

## **Supplementary Materials for**

### **STAIG: Spatial Transcriptomics Analysis via Image-Aided Graph Contrastive Learning for Domain Exploration and Alignment-Free Integration**

**Yitao Yang<sup>1</sup>, Yang Cui<sup>1</sup>, Xin Zeng<sup>1</sup>, Yubo Zhang<sup>1</sup>, Martin Loza<sup>2</sup>, Sung-Joon Park<sup>2</sup>, Kenta Nakai<sup>1,2\*</sup>**

<sup>1</sup>Department of Computational Biology and Medical Science, the University of Tokyo, Japan

<sup>2</sup>Human Genome Center, the Institute of Medical Science, the University of Tokyo, Japan

\*Kenta Nakai (knakai@ims.u-tokyo.ac.jp)

## Supplementary Notes

### Baseline methods and evaluation metrics

To evaluate the clustering efficacy of STAIG comprehensively, we adopted eight prevailing methods as our baseline for comparison. Specifically, this includes the non-spatial Seurat<sup>1</sup> method and seven prominent spatial clustering methods. Among these seven spatial methods, GraphST<sup>2</sup>, DeepST<sup>3</sup>, and STAGATE<sup>4</sup> utilize both genetic and spatial information. SpaGCN<sup>5</sup>, SEDR<sup>6</sup>, ConST<sup>7</sup>, MuCoST<sup>8</sup>, and stLearn<sup>9</sup> incorporated pathological image data. Furthermore, for the integration task, we additionally compared our method with STAligner<sup>10</sup>, which was developed specifically for batch integration. In the zebrafish melanoma dataset, we also compared tumor microenvironment results with TESLA<sup>11</sup>. For all methods, we conducted tests using the default parameter settings. To ensure fairness in the comparison results, we uniformly adopted the same refinement method for the DLPFC dataset. When testing comparative algorithms on experimental platform data that were not evaluated in the original papers, we converted the data into H5AD format. In addition, we modified the code for the model input to facilitate testing. The details of these methods are as follows:

- **Seurat:** Seurat is a widely utilized library for single-cell transcriptomic analysis capable of processing spatial transcriptomic data. We employed the default pipeline and parameters in Seurat, as outlined in the official guidelines. Because of the inability to specify the number of clusters directly, we adjusted the resolutions from higher to lower values and ran the *FindClusters()* function until the outcome yielded the desired number of clusters.
- **GraphST:** GraphST combines graph neural networks with self-supervised contrastive learning to obtain informative and distinctive spot representations. We conducted the experiments using the following default parameters: *epochs* = 600, *k* = 5.
- **DeepST:** DeepST leverages a pre-trained deep neural network to extract image features, which are then integrated with gene expression and spatial location data to form an augmented gene expression matrix. We conducted our experiments using the default parameters: *pca\_n\_comps* = 200, *pre\_epochs* = 800, *epochs* = 1000, *Conv\_type* = "GCNConv", *neighbour\_k* = 4, *distType* = "BallTree".
- **STAGATE:** STAGATE leverages an attention mechanism within a graph autoencoder to adaptively learn low-dimensional latent embeddings that capture both spatial information and gene expression. We conducted our experiments using the following default parameters: *rad\_cutoff* = 150, *pre\_resolution* = 0.2, *alpha* = 0.
- **SpaGCN:** SpaGCN is an end-to-end model that integrates gene expression, spatial locations, and histological pixel intensities from spots into an undirected weighted graph for subsequent analyses using a graph convolutional network. We conducted our experiments using default parameters. The output prior to the clustering layer is extracted as the obtained low-dimensional representation.
- **SEDR:** SEDR utilizes a deep autoencoder to map gene data and concurrently integrates spatial information via a variational graph autoencoder. We conducted experiments using the following default parameters: *k* = 10, *epochs* = 200, *cell\_feat\_dim* = 200, *p\_drop* = 0.2, *deep\_Dim* = 20.
- **ConST:** ConST applies a pre-trained visual model, MAE<sup>12</sup>, to extract histological features, complemented by a three-level contrastive learning strategy to yield low-dimensional mappings of spots. We conducted our experiments using the following default parameters: *k* = 10, *epochs* = 200, *cell\_feat\_dim* = 300, *p\_drop* = 0.2.
- **MuCoST:** MuCoST is a graph contrastive learning framework combining shared multi-view graph convolutional network autoencoder and InfoNCE<sup>13</sup> contrastive loss for deciphering complex ST data from dual scale structural dependency. We conducted our experiments using the following default parameters: *latent\_dim* = 50, *epoch* = 1000, *mode\_his* = 'noh'.

- **stLearn:** stLearn proposes a cluster analysis framework that employs a gene expression normalization algorithm based on the histological features extracted from images and spot locations. We conducted our experiments using default parameters. The parameter `n_comps = 15` for function `st.em.run_pca()`, and the parameter `se_data = 'raw'`, `weights = 'physical_distance'` for function `st.spatial.SME.SME_normalize()`.
- **STAligner:** STAligner is a graph attention neural network designed for integrating and aligning spatial transcriptomics datasets within and across platforms. We used the default parameters for the experiments: `hidden_dims = [512, 30]`, `n_epochs = 1000`, `knn_neigh = 100`.
- **TESLA:** TESLA is a machine learning framework for tissue annotation in spatial transcriptomics (ST) at pixel-level resolution. It utilizes graph convolutional neural networks to integrate gene expression and histology image information, enhancing gene expression to super resolution. It can be used for analyzing the tumor microenvironment and segmenting the tumor center and periphery. We followed the default steps for the analysis, using `cv2.detect_contour()` for contour detection, with the parameter `CANNY_THRESH_2 = 300` to achieve better segmentation results. The resolution parameter was set to `res = 50`.

For datasets with manual annotations, the Adjusted Rand Index (ARI)<sup>14</sup> and Normalized Mutual Information (NMI)<sup>15</sup> served as the evaluation metric.

- ARI assesses the similarity between true and predicted labels while adjusting for chance. The values range from -1 to 1. A value close to 1 indicates that the two data clusters are nearly identical, while a value close to -1 indicates that the two data clusters are opposite. A score close to zero indicates that the clusters are likely to be random.
- NMI measures the mutual information between the true and predicted labels, normalized by the entropy of each labeling. It ranges from 0 to 1, where a higher value indicates a greater agreement between the two labelings. An NMI of 1 means that the predicted labels perfectly match the true labels, while an NMI of 0 suggests that the predicted labels are independent of the true labels.

For datasets without ground-truth labels, Silhouette Coefficient score (SC)<sup>16</sup> and Davies-Bouldin Index (DB)<sup>17</sup> were chosen as the evaluation metric.

- SC measures the similarity of an object to its own cluster compared with other clusters. The coefficient can vary between -1 and 1, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters. If most objects have high values, the clustering configuration is considered appropriate. If many points have low or negative values, the clustering configuration may have too many or too few clusters.
- DB is a metric for evaluating clustering algorithms, where a lower value indicates better separation between clusters and greater compactness within clusters. It is calculated as the average similarity between each cluster and its most similar cluster, where similarity is defined as the ratio of within-cluster distances to between-cluster distances. A good clustering algorithm should have a low DB value, indicating that the clusters are well-separated and compact.

For the integration task, we additionally introduced KL divergence of batch mixing (BatchKL)<sup>18</sup> and integrated Local Inverse Simpson's Index (ILISI)<sup>19</sup> to evaluate the performance batch effect removal based on the embedding representation.

- Batch KL is a metric that assesses the batch integration quality. A smaller KL divergence indicates that the local neighborhood of each cell contains a similar proportion of cells from different batches as the global distribution, suggesting better mixing of the batches in the embedding space. The lower value of BatchKL denotes better mixing performance.

- ILISI is a metric that assesses the batch integration quality. It quantifies the diversity of batch labels in the local neighborhood of each cell in the embedding space. A higher ILISI value indicates that the local neighborhoods of cells are more diverse in terms of batch labels, suggesting better integration of different batches.

### **Justification for the selection of log2fc threshold**

We provide a detailed justification for using a log2 fold change (log2FC) threshold of 0.25 in this study. The Seurat package, employed during our analysis, uses 0.25 as its default parameter in the FindMarkers() function, a setting that is widely adopted in the field. More importantly, spatial transcriptomics, unlike bulk RNA-seq where a log2FC threshold of 1 is common, is characterized by lower sequencing depth and higher levels of noise and data sparsity. These unique aspects of spatial transcriptomics data make a lower threshold of 0.25 more suitable, as it accommodates the inherent variability and ensures the detection of biologically significant differentially expressed genes (DEGs) that might be overlooked with a higher threshold.

### **Hyperparameter optimization on 10x Visium**

#### **● Main hyperparameters**

We systematically evaluated the hyperparameters of the STAIG when image is available on the DLPFC dataset. Our primary focus was on the temperature parameter  $\tau$ , number of epochs  $E$ , number of GCN layers  $l$ , and the number of neighboring nodes  $k$  for each spot. Tested values for  $\tau$  ranged from 0.1 to 50 in specified increments, epochs  $E$  spanned from 50 to 1000,  $l$  was either 1 or 2, and  $k$  took values of 3, 5, or 7. Given a set of parameters, the average ARI and NMI across the 12 slices served as the performance metric.

Our analysis began with 3D surface plots to visualize how variables  $l$  and  $k$  affected performance (Supplementary Fig. S22a-e). We found that, for  $l = 1$ , the average ARI and NMI was consistently higher than that for  $l = 2$  across different  $k$  values (Supplementary Fig. S22a-c). In addition, for a fixed  $l$ ,  $k = 5$  outperformed  $k = 3$  and  $k = 7$  (Supplementary Fig. S22d-e).

After determining the optimal settings ( $l = 1$  and  $k = 5$ ), we used heatmaps to display the ARI and NMI variation with different  $\tau$  and  $E$  combinations (Supplementary Fig. S22f). The best result, an average ARI of 0.63 and an average NMI of 0.68, was achieved with  $\tau = 10$  and  $E = 400$ . To further examine the effects of  $\tau$  and  $E$  we analyzed boxplots from 12 samples. With  $\tau = 10$ , the ARI changed slightly over epochs, peaking at  $E = 400$ , then declining with more epochs, likely due to under-convergence or over-uniform sample embeddings (Supplementary Fig. S22g). For  $E = 400$ , smaller  $\tau$  values led to poorer results, because intense contrasts in early iterations impeded gradient descent, while larger  $\tau$  values provided milder contrasts and more stable results (Supplementary Fig. S22h). In contrast, NMI exhibits larger fluctuations, but overall, it has a positive correlation with ARI and follows the same trend.

Conclusively, based on our comprehensive analysis, the parameters  $k = 5$ ,  $l = 1$ ,  $\tau = 10$ , and  $E = 400$  were designated as the default settings for STAIG when image is available.

#### **● Masked ratio of features in graph augmentation**

During the graph augmentation process, in addition to the changes in graph structure, we randomly mask a certain proportion of gene features for each node in every iteration to enhance the robustness of the training process. To evaluate the impact of different masked ratios on the results, we tested the average ARI and NMI values based on 12 slices of the DLPFC dataset (Supplementary Fig. S23a). We conducted tests with masked ratios ranging from 0 to 50%, using a step size of 10%.



The results revealed that the average ARI and NMI values were highest when the mask ratio was 10%. When the mask ratio was increased to 20%, the ARI slightly decreased, while the NMI remained almost unchanged. However, when the mask ratio exceeded 20%, both ARI and NMI exhibited a significant decline. Clustering performance became highly unsatisfactory when the mask ratio surpassed 40%. This deterioration can be attributed to the excessive loss of information, leading to a reduction in the common information shared between the two augmented graphs, thus hindering the model's ability to learn useful information.

Interestingly, the performance of the model without any masking (0% mask ratio) was not as good as the model with a 10% masking ratio. This observation suggests that partial masking allows the model to receive slightly different information during each iteration, thereby enhancing the model's generalization ability.

Consequently, in the default training process, we set the masked ratio to 10% to strike a balance between preserving sufficient information and introducing beneficial variations in the input data.

### ● **Cluster number in debiased strategy**

In the debiased strategy (DS), we initially assign pseudo labels to spots using the K-means algorithm. During the contrastive learning process, we remove samples with the same class labels from the negative samples. To evaluate the impact of different initial clustering numbers on the results, we assessed the average ARI and NMI values across 12 slices of the DLPFC dataset (Supplementary Fig. S23b) and the ARI and NMI values for a human breast cancer slice (Supplementary Fig. S23c). We tested initial clustering numbers ranging from 10 to 100, with a step size of 10.

The results showed that when the number of clusters was less than 40, it led to a decline in performance. Although there are small number of inherent class labels, the heterogeneity of image textures is more diverse. Having too few clusters results in an excessive number of samples within each cluster, and these samples may not belong to the same class. Such pseudo labels can lead to the exclusion of true positive and negative sample comparisons. When the number of clusters exceeded 60, the performance slightly decreased and then stabilized. This is because finer-grained clustering reduces the number of samples in each class, and the comparison for some nodes shifts towards comparing with the entire set. Although the effectiveness of DS diminishes in this case, other aspects such as graph structure perturbation based on image features continue to play a role. Despite some performance loss, the model still maintains good results.

To further ensure that the K-means clustering results did not produce extreme distributions, where one class has a very large number of samples while the remaining classes have very few, we analyzed the number of samples in each class for the image embeddings of #151673 (Supplementary Fig. S23d) and the human breast cancer slice (Supplementary Fig. S23e) when the number of clusters was set to 40. We sorted the classes based on the number of samples in ascending order. The analysis revealed that most classes had a considerable number of samples, and extreme distributions did not occur. Although some classes had fewer samples, this does not affect the training process for two reasons. First, when a sample is compared with the global set, it effectively transforms into a traditional contrastive learning framework, which can still learn the differences among the entire set. Second, samples grouped into the same class are due to their similar image features, and they are likely to be adjacent and of the same class. Avoiding comparisons with nearby nodes of the same class in a small range helps the model learn local expression similarities and commonalities.

Based on the experimental results, we chose 40 as the default number of clusters for DS.

### ● **Methods for generating the image distance matrix**

To construct the image distance matrix, we tested three methods for calculating the similarity between nodes based on the features extracted from BYOL. We evaluated these methods using the average ARI and NMI values from the DLPFC dataset (Supplementary Fig. S23f). The methods we tested were Euclidean distance, cosine distance, and the RBF kernel. For the RBF kernel, we tested three different values for the gamma parameter: 0.01, 0.1, and 1.

The results showed that Euclidean distance and cosine distance performed similarly, indicating that the embedding differences can be distinguished to some extent based on either distance or direction. However, the performance of the RBF kernel was highly dependent on the choice of the gamma value. When gamma was set to 0.01, the performance was closest to that of Euclidean distance, but there was still a gap. When gamma was set to other values, the performance was worse.

Therefore, in the default construction of the image distance matrix, we chose to use Euclidean distance as the method for measuring the similarity between nodes based on their image features. This choice was based on its simplicity, effectiveness, and consistent performance compared to the other methods tested.

### **Hyperparameter optimization on other platform**

To explore hyperparameter optimization beyond the Visium platform, we tested datasets that typically lack class labels. We selected the Stereo-seq dataset for mouse brain olfactory, evaluating SC and DB scores; the STARmap dataset for ARI and NMI values; and the fluorescence-based MERFISH platform for mouse visual cortex, also assessing SC and DB scores with lower gene quantity.

#### **● Main hyperparameters**

Our primary focus was on tuning four main hyperparameters: the temperature parameter  $\tau$ , the number of training epochs  $E$ , the count of GCN layers  $l$ , and the number of neighboring nodes  $k$  for each spot. We tested  $\tau$  values from 0.1 to 50, epochs  $E$  from 50 to 1000, GCN layers  $l$  at 1 or 2, and  $k$  values at 3, 5, or 7.

For the Stereo-seq platform, we discovered that with a fixed  $k$ , a double-layer GCN ( $l = 2$ ) outperformed a single-layer in terms of SC, and also achieved lower DB values (Supplementary Fig. S24a-c). When holding the number of GCN layers constant,  $k = 5$  generally yielded higher SC and lower DB scores compared to  $k = 3$  and  $k = 7$  (Supplementary Fig. S24d-e).

With the optimal parameters identified ( $l = 2$  and  $k = 5$ ), we visualized the variation of SC and DB scores across different  $\tau$  and  $E$  settings using heatmaps (Supplementary Fig. S24f). Better performance was consistently observed with  $\tau$  greater than or equal to 10 and epochs over 300, a range that encompasses the default parameters of the Visium platform. For usability, we set the default parameters for Stereo-seq to  $k = 5$ ,  $l = 2$ ,  $\tau = 10$ , and  $E = 400$ .

Turning to the STARmap platform, we compared ARI and NMI metrics. Results indicated that a double-layer GCN ( $l = 2$ ) was superior to a single layer (Supplementary Fig. S25a-c). There was a positive correlation between the distributions of ARI and NMI, and  $k = 5$  was consistently more effective than  $k = 3$  or  $k = 7$  (Supplementary Fig. S25d-e). Examining heatmaps (Supplementary Fig. S25f) for the combination of  $l = 2$  and  $k = 5$ , we again observed optimal metrics within the same parameter range as Stereo-seq. Thus, for ease of use, STARmap's default parameters were set to  $k = 5$ ,  $l = 2$ ,  $\tau = 10$ , and  $E = 400$ .

For the MERFISH platform,  $k = 3$  and  $k = 5$  performed better with a single GCN layer ( $l = 1$ ), demonstrating higher SC and lower DB values (Supplementary Fig. S26a-b). Conversely, with  $k = 7$ , a double-layer GCN was more effective (Supplementary Fig. S26c). Overall,  $k = 5$  exhibited superior performance across most datasets (Supplementary Fig. S26d-e). Upon further analysis with heatmaps (Supplementary Fig. S26f) for  $k = 5$  and  $l = 1$ , we found stable results when  $\tau$  was greater than or equal to 10 and epochs exceeded 200. MERFISH data converged more rapidly than the other platforms, possibly due to the smaller number of genes involved. For broad usability, we set MERFISH's default parameters to  $k = 5$ ,  $l = 1$ ,  $\tau = 10$ , and  $E = 200$ , adjusting epochs to reflect the faster convergence.

In general, the difference between the optimal parameters of each platform is mainly  $l$ , and the other parameters are relatively stable. The selection range of  $l$  is 1 or 2. In the actual use process, we recommend choosing according to the actual situation.

### ● Masked ratio of features in graph augmentation

Based on the experience with the 10x Visium platform, we know that excessive mask ratio can lead to a decrease in performance. To investigate this, we tested mask ratios ranging from 0 to 50%, with a step size of 10%.

The results showed that when the mask ratio was set to 10%, all platforms exhibited good performance, with each metric reaching its highest standard. However, at a mask ratio of 20%, the performance of the STARmap (Supplementary Fig. S27a) and MERFISH platforms (Supplementary Fig. S27b) declined significantly. This can be attributed to the fewer number of genes inherently present in these platforms. Due to the limited number of genes, excessive masking led to a noticeable decrease in performance. For the Stereo-seq platform (Supplementary Fig. S27c), the performance started to decline when the mask ratio exceeded 20%. Interestingly, a 10% mask ratio consistently yielded better results compared to not applying any masking at all, which is consistent with the parameter testing on the Visium platform.

Consequently, in the default training process, we set the masked ratio to 10% to strike a balance between preserving sufficient information and introducing beneficial variations in the input data across all platforms.

### Comparing adaptive edge perturbation based on gene similarity with random perturbation

In the absence of imaging data, alongside random perturbation, we also tested an edge deletion strategy based on gene similarity to reduce parameter tuning. Specifically, we first reduced the dimensionality of the nodes' original gene data to 64 dimensions using PCA. As with image feature processing, we constructed a similarity matrix by calculating node-to-node similarities based on the reduced vectors and cosine similarity. This similarity was then translated into a probability of edge deletion, analogous to image feature distance, followed by adaptive neighbor-edge random deletion based on gene similarity.

We employed the average ARI and NMI from the DLPFC dataset to assess the effects of random versus gene-similarity-based adaptive edge deletion on spatial clustering when images are unavailable (Supplementary Fig. S27d). We set the random deletion probability,  $p$ , and investigated values ranging from 0.1 to 0.5. Our simulations revealed that  $p = 0.2$  delivered optimal results. Increasing  $p$  to 0.4 led to a significant decline in both ARI and NMI averages, indicating that excessive disruption of the graph structure and information loss outweigh the benefits of perturbation.

The gene cosine similarity-based adaptive edge perturbation closely matched the results at  $p = 0.2$  and outperformed other values. Although changing the similarity calculation method resulted in slight declines, these were minimal. These findings suggest that while gene similarity does not surpass the peak performance of random perturbation, its results are more stable, making it a preferable choice before finalizing  $p$  values.

Subsequent tests on the Stereo-seq platform and Slide-seqV2 mouse brain olfactory datasets (Supplementary Fig. S27e-f) again indicated that  $p = 0.2$  performed best, achieving the highest SC and lowest DB. These outcomes closely paralleled those obtained with gene cosine similarity. However, from  $p = 0.3$  onwards, both SC and DB metrics began to deteriorate, showing no significant differences at lower values. When comparing the clustering results of cosine similarity of genes to those obtained with  $p = 0.2$ , consistent clustering was observed in Stereo-seq's layers. In contrast, Slide-seqV2 exhibited variations, particularly clustering the accessory olfactory bulb (AOB) region and the glomerular layer (GL) together, likely due to shared gene expressions such as the CCK gene in the GL layer (Supplementary Fig. S27g). This suggests that gene similarity-based adaptive deletion may lead to biases from specific genes, although random perturbation helps mitigate this by uniformly distributing edge deletion probabilities across iterations.

In conclusion, across different platforms, the optimal probability for random edge perturbation was found to be  $p = 0.2$ . Adaptive perturbation based on gene similarity, while potentially introducing biases on high-resolution platforms like Slide-seqV2, remains the preferable choice for initial experiments due to its adaptiveness and consistent performance across most platforms, reducing the need for extensive parameter tuning.

### **Ablation studies on the overall framework**

In our ablation study, we focused on the impact of graph augmentation and debiased strategy (DS) in STAIG, both based on image feature similarity, on the clustering results. The results are shown in Supplementary Fig. S27h shows that STAIG, even without image features, outperformed the other methods in terms of median ARI and NMI. This is because of its unique augmentation strategy, which differs from conventional global node perturbation in DGI<sup>20</sup> by selectively removing edges and balancing global and local information, thereby reducing the impact of incorrect edges. Also, STAIG's contrastive learning, unlike traditional GCL<sup>21</sup> approaches, uses innovative definitions of positive and negative sample pairs, focusing on the consistency across augmented graphs and node affinities.

When the image data were included, the STAIG performance improved further. Both adaptive edge removal and DS significantly increase the ARI and NMI values, demonstrating the effectiveness of STAIG image feature extraction and its ability to utilize image data for enhanced spatial recognition. Image guidance led to more accurate edge removal, thereby lessening the impact of incorrect negative samples during training. The combination of DS with adaptive edge removal optimized the STAIG performance.

### **Patch size selection**

In STAIG, the initial step involves cropping patches from histology images based on the coordinates of spots. To maintain consistency on the Visium platform, we use the dataset-inherent parameter *fiducial\_diameter\_fullres*—the pixel diameter of a fiducial spot in the full-resolution image—as the reference size, denoted as  $d$ . We then test multiple sizes, scaling patches to 512x512 pixels for subsequent filtering processes, despite the variance in information content due to different multiples of  $d$ .

We assessed the impact of patch size using slices #151507 and #151673 from the DLPFC dataset and a dataset from human breast cancer—covering the two most common scenarios: brain and cancer tissues. The patch sizes tested were  $2d$ ,  $3d$ ,  $3.5d$ , and  $4d$ .

With existing class labels, we overlaid contours on the results for a clearer comparison. As shown in Supplementary Fig. S28a, smaller patch sizes resulted in scattered clustering of image features with no distinct

demarcation between tissue contours. For slices #151673 and #151507, the models fixated on color differences rather than textural ones, mainly dividing the images into left and right sections due to insufficient information from smaller patches, hindering the learning of classifiable features. At a  $3d$  size, #151673 and the human breast cancer datasets began to discern textures, aligning with the reference labels; however, #151507 was still disrupted by staining artifacts. At  $3.5d$ , the clustering of image features corresponded to manual labels across all three datasets, with #151507's layer 1, 3 and 5 and #151673's layers 3 and 4, along with the WM sections, closely matching. The human breast cancer imagery adeptly separated different cancer regions. However, increasing to  $4d$  improved the breast cancer clustering but blurred the brain imagery results due to limited textural details in brain slices that could not be further exploited with larger image sizes.

Considering the performance across both datasets, we decided to set the optimal patch size at 3.5 times the diameter  $d$ , balancing detail capture and clarity of clustering.

### **Gaussian blur parameter selection**

After acquiring patches from the original images, we applied Gaussian blur to further process the images and mitigate the effects of noise. We tested kernel sizes of  $3\times 3$ ,  $7\times 7$ , and  $15\times 15$ , using slices #151507 and #151673 from the DLPFC dataset, and a dataset from human breast cancer to assess the influence of the Gaussian kernel parameters across common brain and cancer tissue scenarios.

The results (Supplementary Fig. S28b) indicated that overall, Gaussian blur did not significantly alter the outcomes; changing the parameters yielded similar clustering results that followed the trends of manual labeling. Notably, the human breast cancer slices showed minimal variance in clustering, likely due to their higher image quality and pronounced textures. The brain slices, however, exhibited some differences. With a  $15\times 15$  kernel, while layer 4 in #151673 was discernible, most other regions became confused, suggesting an excessive loss of detail due to over-smoothing. Slice #151507 presented a similar pattern; despite the trend resembling manual labels, there was significant confusion.

In contrast, when applying a  $3\times 3$  Gaussian kernel, the resultant image clustering showed many scattered speckles on the visual representation, a sign that the noise was not sufficiently smoothed out. Upon comparing the outcomes, the  $7\times 7$  Gaussian kernel demonstrated the best performance, balancing detail preservation and noise reduction. Consequently, we have established the  $7\times 7$  kernel size as the default parameter for our Gaussian blur application.

### **Bandpass filter parameters selection**

Before introducing images to the BYOL framework, it is necessary to apply a bandpass filter to the image input. This step is aimed at reducing the impacts of noise and uneven staining. Here, finer textures are removed, leaving behind coarser ones, effectively highlighting the features that the model should focus on before input. We ensure a uniform scaling of the image to  $512\times 512$  pixels so that the frequency range is (0,512). We evaluated the effect of the filter's frequency band on the final outcomes using slices #151507 and #151673 from the DLPFC dataset and a dataset from human breast cancer—these represent the two most common histology image scenarios: brain and cancer tissues.

