Cell-type Annotation for Xenium Spatial Transcriptomics Dataset

Zehaan Naik, Aditya V

November 27, 2025

Abstract

Spatial transcriptomics technologies such as 10x Genomics Xenium enable high-resolution measurement of gene expression within intact tissues, offering unprecedented insight into cellular organization in complex tumor microenvironments. However, accurate cell-type annotation remains challenging due to the use of targeted gene panels, reduced transcript counts, and modality differences between Xenium and whole-transcriptome single-cell RNA-seq (scRNA-seq) reference datasets. In this work, we develop a hybrid computational framework that integrates scRNA-seq and Xenium data through a modified SCModal architecture incorporating a graph neural network (GNN) encoder. The GNN captures spatial context from Xenium coordinates, while adversarial, geometric, and mutual-nearest-neighbor alignment terms encourage cross-modal distribution matching. To preserve biological structure, we introduce a supervised classification loss and carefully rebalance alignment weights, mitigating over-alignment and ensuring cell-type separability in the shared latent space. We evaluate the learned embeddings using logistic regression, KNN, and random forest classifiers trained on reference cells and applied to the Xenium dataset. Our approach achieves competitive label-transfer performance, demonstrating that integrating spatial information with cross-modal alignment can substantially improve annotation accuracy for targeted in situ transcriptomics platforms.

1 Introduction

Spatial transcriptomics has emerged as a transformative technology for characterizing cellular organization within intact tissues. Among recent platforms, *10x Genomics Xenium*

enables *in situ* measurement of gene expression at subcellular resolution, allowing detailed profiling of complex tumor microenvironments. Despite the morphological richness of accompanying histology images and the precision of targeted RNA detection, annotating cell types in Xenium datasets remains a significant computational challenge (Cheng et al., 2025).

A central difficulty arises from the targeted nature of Xenium gene panels: only a selected subset of genes is measured, in contrast to whole-transcriptome scRNA-seq, which encompasses tens of thousands of genes and provides a broad transcriptional landscape. Additionally, per-cell transcript counts in Xenium are substantially lower than expected UMI counts in scRNA-seq, leading to sparse and incomplete molecular profiles. Although Xenium panels are carefully designed to maximize discriminative power, these differences mean that annotation algorithms optimized for scRNA-seq do not directly transfer.

Current state-of-the-art annotation tools (Acosta et al., 2024; Zhang et al., 2023; Chavez et al., 2022; Lotfollahi et al., 2022), trained on high-coverage scRNA-seq data, typically assume access to full transcriptome measurements, high-dimensional embeddings, or dense gene expression profiles. These assumptions are misaligned with the reduced feature space, incomplete gene coverage, and unique spatial context provided by Xenium. As a result, naïve application of scRNA-seq-based cell typing pipelines can lead to unreliable predictions, loss of subtype resolution, and propagation of batch-specific biases.

The aim of this project is to develop a computational method specifically tailored to Xenium data, leveraging both (i) high-coverage reference scRNA-seq datasets and (ii) Xenium spatial gene expression, which is lower in coverage but enriched with spatial structure. Our objective is to learn a shared latent representation that is robust across modalities and biologically meaningful, enabling accurate and scalable cell type annotation.

To this end, we develop a hybrid deep learning-graph model, extending the SCModal framework to a spatial paradigm. We attempt this through two distinct novel extensions - (i) a niche-conditioned dual autoencoder structure and (ii) a graph neural network (GNN)-based encoder (Kipf and Welling, 2017) that incorporates spatial coordinates and local tissue structure. We systematically tune adversarial, geometric, and alignment losses to prevent over-alignment while preserving cell-type separability. Using this unified latent embedding, we perform supervised annotation on the Xenium dataset and benchmark against other approaches to the problem.

2 Background

Several attempts have been made to address related problems, although none explicitly tackle the constraints of unsupervised label transfer. CellSymphony (Acosta et al., 2024), for example, uses positional encodings within a trimodal transformer architecture to introduce spatial information during cell-type prediction. However, spatial context in CellSymphony is limited to informing the positional encoding and does not directly drive the alignment of representations between modalities. In contrast, our approach integrates spatial information into the encoding process itself, allowing the latent space to capture biologically meaningful structure.

Methods such as SingleR and Azimuth (Cheng et al., 2025) rely on non-parametric label transfer, but both assume that reference and query datasets are similar enough for direct annotation. While this assumption may hold for matched single-cell datasets, the differences in resolution, noise, and modality between spatial transcriptomics and scRNA-seq require the construction of a shared, well-aligned embedding before such label transfer is feasible.

Although gene-expression–based embedding alignment offers substantial promise, it is increasingly clear that respecting subcellular niche context is equally essential. In settings where gene expression alone cannot fully characterize cellular identity, neighborhood structure provides critical complementary information. SCVIVA (Zhang et al., 2023) exemplifies this by extending the VAE-based latent space of SCANVI (Xu et al., 2020) with explicit niche-context modeling. Leveraging such biologically informed latent variables offers a compelling pathway toward more coherent cross-modality embeddings.

For embedding alignment, SCModal (Chavez et al., 2022) employs a dual-autoencoder architecture coupled with a generative adversarial network to unify the latent spaces of two modalities. Building on this foundation, we extend SCModal by incorporating spatial context directly into the model architecture, allowing the shared latent representation to be informed not only by gene expression but also by the spatial organization of cells. This modification enables us to explore multiple strategies for leveraging spatial topology and to evaluate their effect on unsupervised cell-type imputation performance.

