nature methods

Supplementary information

https://doi.org/10.1038/s41592-024-02410-7

# Search and match across spatial omics samples at single-cell resolution

In the format provided by the authors and unedited

# Supplementary Information

# Search and match across spatial omics samples at single-cell resolution

Zefang Tang[1,2,3], Shuchen Luo[1,2,3], Hu Zeng[1,2], Jiahao Huang[1,2], Xin Sui[1,2], Morgan Wu[1], Xiao Wang[1,2]*

[1]Broad Institute of MIT and Harvard, Cambridge, MA, USA.

[2]Department of Chemistry, Massachusetts Institute of Technology, Cambridge, MA, USA.

[3]These authors contributed equally to this work.

*Correspondence e-mail: xwangx@mit.edu

## Design and benchmark of CAST architecture

### *Benchmark of CAST Mark*

To investigate each hyperparameter in CAST Mark, we studied all four model architecture hyperparameters (*n_layers*, number of GNN layers; *encoder_dim*, dimension of the MLP encoder; *dfr*, feature dropout rate; *der*, edge dropout rate) and performed ablation studies for the graph convolutional layers (Supplementary Fig. 1-3). We modified one parameter at a time and performed 10 replicates of each combination of hyperparameters (with random seeds generated based on the system time). The median distance of the ground-truth cell pairs in aligned samples S1-S1' was used as the evaluation metric (higher distance value indicates worse performance).

**1. Number of GCNII layers.** We chose the largest number of GCNII layers (*n_layer* = 9) as recommended by the original GCNII paper[1]. Our experimental results on the simulation dataset confirm that **increasing the number of layers improves the accuracy of CAST alignment** (Supplementary Fig. 1a), presumably due to the increased contrast and spatial resolution of learned graph embeddings in layer-shaped anatomical regions (Supplementary Fig. 2). We recapitulated the enhanced resolution in layer structures using real samples S1-S8 (Supplementary Fig. 3). In the particular case of brain slice alignment, as the cortical layers are molecularly similar across different cortical areas, insufficient contrast of different cortical layers will result in inaccurate orientation of the tissue slice. **These results confirmed that the performance gain from a deep GNN architecture is essential for high-resolution spatial alignment and lays an important foundation for spatial integration at single-cell granularity by CAST Projection.**

**2. Encoder dimension.** The purpose of the MLP encoder is to reduce the time and space complexity of the model, especially given datasets with large gene panels. As our test dataset has a gene panel of 2,766 genes, we chose *encoder_dim* from 32 to 2048, along with an experimental group without the MLP encoder module. Results (Supplementary Fig. 1b) showed that encoder dimensions 256 and 512 yielded comparable and even slightly better alignment performance than the “No encoder” group. Therefore, we recommend using 256 and 512 for parameter *encoder_dim* for datasets with large gene panels (larger than 1,000 genes). We recommend using “No encoder” for datasets with limited gene panels (smaller than 1,000 genes).

**3. Feature dropout rate (*dfr*).** This hyperparameter controls the extent of feature dropout for graph augmentation in the CCA-SSG self-supervised learning model[2]. *dfr* = 1 means complete dropout while *dfr* = 0 means no dropout. We scanned the entire possible range for *dfr*. For CAST, we used *dfr* = 0.3 as the default unless otherwise stated, following the CCA-SSG paper[2]. Our parameter sensitivity experiments (Supplementary Fig. 1c) showed that alignment performance is optimal from 0.1 to 0.4. We recommend users use the default *dfr* value unless necessary.

**4. Edge dropout rate (*der*).** This hyperparameter controls the extent of graph edge dropout for graph augmentation in the CCA-SSG self-supervised learning model[2]. *der* = 1 means complete dropout, while *der* = 0 means no dropout. We scanned the entire possible range for *der*. For CAST, we used *der* = 0.5 as the default unless otherwise stated, the same as the default used in CCA-SSG paper[2]. Our sensitivity experiments (Supplementary Fig. 1d) showed that alignment performance is optimal from 0.3 to 0.7. We recommend users use the default *der* value unless necessary.

**5. Ablation studies of the GCNII graph convolutional layer.** We adopted the GCNII graph convolution layer to enable smooth learning with a higher number of graph convolutional layers than traditional GNN architectures. Since we have shown that increasing the number of layers boosts alignment performance (Supplementary Fig. 1a), we further performed ablation studies using the vanilla GCN[3] with 2-9 graph convolution layers, which were visualized alongside our hyperparameter sensitivity experiments performed using 2-9 GCNII layers (Supplementary Fig. 1e). The vanilla GCN layer architecture failed to generate alignment with acceptable performance (median of ground-truth distance > 200 μm), while increasing the number of layers did not improve the performance. In addition to our point made in the main text that GCNII brings performance boosts by enabling learning with more graph convolutional layers, we concluded that the vanilla GCN convolution layers failed the challenge of high-resolution alignment on single-cell resolved spatial datasets.

### *Benchmark of CAST Stack*

To better illustrate the gradient descent process, we benchmarked several important parameters and showed the intermediate results (already wrapped in the CAST Stack code) and final aligned figures (Supplementary Fig. 4 and Supplementary Information Table 1).

**Supplementary Information Table 1 | The parameter list of the trials in the CAST Stack benchmark**

| Batch | Iterations (affine) | Iterations (B-spline) | Mesh size | Translation params | Notes |
|---|---|---|---|---|---|
| b1 | 500 | 400 | 8 | False | Control |
| b2 | **100** | 400 | 8 | False | |
| b3 | 500 | **50** | 8 | False | |
| b4 | 500 | 400 | **4** | False | |
| b5 | 500 | 400 | **10** | False | |
| b6 | 500 | 400 | 8 | **True** | |

**1. The iterations of affine transformation.** We first compared the results after different numbers of iterations of affine transformation gradient descent (500 versus 100, Supplementary Fig. 4a). The profiles of the five parameters *a*, *d*, Φ, $b_1$, and $b_2$ as well as the value of the loss function are shown. With only 100 iterations, we found that in the 100-iteration task, the DG region of the query sample (S4) exhibited a small shift to the one in the reference sample (S1) and the five parameters did not converge, in contrast to the 500-iteration task. Thus, 500 iterations are necessary in this case.

**2. The iterations of B-spline transformation.** Although the 500-iteration task tended to converge and well-aligned the DG region, the CA1 region appears to be less well aligned. Thus, the B-spline phase is important to adjust such differences. With more iterations in the B-spline phase, we found that CA1 could be better aligned (Supplementary Fig. 4b).

**3. The mesh size for B-spline transformation.** Afterwards, we benchmarked the different values of the *meshgrid* parameters. A smaller value of the *meshgrid* generally gives coarse-grained adjustment, while a higher value of the *meshgrid* could adjust more details. We observed that the task with the *Mesh_Size* = 4 (the control points *s*) exhibits poor alignment performance compared to the tasks with *Mesh_Size* = 8 or *Mesh_Size* = 10 (Supplementary Fig. 4c).

**4. Translation in pre-location step.** CAST Stack uses a pre-location strategy to map the query sample to a proper initial location based on the loss function values. By default, this step will try rotations (0°, 90°, 180°, and 270°) as well as reflection and then select the location with the lowest loss function value as the initial position. Here, we tested if allowing translation trials (details are listed below) will help the alignment (Supplementary Fig. 4d). Although the alignment results are similar in this task, the pre-location with translation can provide more trials for better initiations, which benefit a lot in the small-to-big alignment tasks (Fig. 3c and Extended Data Fig. 4f,g).

*Translation trials in the pre-location step.* In this step, we established a range of ±50% for the x and y dimensions of the query sample, dividing each dimension into 10 intervals. Subsequently, all trials yield a loss function value for comparison. CAST Stack automatically selects the one with the lowest value as the initial location of the query sample.

**5. Before and after B-spline transformation.** As we showed previously (Supplementary Fig. 4a,b), the B-spline step is very important to adjust the differences that cannot be solved by the pure affine transformation. We also attached an example (Supplementary Fig. 4e) to showcase the necessity of non-rigid B-spline alignment even for adjacent tissue slices. The arrows in the figure highlighted that non-rigid B-spline alignment can correct local deformation between consecutive RIBOmap and STARmap sections in Mouse 2.

