658	0.690	0.811	0.700	0.713	0.762	0.902	0.686	0.701	0.710	0.805	0.617	0.654	0.692	0.831
151673	0.446	0.624	0.670	0.731	0.440	0.618	0.639	0.708	0.464	0.638	0.659	0.739	0.430	0.618	0.647	0.744	0.496	0.645	0.699	0.804	0.499	0.647	0.656	0.749
151674	0.454	0.584	0.591	0.689	0.483	0.608	0.655	0.801	0.427	0.607	0.610	0.667	0.458	0.544	0.552	0.725	0.420	0.554	0.584	0.743	0.466	0.588	0.599	0.711
151675	0.504	0.628	0.656	0.764	0.530	0.601	0.641	0.817	0.538	0.621	0.674	0.839	0.479	0.629	0.697	0.815	0.486	0.601	0.641	0.764	0.528	0.652	0.665	0.767
151676	0.444	0.601	0.641	0.750	0.548	0.643	0.667	0.773	0.513	0.602	0.632	0.758	0.469	0.634	0.664	0.762	0.488	0.611	0.652	0.780	0.477	0.632	0.642	0.715
Average	0.516	0.620	0.633	0.761	0.562	0.638	0.648	0.789	0.513	0.616	0.623	0.759	0.497	0.608	0.611	0.751	0.494	0.602	0.618	0.761	0.502	0.615	0.611	0.730

Table S4. STMVGAE performs graph combination test results on 12 slices of the DLPFC dataset. STMVGAE integrates the results of four different graphs in a free combination manner to calculate ARI, NMI, HS, and Pur (Purity) respectively. $A^{(1)}$, $A^{(2)}$, $A^{(3)}$, and $A^{(4)}$ represent Radius_balltree, Radius_kdtree, KNN_balltree, and KNN_kdtree respectively. The best result is underlined.

Slice	$A^{(1)} + A^{(2)} + A^{(3)}$				$A^{(1)} + A^{(2)} + A^{(4)}$				$A^{(1)} + A^{(3)} + A^{(4)}$				$A^{(2)} + A^{(3)} + A^{(4)}$				$A^{(1)} + A^{(2)} + A^{(3)} + A^{(4)}$							
	ARI	NMI	HS	Pur	ARI	NMI	HS	Pur	ARI	NMI	HS	Pur												
151507	0.618	0.708	0.827	0.913	0.581	0.699	0.710	0.761	0.688	0.729	0.773	0.866	0.569	0.700	0.715	0.771	0.583	0.704	0.719	0.776				
151508	0.660	0.700	0.721	0.824	0.625	0.674	0.696	0.814	0.705	0.723	0.753	0.839	0.597	0.681	0.724	0.847	0.676	0.708	0.730	0.821				
151509	0.570	0.657	0.641	0.776	0.428	0.612	0.601	0.693	0.570	0.646	0.630	0.747	0.496	0.643	0.618	0.696	0.574	0.656	0.644	0.777				
151510	0.501	0.652	0.635	0.717	0.438	0.634	0.603	0.680	0.411	0.588	0.563	0.678	0.421	0.622	0.593	0.681	0.433	0.635	0.614	0.686				
151669	0.353	0.543	0.506	0.746	0.417	0.551	0.515	0.773	0.374	0.558	0.534	0.739	0.384	0.498	0.495	0.767	0.298	0.513	0.481	0.727				
151670	0.335	0.516	0.459	0.673	0.423	0.537	0.493	0.741	0.481	0.534	0.507	0.764	0.379	0.509	0.462	0.685	0.443	0.512	0.470	0.743				
151671	0.745	0.723	0.748	0.910	0.826	0.740	0.751	0.890	0.790	0.751	0.742	0.896	0.729	0.701	0.785	0.929	0.798	0.751	0.735	0.882				
151672	0.726	0.718	0.755	0.848	0.704	0.705	0.748	0.827	0.725	0.723	0.742	0.855	0.709	0.712	0.728	0.815	0.718	0.717	0.736	0.850				
151673	0.500	0.654	0.659	0.694	0.543	0.675	0.694	0.788	0.532	0.672	0.687	0.741	0.498	0.645	0.648	0.738	0.512	0.665	0.665	0.702				
151676	0.490	0.592	0.625	0.749	0.422	0.587	0.590	0.662	0.477	0.628	0.620	0.672	0.460	0.592	0.625	0.742	0.463	0.607	0.613	0.693				
151675	0.528	0.662	0.691	0.779	0.507	0.624	0.652	0.768	0.462	0.616	0.655	0.760	0.473	0.621	0.680	0.801	0.531	0.661	0.693	0.788				
151677	0.442	0.616	0.635	0.708	0.471	0.643	0.679	0.728	0.445	0.616	0.644	0.721	0.479	0.647	0.687	0.730	0.476	0.642	0.654	0.691				
Average	0.539	0.645	0.658	0.778	0.532	0.640	0.644	0.760	0.555	0.649	0.654	0.773	0.516	0.631	0.647	0.767	0.542	0.648	0.646	0.761				

