
Supplemental Material for “JADE: Joint Alignment and Deep Embedding for Multi-Slice Spatial Transcriptomics”

Anonymous Author(s)

Affiliation

Address

email

1 A Implementation details for JADE and benchmarks

2 **Data Preprocessing.** We followed the standardized preprocessing workflow implemented in the
3 SCANPY package [4] to prepare the input data for our model. For each tissue slice i , raw gene
4 expression counts were normalized by library size and log-transformed. Gene expression values were
5 then scaled to unit variance across spatial locations. The top 3,000 highly variable genes (HVGs)
6 were identified independently in each slice, and only the intersection was retained, resulting in a
7 shared gene set of size $p < 3000$ for both datasets (around 1500 for DLPFC and around 1000 for
8 axolotl brain dataset). This produced two input feature matrices: $X_1 \in \mathbb{R}^{n_1 \times p}$ and $X_2 \in \mathbb{R}^{n_2 \times p}$,
9 where n_1 and n_2 denote the number of spatial locations in slices 1 and 2, respectively.

10 **Spatial Graph Construction.** For each tissue slice i , we construct an unweighted spatial graph
11 $G_i = (V_i, E_i)$ that captures the spatial relationships among spatial locations using spatial information
12 S_i . Here, V_i represents the set of spatial locations, and E_i consists of edges connecting neighboring
13 locations based on their spatial proximity. To determine connectivity between spatial locations, we
14 compute the Euclidean distances between all pairs of locations using their spatial coordinates, then
15 establish edges based on a defined neighborhood size k , where an edge is created between spatial
16 location i and spatial location j if j is among the k nearest neighbors of i , thereby constructing a
17 graph that captures the spatial relationships among spatial locations in the tissue slice. For all real
18 data analyses, we set k to be 3. The resulting graph is mathematically represented by an adjacency
19 matrix $A_i = (a_{ij})$, where $a_{ij} = 1$ indicating a connection between spatial locations i and j , and
20 $a_{ij} = 0$ indicating no connection. After this step, we obtain two adjacency matrices of the spatial
21 graph in two slices, $A_1 \in \mathbb{R}^{n_1 \times n_1}$ and $A_2 \in \mathbb{R}^{n_2 \times n_2}$.

22 **Evaluation metrics.** In this paper, we evaluate both alignment accuracy and representation learning
23 quality across methods. To evaluate spatial alignment, we extend beyond conventional metrics that
24 only consider the proportion of correctly aligned spatial location pairs sharing the same annotation
25 (e.g., layer label) [7, 6]. While commonly used, these traditional metrics can be misleading: they
26 often favor methods that selectively align “easy” spatial location pairs while ignoring harder regions,
27 thereby inflating accuracy at the cost of coverage and interpretability. To address this limitation, we
28 adopt an alignment accuracy metric that incorporates both aligned and unaligned spatial locations,
29 offering a more robust and informative assessment. Specifically, let $\pi \in [0, 1]^{n_1 \times n_2}$ be the alignment
30 matrix where $\pi_{ij} = 1$ if spatial location i in slice 1 is mapped to spatial location j in slice 2 (or for soft
31 alignment methods like JADE). We normalize each column of π to sum up to $1/n_2$, yielding a matrix
32 $\tilde{\pi}$ that retains soft alignments while assigning zero mass to unaligned spots. Our alignment accuracy
33 is then defined as $\text{Acc} = \sum_{i,j} \tilde{\pi}_{ij} \mathbf{1}(l_i = l_j)$, where l_i and l_j denote the biological annotations (e.g.,
34 cell type, tissue layer) for spatial location i and j , respectively. This formulation rewards biologically
35 meaningful alignments—i.e., mappings between spots with consistent labels—while penalizing both
36 misalignments and omissions. When all spatial locations are perfectly aligned within the correct
37 biological regions, $\text{Acc} = 1$. Finally, to ensure symmetry, we repeat the entire procedure swapping
38 the roles of slice 1 and slice 2 and report the average of the two scores.

Beyond alignment, we evaluate the quality of the learned representations produced by each method. We use the adjusted Rand Index (ARI) to assess clustering accuracy by comparing predicted domain labels with ground-truth annotations, providing a quantitative measure of how well the representations capture underlying biological structure. To assess batch effect removal, we compute the local inverse Simpson’s index (iLISI), which quantifies the degree of slice mixing in the embedding space—higher values indicate better integration across slices. We further perform qualitative assessment using Uniform Manifold Approximation and Projection (UMAP): coloring spatial locations by their true domain labels reveals biological coherence, while coloring by slice identity visually demonstrates integration quality and batch correction performance.

Baseline methods We benchmarked our method, JADE, against a set of six representative baseline models, selected for their relevance to either alignment or embedding tasks in spatial transcriptomics. For alignment accuracy, we evaluated against PASTE [7], Seurat [3], and STAligner [8], three widely-used methods designed to align SRT slices based on transcriptomic similarity and spatial proximity. For evaluating the learned embeddings through spatial domain detection, we benchmarked against GraphST [2], STAGATE [1], and STAligner, all of which incorporates spatial or graph-based information to enhance the detection of spatial domains. Below we outline, for each benchmark method, the key preprocessing steps, parameter settings, and clustering or alignment workflows used in our comparisons. All pipelines begin from the same raw count matrices and spatial coordinate annotations.