### *Benchmark of CAST Projection*

**1. CAST Projection is flexible for different low-dimensional embeddings.** Actually, CAST Projection can accommodate various shared embeddings as a bridge between the two datasets. The combination of Combat and Harmony, the default integration strategy in CAST, is a proper and robust way to integrate cells from different modalities. Such strategy was used in the publications of the mouse CNS atlas[4] and RIBOmap[5]. However, there are also a series of great works developed to integrate different omics data into a shared low-dimensional space, like Seurat[6], LIGER[7], and MNN[8]. We tested CAST Projection based on the integrated features generated by these methods (Supplementary Fig. 10c-e). The performances of all three methods are pretty good, especially the LIGER and Seurat, indicating that CAST Projection can easily be adapted to different shared embeddings. Notably, when considering the region similarity, we recommend that the CAST Mark spatial features are a better option because of its ability to precisely capture spatial features.

**2. The spatial constraint in CAST Projection is critical.** As the integration methods are well developed, when designing CAST Projection, we focused on how to precisely and reasonably link two cells across different slices considering the spatial information.

When projecting the cell in the query slice to the one with closest cosine distance in the reference slice without any spatial constraints, we found that the cell pairs were separated by long physical distances, though the consistency of the cell type labels is higher (TP% = 0.93, compared with the initial CAST Projection result: 0.88 for S2-S1; 0.91 for S3-S1; 0.91 for S4-S1. Supplementary Fig. 9). Thus, the spatial constraints are critical for proper spatial-to-spatial projection.

## Methodology advances of CAST

We summarize the novelties of CAST as several points:

1. **Deep GNN increases the range of perception** (Supplementary Fig. 1-3). In contrast to existing graph-based tools that merely utilize shallow layers (2-3 layers), CAST harnesses the power of deep graph layers (GCNII). This approach expands the receptive field, enabling the capture of more precise and consistent spatial features when undertaking tissue segmentation tasks, especially in large-scale single-cell resolved spatial omics data.

2. **New training strategy enables cross-sample spatial feature detection.** CAST has another two major innovations include (1) a sequential training strategy that allows for quick, memory-efficient training (Supplementary Fig. 5), and (2) the CCA training objective to identify consistent spatial features among samples despite drastic sample-level differences in technologies and conditions (Fig. 1c and Extended Data Fig. 2a).

3. **Affine and Non-rigid warping enable handling more variable datasets.** CAST Stack builds affine and B-spline transformations at its core. This enables CAST to tackle the prevalent challenge of significant discrepancies between two slices (Supplementary Fig. 4).

**4. Sub-sampling strategy enables wider application on larger datasets.** CAST uses a fast and low-memory-cost solution to apply the gradient-based non-rigid alignment (Supplementary Fig. 5), particularly in the case of single cell-based data. Thus, it can easily handle datasets with a substantial number of items or different spatial transcriptome technologies (>50k items, like Stereo-seq, Slide-seq, STARmap, and MERFISH (Supplementary Table 4).

## Details of the AD datasets ∆Analysis

*∆Cell analysis*. We calculated ∆Cell (*R* = 50μm) in the aligned cross-sample niches for three disease-relevant glial cell types: astrocytes, microglia, and oligodendrocytes (Supplementary Fig. 7a,b). For easier interpretation, the ∆Cell values of each cell-centered neighborhood are averaged based on their CAST Mark region labels (Extended Data Fig. 5c). The ∆Cell of oligodendrocytes shows the most significant changes in the white matter (WM) region in the 13-mos group and correlates with the accumulation of p-tau intensity (Supplementary Fig. 7b,c), consistent with the previously reported correlation between the oligodendrocytes and the p-tau[9,10]. The abundance of microglia increased in retrosplenial cortex (RSP), cortical layer 5 (L5), and cortical layer 6 (L6) regions at 8 months, and kept increasing in all regions except the hippocampal CA2 and CA3 regions (CA2 and CA3) at 13 months (Supplementary Fig. 7b). The elevated abundance of microglia in disease versus control quantified by ∆Cell shows a high spatial correlation with the density of the increased Aβ plaques (Supplementary Fig. 7d), further validating the activated response of microglia to the Aβ plaque[11,12]. The ∆Cell values with different radii (*R* from 5 μm to 200 μm) show a consistent spatial pattern (Supplementary Fig. 7e).

*Cell-specific ∆Exp screening.* We calculated the delta-sample single-cell gene expression changes (∆Exp, *R* = 50μm) in each niche to interrogate the spatial gradient of differential gene expression (Extended Data Fig. 5d-f). Further correlating ΔExp with spatial locations of the Aβ-plaque enables us to identify spatially resolved plaque-induced genes (PIGs; Methods). Based on this strategy, we identified 30 spatially resolved PIGs (15 PIGs in the 8-mos group; 27 PIGs in the 13-mos group; Extended Data Fig. 5g; Supplementary Table 6). These genes are over-expressed in the disease samples, and their ΔExp patterns are spatially correlated with Aβ plaques (Extended Data Fig. 5h) and enriched in peptide binding (GO:0042277) and the lysosome pathway (mmu04142; Extended Data Fig. 5i; Supplementary Table 6). Notably, our results are consistent with previously reported PIGs identified by Spatial Transcriptomics (ST) technology[11] and the initial STARmap PLUS study (Fisher Exact test, $p < 0.001$), such as *Apoe*, *C1qb*, *Cd63*, *Ctsb*, and *Gfap*. More importantly, the ΔExp analysis further uncovered the regional heterogeneity of changes in PIG expression levels in AD. For instance, *Ctsb*, an AD-related gene[13], does not show uniform over-expression in the microglia across the sample; rather, it exhibits over-expression in the CA stratum lacunosum-moleculare (CAslm), L6, and RSP regions in the 13-mos comparison (Extended Data Fig. 5d-f).

## Supplementary Methods Notes

**Sub-sampling strategy of CAST Mark and CAST Stack.**

To reduce the computational cost and adjust the different spatial resolution between different samples or spatial omics technologies, we designed the sub-sampling strategy for CAST Mark and CAST Stack.

*Sub-sampling for CAST Mark.* Given the number of sub-sampling nodes (e.g. $n$ = 10,000 nodes), CAST will randomly select $n$ cells as the sub-sampling nodes. Each cell in the slice will be assigned to its nearest sub-sampling node. The gene expression (by default, *norm1e4*) of cells grouped at each node are aggregated as the new gene expression for that node. These newly formed sub-sampled datasets become the input for CAST Mark.

*Sub-sampling for CAST Stack.* In this approach, CAST randomly selects $n$ cells (e.g., $n$ = 10,000) to act as anchor cells for the CAST Stack.

*Methods of Sub-sampling Alignment.* There are two primary strategies for sub-sampling alignment:

1. CAST Mark with Sub-sampling, followed by CAST Stack: Here, sub-sampling is initially applied for CAST Mark, and all sub-sampling nodes are subsequently processed through CAST Stack. The learned transformation is then applied to the original sample.

2. Full-size Dataset for CAST Mark, with Sub-sampling in CAST Stack: In this method, CAST Mark utilizes the full-size dataset. Subsequently, sub-sampling is applied for CAST Stack where the original sample is transformed in conjunction with the sub-sampling anchor nodes.

**Sample preparation of the STARmap Mouse 2 sample.** The mouse (C57BL/6 strain) utilized in this study was anesthetized with isoflurane and subsequently decapitated quickly. We collected the brain tissue and embedded it in the Tissue-Tek O.C.T. Compound, which is next frozen in liquid nitrogen and stored at -80 °C. The mouse brain tissue was further transferred to a cryostat (Leica CM1950) at -20°C and was sliced into 20 μm coronal sections. These slices were then placed on glass-bottom 12-well plates that had been pretreated with 3-(Trimethoxysilyl)propyl methacrylate and poly-D-lysine. Next, the slices were fixed with 4% PFA in PBS for 15 minutes (room temperature), permeabilized with cold methanol and kept at -80 °C for an hour. The experimental procedures for STARmap were similar to those used for HeLa cells in Zeng et al.[5], except that all the reaction volumes were doubled because the brain tissue was prepared in 12-well plates. We captured the images using a Leica TCS SP8 confocal microscopy with a 63× oil immersion objective (NA 1.4) and a voxel size of 90.14 nm × 90.14 nm × 300 nm. In the first sequencing round, we also imaged the DAPI staining signals. Overall, 9 imaging cycles were sequenced to detect the 5,413 genes.

**Data processing for STARmap brain tissue sample.**

*STARmap Imaging Preprocessing.* We used Huygens Essential version 21.04 (Scientific Volume Imaging, The Netherlands, http://svi.nl) to deconvolute the raw images (CMLE algorithm; SNR:10 and 10 iterations). The image preprocessing operations, such as image registration, spot calling, and barcode filtering with minor adjustments were applied following the previous report[5].