From the result shown in supplementary Fig. S28c, we observed that broader frequency bands, such as 0-300, 200-300, or 200-512, did not effectively eliminate noise. The model predominantly identified vertical striations in brain images, and the boundaries in breast cancer slice images remained blurred. Upon narrowing the frequency band to 240-280, slice #151673 began to display clustering layers similar to the class labels, and the breast cancer image clustering also started to segregate the main cancerous areas. The classification

effect on slice #151507 was still not very prominent. Further narrowing the band to 245-275, we noticed improved results; slice #151507 aligned with the distribution of class labels, with layer 1 being accurately captured. However, compressing the frequency band further to 250-270 caused the disappearance of detailed sub-clusters in all three images, suggesting excessive information filtering that could hinder the model's learning process.

From this parameter selection exercise, it is clear that information filtering is crucial for unsupervised feature extraction from images. Retaining too much information may distract the model with staining issues, preventing desired outcomes. Conversely, excessive filtering might reduce the model's learning performance. Therefore, we have determined that the optimal frequency band is 245-275. Given that we consistently fix the image size before input, this frequency band should offer a relatively consistent noise filtration and thus possess broad applicability.

### **BYOL ablation study for histology feature extraction**

In efforts to clearly delineate the individual effects of image preprocessing and the Bootstrap Your Own Latent (BYOL) framework on feature extraction from histopathological images, we embarked on a set of ablation experiments. The result is shown in supplementary Fig. S29a-c.

Initially, we scrutinized the BYOL framework's role as an unsupervised feature extractor. Rooted in the ResNet50 architecture, BYOL introduces a self-supervised loss function that doesn't rely on negative samples. STAIG's goal is to leverage BYOL for direct feature extraction from a histopathology image without prior domain-specific pretraining. Hence, we utilized the pretrained ResNet50 model offered by PyTorch, trained on generic image datasets without further tuning for histopathological contexts.

To verify BYOL's ability to learn features autonomously, we compared its results with those from the original pretrained ResNet50. Additionally, we included two representative deep learning models for comparative analysis. Given U-Net's end-to-end structure precludes direct embedding extraction, we used an encoder-decoder U-Net variant<sup>22</sup>. As U-Net operates under supervision, we pretrained it using two approaches: one with the original images from the CPM15 dataset, containing 15 histopathology images and 2905 nuclei; and the other with images from the CPM15 dataset passed through the same bandpass filter utilized in STAIG's preprocessing pipeline. Both methods achieved over 90% accuracy before commencement of the experiments indicating that the information removed by the filter is not critical for cell segmentation tasks, suggesting that basic textures are key for cell recognition.

Furthermore, we considered the Vision Transformer (ViT), another self-supervised model based on the transformer architecture that acquires image features by predicting hidden segments. We analyzed three states of ViT: one with a pretrained version sourced from the internet consistent with the conST study; another where this pretrained model was fine-tuned using patches of original images (size same with STAIG); and the third where it was fine-tuned with images that underwent BYOL's preliminary bandpass filter treatment.

Subsequently, images—original and bandpass-filtered—were input into several models: the original dataset-trained U-Net, the bandpass-filtered dataset-trained U-Net, the baseline pretrained ResNet50, BYOL post-training with bandpass-filtered images (mirroring STAIG's standard protocol), the pretrained ViT, the original-image-fine-tuned ViT, and the bandpass-filtered-image-fine-tuned ViT.

The evaluations revealed that, with original images, no model could effectively extract features from DLPFC images, demonstrating the distinct nature of brain histology images compared to ordinary images. Due to the lack of clear textures, all models focused on vertical stripes caused by uneven staining, basing their classification on color differences rather than cellular morphology.

For human breast cancer images, U-Nets, ResNet50, BYOL, and VIT were capable of extracting features to some extent, with CNN-based models performing better than VIT. Trained U-Net models could effectively recognize different tumor regions, indicating their effectiveness in handling cancer images but not brain images with smaller textural details and significant differences from the training set.

Comparing BYOL and VIT's performance on training with original images, we found no improvement, suggesting that the models were unable to extract new information due to their focus on color and noise, which they considered sufficiently distinguishable.

Interestingly, models trained with filtered images could not directly improve recognition on original images. However, models trained on original images showed improved clustering results when fed with filtered images. This was particularly evident in slice #151507, where the model trained on original images more aligned with class labels when given filtered images. Similar improvements were observed in slice #151673 and human breast cancer images, proving that the filtering preprocess helps the BYOL-based approach by directing the model's output toward basic coarse textures.

Comparing the initial filtered image results with BYOL's self-trained results on filtered images, significant improvements were observed across all three samples, demonstrating the effectiveness of the combined filtering and BYOL self-training approach. The breast cancer images provided a clear example, where the initial ResNet50 performed the worst on filtered images, but after STAIG-style learning (i.e., BYOL self-supervised learning on filtered images), it achieved the best results.

VIT generally underperformed compared to U-Net and BYOL, with its results not aligning with true contours even after self-supervised training, likely due to its large number of parameters requiring extensive datasets for optimization. However, VIT trained on filtered images showed notable improvements compared to its initial version, moving towards texture recognition, highlighting the importance of filtering.

Finally, we demonstrated STAIG's superiority over U-Net by identifying a textural anomaly region in the healthy area of the human breast cancer image (black circle). Among all methods, only STAIG with filtering and the BYOL framework could accurately recognize its contour, proving that STAIG learns more advanced semantic information beyond simple mask classification.

In summary, both image filtering and BYOL play crucial roles in STAIG's effectiveness, enabling the model to learn useful information from filtered images and achieve superior results compared to other methods.

### **Ablation studies for integration**

In this section, we conduct ablation experiments on the integration capabilities of the STAIG framework using four ST slice datasets from two donors in the DLPFC (#151507, #151508, and #151675, #151676) and mouse olfactory bulb datasets from two different platforms, SlideV2 and Stereo-seq. The integration capabilities are evaluated using metrics such as BatchKL and iLISI, where lower BatchKL and higher iLISI values indicate stronger integration.

Initially, we perform PCA dimension reduction followed by UMAP visualization on the raw data preprocessed for integration tasks (see Supplementary Fig. S30a). The UMAP reveals clear separation between samples from different donors in the DLPFC dataset and distinct separations between datasets from different platforms in the mouse olfactory bulb. The observed lower iLISI and higher BatchKL values suggest significant batch effects.

We compare the STAIG framework with another form of contrastive learning framework represented by ConST and GraphST, as well as the classic DGI framework. The STAIG framework employs a contrast strategy where random edges are removed (shown in Supplementary Fig. S30b). For a node in one augmented graph, positive pairs are defined as: 1) the same node in the other augmented graph, 2) the node and its first-order neighbors within the same graph, and 3) the node and its first-order neighbors in the other graph. Negative samples are defined as: 1) the node and all non-neighbor nodes within the same graph, and 2) the node and all non-neighbor nodes in the other graph.

In contrast, DGI, which GraphST and ConST utilize, maintains the original structure while randomly shuffling node positions to generate a contrast graph (Supplementary Fig. S30c). Positive samples in DGI are defined as a node and its neighbors' readouts from original graph, while negative samples are the node's corresponding position in the augmented graph. DGI focuses more on global information, whereas STAIG, with its node-to-node comparisons, prioritizes local information.

Furthermore, we test both frameworks under intra-slice and inter-slice conditions. For intra-slice, STAIG's negative samples consider non-neighbor nodes within the same slice without including disconnected graph nodes. DGI's node perturbations are confined within the corresponding slice. For inter-slice, STAIG considers all non-neighbor nodes, even if disconnected, while DGI's random node changes occur across all slices.

Results shown in Supplementary Fig. S30d-e indicate that both DGI and STAIG perform better in intra-slice scenarios. Moreover, the impact of intra-slice conditions is more pronounced in STAIG, whereas DGI is only slightly affected. The DGI framework fails to integrate data from two different platforms, with iLISI close to 1 and clear clustering separations in UMAP visualizations. On the DLPFC dataset, DGI shows some integration capability; however, the clusters from different donors are not fully overlapping, unlike STAIG, which successfully integrates both intra-slice and inter-slice scenarios, achieving lower BatchKL and higher iLISI values.

We evaluated the integration performance on the DLPFC dataset using ARI and NMI. Under the STAIG framework, both ARI and NMI scores were significantly higher, indicating better clustering and retention of biological features. In contrast, while DGI achieved some degree of integration, the latent space data lost certain biological characteristics, leading to lower ARI and NMI scores. Additionally, STAIG maintained high clustering metrics such as SC and low DB even in cross-platform scenarios, whereas DGI's SC and DB scores were less favorable.

We then conduct further ablation studies on STAIG's loss function to pinpoint the source of its integration capabilities. We compare several similar loss functions: 1) removing first-order neighbor positive samples (Supplementary Fig. S30f) and 2) on top of that, further removing intra-graph negative samples (Supplementary Fig. S30g), making the loss function a classic InfoNCE<sup>13</sup> loss.

The results (Supplementary Fig. S30h-i) demonstrate that regardless of the loss function, under the same graph augmentation strategy, all achieve better integration and clustering results than DGI. The choice of first-order



neighborhood positive samples significantly enhances integration capabilities. Removing these samples results in integration metrics similar to those of InfoNCE, suggesting that removing intra-graph negative samples does not significantly enhance integration. The best integration results are achieved intra-slice when negative samples are restricted to a single slice, likely because node-to-node local comparisons allow the model to focus more on common gene expression patterns. Introducing inter-slice negative samples, however, appears to degrade performance by causing the model to also focus on batch effects across slices.

Lastly, we assess the impact of gene random masking on integration capabilities (Supplementary Fig. S31a-c). Removing gene random masks reduces integration and clustering capabilities for both STAIG and DGI, indicating that random masking enhances model generalization by simulating batch effects and improving robustness. Moreover, the impact of random masking varies between contrastive architectures; it has a lower effect on DGI, while significantly enhancing STAIG's integration capabilities.

From these results, it is evident that STAIG's integration capability primarily stems from its contrastive framework, which differs significantly from the traditional DGI framework. DGI ensures mutual information between a node and its neighbors' summaries but inevitably leads to a loss of local information, such as co-expressed genes. Zhang et al.<sup>8</sup> also support this view, suggesting that compared to DGI, InfoNCE (which shares the same augmentation framework as STAIG) can adaptively learn both spatially adjacent dependency and gene expression relationships. Additionally, the gene perturbation strategy, where random zeroing of some genes simulates batch effects, enhances the model's generalization capabilities during training. Furthermore, focusing on intra-slice negative sample selection yields the best integration results, indicating that the model can learn common gene expression patterns within a single slice without the adverse effects of cross-slice batch effects.

### **Stability and efficiency of the averaged structure**

On hardware beyond our tested range, directly generating augmented graphs for STAIG may pose potential stability issues. To address this concern, we adopted an optional approach that generates multiple graph structures and averages them to form the augmented graphs  $G^1$  and  $G^2$ . Specifically, for  $G^1$  or  $G^2$ , we first generated  $n$  graph structures using the edge removal strategy described in the Methods section of the main text. We then averaged the results for each edge across these  $n$  graphs. An edge was removed if its average value was less than  $P_{i,j}$ ; otherwise, it was retained.

We evaluated the impact of  $n$  on the results using the average ARI and NMI scores on the DLPFC dataset, as well as the ARI and NMI scores on the STARmap dataset, which does not include image features. As shown in Supplementary Fig. S31d-e, the performance of STAIG remained consistent when  $n$  was set to 10 or less. However, when  $n$  exceeded 10, we noticed a slight decline in performance. This decrease may be attributed to the reduced variability in the training structures due to excessive averaging, which could lead to structural rigidity and, consequently, a slight drop in performance. Nonetheless, the overall change in performance was minimal.

Next, we examined the impact on time efficiency (Supplementary Fig. S31f). Using the DLPFC dataset, we set the time efficiency of directly generating augmented graphs ( $n = 1$ ) as the baseline. We observed that as  $n$  increased, the time required also increased, doubling when  $n = 20$ .

Based on our test results, the resampling process produced stable outcomes but increased the time cost. We believe that, within the range of hardware we tested, the methodology described in the main text, where  $n$  is set to 1, is robust across various datasets and scenarios. However, we acknowledge that this approach may not completely eliminate potential issues in every

test or dataset. As a solution, we offer users the option to apply the resampling process in their own tests, which we expect will help mitigate any bias introduced by the sampling process.

If users opt for the resampling strategy, we recommend setting  $n$  to 5 as it offers the best balance between time efficiency and accuracy. At this level, the time required increased by only 18% compared to  $n = 1$ , while maintaining accuracy and improving the stability of the results.

### **The principle of STAIG integration in latent space**

The batch integration capability of STAIG is derived from the generalization ability of the contrastive learning framework, which allows the model to handle multiple similar datasets. Although there is currently no comprehensive theory explaining why contrastive learning at the sample (node) level leads to generalization and ultimately to clustering<sup>23</sup>, Huang et al.<sup>24</sup> have proposed a hypothesis based on results from contrastive learning in the image domain. They suggest that this phenomenon arises because, although samples within the same category may differ in their raw features, their augmented counterparts (positive samples) become closer in the latent space.

While similar theoretical studies in contrastive learning specific to the graph domain are still lacking<sup>25</sup>, Huang et al.'s insights have inspired us to explain STAIG's batch integration mechanism for spatial transcriptomics data. By observing changes in sample positions within the latent space, we interpret how STAIG achieves batch integration.

We take the integration of datasets across platforms with the largest batch effects as an example. We designate the Stereo-seq dataset as slice A and the Slide-seqV2 dataset as slice B. Supplementary Fig. S32a shows their clustering results. We focus on samples from cluster 2 and cluster 7 for closer observation and analysis.

Supplementary Fig. S32b presents the UMAP visualization of the embeddings at the initial stage. As seen in the figure, at the beginning of the latent space (epoch = 0), the two slices are distinctly separated. However, as the number of epochs increases, the embeddings of the two slices begin to integrate.

The term intra-slice refers to negative samples that only consider nodes within the same slice, while inter-slice refers to negative samples that consider nodes across all slices. Supplementary Fig. S32c compares the average cosine distances (where a smaller value indicates closer proximity) between clusters 2 and 7 within slice A, clusters 2 and 7 within slice B, and between clusters 2 and 7 across different slices. Supplementary Fig. S32d shows the cosine distances between the centers of samples from different classes within the same slice, as well as the distances between centers of the same class samples across different slices.

We observe that in the initial latent space, all samples, regardless of whether they belong to the same category or come from the same slice, are positioned close together. At the early stage, the distances between samples of the same class even within the same slice begin to increase. This occurs because contrastive learning operates at the node level, where all nodes, except for a node's immediate neighbors, are treated as negative samples. As a result, every node undergoes a certain degree of separation from others in the latent space.

As the number of epochs increases, under the intra-slice setting, the model tends to prioritize distinguishing between different categories rather than different slices. The distance between samples from different classes continues to increase, while the distance between samples of the same class decreases. The center distance between same-class samples from different slices gradually reduces, leading to successful integration. On the other hand, in the inter-slice setting, the model focuses more on the differences between slices, rather than class distinctions.

Supplementary Fig. S32e and Fig. S33a shows the change of Euclidean distance and angular differences (cosine distance) between the centers of clusters 2 and 7 from different slices after each iteration. We can observe that, under the intra-slice setting, the centers of same-class samples from different slices change by almost the same distance and angle with each iteration. This consistent movement results in their separation from other class samples and their convergence into the same cluster. In contrast, under the inter-slice setting, samples from the same slice are more likely to move in the same direction, leading to less integration between same-class samples across different slices.

The primary reason for this phenomenon is that, under the intra-slice setting, the loss calculations for the same class across different slices tend to be similar (Supplementary Fig. S33b). When negative samples consider nodes across slices, due to the initial angular deviation between the entire slices, the average loss for the same class between different slices varies significantly. This angular discrepancy introduces differences in how the model perceives and learns from samples across slices, leading to variations in the loss values.

In STAIG, cosine similarity is used as the similarity function. When we simplify the STAIG function using the InfoNCE loss function:

$$l(h_i^1) = -\log \frac{e^{\cosine(h_i^1, h_i^2)}}{e^{\cosine(h_i^1, h_i^2)} + \sum_{(j \neq i)} (e^{\cosine(h_i^1, h_j^2)})} \quad (1)$$

the result shows that the term  $\frac{e^{\cosine(h_i^1, h_j^2)}}{e^{\cosine(h_i^1, h_i^2)}}$  (the ratio between negative and positive similarities) for the same class node is nearly identical across different slices. While the embeddings of the two slices may occupy different positions in the initial latent space, the relative angles between different class labels within each slice remain similar. This means that although batch effects exist between slices, the angular differences between classes are consistent across slices. Supplementary Fig. S33c show the heatmaps of cosine similarity between different class labels, comparing cross-platform integration and the integration of different slices from the Visium platform in the initial latent space. From these heatmaps, we can clearly observe that the angular differences between different class labels across slices remain consistent, reflected as similar light-dark trends in the heatmaps. This further validates the aforementioned conclusion.

Consequently, during the training process, STAIG's deep model becomes more inclined to focus on class distinctions. After the initial iterations, samples with the same class label from different slices are pushed in the same direction, leading to their integration.

The integration capability of STAIG is closely linked to the unique characteristics of batch effects in spatial transcriptomics data. As a relatively new data type, spatial transcriptomics has been less explored in terms of its batch effect properties. Previous studies have not focused on the characteristics of consecutive slices in the latent space after graph augmentation and GNN dimensionality reduction. This led them to apply experience from single-cell RNA sequencing (sc-RNA) analysis, which added unnecessary complexity to the integration process.

In our research, we discovered that the batch effects in spatial transcriptomics are more about directional differences at the global level. In the latent space, coordinate basis vectors differ between slices, while the angular differences between domains within the same slice are similar. This is reflected in UMAP visualizations, where methods with weaker integration capabilities show shapes that are similar across consecutive slices but exhibit a certain degree of rotation (Supplementary Fig. S13). Describing batch effects in spatial transcriptomics is not our primary focus, but through extensive experiments across multiple platforms and datasets, we demonstrated that STAIG consistently ensures batch integration within our test range.

STAIG's integration capability is not coincidental; it stems from its node-level contrastive approach. From a broader perspective, the model pays attention to differences between categories from the very beginning, allowing it to overlook the angular differences caused by batch effects between slices. In subsequent iterations, the model increasingly learns and extracts

features relevant to category distinctions, further reducing the impact of batch effects. Ultimately, samples from the same category, regardless of their slice origin, converge into the same cluster, resulting in embeddings that are effectively free from batch effects.

### **Comparison between whole batch and separate training in STAIG.**

Here, we demonstrate the differences between whole batch and separate training in STAIG. For whole batch training, the GCN computation formula is:

$$H = \sigma(AW),$$

where  $A$  is the merged adjacency matrix,  $X$  is the input data,  $W$  represents the GCN parameters, and  $\sigma$  is the activation function.

For separate training, the GCN computation formulas are:

$$H_1 = \sigma(A_1X_1W_1), H_2 = \sigma(A_2X_2W_2),$$

where  $A_1$ ,  $A_2$  are adjacency matrices for the two datasets,  $X_1$ ,  $X_2$  are their respective input data,  $W_1$  and  $W_2$ , are independent GCN parameters, and  $\sigma$  is the activation function.

From a formulaic perspective, the distinction between whole batch and separate training lies in the iteration process. In whole batch training, the loss is calculated as the average of all points across all slices, with  $W$ 's gradient updates computed based on the total loss. In separate training,  $W_1$  and  $W_2$  are updated independently, with each backward gradient calculated based on the loss from corresponding slice nodes. This process amplifies differences during iteration, causing nodes from different slices to map to distinct locations in the latent space.

According to Hu et al.'s research<sup>26</sup>, mini-batch performance is inferior to whole batch training. Additionally, their experiments involving dataset division into training and test sets for GCN training revealed that test set points did not completely align with training set points in UMAP visualization, further demonstrating the importance of shared parameters.

To further validate these findings, we conducted cross-platform slice integration experiments under four scenarios: First, whole batch training. Second, training using the Stereo-seq platform model and generating embeddings for both Stereo-seq and slide-seqV2 platform datasets. Third, sequential training - first with the Stereo-seq platform dataset, followed by the slide-seqV2 platform dataset. Fourth, alternating training between Stereo-seq and slide-seqV2 platform datasets (switching every epoch).

The UMAP visualization results are presented in Supplementary Figure S34a. The results clearly demonstrate that only whole batch training ensures proper slice integration. Alternating training ranks second in performance, though its SC and DB metrics are suboptimal. The other two approaches fail to achieve integration, exhibiting high BatchKL values and very low iLISI scores.

### **STAIG's performance with different feature gene selection methods.**

In addition to the default high-expression gene selection method used by Scanpy (VST, or the variance of standardized counts), we incorporated four additional gene selection methods in STAIG to provide users with more options. The first method is an empirical distribution-based approach, for which we selected the MVP method<sup>1</sup> (mean.var.plot) that applies the Z statistic of mean and variance. The second approach focuses on minimizing redundancy in gene selection, represented by the

scGeneClust method<sup>27</sup>. Thirdly, we included a method based on co-expression gene networks, using PyWGCNA<sup>28</sup>. Lastly, for spatially variable gene selection, we selected SpatialDE<sup>29</sup>.

Users can select any of these four methods within STAIG. To evaluate their performance, we tested these methods along with the default VST on the DLPFC dataset. All methods were applied with default parameters, except for PyWGCNA, where we selected the top 100 hub genes as variable genes from each module.

A box plot of the results on the DLPFC dataset is shown in Supplementary Fig. S34b. From these results, there is no significant difference in performance between VST, MVP, scGeneClust, and SpatialDE, likely because the self-supervised learning capabilities of GNNs are already robust enough to capture relevant information, reducing dependency on gene pre-selection. PyWGCNA showed slightly lower performance, likely due to its original design for analyzing gene modules rather than directly identifying variable genes.

### **Exploring the integration capacity of STAIG with artificially modified batch effects.**

The integration capacity of STAIG stems from its strong generalization capability. To further examine the limits of STAIG's integration performance, we used scDesign3<sup>30</sup> to artificially modify batch effects in the data, allowing us to systematically explore the model's integration threshold.

Specifically, we applied scDesign3 to data from two donors (#151507 and #151675) in the DLPFC dataset. Initially, we measured the original batch marginal distribution (Supplementary Fig. S34c), approximated a normal distribution with a mean of 0.12 and an SD of 0.23. We then simulated datasets with specified batch effect intensities by varying the mean and SD values. The mean values used were 0.1, 0.3, 0.5, and 0.8, while the SD values were set to 0.2, 0.5, and 0.8. STAIG's integration capability was tested on each combination. The results are presented in Supplementary Table S4, with UMAP visualizations of selected combinations shown in Supplementary Fig. S34d. Based on previous integration metrics, we set the success thresholds for integration at a BatchKL below 0.5 and iLISI above 1.5. As shown in the table, when the mean is below 0.3, STAIG achieves good integration across all SD values. For mean values of 0.5, integration remains effective with SDs below 0.5, while for a mean of 0.8, integration is still effective when SD is 0.2.

These results demonstrate that STAIG maintains a high integration capacity even under substantial batch effects. Users can apply scDesign3 to test whether the batch effects in their data fall within STAIG's effective range, helping ensure optimal application of the model.