- **JADE.** For each pairwise alignment task, we first selected the top 3,000 highly variable genes from each slice, then took the intersection of these two sets, yielding approximately 1,500 genes for every pair in DLPFC and 1000 genes for axolotl brain dataset. We then applied log-normalization to the expression matrix, and constructed a spot-by-gene feature matrix. For the DLPFC dataset, we normalized each inter-slice distance matrix by the minimum distance between any two distinct spots within the same slice. For the axolotl brain dataset, because the juvenile slice is slightly larger physically than the adult slice, we rescaled the coordinates of the adult slice to match that of the juvenile slice. We ended up multiplying the x coordinates of the adult slice by approximately 1.3 to eliminate any alignment bias induced by scaling. The data-driven approach of selecting λ_2 - λ_5 can be found in Section D. We employed a GCN with a single hidden layer to project the original expression matrix into a 64-dimensional latent space. Following [5], we set the neighborhood size to $k = 3$, which has been shown to yield strong performance. Before joint training, we pretrained the model to obtain an initial estimate of the alignment matrix Π , since the embeddings H are essentially uninformative at the start of optimization.

$$\mathcal{L}_{\text{pretrain}} = \mathcal{L}_{\text{SCL}} + \lambda_2 \mathcal{L}_{\text{recon}} + \lambda_3 \mathcal{L}_{\text{maintain}} + \lambda_4^0 \mathcal{L}_{\text{align}}^0 + \lambda_5 \mathcal{L}_{\text{marginal}}.$$

Similar to the Mis-Alignment loss $\mathcal{L}_{\text{align}}$, we define

$$\mathcal{L}_{\text{align}}^0 = \frac{1}{n_1} \|X_1 - n_2 \Pi X_2\|_F + \frac{1}{n_2} \|X_2 - n_1 \Pi^T X_1\|_F.$$

Throughout, we fixed $\lambda_4^0 = 5.0$. We used Adam optimizer with learning rate set as 0.002, number of pretraining epochs as be 200 and number of training epochs set as 800.

- **GraphST.** Following the GraphST protocol, we used the same preprocessing steps as JADE and then assembled each spot’s neighborhood by integrating within-slice 2D spatial locations. Specifically, we fixed the neighborhood size to $k = 3$ spots for each query location, which is also recommended by GraphST. All other GraphST hyperparameters were left at their package defaults. After computing the spatially informed graph, we applied the mclust Gaussian mixture model for final cluster assignment, using a smoothing radius of 20 spots in the refinement stage, which again mirroring GraphST’s recommended default.
- **STAligner.** We subjected the data to our standard HVG filtering and normalization pipeline before invoking STAligner’s built-in neighbor selection routine. For the human DLPFC slices, we set the cutoff radius to 8 spatial units, which yielded on average approximately 5 neighbors per spot; for the axolotl brain slices, we increased the cutoff to 50 units to account for differences in spot density, also achieving roughly 5 neighbors per spot. All other STAligner hyperparameters remained at their defaults. Clustering on the aligned feature space was again performed with mclust, allowing direct comparability to GraphST and JADE.

- **STAGATE.** After the shared preprocessing steps, we turned to STAGATE’s graph attention framework. We utilized its default neighbor selection within each slice, specifying a radius of 8 units for DLPFC (approximately 5 neighbor spots) and 50 units for the axolotl data (approximately 5 neighbor spots). This radius-based neighborhood ensures that each spot aggregates information from a consistent local context before passing through the graph attention layers. We retained STAGATE’s default training hyperparameters and extracted the learned embeddings for clustering via mclust.
- **Seurat.** We applied the canonical Seurat integration workflow across all four consecutive slices of each sample in DLPFC dataset. We selected the top 3,000 variable genes for integration and computed 15 principal components on the combined expression matrix.
- **PASTE.** Finally, we ran the original PASTE algorithm with its default settings. Throughout, we set $\alpha = 0.1$ as recommended. No parameter tuning was performed beyond the defaults supplied by the PASTE package.

Biological applications. To assess the biological interpretability of the learned representations, we performed downstream analysis of domain-specific gene expression. For each spatial domain identified from the embeddings, we conducted differential expression analysis using the Wilcoxon rank-sum test to identify domain-enriched marker genes. These markers were then validated against known anatomical annotations and literature-curated gene lists. This analysis demonstrates the ability of our method to recover spatially organized, functionally coherent tissue structures and to reveal biologically meaningful gene–domain associations.

B Additional results for DLPFC

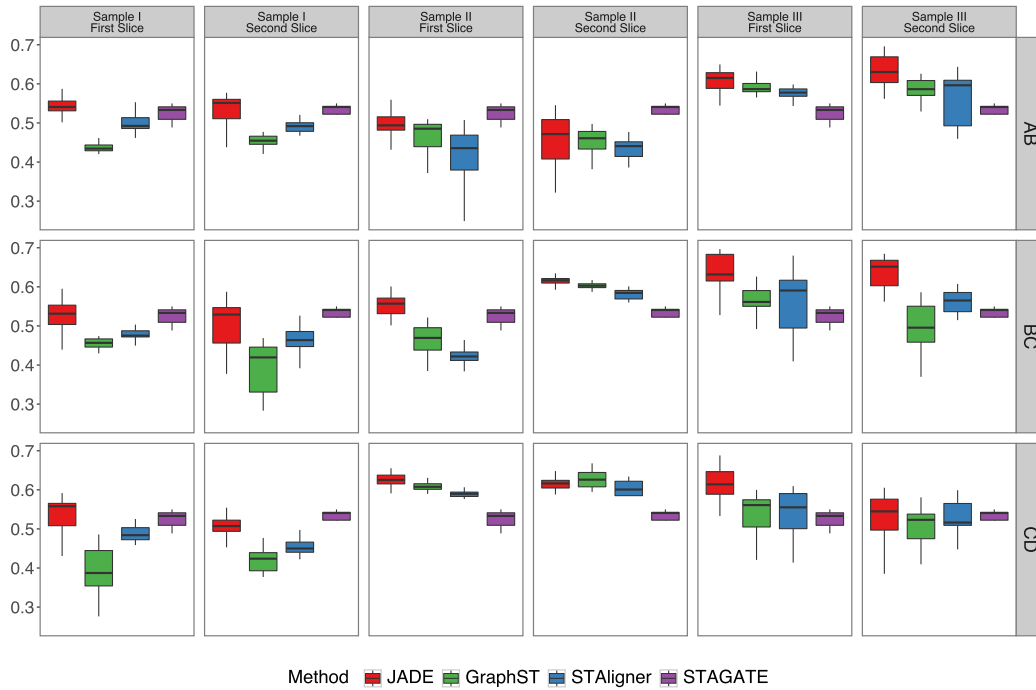


Figure 1: Adjusted Rand Index (ARI) results for three DLPFC samples (I–III, left to right), illustrating pairwise clustering of adjacent sections. Rows correspond to slice pairs AB, BC, and CD (top to bottom), and each boxplot summarizes ARI scores for both slices over 20 independent replicates.