*Cell Segmentation.* The RNA amplicon-based cell segmentation method ClusterMap[14] was utilized to automatically detect the cells. We applied the default pipeline with minor adjustments to the DAPI signal preprocessing to capture cells in each field of view (FOV). The following parameters were used in the ClusterMap: *cell_num_threshold* = 0.7; *dapi_grid_interval* = 5; *pct_filter* = 0.1; *window_size* = 550. After cell segmentation for each FOV, cells were stitched to generate the cell-by-gene matrix.

*Quality control and Cell typing.* We adopted the quality control strategy reported by Zeng et al.[5] for the STARmap Mouse 2 sample. The median absolute deviation (MAD) was used to estimate the lower/upper boundaries for cell filtration:

$$\begin{aligned}\text{Lower boundary} &= \text{median}(\text{reads per cell}) - 3 \times MAD\\ \text{Upper boundary} &= \text{median}(\text{reads per cell}) + 3 \times MAD\end{aligned}$$

The cells with reads fewer than the lower boundary or greater than the upper boundary were filtered out, resulting in 44,751 cells and 5,413 genes. Employing the similar strategy reported by Zeng et al.[5], we further identified 11 cell types across 4 samples.

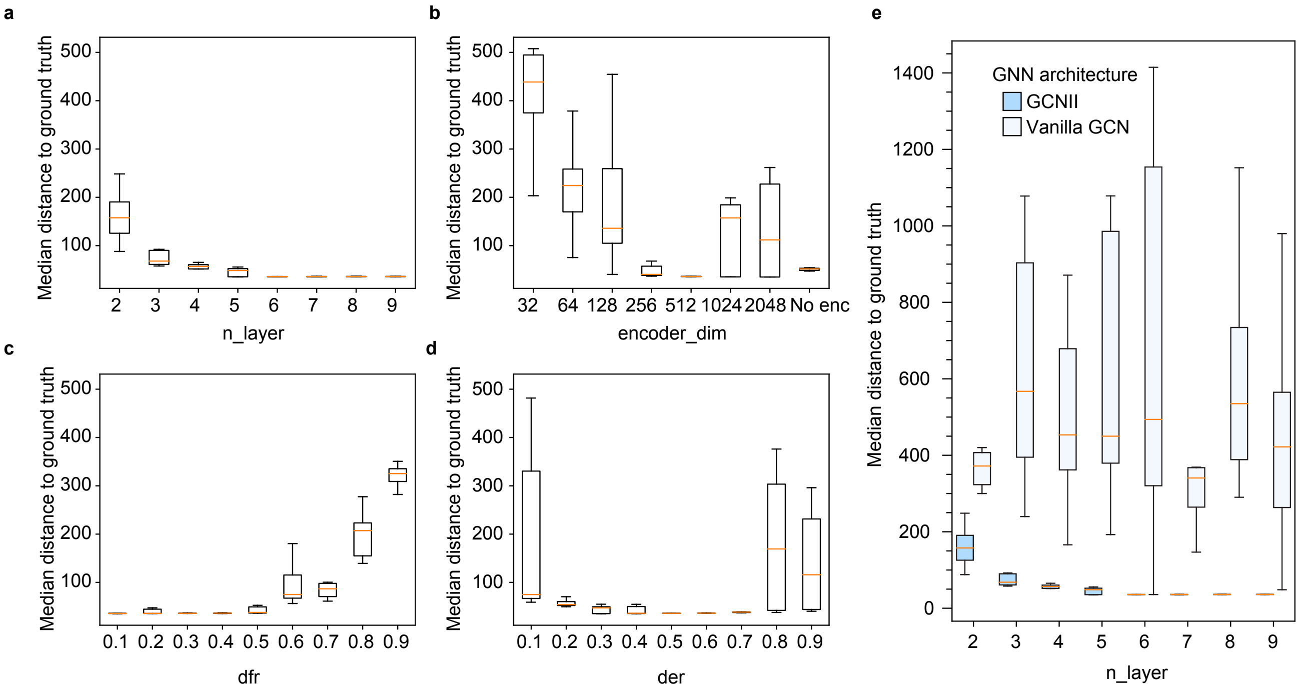
**CAST Projection performance benchmark.** To check the differences between CAST Projection and projection without spatial constraints, we used the strategy of projecting the cell from the query slice (S2, S3 or S4) to the cell in the reference slice (S1) with the closest cosine distance value (Supplementary Fig. 9).

We also compared the results of the algorithms which were designed to project single cells to spatial cells (Tangram[15] and Cell2Location[16]; Supplementary Fig. 10a,b). The *norm1e4* expression data was used in Tangram (achieved the best results in our hands) and the *raw* expression data was used in the Cell2Location (achieved the best results in our hands). The default parameters shown in the tutorials in these two methods were used in the benchmark.

In addition, we further compared the CAST Projection performance based on the integration embeddings of Seurat CCA[6], MNN[8], and LIGER[7] (Supplementary Fig. 10c-e). The default parameters shown in the tutorials are used in the benchmark.