## Supplementary Tables

**Table S1.** Description of all ST datasets used in this study.

| Platform    | Tissue                                       | Section                                    | #Spots/Bins | Related figures                                                                                                                               | Reference         |
|-------------|----------------------------------------------|--------------------------------------------|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| 10x Visium  | Human dorsolateral prefrontal cortex (DLPFC) | 151507                                     | 4226        | Fig. 2a, Fig. 3a, Fig. 5c Fig. S1, Fig. S13, Fig. S22, Fig. S23a-b, Fig. S23f, Fig. S27d, h, Fig. S28, Fig. S29, Fig. S30, Fig. S31, Fig. S33 | [ <sup>31</sup> ] |
|             |                                              | 151508                                     | 4384        | Fig. 2a, Fig. 5c, Fig. S1, Fig. S13, Fig. S22, Fig. S23a-b, Fig. S23f, Fig. S27d, h, Fig. S30, Fig. S31                                       |                   |
|             |                                              | 151509                                     | 4789        | Fig. 2a, Fig. S1, Fig. S22, Fig. S23a-b, Fig. S23f, Fig. S27d, h, Fig. S31                                                                    |                   |
|             |                                              | 151510                                     | 4634        | Fig. 2a, Fig. S1, Fig. S22, Fig. S23a-b, Fig. S23f, Fig. S27d, h, Fig. S31                                                                    |                   |
|             |                                              | 151669                                     | 3661        | Fig. 2a, Fig. S1, Fig. S22, Fig. S23a-b, Fig. S23f, Fig. S27d, h, Fig. S31                                                                    |                   |
|             |                                              | 151670                                     | 3498        | Fig. 2a, Fig. S1, Fig. S22, Fig. S23a-b, Fig. S23f, Fig. S27d, h, Fig. S31                                                                    |                   |
|             |                                              | 151671                                     | 4110        | Fig. 2a, Fig. S2, Fig. S22, Fig. S23a-b, Fig. S23f, Fig. S27d, h, Fig. S31                                                                    |                   |
|             |                                              | 151672                                     | 4015        | Fig. 2a, Fig. 2b, Fig. S2, Fig. S22, Fig. S23a-b, Fig. S23f, Fig. S27d, h, Fig. S31                                                           |                   |
|             |                                              | 151673                                     | 3639        | Fig. 2a, Fig. 2c-e, Fig. 3b, Fig. S2, Fig. S22, Fig. S23a-b, Fig. S23d, Fig. S23f, Fig. S27d, h, Fig. S28, Fig. S29, Fig. S31                 |                   |
|             |                                              | 151674                                     | 3673        | Fig. 2a, Fig. S2, Fig. S22, Fig. S23a-b, Fig. S23f, Fig. S27d, h, Fig. S31                                                                    |                   |
|             |                                              | 151675                                     | 3592        | Fig. 2a, Fig. 5a, Fig. 5c Fig. S2, Fig. S11, Fig. S13, Fig. S22, Fig. S23a-b, Fig. S23f, Fig. S27d, h, Fig. S30, Fig. S31                     |                   |
|             |                                              | 151676                                     | 3460        | Fig. 2a, Fig. 5a, Fig. 5c, Fig. S2, Fig. S11, Fig. S13, Fig. S22, Fig. S23a-b, Fig. S23f, Fig. S27d, h, Fig. S30, Fig. S31, Fig. S33          |                   |
|             | Mouse brain                                  | Mouse Brain Section 1 (Sagittal-Anterior)  | 2695        | Fig. 2g, Fig. 5b, Fig. S4, Fig. S12                                                                                                           | 10x Visium demo   |
|             |                                              | Mouse Brain Section 1 (Sagittal-Posterior) | 3355        | Fig. 2f, Fig. 5b, Fig. S3, Fig. S12                                                                                                           |                   |
|             | Human breast                                 | Human Breast Cancer Section 1              | 3798        | Fig. 3c, Fig. 6, Fig. S16, Fig. S17, Fig. S13, Fig. 23c, Fig. 23e, Fig. S28, Fig. S29                                                         |                   |
|             | zebrafish melanoma                           | GSM4838131 Visium A                        | 2179        | Fig. 7, Fig. S18, Fig. S19, Fig. S20, Fig. S21                                                                                                | [ <sup>32</sup> ] |
|             |                                              | GSM4838132 Visium B                        | 2677        |                                                                                                                                               |                   |
| Stereo-seq  | Mouse olfactory bulb                         | X                                          | 19109       | Fig. 4a-c, Fig. 5c, Fig. S5, Fig. S15, Fig. S24, Fig. S27b-c, e-f, Fig. S30, Fig. S32, Fig. S33                                               | [ <sup>33</sup> ] |
| Slide-seqV2 | mouse hippocampus                            | Puck_200115_08                             | 52,869      | Fig. 4f, Fig. S7                                                                                                                              | [ <sup>34</sup> ] |
|             | mouse olfactory bulb                         | Puck_200127_15                             | 20139       | Fig. 4d-e, Fig. 5c, Fig. S6, Fig. S15, Fig. 27e-g, Fig. S30, Fig. S32, Fig. S33                                                               |                   |
| STARmap     | Mouse visual cortex                          | X                                          | 1207        | Fig. 4g, Fig. S8, Fig. S25, Fig. 27a, Fig. S31                                                                                                | [ <sup>35</sup> ] |
| MERFISH     | Mouse visual cortex                          | Mouse1.AUD_TEA_VIS.242.unexpand            | 5995        | Fig. 4h, Fig. 5d, Fig. S10, Fig. S14, Fig. S26, Fig. S27b-c                                                                                   | [ <sup>36</sup> ] |
|             | Mouse visual cortex                          | Mouse2.AUD_TEA_VIS.242.unexpand            | 2479        | Fig. 5d, Fig. S14                                                                                                                             |                   |
|             | Human middle temporal gyrus                  | H18.06.006.MTG.4000.expand.rep3            | 3970        | Fig. 4h, Fig. S9                                                                                                                              |                   |

**Table S2.** Sample IDs and URLs of data downloaded from the Allen Brain Atlas in this study.

| Related figures   | ABA ID    | ABA URL                                                                                                                           |
|-------------------|-----------|-----------------------------------------------------------------------------------------------------------------------------------|
| Fig. 2f           | 100884070 | <a href="https://atlas.brain-map.org/atlas?atlas=2&amp;plate=100884070">atlas.brain-map.org/atlas?atlas=2&amp;plate=100884070</a> |
| Fig. 4d, Fig. S6  | 100960460 | <a href="https://atlas.brain-map.org/atlas?atlas=1&amp;plate=100960460">atlas.brain-map.org/atlas?atlas=1&amp;plate=100960460</a> |
| Fig. 4f, Fig. S7  | 100960232 | <a href="https://atlas.brain-map.org/atlas?atlas=1&amp;plate=100960232">atlas.brain-map.org/atlas?atlas=1&amp;plate=100960232</a> |
| Fig. 5c, Fig. S11 | 100883770 | <a href="https://atlas.brain-map.org/atlas?atlas=2&amp;plate=100883770">atlas.brain-map.org/atlas?atlas=2&amp;plate=100883770</a> |

**Table S3.** Time cost of STAIG on various datasets using A100 GPU.

| Dataset              | Platform    | #Spots/Bins | Extract image features | Data preprocessing | Train GNN |
|----------------------|-------------|-------------|------------------------|--------------------|-----------|
| #151673              | Visium 10x  | 3639        | 2min 52s               | 7.05s              | 10.74s    |
| Human breast Cancer  | Visium 10x  | 3798        | 3min 04s               | 8.89s              | 11.82s    |
| Mouse olfactory bulb | Stereo-seq  | 19109       | X                      | 2min 16s           | 1min 53s  |
| mouse olfactory bulb | Slide-seqV2 | 20139       | X                      | 2min 27s           | 2min 05s  |
| Mouse visual cortex  | STARmap     | 1207        | X                      | 0.98s              | 6.53s     |

**Table S4.** STAIG performance under varying batch effects on DLPFC slices #151507 and #151675 integration tasks.

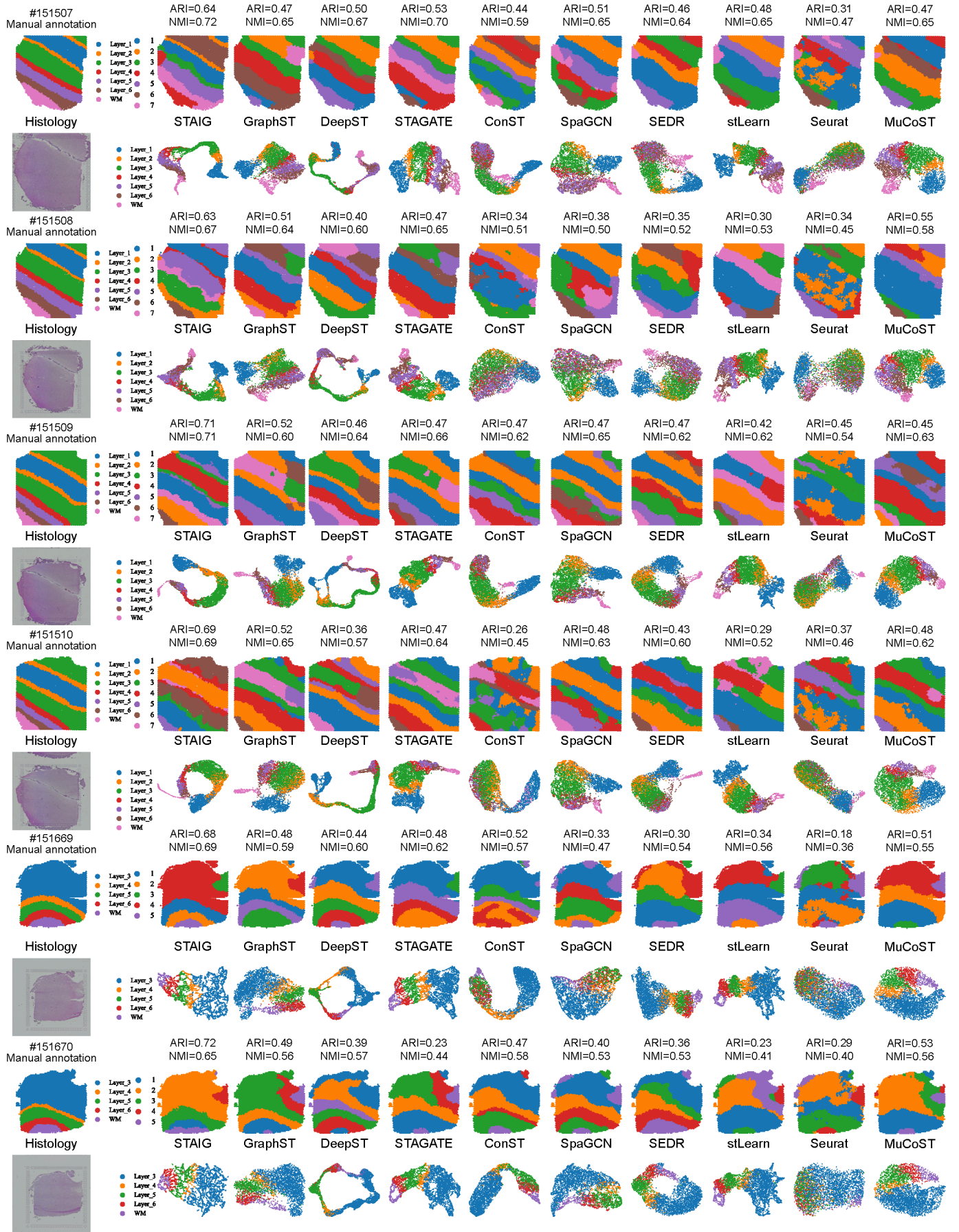
| Mean     | SD       | BatchKL | iLISI | ARI  | NMI  |
|----------|----------|---------|-------|------|------|
| Raw data | Raw data | 0.16    | 1.83  | 0.61 | 0.72 |
| 0.1      | 0.2      | 0.14    | 1.86  | 0.60 | 0.70 |
| 0.1      | 0.5      | 0.20    | 1.81  | 0.58 | 0.70 |
| 0.1      | 0.8      | 0.28    | 1.73  | 0.56 | 0.74 |
| 0.3      | 0.2      | 0.19    | 1.78  | 0.57 | 0.69 |
| 0.3      | 0.5      | 0.26    | 1.73  | 0.55 | 0.65 |
| 0.3      | 0.8      | 0.35    | 1.66  | 0.55 | 0.62 |
| 0.5      | 0.2      | 0.27    | 1.72  | 0.51 | 0.63 |
| 0.5      | 0.5      | 0.39    | 1.55  | 0.52 | 0.59 |
| 0.5      | 0.8      | 0.55    | 1.36  | 0.49 | 0.55 |
| 0.8      | 0.2      | 0.42    | 1.52  | 0.51 | 0.58 |
| 0.8      | 0.5      | 0.55    | 1.34  | 0.38 | 0.47 |
| 0.8      | 0.8      | 0.66    | 1.22  | 0.33 | 0.42 |

## Supplementary Figures

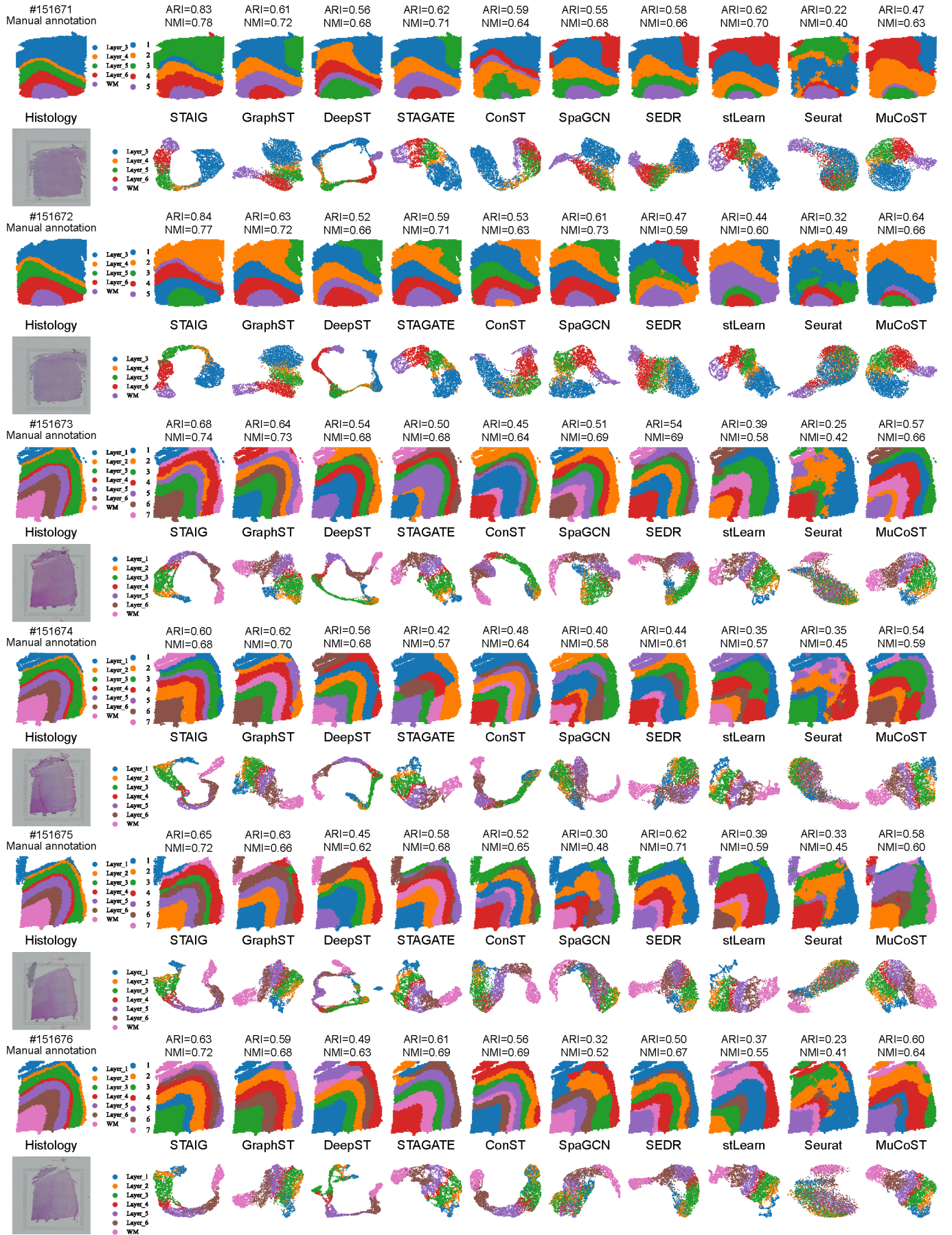
|                                                                                                                                                                                                                                                                                            |    |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Fig. S1 H&E stained image, manual annotations, clustering result with ARI, NMI and UMAP visualization by STAIG and baseline methods (GraphST, DeepST, STAGATE, ConST, SpaGCN, SEDR, stLearn, Seurat, MuCoST) on DLPFC slices #151507, #151508, #151509, #151510, #151669, and #151670..... | 22 |
| Fig. S2 H&E stained image, manual annotations, clustering result with ARI, NMI and UMAP visualization by STAIG and baseline methods on DLPFC slices #151671, #151672, #151673, #151674, #151675, and #151676. ....                                                                         | 23 |
| Fig. S3 H&E stained image and clustering results with SC and DB by STAIG and baseline methods on mouse brain posterior tissue. ....                                                                                                                                                        | 24 |
| Fig. S4 Manual annotations and clustering results with ARI and NMI by STAIG and baseline methods on mouse brain anterior tissue, including a bar chart of NMI results.....                                                                                                                 | 25 |
| Fig. S5 supplementary figures on Stereo-seq mouse olfactory bulb tissue. ....                                                                                                                                                                                                              | 26 |
| Fig. S6 supplementary figures on Slide-seqV2 mouse olfactory bulb tissue.....                                                                                                                                                                                                              | 27 |
| Fig. S7 Annotations from the Allen Reference Atlas and clustering results with SC and DB by STAIG and baseline methods on Slide-seqV2 mouse hippocampus tissue.....                                                                                                                        | 28 |
| Fig. S8 Manual annotations and clustering results with ARI and NMI by STAIG and baseline methods on STARmap mouse visual cortex dataset. ....                                                                                                                                              | 29 |
| Fig. S9 Manual annotations and clustering results with SC and DB by STAIG and baseline methods on the MERFISH human middle temporal gyrus dataset.....                                                                                                                                     | 30 |
| Fig. S10 Manual annotations and clustering results with SC and DB by STAIG and baseline methods on the MERFISH mouse visual cortex dataset. ....                                                                                                                                           | 31 |
| Fig. S11 Vertical integration of STAIG and compared methods on DLPFC slices from the same source (#151675, #151676). Includes manual annotations, UMAP, ARI, NMI, BatchKL, and ILISI metrics.....                                                                                          | 32 |
| Fig. S12 Horizontal integration of STAIG and compared methods on mouse anterior and posterior brain tissue slices. Includes H&E stained images, annotations from the Allen Reference Atlas, integrated results, UMAP, SC, DB, BatchKL and ILISI metrics. ....                              | 33 |
| Fig. S13 Integration of STAIG and compared methods on DLPFC slices from the different source (DLPFC #151507 with #151508 and #151675 and #151676). Includes manual annotations, UMAP, ARI, NMI, BatchKL, and ILISI metrics.....                                                            | 34 |
| Fig. S14 Integration of STAIG and compared methods on MERFISH mouse visual cortex datasets from the different mouse. Includes manual annotations, UMAP, SC, DB, BatchKL, and ILISI metrics.....                                                                                            | 35 |
| Fig. S15 Cross platform integration of STAIG and compared methods on mouse olfactory bulb datasets (Stereo-seq and Slide-seqV2). Includes clustering results, UMAP, SC, DB, BatchKL, and ILISI metrics.....                                                                                | 36 |
| Fig. S16 Manual annotation and clustering results with ARI and NMI by STAIG and baseline methods on human breast cancer dataset.....                                                                                                                                                       | 37 |



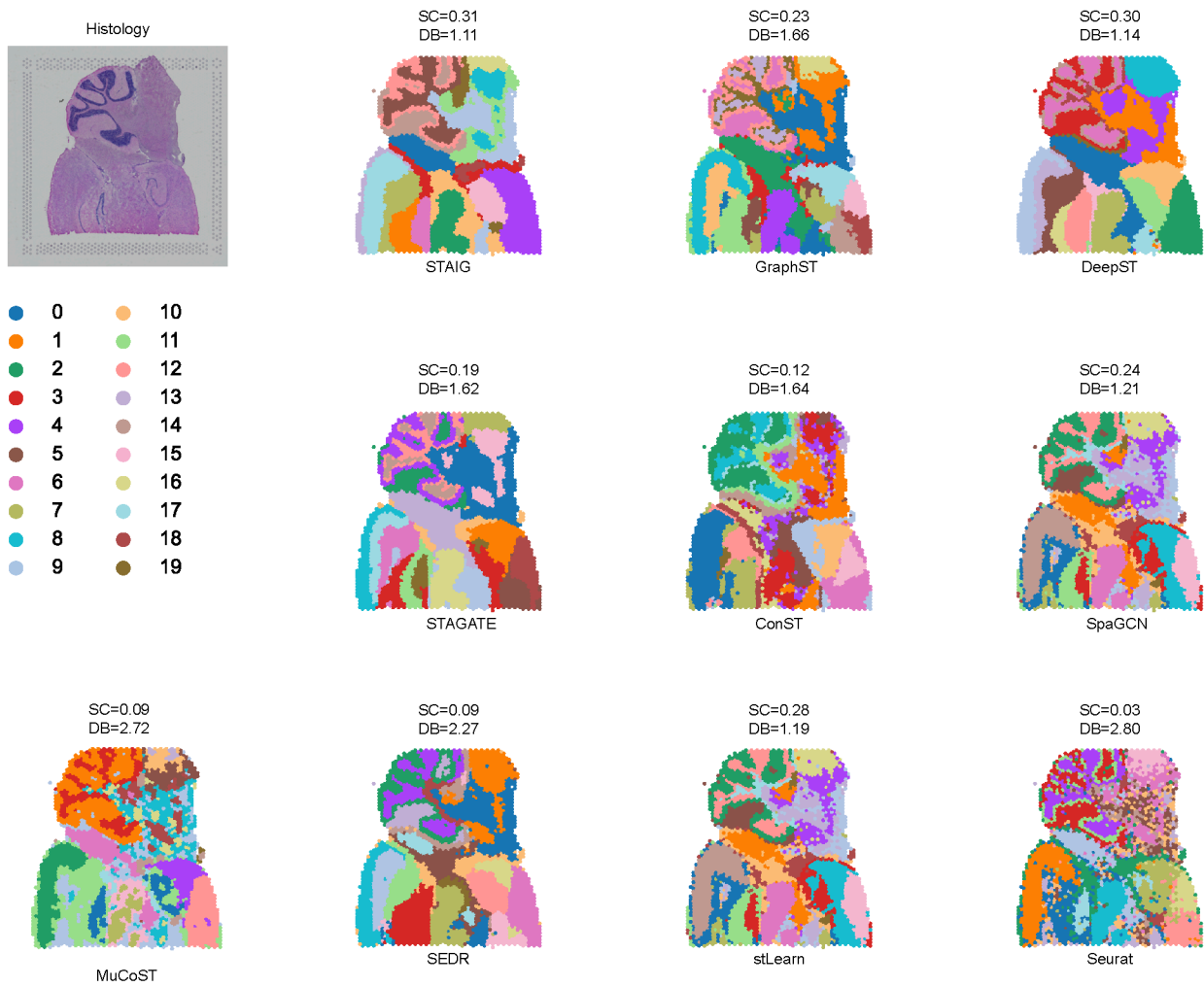
|                                                                                                                                                                                                                                       |           |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| <b>Fig. S17 Differential Gene Expression (DGE) and Gene Ontology (GO) analysis of Clusters 3 and 4 on human breast cancer dataset.....</b>                                                                                            | <b>38</b> |
| <b>Fig. S18 Annotation and cluster results with SC and DB by STAIG and baseline methods on zebrafish melanoma on slice A. ....</b>                                                                                                    | <b>39</b> |
| <b>Fig. S19 Annotation and cluster results with SC and DB by STAIG and baseline methods on zebrafish melanoma on slice B. ....</b>                                                                                                    | <b>40</b> |
| <b>Fig. S20 Comparison of STAIG and TESLA performance on zebrafish melanoma dataset.....</b>                                                                                                                                          | <b>41</b> |
| <b>Fig. S21 DGE and GO analysis of Fine-tuned subregions on zebrafish melanoma dataset.....</b>                                                                                                                                       | <b>42</b> |
| <b>Fig. S22 Main hyperparameter optimization on 10x Visium .....</b>                                                                                                                                                                  | <b>43</b> |
| <b>Fig. S23 Analysis of masked feature ratios in graph augmentation on Visium and cluster number assessments in a debiased strategy. ....</b>                                                                                         | <b>45</b> |
| <b>Fig. S24 Main hyperparameter optimization on Stereo-seq platform.....</b>                                                                                                                                                          | <b>46</b> |
| <b>Fig. S25 Main hyperparameter optimization on STARmap platform. ....</b>                                                                                                                                                            | <b>47</b> |
| <b>Fig. S26 Main hyperparameter optimization on MERFISH platform.....</b>                                                                                                                                                             | <b>48</b> |
| <b>Fig. S27 Analysis of masked feature ratios in graph augmentation on other platform, comparing adaptive edge perturbation based on gene similarity with random perturbation and ablation studies on the overall framework. ....</b> | <b>49</b> |
| <b>Fig. S28 Analysis of parameter impacts on feature extraction (DLPFC #151507, #151673 and human breast cancer). ....</b>                                                                                                            | <b>50</b> |
| <b>Fig. S29 BYOL ablation study for histology feature extraction.....</b>                                                                                                                                                             | <b>51</b> |
| <b>Fig. S30 Ablation studies for integration.....</b>                                                                                                                                                                                 | <b>52</b> |
| <b>Fig. S31 Ablation studies for integration and stability of the averaged structure. ....</b>                                                                                                                                        | <b>53</b> |
| <b>Fig. S32 The principle of STAIG integration in latent space. ....</b>                                                                                                                                                              | <b>54</b> |
| <b>Fig. S33 The principle of STAIG integration in latent space and the heatmaps of cosine similarity between different class labels. ....</b>                                                                                         | <b>55</b> |
| <b>Fig. S34 Effect of different training methods and different gene selection methods in STAIG and Batch effect performance test. ....</b>                                                                                            | <b>56</b> |



**Fig. S1** H&E stained image, manual annotations, clustering result with ARI, NMI and UMAP visualization by STAIG and baseline methods (GraphST, DeepST, STAGATE, ConST, SpaGCN, SEDR, stLearn, Seurat, MuCoST) on DLPFC slices #151507, #151508, #151509, #151510, #151669, and #151670.

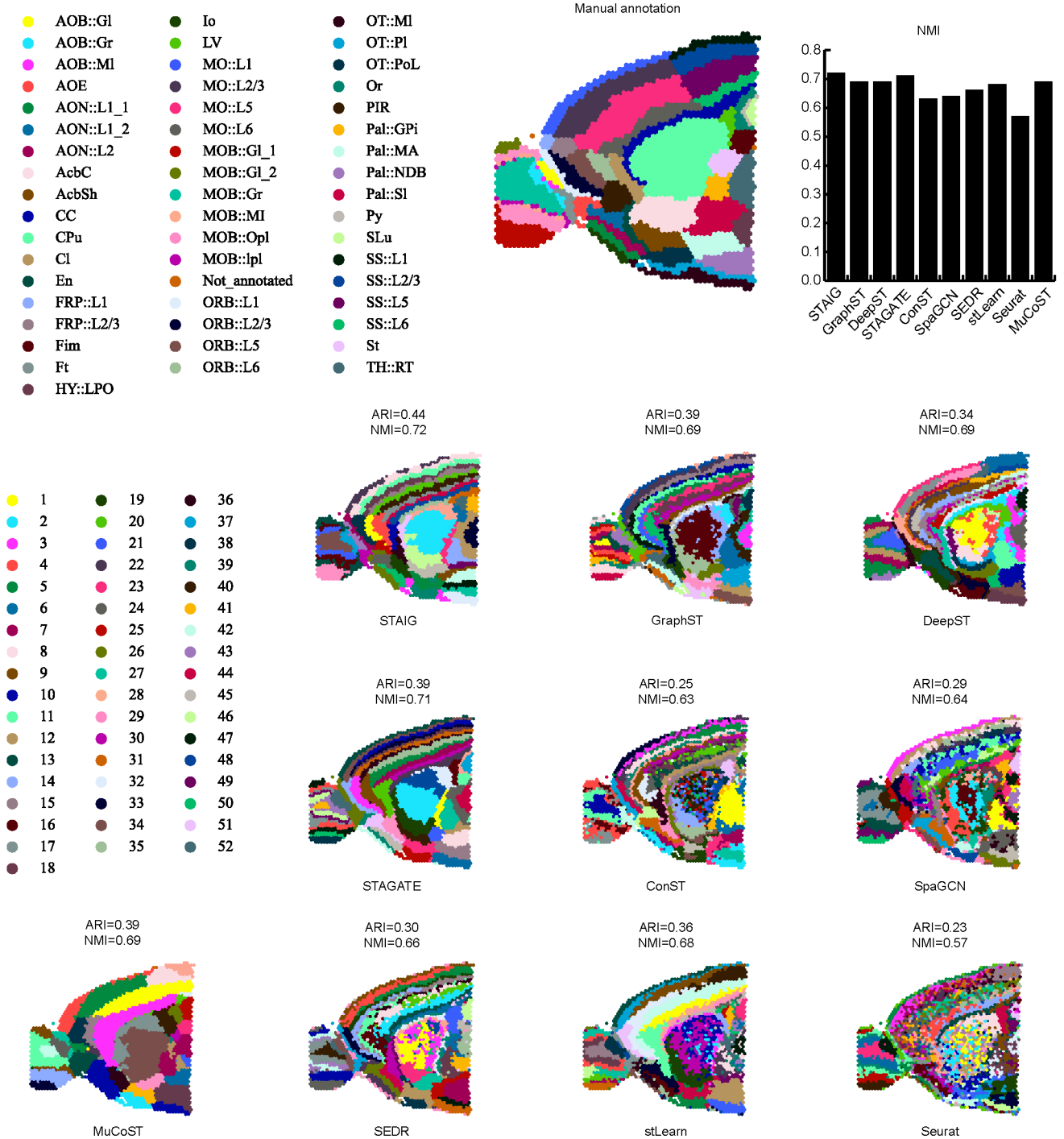


**Fig. S2** H&E stained image, manual annotations, clustering result with ARI, NMI and UMAP visualization by STAIG and baseline methods on DLPFC slices #151671, #151672, #151673, #151674, #151675, and #151676.

