

# QUEST: QUERYING FUNCTIONAL AND STRUCTURAL NICHES ON SPATIAL TRANSCRIPTOMICS DATA VIA CONTRASTIVE SUBGRAPH EMBEDDING

**Anonymous authors**

Paper under double-blind review

## ABSTRACT

The functional or structural spatial regions within tissues, referred to as spatial niches, are elements for illustrating the spatial contexts of multicellular organisms. A key challenge is querying shared niches across diverse tissues, which is crucial for achieving a comprehensive understanding of the organization and phenotypes of cell populations. However, current data analysis methods predominantly focus on creating spatial-aware embeddings for cells, neglecting the development of niche-level representations for effective querying. To address this gap, we introduce QueST, a novel niche representation learning model designed for querying spatial niches across multiple samples. QueST utilizes a novel subgraph contrastive learning approach to explicitly capture niche-level characteristics and incorporates adversarial training to mitigate batch effects. We evaluate QueST on established benchmarks using human and mouse datasets, demonstrating its superiority over state-of-the-art graph representation learning methods in accurate niche queries. Overall, QueST offers a specialized model for spatial niche queries, paving the way for deeper insights into the patterns and mechanisms of cell spatial organization across tissues.

## 1 INTRODUCTION

A spatial niche in tissue sections is a cell and its surroundings with specific functionality or structure. Cells within tissues operate in complex spatial niches, where their functions are significantly influenced by interactions with neighboring cells. These niches are characterized by the spatial arrangement of cells and the molecular signals exchanged among them, playing a crucial role in regulating higher-order tissue functions such as immune responses, tissue regeneration, and disease progression (Wu et al., 2021; Meylan et al., 2022; Palla et al., 2022). Importantly, the behavior of the same cell type can vary dramatically depending on its niche (Wagner et al., 2016), underscoring the necessity of understanding the spatial context in cellular behavior.

Recent advances in spatial transcriptomics technologies (Xia et al., 2019; Eng et al., 2019; Rao et al., 2020; Rodrigues et al., 2019; Chen et al., 2022) have enabled scientists to inspect this important biological component from a computational view, as these technologies manage to generate gene expression data for each cell or sequencing spot while preserving its location information. Some researchers have endeavored to develop context-aware representation learning methods on spatial transcriptomics data (Hu et al., 2021; Dong & Zhang, 2022; Long et al., 2023). However, these methods focus on cell or node embedding clustering on a single spatial transcriptomics sample, limiting their application to studying spatial niches. Also, most methods are unable to remove the batch effect, limiting their usability on multiple samples.

Extending the analysis of spatial niches across multiple samples amplifies our ability to discern universal patterns and unique variations that single-sample studies might overlook. By integrating spatial transcriptomics data across different samples, including, different tissues, developmental stages, or disease conditions, we may uncover how similar niches or microenvironmental features manifest in diverse biological contexts, shedding light on the underlying mechanisms that drive normal physiology and pathological states. Recent researchers have made a step towards this multiple sample integration target (Zeira et al., 2022; Xia et al., 2023; Zhou et al., 2023), which are the

so-called ‘alignment’ methods. However, these methods focus on cross-sample similarity on the cell-cell level rather than considering it on a niche level. Although the cell-cell alignment results may generally indicate niche similarity structure, they cannot directly serve as niche similarity measurement. Furthermore, these methods are limited to aligning only two samples per training run, making them impractical for analyzing large cohorts of spatial transcriptomics samples.

Querying spatial niches across different samples presents several unique challenges. First, explicit representations are needed to compute the similarities among spatial niches and perform query tasks. However, existing graph deep learning methods typically focus on node embedding rather than sub-graph embedding. Some methods (Wu et al., 2022; Fischer et al., 2022; Wang et al., 2023) have tried to represent a subgraph via posing a pooling layer or readout function on the node embedding, but this simple approach cannot guarantee proper utilization of information within and between subgraphs under the self-supervised setting. Second, the gene expression data between different samples may contain considerable batch effect (Korsunsky et al., 2019), which must be removed before comparing niches on a sample-wise level.

To address the above challenges, we present QueST, a novel niche representation learning model that explicitly learns biologically meaningful representations for spatial niches and can perform spatial niche queries across multiple samples. We construct benchmarks for the niche query problem with the Human Dorsolateral Prefrontal Cortex (DLPFC) dataset (Maynard et al., 2021) and the Mouse Olfactory Bulb Tissue (MOBT) dataset (Guo et al., 2023), and evaluate QueST against several state-of-the-art graph representation learning and alignment methods on these benchmarks. Our framework makes the following key contributions:

- **Novel Niche Representation Learning Strategy:** We introduce a novel graph contrastive learning strategy to explicitly learn the representation of a niche, which includes obtaining subgraph embedding via a pooling layer and partially shuffling the graph while leaving certain parts fixed to generate positive and negative niche pairs.
- **Adversarial Batch Effect Removal:** We borrow the idea of Adversarial Auto-encoder (Makhzani et al., 2015) and introduce an adversarial training strategy to remove batch information from the model’s latent space during the training stage and empower the encoder with the ability to remove batch effect from unseen samples.

## 2 TASK FORMULATION

Consider a spatial transcriptomics dataset  $\mathcal{D}$  that includes gene expression measurements for  $N$  cells,  $G$  genes, and  $T$  samples. For each cell  $i$ , we have access to the gene expression vector  $x_i \in \mathbb{R}^G$ , its spatial location  $s_i \in \mathbb{R}^2$ , and a one-hot batch annotation  $b_i \in \mathbb{N}^T$  indicating the sample from which cell  $i$  originates.

We construct a spatial graph  $G = (V, E)$  based on the spatial proximity relationship among cells within each sample, where each node has node features that correspond to a cell’s gene expression vector. We define the  $k$ -hop subgraph centered at cell  $i$  as its corresponding spatial niche, denoted as  $\text{Niche}(i)$ . Consequently, the dataset  $\mathcal{D}$  will contain up to  $N$  niches in total. We categorize these niches into two sets:

**Query niche set  $\mathcal{Q}$ :** a small set of niches that are well studied, possessing known properties of interest, such as specific cell type patterns.

**Reference niche set  $\mathcal{R}$ :** a larger set of niches where the properties of interest remain largely unknown and are worth exploration.

Now we introduce the main objective of this paper:

**Niche Query Problem: What is the most similarity niche in  $\mathcal{R}$  to a query niche in  $\mathcal{Q}$ ?**

Figure 1 provides a schematic illustration of this problem. Each query niche corresponds to a “similar niche(s)” area on a reference sample, where a series of niches will all serve as the correct answers to the **Niche Query Problem**. We aim for the model not only to identify a single correct niche within the “similar niche(s)” area but also to reveal this entire area.

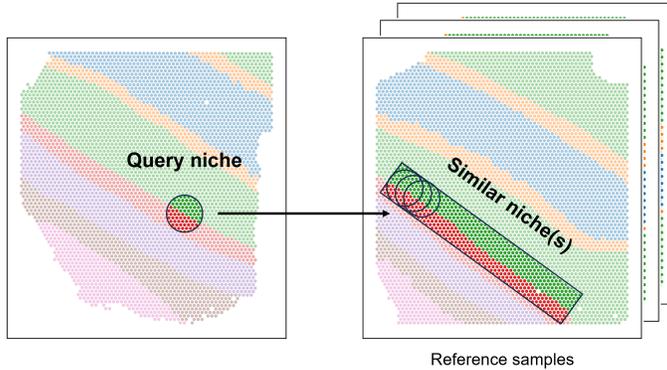


Figure 1: Illustration of the Niche Query Problem

Assume  $\text{Niche}(q) \in \mathcal{Q}$  is a given query niche and  $\text{Niche}(r) \in \mathcal{R}$  is an arbitrary niche from the reference set. The objective of the niche query problem is:

$$\text{Niche}(\hat{r}) = \arg \max_{\text{Niche}(r) \in \mathcal{R}} \text{sim}_{\mathcal{M}}[\text{Niche}(q), \text{Niche}(r)] \quad (1)$$

where  $\text{sim}_{\mathcal{M}}(\cdot, \cdot)$  is a pairwise similarity function defined by the model  $\mathcal{M}$ .

We then define several metrics to evaluate  $\mathcal{M}$ 's performance from the following two aspects:

**Best Niche Match Accuracy:** This perspective examines whether  $\text{Niche}(\hat{r})$  is accurate, requiring a ground truth metric for pairwise niche similarity. Since no existing niche similarity measurement is available, we derive this metric from ground truth cell type annotations, transforming the niche similarity problem into a subgraph similarity problem. We adopt the Wasserstein Weisfeiler-Lehman (WWL) Graph Kernel method (Togninalli et al., 2019) to calculate the similarity value between two arbitrary subgraphs. This method involves multiple information aggregation iterations, where each vertex aggregates the node embeddings (initialized with cell type labels) from its neighbors and hashes the aggregated label into a new embedding for itself. This process can be defined as:

$$\mathbf{a}^{h+1}(v) = \text{hash}(\mathbf{a}^h(v), \mathcal{N}^h(v)) \quad (2)$$

After these iterations, we compute the Wasserstein distance between the node embedding matrices of the two niches, reversing it to establish the ground truth similarity measure:

$$\text{sim}_{\text{wwl}}[\text{Niche}(q), \text{Niche}(\hat{r})] = 1 - \text{WassersteinDistance}[\text{Niche}(q), \text{Niche}(\hat{r})] \quad (3)$$

**Overall Similarity Accuracy:** This perspective assesses the accuracy of  $\text{sim}_{\mathcal{M}}$ , which in essence reflects the model's ability to reveal the entire "similar niche(s)" region in Figure 1. We compare the similarity results between  $\text{sim}_{\text{wwl}}$  and  $\text{sim}_{\mathcal{M}}$  across the entire sample. Given a query niche  $\text{Niche}(q)$  and a reference sample  $\mathcal{T}$  with its corresponding niche set  $\text{Niche}(\mathcal{T}) = \{\text{Niche}(r) | r \in \mathcal{T}\}$ , each similarity function  $\text{sim}$  generates a similarity vector  $S$ , with each entry representing the similarity value:

$$S_{\text{sim}}[r] = \text{sim}[\text{Niche}(q), \text{Niche}(r)] \quad (4)$$

We then evaluate the agreements between  $S_{\text{sim}_{\text{wwl}}}$  and  $S_{\text{sim}_{\mathcal{M}}}$  using the Pearson correlation metric:

$$\text{Accuracy}_{\text{sim}_{\mathcal{M}}} = \text{PearsonCorrelation}(S_{\text{sim}_{\text{wwl}}}, S_{\text{sim}_{\mathcal{M}}}) \quad (5)$$

### 3 THE QUEST APPROACH

In this section, we present QueST, the self-supervised graph auto-encoder framework, to query niches on multi-sample spatial transcriptomics data based on graph neural networks. Besides the main graph auto-encoder architecture, QueST has a contrastive learning module and an adversarial training module, which are two critical components that distinguish QueST from previous approaches. The contrastive learning module enables the model to discriminate between niches with different cell type compositions and topological structures, while the adversarial training module removes the batch effect from the latent niche representation. We present a schematic illustration of QueST model architecture in Figure 2.

