Statistical Inference for Spatial Transcriptomics in the Age of Deep Learning

by

Roman Kouznetsov

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Statistics)
in the University of Michigan
2025

Doctoral Committee:

Assistant Professor Jeffrey Regier, Chair
Associate Professor Johann A. Gagnon-Bartsch
Associate Professor Hyun Min Kang
Associate Professor Jonathan Terhorst

Roman Kouznetsov
roko@umich.edu
ORCID iD: 0009-0001-2147-3590

ACKNOWLEDGEMENTS

This work has been made possible by the tremendous support I received throughout my studies, and I am grateful and deeply indebted to everyone who contributed to this journey.

First and foremost, I extend my gratitude to my advisor, Dr. Jeffrey Regier, and my co-advisor in spirit, Dr. Jackson Loper. It has been an honor to work alongside and learn from them; their expertise and mentorship have been instrumental in shaping my research. Their enthusiasm and support not only strengthened my confidence but also fueled my passion for interdisciplinary work. I appreciate their motivation, compassion, support, and valuable contributions to our work together. From both of them, I have learned so much about statistics, project design, proper coding practices, machine learning, and how to stay at the cutting edge of the field. I am excited to transfer all that I have gleaned from their mentorship in my next chapter.

I am profoundly grateful to my family. My mother (Elena Kuznetsov), father (Sergei Kuznetsov), sisters (Leeza Kuznetsov and Alisa Kuznetsov), and grandmother (Serafima Krivolapova) have invested so much in me and have been steadfast in their love and support throughout this journey. Their belief in me has been a source of strength in times of doubt, and I owe them more than words can express. Their consistent outpouring love, words of wisdom, persistent desire to help, and support were the foundation upon which I have ever managed to complete this endeavor—or anything, for that matter.

To my friends, a formative video game quote of my childhood was, "My friends are my power." So, I thank you all for being such a strong and consistent source of it. There are too many to thank in this outing but all of you know who you are. Among the many who have cheered me on, I would like to single out Erik Craig, Travis Pavletich, Rueben Dockery, Eric Maccaro, Rohit Mishra, Patrick Hogrell, Maria Sidulova, Elijah Fourre, Thomas Nemec, Sunag Udupa, Adam Boldenow, Gregory Star, Turner Gunderson, Austin Sparkman, Michael Huynh, and Alexander Kan for always making time for me. Keeping in touch with all of you has been incredibly motivating; your words of encouragement and the knowledge that I could turn to any of you at a moment's notice mean the world to me.

Of the many things I gained during my time at the University of Michigan, the best has been the lifelong friends I made within the statistics department. I thank them all for being

an intellectual and emotional support system. In particular, I want to thank my cohort friends—Seamus Somerstep, Gabriel Durham, Derek Hansen, Moritz Korte-Stapff, Brian Manzo, Luke Francisco, Declan McNamara, and Simon Fontaine—who have been like older brothers, providing guidance, camaraderie, and countless memories along the way. A stand-out thank you to Simon Fontaine for opening his home to me during the pandemic, being a consistent reviewer of everything I have ever written because there is no one's criticism I value more than his, and answering every question no matter how silly it was at the time of proposition.

Finally, to my partner, Reed Momjian, your support has been the most valuable thing I have ever had. You believed in me unconditionally, especially in times when I did not believe in myself. Through every win, setback, breakdown, and emotion, I had you by my side. In a world of chaos, you were the constant. I have studied the concept of luck for many years, and no matter how long I study the discipline, I will never understand how I got so lucky to have you. I am forever grateful to have you.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE

LIST OF TABLES

TABLE

LIST OF APPENDICES

ABSTRACT

Single-cell spatial transcriptomics enables the measurement of gene expression of individual cells while simultaneously capturing the spatial positions of these cells within a tissue sample. To utilize these spatial positions effectively, careful model selection is required to ensure conclusions reflect spatial dependencies in the underlying biology. In this dissertation, we contribute three novel methodologies that merge deep learning with statistical inference for spatial transcriptomics data.

First, we attempt to better predict gene expression by leveraging the spatial context included in spatial transcriptomics data. Comparing predictions from a spatial model to those from a baseline regressor without cell neighborhood information offers insights into how expression changes as a result of cell-cell communication (CCC) signals. However, to trust conclusions reached from such a paired modeling framework, we need to ensure that the baseline version of a model provides a valid non-spatial reference point. To this end, we develop a graph convolutional network (GCN) that uses graphs defined by cellular positions to predict gene expression. By encoding tissue samples as a graph, in which nodes represent cells and edges indicate spatial proximity between cells, we can leverage the full spatial layout and gene expression profile of the tissue. We find a marked performance gap between spatially aware and spatially ignorant models, highlighting the GCN's ability to model spatial effects in both real and semi-synthetic settings. These results underscore the importance of model structure in spatial inference because a spatially ignorant version of GCNs can make better predictions than spatially aware versions of previous methods.

Second, we study a clustering task for spatial transcriptomics data through a Bayesian framework. A central challenge in spatial transcriptomics is to identify distinct cell communities that not only reflect transcriptional heterogeneity but also preserve spatial coherence across tissue. These clusters often represent biological components such as cortical layers, tissue micro-environments, or pathological regions, whose spatial organization is critical for interpreting tissue structure and function. However, spatial transcriptomics data are collected at varying resolutions; as such, any spatial unit indexed by the data may contain multiple communities of varying memberships. Many exact Bayesian approaches model hard cluster assignments in their models, which limits their adaptability to datasets of vary-

ing resolutions. To address this limitation, we introduce a stochastic variational inference (SVI) method designed to learn posterior spot cluster distributions that are both spatially coherent and biologically interpretable. Our approach enhances clustering accuracy by incorporating spatial relationships through carefully designed prior distributions, allowing it to balance the trade-off between smoothness and expression differences. Furthermore, the method is scalable and effective across data resolutions. As spot data scales polynomially with finer resolution, SVI becomes a more favorable approach. It is more computationally efficient than previous methods that rely on posterior sampling techniques, such as Markov Chain Monte Carlo (MCMC), which can be prohibitively expensive to retrain. This method groups tissues into more contiguous regions compared to previous methods while preserving expression heterogeneity consistent with earlier studies, offering a competitive alternative to existing approaches.

Third, to expand the work of Bayesian clustering with SVI, we leverage normalizing flows as the approximate posterior distributions for variational inference. Normalizing flows transform simple base distributions (e.g., Gaussian) into more expressive ones by stacking L invertible transformations based on the change-of-variables formula. By using normalizing flows instead of standard choices like a mean-field or full-covariance Gaussian as the approximate posterior, we can model more flexible, multi-modal posteriors over soft cluster assignments in a way that simpler variational families cannot express. We demonstrate that the posteriors learned by these normalizing flows accurately recover cluster membership compositions, guided by prior distributions that encode spatial dependencies.

CHAPTER 1

Introduction

Spatial transcriptomics (ST) refers to a class of biotechnologies that integrate spatial context (such as the precise cellular coordinates within a tissue) with high-dimensional transcriptomics data (quantifying expression levels of many genes). This integration enables the modeling of cellular ecosystems directly within their original structural and functional contexts. The ability to model such phenomena motivates statistical formulations grounded in uncertainty quantification, maximum likelihood estimation, and posterior inference.

Prior to the emergence of spatial transcriptomics in 2015, much of the available gene expression data was collected either via bulk RNA-sequencing (bulk RNA-seq) or single-cell RNA-sequencing (scRNA-seq). Bulk RNA-seq measures gene expression over entire tissue samples, while scRNA-seq captures gene expression from each individual cell. Both methods are capable of high gene throughput; bulk RNA-seq is known to be effective at covering the entire transcriptome and scRNA-seq is capable of measuring thousands of genes at the cellular level. However, they also require dissociating cells from tissue, which prevents tracking where cells were originally located in the tissue sample prior to expression measurement. The datasets produced by these technologies were used to analyze paired tissues for applications such as differential expression of genes and biomarker discovery (Li and Wang, 2021). However, the scope of questions and answers that can be addressed by gene-level summaries for entire tissue samples obtained via bulk RNA-seq and scRNA-seq data is limited. These datasets help us perform analyses *between* and *across* tissues but *not within* a tissue sample. As a result, applications such as differential expression analysis of a single tissue or between healthy and diseased tissues could be supported by non-spatial transcriptomic technologies, but inferences that assume intra-tissue heterogeneity could not be supported. Phenomena such as spatial domains, cell-cell communication, and microenvironment-specific signaling, among many others, did not have the appropriate data available for statistical modeling with the spatial context of tissue incorporated. Despite this, there are spatially aware methods that predate modern spatial transcriptomics. Single Molecule Fluorescence in situ Hybridiza-

tion (smFISH) was able to measure transcripts at cellular resolutions without dissociating from the tissue but suffered from incredibly low gene throughput (Femino et al., 1998). Despite subsequent improvements, this approach typically processes only tens of genes, with the most best version reaching a few hundred (Buxbaum et al., 2014; Pichon et al., 2018). With such low gene throughput, relevant genes may be excluded that are important for downstream inference tasks. On the other hand, Laser Capture Microdissection (LCM) dissects regions to be processed for bulk RNA sequencing, which provides access to more genes in a localized region (Emmert-Buck et al., 1996). However, LCM still requires physically isolating tissue regions, which is a form of tissue dissociation. Prior to the adoption of spatial transcriptomics, inferential experiments about gene regulation, tissue architecture, and cell-cell communication had to balance a trade-off between spatial precision and gene throughput. Modern ST technologies represent an era where this compromise is substantially reduced, allowing us to answer a wider range of biological questions with improved fidelity and confidence.