Complementing Figure 2(F), Figure 1 presents the ARI results for all samples and their adjacent slice pairs in the DLPFC dataset. In nearly every comparison, JADE outperforms the other methods by a substantial margin. Across all three DLPFC samples (I–III) and for each adjacent-slice pairing (AB, BC, and CD), JADE not only achieves the highest median ARI but also exhibits consistently

116 tighter score distributions than the competing methods. For example, in Sample I's AB pair, JADE's
 117 median ARI is roughly 0.55, compared with about 0.45 for GraphST, 0.50 for STAligner, and 0.40 for
 118 STAGATE—an improvement of 0.05–0.15. Similarly, in Sample II's BC pairing, JADE scores near
 119 0.62, while GraphST and STAligner both cluster around 0.60 and STAGATE lags at approximately
 120 0.42. Moreover, JADE exhibits uniformly narrower interquartile ranges, particularly when compared
 121 to STAligner in Samples II and III and to STAGATE in Sample I. This indicates less variability and
 122 greater robustness to random initialization of JADE algorithm. Together, Figure 1 and Figure 2(E)
 123 highlight the satisfactory improvement achieved by JADE and underscore that joint alignment and
 124 embedding delivers more accurate and stable clustering across consecutive tissue sections.

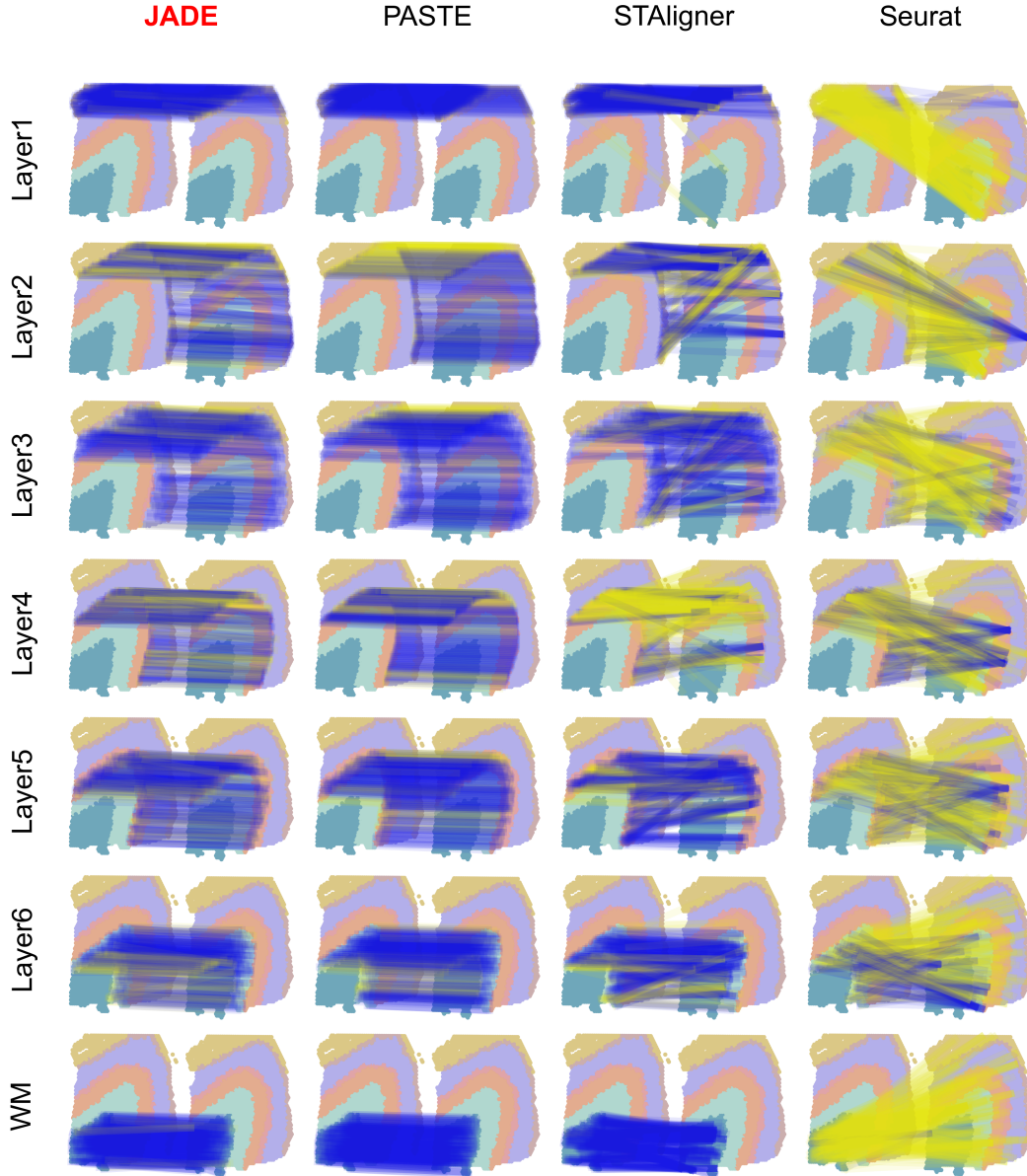


Figure 2: Illustration of alignment results from a fixed layer of Slice A to Slice B for Sample III, comparing four methods; blue lines denote correct correspondences, and yellow lines denote incorrect links.