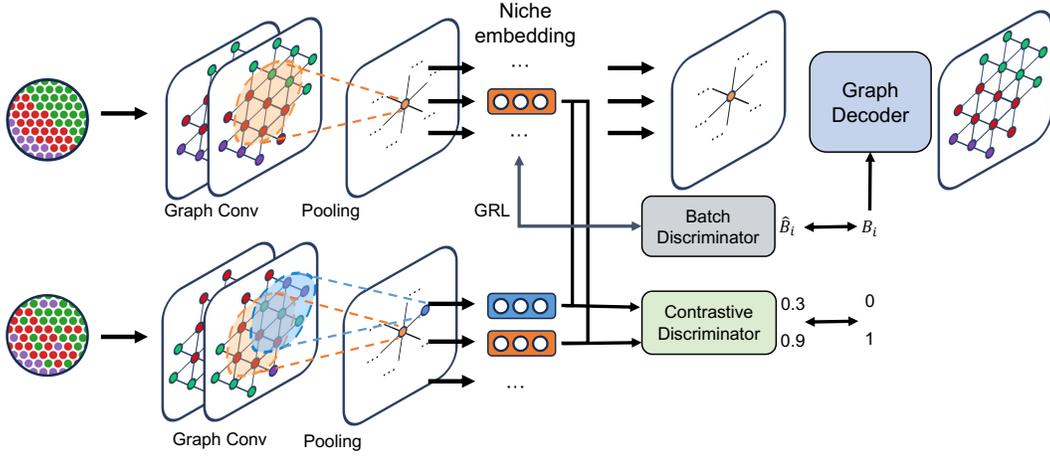


Figure 2: QueST model architecture

### 3.1 CONTRASTIVE NICHE REPRESENTATION LEARNING

The main architecture of QueST uses a graph auto-encoder with GIN (Xu et al., 2018). The graph encoder maps the normalized gene expression vector to a  $d$  dimensional latent space:

$$\mathbf{h}_i^{(l+1)} = \text{MLP}^{(l)} \left( (1 + \epsilon) \cdot \mathbf{h}_i^{(l)} + \sum_{j \in \mathcal{N}(i)} \mathbf{h}_j^{(l)} \right) \quad (6)$$

where  $\mathbf{h}_i^{(l)}$  is the node embedding of cell  $i$  after the  $l$ -th encoder layer of GNN, with  $\mathbf{h}_i^{(0)}$  initialized as  $\mathbf{x}_i$ .  $l_{\text{enc}}$  is the number of layers of the encoder, and  $\mathbf{z}_i = \mathbf{h}_i^{(l_{\text{enc}})} \in \mathbb{R}^d$  is the node embedding of cell  $i$  after the encoder.  $\epsilon$  is a scalar that controls the weight of each node’s own feature in the aggregation process.

However, this typical node-embedding fashion only learns a representation for each node or cell rather than for each niche. Even if the node embedding considers the  $k$ -hop neighborhood via information aggregation in Graph Neural Networks, the signal faints as the distance from the central node increases, while each cell should be treated equally within a niche when the prior knowledge of importance for each cell is absent. To emphasize this equal information aggregation within a niche, we add a pooling layer to force the model to learn the unbiased representation for the entire subgraph or niche centered at each cell:

$$\mathbf{z}_{\text{niche}(i)} = \text{POOL}(\mathbf{z}_j | j \in \mathcal{N}(i)) \quad (7)$$

where  $\mathbf{z}_{\text{niche}(i)}$  is the embedding of the niche centered at cell  $i$ . We then introduce a typical graph decoder  $f_{\text{dec}}$  and a reconstruction loss to make sure that the embedding contains enough information to reconstruct the input:

$$L_{\text{recon}} = \frac{1}{N} \sum_{i=1}^N \|f_{\text{dec}}(\mathbf{z}_{\text{niche}(i)}) - \mathbf{x}_i\|^2 \quad (8)$$

To make sure the niche embedding encodes both the local and global topological and structural information, we introduce a contrastive learning framework to enhance the quality of the niche embedding. Consider the spatial graph  $G_k = (V_k, E_k)$  for sample  $k, k = 1, \dots, T$ . During each training epoch, before  $G_k$  goes through the model, we conduct data augmentation via the following steps:

**1. Niche Fixing:** we randomly sample a certain subset of nodes  $V_{\text{fix}} \subset V$ . For each niche with a central node in  $V_{\text{fix}}$ , we generate a positive and a negative pair in the next step. Denote this fixed niche set as  $\text{Niche}_{\text{fix}} = \{\text{Niche}(v) | v \in V_{\text{fix}}\}$ .

**2. Graph Corruption:** We generate a corrupted graph by randomly shuffling node features while preserving the graph structure. During this process, the nodes included by  $Niche_{\text{fix}}$  will have their node features fixed, while the rest of the nodes will randomly exchange their node feature vectors. Denote the corrupted graph as  $G'_k = (V'_k, E'_k)$ . For each node  $v \in V$ , denote  $v'$  as its corresponding node in the corrupted graph  $G'$ .

**3. Dual Encoding:** Once the corrupted graph  $G'$  is generated, both  $G$  and  $G'$  will be fed into the model and will have its own latent niche representation denoted as  $\mathbf{z}_{\text{Niche}(v)}$  and  $\mathbf{z}_{\text{Niche}(v')}$ .

**4. Contrastive Sampling:** After obtaining  $\mathbf{z}_{\text{Niche}(v)}$  and  $\mathbf{z}_{\text{Niche}(v')}$ , for each node  $v$  in  $V_{\text{fix}}$  and its corresponding niche, we can define following the positive and negative embedding pair:

- Positive pair:  $(\mathbf{z}_{\text{Niche}(v)}, \mathbf{z}_{\text{Niche}(v')})$ ;
- Negative pair:  $(\mathbf{z}_{\text{Niche}(v)}, \mathbf{z}_{\text{Niche}(u')})$ , where  $u'$  is randomly sampled from  $V' \setminus \{v\}$ .

After the data augmentation step, we define the following contrastive loss:

$$L_{\text{contrast}} = \frac{1}{|V_{\text{fix}}|} \sum_{v \in V_{\text{fix}}} \log [f_{\text{contrast}}(\mathbf{z}_{\text{Niche}(v)}, \mathbf{z}_{\text{Niche}(v')})] + \log [1 - f_{\text{contrast}}(\mathbf{z}_{\text{Niche}(v)}, \mathbf{z}_{\text{Niche}(u')})] \quad (9)$$

where we build the contrastive loss following Deep Graph Infomax (Velickovic et al., 2019), with  $f_{\text{contrast}}$  being a bilinear discriminator  $f_{\text{contrast}} : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$  which distinguishes the positive pair from the negative pair.

### 3.2 BATCH REMOVAL VIA ADVERSARIAL TRAINING

The above architecture is designed to learn niche representation on one single spatial transcriptomics sample. To enable QueST query niches across multiple samples, we develop an adversarial training method to remove the batch effect from the latent niche representation when training the model on multiple samples. The basic logic of this adversarial training is to make a discriminator to discriminate batch labels from the niche embeddings while simultaneously training the encoder adversarially to fool the batch discriminator. To achieve this, we input the latent niche representation  $\mathbf{z}_{\text{Niche}(v)}$  to a batch discriminator  $f_{\text{batch}}$  to predict which sample this niche is from and use a Gradient Reversal Layer (Ganin et al., 2016) for giving reverse gradient to the encoder. We denote the predicted batch labels as

$$\hat{\mathbf{b}}_v = f_{\text{batch}}(GRL(\mathbf{z}_{\text{Niche}(v)})) \quad (10)$$

where GRL is:

$$GRL(\mathbf{z}_{\text{Niche}(v)}) = \mathbf{z}_{\text{Niche}(v)}; \quad \frac{\partial GRL(\mathbf{z}_{\text{Niche}(v)})}{\partial \mathbf{z}_{\text{Niche}(v)}} = -1 \quad (11)$$

then we have the following batch prediction loss:

$$L_{\text{batch}} = -\frac{1}{N} \sum_{i=1}^N \left( \sum_{j=1}^T \mathbf{b}_i^{(j)} \log \hat{\mathbf{b}}_i^{(j)} + (1 - \mathbf{b}_i^{(j)}) \log (1 - \hat{\mathbf{b}}_i^{(j)}) \right) \quad (12)$$

However, the removal of batch information from the latent representation will add difficulties to reconstruction, since the gene expression that which decoder is required to reconstruct still contains the batch effect. To mitigate the negative impact of batch removal on the reconstruction task, we add batch information back to the input of the decoder:

$$\mathbf{x}_{\text{recon}_v} = f_{\text{dec}}(\mathbf{z}_{\text{Niche}(v)}, \hat{\mathbf{b}}_v) \quad (13)$$

and the reconstruction loss becomes

$$L_{\text{recon}} = \frac{1}{N} \sum_{i=1}^N \|\mathbf{x}_{\text{recon}_i} - \mathbf{x}_i\|^2 \quad (14)$$

### 3.3 QUEST OVERALL LOSS FUNCTION

The overall loss function of QueST is defined as follows:

$$L = \lambda_1 L_{\text{recon}} + \lambda_2 L_{\text{contrast}} + \lambda_3 L_{\text{batch}} \quad (15)$$

where we set  $\lambda_1 = 1$ ,  $\lambda_2 = 1$ ,  $\lambda_3 = 0.1$  as default across all experiments. For other hyperparameters and implementation details, please refer to section A.3.