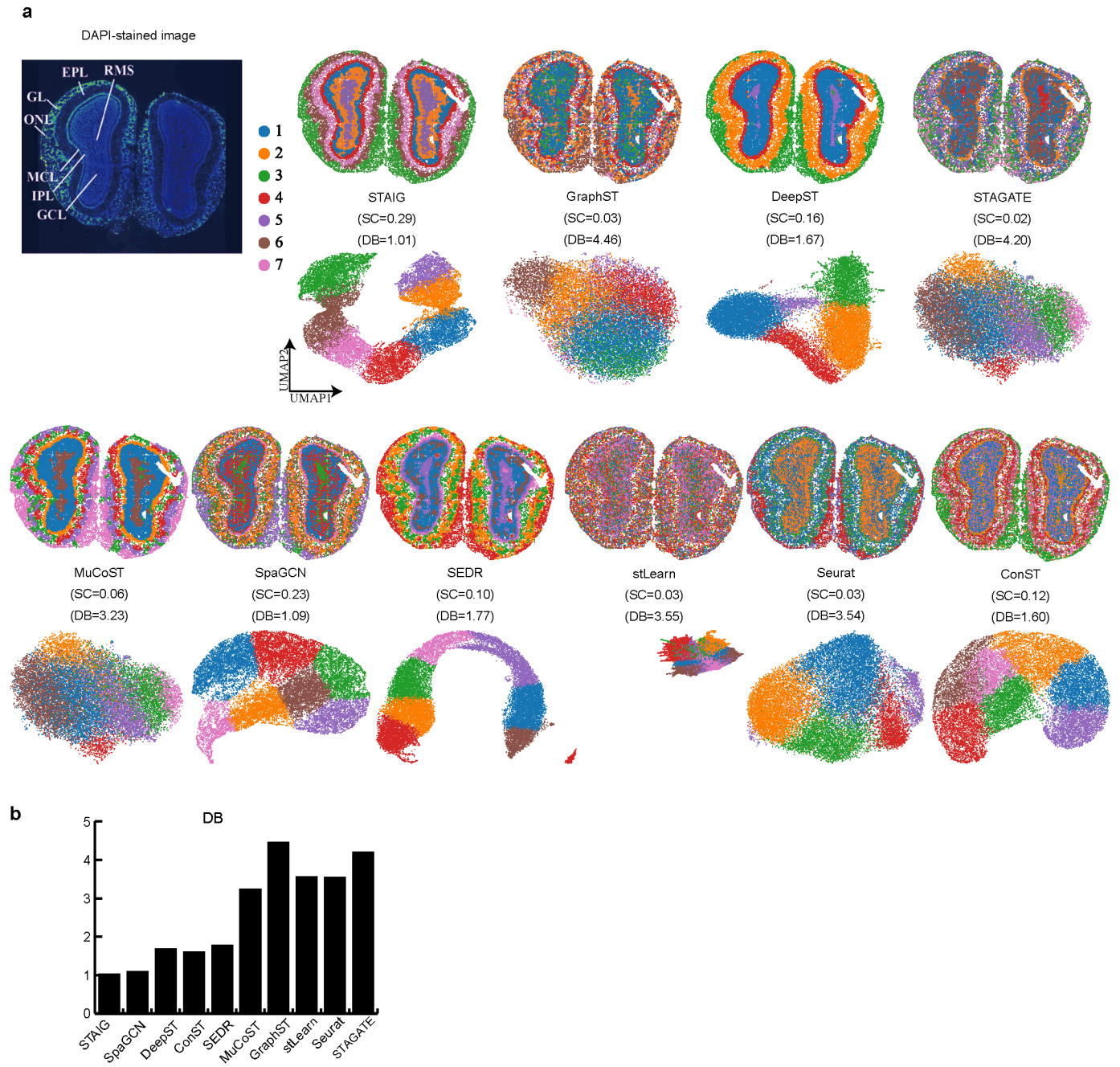


**Fig. S3 H&E stained image and clustering results with SC and DB by STAIG and baseline methods on mouse brain posterior tissue.**



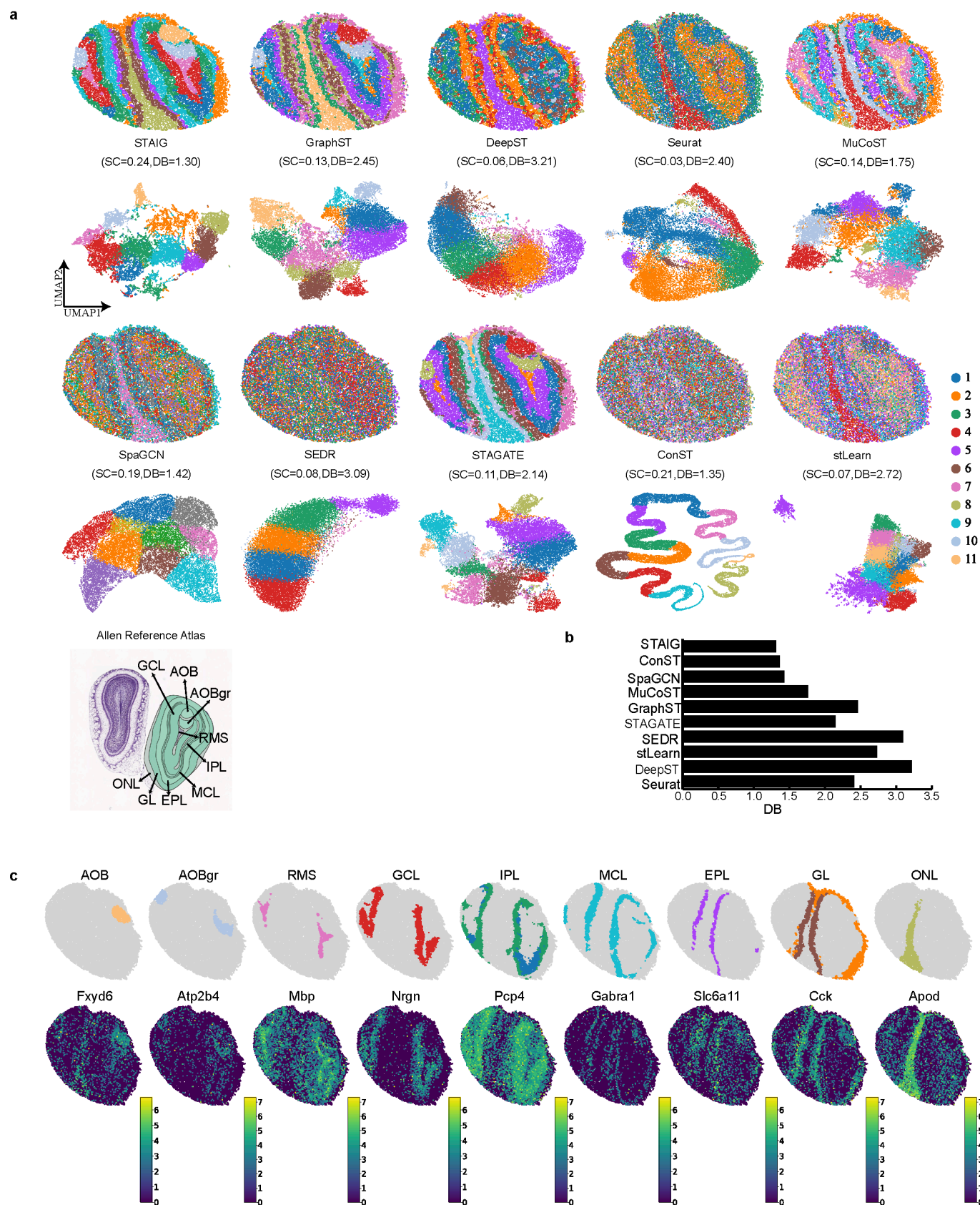


**Fig. S4 Manual annotations and clustering results with ARI and NMI by STAIG and baseline methods on mouse brain anterior tissue, including a bar chart of NMI results. The x-axis represents methods and y-axis represents NMI.**



**Fig. S5 supplementary figures on Stereo-seq mouse olfactory bulb tissue.**

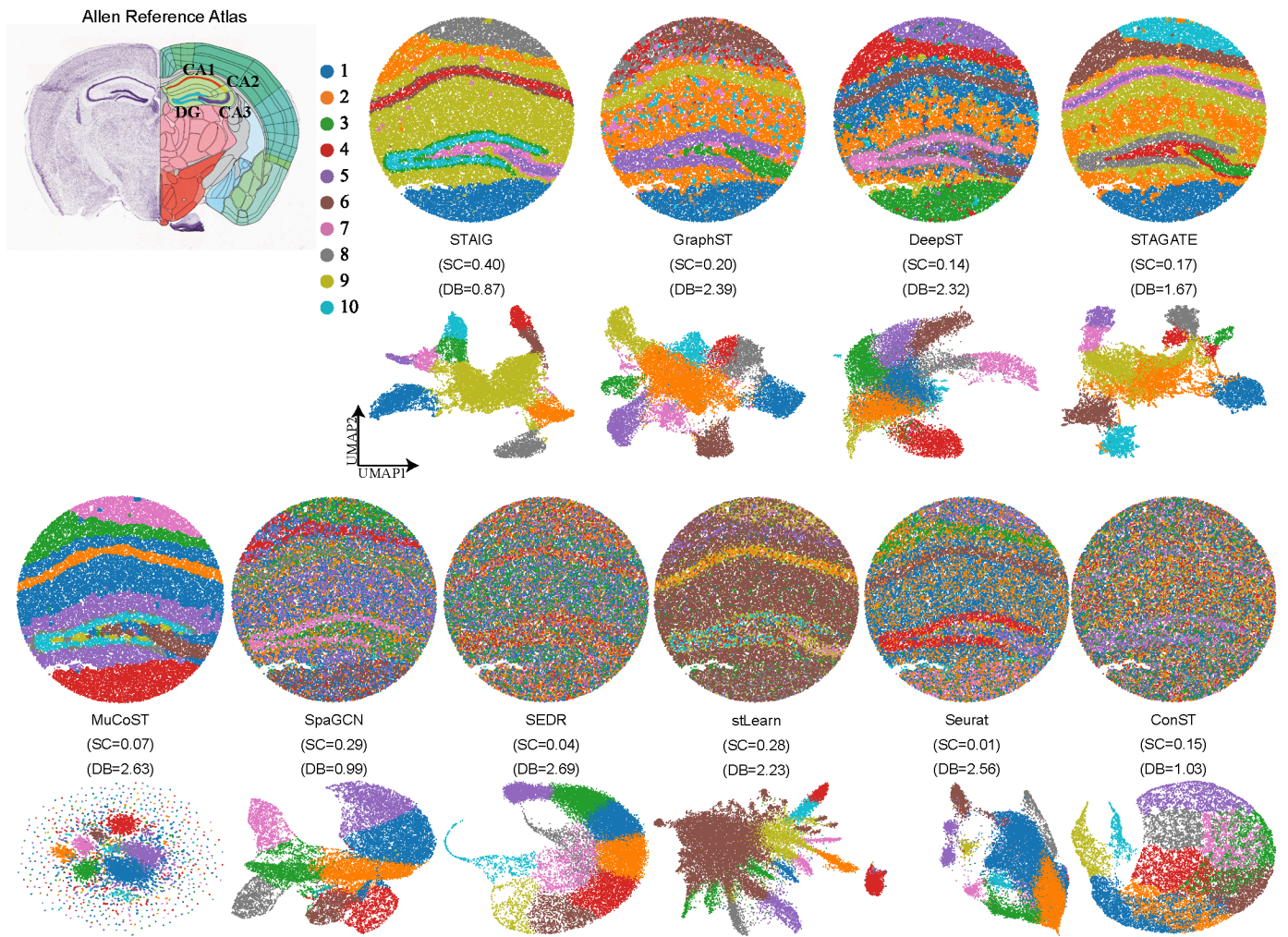
(a) DAPI-stained image with annotated laminar organization and clustering results with Silhouette Coefficient (SC) and Davies-Bouldin Index (DB) by STAIG and baseline methods on Stereo-seq mouse olfactory bulb tissue. (b) A bar chart of DB results. The x-axis represents methods and the y-axis represents DB.



**Fig. S6 supplementary figures on Slide-seqV2 mouse olfactory bulb tissue.**

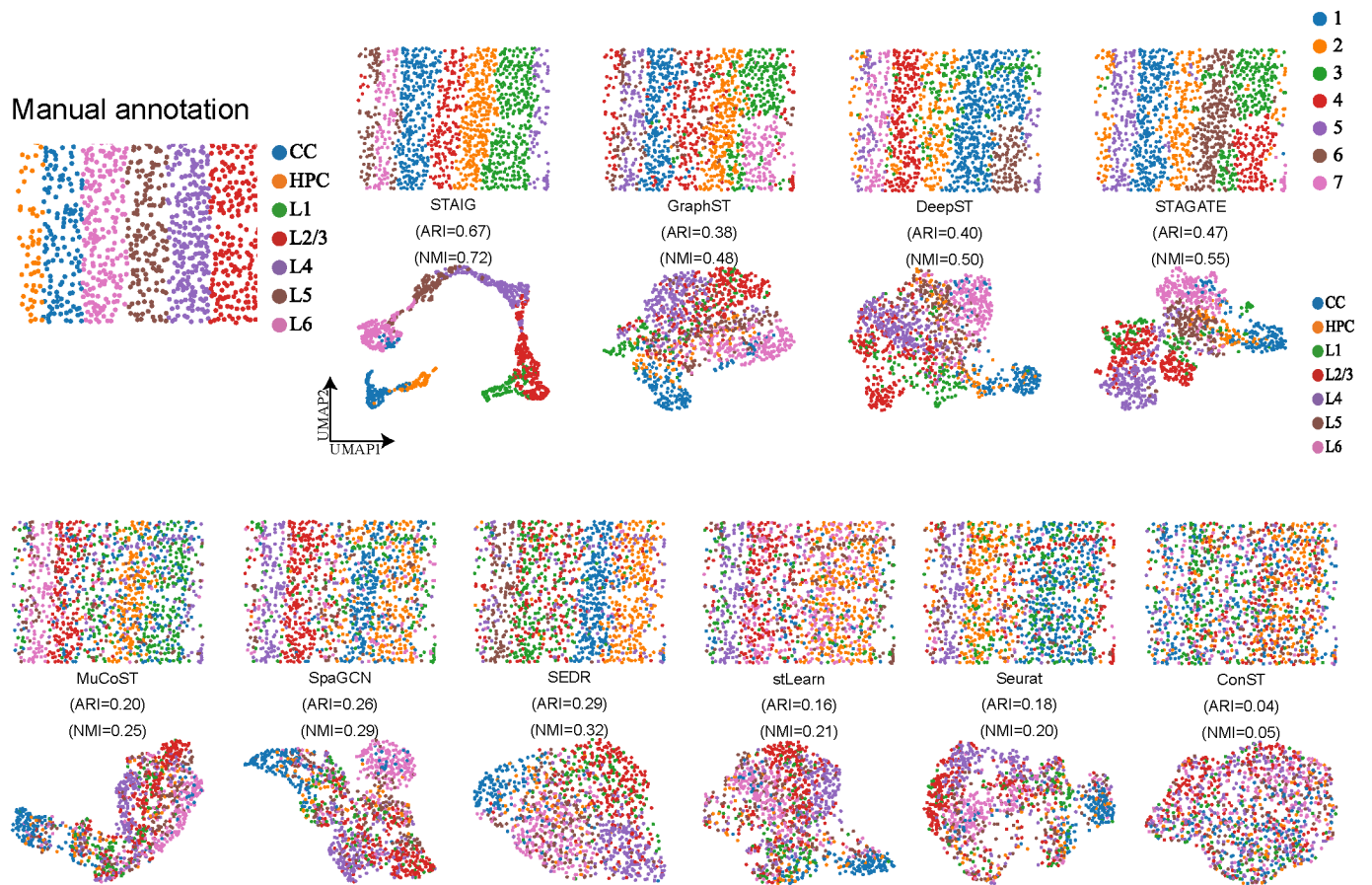
(a) Annotations from the Allen Reference Atlas and clustering results with SC and DB by STAIG and baseline methods on Slide-seqV2 mouse olfactory bulb tissue. (b) A bar chart of DB results. The y-axis represents methods, and the x-axis represents DB. (c) Visualization of spatial domains identified by STAIG and the corresponding marker gene expressions on Slide-seqV2 mouse olfactory bulb tissue.



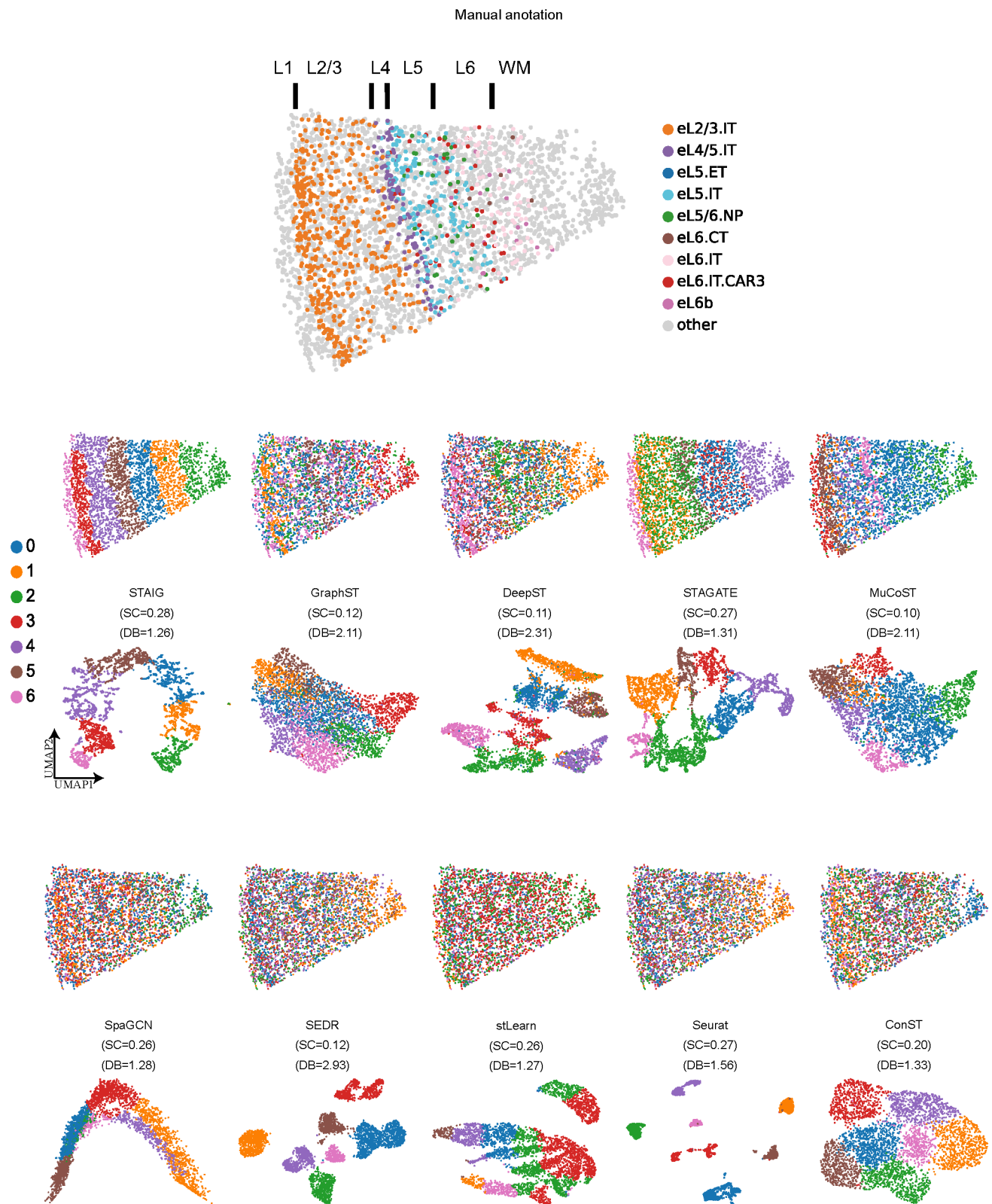


**Fig. S7** Annotations from the Allen Reference Atlas of Cornu Ammonis (CA) and Dentate Gyrus (DG) domains and clustering results with SC and DB by STAIG and baselines methods on Slide-seqV2 mouse hippocampus tissue.

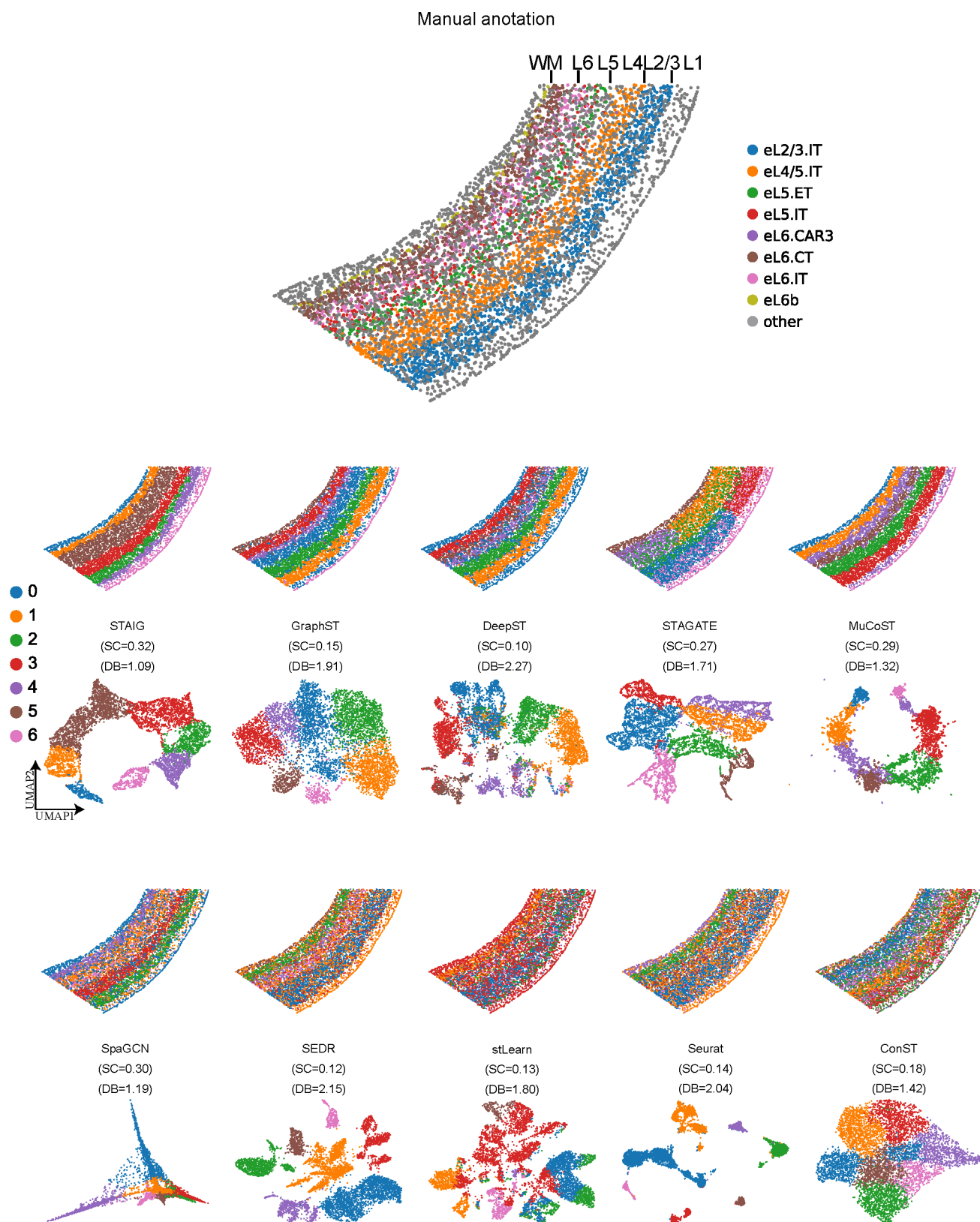




**Fig. S8 Manual annotations and clustering results with ARI and NMI by STAIG and baseline methods on STARmap mouse visual cortex dataset.**