Preserving positional information of cells or tissue spots during statistical inference represents a pivotal breakthrough in computational biology. At these higher resolutions, researchers have sufficient data to determine whether the expression of ligands and receptors reflects structured patterns of cell-cell communication, whether immune or stromal cells occupy distinct niches, or how pathological regions disrupt normal signaling pathways. This represents a qualitative shift from previous transcriptomic technologies, which often relied on cell-type identification or broad assumptions about expression distributions across spatial regions of a sample. With spatial data, we can model biological processes as inherently situated in space, a modeling paradigm more in line with biological reality. At present, there is an abundance of ST datasets. The chronology of notable developments is depicted in Figure 1.1. After the advent of the original spatial transcriptomics method introduced by Ståhl et al. (2016), several ST advancements have emerged, including but not limited to: seqFISH+ (Eng et al., 2019), MERFISH (Chen et al., 2015), Slide-seq (Rodriques et al., 2019) and its improved version Slide-seqV2 (Stickels et al., 2020a), Visium (10x Genomics) (10x Genomics, nda), Stereo-seq (Chen et al., 2022), CosMx SMI (He et al., 2022), Xenium (10x Genomics) (Janesick et al., 2023b), MERSCOPE (Vizgen), and Molecular Cartography (Resolve). These technologies have produced a diverse array of spatially aware transcriptomic datasets that vary in spatial resolution and molecular coverage. Practitioners sought ways to model the spatial information that generalize across datasets produced by these technologies. This is evident in the rise of software ecosystems that aim to support preprocessing, graph construction, clustering, regression, visualization, and downstream analysis of ST data (Hao et al., 2023; Palla et al., 2022; Wolf et al., 2018). Despite potential differences,

these advancements share a unifying goal of preserving and utilizing spatial information to study biological patterns that might go undetected in spatially ignorant settings.

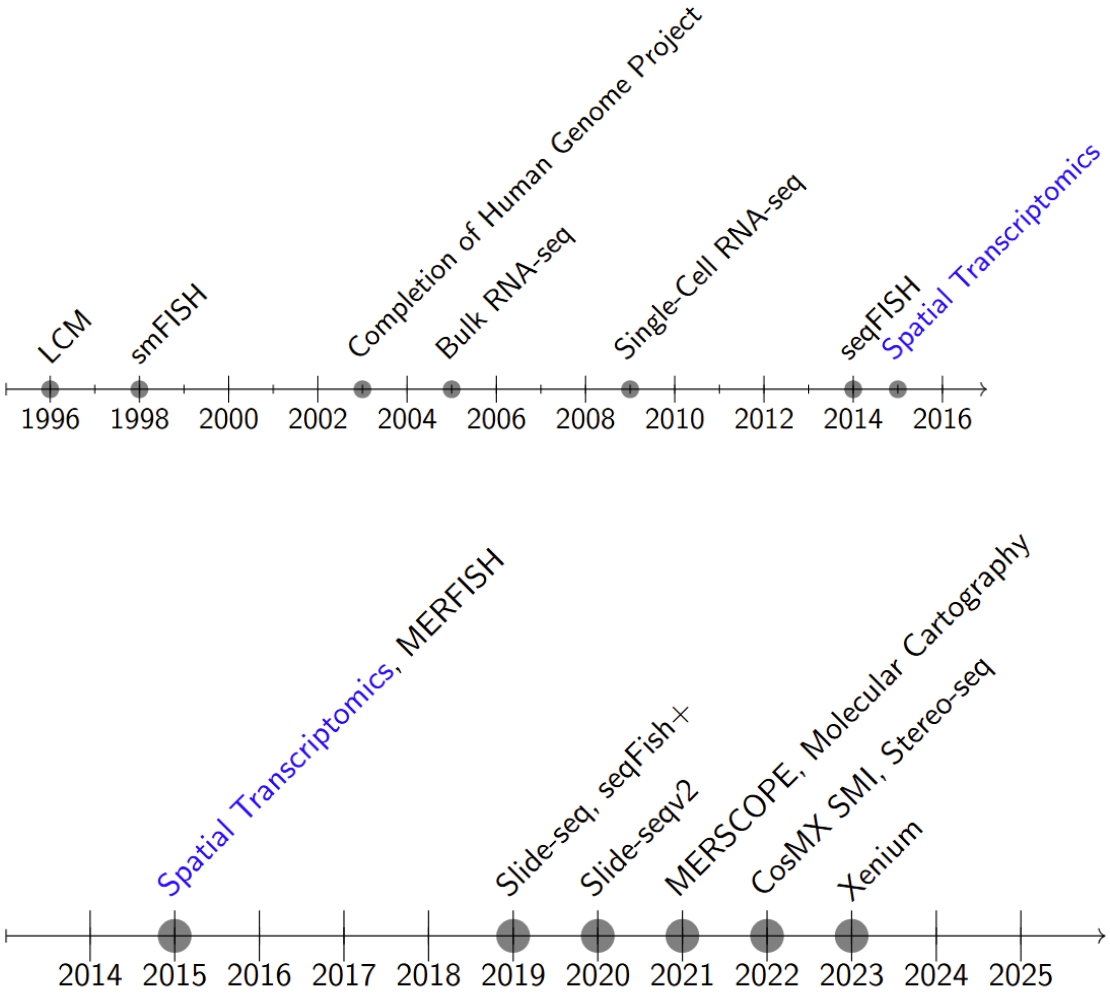


Figure 1.1: Timeline of major milestones in genomic and transcriptomic technologies. Top: pre-ST era. Bottom: post-ST advancements.

Modern spatial transcriptomics catalyzed the development of disparate models, including graph neural networks, decision trees, mixture-of-experts models, hierarchical Bayesian frameworks, variational autoencoders, Gaussian processes, and many others. These models leverage spatially resolved gene expression data to capture intricate biological structures. Each approach offers unique mechanisms for representing spatial dependencies, cell-cell interactions, and localized gene expression patterns. The combination of thousands of gene expression measurements per cell with low-dimensional spatial coordinates—usually in two or three dimensions—introduces modeling challenges: How can spatial and molecular dimen-

sions be optimally combined to capture both local and global patterns in cellular ecosystems? Which models most effectively harness spatial relationships to enhance interpretability and predictive power without sacrificing statistical rigor? How do we ensure the thousands of gene features do not dilute spatial signals? How do we guarantee that our model construction answers a biological question? As ST technologies scale and yield datasets that vary in resolution, innovative approaches are required to bridge spatial and molecular dimensions. Developing methods that take into consideration spatial context and resolution demands careful statistical design in order to draw biologically valid conclusions that generalize across tissues and technologies. This fusion of biology, statistics, and machine learning also requires careful consideration of what constitutes meaningful inference in a clinical setting; modeling improvements should ideally translate to improved biological insight. For instance, while lower loss or higher accuracy of a model serves as a useful baseline for model evaluation, marginal improvements should lead to more reliable conclusions—ones that provide biological or clinical insights that were previously uncertain, inaccessible, or unsupported. Model comparison assesses how altering the information available to the model changes its predictive accuracy. This type of assessment may not always be intrinsically interesting, but it is valuable for answering questions of scientific interest. The lens of model comparison, like all lenses, has both clarifying power and inherent distortion. When used thoughtfully, model comparison can highlight functional dependencies and suggest mechanisms worth further exploration. However, it can be misleading without rigorous attention to issues such as model flexibility, estimation quality, error quantification, noise, and biological context. Our work suggests that machine learning architectures such as GCNs and scalable variational inference frameworks can be helpful in addressing some of these subtle issues.

In this dissertation, we introduce three novel statistical inference techniques for leveraging emerging spatial transcriptomics technology.

In Chapter 2, we consider the regression task of predicting gene expressions using sender (ligand) and receiver (receptor) genes as inputs to the model. This focuses on studying cell-cell communication (CCC) effects, typically with the aim of identifying which genes are signaling or spatially dependent. This problem has been previously studied in non-spatial omics data and early spatial transcriptomics with the work of Li et al. (2020) catalyzing the inclusion of the spatial context available in ST datasets. The conclusion of spatial dependence is reached by comparing pairs of models with identical architectures, differing only in their access to neighboring information as predictors. However, many of the existing modeling families used for this inference procedure underfit the true relationship between local signaling environments and transcriptional responses. For instance, while CCC primarily involves paracrine signaling—where a cell sends a signal to a neighboring target—a target

cell does not necessarily need a separate cell to receive a signal; it can receive signals from itself in a process known as autocrine signaling. Too often, signals delivered via autocrine and paracrine signaling are treated as indistinguishable features, which can confound the interpretability of inferred CCC effects.