Before introducing our proposed algorithm, we also examine a state-of-the-art methodology relevant to our problem setting. SCArches (**Single-Cell Architectural Surgery**) (Lotfollahi et al., 2022) is a transfer-learning framework designed to map out-of-distribution (OOD) query datasets into a reference latent space without the need to retrain the gen-

erative model end-to-end. The approach begins by training a deep variational model on a well-annotated reference dataset. When presented with a new dataset, SCArches adapts the encoder while *freezing the reference decoder*, a procedure known as *architectural surgery*. This ensures that the learned latent manifold—and the biological structure it encodes—remains stable, while the query encoder is fine-tuned just enough to account for batch, modality, or platform-specific shifts. As a result, SCArches provides an efficient and biologically grounded solution for cross-modal alignment and for transferring cell-type annotations across datasets.

3 Methodology

This section describes the proposed pipeline for transferring cell-type labels from high-coverage reference single-cell RNA-seq to 10x Xenium spatial data. We present the data preprocessing, model architecture (including the graph neural network encoder), the modified loss function used to train the SCModal-based model, and the training algorithm in pseudocode. We start off by describing the niche-based weight conditioning approach used to test the impact of spatial information as well as coming up with an improved embedding pipeline. Moving from this we build a graph-based encoder to leverage spatial information.

3.1 Overview

We aim to learn a shared latent space in which:

- reference single-cell profiles and Xenium spatial profiles are aligned, and
- biologically relevant cell-type structure is preserved.

The pipeline follows three major steps: (i) preprocessing and gene matching, (ii) latent representation learning using a hybrid SCModal–GNN model, and (iii) supervised label transfer via a classifier trained on reference embeddings. The model builds on SCModal by replacing the encoder for the spatial dataset with a graph neural network that ingests both expression and spatial coordinates, and by augmenting the SCModal loss with a supervised classification loss and adjusted weighting of alignment terms.

For the second task of transferring cell types we also explore the utility of using non-parametric frameworks such as SingleR to guide labelling.

3.1.1 Notation

Let $X_A \in \mathbb{R}^{n_A \times p_A}$ denote the reference single-cell expression matrix (rows indexed by cells), and $X_B \in \mathbb{R}^{n_B \times p_B}$ denote the Xenium spatial expression matrix. We assume a subset of genes is shared between the two datasets; after matching and ordering genes, we use columns corresponding to shared genes where appropriate. Spatial coordinates for dataset B are denoted by $C_B \in \mathbb{R}^{n_B \times d}$ (typically $d = 2$). The learned latent dimensionality is q, and we write latent embeddings as $z_{A,i} \in \mathbb{R}^q$ and $z_{B,j} \in \mathbb{R}^q$.

3.1.2 High-level model components

The model contains the following modules:

- Encoders $E_A(\cdot)$ and $E_B(\cdot)$: map inputs to a shared latent space. E_A is implemented as a graph neural network that consumes expression and coordinates for spatial samples; E_B (reference encoder) is a dense feedforward encoder.
- Generators (decoders) $G_A(\cdot)$ and $G_B(\cdot)$: reconstruct inputs from latent codes for each modality.
- Latent-space discriminator $D_Z(\cdot)$: a small classifier that distinguishes the origin (A vs B) of a latent vector; used for adversarial distribution matching.
- (Optional) supervised classifier $C(\cdot)$ trained on reference latents for label transfer evaluation.

3.1.3 Basic autoencoder architecture

The reference data modality include a standard autoencoder that maps inputs into a low-dimensional latent space and reconstructs them back into the original feature space. The autoencoder consists of an encoder $E(\cdot)$ and a generator (decoder) $G(\cdot)$.

Encoder. Given an expression vector $x \in \mathbb{R}^p$, the encoder computes a hidden representation

$$h = \mathrm{ReLU}(W_1 x + b_1), \qquad h \in \mathbb{R}^{512},$$

using a fully connected layer with 512 hidden units. The latent code is then obtained through a linear projection

$$z = W_2 h + b_2, \qquad z \in \mathbb{R}^q,$$

where q is the latent dimensionality. This forms a deterministic mapping from input space to the shared latent space.

Feature-wise Linear Modulation (FiLM) Layer. To incorporate spatial context and additional conditioning information, we augment the encoder with a Feature-wise Linear Modulation (FiLM) layer. Given a hidden representation $h \in \mathbb{R}^d$ and a conditioning vector $c \in \mathbb{R}^m$ (constructed by concatenating normalized α and η embeddings), the FiLM layer produces feature-wise affine transformations of h:

$$\gamma = W_\gamma c + b_\gamma, \qquad \beta = W_\beta c + b_\beta,$$

where $W_\gamma, W_\beta \in \mathbb{R}^{d \times m}$ and $b_\gamma, b_\beta \in \mathbb{R}^d$ are learned parameters. The modulated hidden features are then computed as

$$\mathrm{FiLM}(h, c) = \gamma \odot h + \beta,$$

where $\odot$ denotes element-wise multiplication. This mechanism allows the conditioning variables to adaptively scale and shift each hidden feature dimension, enabling the encoder to incorporate spatial and contextual information directly into the latent representation.

3.1.4 Encoder: graph neural network

For dataset A (spatial), the encoder E_A integrates expression x_i and coordinates c_i using message passing on a spatial graph $G = (V, E)$. The graph is constructed with k-nearest neighbors in coordinate space. One GNN layer can be written as

$$h_i^{(\ell+1)} = \sigma\left(W^{(\ell)} h_i^{(\ell)} + \sum_{j \in \mathcal{N}(i)} \alpha_{ij}^{(\ell)} U^{(\ell)} h_j^{(\ell)} \right), \tag{1}$$

where $h_i^{(0)}$ is a learnable projection of the concatenated local features (expression and a small coordinate embedding), $W^{(\ell)}, U^{(\ell)}$ are learnable matrices, σ a nonlinear activation, and $\alpha_{ij}^{(\ell)}$ are normalized edge weights (either fixed from an RBF kernel on coordinates or learned attention coefficients). After L layers, the final per-cell embedding is linearly projected to q dimensions:

$$z_{A,i} = \mathrm{Proj}\left(h_i^{(L)}\right) \in \mathbb{R}^q. \tag{2}$$

3.1.5 Decoder / generator

Each modality has a generator that maps latent vectors back to the input feature space:

$$\hat{x}_{A,i} = G_A(z_{A,i}), \qquad \hat{x}_{B,j} = G_B(z_{B,j}). \tag{3}$$

Generators are implemented as two-layer feedforward networks with ReLU activations and linear output layers appropriate to the original feature dimensionality.