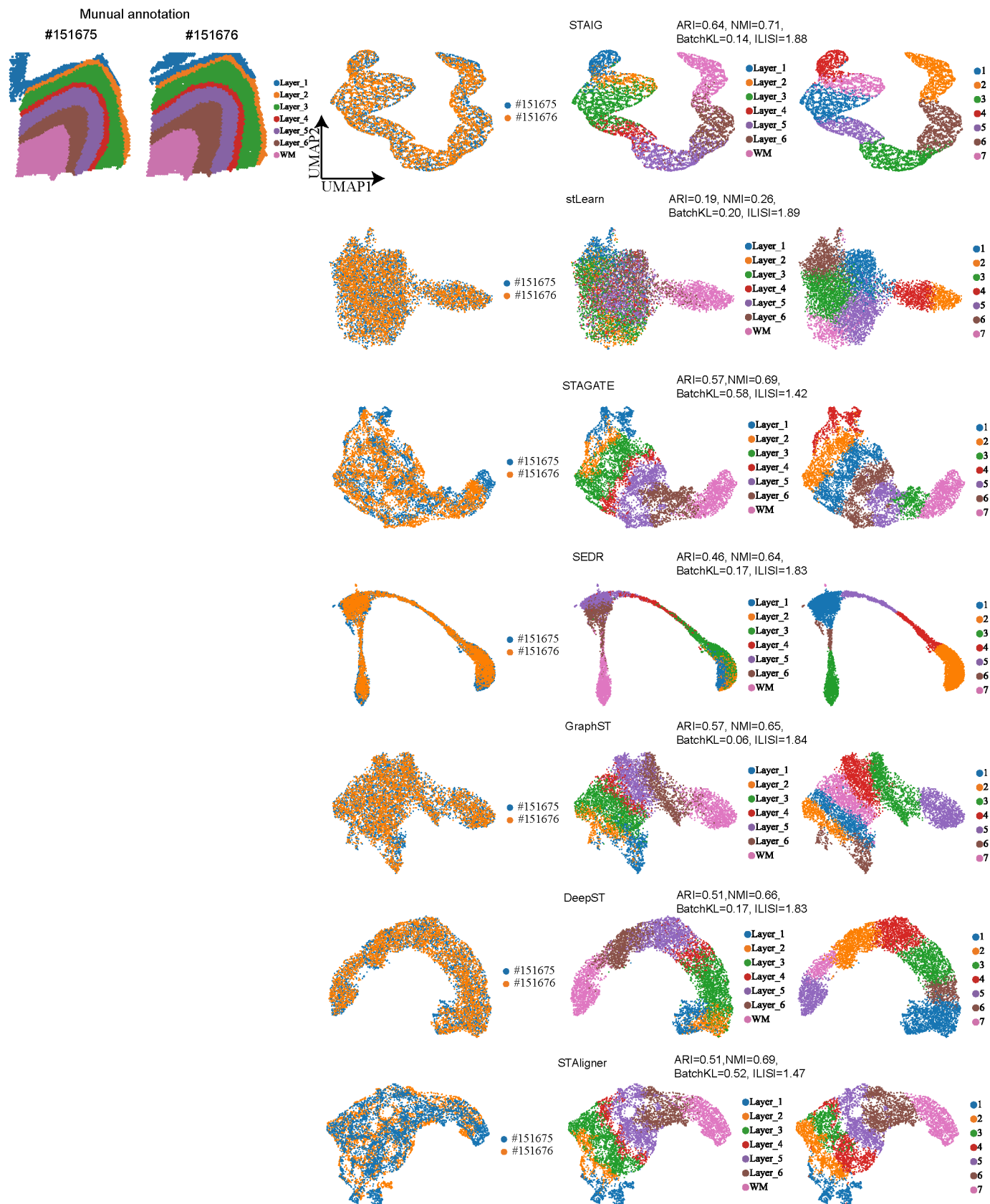


**Fig. S9 Manual annotations and clustering results with SC and DB by STAIG and baseline methods on the MERFISH human middle temporal gyrus dataset.**

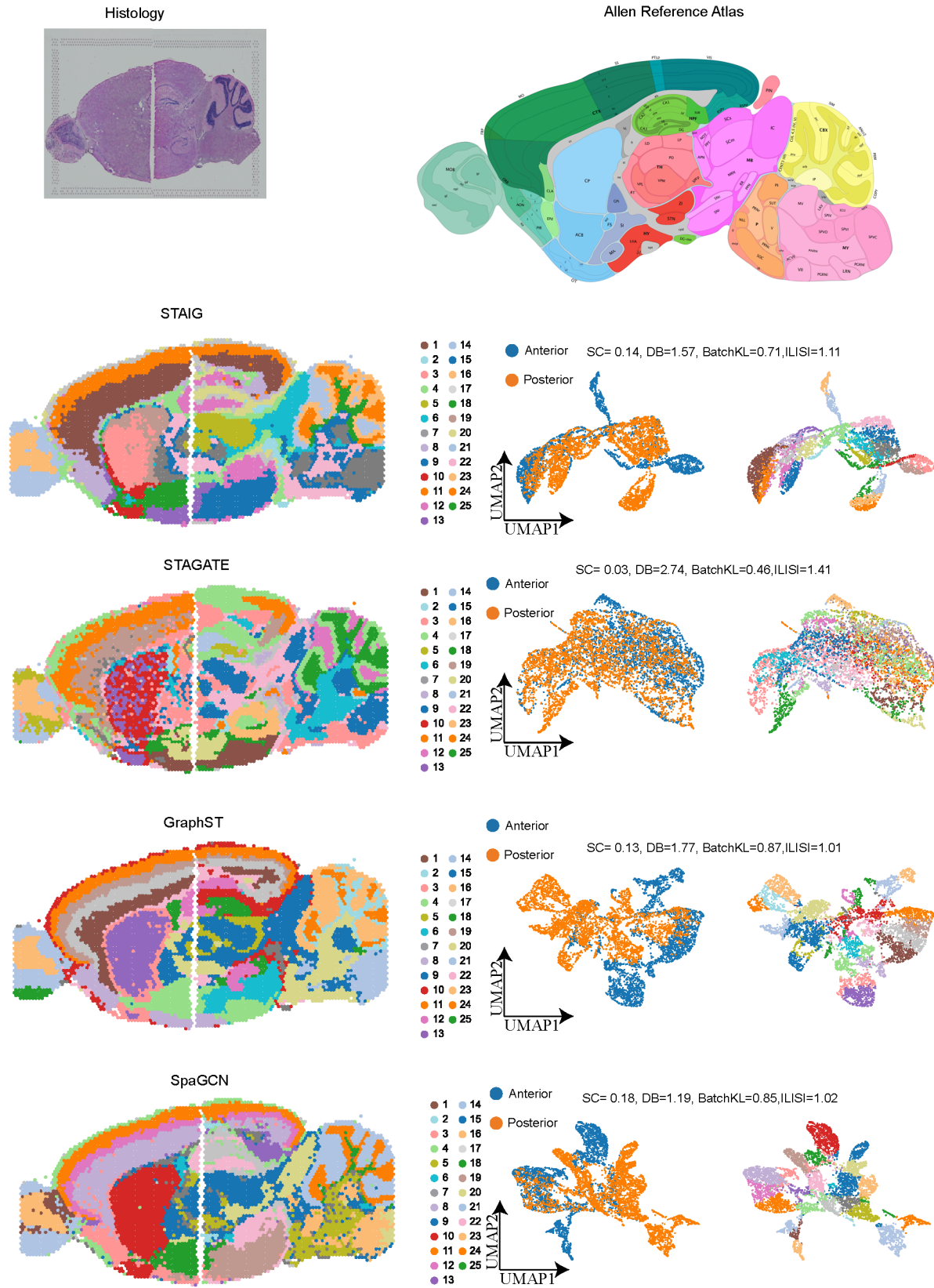


**Fig. S10** Manual annotations and clustering results with SC and DB by STAIG and baseline methods on the MERFISH mouse visual cortex dataset.

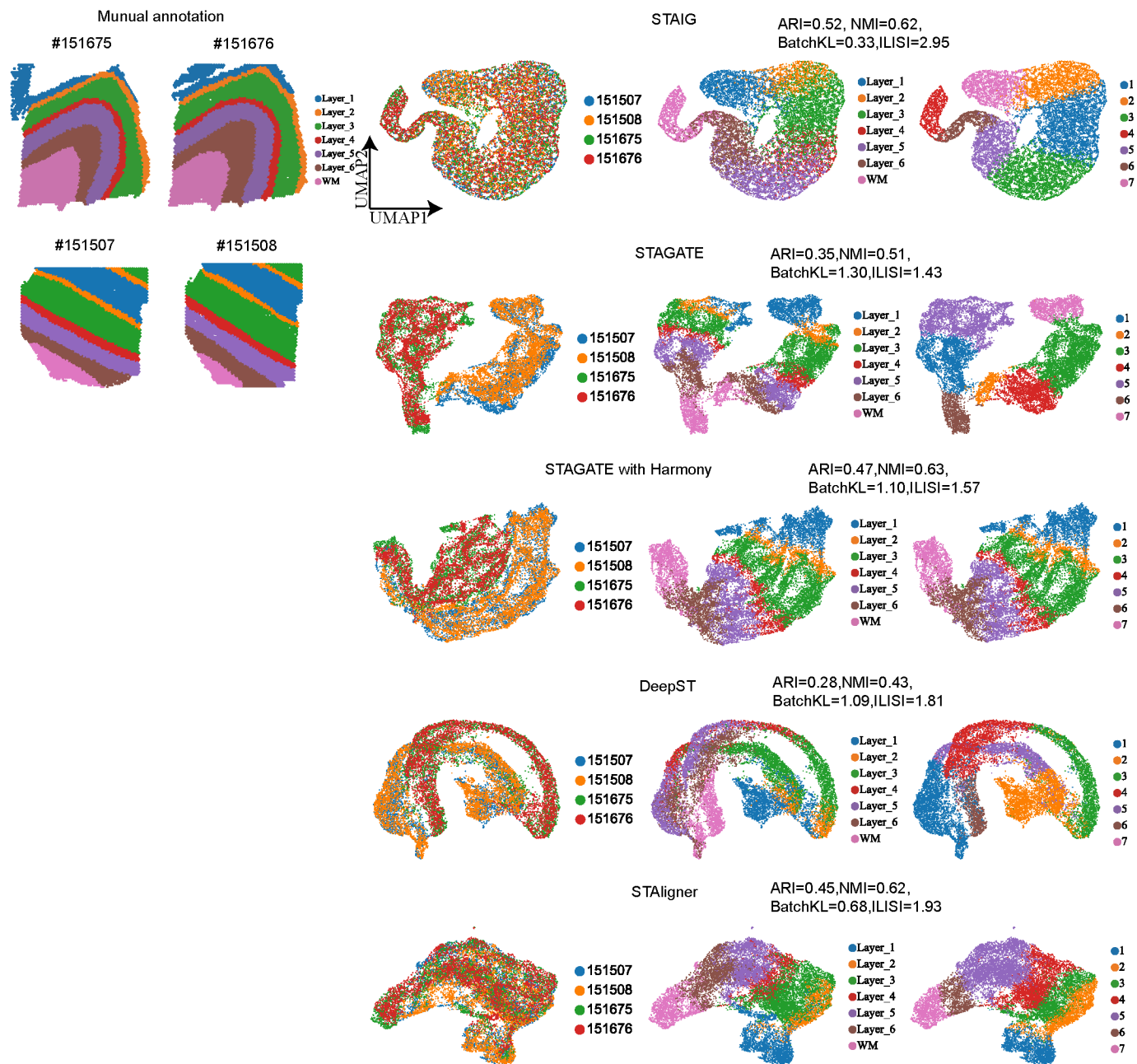




**Fig. S11 Vertical integration of STAIG and compared methods on DLPCF slices from the same source (#151675, #151676). Includes manual annotations, UMAP, ARI, NMI, BatchKL, and ILISI metrics.**

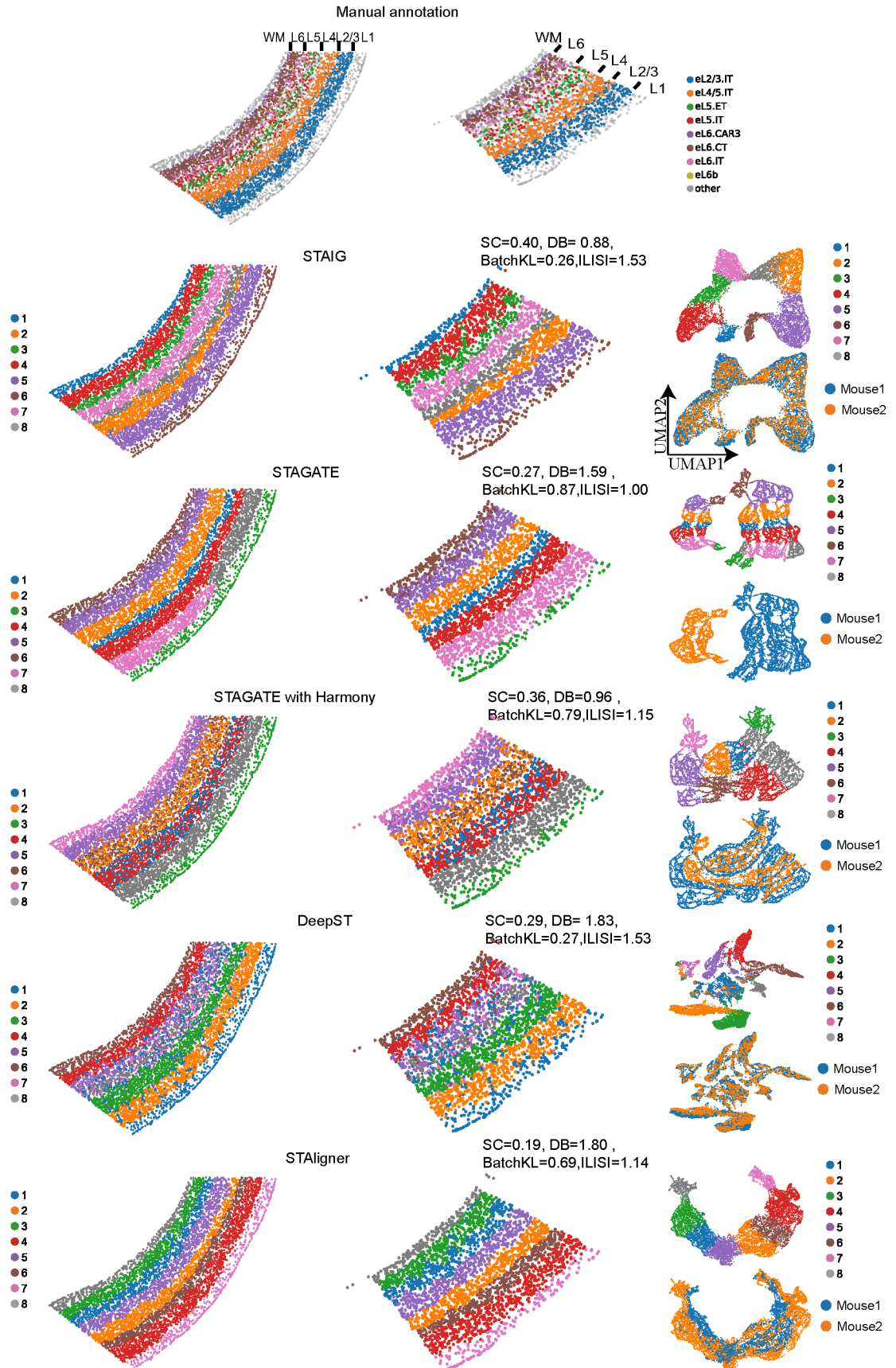


**Fig. S12** Horizontal integration of STAIG and compared methods on mouse anterior and posterior brain tissue slices. Includes H&E stained images, annotations from the Allen Reference Atlas, integrated results, UMAP, SC, DB, BatchKL and ILISI metrics.



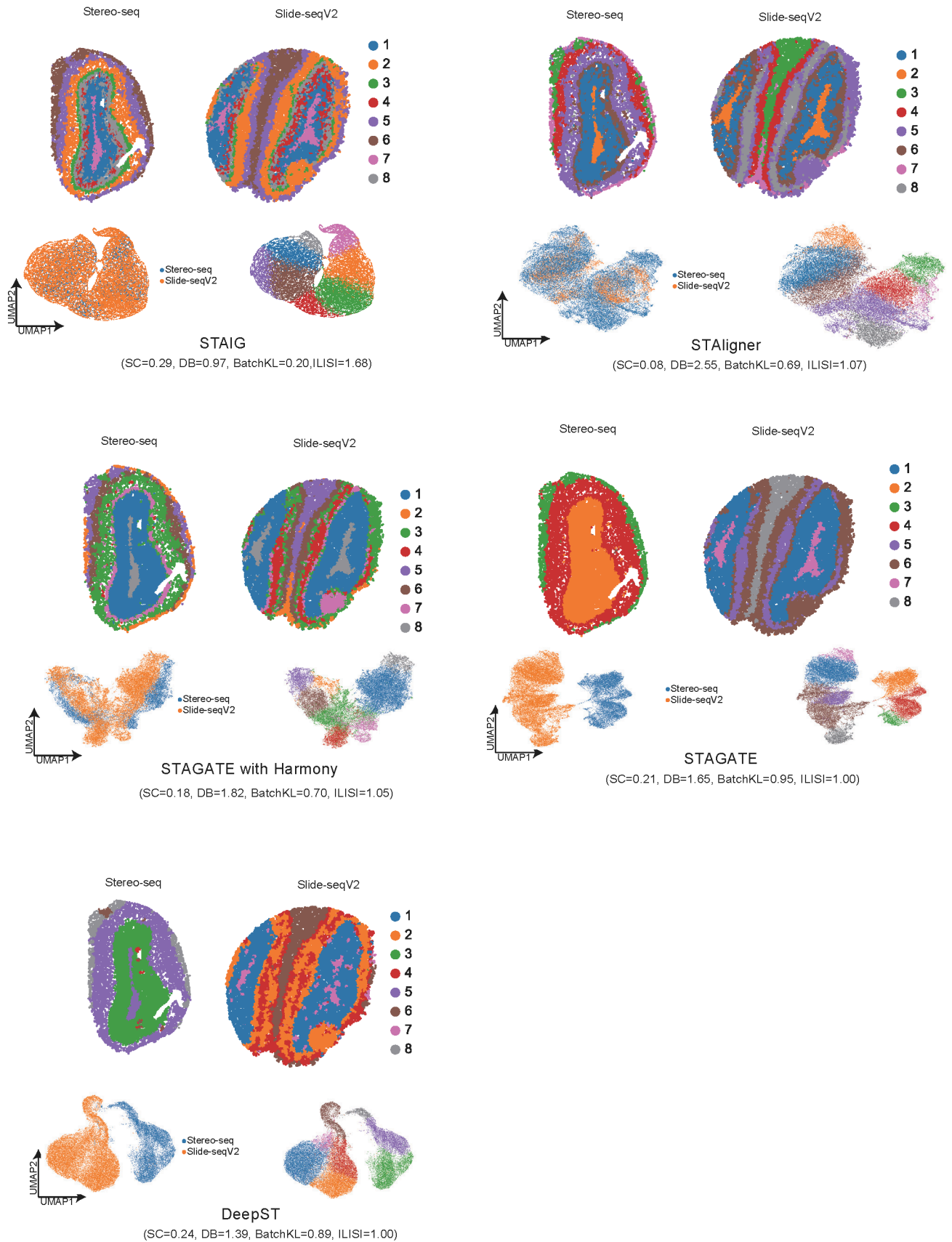
**Fig. S13** Integration of STAIG and compared methods on DLPFC slices from the different source (DLPFC #151507 with #151508 and #151675 and #151676). Includes manual annotations, UMAP, ARI, NMI, BatchKL, and ILISI metrics.





**Fig. S14 Integration of STAIG and compared methods on MERFISH mouse visual cortex datasets from the different mouse. Includes manual annotations, UMAP, SC, DB, BatchKL, and ILISI metrics.**

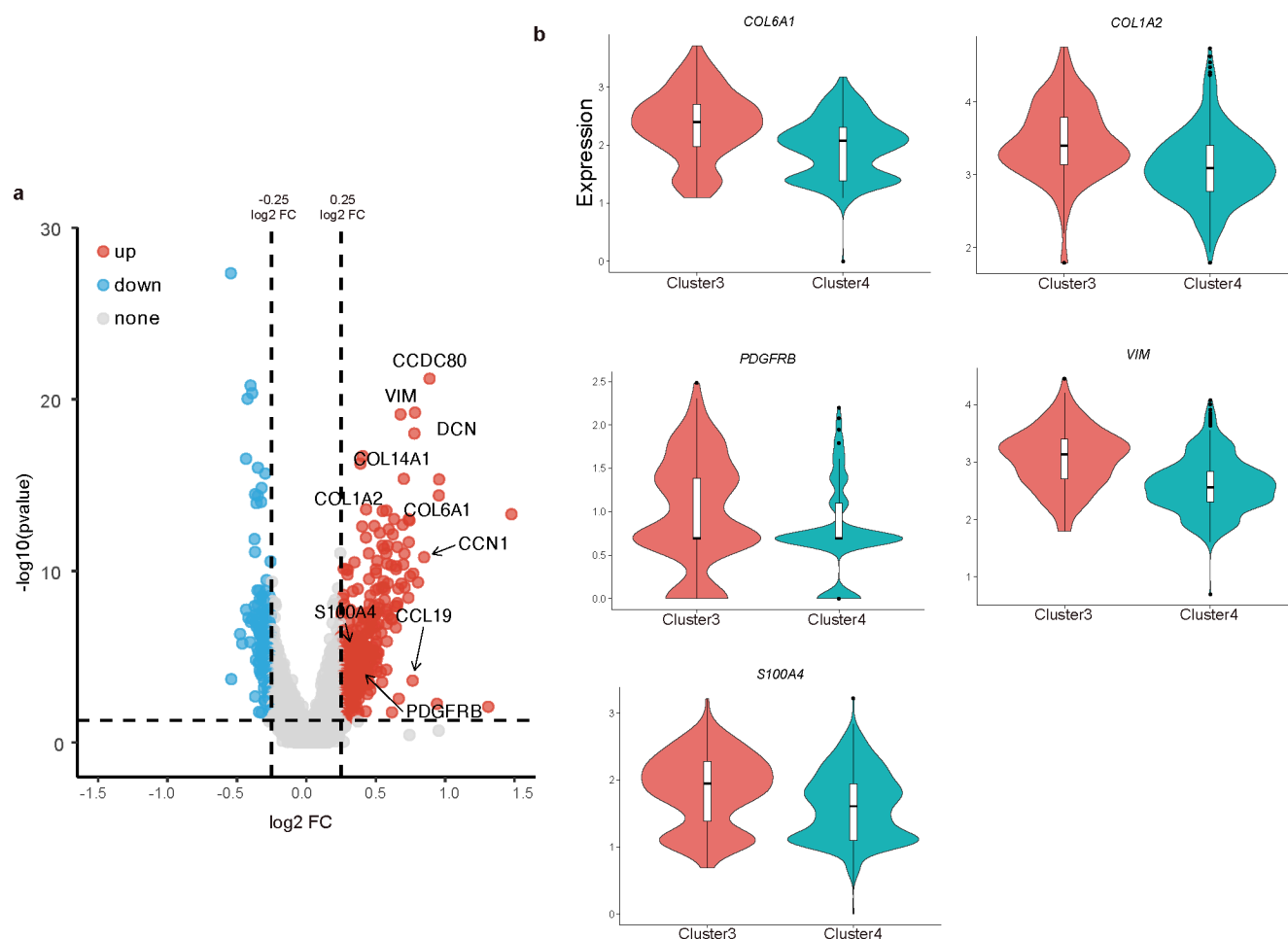




**Fig. S15** Cross platform integration of STAIG and compared methods on mouse olfactory bulb datasets (Stereo-seq and Slide-seqV2). Includes clustering results, UMAP, SC, DB, BatchKL, and ILISI metrics.



**Fig. S16 Manual annotation and clustering results with ARI and NMI by STAIG and baseline methods on human breast cancer dataset.**



**Fig. S17 Differential Gene Expression (DGE) and Gene Ontology (GO) analysis of Clusters 3 and 4 on human breast cancer dataset.**

(a) DGE analysis of Cluster 3 vs Cluster 4. Genes significantly upregulated in Cluster 3 are indicated in red, while significantly downregulated genes are shown in blue. Each point represents a gene, the vertical axis represents the  $-\log_{10}$  of the p-value and the horizontal axis represents the  $\log_2$  Fold Change ( $\log_2$  FC). P-values were derived from two-sided Wilcoxon rank-sum test. The significance thresholds were set at  $|\log_2 \text{FC}| > 0.25$  and  $\text{p-value} < 0.05$ . (b) Violin plots of expression of CAF marker genes (*COL6A1*, *COL1A2*, *VIM*, *PDGFRB*, *S100A4*) in Cluster 3 (n = 166 spots) vs Cluster 4 (n = 382 spots). The vertical axis represents gene expression levels. Each violin represents the distribution of expression for a particular gene, with the width indicating frequency. The central white dot denotes the median expression level, the thick black bar within each violin represents the interquartile range (IQR; 25th to 75th percentile), and the thin black line (whiskers) extends to  $1.5 \times \text{IQR}$  from the 25th and 75th percentiles, capturing the data range excluding outliers. Data represent biological replicates (individual spots).

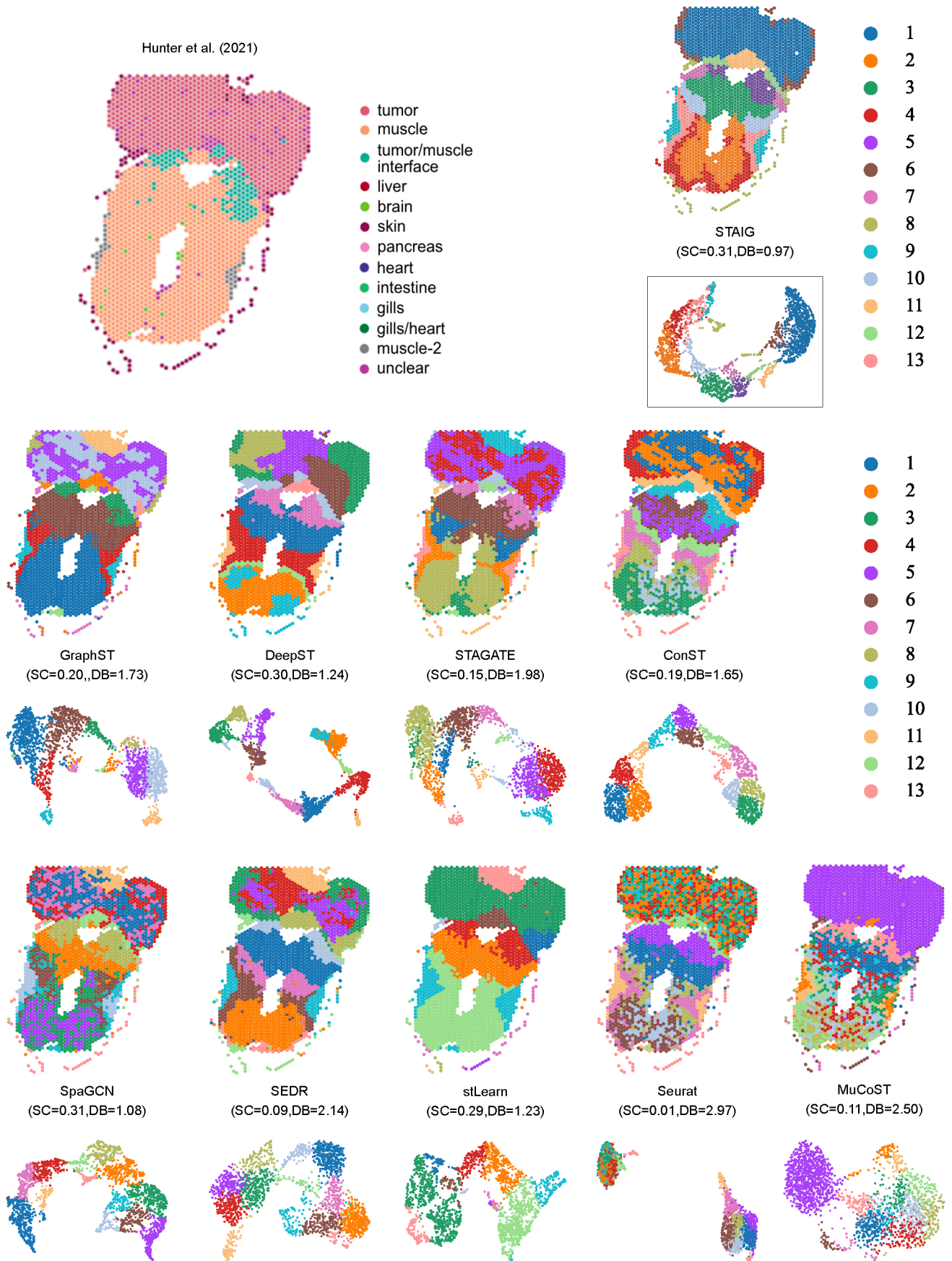


Fig. S18 Annotation and cluster results with SC and DB by STAIG and baseline methods on zebrafish melanoma on slice A.