To address this, we propose a pipeline for estimating response gene expressions with a graph convolutional network (GCN) as the regression model. This GCN incorporates Gaussian mixture-inspired kernels, enabling the model to learn weighted signals from neighboring cells more effectively. Our approach, termed DeepST, uses spatial information to better predict target gene expressions in large-scale ST datasets without requiring manual filtering or feature selection. This is accomplished by creating tissue graphs based on the positional information; cells represent the nodes and edges represent possible signals sent between cells. Furthermore, identifying spatial dependence within target genes can be formulated as a model comparison problem. By contrasting paired models—one incorporating spatial information and one omitting it—decision rules can be established to determine whether a target exhibits spatial dependence. DeepST improves the reliability of spatial dependence detection, outperforming contemporary methods and maintaining accuracy even when the constructed tissue graphs do not perfectly represent the underlying cell-cell interaction topology.

Our model comparison framework assumes that each candidate model, given its specific inputs, produces the predictions that minimize average error. This assumption is only viable if the models are all sufficiently flexible and accurately estimated. To see how this could create difficulties, we first consider two abstract examples. In both cases, assume that the spatially ignorant model class is less flexible and the spatially informed model class is more flexible. First, if the target gene depends solely on local (non-spatial) node attributes, a spatially informed model might still outperform a simpler spatially ignorant model. This could occur because the greater flexibility of the spatially informed model allows it to capture complex relationships with local attributes, even though spatial information isn't relevant. Second, if the target gene depends on neighborhood gene expression, a spatially ignorant model might still outperform a spatially informed model. This can happen if estimating the spatially informed model's many parameters requires more data than available, making it difficult to estimate accurately. Consequently, the simpler spatially ignorant model may perform better on held-out data. In practice, choices about flexibility, estimation, and error can be challenging to reason about. Even minor subtleties can have outsized impacts. Seemingly straightforward model comparisons can become misleading when subtle interactions between flexibility, estimation accuracy, and evaluation criteria are overlooked. Careful attention to these interactions is essential to reliably identify spatially varying genes. The GCN underly-

ing DeepST is sufficiently flexible, and the pipeline is explicitly designed to treat autocrine and paracrine signals separately, which helps avoid these pitfalls. DeepST exemplifies how advances in machine learning not only improve predictive accuracy, but also establish how thoughtful model design can more appropriately leverage spatial context, avoiding errors and increasing reliability in detecting spatial dependence.

While Chapter 2 focuses on using ST data in a regression setting, in Chapter 3, we address the task of spatial clustering in ST data. Many existing methods for clustering transcriptomics data try to identify groupings with similar gene expression profiles. However, practical applications often require clusters to also reflect an interpretable neighborhood structure. These objectives are often at odds, requiring models that can balance this trade-off. Several approaches have been proposed to address this challenge, ranging from deep learning (Dong and Zhang, 2022; Hu et al., 2021), graph-based clustering (Blondel et al., 2008; Miller et al., 2021; Traag et al., 2019), novel spatial data construction (Pham et al., 2023), and fully Bayesian spatial models (Dries et al., 2021; Yang et al., 2021; Zhao et al., 2021) among others.

Clustering with spatial contiguity in mind highlights the modeling challenge posed by the high dimensionality of gene expression relative to spatial coordinates. Including positional information as direct inputs to clustering algorithms could ensure that clusters follow spatial patterns, but this signal can be dwarfed by transcriptomic features. One could amplify spatial features to manually weight their influence, but this is a form of inductive bias that may be too rigid or easy to exploit. Instead, a common approach is to define a Markov random field over tissue regions, encoding spatial dependence consistent with prior assumptions about tissue organization. Now, positional information defines a global assumption of spatial dependence instead of being treated as input features. The graph characterizing the Markov random field can be constructed similarly to how it was done in Chapter 2; changing the graph in turn changes the type of spatial dependence we expect to be present in a tissue sample.

Bayesian models are especially appealing for this problem because they provide uncertainty in the clustering assignments and enforce spatial smoothness with a prior distribution. This uncertainty is especially important due to the variable resolution of ST data. In high resolution datasets, each spatial region might only belong to a single cluster, making modeling hard assignments a reasonable design choice. Still, Bayesian models offer uncertainty estimates about cluster labels that can indicate ambiguous boundaries or low posterior confidence due to overlapping expression profiles. In lower resolutions, where each spatial region often contains expression from multiple cells or even multiple cell communities, it is more appropriate to use soft assignments to characterize these regions. Yet, most existing Bayesian

models do not adapt to this reality, making them less capable of representing mixture memberships.

Furthermore, Bayesian methods are less scalable than alternative approaches, making it difficult to apply them to modern ST datasets as their resolutions increase. This challenge is compounded when tissue graphs are dense, with high neighborhood connectivity. Even in sparse cases, as ST technologies continue to increase in resolution, a small proportion of connections between regions can create expensive message passing and inference procedures.

We propose using variational inference with empirical priors to address the scalability limitation. Our method, BayXenSmooth, learns an approximate posterior cluster membership distribution given some initial estimate based only on gene expression similarity. We construct prior distributions that encourage spatial smoothness among neighboring regions, allowing for inference where our prior belief is that nearby cells are more likely to belong to the same cluster. Shifting from an exact to approximate Bayesian inference regime allows us to also explore a larger set of model families that may accurately reflect the structure of ST data. Exact Bayesian inference is primarily useful when we have high confidence in the model's structural assumptions, the dimensionality is modest, and the cost of posterior sampling is justifiable. However, these conditions are not met for spatial transcriptomics, in which the spatial dependencies are not fully understood and should not be constrained to a fixed parametric form just to support Bayesian inference.

In Chapter 4, we extend the work from Chapter 3 with normalizing flows to capture more expressive approximate posteriors over spatial cluster assignments. Potential poor mixing, model mis-specification, and high computational cost from established Bayesian methods suggest that a more complicated distribution may be necessary for posterior estimation. Chapter 3 alleviates this concern by moving to an approximate Bayesian inference procedure with a tractable variational family. However, using traditional parametric distributions as approximate posteriors may underfit the true posterior and be a significant restriction on the quality of inference (Rezende and Mohamed, 2016). This would be especially true if the relationship between the expression data and cluster assignment—where cluster assignment can represent spatial domains, tissue regions, anatomical layers, pathological boundaries, etc.—exhibits nonlinear and multi-modal associations.

Normalizing flows enable the modeling of intricate distributions by applying a series of complex, invertible transformations to a simple base distribution. This invertibility allows for exact likelihood computation—a feature not generally available in deep learning approaches like diffusion models and variational autoencoders which rely on stochastic approximations. This approach builds on recently proposed Bayesian models that encourage neighboring entities to be similarly clustered in the prior distribution. Intuitively, this means the approximate

posterior is incentivized to cluster adjacent regions differently only when the likelihood of such an occurrence is exceedingly low. The result is a smoother spatial clustering, with each location having a distribution over cluster memberships. Despite normalizing flows being inherently uninterpretable, we embed them in a Bayesian framework that offers interpretability. We assume a prior where biological spatial organization underlies the data, and allow the flexible approximate posterior to express disagreement if it finds compelling evidence to push against the prior. If it does not, the learned posterior simply reinforces the spatial structure, providing support for our prior belief and increasing trust in downstream analyses.

The spatially aware statistical methods we have developed advance predictive inference in spatial transcriptomics in ways that directly contribute to clinical benefit. This spatial awareness is crucial to valid biological inference because—at bare minimum—spatial location and gene expression co-vary in ways that give rise to organized tissue regions and distinct groups of interacting cells. A cell's physical neighborhood can shape its functional identity. Cells can coordinate their behaviors based on the signals they receive from neighboring cells, yielding spatially coherent patterns that organize cellular neighborhoods, boundaries, and functional zones within the tissue. Through demonstrated improvements in regression and clustering performance, we enable the identification of spatially variable genes and spatial domains. These contributions support practical applications including targeted therapy, disease progression analysis, and tissue-specific regulatory discovery. Furthermore, they offer computational runtime improvements that do not compromise statistical rigor. Together, the models developed in this dissertation contribute to a growing class of models incorporating spatial structure into inference, advancing how we extract biologically meaningful patterns from spatially structured data.

CHAPTER 2

Modeling Gene Expression with Graph Convolutional Networks

2.1 Introduction

Cell-cell communication (CCC) is an essential function of multicellular organisms. Much of this communication takes place through ligand-receptor binding. To send signals, a cell constructs molecules known as a ligands and sends them outside the cell membrane. Ligands then bind to receptor molecules on the surface of nearby target cells. This binding event affects gene expression in the target cell. Changes in gene expression, in turn, can affect almost all cell functions. Thus, understanding cell-cell communication is an active research avenue with widespread applications.