125 Complementing Figure 2(D), Figure 2 displays the top 250 spot correspondences between a fixed
 126 layer of Slice A and Slice B for Sample III. Blue lines indicate correct matches, while yellow lines
 127 denote incorrect ones. Overall, JADE delivers superior alignment quality compared to all three other

128 benchmarks: it produces markedly smoother, less noisy mappings than STAligner or Seurat across
 129 all seven layers, and it achieves higher correspondence accuracy than PASTE with only a minimal
 130 trade-off in smoothness.

Table 1: Accuracy of top 250 alignment correspondences by domain.

Method	Layer1	Layer2	Layer3	Layer4	Layer5	Layer6	WM
JADE	0.996	0.636	0.908	0.776	0.912	0.752	1.000
PASTE	1.000	0.584	0.840	0.724	0.848	0.836	1.000
STAligner	0.936	0.648	0.832	0.252	0.752	0.740	0.992
Seurat	0.016	0.084	0.136	0.296	0.276	0.344	0.020

Table 2: Fraction of unaligned cells per domain for each alignment method.

Method	Layer1	Layer2	Layer3	Layer4	Layer5	Layer6	WM
JADE	0.000	0.000	0.000	0.000	0.000	0.000	0.000
PASTE	0.000	0.000	0.000	0.000	0.000	0.000	0.000
STAligner	0.739	0.683	0.709	0.818	0.738	0.705	0.790
Seurat	0.953	0.960	0.932	0.862	0.889	0.818	0.984

131 In Table 1, we report, for each spatial domain within DLPFC, the alignment accuracy achieved by
 132 the top 250 correspondence links when matching Slice A to Slice B in Sample III. Each entry shows
 133 the proportion of correctly paired spots among the first 250 alignments for that domain, providing a
 134 domain-specific assessment of alignment performance. JADE emerges as the most reliable method,
 135 achieving nearly perfect accuracy (1.00) in Layer 1 and WM and maintaining at least 0.63 accuracy
 136 in every layer. PASTE also attains perfect matches in Layer 1 and WM but shows greater fluctuation
 137 in intermediate layers (ranging from 0.584 to 0.848). STAligner also delivers competitive results in
 138 early layers—0.936 in Layer 1 and 0.648 in Layer 2—but its performance drops markedly in deeper
 139 regions (as low as 0.252 in Layer 4), although it still nearly reaches perfection in WM (0.992). In
 140 contrast, Seurat fails to produce meaningful alignments across all domains, with accuracy scores
 141 below 0.35 in every layer (0.016–0.344). Table 2 reports, for each cortical layer (Layers 1–6) and
 142 white matter (WM), the fraction of spots left unaligned by each method. Both JADE and PASTE
 143 achieve a fully dense correspondence matrix—no spot remains unaligned in any domain. In contrast,
 144 STAligner fails to align a substantial fraction of spots (68–82% across layers, and 79% in WM), while
 145 Seurat leaves even more spots unmatched (93–96% in Layers 1–3, tapering to 81–82% in Layers 6
 146 and 4–5, and 98% in WM). These results highlight that only JADE and PASTE guarantee fine-grained
 147 spot-spot alignment, whereas STAligner and Seurat produce large numbers of unaligned spots.

148 C Additional results for axolotl brain dataset

149 Complementing Figure 3(D), Figure 3 displays the top spot correspondences between a fixed layer of
 150 juvenile slice to adult slice. The number of correspondences is determined by half of the corresponding
 151 spots. Blue lines indicate correct matches, while gray lines denote incorrect ones. Similar to DLPFC,
 152 JADE delivers superior alignment quality compared to all three other benchmarks overall.

153 Table 3 compares, for each common cell type between juvenile and adult slices, the accuracy of the
 154 top correspondences and the average fraction of spots left unaligned. Across all cell types, JADE
 155 attains the highest or near-highest top-link accuracy, with values ranging from 0.105 (sstIN) up to
 156 0.766 (nptxEX). PASTE closely follows JADE—in several cases enjoying the highest accuracy for
 157 dpEX—but generally trails by 5–10 percentage points. While STAligner exceeds JADE on individual
 158 domains (e.g. scgnIN: 0.778 vs. 0.722; sfrpEGC: 0.171 vs. 0.145), JADE remains close to STAligner
 159 in some cases (e.g. scgnIN, sfrpEGC). However, STAligner’s average fraction of unaligned spots is
 160 0.724, meaning over 70 % of cells remain unaligned. Seurat performs poorly throughout, with top-link
 161 accuracies below 0.10 in every domain and an average of 0.925 unaligned spots. Taken together, these
 162 results underscore that although STAligner can rival JADE in top-ranked matches for certain cell types,
 163 its high rate of unaligned spots prevents a truly fine-grained, cell-to-cell alignment. In contrast, JADE
 164 (and PASTE) deliver both high accuracy and cell-cell alignment, ensuring no cell is left unmatched.