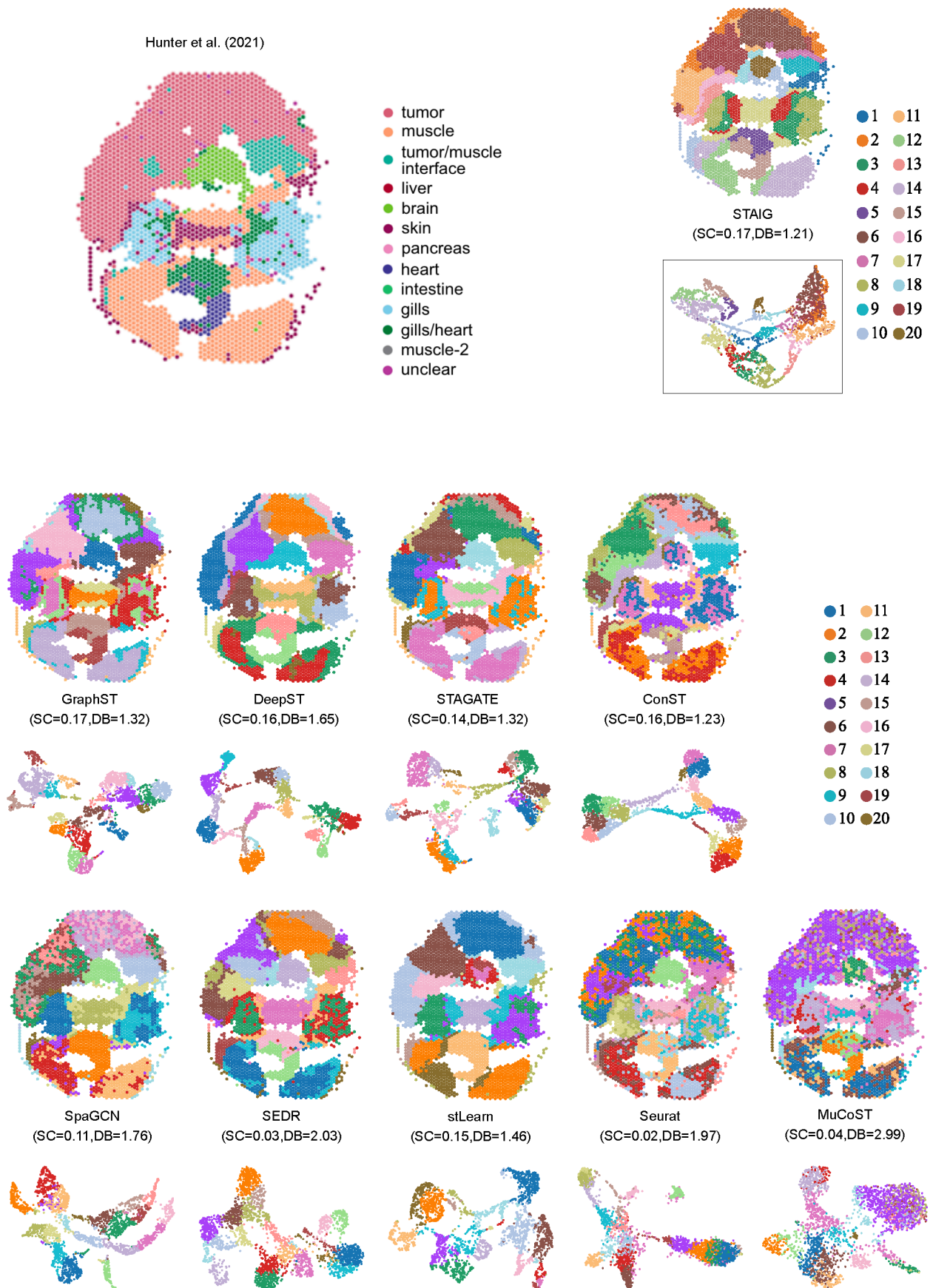
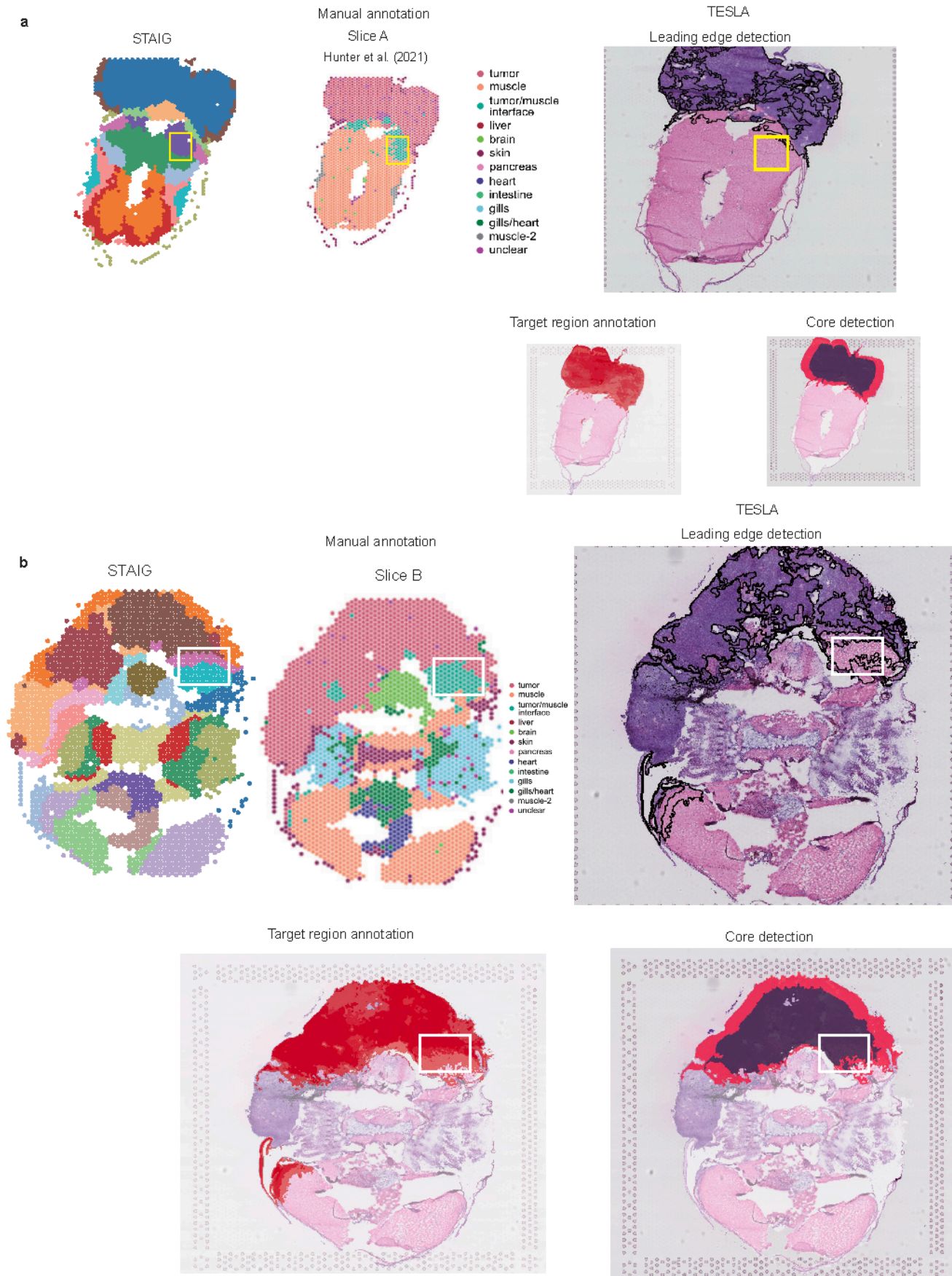


Fig. S19 Annotation and cluster results with SC and DB by STAIG and baseline methods on zebrafish melanoma on slice B.

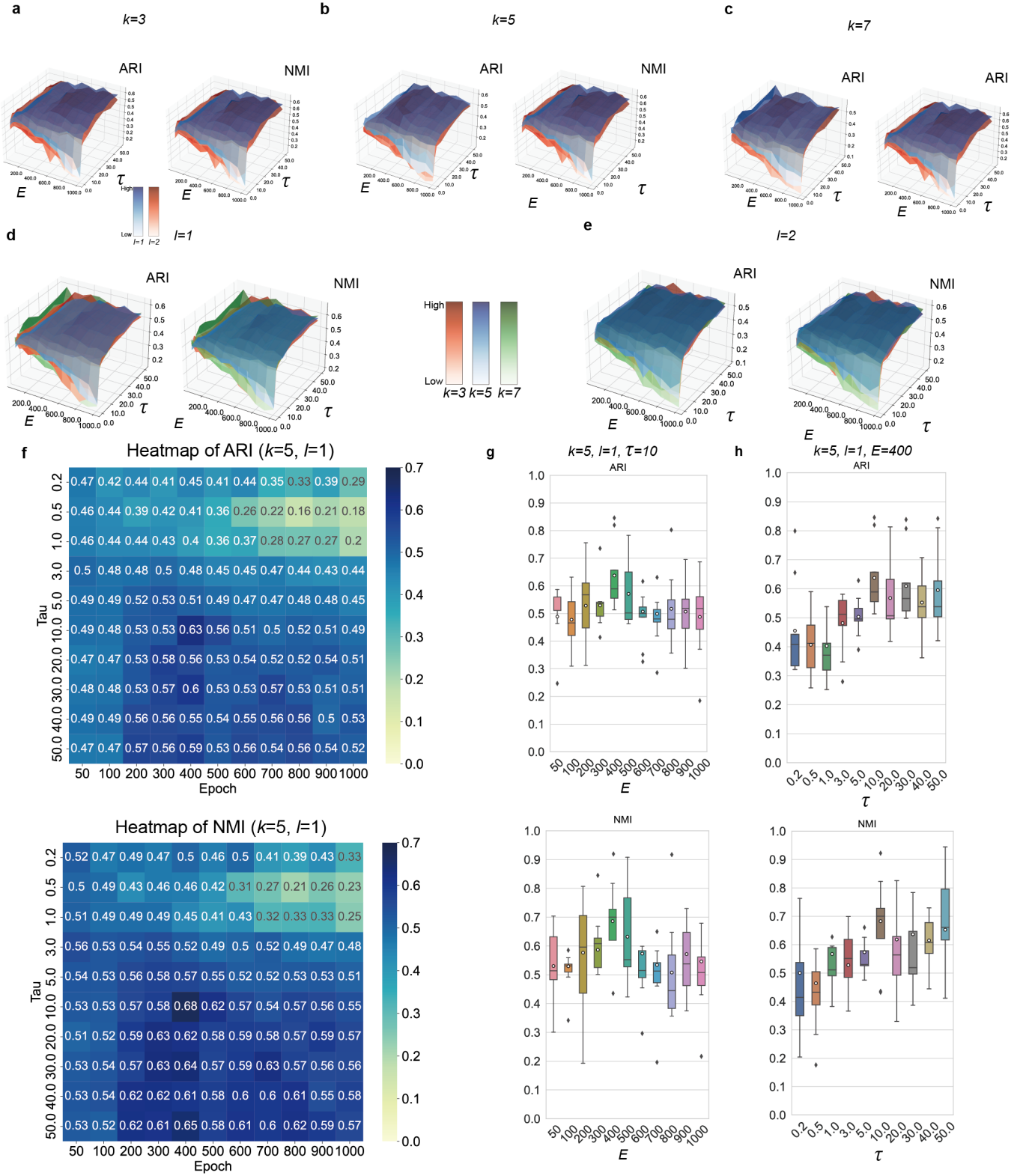


**Fig. S20 Comparison of STAIG and TESLA performance on zebrafish melanoma dataset.**

(a) STAIG clustering results, manual annotations, TESLA leading edge detection, core detection, and target region annotations on slice A. The yellow frame is part of interface. (b) STAIG clustering results, manual annotations, TESLA leading edge detection, core detection, and target region annotations on slice B. The white frame is part of interface.



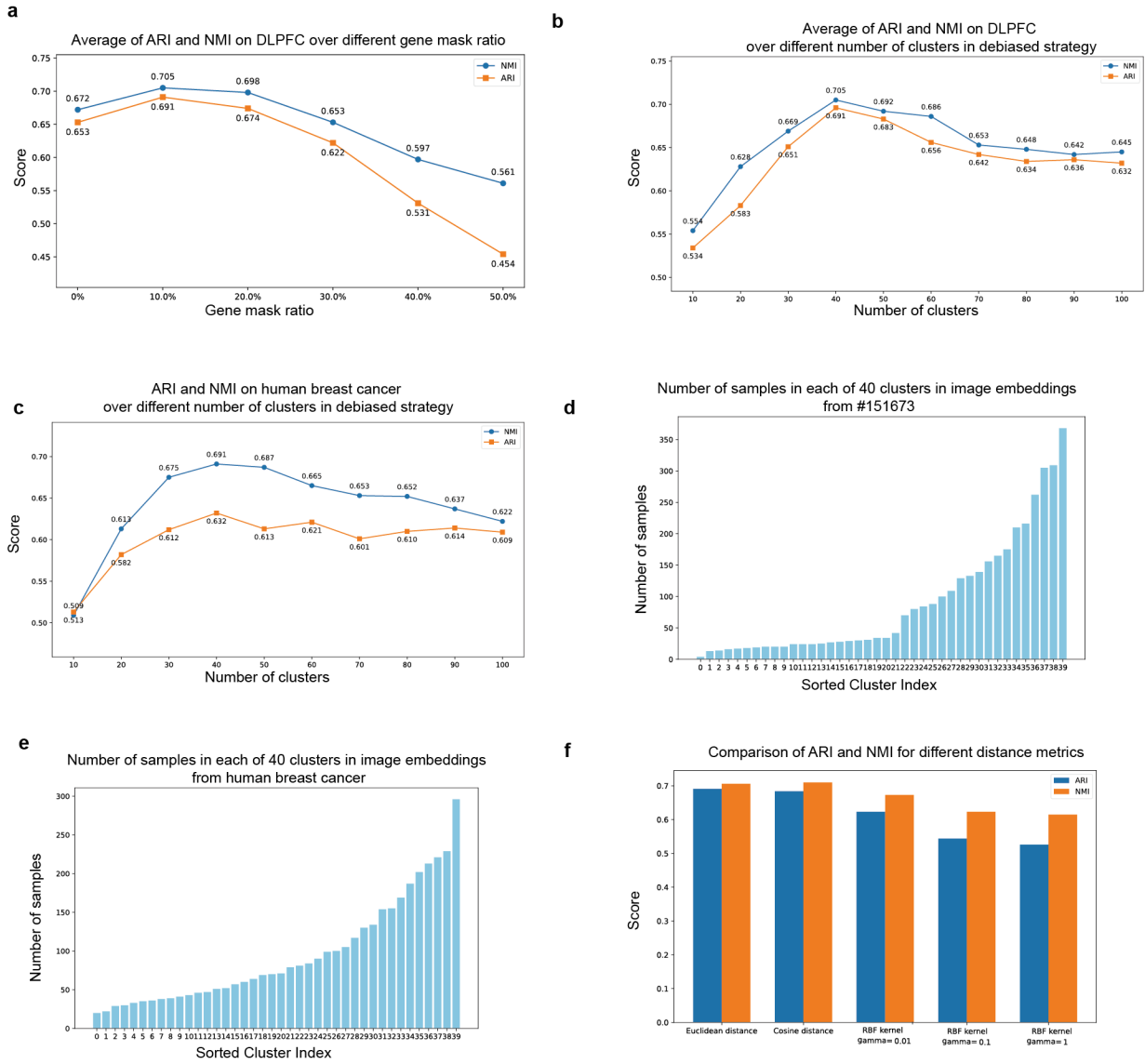




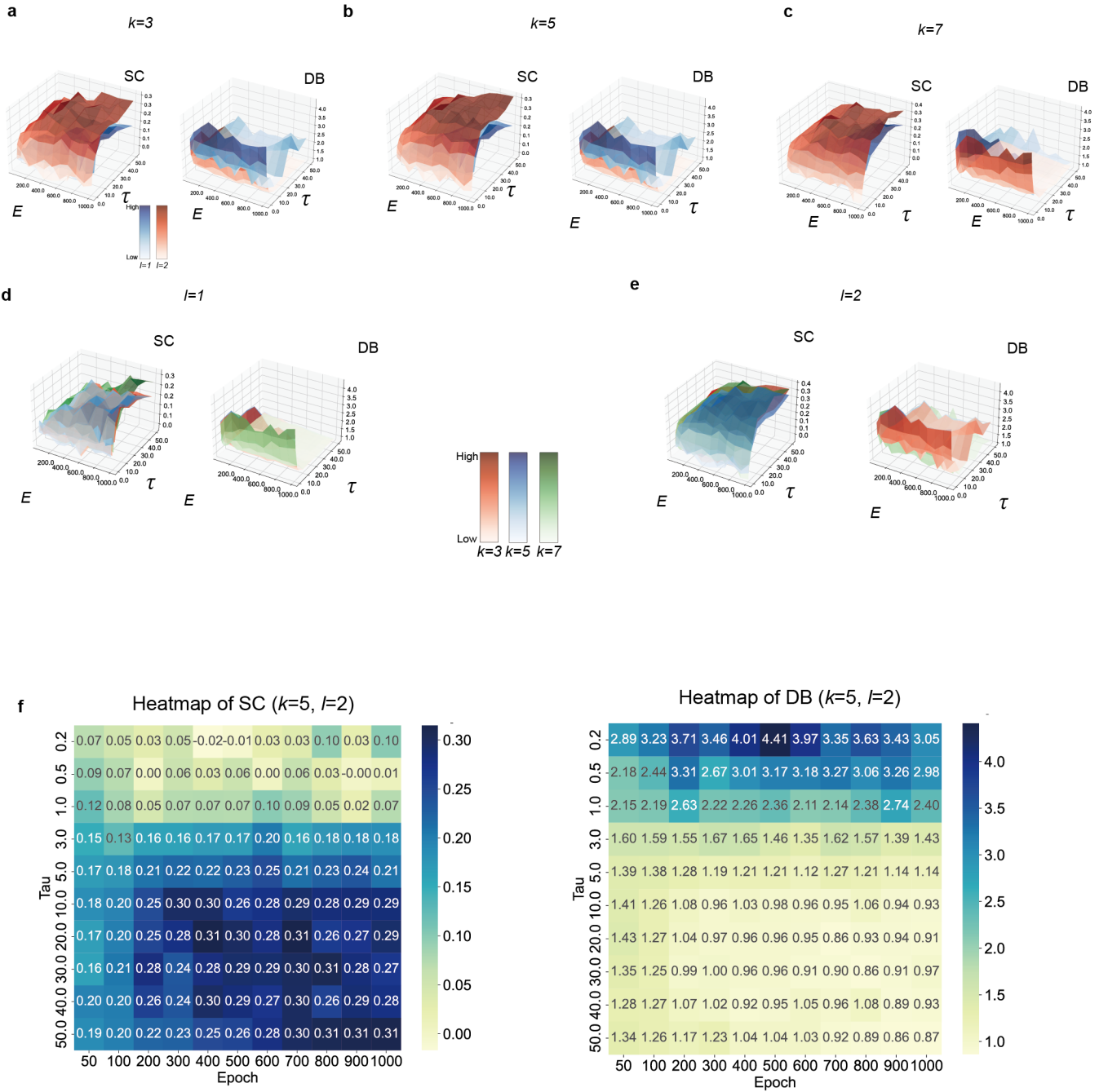
**Fig. S22 Main hyperparameter optimization on 10x Visium**

(a) 3D surface plots of average ARI and NMI on DLPFC with the neighboring node  $k$  of 3. The x-axis represents epochs  $E$ , the y-axis represents the temperature parameter  $\tau$ , and the z-axis represents ARI or NMI. (b) 3D surface plots of average ARI and NMI on DLPFC with  $k$  of 5. (c) 3D surface plots of average ARI and NMI on DLPFC with  $k$  of 7. (d) 3D surface plots of average ARI and NMI on DLPFC the with GCN layers number  $l$  of 1. (e) 3D surface plots of average ARI and NMI on DLPFC with  $l$  of 2. (f) Heatmap of average ARI and NMI on DLPFC with  $k$  of 5 and  $l$  of 1. The horizontal axis represents  $E$ , and the vertical axis represents  $\tau$ . (g) Boxplots of ARI and NMI by STAIG on the DLPFC dataset under various  $E$  with  $k$  of 5,  $l$  of 1, and  $\tau$  of 10. In the boxplot, the center line denotes the median, the white dot denotes the mean value, box limits denote the upper and lower quartiles, black dots denote individual slices, and whiskers denote the  $1.5 \times$  interquartile range. (h) Boxplots of ARI and NMI by STAIG on the DLPFC dataset under various  $\tau$  with  $k$  of 5,  $l$  of 1, and  $E$  of 400.





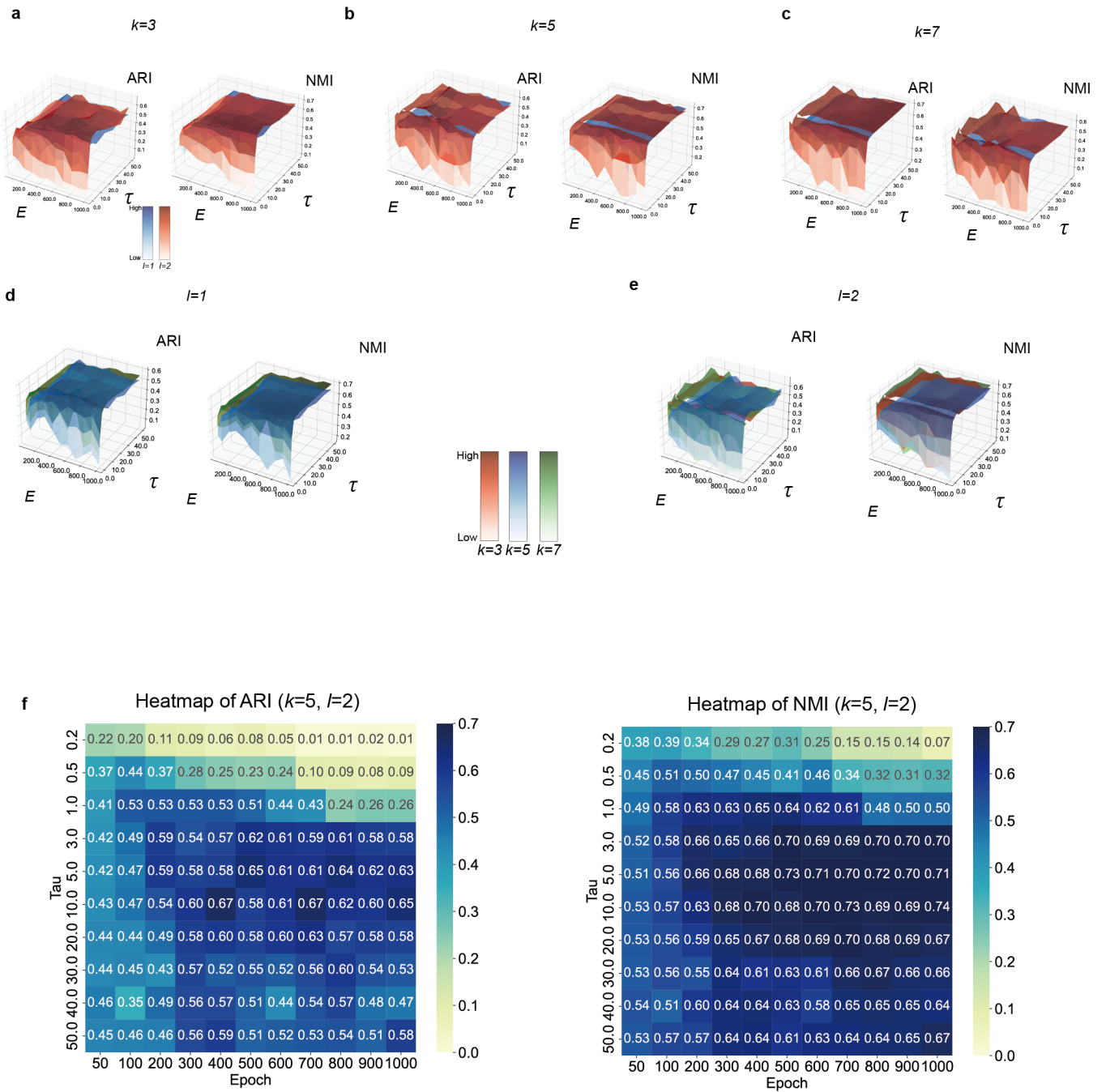
**Fig. S23 Analysis of masked feature ratios in graph augmentation on Visium and cluster number assessments in a debiased strategy.** (a) Line graphs showing average ARI and NMI across different gene mask ratios. The x-axis represents gene mask ratio, and the y-axis represents scores. (b) Line graphs of average ARI and NMI for varying cluster numbers in the debiased strategy on the DLPFC dataset. The x-axis represents number of clusters, and the y-axis represents scores. (c) Line graphs of average ARI and NMI for varying cluster numbers in the debiased strategy on the human breast cancer dataset. The x-axis represents number of clusters, and the y-axis represents scores. (d) Ascending distribution graph of cluster counts for  $k$ -means on DLPFC image features under 40 categories using STAIG (#151673). The x-axis represents cluster index, and the y-axis represents number of samples in corresponding cluster. (e) Ascending distribution graph of cluster counts for  $k$ -means on human breast cancer dataset image features under 40 categories using STAIG. The x-axis represents cluster index, and the y-axis represents number of samples in corresponding cluster. (f) ARI and NMI under different distance metrics for the DLPFC dataset. The x-axis represents different distance metrics, and the y-axis represents scores.