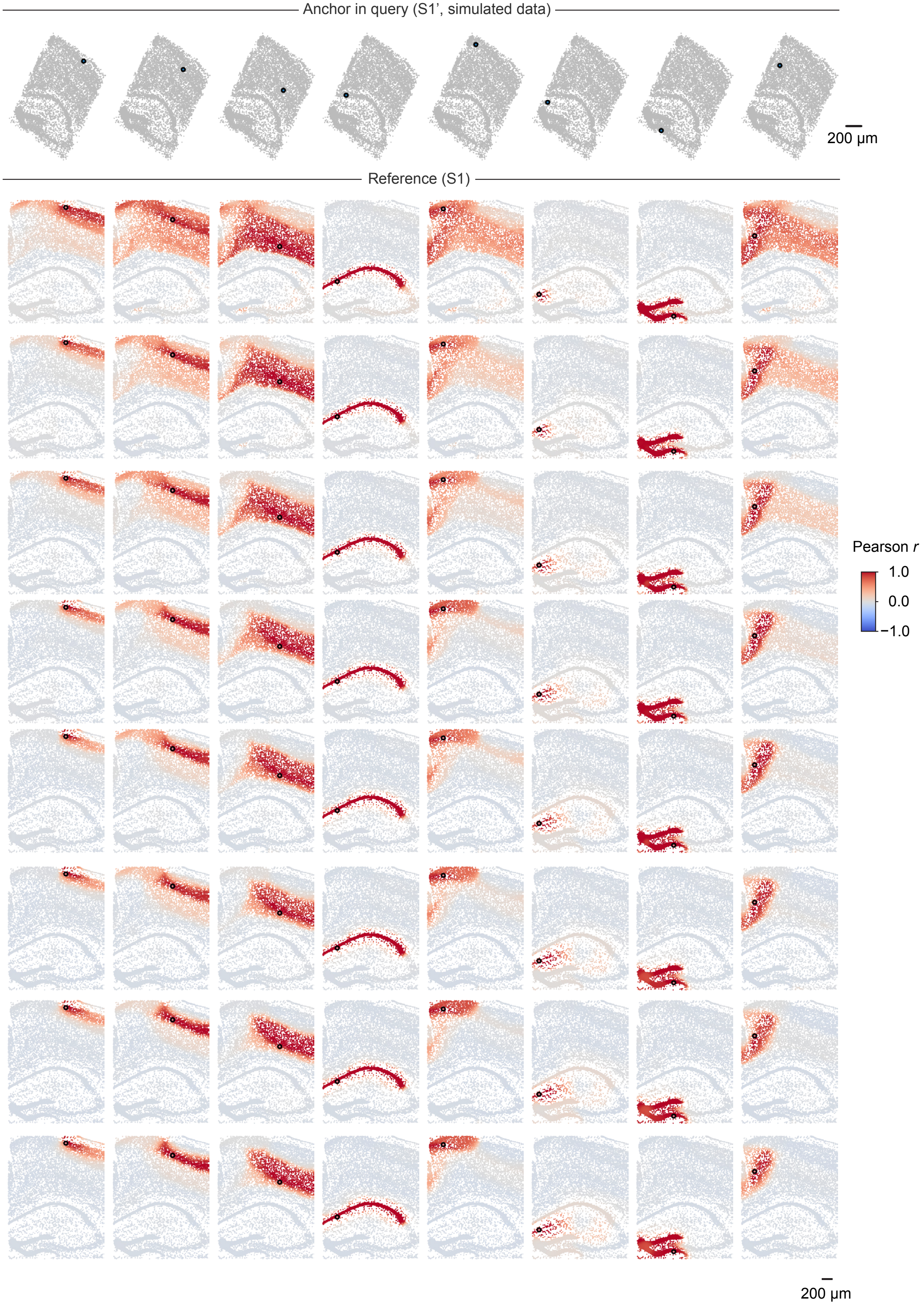
# Supplementary Figures

## Supplementary Fig. 1



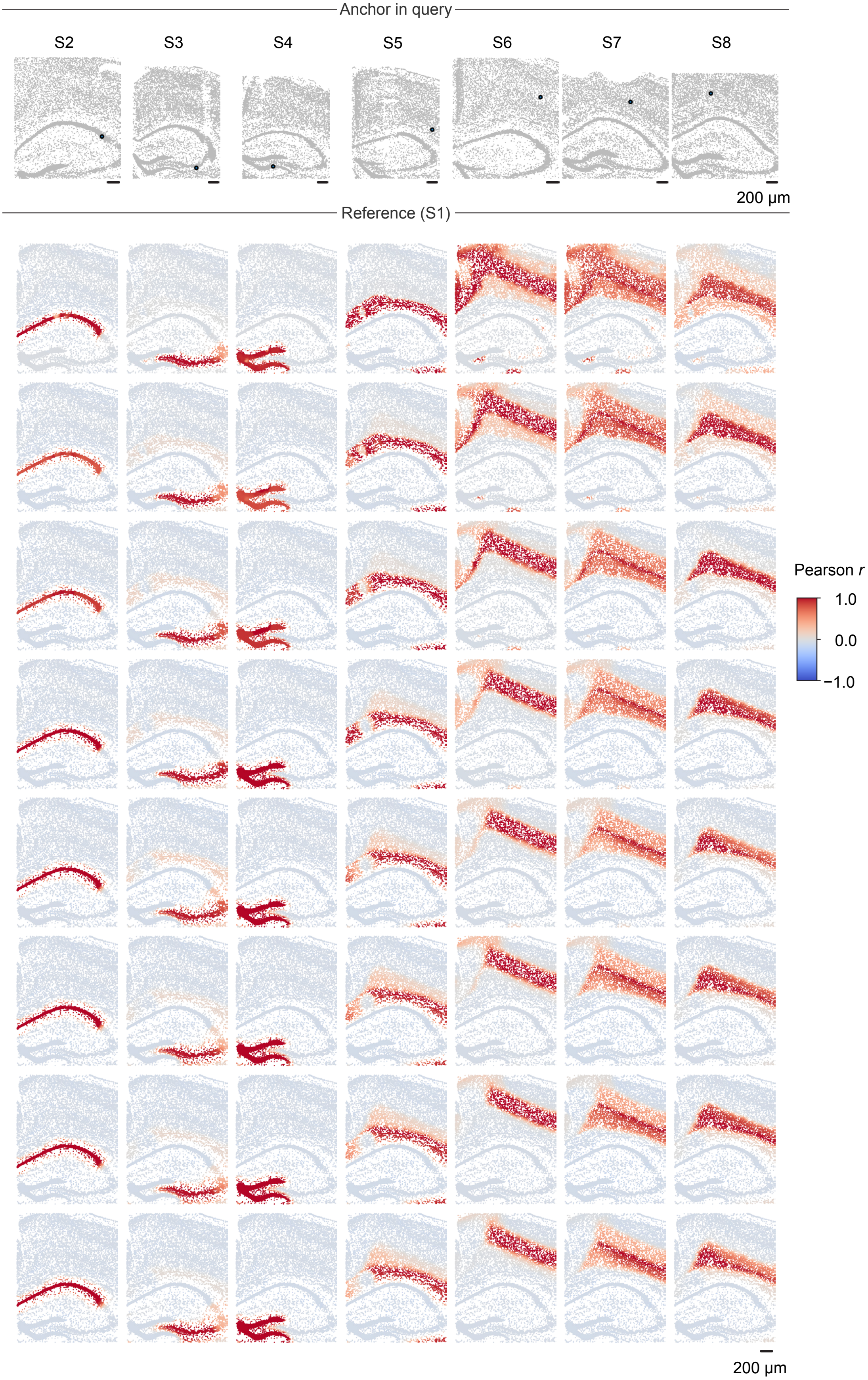
**Supplementary Fig. 1 | Parameter sensitivity and ablation studies of the CAST Mark architecture.** The median distance of each ground-truth cell pair of samples S1-S1' after alignment is used as the performance evaluation metric (sample size = 10 for each box). **a-d**, The benchmark results for different CAST Mark hyperparameters: *n_layer*, number of GCNII layers (**a**); *encoder_dim*, encoder dimension (**b**); *dfr*, feature dropout rate (**c**); *der*, edge dropout rate (**d**). **e**, Ablation studies of the graph convolution layer (blue, GCNII, the same data as in panel **a**; light blue, replacing GCNII with the vanilla GCN layers) used in CAST Mark. In the boxplots **a-e**, the middle line indicates the median; the first and third quartiles are shown by the lower and upper lines, respectively; the upper and lower whiskers extend to values not exceeding 1.5 times the IQR.

## Supplementary Fig. 2



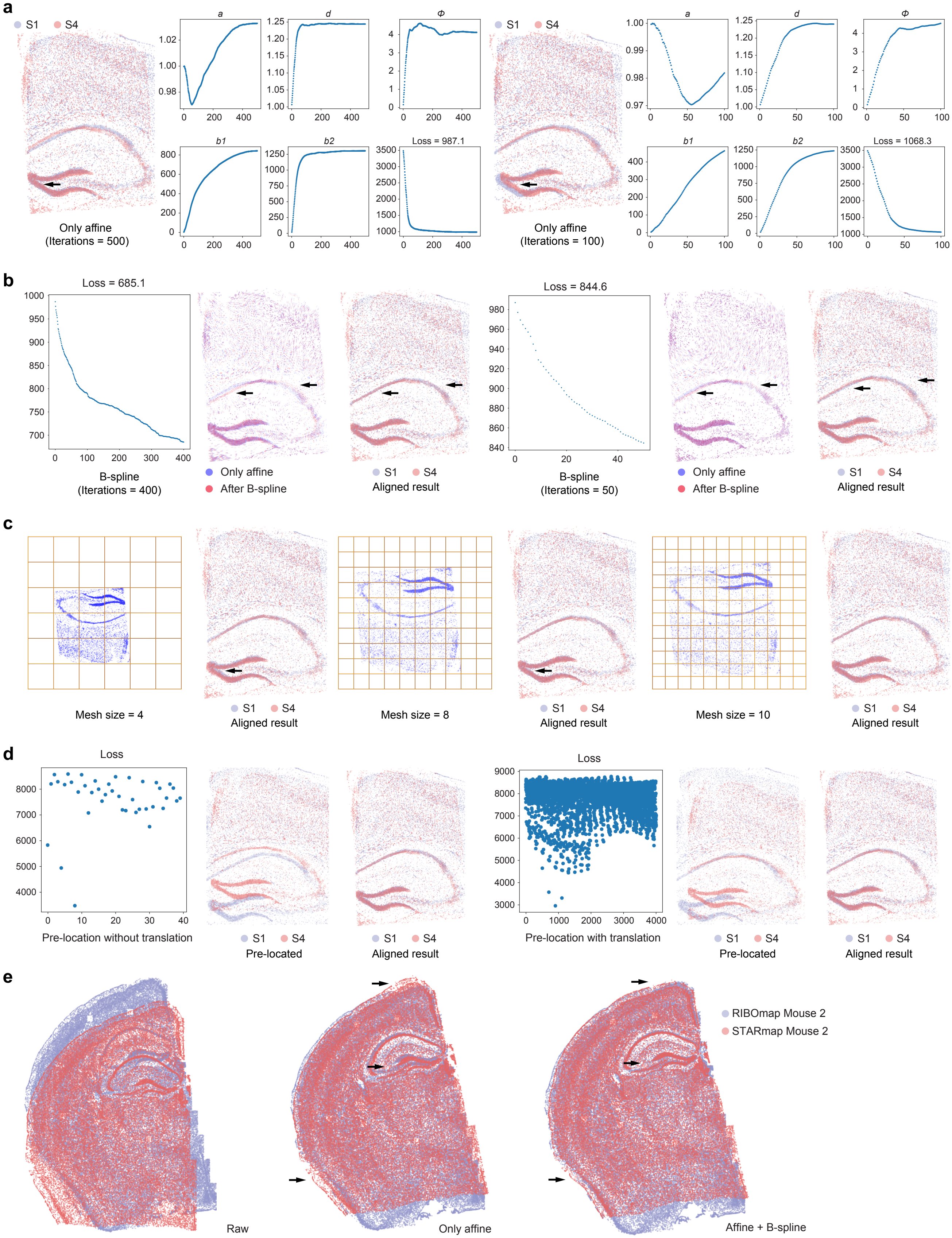
**Supplementary Fig. 2 | Spatial resolution of CAST Mark embedding with respect to the number of GCNII layers between S1 and simulated dataset S1'.** Given the query cell in the query sample (simulated dataset S1'), the cells in the reference sample (S1) are colored by Pearson correlation of the graph embedding between the reference cells and the given query cell (the same as Fig. 2c). Data shown is the mean value for 10 CAST training replicates.