### 3.4 QUERYING NICHES IN LATENT SPACE

Finally, we measure the similarity between two arbitrary spatial niches using the cosine similarity of their latent representation, which is used for the query:

$$\text{sim}_{\text{QueST}}[\text{Niche}(q), \text{Niche}(r)] = \frac{\mathbf{z}_{\text{Niche}(q)}^\top \mathbf{z}_{\text{Niche}(r)}}{\|\mathbf{z}_{\text{Niche}(q)}\| \cdot \|\mathbf{z}_{\text{Niche}(r)}\|} \quad (16)$$

## 4 EXPERIMENTS

In this section, we present our experimental results. We set  $k = 3$  as the default for the definition of a spatial niche across all experiments. For the WWL Graph Kernel, We set the iteration number to be 3 and use it as the ground truth similarity measurement.

### 4.1 BENCHMARK CONSTRUCTION

As there are currently no available multi-sample datasets with niche-level annotations that can serve as benchmarks, we synthesize our niche query benchmarks based on existing spatial transcriptomics datasets with cell-type annotation.

Specifically, we use two widely used datasets in spatial transcriptomics: the Human Dorsolateral Prefrontal Cortex (DLPFC) dataset and the Mouse Olfactory Bulb Tissue (MOBT) dataset. These datasets represent niche query problems with varying levels of challenges. On each dataset, we first separate samples into query and reference set, and then generate niches that need to be queried. The DLPFC dataset consists of 12 samples from the human brain, with ground truth cell type annotation illustrated in Figure A.1. We partition this dataset into a query set with 1 sample and a reference set with 11 samples. This setting requires the model to perform batch effect removal and niche query on a large cohort of spatial transcriptomics samples. The MOBT dataset contains 3 samples from different sequencing technologies including 10X Visium, Stereo-seq (Chen et al., 2022) and Slide-seq V2 (Stickels et al., 2021). The radius of data spots for each technology differs; they are  $50\mu\text{m}$ ,  $35\mu\text{m}$  and  $10\mu\text{m}$ , respectively. As the ground truth cell type annotation shown in Figure A.2, the variation in spot resolution affects the spatial distribution of the sequencing spots or cells, which is an even harder task. It requires the niche query methods not only to remove the batch effect in gene expression but also to model the spatial context on heterogeneous graphs. We split the MOBT dataset into a query set with 1 sample from Stereo-seq and a reference set with 2 samples from 10X Visium and Slide-seq V2. The detailed description of these two datasets can be found in section A.1.

Once the query and reference sample sets are established, any niche within the query sample can correspond to a niche query problem. In this paper, we generate several query niches by changing these two parameters:

**1. Size:** This refers to the number of cells included in the niche. We consider three scenarios with the query niche containing 50, 100, and 200 cells.

**2. Cell Type Pattern:** This refers to the number of cell types contained in the niche. We also examine three scenarios, where the query niche includes 1, 2, and 3 cell types.

By combining these two aspects of variation, in total 9 query niches can be defined for each query sample, as illustrated in Figure A.3 and Figure A.4. See section A.2 for detailed generation of query niches.

We evaluate QueST’s performance against three popular graph deep learning methods in spatial transcriptomics: GraphST, SLAT, and STAGATE. GraphST and STAGATE are representation learning

Table 1: Results for Pearson correlation on DLPFC dataset. We report the mean value on the 11 reference samples for each query niche.

Niche Name	GraphST	SLAT	STAGATE	QueST
Layer3_Layer4_Layer5_100	0.476	<b>0.528</b>	0.130	0.383
Layer3_Layer4_Layer5_200	0.477	0.607	0.102	<b>0.620</b>
Layer3_Layer4_Layer5_50	0.335	0.506	0.068	<b>0.585</b>
Layer4_100	0.288	0.418	0.062	<b>0.535</b>
Layer4_200	0.285	0.430	0.063	<b>0.535</b>
Layer4_50	0.282	0.415	0.058	<b>0.529</b>
Layer5_Layer6_100	0.386	0.658	0.132	<b>0.802</b>
Layer5_Layer6_200	0.463	0.729	0.152	<b>0.743</b>
Layer5_Layer6_50	0.433	<b>0.720</b>	0.140	0.714

methods designed for single samples, utilizing GCN (Kipf & Welling, 2016) and GAT (Veličković et al., 2017) as their respective model backbones, while SLAT is tailored for the alignment of two spatial transcriptomics slides, employing LGCN (Xia et al., 2023) as its backbone and SVD pre-processing for batch effect removal. Additionally, SLAT solves an optimal transport optimization problem in its latent space to establish node correspondence relationships between samples as alignment results. However, all these competing methods focus on node embedding learning; thus, we manually perform pooling on their embeddings to obtain niche representations. We utilize the cosine similarity between the representations of niches in reference samples and the query niche as the similarity measurement corresponding to these representation learning methods.

#### 4.2 QUEST QUERIES NICHES ACCURATELY ON LARGE COHORTS OF REFERENCE SAMPLES

We first evaluate QueST’s performance on the DLPFC datasets. Table 1 presents the performance of GraphST, STAGATE, SLAT, and QueST on the DLPFC dataset, measured by the Overall Similarity Accuracy. As shown, the niche similarity results generated by QueST closely resemble the ground truth, achieving the highest Pearson correlation for 7 out of the 9 query niches. SLAT, utilizing its LGCN and batch correction module, also performs well, attaining the best results for the remaining 2 query niches. In contrast, the representation learning methods designed for single samples, including GraphST and STAGATE, perform poorly on this niche query task.

Figure 3 and Figure A.5-A.9 showcase the spatial distribution of cosine similarity for the example query niche Layer5\_Layer6\_100. As shown in these figures, QueST visually aligns most closely with the ground truth, with the high-similarity regions closely following the ground truth patterns, exhibiting fewer noisy artifacts and a smoother gradient across the reference sample. This indicates that QueST excels at reproducing the similarity structure of the ground truth. In contrast, the other methods capture the general spatial similarity pattern but lack niche embedding learning, making their latent spaces unsuitable for accurately measuring niche similarity.

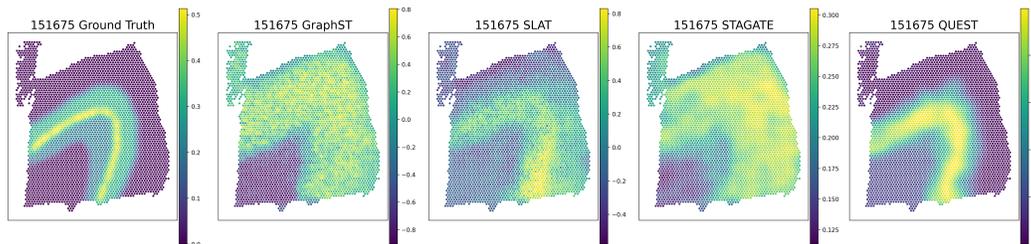


Figure 3: Visualization of cosine similarity of niche Layer5\_Layer6\_100 on sample 151675

Table 2 presents the performance of these methods based on **Best Niche Match Accuracy**. From this table, it is evident that QueST achieves the best results for nearly all query niches. These results are consistent with those from the **Overall Similarity Accuracy**, as a more reasonable similarity measurement is more likely to accurately address the **Niche Query Problem**. Figure 4 visualizes

Table 2: Results for Subgraph Wasserstein Distance on DLPFC dataset. We report the mean value on the 11 reference samples.

Niche Name	GraphST	SLAT	STAGATE	QueST
Layer3_Layer4_Layer5_100	0.917	<b>0.809</b>	0.936	0.870
Layer3_Layer4_Layer5_200	0.840	0.842	0.896	<b>0.837</b>
Layer3_Layer4_Layer5_50	0.941	0.869	0.957	<b>0.858</b>
Layer4_100	0.946	0.907	0.989	<b>0.883</b>
Layer4_200	0.967	0.910	0.989	<b>0.897</b>
Layer4_50	0.938	0.891	0.989	<b>0.889</b>
Layer5_Layer6_100	0.758	0.803	0.969	<b>0.751</b>
Layer5_Layer6_200	0.797	0.776	0.963	<b>0.771</b>
Layer5_Layer6_50	0.781	0.782	0.968	<b>0.775</b>

the niche query results for sample 151675, highlighting only the queried region’s cell types. We observe that QueST generates results that most closely match the results generated by ground truth similarity measurement, accurately capturing both the correct cell types and their adjacency relationships within the marked region. We also observe that alignment methods based on OT, such as SLAT, can find regions with cell type largely correct, but fail to produce a concentrated query area that corresponds to a local neighborhood or niche. This limitation arises because OT-based methods focus on computing cell-cell similarity rather than niche-niche similarity, mapping each cell in the query niche to its most similar counterpart in the reference niche. Therefore, while popular in multi-sample integration, these methods are not well-suited for the niche query task.

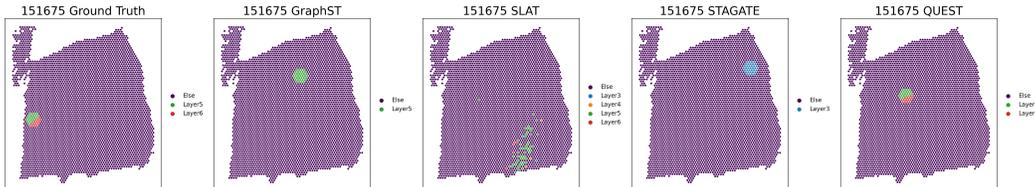


Figure 4: Visualization of query result on example niche Layer5\_Layer6\_100. We show the cell type of the region which is determined by the method as the query result and mask the cell type of the rest part.

### 4.3 QUEST QUERIES NICHE ACCURATELY ACROSS DIFFERENT SEQUENCING TECHNOLOGIES

Next, we evaluate QueST’s performance on the Mouse Olfactory Bulb Tissue dataset. Table 3 shows the performance on the MOBT dataset in terms of the **Overall Similarity Accuracy**. Again, QueST achieves the highest Pearson correlation value on most of the query niches, demonstrating QueST’s superiority for querying niches across samples with varying sequencing technologies. Figure 5 and Figure A.10-A.14 showcase the spatial distribution of cosine similarity of an example query niche GL\_ONL\_100. We observe that most of the results degrade significantly compared to those on DLPFC datasets, largely due to the substantial variations between samples introduced by the difference in sequencing technology. Nevertheless, QueST successfully retains most of the underlying similarity structure. In contrast, other methods suffer badly from this technical challenge, producing cosine similarity that almost spreads uniformly across the entire tissue, with low contrast between the target regions and the others.