References

1. F Alexander Wolf, Philipp Angerer, et al. SCANPY: large-scale single-cell gene expression data analysis. *Genome Biology*, 19:1–5, 2018.
2. Duy Pham, Xiao Tan, et al. stlearn: Robust mapping of spatiotemporal trajectories and cell-cell interactions in healthy and diseased tissues. *Nature Communications*, 14(1):7739, 2023.
3. Hang Xu, Huazhu Fu, et al. SEDR: Unsupervised spatially embedded deep representation of spatial transcriptomics. *Genome Medicine*, 16(1):1–15, 2024.
4. Jian Hu, Xiangjie Li, et al. SpaGCN: Integrating gene expression, spatial location and histology to identify spatial domains and spatially variable genes by graph convolutional network. *Nature Methods*, 18(11):1342–1351, 2021.
5. Chang Xu, Xiyun Jin, et al. DeepST: identifying spatial domains in spatial transcriptomics by deep learning. *Nucleic Acids Research*, 50(22):e131–e131, 2022.
6. Kangning Dong and Shihua Zhang. STAGATE: Deciphering spatial domains from spatially resolved transcriptomics with an adaptive graph attention auto-encoder. *Nature Communications*, 13(1):1739, 2022.
7. Chihao Zhang, Kangning Dong, et al. STAMarker: determining spatial domain-specific variable genes with saliency maps in deep learning. *Nucleic Acids Research*, 51(20):e103–e103, 2023.
8. Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2:193–218, 1985.
9. Zhiyuan Yuan, Fangyuan Zhao, Senlin Lin, Yu Zhao, Jianhua Yao, Yan Cui, Xiao-Yong Zhang, and Yi Zhao. Benchmarking spatial clustering methods with spatially resolved transcriptomics data. *Nature methods*, 21(4):712–722, 2024.
10. Edward C Gilmore and Karl Herrup. Cortical development: layers of complexity. *Current Biology*, 7(4):R231–R234, 1997.
11. Rachel J Waldemer-Streyer, Adriana Reyes-Ordoñez, et al. Cxcl14 depletion accelerates skeletal myogenesis by promoting cell cycle withdrawal. *NPJ Regenerative Medicine*, 2(1):1–10, 2017.
12. Kim Thrane, Hanna Eriksson, et al. Spatially resolved transcriptomics enables dissection of genetic heterogeneity in stage iii cutaneous malignant melanoma. *Cancer Research*, 78(20):5970–5979, 2018.
13. Mohammad Nuwaisir Rahman, Abdullah Al Noman, et al. ScribbleDom: using scribble-annotated histology images to identify domains in spatial transcriptomics data. *Bioinformatics*, 39(10):btad594, 2023.
14. Ao Chen, Sha Liao, et al. Spatiotemporal transcriptomic atlas of mouse organogenesis using dna nanoball-patterned arrays. *Cell*, 185(10):1777–1792, 2022.
15. Jörn Lötsch, Thomas Hummel, et al. Disorganized expression of odorant receptors in the olfactory epithelium of morbidly obese subjects. *Chemical Senses*, 39(8):707–716, 2014.
16. Ye Zhang, Kenian Chen, et al. An rna-sequencing transcriptome and splicing database of glia, neurons, and vascular cells of the cerebral cortex. *Journal of Neuroscience*, 34(36):11929–11947, 2014.
17. Ilya Korsunsky, Nghia Millard, et al. Fast, sensitive and accurate integration of single-cell data with harmony. *Nature Methods*, 16(12):1289–1296, 2019.
18. Kristen R Maynard, Leonardo Collado-Torres, et al. Transcriptome-scale spatial gene expression in the human dorsolateral prefrontal cortex. *Nature Neuroscience*, 24(3):425–436, 2021.