## Supplementary Fig. 3



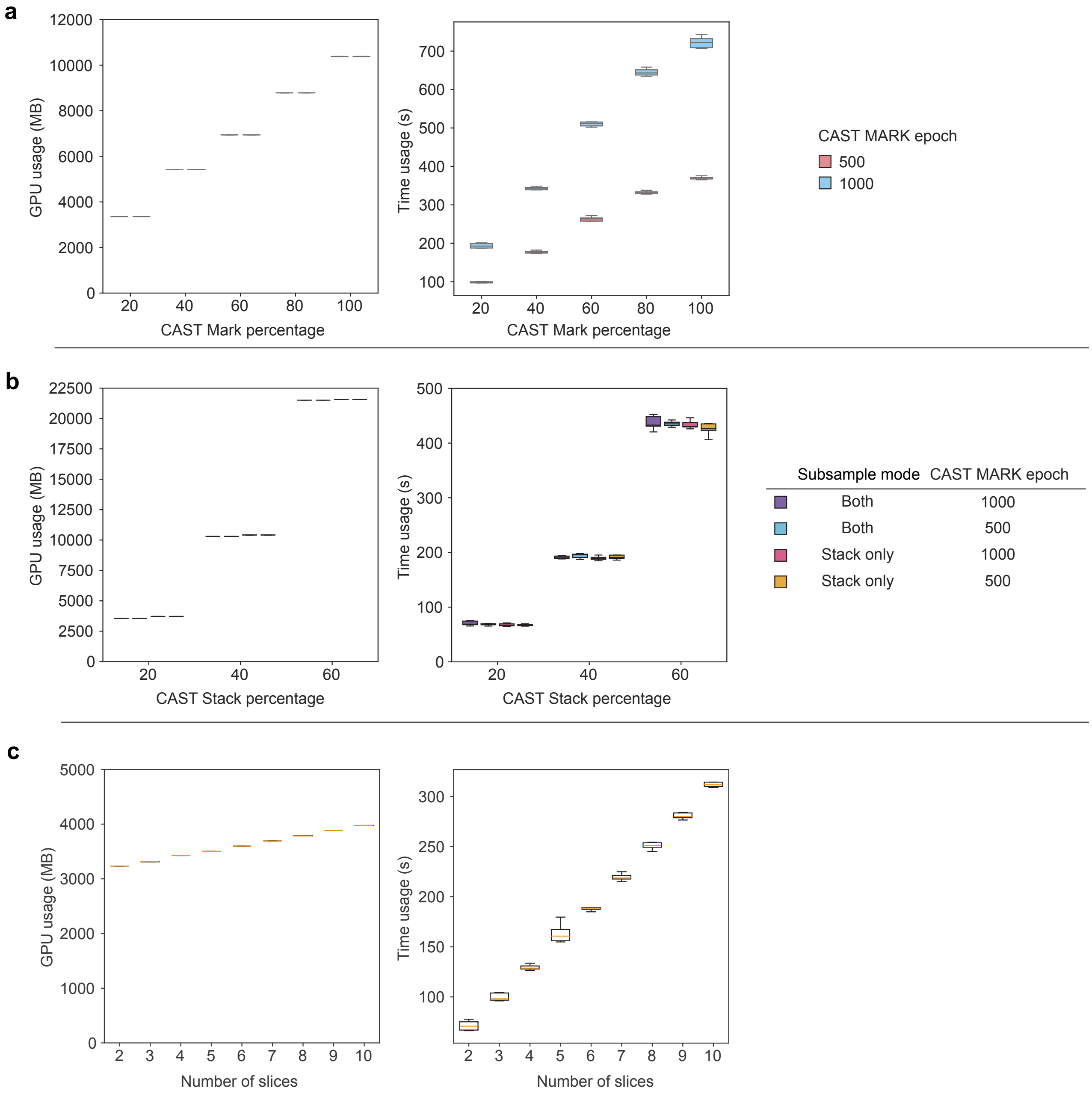
**Supplementary Fig. 3 | Spatial resolution of CAST Mark embedding with respect to the number of GCNII layers between S1 and real biological samples S2-S8.** Given the query cell in the query sample (S2-S8), the cells in the reference sample (S1) are colored by Pearson correlation of the graph embedding between the reference cells and the given query cell (the same as Fig. 2c). Data shown is the mean value for 10 CAST training replicates.

## Supplementary Fig. 4



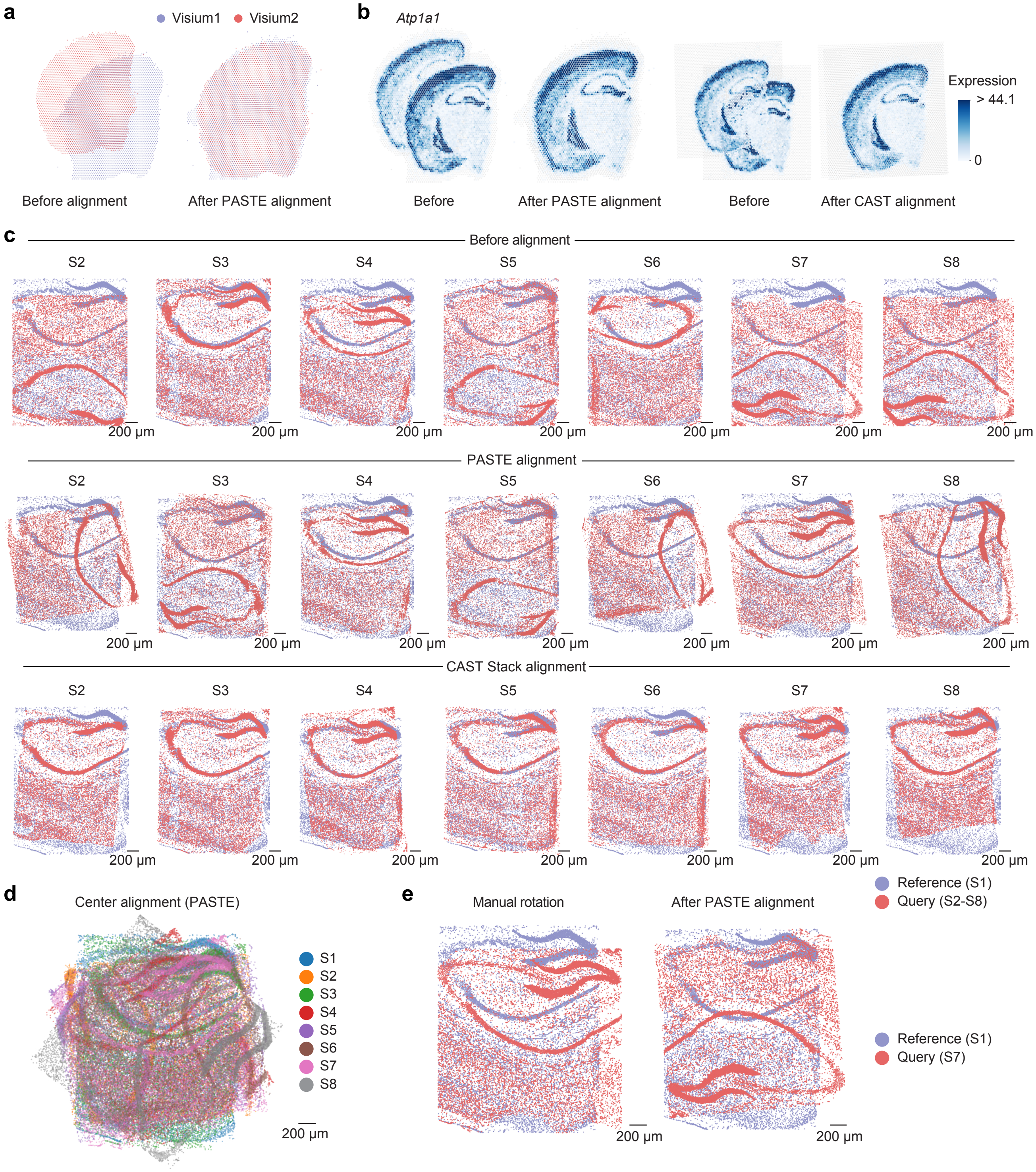
**Supplementary Fig. 4 | Benchmark of the CAST Stack initialization. a**, Different affine transformation iterations (left panel, 500 iterations; right panel, 100 iterations) were applied to align S4 to S1. The alignment results using only affine transformation are displayed, alongside the profiles of corresponding parameters observed throughout the gradient descent process. **b**, Different B-spline iterations (left panel, 400 iterations; right panel, 50 iterations) are applied to align S4 to S1. The loss function, before and after the B-spline warping are displayed, respectively. **c**, Different mesh grid settings alongside the alignment results are exhibited respectively. **d**, Results of the pre-location screening with or without translation trials are shown. Each panel shows the loss function profiles of different pre-location trials as well as the two samples before and after alignment. For visualization, the aligned results of **b** (left panel), **c** (middle panel), and **d** (left panel) are same. **e**, Alignment of STARmap_mouse2 and RIBOmap_mouse2. The results show the before alignment, after affine transformation, and following the subsequent B-spline warping.