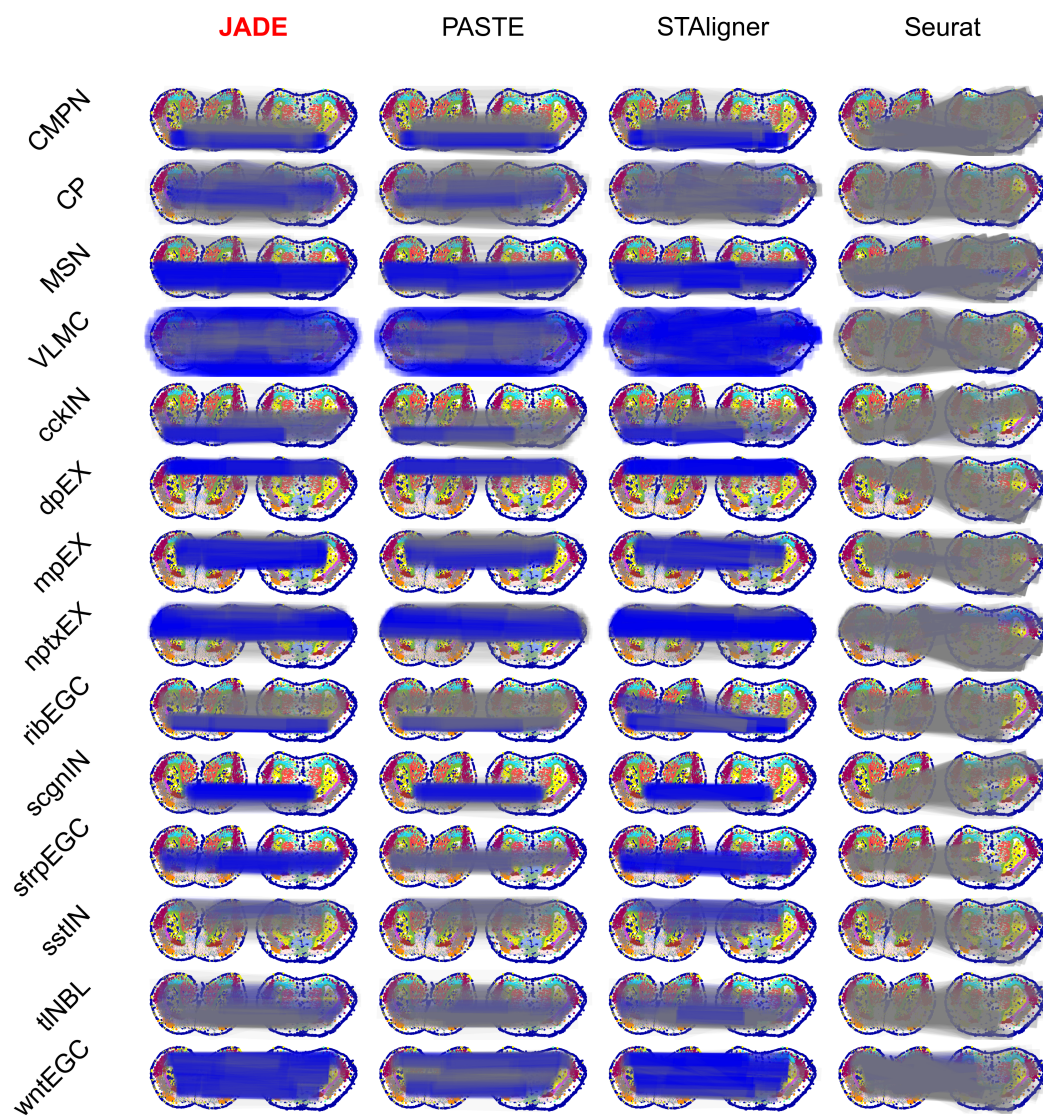


Figure 3: Alignment from a given cell type in juvenile slice to the adult slice, comparing four methods (JADE, PASTE, STAligner, and Seurat). Blue lines denote correct correspondences, while grey lines denote incorrect links.

Table 3: Alignment accuracy of the top correspondences and average fraction of unaligned spots for each common cell type between juvenile and adult slices. For each cell type, the number of correspondences equals half of its total cells.

Domain	# cells	JADE	PASTE	STAligner	Seurat
CMPN	782	0.357	0.288	0.448	0.012
CP	1500	0.224	0.157	0.121	0.069
MSN	54	0.685	0.704	0.741	0.037
VLMC	523	0.585	0.539	0.772	0.011
cckIN	99	0.211	0.169	0.237	0.019
dpEX	456	0.809	0.568	0.721	0.011
mpEX	464	0.623	0.394	0.591	0.012
nptxEX	840	0.751	0.637	0.744	0.031
ribEGC	187	0.337	0.096	0.690	0.011
scgnIN	713	0.421	0.445	0.372	0.059
strpEGC	349	0.630	0.166	0.745	0.011
sstIN	294	0.109	0.075	0.306	0.003
tlNBL	257	0.183	0.105	0.237	0.000
wntEGC	509	0.658	0.326	0.633	0.069
<i>Avg. frac. unaligned</i>	—	0.000	0.000	0.724	0.925

JADE (and PASTE) combine high top-link accuracy with exhaustive alignment—leaving none of spots unaligned—making them the only methods to guarantee both precision and completeness of correspondences.

D Hyperparameter tuning and sensitivity analysis

The training objective of JADE is

$$\underbrace{\mathcal{L}_{\text{SCL}} + \lambda_2 \mathcal{L}_{\text{recon}}}_{\text{single slice loss}} + \underbrace{\lambda_3 \mathcal{L}_{\text{maintain}} + \lambda_4 \mathcal{L}_{\text{align}}}_{\text{alignment loss}} + \lambda_5 \mathcal{L}_{\text{marginal}}.$$