Studying CCC is challenging. Ideally, one could track ligands from the moment of their production to the moment of their receptor binding. Some methods have been developed to approach this ideal, but in all such methods the practitioner can only measure a very small number (typically one) of ligand-receptor pairs at a time (Stockmann et al., 2017). Single-cell transcriptomics experiments offer another approach: cells from a tissue are dissociated from each other, and then the expression of genes associated with ligands and receptors can be measured for each cell separately. This allows a practitioner to measure many receptors and ligands in a single tissue. However, it is difficult to infer communication effects from looking at these gene expressions alone. These experiments do not record the positions of the cells in the tissue. Nonetheless, efforts such as PIC-seq and probabilistic models on integrated data have attempted to study communication events between cells without positional information (Giladi et al., 2020; Wilczynski et al., 2012).

Spatial transcriptomics experiments offer a promising new path to studying CCC, enabling practitioners to measure gene expression for individual cells as well as the positions of these cells within a tissue sample. As ligands typically bind to receptors on nearby cells, this positional information should facilitate our study of CCC (Wolpert, 1969). While we

understand the general roles that ligand-receptor interactions and signaling mechanisms play in CCC, quantifying the effect of communication events remains an open problem (Foster et al., 2021). Modeling expression changes affected by CCCs has several major applications. Determining the spatial dependence among genes is useful for identifying genes that are candidate therapeutic targets—genes that drugs can focus on to inhibit or activate specific pathways, thereby altering disease progression (Yang et al., 2024). A statistical framework for ranking such genes can prioritize which genes to develop drugs for, reducing the costs associated with experimental validation across a large set of candidate genes. Even when a spatially dependent gene is not a therapeutic target, modeling genes that have a dysregulated expression in healthy vs. disease tissue can help guide experiments. Another application is the study of tumor heterogeneity, which can be explored by analyzing the spatial context of oncogenes, that is, genes that drive cancer. If a known oncogene expression has strong spatial patterns, that could indicate a tumor micro-environment effect: an effect where cancer cells interact with neighboring stromal, immune, and vascular cells to influence tumor progression.

To better identify the relationship between cell-cell communication and gene expression, we propose a graph convolutional network (GCN) that can be utilized to determine if a gene's expression is better modeled when neighboring information is considered. By representing the tissue sample as a graph, we avoid the restricted spatial information that can be represented in a finite dimension while adding a more spatially aware model that utilizes deep neural networks as opposed to classical methods. Our method also attempts to make predictions as an independent network; it does not perform any clustering or dimensionality reduction for downstream tasks. With this method, we identify genes that exhibit spatial dependence by modeling how cell-cell communication affects gene expression in neighboring cells.

When spatial patterns occur in tissue samples, the high volume of unique genes make it difficult to robustly identify the subset of them that vary spatially; the challenge is compounded when the spatial effect is predicated on an interaction between genes. Identifying spatially dependent genes can be an appropriate substitute in settings where histology images are unobtainable. Additionally, inference on expression spatial dependence has a relevant application in identifying cell subtypes that are correlated with other indications of disease (e.g., tumorous tissue) (Luo et al., 2021).

With spatial transcriptomic data, understanding how neighboring cells interact can be cast in terms of model selection. We first model gene expressions in each cell using the attributes of the cell. We then model the same responses using the attributes of the cell and its neighbors. The comparison between these two models is a generalization of leave-one-covariate-out (LOCO) (Lei et al., 2018). This framework allows us to assess the effect of

neighboring signals ceteris paribus.

Genes may then be ranked according to difference in predictive performance: genes for which the predictions from the second model are more accurate than the first may carry important spatial correlates. Highly ranked genes may warrant further investigation through follow-up experiments.

2.2 Related Work

The ability to couple gene expressions with spatial positions created a boom in interest to derive insights about CCCs. This section serves as a library of prior work in spatial transcriptomics inference.

Existing models often rely on predefined summary statistics of neighboring gene expression, constraining their ability to fully leverage spatial transcriptomics data. This limitation can lead to incorrect conclusions about spatial dependencies in gene expression that are potentially spurious. In most cases, the summary statistics represent the expression of neighboring cells by aggregating or encoding it into predefined feature sets. Such encodings pigeonhole valuable neighborhood information into a finite-dimensional summary, and may fail to capture the full complexity of spatial organization. Figure 2.1 illustrates this limitation, showing how identical summary statistics can arise from drastically different spatial structures. Graph-based methods provide greater flexibility than fixed-dimensional embeddings, enabling more precise reconstruction of response gene expression. While our approach shares similarities with previously proposed graph-based methods for analyzing spatial transcriptomics data, these models are typically developed for different tasks — such as clustering, niche modeling, or spatial domain detection — and they lack explicit model comparisons between spatially aware and spatially ignorant settings, making statistical validation of spatial dependencies difficult. We focus on the gene expression prediction problem and emphasize that a pair of well-specified prediction models can assess whether response expressions are conditionally independent of neighboring signals. The first model is a spatially ignorant baseline that considers every cell to be isolated while the other model is a spatially aware model that considers neighboring cells for each target cell.

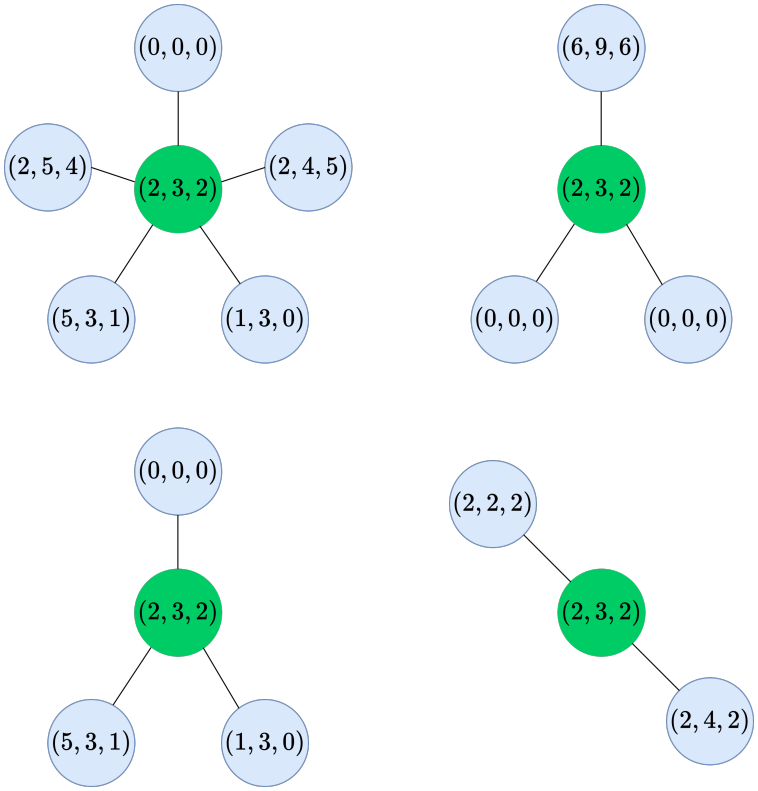


Figure 2.1: Illustration of the limitations of fixed-dimensional vector summaries for neighboring gene expression. Each circle represents a cell, with target cells (green) receiving signals from neighboring cells (blue). A fixed-dimensional encoding of the neighborhood of each target cell is produced by averaging the gene expression vectors from neighboring cells, yielding the same encoding, (2,3,2), for every target cell. However, the four target cells differ significantly in the number of neighboring cells, the distribution of their gene expression values, and the spatial arrangement of cells.

2.2.1 Mixture of Experts

One approach taken to make inferences from CCCs involves inferring cell subtypes for response expression prediction using spatial information as an input. Mixture of Experts for Spatial Signaling genes Identification (MESSI) is one method that has achieved promising results by leveraging a mixture of experts model (Li et al., 2020). MESSI compares spatially aware predictions with spatially ignorant predictions to identify spatially dependent genes. This framework identifies signaling genes responsible for response expressions by evaluating learned coefficients from each expert. Each expert is intended to model a subtype of a cell, an effect they enforce by filtering inputs to a specific cell type in advance. Identifying cell subtypes can prove useful for predicting response gene expression. However, unlike our proposed approach, MESSI represents neighborhood expression by aggregating it into a

summary statistics that may fail to capture the full complexity of spatial organization. By comparing the predictive efficacy of various mixture model architectures, MESSI infers the relevant number of cell subtypes within each cell type.

However, MESSI has several limitations. Their neighborhood construction is accomplished with Delaunay triangulation, which is restricting and may fail to capture the full range of CCC interactions. Additionally, while MESSI did compare against gradient boosting, they compared predictive quality, measured by mean absolute deviance, against an XGBoost implementation trained with a mean squared loss. We found that a LightGBM implementation trained with a mean absolute deviance loss outperforms MESSI under matched conditions.