## Supplementary Fig. 5



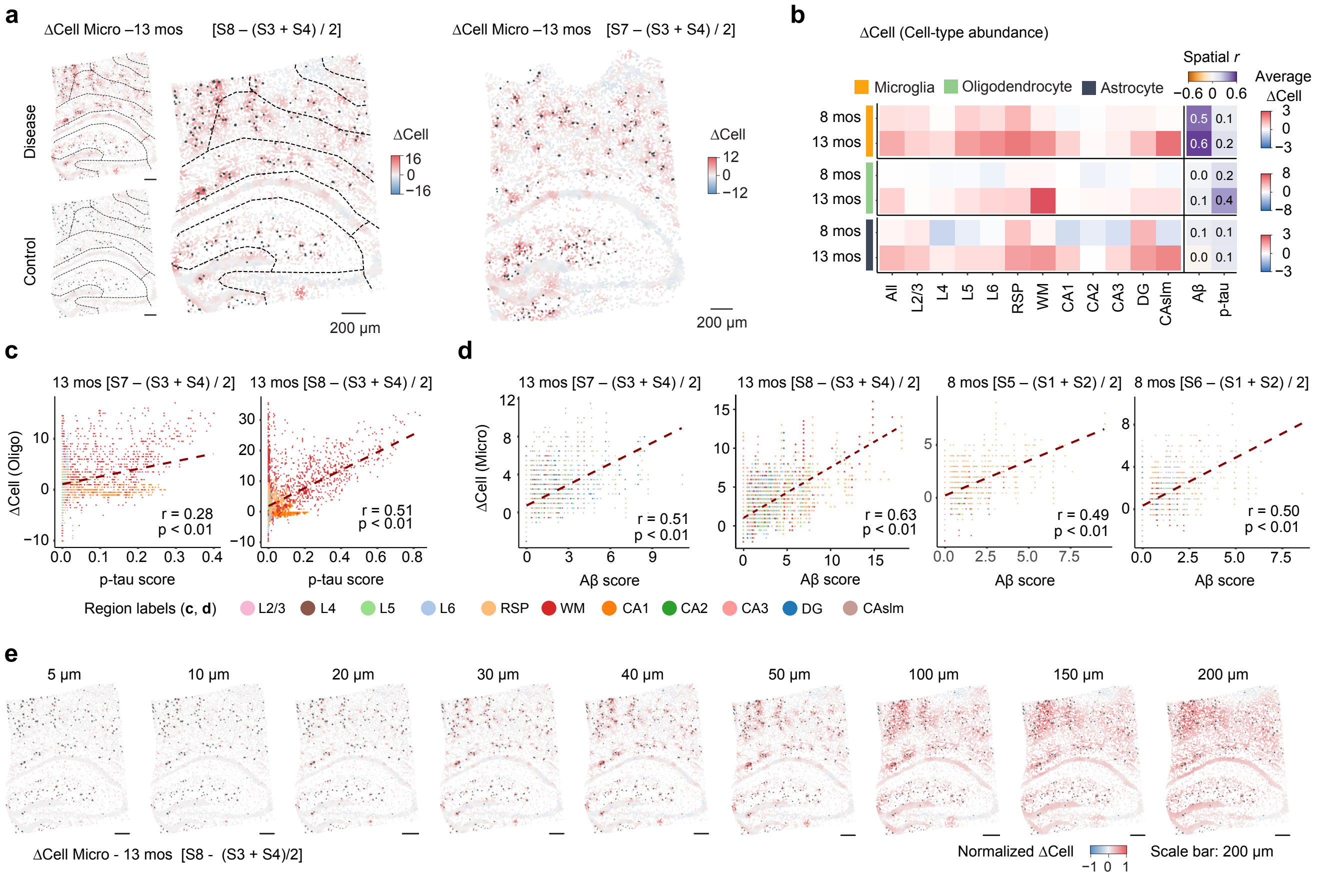
**Supplementary Fig. 5 | Efficiency of the CAST algorithm in time and space scales with the number of cells and number of slices with the subsampling strategy. a,** CAST Mark memory usage scales linearly with the number of cells per sample in subsampling while the time cost scales sublinearly. RIBOmap Mouse 2 and STARmap Mouse 2 containing 60,481 cells and 44,751 cells, respectively, were used as the training datasets for this task. **b,** CAST Stack time and memory usage scales superlinearly with the number of cells per sample in subsampling. RIBOmap Mouse 2 and STARmap Mouse 2 were used as the training datasets for this task. **c**, The memory efficiency of the CAST algorithm is robust to the number of samples. The time cost of the CAST algorithm scales linearly with the number of samples. S1 and S1' were used as the training datasets for this sample. The sample size is 10 for each box in **a-c**. The middle line indicates the median; the first and third quartiles are shown by the lower and upper lines, respectively; the upper and lower whiskers extend to values not exceeding 1.5 times the IQR.

# Supplementary Fig. 6



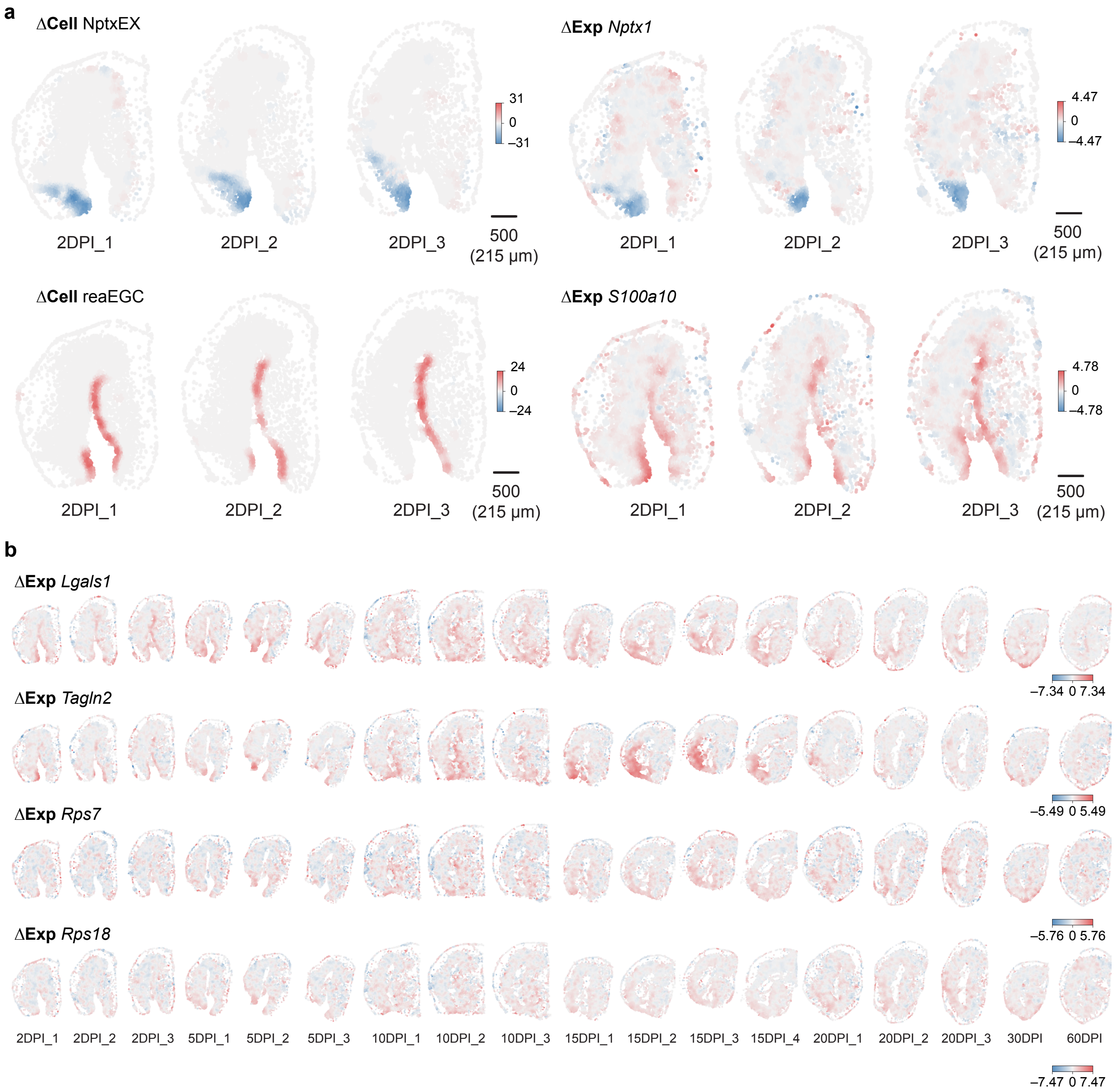
**Supplementary Fig. 6 | Benchmark of PASTE. a**, PASTE alignment from the Visium2 sample to the Visium1 sample. **b**, The *Atp1a1* raw count gene expression profile of the before and after alignment (left, PASTE; right, CAST). **c**, Benchmarking the performance of CAST and PASTE in one-to-one alignment tasks. Spatial coordinates of query (S2-S8, respectively, colored in pink) and reference samples (S1, colored in blue) are overlaid before alignment (top panel), after PASTE alignment (middle panel), or after CAST Stack alignment (bottom panel). **d**, PASTE center-align alignment results of samples S1–S8. Spatial coordinates of these samples are overlaid. Cells are colored by samples. **e**, Manual rotation preprocessing fails to improve PASTE alignment performance. Human-preprocessed spatial coordinates were used as input for PASTE alignment. S7 to S1 alignment is shown as an example. Overlaid spatial coordinates of the two samples are plotted before and after PASTE alignment, the query sample (S7) colored pink and the reference sample (S1) colored blue.