The overall loss can be decomposed into three terms: the single-slice reconstruction loss, the alignment loss, and a regularization term. Throughout, we fix $\lambda_2 = 10$ and $\lambda_5 = 1$ and choose a neighborhood size of $k = 3$. The hyperparameters λ_3, λ_4 govern the trade-off between enforcing accurate slice-to-slice alignment and preserving meaningful, slice-specific embeddings. Within the alignment loss, $\mathcal{L}_{\text{maintain}}$ acts as a regularization term or prior belief that encodes our prior belief in the spatial coordinate similarity between the two slices. For the DLPFC dataset, we normalize each inter-slice distance matrix by the minimum distance between any two distinct spots within the same slice. We recommend using $\lambda_4 = 0.1$ and $\lambda_3 = 2.0$ by default, except in two cases—where the slices are a priori less similar—in which we set $\lambda_3 = 0.2$. To quantify slice-to-slice similarity—and thereby guide hyperparameter selection—we define the mini-max distance, which measures the average feature mismatch between corresponding spots after neighborhood smoothing. Specifically, suppose we have two slices with n_1 and n_2 spots. We first perform joint PCA and reduce the normalized gene expression matrices into low-dimensional features $Z_1 \in \mathbb{R}^{n_1 \times d}$ and $Z_2 \in \mathbb{R}^{n_2 \times d}$. Denote the adjacency matrices of these two slices by A_1 and A_2 , respectively. We define the mini-max measure as follows:

$$\text{Mini-max}(Z_1, Z_2) = \frac{\text{quantile} \{ \min_{1 \leq j \leq n_2} \|(A_1 Z_1)_i - (A_2 Z_2)_j\| : 1 \leq i \leq n_1, 0.99 \}}{\sqrt{\text{mean}_i \|(A_1 Z_1)_i\| \times \text{mean}_j \|(A_2 Z_2)_j\|}},$$

where the inner minimum is taken row-wise over the first slice’s smoothed features, quantifying the distance between each spot of the first slice and its closest counterpart on the second slice. The mini-max distance is then defined as the 99%th-largest quantile among these minimal distances. Table 4 reports the mini-max distances for three samples across their adjacent slice pairs (AB, BC, CD). We note that the BC pairs in Sample I and Sample II have substantially larger distances than the rest. For these two case, we set $\lambda_3 = 0.2$. Accordingly, for all other cases we set $\lambda_3 = 2.0$ to promote joint learning between slices. Figure 4 and Figure 5 present sensitivity analyses for the hyperparameters λ_3 and λ_4 , respectively. Both the adjusted Rand index and the alignment score

Table 4: Mini-max distances quantifying slice-to-slice similarity for three samples.

Sample	AB	BC	CD
Sample I	0.2726	0.5670	0.2609
Sample II	0.2958	0.4713	0.2764
Sample III	0.2439	0.2127	0.2368

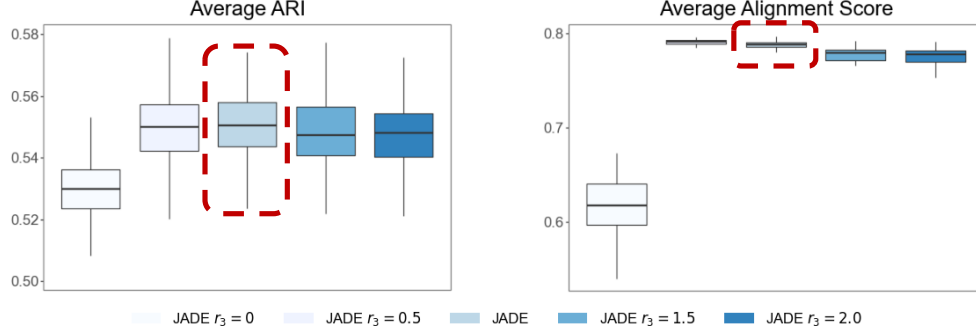


Figure 4: Sensitivity analysis: Varying λ_3 , where r_3 is the magnifier of λ_3 against JADE. Each boxplot summarizes the outcomes of 100 independent replications.

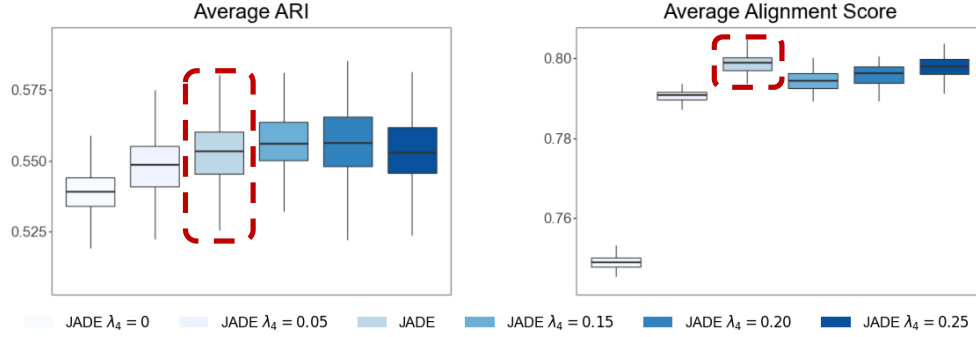


Figure 5: Sensitivity analysis: Varying λ_4 . Each boxplot summarizes the outcomes of 100 independent replications.

193 remain stable over a wide range of values: r_3 from 0.5 to 2.0; λ_4 from 0.05 to 0.25, deteriorating
194 only when either hyperparameter is set to very low levels.

195 E Ablation study

196 Figure 6 compares JADE to versions with no mismatch loss ($\lambda_3 = 0$) and no misalignment loss
197 ($\lambda_4 = 0$). Results show that performance deteriorates markedly when either loss term is omitted,
198 demonstrating that both the misalignment and mismatch losses are crucial for optimal embedding
199 and alignment quality. Setting $\lambda_4 = 0$ causes the ARI to drop from 0.55 to 0.53 (a 4% decrease) and
200 the alignment score to fall from 0.8 to 0.7 (over a 10% decrease), underscoring the critical role of the
201 misalignment loss. Setting $\lambda_3 = 0$ causes the ARI to drop from 0.55 to 0.525 (a 5% decrease) and
202 the alignment score to fall from 0.8 to 0.62 (about 20% decrease), underscoring the critical role of the
203 mismatch loss.