LightGBM is more computationally feasible for large datasets as well as supporting categorical features (like cell type) without the need for one-hot encoding. Furthermore, the decision tree learned by LightGBM is learned leaf-wise—as opposed to the level-wise approach learned by XGBoost. This means that LightGBM generally leads to higher accuracy by making the split on the leaf that most reduces loss, regardless of its depth, resulting in faster convergence and likely a better performance on large datasets. As higher resolutions increase the number of cells and lead to more complex tree structures learned by boosting algorithms, LightGBM's efficiency and scalability make it a natural choice for large-scale ST datasets, as our results will demonstrate. MESSI performs optimally when the data is stratified by both demographic factors and cell type. The availability of cell type is not always present in ST data and therefore, can lead to poorer performance in practice. Lastly, although MESSI identifies signaling genes via expert-specific coefficients, these interpretations may be misleading if the model lacks the flexibility to capture complex relationships in spatially naive contexts. These problems motivate the need for a more expressive model that leads to correct conclusions about genes' spatial dependencies when compared against a spatially ignorant baseline.

2.2.2 Gaussian Process Regression

While any regression model can be used to make gene expression predictions, they often lack spatial context. However, a subset of spatially aware regression models has emerged to bridge this gap. nnSVG is a prime example of a spatially informed regression model that identifies genes that vary across tissue samples (Weber et al., 2023). At its core, nnSVG uses a nearest-neighbor Gaussian process to encode spatial correlation through a covariance structure that decays with distance.

To achieve this, the model decomposes the total variance into spatial and non-spatial

components

$$\Sigma(\boldsymbol{\theta}, \tau^2) = C(\boldsymbol{\theta}) + \tau^2 \boldsymbol{I}$$

with C serving as a spatial proximity kernel

$$C_{ij}(\boldsymbol{\theta}) = \sigma^2 \exp\left(\frac{-\|\mathbf{s_i} - \mathbf{s_j}\|}{l}\right).$$

The spatial variance is a function of the euclidean distance between two spatial locations s_i and s_j. Maximum likelihood estimates are obtained for parameters (σ^2, l, τ^2) which helps quantify how much variation is attributable to spatial structure rather than residual noise.

The model assumes that gene expression follows the following multivariate normal distribution:

$$\mathbf{y} \sim N\left(\mathbf{X}\boldsymbol{\beta}, \Sigma\left(\boldsymbol{\theta}, \tau^2\right)\right).$$

This approach extends traditional regression by replacing the i.i.d. noise assumption with a spatially correlated error structure, thereby introducing spatially aware predictions. Furthermore, nnSVG can compare the effect of spatial influence by conducting a likelihood ratio test between a null Gaussian process where $\sigma^2 = 0$ against their proposed method which learns $\sigma^2 > 0$. Furthermore, an effect size of the spatial signal can be represented by $\frac{\sigma^2}{\sigma^2+\tau^2}$.

However, these predictions assume Gaussianity and the model flexibility is constrained by a pre-determined spatial kernel function. Ideally, the predictive model would be a universal approximator, capable of modeling signal propagation that may be nonlinear, sparse, or cell-specific.

2.2.3 Graph Neural Networks

Graph convolutional networks have been used in a variety of ways for gaining insights about CCC.

Graph Convolutional Neural networks for Genes (GCNG) casts detecting CCC gene relationships as a GCN link prediction problem (Yuan and Bar-Joseph, 2020). In contrast, our approach defines edges and edge attributes based on cell proximity and uses them for gene expression prediction.

SpaGCN uses a GCN to learn an embedding that is used for spatial domain clustering (Hu et al., 2021). GraphST extends the effectiveness of graph neural networks for clustering tasks by leveraging self-supervised contrastive learning to learn spatially informed representations, while also proving useful for cell-type deconvolution. These clustering tasks are different from

the tasks considered in this article. Our task focuses more on using GCNs for predicting response expressions conditioned on neighboring ligand and receptor activity, which we frame as a straightforward regression problem.

Node-centric expression models (NCEM) is more aligned with our work in the regression problem we aim to solve and uses graph neural networks (GNNs) as spatial encoders within their modeling pipeline (Fischer et al., 2023). However, NCEM presents introduces a family of related models rather than a singular one; in most of these approaches, using the cell-type annotations of a target cell and its neighbors is central to most of its modeling strategies. One variant of NCEM deemed Non-Linear Ligand-Receptor NCEM (NL-NCEM-LR) is the closest to our method, using only ligand and receptor expressions for predicting target expressions, and the authors demonstrated this version achieved the highest test performance (in terms of R^2) on several real datasets. However, rather than allowing for the encoder and decoders themselves be GNNs or GCNs, they typically are typically trained to learn a fixed-dimensional embedding that the the NCEM encoder and decoder MLPs take as input. We will demonstrate that using the GCN as the regression model, outperforms non-spatial baselines and has a more stable performance across a range of neighborhood radii in tissue graphs.

HoloNet utilizes graph-attention network architectures for niche modeling and ligand-receptor analysis (Li et al., 2023a). While this method can be posed as a regression problem, it does not directly assess the added benefit of spatial information, limiting the strength of their conclusions about spatial dependence.

The explicit comparison of spatially ignorant and spatially informed GCN regression models differentiates our work from existing GCN approaches for understanding CCC. By explicitly comparing these paired models for each gene of interest, we directly quantify the predictive value of spatial context, enabling principled identification of spatially regulated genes.

2.3 Graph Representations and Learning for Cell-Cell Communication

Here, we present the motive and relevant background for DeepST, a GCN that takes as input a tissue sample mapped to a graph. The goal is to leverage cell neighborhood structure to improve predictions of gene expression patterns within individual cells.

Furthermore, we examine cell-cell communication through the perspective of model selection. Specifically, we consider gene expression at the single-cell level and distinguish two

types of expression: local expression, defined as the expression level of each gene within a specific cell, and neighborhood expression, defined as the gene expression levels observed in neighboring cells. To assess spatial dependence, we compare spatially aware and spatially ignorant models based on their ability to predict gene expression. A gene is considered spatially dependent if the spatially aware model achieves a significant decrease in prediction error than its spatially ignorant counterpart.

However, in a spatial graph where all cells within a fixed radius are considered neighbors, the number of edges per cell scales quadratically with the radius. This can be expensive in memory if we plan to store the expressions of each neighbor as separate covariates. A possible work around is to provide finite dimensional summaries of neighboring gene expressions. Assuming some distribution over the gene expressions, these summaries could be sufficient statistics which provide the same inference at a fraction of the dimensionality. MESSI, for example, uses a vector of sums of ligand expressions across all neighboring genes. As previously discussed, fixed-dimensional summaries may obscure spatial structure and require constructing summaries without knowing which ones are meaningful a priori. This practice is common and not inherently negative, but being able to take advantage of edge weights or other relevant aspects of graphs can prove useful in properly quantifying cell-cell communications (Kim and Cho, 2019; Wei et al., 2022). The relationship between a neighboring signal and a target expression may be a nonlinear function and each ligand can affect a response gene uniquely. Additionally, the summary limits any insights that happen from multi-hop communications.

To address these limitations, we choose a graph-based modeling strategy, converting tissue samples into graphs that are used as inputs for a GCN. In the following subsections, we introduce relevant background on the two core components of our method.

2.3.1 Graphs

A graph $G = (V, E)$ is a data structure determined by a set of nodes V and the edges that connect these nodes E. Each edge in the graph can have a weight that quantifies the strength of the relationship between nodes. A graph that does not weight its edges—in effect treating all edges equally—is defined as unweighted. A graph can be uniquely determined by its adjacency matrix A such that A_{ij} represents the weight of the edge between node i and node j should the edge exist. The adjacency matrix is a matrix that stores the edges that exist as 1s and 0s otherwise. Edges can either flow in one direction (directed), or in both directions (undirected). An undirected graph has the requirement that ($A = A^T$), whereas a directed graph can have complete freedom in its off-diagonal elements. For our purposes, we assume

the cell graph is undirected. Each node ($v \in V$) in the graph can have node attributes (v_1, v_2, ..., v_p) that describe certain properties of the nodes.

With this, we complete the analogy between a graph and cells dispersed about a tissue slice; the nodes are represented by cells, cell-cell communication events are represented by the edges, and cell expressions are represented by node attributes.

2.3.2 Graph Convolutional Networks

GCNs provide similar benefits to learning on graph inputs as convolutional neural networks provide for fixed-dimensional data. The inputs provided to a GCN are the node attributes of a targeted node provided by a dataset; the outputs are a set of predicted attributes for the same targeted node. The nodes on the graph have a specific relationship exhibited by their edges; a model that wants to make the most of the inference available from this information should use the edges.

The generalized update step in a GCN is $H^{l+1} = f(H^l, A)$, where l represents the layer depth ($l = 0$ being the input dimension), H^l represents the node attributes of the network at layer l, A is the adjacency matrix, and f a nonlinear mapping. The fundamental idea of the update step is that forward passes are non-linear functions of the previous weights that create new hidden values as an aggregation from neighboring nodes. The specific choice for f depends on the GCN. Some commonly used architectures include the original GCN, Graph Attention Network, Graph Isomorphism Network, and GraphSAGE (Hamilton et al., 2017; Kipf and Welling, 2016; Veličković et al., 2017; Xu et al., 2018a).