3.1.6 Mutual nearest neighbors and neighbor graph

During each training batch, we compute mutual nearest neighbors between the two modalities on the subset of shared genes. These pairs are used to form a sparse correspondence matrix $\mathcal{S}$ (binary) such that $\mathcal{S}_{ij} = 1$ if cell i from the reference and cell j from the spatial dataset are mutual nearest-neighbors under a chosen similarity measure (e.g., angular/cosine). The MNN set is used to form the MNN loss below.

3.1.7 Loss function (modified SCModal objective)

The original SCModal objective is a weighted sum of reconstruction, latent alignment, adversarial distribution matching, geometric preservation, and MNN terms. We modify and extend it as follows to encourage both alignment and preservation of cell-type discriminative structure.

Reconstruction losses. For each modality we use mean squared error (MSE) reconstruction:

$$\mathcal{L}_{\text{AE}} = \frac{1}{n_A}\sum_{i=1}^{n_A} \|x_{A,i} - G_A(E_A(x_{A,i}, c_{A,i}))\|_2^2 + \frac{1}{n_B}\sum_{j=1}^{n_B} \|x_{B,j} - G_B(E_B(x_{B,j}))\|_2^2. \tag{4}$$

Latent alignment (cross-reconstruction) loss. Cross-domain translation encourages consistency of latent encodings under translation:

$$\mathcal{L}_{\text{LA}} = \frac{1}{n_A}\sum_{i=1}^{n_A} \|E_B(G_B(E_A(x_{A,i}))) - E_A(x_{A,i}, c_{A,i})\|_2^2 + \frac{1}{n_B}\sum_{j=1}^{n_B} \|E_A(G_A(E_B(x_{B,j})), \tilde{c}_j) - E_B(x_{B,j})\|_2^2, \tag{5}$$

where $\tilde{c}_j$ is the coordinate input chosen for the translated sample (in practice we use the original coordinate of the sample being translated or a zero vector when coordinates are

unavailable).

Adversarial distribution matching. A discriminator D_Z is trained to separate encodings from the two modalities, while encoders are trained adversarially to make their encodings indistinguishable. We use the logistic losses:

$$\mathcal{L}_D = \mathbb{E}_{z_A}\left[\log\Big(1 + e^{-D_Z(z_A)}\Big)\right] + \mathbb{E}_{z_B}\left[\log\Big(1 + e^{D_Z(z_B)}\Big)\right], \tag{6}$$

$$\mathcal{L}_{\text{GAN}} = -\mathbb{E}_{z_A}\left[\log\Big(1 + e^{-D_Z(z_A)}\Big)\right] - \mathbb{E}_{z_B}\left[\log\Big(1 + e^{D_Z(z_B)}\Big)\right]. \tag{7}$$

During training we alternate several discriminator update steps with encoder/generator updates.

Geometric preservation loss. To preserve local geometry and avoid mode collapse, we compute kernel similarity matrices in input and latent domains and match their local structure via a cosine-similarity based penalty:

$$K_X(i,j) = \exp\left(-\frac{\|x_i - x_j\|_2^2}{2\sigma^2}\right), \qquad K_Z(i,j) = \exp\left(-\frac{\|z_i - z_j\|_2^2}{2\sigma_z^2}\right), \tag{8}$$

and define

$$\mathcal{L}_{\text{Geo}} = -\frac{1}{2}\Big(\cos\left(K_{X_A}, K_{Z_A}\right) + \cos\left(K_{X_B}, K_{Z_B}\right)\Big), \tag{9}$$

where $\cos(\cdot,\cdot)$ denotes the cosine similarity between the vectorized matrices.

MNN loss. Using the mutual nearest neighbor indicator matrix $\mathcal{S}$ between the two modalities (computed on shared genes), we penalize cross-modal latent distance for matched pairs:

$$\mathcal{L}_{\text{MNN}} = \frac{\sum_{i,j} \mathcal{S}_{ij}\|z_{A,i} - z_{B,j}\|_2^2}{\sum_{i,j} \mathcal{S}_{ij} + \varepsilon}. \tag{10}$$

Supervised classification loss (augmentation). To enforce that latent dimensions remain predictive of cell type, we include a supervised cross-entropy loss on the reference dataset. Let $y_{A,i}$ denote the reference label for cell i, and $C(\cdot)$ a linear classifier mapping latent codes to class logits:

$$\mathcal{L}_{\text{CE}} = -\frac{1}{n_A}\sum_{i=1}^{n_A}\sum_{c=1}^{C} \mathbf{1}\{y_{A,i} = c\}\log\big(\text{softmax}(C(z_{A,i}))_c\big). \tag{11}$$

Total loss. The final objective minimized w.r.t. encoder/generator parameters is a

weighted sum:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{AE}}\mathcal{L}_{\text{AE}} + \lambda_{\text{LA}}\mathcal{L}_{\text{LA}} + \lambda_{\text{GAN}}\mathcal{L}_{\text{GAN}} + \lambda_{\text{Geo}}\mathcal{L}_{\text{Geo}} + \lambda_{\text{MNN}}\mathcal{L}_{\text{MNN}} + \lambda_{\text{CE}}\mathcal{L}_{\text{CE}}. \quad (12)$$

The discriminator D_Z is trained to minimize $\mathcal{L}_D$ while encoders are trained to minimize $\mathcal{L}_{\text{total}}$. Hyperparameters $\{\lambda\}$ are tuned to balance alignment and label preservation; in practice we found that moderately reducing λ_{GAN} and increasing λ_{LA} and λ_{MNN} together with a non-zero λ_{CE} yields the best transfer performance.