# Supplementary Fig. 7



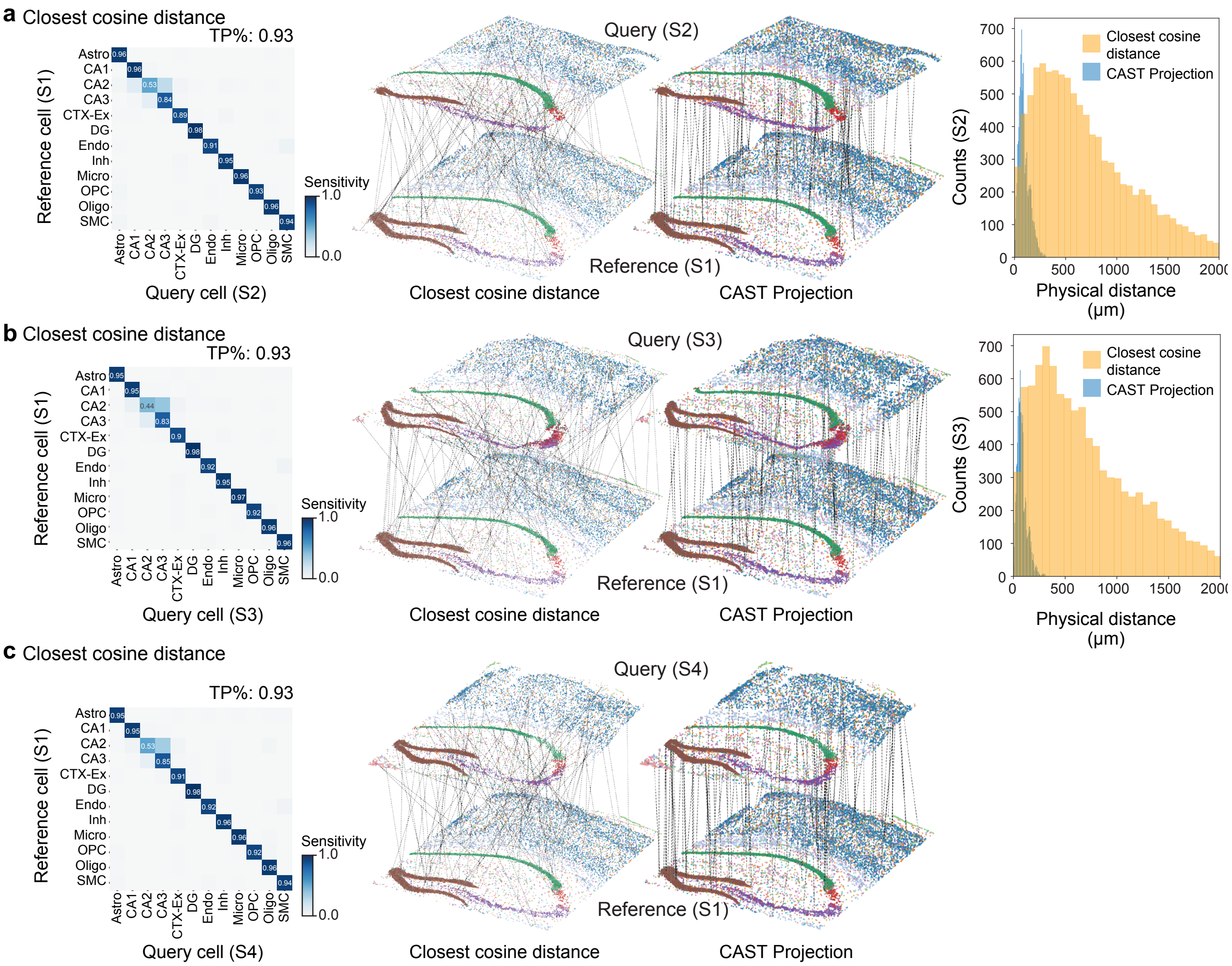
**Supplementary Fig. 7 | Delta-sample analysis (ΔAnalysis) detects spatial differences of molecular characteristics between disease and normal conditions. a**, Left panel: the spatial gradient map (S8 coordinates) shows the ΔCell pattern of the microglia in 13 mos comparison (S8 − (S3 + S4) / 2). The dark green dots represent the Aβ-plaque in the S8 sample, and the size of the dots indicates the area of the Aβ-plaque. The dashed lines indicate the different CAST Mark regions (Extended Data Fig. 5c). Similarly, the right panel is the ΔCell pattern of S7 (S7 − (S3 + S4) / 2). **b**, The average delta cell type abundance (ΔCell) in different regions, cell types and comparison groups. The average spatial correlation (Pearson *r*) between the ΔCell and the Aβ-plaque as well as p-tau scores (Methods) are displayed aside. For visualization, The values of 4 combinations in each comparison are averaged (13 mos comparison: S7 − S3, S7 − S4, S8 − S3, S8 − S4; 8 mos comparison: S5 − S1, S5 − S2, S6 − S1, S6 − S2). **c**, The scatter plot shows the significant high spatial correlation between the ΔCell (Oligodendrocyte) and the p-tau score in S7 (left panel, n = 9,634) and S8 (right panel, n = 10,372). The ΔCell values (*y* axis) are the subtraction of the disease sample and the average value of the control samples (left panel, S7 − (S3 + S4) / 2; right panel, S8 − (S3 + S4) / 2). In the Pearson correlation tests, all p-values are less than 2.2e-16. **d**, Analogous to **c**, the ΔCell (Microglia) shows significant high spatial correlation with Aβ-plaque score in 13 mos (S8 − (S3 + S4) / 2, n=10,372; S7 − (S3 + S4) / 2, n = 9,634) and 8 mos (S5 − (S1 + S2) / 2, n = 8,202; S6 − (S1 + S2) / 2, n = 8,186) comparisons. In the Pearson correlation tests, all p-values are less than 2.2e-16. **e**, Analogous to **a**, The ΔCell pattern of the microglia in 13 mos comparison (S8 − (S3 + S4) / 2) with different physical radii *R* (from 5 μm to 200 μm). For visualization, the ΔCell pattern in **a** (left) is also displayed here.