Unlike classical machine learning methods, forward passes through an untrained GCN can still provide insight into node features due to the adjacency matrix being known a priori (Kipf and Welling, 2016). As we will show later on, GCNs devoid of any spatial information can outperform models from previous works tailored to include it.

Highly parameterized networks are universal approximators and can learn complicated functions (Cybenko, 1989). Leveraging these graphs and applying flexible convolutions to them can resolve the limitations of fixed-dimensional representations.

2.4 Methodology

In this section, we present the complete modeling and training procedure of DeepST. We also introduce a baseline method—technically a special case of DeepST where no cells are connected by edges. In this special case, DeepST is equivalent to a feed-forward neural network that takes as input the covariate gene expression levels.

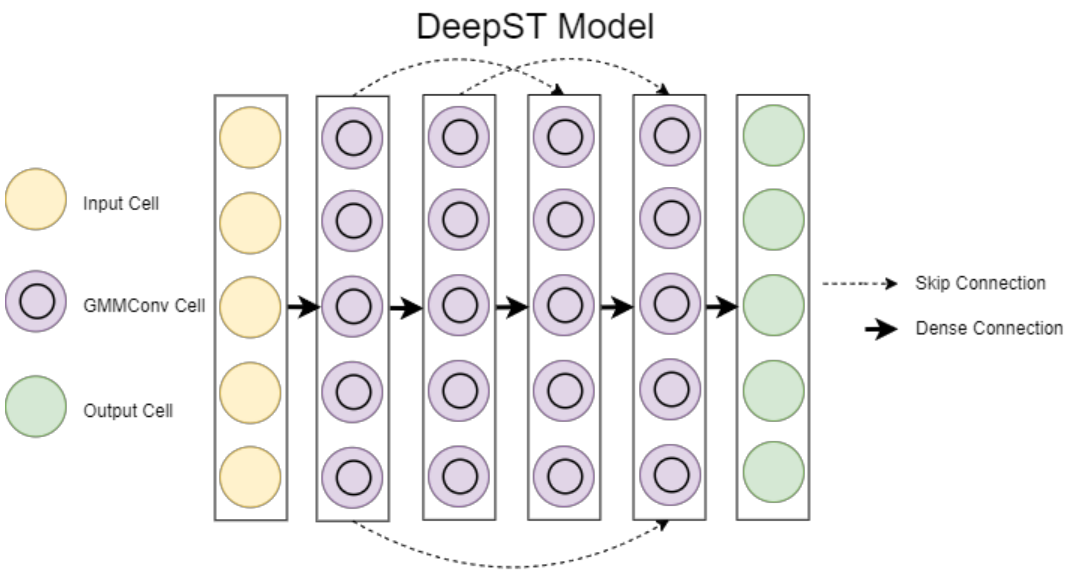


Figure 2.2: DeepST model architecture. Skip connections are only included in scenarios where hidden layers have the same dimensions; results shown in this section are from a model with all hidden layers having the same dimension and skip connections included whenever possible.

2.4.1 Introduction and Notation

We start by outlining the problem of predicting response gene expression from ligand and receptor activity within a local neighborhood. The relevant notation is provided in Table 2.1.

Variable	Description
X	dataset of all G gene expressions for all cells in a tissue sample
Y	Response gene expressions for all cells in a tissue sample
X_c	All gene expressions for cell c
$X_{c,L}$	Expressions of all L ligands for cell c
$X_{C,L}$	Expressions of all L ligands for cells in set C
$X_{c,R}$	Expressions of all R receptors for cell c
$X_{C,R}$	Expressions of all R receptors for cells in set C
Y_c	Expressions of all response genes for cell c
$Y_{c,g}$	Expression of response gene g for cell c
$Y_{C,g}$	Expression of response gene g expressions for cells in set C
$\mathcal{N}(c)$	The set of neighboring cells for cell c
$X_{c,g}$	Expression of gene g for cell c

Table 2.1: Description of variables for DeepST modeling.

DeepST attempts to learn the expressions of response genes based on the ligand and receptor expressions $E[Y_c] = f_r(X_{c,L}, X_{c,R}, X_{\mathcal{N}(c),L}, X_{\mathcal{N}(c),R})$ where f_r represents the GCN.[1]

This method can be understood in terms of two parts: the graph that it is based on, and the convolutional operation applied to that graph. The next sections explain how tissue samples are translated into graphs, how the DeepST GCN performs a convolution over neighboring cells, and provides a comprehensive overview of how each stage contributes to the broader modeling framework.

[1]In the event that $\mathcal{N}(c)$ is defined to include c (i.e. the graphs includes self edges), then this equation can be written without the first 2 inputs: $E[Y_c] = f_r(X_{\mathcal{N}(c),L}, X_{\mathcal{N}(c),R})$.

2.4.2 Graph Creation

To use GCNs, we can construct graphs from spatial transcriptomics data, treating the cells as nodes and defining edges based on spatial proximity. Specifically, we include an edge in the graph if the distance between any two cells (i, j) is beneath a threshold (r):

$$A_{ij} = \mathbb{I}\left\{d(i,j) < r\right\}. \tag{2.1}$$

In eq. (2.1), we call r the radius of consideration. This value can be customized if using a graph with $r^* > r'$ better predicts response genes. For applications, $d(\cdot,\cdot)$ represent the Euclidean distance between two cells in micrometers. While Euclidean distance seems like an intuitive rule to decide edge placements, previous work has explored alternative edge construction methods such as Delaunay Triangulation and K-Nearest Neighbors (Delaunay, 1934; Fix and Hodges, 1951). This can lead to edges that may not be representative of communication events that can happen naturally in an organism (see Figure 2.3). Communication events such as endocrine signaling may not be represented by edges created in high density regions and considered exclusively for more isolated cells. Signals get sent through the bloodstream to the target cells that may exceed distances between neighbors assumed by alternative neighborhood structures. Similarly, certain edges could connect nodes that are far apart and would be unrealistic in practice to facilitate cell-cell communication.

Node attributes encode relevant information about each node in the graph, allowing models like graph neural networks to learn relationships between nodes beyond their connectivity. These features help distinguish nodes with similar connections but different properties, improving the model's ability to capture meaningful patterns. In our case, we populate each node with the following node attributes:

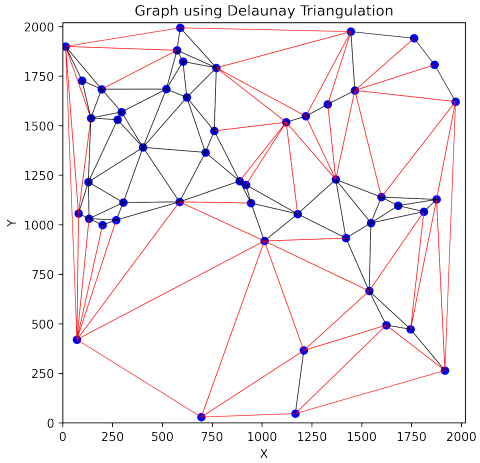
CCC features: ligand and receptor expression levels.

Response values: response gene expression levels

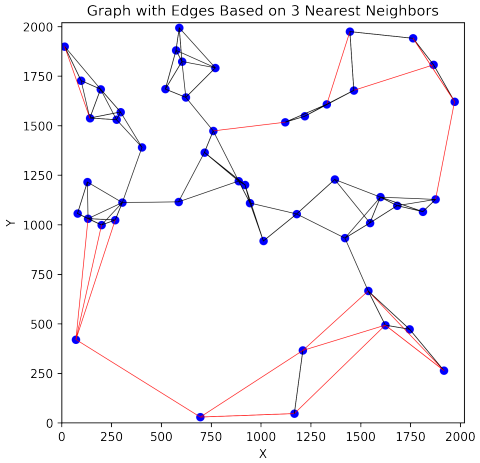
Metadata (Optional): cell type.

As a result, each graph represents a single tissue slice taken from an animal at a specific bregma. For model training and evaluation, we craft training, validation, and test sets that consist of disjoint sets of animal tissue slices. This is to ensure that the model generalizes across biologically distinct animals.

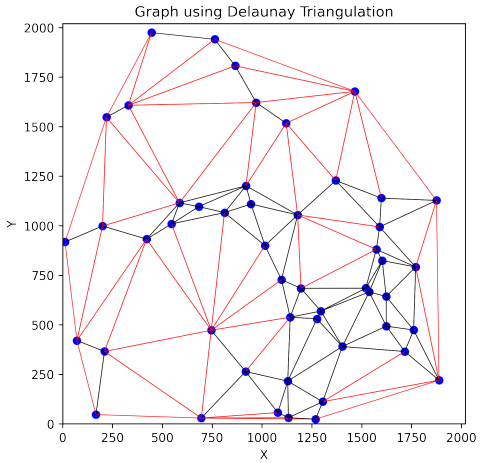
Thus, the graph structure encodes spatial transcriptomics data by treating cells as nodes and available CCC channels as edges.