Similar trends are observed when assessing these methods on the MOBT dataset using **Best Niche Match Accuracy**. Table 4 summarizes the mean performance of the nine query niches across two reference samples, with QueST consistently delivering the best results for most query niches. Figure 6 visualizes the query results for the example niche GL\_ONL\_100, again showing that QueST’s outputs most closely resemble the ground truth.

Table 3: Results for Pearson Correlation on Mouse Olfactory Bulb Tissue dataset. We report the mean value on the 2 reference samples.

Niche Name	GraphST	SLAT	STAGATE	QueST
GCL_100	0.437	0.769	0.576	<b>0.777</b>
GCL_150	0.518	0.669	0.539	<b>0.771</b>
GCL_50	0.359	0.747	0.551	<b>0.773</b>
GCL_MCL_EPL_100	0.130	0.515	0.215	<b>0.544</b>
GCL_MCL_EPL_150	0.116	<b>0.572</b>	0.283	0.557
GCL_MCL_EPL_50	0.223	<b>0.541</b>	0.081	0.288
GL_ONL_100	0.211	0.573	-0.256	<b>0.766</b>
GL_ONL_150	0.156	0.691	-0.237	<b>0.768</b>
GL_ONL_50	0.241	0.663	-0.428	<b>0.812</b>

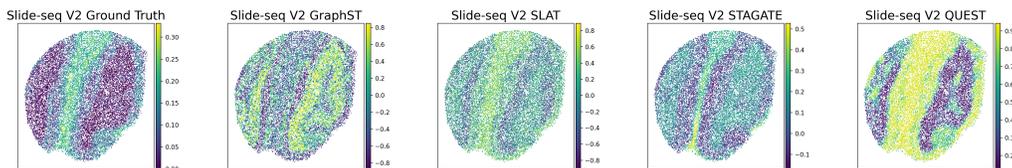


Figure 5: Visualization of cosine similarity on example niche GL\_ONL\_100

#### 4.4 QUEST ABLATION EXPERIMENTS

In this section, we present the results of the ablation experiments, as shown in Table 5. From these results, we can draw several important observations:

First, removing the contrastive learning module significantly degrades the model’s performance. This finding is intuitive, as the primary purpose of incorporating contrastive learning is to help the model distinguish between potentially similar yet fundamentally different niches (for instance, niches with the same cell type composition but different arrangements are treated as negative pairs) and to recognize identical niches in varying external environments (positive pairs). Our model employs subgraph pooling to generate niche embeddings, which can obscure the topological structure within the niche or subgraph. The introduction of contrastive learning effectively mitigates this issue.

Second, the absence of pooling also severely decreases the model’s performance. We argue that node embedding methods are ill-suited for the niche query problem. This is primarily due to the fact that, while the information aggregated in node embeddings considers the microenvironments surrounding each cell, the signal diminishes with increasing distance from neighboring cells to the central cell. Given that a spatial niche comprises dozens of cells, it is challenging to identify a

Table 4: Results for Subgraph Wasserstein Distance on Mouse Olfactory Bulb Tissue dataset. We report the mean value on the 2 reference samples.

Niche Name	GraphST	SLAT	STAGATE	QueST
GCL_100	0.794	0.741	0.812	<b>0.729</b>
GCL_150	0.800	0.762	0.883	<b>0.738</b>
GCL_50	0.808	0.729	0.815	<b>0.692</b>
GCL_MCL_EPL_100	0.819	0.835	0.894	<b>0.803</b>
GCL_MCL_EPL_150	0.822	<b>0.772</b>	0.881	0.797
GCL_MCL_EPL_50	0.839	0.842	0.926	<b>0.838</b>
GL_ONL_100	0.918	0.829	0.865	<b>0.782</b>
GL_ONL_150	0.848	0.810	0.870	<b>0.763</b>
GL_ONL_50	0.906	0.810	0.887	<b>0.765</b>

486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539

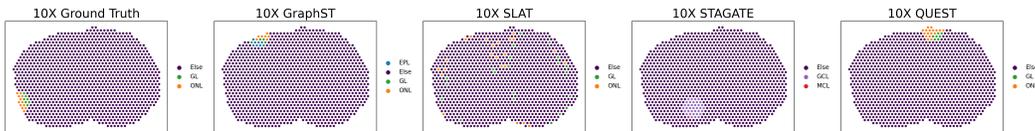


Figure 6: Visualization of query result on example niche GL\_ONL\_100

Table 5: Ablation study for batch discriminator, contrastive learning and pooling on QueST model, where No-X means the QueST model without the X module. We report mean Pearson Correlation on the 11 reference samples of the DLPFC dataset.

Niche Name	No-Batch	No-Contrastive	No-Pooling	QueST
Layer3_Layer4_Layer5_100	0.302	0.316	0.260	<b>0.383</b>
Layer3_Layer4_Layer5_200	0.402	0.263	0.306	<b>0.620</b>
Layer3_Layer4_Layer5_50	0.524	0.272	0.369	<b>0.585</b>
Layer4_100	<b>0.540</b>	0.258	0.226	0.535
Layer4_200	0.527	0.259	0.343	<b>0.535</b>
Layer4_50	0.508	0.248	0.213	<b>0.529</b>
Layer5_Layer6_100	0.647	0.294	0.496	<b>0.802</b>
Layer5_Layer6_200	0.498	0.393	0.280	<b>0.743</b>
Layer5_Layer6_50	0.549	0.399	0.389	<b>0.714</b>

dominant cell. Consequently, using the central cell’s node embedding as the representative for the entire niche can introduce significant bias when calculating the similarity between different niches.

Lastly, the removal of the batch discriminator has a relatively minor impact on the model’s performance. This may be attributed to the relatively small batch effects present in the DLPFC dataset. Nonetheless, the overall performance indicates that the batch discriminator contributes to more accurate niche similarity measurements across different samples.

## 5 DISCUSSION AND CONCLUSION

In this study, we introduce the **Niche Query Problem** and develop QueST, a novel and specialized framework for querying spatial niches across diverse spatial transcriptomics samples. QueST employs a subgraph contrastive learning approach to explicitly learn niche representation, incorporates an adversarial training framework to perform batch correction, and finally uses cosine similarity between latent representation vectors to perform spatial niche query. Experimental results demonstrate the efficacy of these critical modules and validate QueST’s capability for accurately querying spatial niches in various settings. To the best of our knowledge, QueST is the first graph deep learning model tailored for querying spatial niches in spatial transcriptomics data. We believe that with the increased volume of spatial transcriptomic data, this niche query problem will be a foundation for investigating pan-cancer or cross-species studies, and both the QueST method and the query benchmark we developed will lay the groundwork for addressing this problem.

Certain limitations exist for QueST and further work is desired to improve the framework. At present, QueST requires computing  $k$ -hop subgraphs with a fixed  $k$  to extract spatial niches on reference samples for comparison with the query niche. Handling niches with various sizes and shapes simultaneously on reference samples is a potential future direction for improvement. Furthermore, the graph auto-encoder architecture limits the scale of the model. Introducing novel architecture such as transformers may help scale up the model for larger pre-training and querying tasks on datasets with millions of cells or even more.

## REFERENCES

- 540  
541  
542 Ao Chen, Sha Liao, Mengnan Cheng, Kailong Ma, Liang Wu, Yiwei Lai, Xiaojie Qiu, Jin Yang,  
543 Jiangshan Xu, Shijie Hao, et al. Spatiotemporal transcriptomic atlas of mouse organogenesis  
544 using dna nanoball-patterned arrays. *Cell*, 185(10):1777–1792, 2022.
- 545 Kangning Dong and Shihua Zhang. Deciphering spatial domains from spatially resolved transcrip-  
546 tomics with an adaptive graph attention auto-encoder. *Nature communications*, 13(1):1739, 2022.
- 547 Chee-Huat Linus Eng, Michael Lawson, Qian Zhu, Ruben Dries, Noushin Koulona, Yodai Takei,  
548 Jina Yun, Christopher Cronin, Christoph Karp, Guo-Cheng Yuan, et al. Transcriptome-scale  
549 super-resolved imaging in tissues by rna seqfish+. *Nature*, 568(7751):235–239, 2019.
- 550 David S Fischer, Mayar Ali, Sabrina Richter, Ali Ertürk, and Fabian Theis. Graph neural networks  
551 learn emergent tissue properties from spatial molecular profiles. *bioRxiv*, pp. 2022–12, 2022.
- 552 Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François  
553 Laviolette, Mario March, and Victor Lempitsky. Domain-adversarial training of neural networks.  
554 *Journal of machine learning research*, 17(59):1–35, 2016.
- 555 Tiantian Guo, Zhiyuan Yuan, Yan Pan, Jiakang Wang, Fengling Chen, Michael Q Zhang, and Xi-  
556 angyu Li. Spiral: integrating and aligning spatially resolved transcriptomics data across different  
557 experiments, conditions, and technologies. *Genome Biology*, 24(1):241, 2023.
- 558 Jian Hu, Xiangjie Li, Kyle Coleman, Amelia Schroeder, Nan Ma, David J Irwin, Edward B Lee,  
559 Russell T Shinohara, and Mingyao Li. Spagcn: Integrating gene expression, spatial location and  
560 histology to identify spatial domains and spatially variable genes by graph convolutional network.  
561 *Nature methods*, 18(11):1342–1351, 2021.
- 562 Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional net-  
563 works. *arXiv preprint arXiv:1609.02907*, 2016.
- 564 Ilya Korsunsky, Nghia Millard, Jean Fan, Kamil Slowikowski, Fan Zhang, Kevin Wei, Yuriy  
565 Baglaenko, Michael Brenner, Po-ru Loh, and Soumya Raychaudhuri. Fast, sensitive and accurate  
566 integration of single-cell data with harmony. *Nature methods*, 16(12):1289–1296, 2019.
- 567 Yahui Long, Kok Siong Ang, Mengwei Li, Kian Long Kelvin Chong, Raman Sethi, Chengwei  
568 Zhong, Hang Xu, Zhiwei Ong, Karishma Sachaphibulkij, Ao Chen, et al. Spatially informed  
569 clustering, integration, and deconvolution of spatial transcriptomics with graphst. *Nature Com-  
570 munications*, 14(1):1155, 2023.
- 571 Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial  
572 autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.
- 573 Kristen R Maynard, Leonardo Collado-Torres, Lukas M Weber, Cedric Uyttingco, Brianna K Barry,  
574 Stephen R Williams, Joseph L Cattalini, Matthew N Tran, Zachary Besich, Madhavi Tippiani,  
575 et al. Transcriptome-scale spatial gene expression in the human dorsolateral prefrontal cortex.  
576 *Nature neuroscience*, 24(3):425–436, 2021.
- 577 Maxime Meylan, Florent Petitprez, Etienne Becht, Antoine Bougouin, Guilhem Pupier, Anne  
578 Calvez, Ilenia Giglioli, Virginie Verkarre, Guillaume Lacroix, Johanna Verneau, et al. Tertiary  
579 lymphoid structures generate and propagate anti-tumor antibody-producing plasma cells in renal  
580 cell cancer. *Immunity*, 55(3):527–541, 2022.
- 581 Giovanni Palla, David S Fischer, Aviv Regev, and Fabian J Theis. Spatial components of molecular  
582 tissue biology. *Nature Biotechnology*, 40(3):308–318, 2022.
- 583 Nikhil Rao, Sheila Clark, and Olivia Habern. Bridging genomics and tissue pathology: 10x ge-  
584 nomics explores new frontiers with the visium spatial gene expression solution. *Genetic Engi-  
585 neering & Biotechnology News*, 40(2):50–51, 2020.
- 586 Samuel G Rodriques, Robert R Stickels, Aleksandrina Goeva, Carly A Martin, Evan Murray,  
587 Charles R Vanderburg, Joshua Welch, Linlin M Chen, Fei Chen, and Evan Z Macosko. Slide-  
588 seq: A scalable technology for measuring genome-wide expression at high spatial resolution.  
589 *Science*, 363(6434):1463–1467, 2019.