**Fig. S24 Main hyperparameter optimization on Stereo-seq platform.**

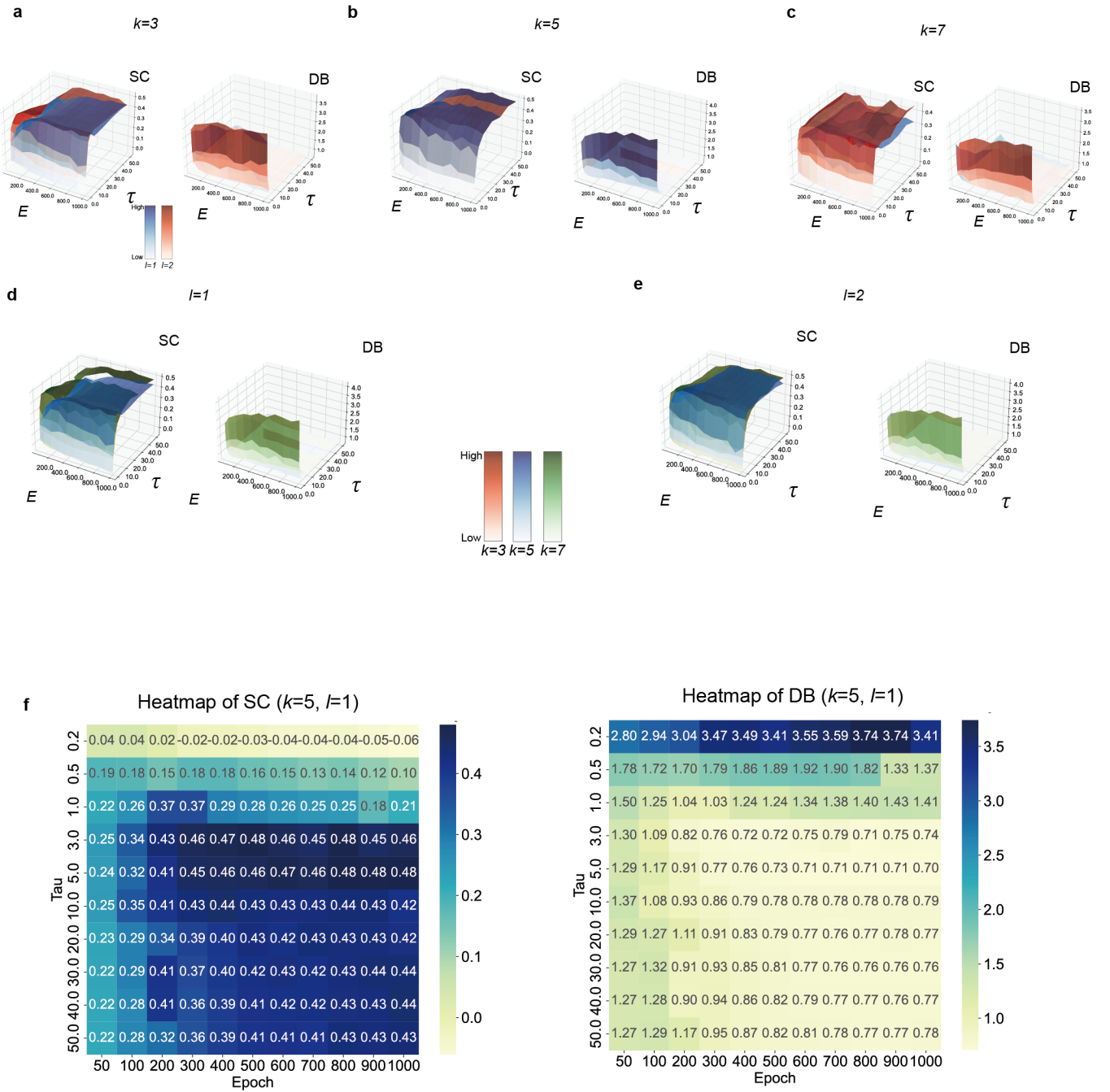
(a) 3D surface plots of SC and DB on Stereo-seq platform mouse olfactory bulb dataset with the neighboring node  $k$  of 3. The x-axis represents epochs  $E$ , the y-axis represents the temperature parameter  $\tau$ , and the z-axis represents SC or DB. (b) 3D surface plots of SC and DB on Stereo-seq platform mouse olfactory bulb dataset with  $k$  of 5. (c) 3D surface plots of SC and DB on Stereo-seq platform mouse olfactory bulb dataset with  $k$  of 7. (d) 3D surface plots of SC and DB on Stereo-seq platform mouse olfactory bulb dataset with GCN layers number  $l$  of 1. (e) 3D surface plots of SC and DB on Stereo-seq platform mouse olfactory bulb dataset with  $l$  of 2. (f) Heatmap of SC and DB on Stereo-seq platform mouse olfactory bulb dataset with  $k$  of 5 and  $l$  of 2. The horizontal axis represents  $E$ , and the vertical axis represents  $\tau$ .





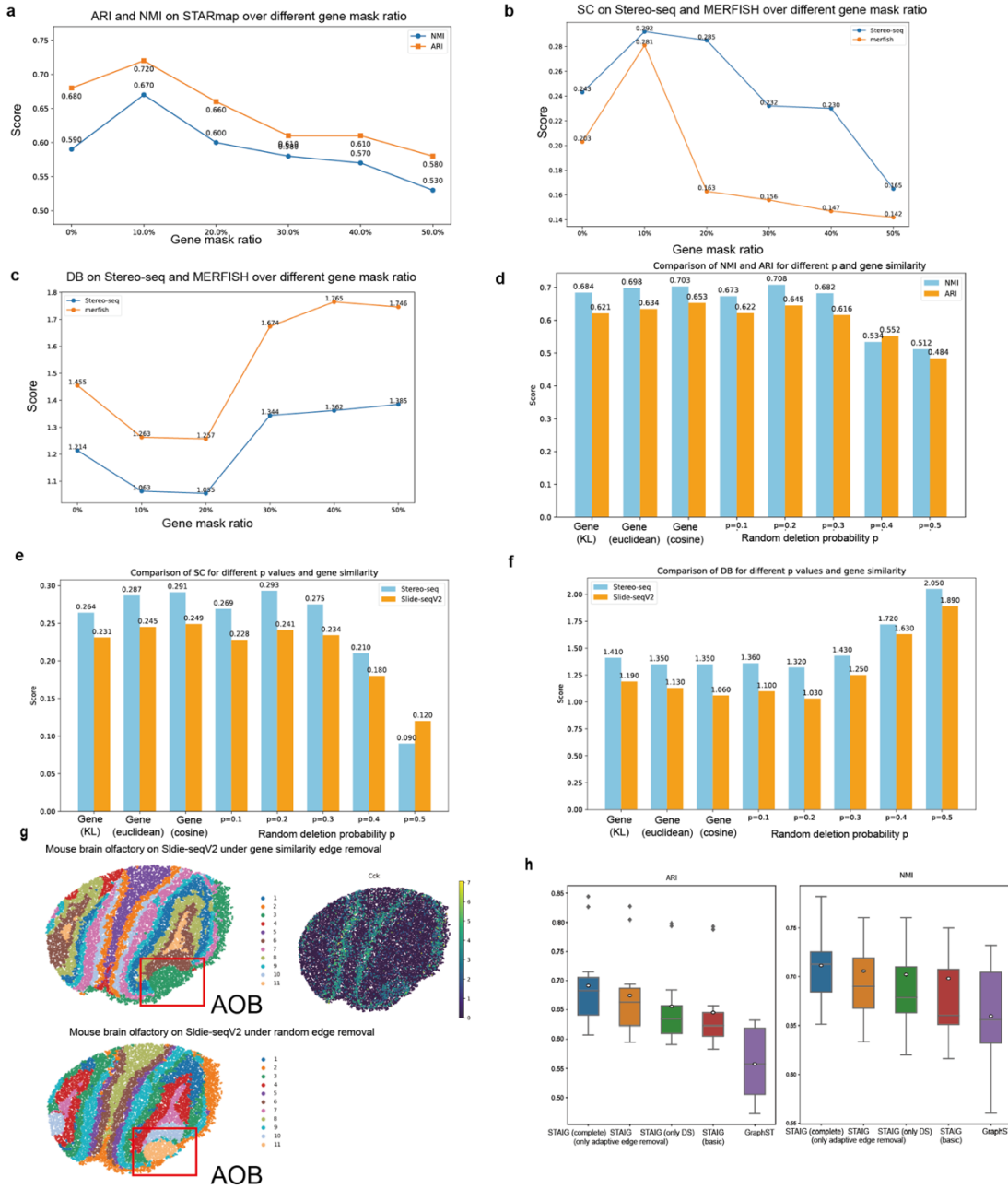
**Fig. S25 Main hyperparameter optimization on STARmap platform.**

(a) 3D surface plots of ARI and NMI on STARmap mouse visual cortex dataset with the neighboring node  $k$  of 3. The x-axis represents epochs  $E$ , the y-axis represents the temperature parameter  $\tau$ , and the z-axis represents ARI or NMI. (b) 3D surface plots of ARI and NMI on STARmap mouse visual cortex dataset with  $k$  of 5. (c) 3D surface plots of ARI and NMI on STARmap mouse visual cortex dataset with  $k$  of 7. (d) 3D surface plots of ARI and NMI on STARmap mouse visual cortex dataset with GCN layers number  $l$  of 1. (e) 3D surface plots of ARI and NMI on STARmap mouse visual cortex dataset with  $l$  of 2. (f) Heatmap of ARI and NMI on Stereo-seq platform mouse olfactory bulb dataset with  $k$  of 5 and  $l$  of 2. The horizontal axis represents  $E$ , and the vertical axis represents  $\tau$ .



**Fig. S26 Main hyperparameter optimization on MERFISH platform.**

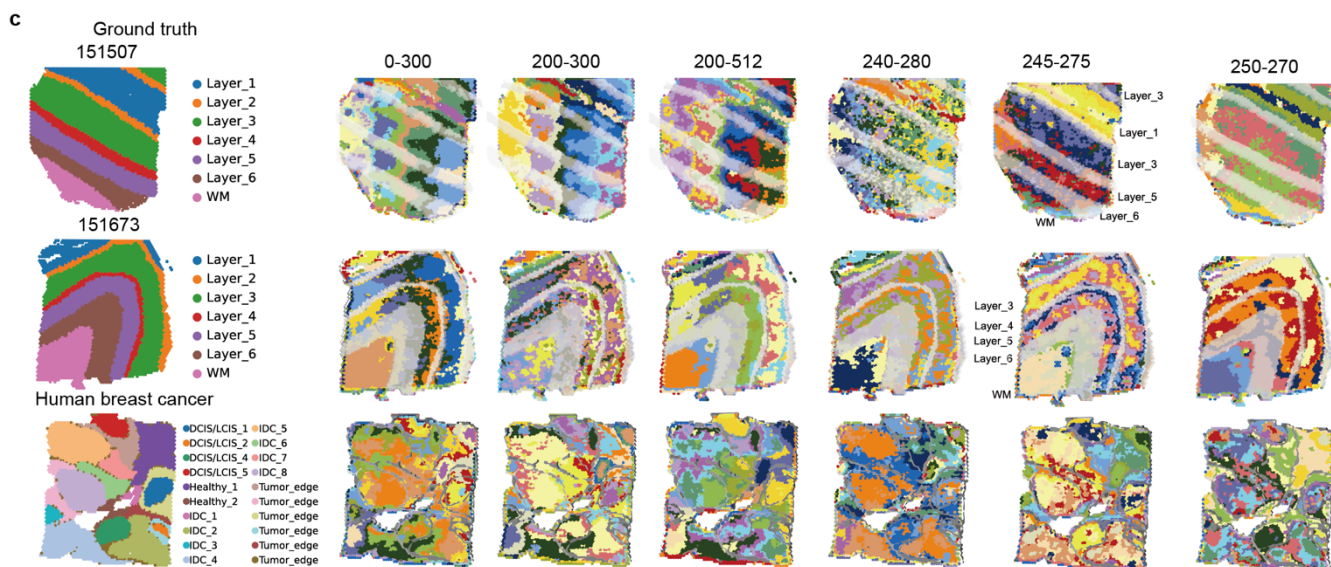
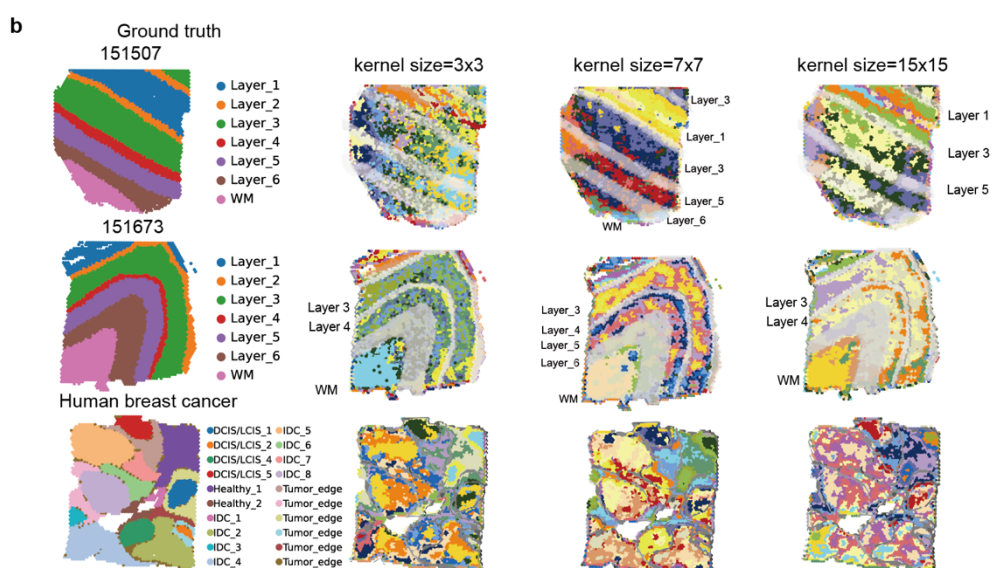
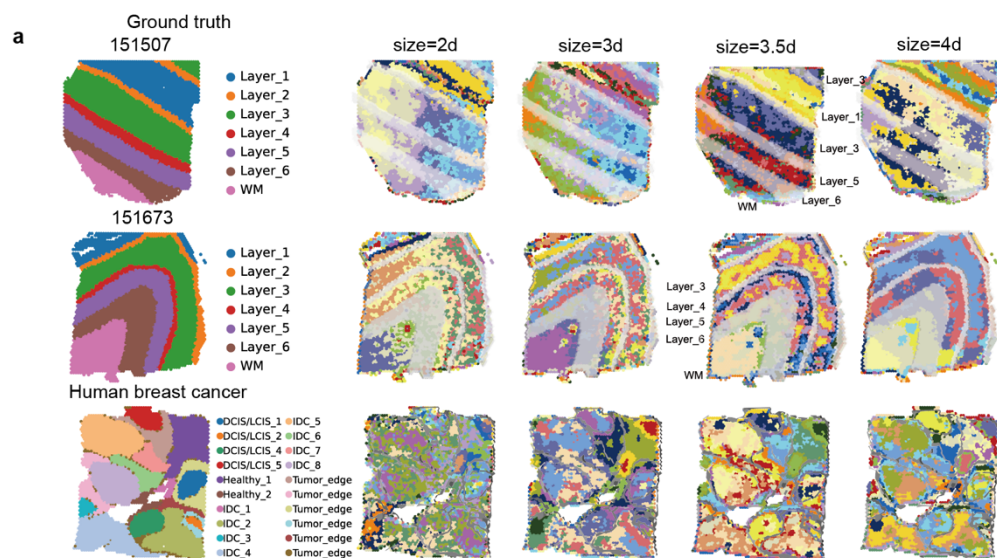
(a) 3D surface plots of SC and DB on MERFISH mouse visual cortex dataset with the neighboring node  $k$  of 3. The x-axis represents epochs  $E$ , the y-axis represents the temperature parameter  $\tau$ , and the z-axis represents SC or DB. (b) 3D surface plots of SC and DB on MERFISH mouse visual cortex dataset with  $k$  of 5. (c) 3D surface plots of SC and DB on MERFISH mouse visual cortex dataset with  $k$  of 7. (d) 3D surface plots of SC and DB on MERFISH mouse visual cortex dataset with GCN layers number  $l$  of 1. (e) 3D surface plots of SC and DB on MERFISH mouse visual cortex dataset with  $l$  of 2. (f) Heatmap of SC and DB on MERFISH mouse visual cortex dataset with  $k$  of 5 and  $l$  of 1. The horizontal axis represents  $E$ , and the vertical axis represents  $\tau$ .



**Fig. S27 Analysis of masked feature ratios in graph augmentation on other platform, comparing adaptive edge perturbation based on gene similarity with random perturbation and ablation studies on the overall framework.**

(a) Line graphs showing ARI and NMI across different gene mask ratios on STARmap. The x-axis represents gene mask ratio, and the y-axis represents scores. (b) Line graphs showing SC across different gene mask ratios on Stereo-seq and MERFISH. The x-axis represents gene mask ratio, and the y-axis represents scores. (c) Line graphs showing DB across different gene mask ratios on Stereo-seq and MERFISH. The x-axis represents gene mask ratio, and the y-axis represents scores. (d) ARI and NMI across random perturbations probabilities and different gene similarity calculation metrics on DLPFC dataset. The x-axis represents perturbations strategy, and the y-axis represents scores. (e) A bar chart depicting the SC for Stereo-seq and Slide-V2 platform under various random perturbations probabilities and different gene similarity calculation metrics. The x-axis represents perturbations strategy, and the y-axis represents scores. (f) A bar chart depicting the DB for Stereo-seq and Slide-V2 platform under various random perturbations probabilities and different gene similarity calculation metrics. The x-axis represents perturbations strategy, and the y-axis represents scores. (g) Comparative analysis of clustering results for Slide-seq V2 under random removal and gene similarity removal. Red boxes represent AOB areas. (h) Boxplots of ARI and NMI across various STAIG configurations. From left to right on the x-axis, the configurations are: 1) Complete STAIG, which includes all features; 2) STAIG with only adaptive edge removal, excluding the debiased strategy (DS); 3) STAIG exclusively utilizing the DS, without adaptive edge removal (i.e., using a fixed removal probability); 4) The basic framework of STAIG when no graph is available, omitting both DS and adaptive edge removal; and 5) The results of the baseline method, GraphST. In the boxplot, the center line denotes the median, the white dot denotes the mean value, box limits denote the upper and lower quartiles, black dots denote individual slices, and whiskers denote the  $1.5 \times$  interquartile range.

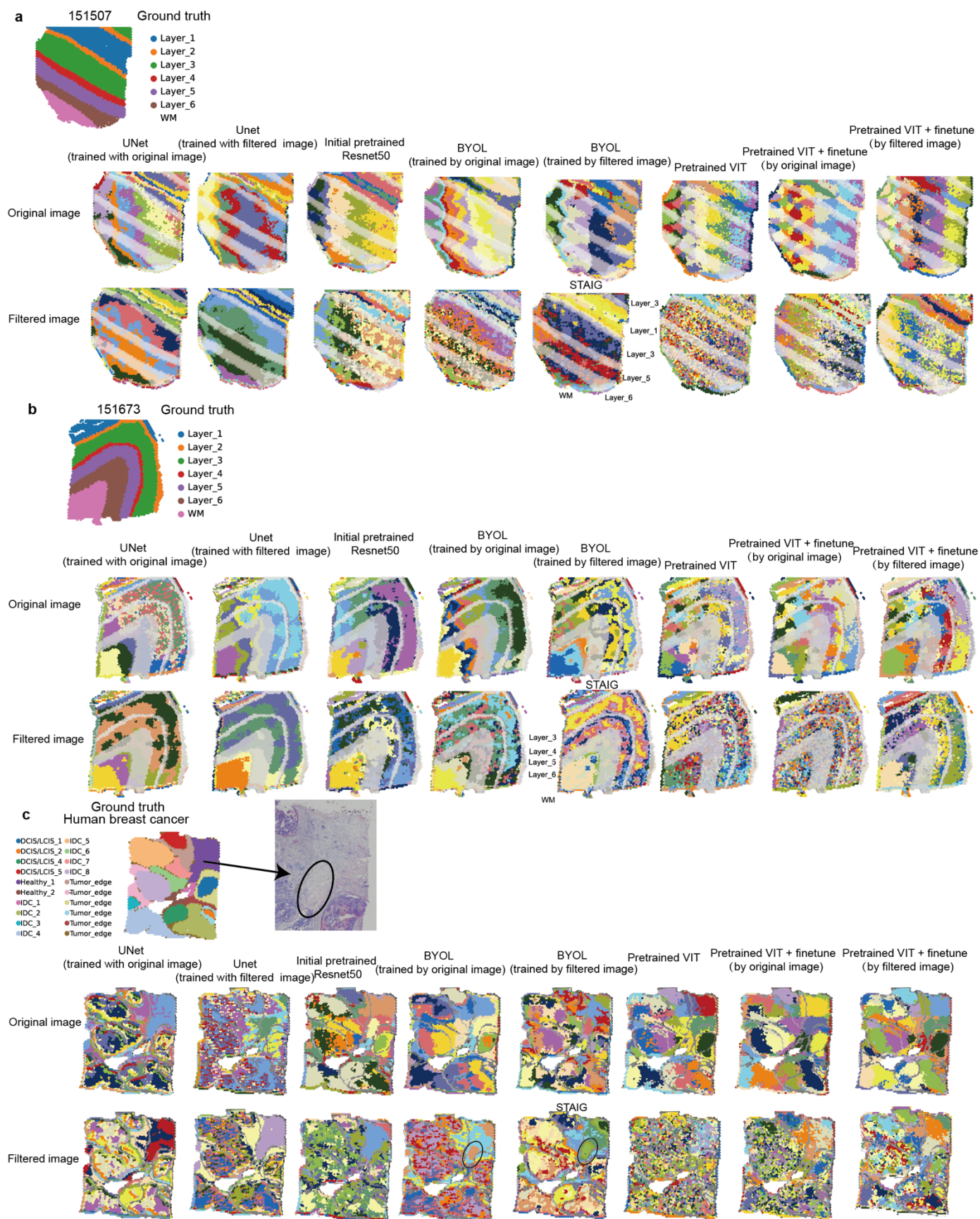




**Fig. S28 Analysis of parameter impacts on feature extraction (DLPFC #151507, #151673 and human breast cancer).**

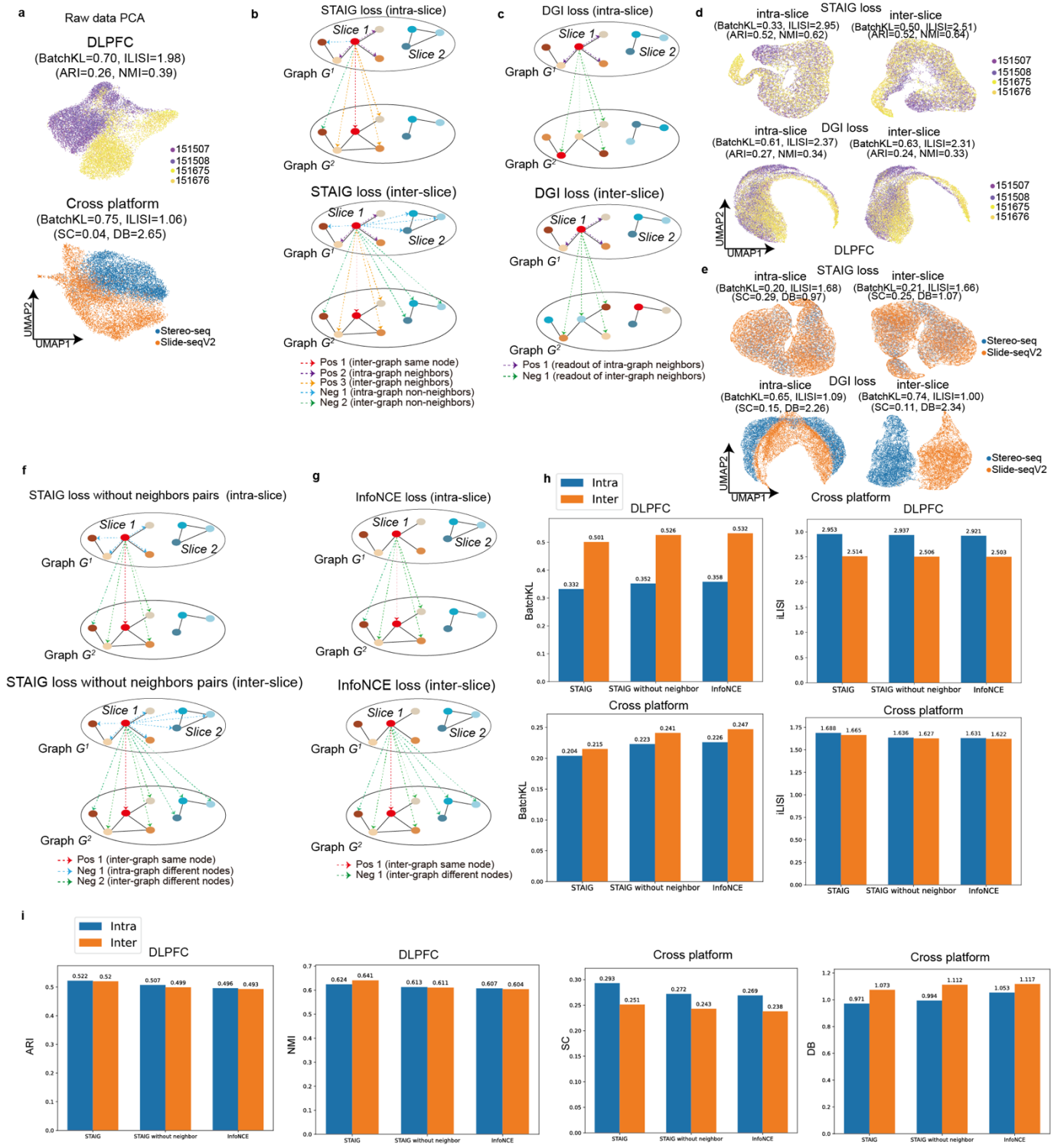
(a) The impact of varying patch sizes on feature extraction outcomes. (b) The influence of different Gaussian kernel parameters on feature extraction. (c) The effects of various filtering settings on feature extraction results. Semi-transparent masks overlaying ground truth contours are applied on slices #151507 and #151673. Additionally, clustering results for human breast cancer features include manually annotated edge contours (depicted with gray lines).