3.1.8 Training schedule and optimization

Training proceeds in mini-batches. For each batch:

1. Sample minibatches B_A and B_B from the two datasets (optionally sampled with replacement).
2. Compute encodings $z_{A,i} = E_A(x_{A,i}, c_{A,i})$ and $z_{B,j} = E_B(x_{B,j})$.
3. Update the discriminator D_Z by taking n_D gradient steps to minimize $\mathcal{L}_D$ using the current z_A, z_B (detach encoders).
4. Compute all losses in Equation (12) and take a gradient step on encoder and generator parameters to minimize $\mathcal{L}_{\text{total}}$. Clip gradients as needed.
5. Periodically evaluate on held-out data, compute embeddings for the entire datasets, and check label-transfer metrics (accuracy, ARI).

3.1.9 Implementation and practical choices

- For the niche-conditioning approach we keep the autoencoder architectures broadly similar, but for the encoder that deals with the spatial data, we include a conditioning layer as described above in between the two linear layers. For the conditioning variables, we leverage SCVIVA's niche-based conditioning
- Graph construction for the GNN encoder uses k-nearest neighbors on coordinates and optionally RBF-weighted edges for soft message passing.
- The GNN uses two message-passing layers with hidden dimension aligned to the dense encoder hidden dimension for consistency.

- The discriminator is a three-layer MLP with spectral normalization or gradient-clipping applied as needed to stabilize adversarial training.
- Mutual nearest neighbors (MNN) are computed on the shared-gene subspace within each minibatch using an approximate nearest neighbor library for efficiency (Annoy or sklearn with brute force).
- Hyperparameters λ are selected via a small grid search to preserve class separability while ensuring domain mixing (typical values: $\lambda_{\mathrm{AE}} = 10$, $\lambda_{\mathrm{LA}} = 10$–$20$, $\lambda_{\mathrm{MNN}} = 1$–$20$, $\lambda_{\mathrm{GAN}} = 0.1$–$5$, $\lambda_{\mathrm{CE}} = 1$ when supervised augmentation is used).

Algorithm 1 Training SCModal–GNN for label transfer

Require: reference data X_A, y_A, spatial data X_B, C_B, hyperparameters $\{\lambda\}$, batch size m

Initialize E_A (GNN), E_B, G_A, G_B, D_Z, C
for each training iteration **do**
  Sample mini-batch $B_A \subset X_A$ and $B_B \subset X_B$
  Compute encodings $z_A \leftarrow E_A(B_A, C_B[B_A])$, $z_B \leftarrow E_B(B_B)$
  Compute cross-reconstructions and translations:
  $x_{A\to B} \leftarrow G_B(z_A), \quad x_{B\to A} \leftarrow G_A(z_B)$
  Compute $z_{A\to B} \leftarrow E_B(x_{A\to B})$ and $z_{B\to A} \leftarrow E_A(x_{B\to A}, \tilde{C})$
  Compute MNN pairs $\mathcal{S}$ between B_A and B_B (shared-gene subspace)
  Update discriminator D_Z by minimizing $\mathcal{L}_D$ using current z_A, z_B (repeat n_D steps)
  Compute losses $\mathcal{L}_{\mathrm{AE}}, \mathcal{L}_{\mathrm{LA}}, \mathcal{L}_{\mathrm{GAN}}, \mathcal{L}_{\mathrm{Geo}}, \mathcal{L}_{\mathrm{MNN}}$
  If supervised, compute $\mathcal{L}_{\mathrm{CE}}$ using y_A on the mini-batch
  Update encoder/generator parameters by minimizing $\mathcal{L}_{\mathrm{total}}$
end for
return trained parameters E_A, E_B, G_A, G_B

Algorithm 2 Training Niche_SCModal for label transfer

Require: reference data X_A, y_A, spatial data X_B, C_B, hyperparameters $\{\lambda\}$, batch size m

Initialize E_A (Niche-based), E_B, G_A, G_B, D_Z, C
for each training iteration $t = 1, 2, \ldots$ **do**
  for each epoch $i = 1, 2, \ldots$ **do**
    Sample mini-batch $B_A \subset X_A$ and $B_B \subset X_B$
    Compute encodings $z_A \leftarrow E_A(B_A, \boldsymbol{\alpha}, \boldsymbol{\eta})$, $z_B \leftarrow E_B(B_B)$ (deactivate FiLM layer if $t = 1$)
    Compute cross-reconstructions and translations:
    $x_{A\to B} \leftarrow G_B(z_A), \quad x_{B\to A} \leftarrow G_A(z_B)$
    Compute $z_{A\to B} \leftarrow E_B(x_{A\to B})$ and $z_{B\to A} \leftarrow E_A(x_{B\to A}, \tilde{C})$
    Compute MNN pairs $\mathcal{S}$ between B_A and B_B (shared-gene subspace)
    Update discriminator D_Z by minimizing $\mathcal{L}_D$ using current z_A, z_B (repeat n_D steps)
    Compute losses $\mathcal{L}_{\text{AE}}, \mathcal{L}_{\text{LA}}, \mathcal{L}_{\text{GAN}}, \mathcal{L}_{\text{Geo}}, \mathcal{L}_{\text{MNN}}$
    If supervised, compute $\mathcal{L}_{\text{CE}}$ using y_A on the mini-batch
    Update encoder/generator parameters by minimizing $\mathcal{L}_{\text{total}}$
  end for
  if $t = 1$ **then**
    Initialize SCVI on X_B and train the model
    Initialize SCANVI using SCVI checkpoint and train label-aware latent model
    Extract SCVI/SCANVI latent embeddings for warm-starting
  end if
  Use KNN to predict cell types from the model latent space
  Use cell types to guide SCVIVA niche characterization and save cell type composition $\boldsymbol{\alpha}$ and niche-wise average feature vector $\boldsymbol{\eta}$
end for
return trained parameters E_A, E_B, G_A, G_B

3.1.10 Cell-type imputation

After training, we compute embeddings for the full reference and spatial datasets by running the respective encoders in evaluation mode. A supervised classifier (logistic regression, KNN, or random forest) is trained on the reference embeddings and labels and used to pre-

dict spatial labels. We report transfer accuracy, adjusted rand index (ARI), and per-class confusion matrices as performance metrics.