- 594 Robert R Stickels, Evan Murray, Pawan Kumar, Jilong Li, Jamie L Marshall, Daniela J Di Bella,  
595 Paola Arlotta, Evan Z Macosko, and Fei Chen. Highly sensitive spatial transcriptomics at near-  
596 cellular resolution with slide-seq2. *Nature biotechnology*, 39(3):313–319, 2021.  
597
- 598 Matteo Togninalli, Elisabetta Ghisu, Felipe Llinares-López, Bastian Rieck, and Karsten Borgwardt.  
599 Wasserstein weisfeiler-lehman graph kernels. *Advances in neural information processing systems*,  
600 32, 2019.
- 601 Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua  
602 Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.  
603
- 604 Petar Velickovic, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon  
605 Hjelm. Deep graph infomax. *ICLR (Poster)*, 2(3):4, 2019.
- 606 Allon Wagner, Aviv Regev, and Nir Yosef. Revealing the vectors of cellular identity with single-cell  
607 genomics. *Nature biotechnology*, 34(11):1145–1160, 2016.  
608
- 609 Hanchen Wang, Jianke Yu, Xiaoyang Wang, Chen Chen, Wenjie Zhang, and Xuemin Lin. Neu-  
610 ral similarity search on supergraph containment. *IEEE Transactions on Knowledge and Data  
611 Engineering*, 36(1):281–295, 2023.
- 612 Rui Wu, Wenbo Guo, Xinyao Qiu, Shicheng Wang, Chengjun Sui, Qiuyu Lian, Jianmin Wu, Yiran  
613 Shan, Zhao Yang, Shuai Yang, et al. Comprehensive analysis of spatial architecture in primary  
614 liver cancer. *Science Advances*, 7(51):eabg3750, 2021.
- 615 Zhenqin Wu, Alexandro E Trevino, Eric Wu, Kyle Swanson, Honesty J Kim, H Blaize D’Angio,  
616 Ryan Preska, Gregory W Charville, Piero D Dalerba, Ann Marie Egloff, et al. Space-gm: ge-  
617 ometric deep learning of disease-associated microenvironments from multiplex spatial protein  
618 profiles. *bioRxiv*, pp. 2022–05, 2022.  
619
- 620 Chen-Rui Xia, Zhi-Jie Cao, Xin-Ming Tu, and Ge Gao. Spatial-linked alignment tool (slat) for  
621 aligning heterogeneous slices. *Nature Communications*, 14(1):7236, 2023.
- 622 Chenglong Xia, Jean Fan, George Emanuel, Junjie Hao, and Xiaowei Zhuang. Spatial transcriptome  
623 profiling by merfish reveals subcellular rna compartmentalization and cell cycle-dependent gene  
624 expression. *Proceedings of the National Academy of Sciences*, 116(39):19490–19499, 2019.  
625
- 626 Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural  
627 networks? *arXiv preprint arXiv:1810.00826*, 2018.
- 628 Ron Zeira, Max Land, Alexander Strzalkowski, and Benjamin J Raphael. Alignment and integration  
629 of spatial transcriptomics data. *Nature Methods*, 19(5):567–575, 2022.  
630
- 631 Xiang Zhou, Kangning Dong, and Shihua Zhang. Integrating spatial transcriptomics data across  
632 different conditions, technologies and developmental stages. *Nature Computational Science*, 3  
633 (10):894–906, 2023.  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647

## A APPENDIX

### A.1 DETAILED DESCRIPTION OF DATASETS

The DLPFC dataset contains 12 samples as summarized in Table A.1:

Table A.1: Summary of the DLPFC dataset

Sample ID	Spot Number	Gene Number
151507	4221	33538
151508	4381	33538
151509	4788	33538
151510	4595	33538
151669	3636	33538
151670	3484	33538
151671	4093	33538
151672	3888	33538
151673	3611	33538
151674	3635	33538
151675	3566	33538
151676	3431	33538

Besides, the DLPFC contains 7 cell types in total, namely Layer1, Layer2, Layer3, Layer4, Layer5, Layer6, and WM, which corresponds to different layers of cortex and the white matter. The ground truth cell type annotation is depicted in Figure A.1.

The MOBT dataset contains 3 samples as summarized in Table A.2:

Table A.2: Summary of the MOBT dataset

Sequencing Technology	Spot Number	Gene Number
10X Visium	1185	5531
Stereo-seq	8762	5531
Slide-seq V2	18173	5531

The MOBT dataset contains 5 cell types in total, which are EPL (External Plexiform Layer), GCL (Granular Cell Layer), GL (Glomerular Layer), MCL (Mitral Cell Layer), ONL (Olfactory Nerve Layer). We show the ground truth cell type annotation in Figure A.2.

### A.2 DETAILED GENERATION OF QUERY NICHES

In this section, we describe how the query niches are selected in detail.

For the DLPFC dataset, we choose sample 151507 as the query sample, upon which we define the query niches. We propose the following three cell-type patterns:

1. Layer4;
2. Layer5 and Layer6;
3. Layer3, Layer4 and Layer6.

With the niche size determined as 50, 100, and 200 cells beforehand, we select the region that satisfies these 9 different requirements respectively. For each setting, we first randomly determine an anchor cell for the query niche and expand this subgraph with its k-hop neighbors until the subgraph satisfies the size requirements. The query niches selected are visualized in Figure A.3.

For the MOBT dataset, we choose the Stereo-seq sample as the query sample and define query niches on it. We choose the following cell-type patterns:

1. GCL;

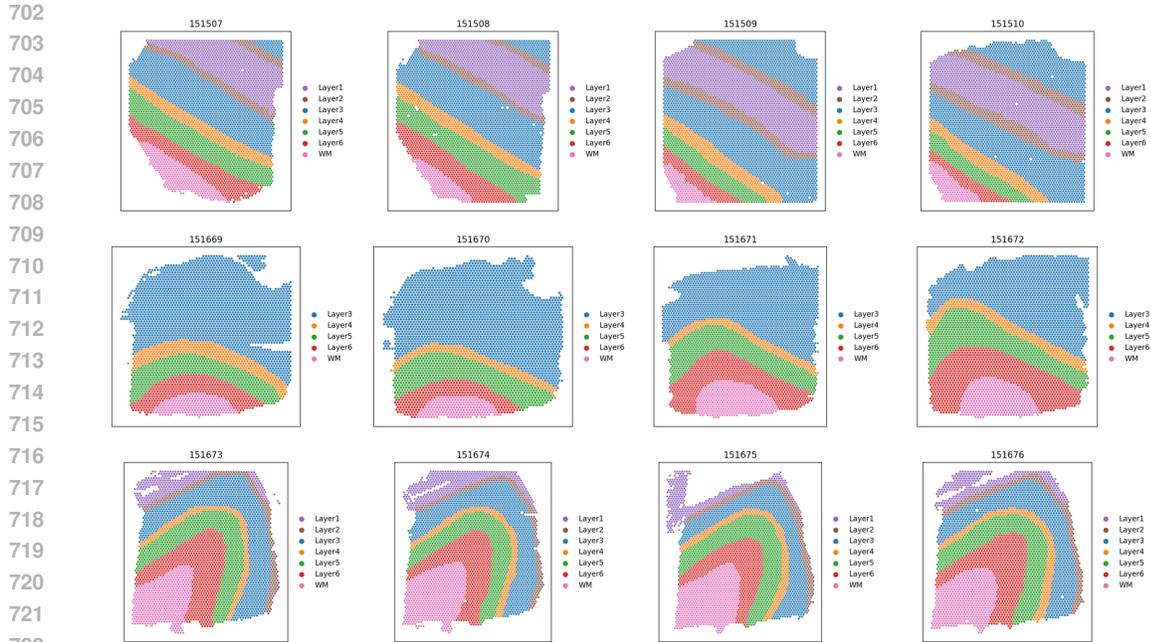


Figure A.1: ground truth cell type annotation on DLPFC dataset

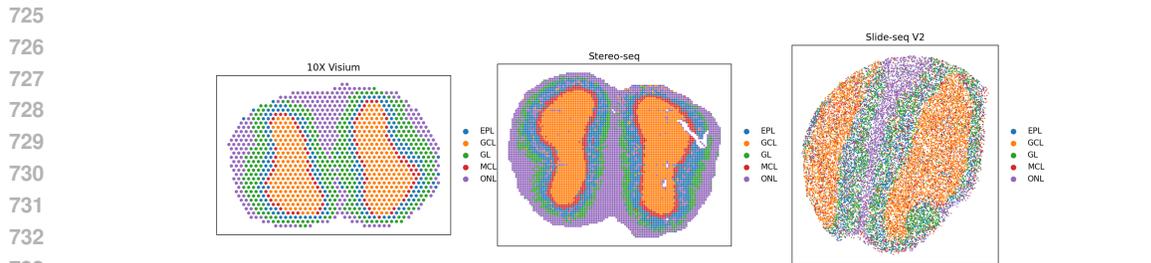


Figure A.2: ground truth cell type annotation on MOBT dataset

- 740 2. GL and ONL;
- 741 3. GCL, MCL, and EPL.