**Fig. S29 BYOL ablation study for histology feature extraction.**

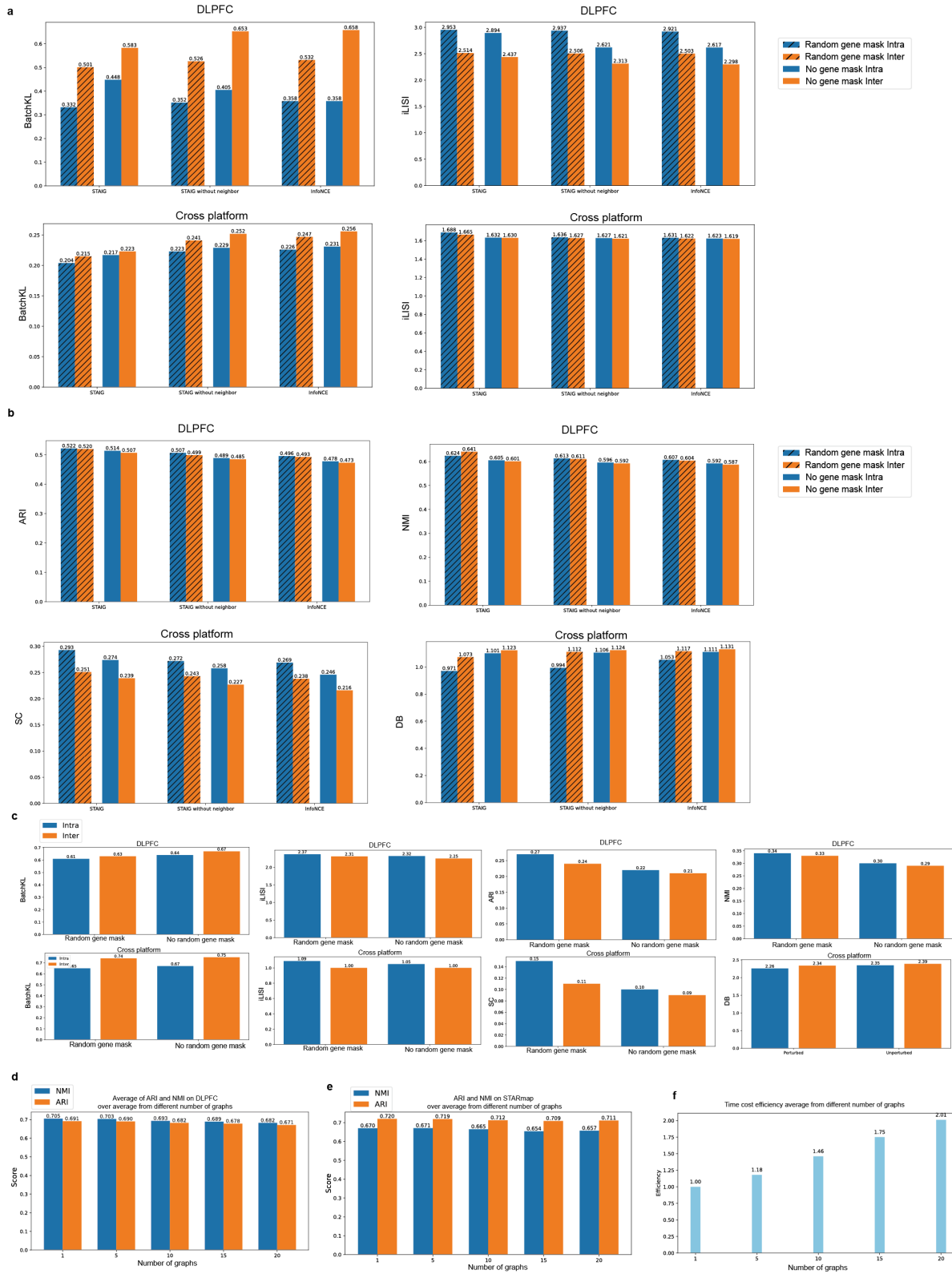
(a) DLPFC #151507 ablation result with annotation. (b) DLPFC #151673 ablation result with annotation. (c) Human breast cancer ablation result with annotation and a part of original histology image from Healthy\_1. Black circle is the region BYOL recognized. From left to right, the models tested are a U-Net trained on original images, a U-Net trained on filtered images, an initially imported pretrained ResNet50, a BYOL trained on original images, a BYOL trained on filtered images, a pretrained ViT, a pretrained ViT fine-tuned with original images, and a pretrained ViT fine-tuned with filtered images.



**Fig. S30 Ablation studies for integration.**

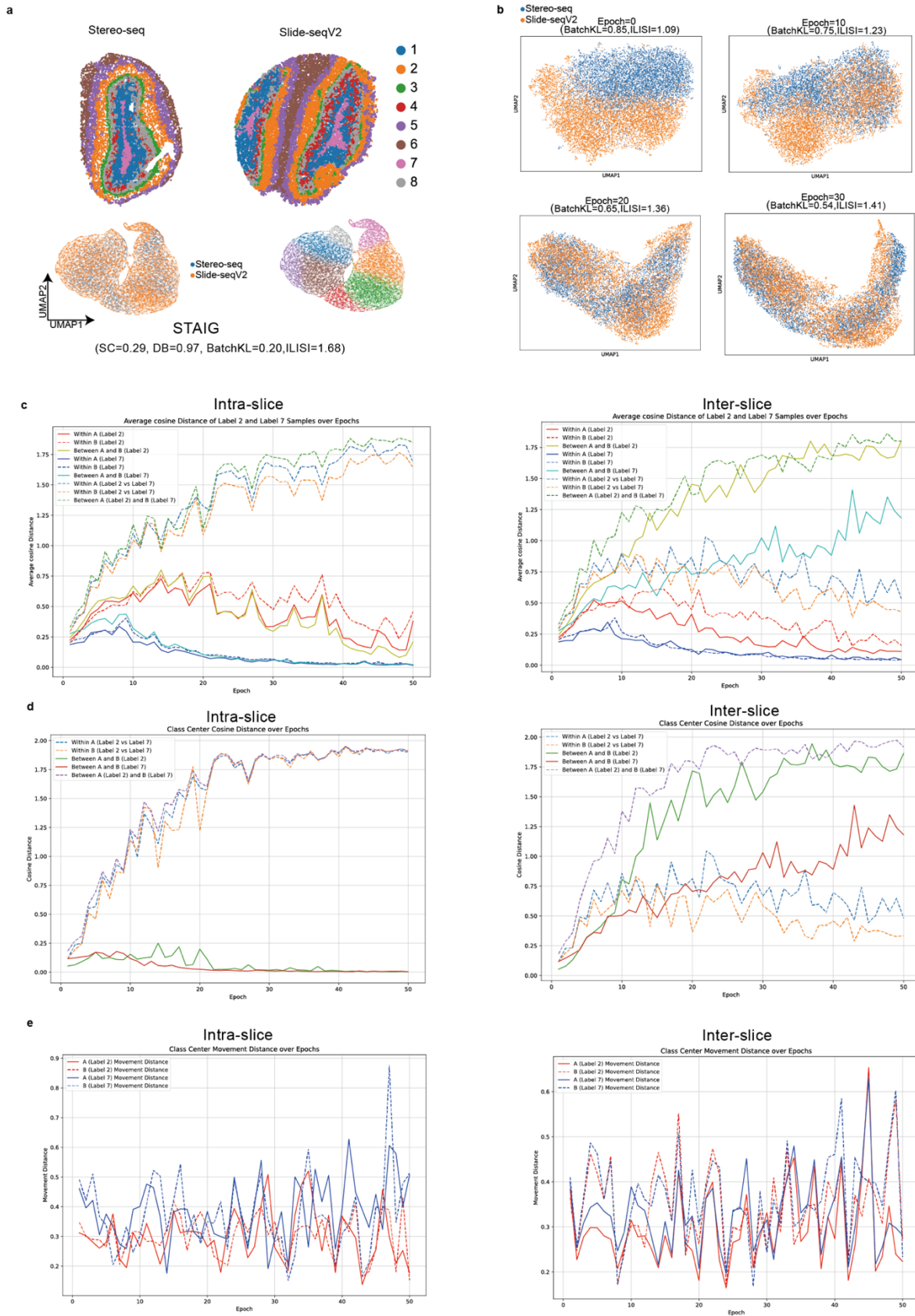
(a) UMAP visualization of Raw data after PCA. (b) Schematic diagram of the STAIG loss. (c) Schematic diagram of the DGI loss. (d) Integration of STAIG and DGI in DLPFC. (e) Integration of STAIG and DGI in cross platform. (f) Schematic diagram of the STAIG loss without neighbors. (g) Schematic diagram of the InfoNCE loss. (h) Comparison of BatchKL and iLISI in different loss functions on intra- and inter-slice integration on DLPFC and cross platform task. The x-axis represents different functions, and the y-axis represents scores. (i) Comparison of ARI, NMI, SC and DB in different loss functions on intra- and inter-slice integration on DLPFC and cross platform task. The x-axis represents different functions, and the y-axis represents scores.





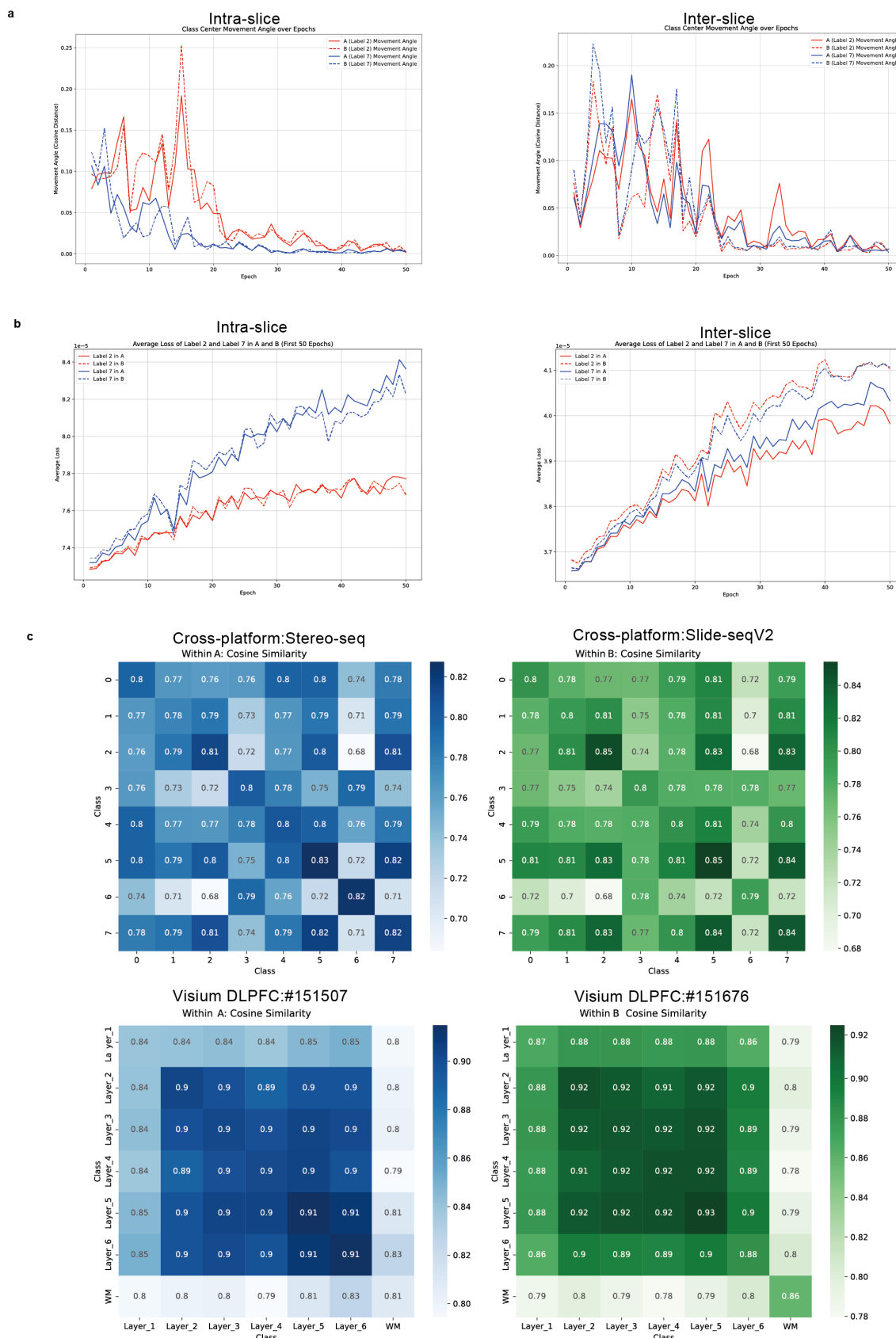
**Fig. S31 Ablation studies for integration and stability of the averaged structure.**

(a) Comparison of BatchKL and iLISI in different loss functions on intra- and inter-slice integration with random gene mask. on DLPFC and cross platform task. The x-axis represents different functions, and the y-axis represents scores. (b) Comparison of ARI, NMI, SC and DB in different loss functions on intra- and inter-slice integration with random gene mask. The x-axis represents different functions, and the y-axis represents scores. (c) Comparison of DGI loss functions on intra- and inter-slice integration with random gene mask. The x-axis represents different functions, and the y-axis represents scores. (d) ARI and NMI on DLPFC from different number of graphs. The x-axis represents number of graphs, and the y-axis represents scores. (e) The ARI and NMI on STARmap from different number of graphs. The x-axis represents number of graphs, and the y-axis represents scores. (f) Time cost from different number of graphs. The x-axis represents number of graphs, and the y-axis represents efficiency ratio.



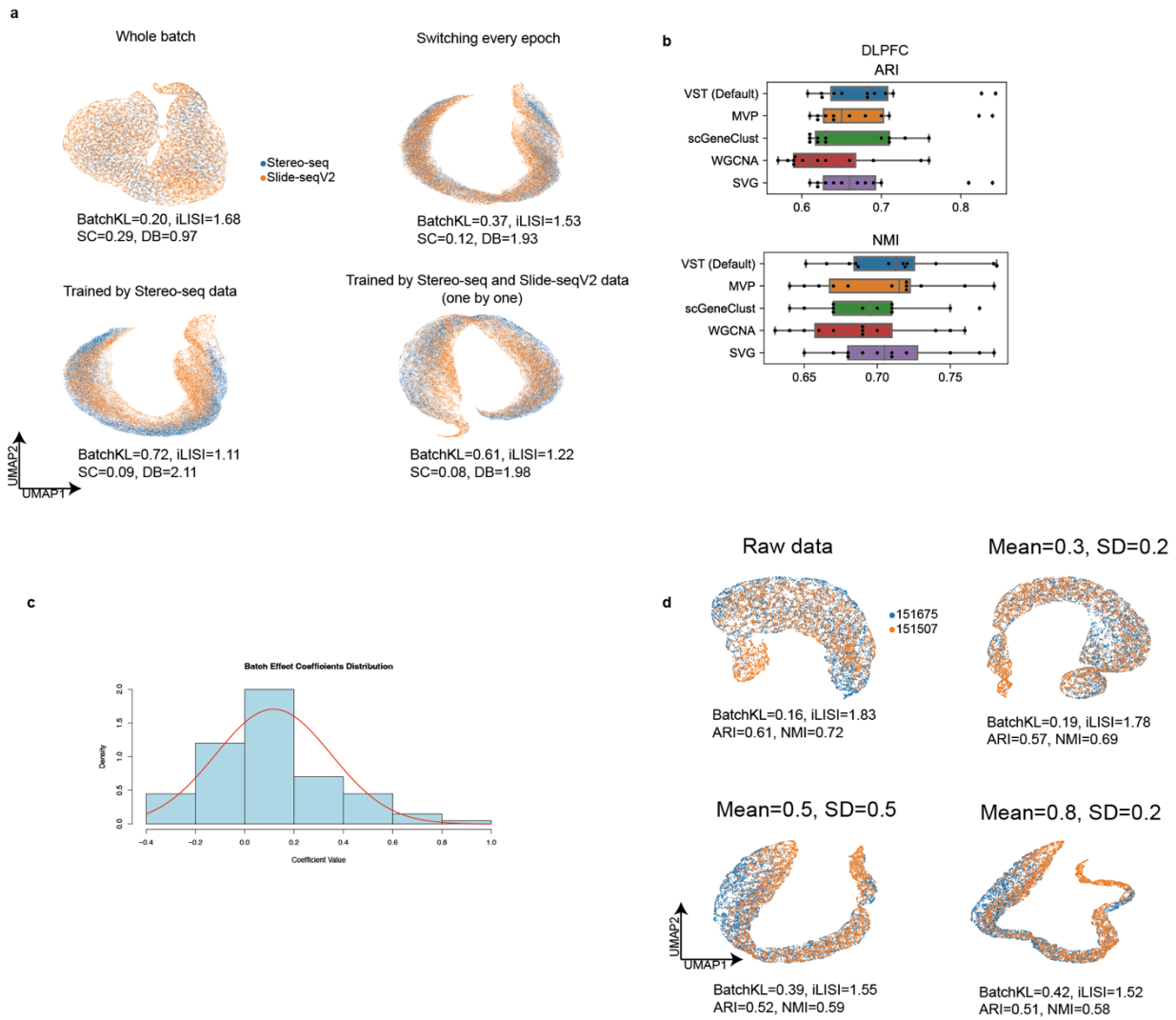
**Fig. S32 The principle of STAIG integration in latent space.**

(a) Cross platform integration result. (b) UMAP visualization for first 30 epochs in latent space. (c) Average cosine distance between clusters 2 and 7 within slice A, within slice B, and between clusters 4 and 7 across different slices. The x-axis represents epochs, and the y-axis represents cosine distance. (d) The cosine distances between the centers of samples from different classes within the same slice, as well as the distances between centers of the same class samples across different slices. The x-axis represents epochs, and the y-axis represents cosine distance. (e) The movement of Euclidean distance between the centers of clusters 2 and 7 from different slices after each iteration. The x-axis represents epochs, and the y-axis represents movement distance.



**Fig. S33 The principle of STAIG integration in latent space and the heatmaps of cosine similarity between different class labels.**

(a) The movement of angular differences (cosine distance) between the centers of clusters 2 and 7 from different slices after each iteration. The x-axis represents epochs, and the y-axis represents movement angular. (b) The average loss of cluster 2 and 7 in slice A and B. The x-axis represents epochs, and the y-axis represents average loss. (c) The heatmaps of cosine similarity between different class labels, comparing cross-platform integration and the integration of different slices from the Visium platform in the initial latent space. The x-axis and the y-axis represents class.



**Fig. S34 Effect of different training methods and different gene selection methods in STAIG and Batch effect performance test.**

(a) The UMAP of Integration with different training styles. (b) ARI and NMI of DLPFC dataset by different gene selection methods. In the boxplot, the center line denotes the median, the white dot denotes the mean value, box limits denote the upper and lower quartiles, black dots denote individual slices, and whiskers denote the  $1.5 \times$  interquartile range. (c) The original batch marginal distribution of #151507 and #151675. The x-axis represents coefficient value, and the y-axis represents density. (d) Performance of integration by different batch effect on #151507 and #151675.

## Supplementary References

1. Hao, Y. *et al.* Integrated analysis of multimodal single-cell data. *Cell* **184**, 3573–3587.e29 (2021).
2. Long, Y. *et al.* Spatially informed clustering, integration, and deconvolution of spatial transcriptomics with GraphST. *Nat Commun* **14**, 1155 (2023).
3. Xu, C. *et al.* DeepST: identifying spatial domains in spatial transcriptomics by deep learning. *Nucleic Acids Research* **50**, e131–e131 (2022).
4. Dong, K. & Zhang, S. Deciphering spatial domains from spatially resolved transcriptomics with an adaptive graph attention auto-encoder. *Nat Commun* **13**, 1739 (2022).
5. Hu, J. *et al.* SpaGCN: Integrating gene expression, spatial location and histology to identify spatial domains and spatially variable genes by graph convolutional network. *Nat Methods* **18**, 1342–1351 (2021).
6. Fu, H. *et al.* *Unsupervised Spatially Embedded Deep Representation of Spatial Transcriptomics*. <http://biorxiv.org/lookup/doi/10.1101/2021.06.15.448542> (2021) doi:10.1101/2021.06.15.448542.
7. Zong, Y. *et al.* *conST: An Interpretable Multi-Modal Contrastive Learning Framework for Spatial Transcriptomics*. <http://biorxiv.org/lookup/doi/10.1101/2022.01.14.476408> (2022) doi:10.1101/2022.01.14.476408.
8. Zhang, L., Liang, S. & Wan, L. A multi-view graph contrastive learning framework for deciphering spatially resolved transcriptomics data. *Briefings in Bioinformatics* **25**, bbae255 (2024).
9. Pham, D. *et al.* *stLearn: Integrating Spatial Location, Tissue Morphology and Gene Expression to Find Cell Types, Cell-Cell Interactions and Spatial Trajectories within Undissociated Tissues*. <http://biorxiv.org/lookup/doi/10.1101/2020.05.31.125658> (2020) doi:10.1101/2020.05.31.125658.
10. Zhou, X., Dong, K. & Zhang, S. Integrating spatial transcriptomics data across different conditions, technologies and developmental stages. *Nat Comput Sci* **3**, 894–906 (2023).
11. Hu, J. *et al.* Deciphering tumor ecosystems at super resolution from spatial transcriptomics with TESLA. *Cell Systems* **14**, 404–417.e4 (2023).
12. He, K. *et al.* Masked Autoencoders Are Scalable Vision Learners. in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 15979–15988 (IEEE, New Orleans, LA, USA, 2022). doi:10.1109/CVPR52688.2022.01553.
13. Oord, A. van den, Li, Y. & Vinyals, O. Representation Learning with Contrastive Predictive Coding. Preprint at <https://doi.org/10.48550/ARXIV.1807.03748> (2018).
14. Steinley, D. Properties of the hubert-arable adjusted rand index. *Psychological methods* **9**, 386 (2004).
15. Estevez, P. A., Tesmer, M., Perez, C. A. & Zurada, J. M. Normalized Mutual Information Feature Selection. *IEEE Trans. Neural Netw.* **20**, 189–201 (2009).
16. Shahapure, K. R. & Nicholas, C. Cluster quality analysis using silhouette score. in *2020 IEEE 7th international conference on data science and advanced analytics (DSAA)* 747–748 (IEEE, 2020).
17. Vergani, A. A. & Binaghi, E. A Soft Davies-Bouldin Separation Measure. in *2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)* 1–8 (IEEE, Rio de Janeiro, 2018). doi:10.1109/FUZZ-IEEE.2018.8491581.
18. Li, X. *et al.* Deep learning enables accurate clustering with batch effect removal in single-cell RNA-seq analysis. *Nat Commun* **11**, 2338 (2020).
19. Korsunsky, I. *et al.* Fast, sensitive and accurate integration of single-cell data with Harmony. *Nat Methods* **16**, 1289–1296 (2019).
20. Veličković, P. *et al.* Deep Graph Infomax. Preprint at <http://arxiv.org/abs/1809.10341> (2018).
21. Zhu, Y. *et al.* Deep Graph Contrastive Representation Learning. (2020) doi:10.48550/ARXIV.2006.04131.
22. Pun, N. S. & Agarwal, S. BT-Unet: A self-supervised learning framework for biomedical image segmentation using barlow twins with U-net models. *Mach Learn* **111**, 4585–4600 (2022).



23. Parulekar, A., Collins, L., Shanmugam, K., Mokhtari, A. & Shakkottai, S. Infonce loss provably learns cluster-preserving representations. in *The Thirty Sixth Annual Conference on Learning Theory* 1914–1961 (PMLR, 2023).
24. Huang, W., Yi, M., Zhao, X. & Jiang, Z. Towards the Generalization of Contrastive Self-Supervised Learning. in *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023* (OpenReview.net, 2023).
25. Yuan, Y. *et al.* Towards generalizable Graph Contrastive Learning: An information theory perspective. *Neural Networks* **172**, 106125 (2024).
26. Hu, W. *et al.* Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems* **33**, 22118–22133 (2020).
27. Deng, T. *et al.* A cofunctional grouping-based approach for non-redundant feature gene selection in unannotated single-cell RNA-seq analysis. *Briefings in Bioinformatics* **24**, bbad042 (2023).
28. Rezaie, N., Reese, F. & Mortazavi, A. PyWGCNA: a Python package for weighted gene co-expression network analysis. *Bioinformatics* **39**, btad415 (2023).
29. Svensson, V., Teichmann, S. A. & Stegle, O. SpatialDE: identification of spatially variable genes. *Nat Methods* **15**, 343–346 (2018).
30. Song, D. *et al.* scDesign3 generates realistic in silico data for multimodal single-cell and spatial omics. *Nat Biotechnol* (2023) doi:10.1038/s41587-023-01772-1.
31. Maynard, K. R. *et al.* Transcriptome-scale spatial gene expression in the human dorsolateral prefrontal cortex. *Nat Neurosci* **24**, 425–436 (2021).
32. Hunter, M. V., Moncada, R., Weiss, J. M., Yanai, I. & White, R. M. Spatially resolved transcriptomics reveals the architecture of the tumor-microenvironment interface. *Nat Commun* **12**, 6278 (2021).
33. Chen, A. *et al.* Spatiotemporal transcriptomic atlas of mouse organogenesis using DNA nanoball-patterned arrays. *Cell* **185**, 1777–1792.e21 (2022).
34. Stickels, R. R. *et al.* Highly sensitive spatial transcriptomics at near-cellular resolution with Slide-seqV2. *Nat Biotechnol* **39**, 313–319 (2021).
35. Wang, X. *et al.* Three-dimensional intact-tissue sequencing of single-cell transcriptional states. *Science* **361**, eaat5691 (2018).
36. Fang, R. *et al.* Conservation and divergence of cortical cell organization in human and mouse revealed by MERFISH. *Science* **377**, 56–62 (2022).