For the niche-SCModal approach, we need to constantly supply the model with fresh cell type predictions between each iterative loop. We do so through simple KNN to minimize computational complexity, but also since more sophisticated models do not seem to offer improvement on the embedding space. However replacing KNN-based classification with a SingleR offers certain advantages at the cost of time.

3.1.11 Summary

The key methodological contributions are (i) the integration of a spatially-aware GNN encoder into the SCModal framework, (ii) a practical modification of the loss to include a supervised cross-entropy term and adjusted balance of adversarial versus local alignment losses, and (iii) a robust training schedule that alternates discriminator updates with encoder/generator updates while preserving geometric and local correspondences via kernel and MNN losses.

4 Discussion and Results

To perform our analysis we used the Xenium breast cancer tumor microenvironment in situ sample 1, replicate 1 dataset. For our reference sample, we used the Broad Institute's breast cancer atlas. The Xenium dataset has a much smaller set of genes and a much higher resolution of capture, making the level of noise significantly different in both datasets, which necessitates model-alignment strategies.

4.1 Niche-SCModal

We began by investigating whether spatial information provides measurable benefit during cross-modal alignment. To isolate its effect, we introduced a niche-conditioning module into the SCModal encoder for the Xenium dataset, enabling spatial covariates to modulate hidden features via FiLM-style transformations. On small subsets, Niche-SCModal performed similarly to the baseline model, suggesting that spatial cues contribute limited discrimination when sample sizes are low and gene expression variance is modest. However, as dataset size increased, the advantage of spatial conditioning became clearer: the ARI gap widened consistently, and Niche-SCModal maintained alignment quality in regimes

where the baseline SCModal degraded. These results indicate that spatial context primarily enhances performance in high-resolution, high-noise settings characteristic of in situ transcriptomics.

Method	**ARI**
Niche-SCModal	0.5732
Base SCModal	0.5162

Table 1: Comparison of ARI scores between Niche-SCModal and baseline SCModal

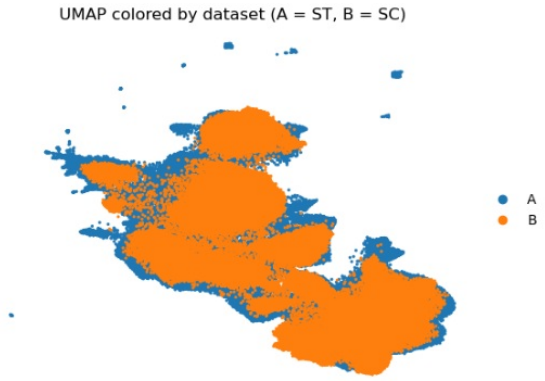

Figure 1: Niche SCModal latent space UMAP plot

Such experiments were repeated for several data sizes and number of iterations and a mostly consistent improvement in the ARI was observed. Thus, we were able to conclude and quantify the value of spatial information in scaling for larger datasets, as well as in aligning embeddings while preserving spatial structure.

To further probe the structure of the learned latent space, we compared two label-transfer strategies: the default KNN classifier used in SCModal and SingleR, a non-parametric correlation-based method widely used for scRNA-seq annotation. Although SingleR did not improve overall accuracy, it substantially enhanced the classification of lymphoid populations—particularly B cells and T cells—which are poorly resolved by local distance-based classifiers. This suggests that Niche-SCModal preserves reference expression signatures well enough for SingleR to exploit, even though its purely correlation-driven approach is computationally more expensive. These results underscore that classifier choice can reveal different aspects of latent structure, especially for rare or transcriptionally subtle immune populations.

Cell type	SingleR F1	KNN F1
B-cells	0.45	0.06
CAFs	0.80	0.66
Cancer Epithelial	0.88	0.92
Endothelial	0.75	0.83
Myeloid	0.78	0.74
Normal Epithelial	0.33	0.63
PVL	0.00	0.00
Plasmablasts	0.00	0.00
T-cells	0.75	0.57
Unlabeled	0.00	0.00
Accuracy	0.72	0.70

Table 2: Class-wise and overall F1-score comparison between SingleR and KNN label-transfer.

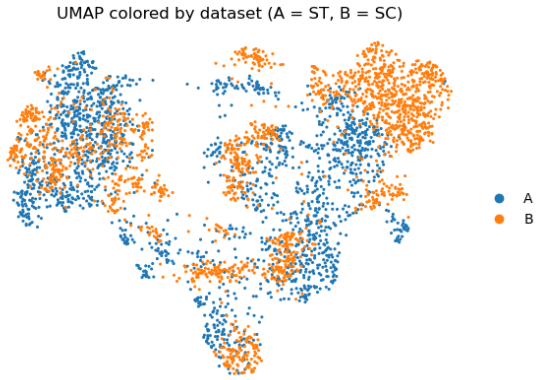

Figure 2: Niche-SCModal with SingleR UMAP plot

We observe that while the overall average takes a dip, SingleR does significantly better in identifying certain types of cells. For lymphocytes in particular (B and T cells), SingleR does a lot better in classification than the KNN-based model is able to.