742 We then set out to select appropriate regions with the same niche size requirements. Since the cell  
743 type distribution in this tissue is not as smooth as in the DLPFC dataset, it is hard to find a region  
744 that satisfies the cell type requirements. Instead, we choose regions that contain the required cell  
745 types with a proportion over 95%. The query niches of the MOBT dataset are shown in Figure A.4.

### 746 A.3 IMPLEMENTATION DETAILS FOR QUEST MODEL

747 Here we present the default configuration for QueST in Table A.3:

748 where the fix ratio refers to the fixed number of niches when generating positive and negative pairs,  
749 and the negative pair shuffle ratio is used to determine whether a shuffled subgraph can be sampled  
750 as a negative pair. QueST is implemented via PyTorch 2.2.1 and PyTorch Geometric 2.5.2. All  
751 experiments are done on an Nvidia A100-80G GPU.

752  
753  
754  
755

756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809

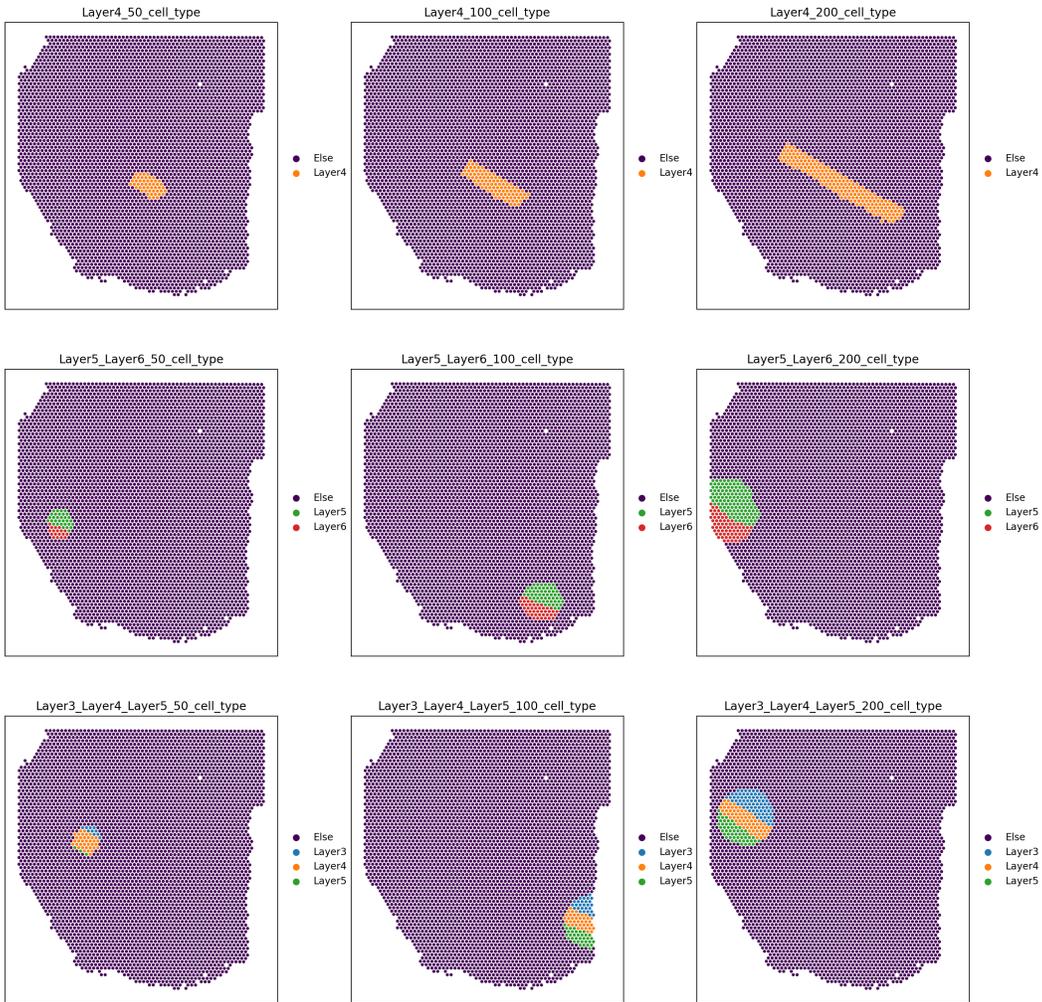


Figure A.3: Query niches of the DLPFC dataset

810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863

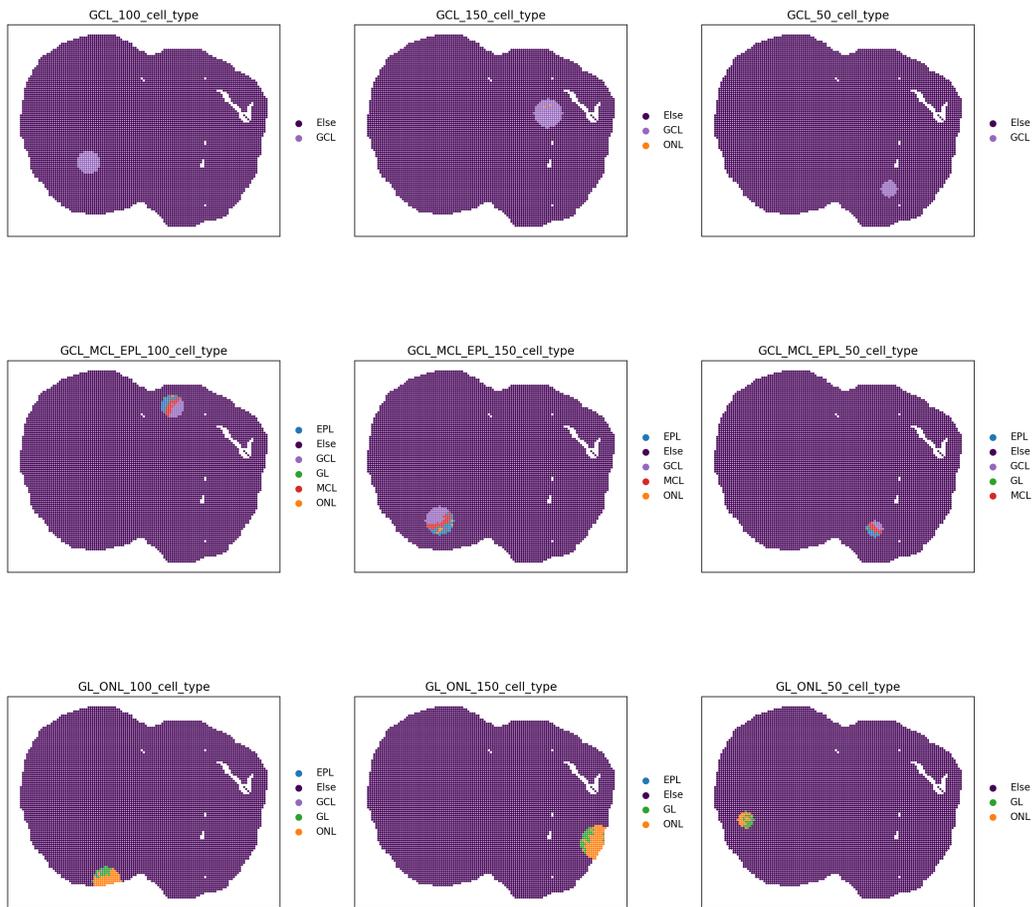


Figure A.4: Query niches of the MOBt dataset

Table A.3: Default configuration for QueST

config	value
gnn type	GIN
$\epsilon$ of GIN	0
encoder gnn layer number	3
decoder gnn layer number	1
bottleneck dimension	32
k	3
fix ratio	2%
negative pair shuffle ratio	25%-75%
optimizer	Adam
learning rate	0.001
weight decay	5e-4

#### A.4 EXTENDED FIGURES

In this section, we present the results of the niche Layer5\_Layer6\_100 on all samples of the DLPFC dataset, and GL\_ONL\_100 on all samples of the MOBT dataset. For results of the other niches, please refer to the supplementary materials.