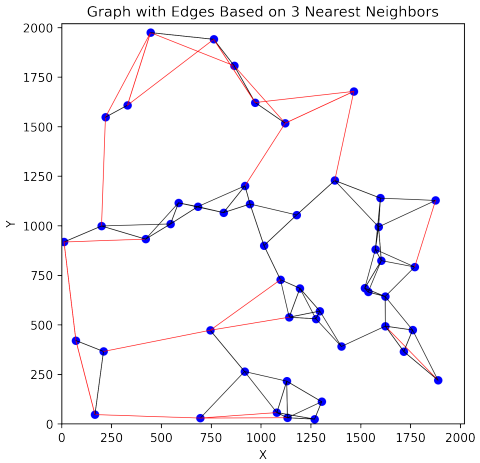
(a) Example graph using Delaunay Triangulation with non-uniform node distribution.



(b) Example graph using K-Nearest Neighbors (k = 3) with non-uniform node distribution.



(c) Example graph using Delaunay Triangulation with uniform node distribution.



(d) Example graph using K-Nearest Neighbors (k = 3) with uniform node distribution.

Figure 2.3: Example graphs created via Delaunay triangulations and K-Nearest Neighbors under non-uniform and uniform node distributions. These edge generation techniques may be misaligned with cell-communication patters and require a more sophisticated decision rule. Red edges represent edges with a distance of greater than 350.

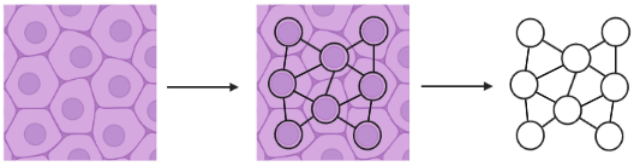


Figure 2.4: A graph overlaid on spatial transcriptomics data. Nodes are placed at cell positions and edges are created based on a neighborhood criterion.

2.4.3 Graph Convolution

GCNs represent a family of functions that map the node attributes of a graph, together with trainable parameters, to transformed node attributes. Equivalently, GCNs can be viewed as functions parameterized by weights that take node-level inputs and produce node-level outputs. Unlike traditional neural networks, GCNs operate directly on graph-structured data rather than on tabular or grid-like inputs (Chami et al., 2021). GCNs are spatially aware and often rotation-invariant, making them state-of-the-art tools for deep learning on graphs (Mac and Nguyen, 2021). Stacking convolutional layers allows GCNs to integrate information from nodes progressively further away, effectively modeling how signals propagate through interconnected structures such as biological pathways. The final GCN output at each node encodes both immediate neighborhood interactions and more complex, long-range structural relationships within the graph.

For DeepST, we use a GCN based on the Gaussian mixture model convolutional operator (GMMConv) (Monti et al., 2016). Our full GCN is designed by stacking these operators in layers. GMMConv is defined in terms of many simpler units, referred to as "kernels." Each kernel k is associated with a matrix Θ_k, a mean vector μ_k, and a covariance Σ_k. The GMMConv operator uses the kernel parameters, together with edge attributes ($\mathbf{e}$) and node attributes ($\mathbf{x}$) to produced transformed node attributes ($\mathbf{x}'$). Specifically, for node i, let $\mathcal{N}(i)$ denote the set of its neighboring nodes (i.e., the set of other nodes with an edge connecting them to node i). Then, the GMMConv operator computes the transformed node attributes according to the equation

$$\mathbf{x}'_i = \frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} \frac{1}{K} \sum_{k=1}^{K} \mathbf{w}_k \left(\mathbf{e}_{i,j}\right) \odot \Theta_k \mathbf{x}_j, \tag{2.2}$$

where each $\mathbf{w}_k$ is a weighting function defined by

$$\mathbf{w}_k(\mathbf{e}) = \exp\left(-\frac{1}{2}\left(\mathbf{e} - \mu_k\right)^\top \Sigma_k^{-1} \left(\mathbf{e} - \mu_k\right)\right). \tag{2.3}$$

Averaging all of these learned signals over all kernels and neighbors provides us the final output for a GMMConv layer. The analogy between the gaussian mixture model convolution and the ligand signals sent to the target cell is visualized in Figure 2.5. The added flexibility of letting each pair of connected nodes in a graph have a weight learned by a network has been shown to outperform other convolutional approaches to applied deep learning problems including classical Euclidean convolutional neural networks (CNN), spectral CNNs, GCNs, and diffusion CNNs (DCNN) (Monti et al., 2016).

$$\mathbf{x}_i' = \frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} \frac{1}{K} \sum_{k=1}^{K} \mathbf{w}_k \left(\mathbf{e}_{i,j} \right) \odot \mathbf{\Theta}_k \mathbf{x}_j$$

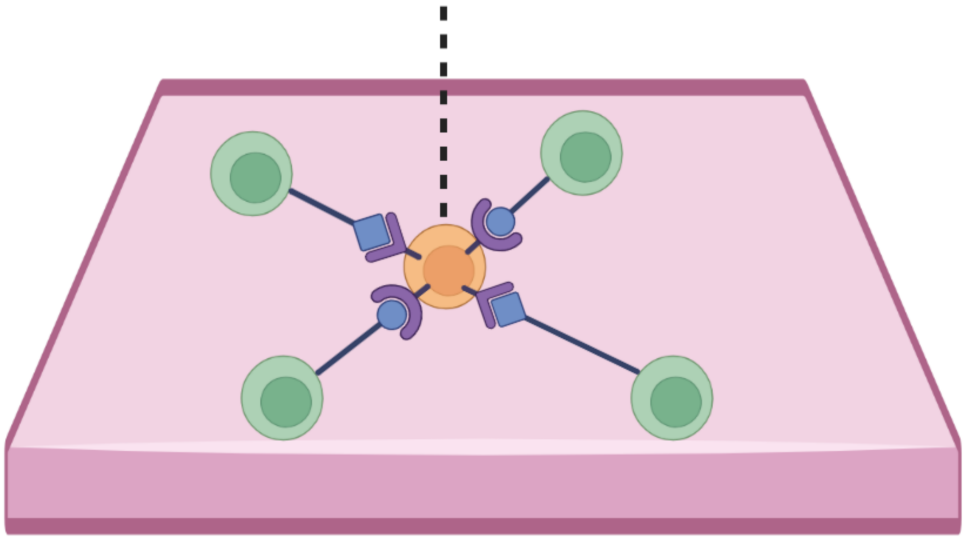


Figure 2.5: Example gaussian mixture model convolution. The target cell (orange) receives signals of some transformation of expressions from the signaling cells (green). The convolution $\mathbf{w}_k \left(\mathbf{e}_{i,j} \right) \odot \Theta_k \mathbf{x}_j$ is represented as the sum over K kernels parameterized by the weights w_k learned using edge weights $e_{i,j}$ that relates a specific signal to the target.

The architecture of the GCN in the DeepST model contains three hidden layers each constructed as the sum of a GMMConv and a fully connect layer followed by a ReLU activation:

$$h_{i+1} = \text{ReLU}(\text{GMMConv}(h_i, \texttt{edge\_index}, \texttt{edge\_attr}) + W_i h_i).$$

We use polar coordinate vectors describing the distance between 2 cells as the edge attributes

$$\mathbf{e}_{i,j} = \left(\sqrt{(x_j - x_i)^2 + (y_j - y_i)^2},\ \arctan\left(\frac{y_j - y_i}{x_j - x_i} \right) \right)$$

and use $K = 10$ kernels for each GMMConv layer. The GCN is trained with an Adam optimizer with a learning rate of 0.001 and ended early upon observing 10 training epochs with no improvement on the validation set. Further details about model parameters can be found in Appendix A.2.

2.4.4 The DeepST Model

So far, we have shown how tissue sections of ST data can be represented as graphs and how we can perform a GMMConv on every node. Here, we complete our pipeline—as presented in Figure 2.6—by detailing how DeepST extracts spatial patterns and quantifies its predictive accuracy for response gene expressions.

We can understand this approach more formally as performing model selection. Let $C = \{j : j \in \mathcal{N}(c)\}$ denote the neighboring cells of a target cell c. Our goal is to estimate the expression of a response gene g for a target cell c, denoted $Y_{c,g}$, using the total ligand and receptor expression of the neighborhood, $X_{C,L}, X_{C,R}$, and (optionally) cell-type metadata from those same neighbors, M_C.

For each radius r and each response gene g, we consider the hypothesis H_r that the conditional expectation of the response gene expression at each cell can be modeled as

$$\mathbb{E}[Y_{c,g} \mid X_{C,L}, X_{C,R}, M_C] = f_r(X_{C,L}, X_{C,R}, M_C), \tag{2.4}$$

where f_r refers to the DeepST GCN trained on graphs with neighborhood radius r.