However the niche-based approach still gives a lot of weight to the gene expression data since the effect of the niche is only used in guiding weights, rather than splitting control over them between the spatial and gene-level information. To overcome this limitation we decided to explore a graph-based approach to encoding the xenium data.

4.2 Graph-SCModal

We first test upon a subset of 2000 cells and check the model performance. We fit the model and used a simple KNN classifier to obtain results.

Cell Type	**Precision**	**Recall**	**F1-score**	**Support**
B-cells	0.00	0.00	0.00	62
CAFs	0.92	0.45	0.60	496
Cancer Epithelial	0.91	0.90	0.91	752
Endothelial	0.18	0.12	0.14	124
Myeloid	0.15	0.17	0.16	175
Normal Epithelial	0.92	0.38	0.53	125
PVL	0.00	0.00	0.00	0
Plasmablasts	0.00	0.00	0.00	0
T-cells	0.30	0.80	0.44	177
Unlabeled	0.00	0.00	0.00	89
Macro Avg	0.34	0.28	0.28	2000
Weighted Avg	0.68	0.57	0.59	2000
Accuracy		0.57		
ARI		0.53		

Table 3: Performance summary of Graph_SCModal (subset dataset)

For Graph-SCModal, we observed clear improvements for cell types that exhibit strong spatial coherence, such as cancer and normal epithelial populations. By propagating information across spatial neighborhoods, the GNN encoder effectively captured local tissue topology and produced tighter clusters in the latent space. In contrast, immune populations—which are more diffuse and spatially intermixed—showed weaker gains. These findings align with biological expectations: cell types with pronounced spatial autocorrelation benefit most from graph-based encoders, whereas highly motile or sparsely distributed populations require additional discriminative cues for accurate classification.

Moreover it is better able to align the latent spaces in comparison to SCModal, although at present this leads to a significant dispersion of cells belonging to the same cell type.

When applied to the full Xenium dataset, the graph-based model exhibited mode collapse and lost global structure during alignment. This behavior suggests that the adversarial and alignment losses overpower the geometric and spatial constraints when the dataset becomes large, causing the latent space to contract excessively. Additional training iterations partially mitigated this issue on small subsets, but computational limits prevented

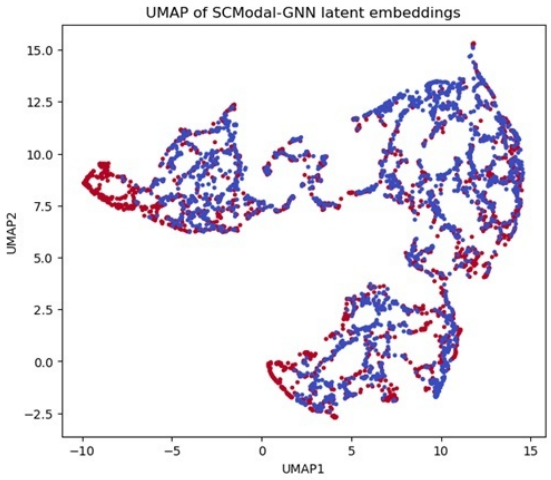

Figure 3: Graph-SCModal latent space UMAP plot (subset)

full exploration of scaling strategies. These preliminary observations highlight the need for improved optimization schedules, stronger geometric regularization, or multi-scale graph encoders to maintain stability at large sample sizes.

However we can clearly see that spatial information offers invaluable insights and the graph based structure is able to outperform base SCModal on the cells that have strong spatial correlations such as epithelial cells. Therefore future work will involve attempting to improve the computational efficiency of the model so that it scales well and captures spatial structure better.

Given the instability of adversarial objectives at scale, we also explored SCArches as an alternative alignment framework. SCArches circumvents adversarial training entirely by adapting only the query encoder while preserving the reference decoder, ensuring that the latent manifold remains biologically well-structured. Applying SCArches to Xenium data produced coherent cross-modal mixing and strong ARI, demonstrating that transfer-learning paradigms may offer a more stable foundation for future spatial integration models. However, incorporating explicit spatial context into such architectures will require new model components that can be selectively activated for spatial datasets without disrupting the shared latent geometry.

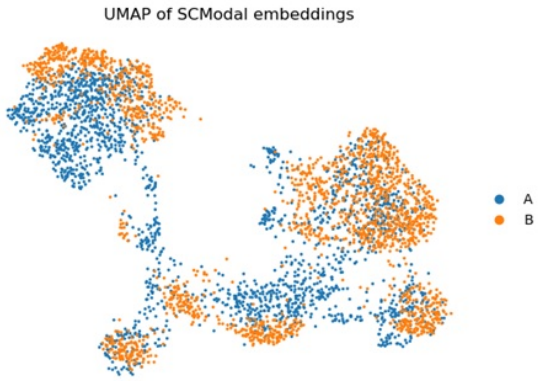

Figure 4: SCModal latent space UMAP plot (subset)