## Supplementary Fig. 8



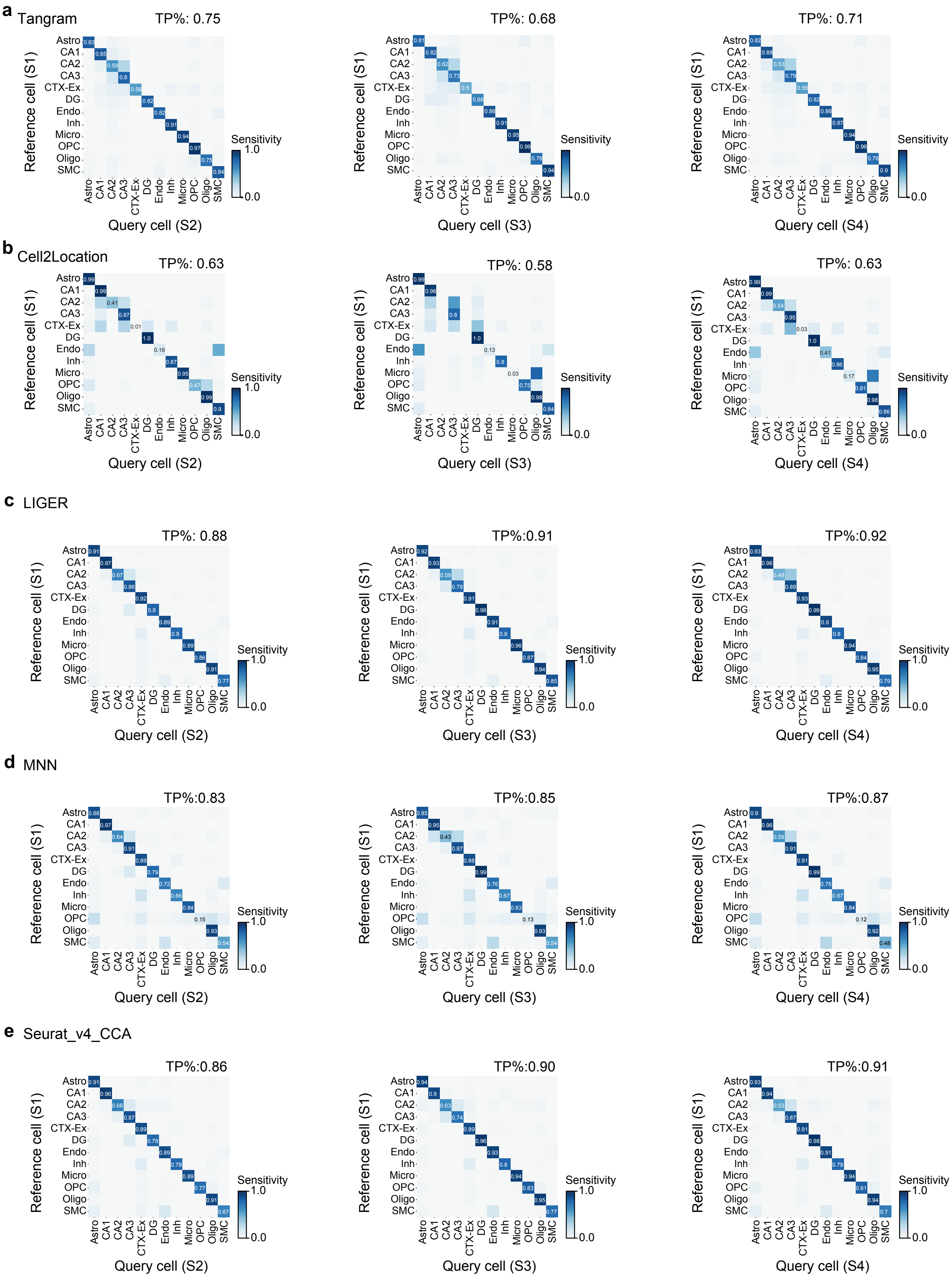
**Supplementary Fig. 8 | The ΔAnalysis delineates the spatial differences of injury-associated cell types and gene expression. a**, ΔCell results of the nptxEx (*Nptx*$^+$ lateral pallium excitatory neuron) and reaEGC (reactive ependymoglial cells), and associated ΔExp profiles (*log2_norm1e4*) of the nptxEX marker gene *Nptx1* and the reaEGC marker gene *S100a10* in 2DPI samples (2DPI_1, 2DPI_2, 2DPI_3, respectively). **b**, The ΔExp profiles (*log2_norm1e4*) of the example genes in Cluster 6 (*Lgals1* and *Tagln2*) and 9 (*Rps7* and *Rps18*) across all regeneration stages.

## Supplementary Fig. 9



**Supplementary Fig. 9 | Projecting cells without spatial constraints. a-c**, The cells in the query slice (S2, **a**; S3, **b**; S4, **c**) were projected to the cell in the reference slice (S1) with the closest cosine distance value. In each plot, the left panel is the confusion matrix of the projection (True positive rate = 0.93 for each group). In order to analyze cell types with an adequate sample size, we filtered out those with less than 10 cells in the reference sample. The middle panel is the schematic for projection results. For visualization, projection profiles of the initial CAST Projection (Fig. 5b and Extended Data Fig. 7c) are also displayed here. Dashed lines (100 randomly sampled assignment pairs for visualization) connect cells from the query sample (S2-S4) with its destination cell in the reference sample (S1). Colors represent cell types. The right panel is the distribution of the physical distance in the spatial single-cell projection results (yellow) compared with the initial CAST Projection results (blue).

## Supplementary Fig. 10



**Supplementary Fig. 10 | Benchmark of the CAST Projection algorithm. a, b**, The confusion matrix of the true label and the predicted cell type label generated by Tangram (**a**) and Cell2Location (**b**) based on the query sample (S2, S3, S4, respectively). The cell type with the highest probability is assigned as the predicted cell type label. **c-e**, CAST Projection performance using different integration methods. The confusion matrix of the true label and the predicted cell type label generated by LIGER (**c**), MNN (**d**) and Seurat (**e**, based on CCA integration of Seurat version 4) based on the query sample (S2, S3, S4, respectively). All the parameters are the same as the initial CAST Projection based on the Combat and Harmony integration.

## Reference

1. Chen, M., Wei, Z., Huang, Z., Ding, B. & Li, Y. Simple and deep graph convolutional networks. in *Proceedings of the 37th International Conference on Machine Learning* 1725–1735 (PMLR, 2020).
2. Zhang, H., Wu, Q., Yan, J., Wipf, D. & Yu, P. S. From canonical correlation analysis to self-supervised graph neural networks. in *Advances in Neural Information Processing Systems* **34**, 76–89 (2021).
3. Kipf, T. N. & Welling, M. Semi-supervised classification with graph convolutional networks. Preprint at *arXiv* https://arxiv.org/abs/1609.02907 (2016).
4. Shi, H. *et al.* Spatial atlas of the mouse central nervous system at molecular resolution. *Nature* **622**, 552–561 (2023).
5. Zeng, H. *et al.* Spatially resolved single-cell translatomics at molecular resolution. *Science* **380**, eadd3067 (2023).
6. Stuart, T. *et al.* Comprehensive integration of single-cell data. *Cell* **177**, 1888–1902.e21 (2019).
7. Welch, J. D. *et al.* Single-cell multi-omic integration compares and contrasts features of brain cell identity. *Cell* **177**, 1873–1887.e17 (2019).
8. Haghverdi, L., Lun, A. T. L., Morgan, M. D. & Marioni, J. C. Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors. *Nat. Biotechnol.* **36**, 421–427 (2018).
9. Zeng, H. *et al.* Integrative in situ mapping of single-cell transcriptional states and tissue histopathology in a mouse model of Alzheimer's disease. *Nat. Neurosci.* **26**, 430–446 (2023).
10. Busche, M. A. & Hyman, B. T. Synergy between amyloid-β and tau in Alzheimer's disease. *Nat. Neurosci.* **23**, 1183–1193 (2020).
11. Chen, W.-T. *et al.* Spatial transcriptomics and in situ sequencing to study Alzheimer's disease. *Cell* **182**, 976–991.e19 (2020).
12. Keren-Shaul, H. *et al.* A unique microglia type associated with restricting development of Alzheimer's disease. *Cell* **169**, 1276–1290.e17 (2017).
13. Hook, G. *et al.* Cathepsin B gene knockout improves behavioral deficits and reduces pathology in models of neurologic disorders. *Pharmacol. Rev.* **74**, 600–629 (2022).
14. He, Y. *et al.* ClusterMap for multi-scale clustering analysis of spatial gene expression. *Nat. Commun.* **12**, 1–13 (2021).
15. Biancalani, T. *et al.* Deep learning and alignment of spatially resolved single-cell transcriptomes with Tangram. *Nat. Methods* **18**, 1352–1362 (2021).
16. Kleshchevnikov, V. et al. Cell2location maps fine-grained cell types in spatial transcriptomics. *Nat. Biotechnol.* **40**, 661–671 (2022).