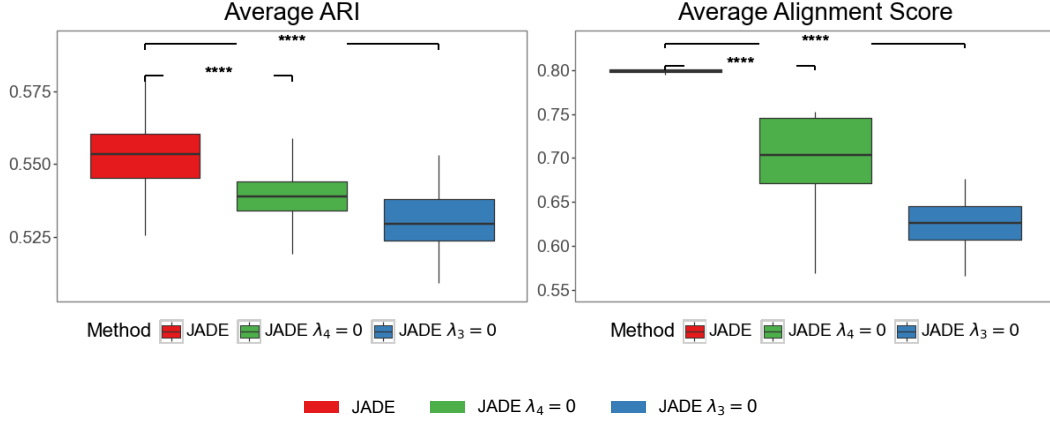


Figure 6: Ablation study: each boxplot summarizes the outcomes of 100 independent replications. We calculate the p -value using the Wilcoxon-rank sum test. **** stands for p -value lower than 0.01%

204 F Pseudo code of JADE

Input: Slices (A_i, X_i) and within slice distance matrix D_i for $i = 1, 2$; hyperparameters $\lambda_2, \dots, \lambda_5$; epochs T ; dim d ; number of Sinkhorn iterations s .

Output: Alignment $\Pi \in \mathbb{R}^{n_1 \times n_2}$, embeddings $H_i \in \mathbb{R}^{n_i \times d}$.

Initialize $\{W_{ie}, b_{ie}, W_{id}, b_{id}, M, \Phi\}$

for $epoch = 1$ **to** T **do**

1. Encode-Decode

Compute $\tilde{A}_i = D_i^{-\frac{1}{2}} A_i D_i^{-\frac{1}{2}}$, $H_i = \text{ReLU}(\tilde{A}_i X_i W_{ie} + b_{ie})$,
 $\hat{X}_i = \text{ReLU}(\tilde{A}_i H_i W_{id} + b_{id})$, $\mathcal{L}_{\text{recon}} = \frac{1}{n_1} \|X_1 - \hat{X}_1\|_F^2 + \frac{1}{n_2} \|X_2 - \hat{X}_2\|_F^2$.

2. Alignment via Sinkhorn iterations

Compute $C = \text{Softmax}(-H_1^\top M M^\top H_2 / \sqrt{d})$ and get Π by s Sinkhorn iterations and

calculate the marginal loss $\mathcal{L}_{\text{marginal}} = \text{KL}(\frac{1_{n_1}}{n_1} \Pi \| \frac{1_{n_2}}{n_2})$.

3. Mis-Alignment and Mis-Maintain loss

Compute $\mathcal{L}_{\text{maintain}} = \frac{1}{n_1} \|D_1 - n_2^2 \Pi D_2 \Pi^\top\|_F + \frac{1}{n_2} \|D_2 - n_1^2 \Pi^\top D_1 \Pi\|_F$, and
 $\mathcal{L}_{\text{align}} = \frac{1}{n_1} \|H_1 - n_2 \Pi H_2\|_F + \frac{1}{n_2} \|H_2 - n_1 \Pi^\top H_1\|_F$.

4. Contrastive loss

Compute $\mathcal{L}_{\text{SCL}} = \sum_{i=1}^2 -\frac{1}{n_i} \sum_{j=1}^{n_i} [\log \Phi(h_{ij}, r_{ij}) + \log(1 - \Phi(h'_{ij}, r'_{ij}))]$.

5. Update $\mathcal{L}_{\text{tot}} = \mathcal{L}_{\text{recon}} + \lambda_2 \mathcal{L}_{\text{SCL}} + \lambda_3 \mathcal{L}_{\text{maintain}} + \lambda_4 \mathcal{L}_{\text{align}} + \lambda_5 \mathcal{L}_{\text{marginal}}$
optimizer.step($\nabla \mathcal{L}_{\text{tot}}$).

end

return Π, H_1, H_2

Algorithm 1: JADE

Input: Slices (A_i, X_i) and within slice distance matrix D_i for $i = 1, 2$; hyperspot counts m_i ; spot-hyperspot assignments N^i ; hyperparameters $\lambda_2, \dots, \lambda_5$; epochs T ; dim d ; number of Sinkhorn iterations s .

Output: Alignment $\Pi \in \mathbb{R}^{n_1 \times n_2}$, embeddings $H_i \in \mathbb{R}^{n_i \times d}$.

Initialize $\{W_{ie}, b_{ie}, W_{id}, b_{id}, M, \Phi\}$

for $epoch = 1$ **to** T **do**

1. Encode-Decode

 Compute $\tilde{A}_i = D_i^{-\frac{1}{2}} A_i D_i^{-\frac{1}{2}}$, $H_i = \text{ReLU}(\tilde{A}_i X_i W_{ie} + b_{ie})$,

$\hat{X}_i = \text{ReLU}(\tilde{A}_i H_i W_{id} + b_{id})$, $\mathcal{L}_{\text{recon}} = \frac{1}{n_1} \|X_1 - \hat{X}_1\|_F^2 + \frac{1}{n_2} \|X_2 - \hat{X}_2\|_F^2$.

2. Hyperspot embeddings

 Compute $H_i^{\text{hyper}}[l] = \frac{1}{|N_i^l|} \sum_{j \in N_i^l} h_j^i$.