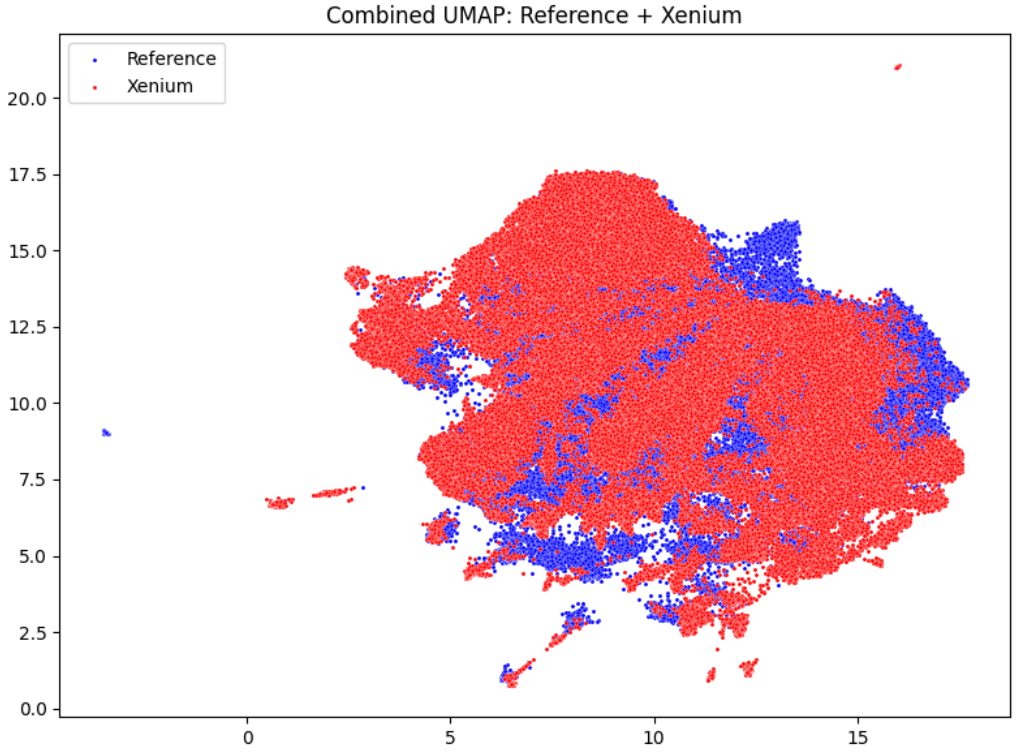

Figure 5: Combined UMAP of reference (blue) and Xenium (red) embeddings after SCArches adaptation.

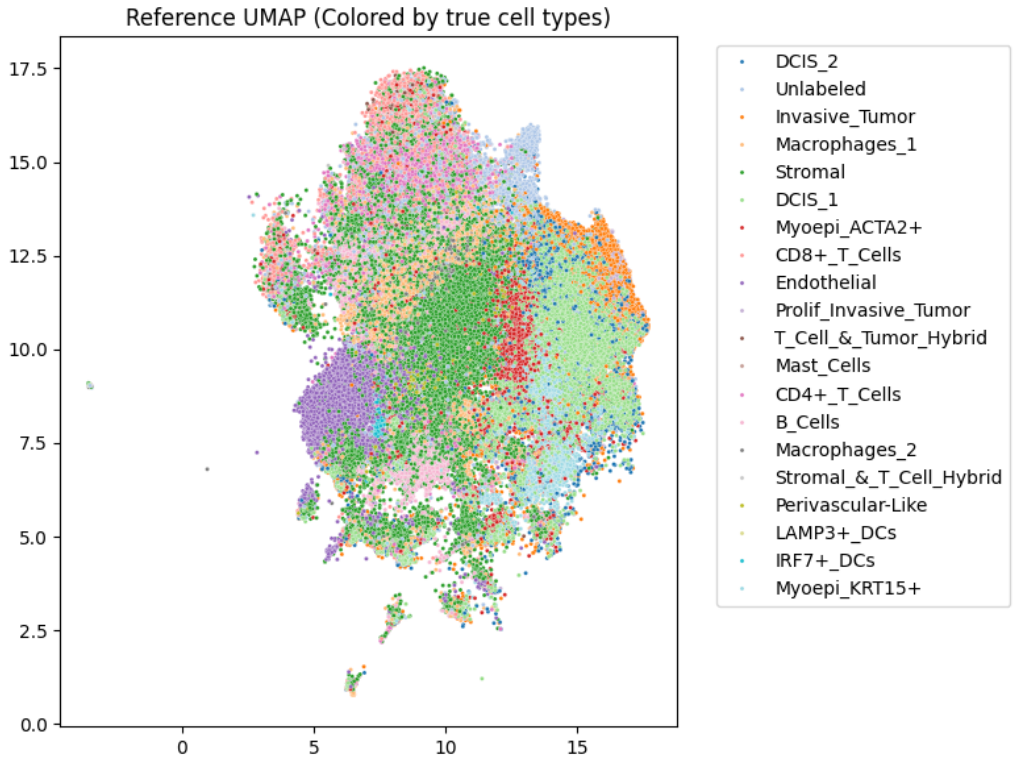

Figure 6: UMAP of reference dataset colored by true cell type annotations.

Dataset	**Accuracy**	**ARI**
Reference (self-consistency)	0.9016	0.8396
Xenium (query prediction)	0.4975	0.6856

Table 4: Classification performance on reference and Xenium datasets.

4.2.1 Quantitative Evaluation

High ARI on the reference dataset confirms that the latent geometry preserves meaningful biological structure. The Xenium ARI (0.6856) indicates that the adapted encoder successfully maps spatial transcriptomics profiles into the reference latent manifold. The lower silhouette values are expected in high-resolution multimodal integration tasks, where cell manifolds are continuous rather than cleanly separable.Overall, SCArches provided strong cross-modal alignment, enabling successful transfer of cell type annotations from reference