##### A.4.1 PEARSON CORRELATION ON THE DLPFC DATASET

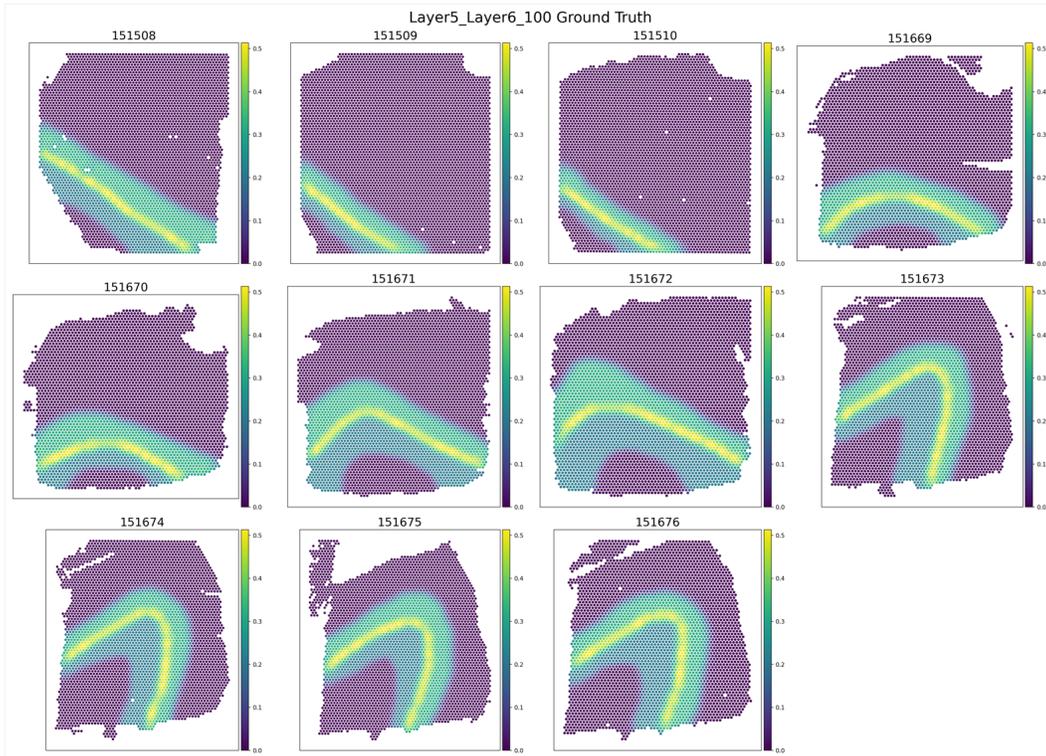


Figure A.5: Pearson Correlation of Ground Truth on the DLPFC dataset

918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971

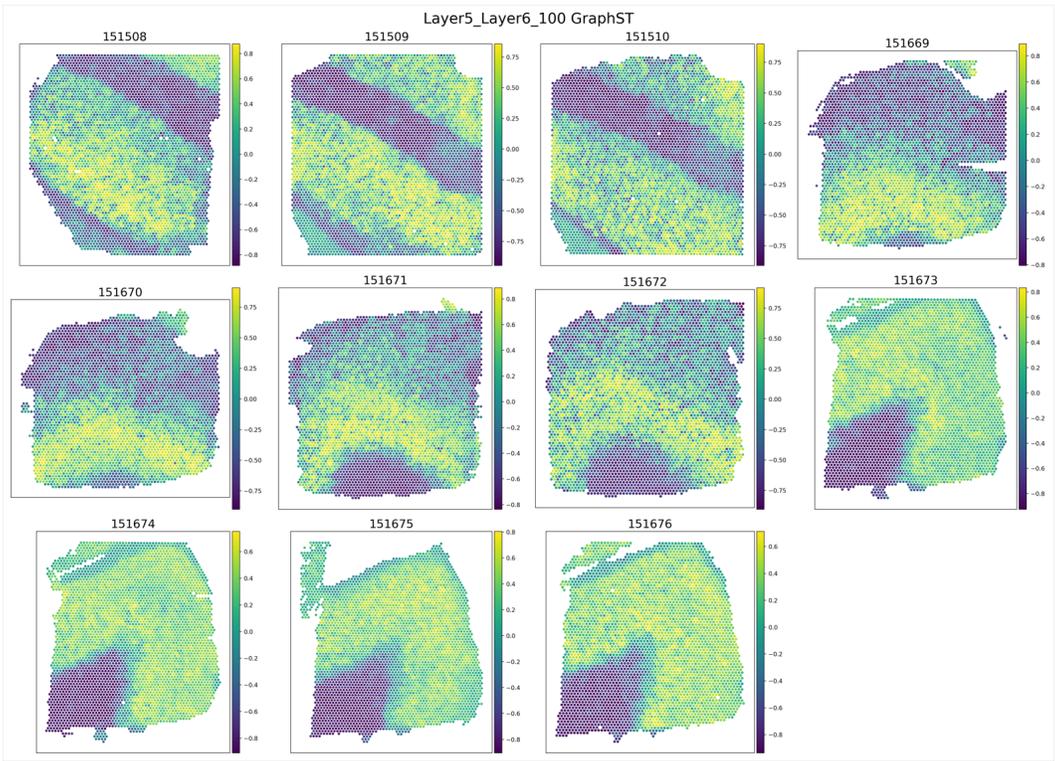


Figure A.6: Pearson Correlation of GraphST on the DLPFC dataset

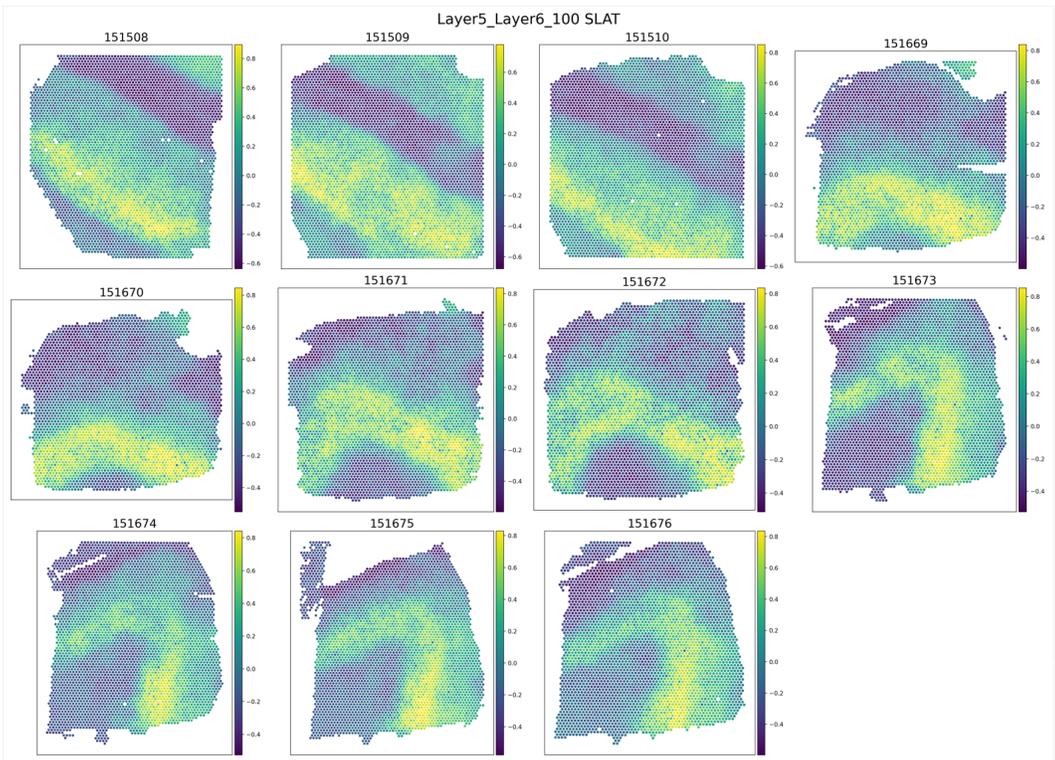


Figure A.7: Pearson Correlation of SLAT on the DLPFC dataset

972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000  
1001  
1002  
1003  
1004  
1005  
1006  
1007  
1008  
1009  
1010  
1011  
1012  
1013  
1014  
1015  
1016  
1017  
1018  
1019  
1020  
1021  
1022  
1023  
1024  
1025