3. Alignment via Sinkhorn iterations

 Compute $C^0 = \text{Softmax}(-H_1^{\text{hyper}\top} M M^\top H_2^{\text{hyper}} / \sqrt{d})$, get Π^0 by s Sinkhorn iterations

$\mathcal{L}_{\text{marginal}} = \text{KL}(\frac{1}{m_1} \Pi^0 \| \frac{1}{m_2})$.

4. Mis-Alignment and Mis-Maintain loss

 Compute

$\mathcal{L}_{\text{maintain}} = \frac{1}{m_1} \|D_1^{\text{hyper}} - m_2^2 \Pi^0 D_2^{\text{hyper}} \Pi^{0,\top}\|_F + \frac{1}{m_2} \|D_2^{\text{hyper}} - m_1^2 \Pi^{0,\top} D_1^{\text{hyper}} \Pi^0\|_F$,

 and $\mathcal{L}_{\text{align}} = \frac{1}{m_1} \|H_1^{\text{hyper}} - m_2 \Pi^0 H_2^{\text{hyper}}\|_F + \frac{1}{m_2} \|H_2^{\text{hyper}} - m_1 \Pi^{0,\top} H_1^{\text{hyper}}\|_F$.

5. Contrastive loss

 Compute $\mathcal{L}_{\text{SCL}} = \sum_{i=1}^2 -\frac{1}{n_i} \sum_{j=1}^{n_i} [\log \Phi(h_{ij}, r_{ij}) + \log(1 - \Phi(h'_{ij}, r'_{ij}))]$.

6. Update $\mathcal{L}_{\text{tot}} = \mathcal{L}_{\text{recon}} + \lambda_2 \mathcal{L}_{\text{SCL}} + \lambda_3 \mathcal{L}_{\text{maintain}} + \lambda_4 \mathcal{L}_{\text{align}} + \lambda_5 \mathcal{L}_{\text{marginal}}$
 optimizer.step($\nabla \mathcal{L}_{\text{tot}}$).

end

Recover Π : Given trained M , compute Π as in Section 2 and Section 3.

return Π, H_1, H_2

Algorithm 2: JADE

We present the self-contained pseudo code for JADE and its accelerated version Fast-JADE in Algorithm 1 and Algorithm 2 respectively.

G Analysis of JADE and Fast-JADE

Table 5: Runtime comparison of JADE and Fast-JADE (in seconds).

Method	DLPFC	Axolotl Brain
JADE	1549	3602
Fast-JADE (with 2000 hyperspots)	780	2305
Fast-JADE (with 1000 hyperspots)	310	1201

Table 6: Average ARI comparison of JADE and Fast-JADE in DLPFC.

Method	DLPFC
JADE	0.551 (0.002)
Fast-JADE (with 1000 hyperspots)	0.536 (0.001)

Table 5 and 6 present the runtime and average ARI for JADE and Fast-JADE, the accelerated version of JADE using hyperspots as introduced in Section 3. We see that as the runtime reduce significantly when using Fast-JADE instead of the standard version of JADE while the ARI remains almost the same.

References

- [1] Kangning Dong and Shihua Zhang. Deciphering spatial domains from spatially resolved transcriptomics with an adaptive graph attention auto-encoder. *Nature Communications*, 13(1):1739, 2022. doi: 10.1038/s41467-022-29439-6.

- 218 [2] Yahui Long, Kok Siong Ang, Mengwei Li, Kian Long Kelvin Chong, Raman Sethi, Chengwei
219 Zhong, Hang Xu, Zhiwei Ong, Karishma Sachaphibulkij, Ao Chen, Li Zeng, Huazhu Fu, Min Wu,
220 Lina Hsiu Kim Lim, Longqi Liu, and Jinmiao Chen. Spatially informed clustering, integration,
221 and deconvolution of spatial transcriptomics with graphst. *Nature Communications*, 14(1):1155,
222 2023. doi: 10.1038/s41467-023-36796-3.
- 223 [3] Tim Stuart, Andrew Butler, Paul Hoffman, Christoph Hafemeister, Efthymia Papalexi, William
224 M. III Mauck, Yuhao Hao, Marlon Stoeckius, Peter Smibert, and Rahul Satija. Comprehensive
225 integration of single-cell data. *Cell*, 177(7):1888–1902.e21, 2019. doi: 10.1016/j.cell.2019.05.
226 031.
- 227 [4] Fabian Alexander Wolf, Philipp Angerer, and Fabian J Theis. Scanpy: large-scale single-cell
228 gene expression data analysis. *Genome biology*, 19(1):15, 2018.
- 229 [5] Junlin Xu, Jielin Xu, Yajie Meng, Changcheng Lu, Lijun Cai, Xiangxiang Zeng, Ruth Nussinov,
230 and Feixiong Cheng. Graph embedding and gaussian mixture variational autoencoder network
231 for end-to-end analysis of single-cell rna sequencing data. *Cell Reports Methods*, 3(1):100382,
232 2023. doi: 10.1016/j.crmeth.2022.100382.
- 233 [6] Zhiyuan Yuan, Fangyuan Zhao, Senlin Lin, et al. Benchmarking spatial clustering methods
234 with spatially resolved transcriptomics data. *Nature Methods*, 21:712–722, 2024. doi: 10.1038/
235 s41592-024-02215-8.
- 236 [7] Ron Zeira, Max Land, Alexander Strzalkowski, and Benjamin J Raphael. Alignment and
237 integration of spatial transcriptomics data. *Nature Methods*, 19(5):567–575, 2022.
- 238 [8] Xiang Zhou, Kangning Dong, and Shihua Zhang. Integrating spatial transcriptomics data across
239 different conditions, technologies and developmental stages. *Nature Computational Science*, 3
240 (10):894–906, 2023. doi: 10.1038/s43588-023-00528-w.