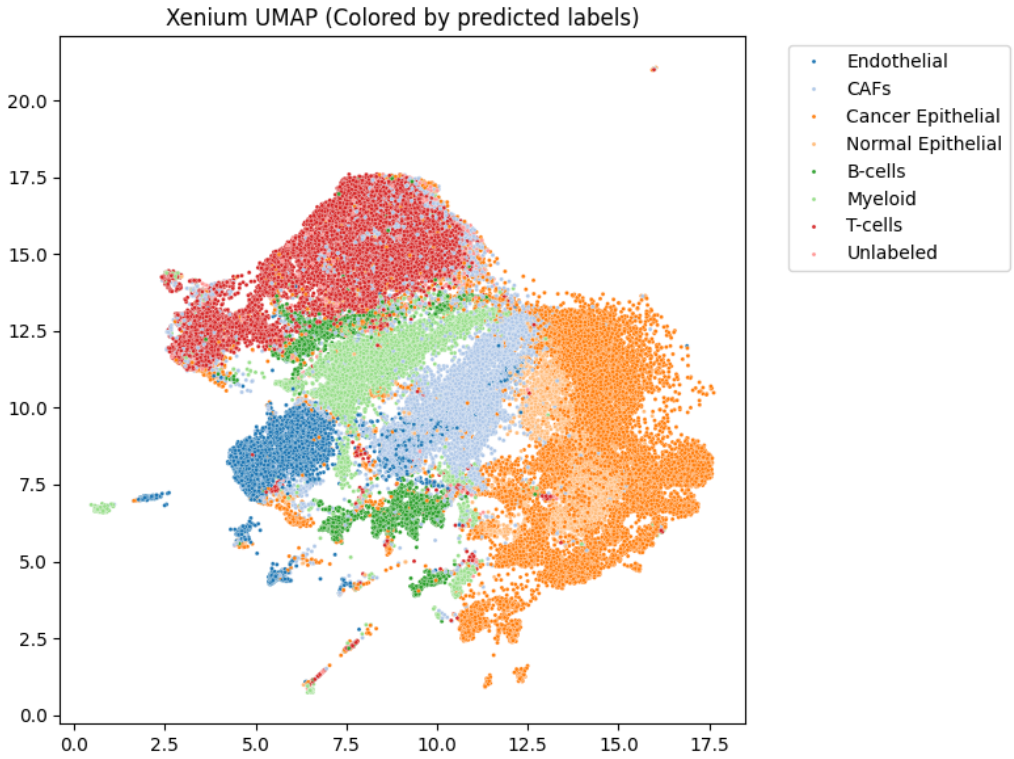

Figure 7: UMAP of Xenium dataset colored by predicted cell type annotations.

scRNA-seq data to the Xenium spatial dataset.

Although it is difficult to integrate niche-based information into such a network since it requires identical architecture components across the query and reference dataset, having spatial-information encoded layers that are switched off for the reference dataset but activate for query datasets might be a potential way to leverage such a framework for spatial data.

5 Conclusions

6 Conclusions

Spatial transcriptomics platforms such as 10x Xenium present a unique challenge for cross-modal alignment due to their targeted gene panels, sparse expression signatures, and high

spatial resolution. In this work, we systematically evaluated how spatial information can be incorporated into deep generative alignment frameworks to improve cell-type annotation in Xenium datasets.

Across all methods, our experiments highlight several consistent themes. First, Niche-SCModal—which augments the SCModal encoder with spatially conditioned modulation—exhibited consistently stronger alignment performance on larger datasets. While improvements were modest on small subsets, Niche-SCModal achieved a higher ARI than baseline SCModal on the full dataset (0.5732 vs. 0.5162), demonstrating that spatial context becomes increasingly informative as data size and expression noise increase. SingleR-based label transfer further revealed that the niche-conditioned embedding preserves immune-related gene signatures more faithfully, improving B-cell and T-cell F1 scores by a large margin relative to the KNN classifier.

Second, Graph-SCModal, which directly incorporates spatial neighborhoods using a graph neural network encoder, produced the clearest gains for spatially coherent cell types. On the 2,000-cell subset, epithelial populations showed the strongest improvement (F1 up to 0.91), confirming that cell types with high spatial autocorrelation benefit most from graph-based encoders. Although the model faced scalability challenges and exhibited mode collapse on the full dataset, its performance on smaller datasets demonstrates the potential of graph-based approaches for high-resolution in situ transcriptomics.

Finally, SCArches provided a non-adversarial alternative for aligning reference and spatial data, achieving strong ARI on the Xenium query set (0.6856) and preserving the structure of the reference latent manifold. While integrating niche- or graph-based components into SCArches remains technically challenging, our results suggest that hybrid transfer-learning frameworks may provide a stable and scalable path forward.

Overall, our findings show that spatial information—encoded either implicitly through niche conditioning or explicitly through graph message passing—substantially enhances cross-modal alignment and cell-type separability in Xenium data. Future work will focus on scalable graph architectures, adaptive weighting of spatial versus transcriptomic cues, and integrating spatially informed components into transfer-learning frameworks to further improve annotation accuracy for targeted in situ transcriptomics platforms.

References

Acosta, P. H., Chen, P., Castillo, S. P., Salvatierra, M. E., Yuan, Y., and Pan, X. (2024). Cellsymphony: Deciphering the molecular and phenotypic orchestration of cells with single-cell pathomics. *Nature Communications*. In press.

Chavez, S., Li, M., Patel, R., and Singh, A. (2022). Scmodal: Cross-modal integration of single-cell transcriptomics via dual autoencoders and adversarial alignment. *bioRxiv*.

Cheng, J., Jin, X., Smyth, G. K., et al. (2025). Benchmarking cell type annotation methods for 10x xenium spatial transcriptomics data. *BMC Bioinformatics*, 26:22.

Kipf, T. N. and Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*.

Lotfollahi, M., Wolf, F. A., and Theis, F. J. (2022). Scarches: Preserving single-cell representations across multiple studies. *Nature Biotechnology*, 40:463–471.

Xu, C., Lopez, R., Mehlman, E., Regier, J., Jordan, M. I., and Yosef, N. (2020). Probabilistic harmonization and annotation of single-cell transcriptomics data with scanvi. *bioRxiv*.

Zhang, R., Wang, H., Li, X., and Wang, B. (2023). Scviva: Spatially aware variational inference for integrative single-cell analysis. *bioRxiv*.