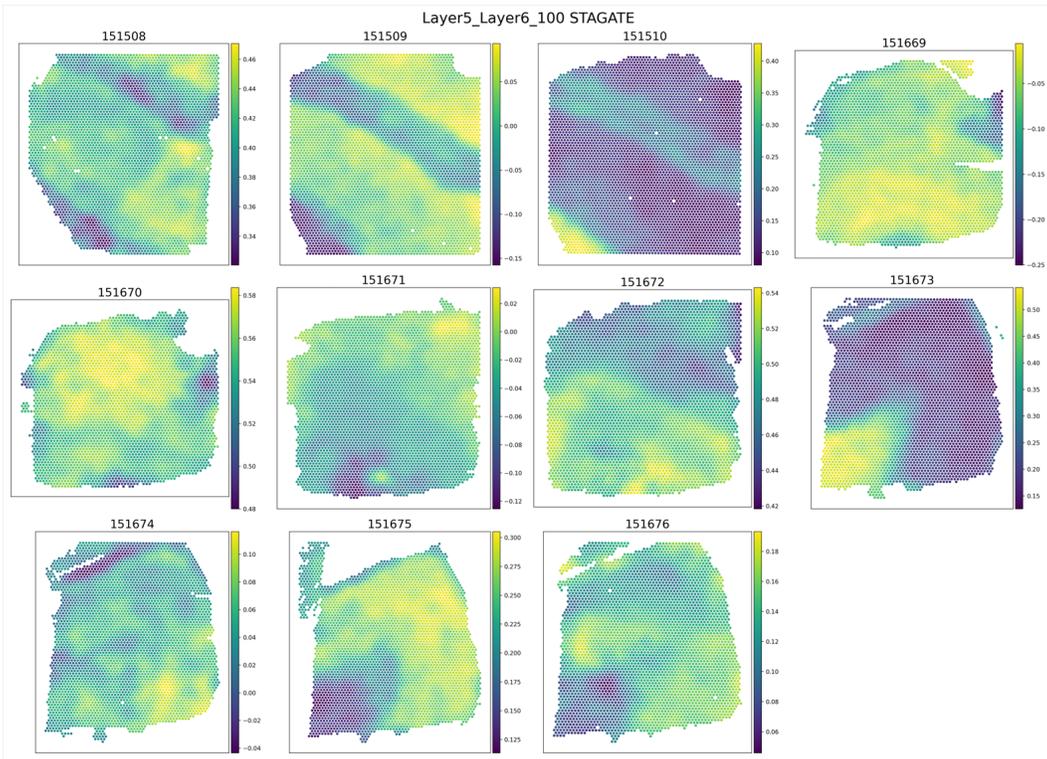


Figure A.8: Pearson Correlation of STAGATE on the DLPFC dataset

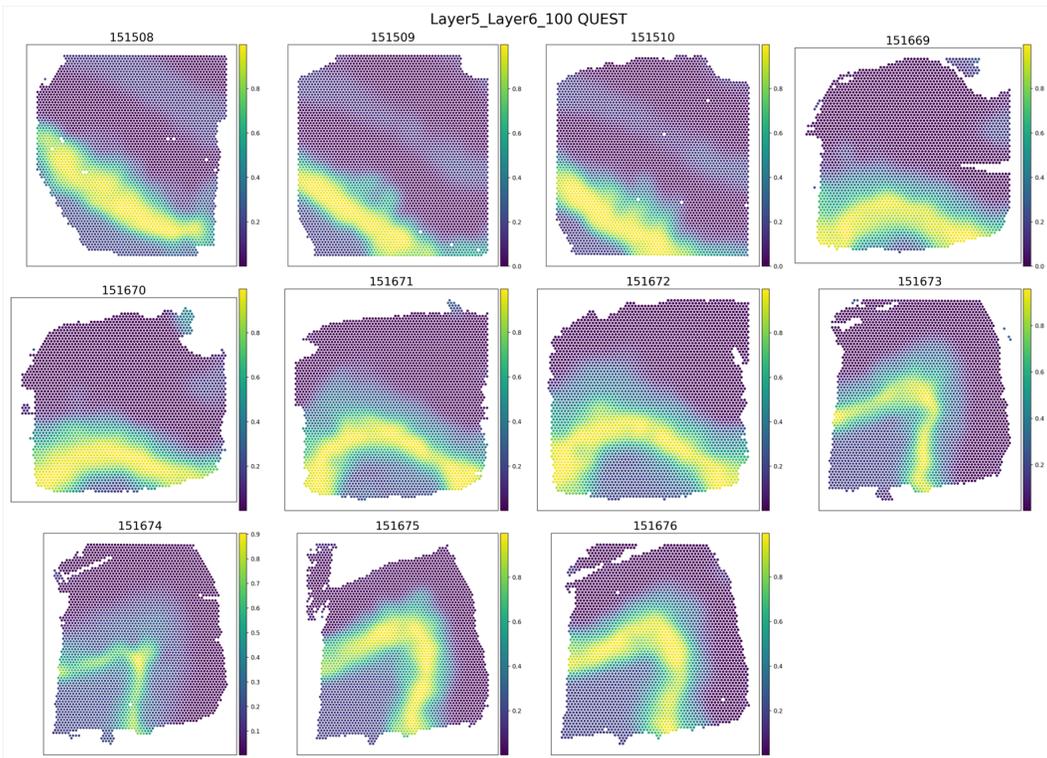


Figure A.9: Pearson Correlation of QUEST on the DLPFC dataset

A.4.2 PEARSON CORRELATION ON THE MOBT DATASET

1026  
1027  
1028  
1029  
1030  
1031  
1032  
1033  
1034  
1035  
1036  
1037  
1038  
1039  
1040  
1041  
1042  
1043  
1044  
1045  
1046  
1047  
1048  
1049  
1050  
1051  
1052  
1053  
1054  
1055  
1056  
1057  
1058  
1059  
1060  
1061  
1062  
1063  
1064  
1065  
1066  
1067  
1068  
1069  
1070  
1071  
1072  
1073  
1074  
1075  
1076  
1077  
1078  
1079

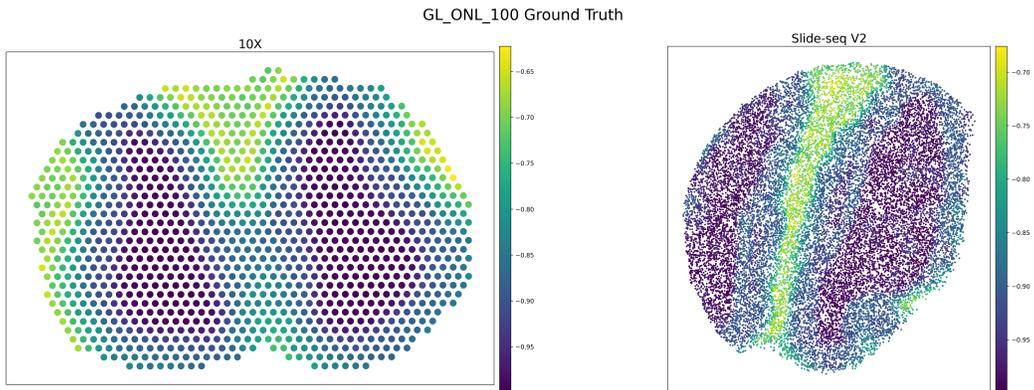


Figure A.10: Pearson Correlation of Ground Truth on the MOBT dataset

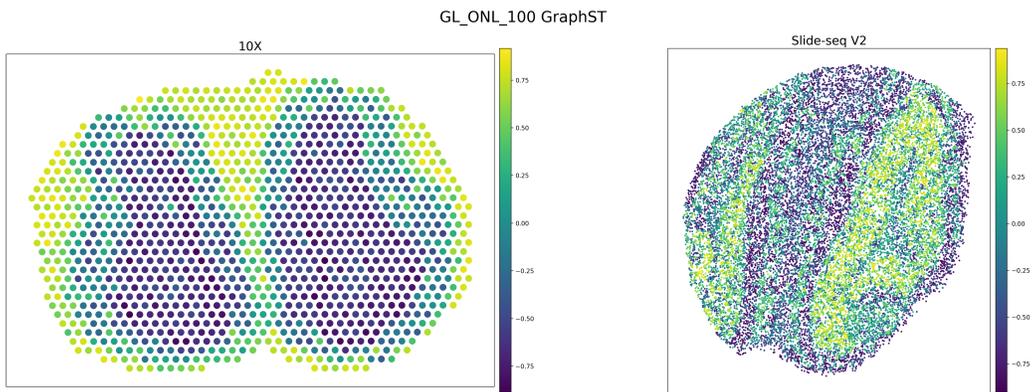


Figure A.11: Pearson Correlation of GraphST on the MOBT dataset

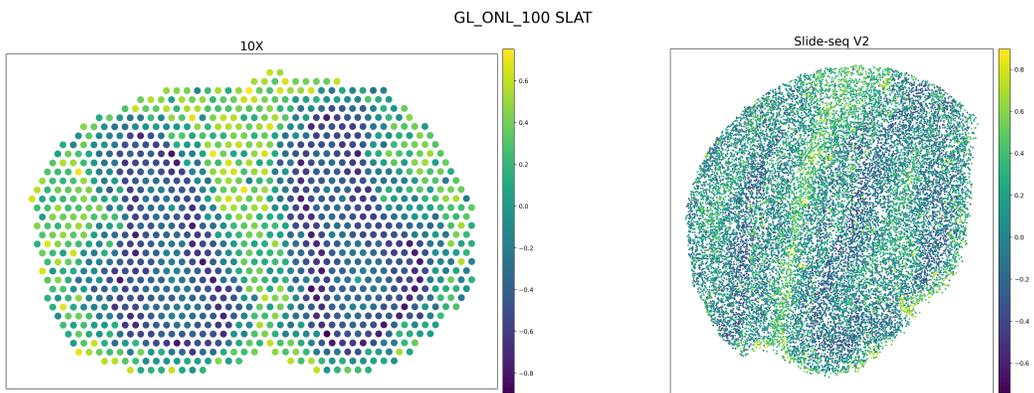


Figure A.12: Pearson Correlation of SLAT on the MOBT dataset

1080  
1081  
1082  
1083  
1084  
1085  
1086  
1087  
1088  
1089  
1090  
1091  
1092  
1093  
1094  
1095  
1096  
1097  
1098  
1099  
1100  
1101  
1102  
1103  
1104  
1105  
1106  
1107  
1108  
1109  
1110  
1111  
1112  
1113  
1114  
1115  
1116  
1117  
1118  
1119  
1120  
1121  
1122  
1123  
1124  
1125  
1126  
1127  
1128  
1129  
1130  
1131  
1132  
1133

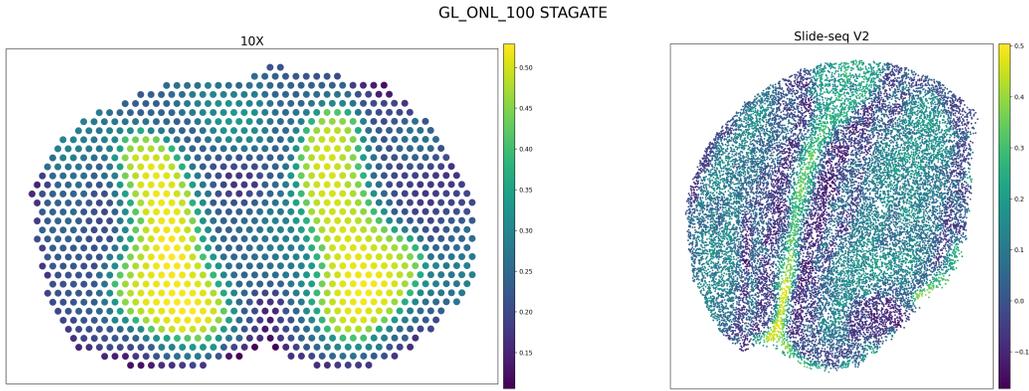


Figure A.13: Pearson Correlation of STAGATE on the MOBT dataset

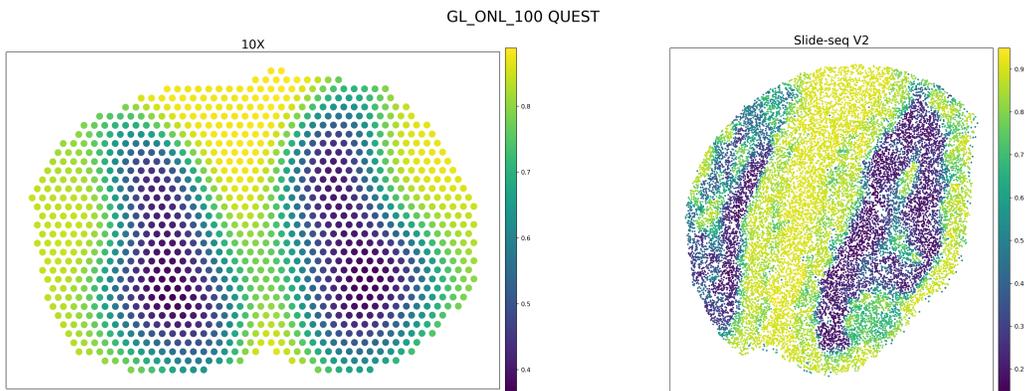


Figure A.14: Pearson Correlation of QUEST on the MOBT dataset