We can evaluate the hypothesis H_r for different values of r by measuring how closely the corresponding DeepST predictions match the true observed values. Specifically, for each gene g and each method r, we calculate the mean squared difference between the actual observed gene expression, $Y_{c,g}$, and the predicted expression produced by DeepST, $f_r(X_{C,L}, X_{C,R}, M_C)$, within our test dataset. This calculation gives us an unbiased estimate of

$$\mathbb{E}\left[\left\| Y_{c,g} - f_r(X_{C,L}, X_{C,R}, M_C) \right\|^2 \,\middle|\, X_{C,L}, X_{C,R}, M_C \right],$$

which we denote as $\hat{\sigma}^2_{r,g}$. The true conditional expectation would minimize this mean squared error (see Appendix A.4.1). Therefore, we suggest rejecting the hypothesis H_r (that DeepST with radius r accurately reflects true expression patterns) if we can find another value r'

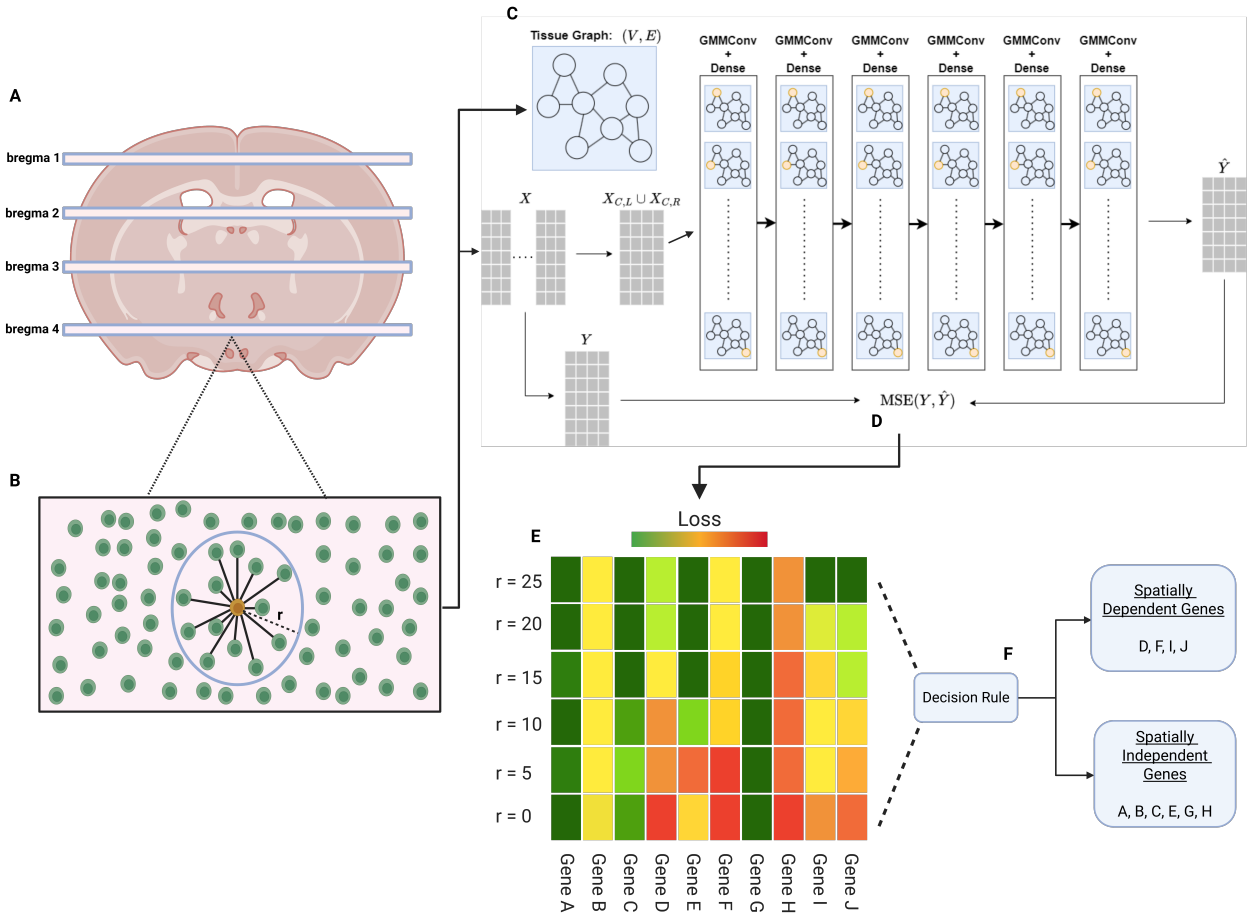


Figure 2.6: The DeepST modeling pipeline. **A**) Tissue extraction. Source tissues are collected from the data source at varying locations. **B**) Graph creation. For each target cell i (orange) measured in the tissue graphs, a neighborhood is created by adding bidirectional edges from i to all cells less than r µm away. **C**) Forward model. Each graph is passed into the model, only including cell ligands and receptors as input. After several convolution and dense layers, the model outputs predicted values for the response genes. **D**) Model evaluation. The true response expressions for all genes across all cells (Y) are compared against DeepST's output via the MSE ($\hat{Y}$). **E**) Spatial analysis. DeepST model evaluation for multiple neighborhood radii. Downstream analysis with this information can help identify spatially variable genes that are affected by CCCs. **F**) Identifying spatially dependent genes. By comparing model performances from a spatially ignorant ($r = 0$ µm) and a spatially aware ($r > 0$ µm, 25 µm depicted in figure), we can craft a decision rule that separates genes with inferred spatial dependence from those without.

whose estimated mean squared error $\hat{\sigma}_{r',g}$ is substantially smaller than $\hat{\sigma}_{r,g}$. For DeepST, we are particularly interested in the case where $\hat{\sigma}_{r,g} \ll \hat{\sigma}_{0,g}$ for some r, as it suggests that a spatially ignorant model may be inadequate to explain the gene expression. This, in turn, suggests that the expression of gene g may exhibit spatial variations.

The architecture for the DeepST model is visually represented in Figure 2.6C. By stacking multiple layers, DeepST can capture multi-hop signaling pathways, allowing the model to infer how chains of CCCs propagate gene expression changes beyond solely direct neighbors.

Each GMMConv layer effectively returns a weighted sum where each neighboring gene expression is provided a unique weight learned by the network. This allows the network to learn which neighbors and neighboring features of a target cell are important for predicting response genes. Additionally, this avoids potential pitfalls in assuming an incorrect spatial structure between cells because the weights of the GMMConv can properly weigh the neighboring contributions.

DeepST leverages residual connections to prevent vanishing gradients when using deeper architectures. Specifically, the output of the k^{th} hidden layer is connected to the $(K-k+1)^{\text{th}}$ layer (which ensures symmetric U-Net style connections). We defer discussion of complete model details to Section A.1.

DeepST is evaluated on the mean squared error (MSE) between the true target response expressions Y and DeepST's output $\hat{Y}$ (Figure 2.6D):

$$\text{MSE}(Y, \hat{Y}) = \frac{1}{CG} \sum_{c=1}^{C} \sum_{g=1}^{G} (Y_{c,g} - \hat{Y}_{c,g})^2. \tag{2.5}$$

The radius of consideration (r) is a hyperparameter in our graph construction, which allows us to better understand the radius of influence in cell communication. Evaluation across every gene and several candidate values for r identifies genes whose response genes are reconstructed better with the inclusion of spatial information (Figure 2.6E). As we will demonstrate in the next section, sequential modeling across multiple graphs with varying r values is less necessary with DeepST than with previous methods.

2.4.5 Hypothesis Testing

Ultimately, our goal is to identify spatially dependent genes, so we require an appropriate decision rule to reach and support such conclusions. Evaluating DeepST's performance through MSE reduction provides direct insight into spatial gene dependence. Minimizing the MSE of predictions generated by DeepST is equivalent to maximizing a fixed scale Gaussian distribution. Therefore, we can design a hypothesis test as a principled approach to identify

genes with significant spatial effects and prioritize them for further biological validation.

We will demonstrate in the coming sections that DeepST is a more appropriate model for spatial dependence inference. Proceeding from this, we can use DeepST predictions in the following hypothesis test:

$$H_0 : \mu_{c,g} = f_{r=0}(X_{C,L}, X_{C,R}, M_C) \text{ v. } H_1 : \mu_{c,g} = f_{r=r^*}(X_{C,L}, X_{C,R}, M_C).$$

For shorthand we will define $\mu_0 := f_{r=0}(X_{C,L}, X_{C,R}, M_C)$ and $\mu_{r^*} := f_{r=r^*}(X_{C,L}, X_{C,R}, M_C)$.

Because the models already provide predictions and we assume fixed variance, we can compute the likelihood ratio directly:

$$\Lambda(Y_{c,g}) = \frac{\mathcal{L}(\mu_0 \mid Y_{c,g})}{\mathcal{L}(\mu_{r*} \mid Y_{c,g})}.$$

Therefore, a likelihood ratio test (LRT) can be performed between a spatially ignorant model and a spatially aware one. We prove that this LRT is proportional to the difference in squared errors under each model, given by:

$$(Y_{c,g} - \mu_{r^*})^2 - (Y_{c,g} - \mu_0)^2.$$

Furthermore, If we use the decision rule $\Lambda < c$ to reject the null hypothesis such that $P(\Lambda < c \mid H_0) = \alpha$ and fail to reject otherwise, then—by the Neyman-Pearson lemma—this is the most powerful α-level test for detecting whether incorporating spatial information improves prediction accuracy under the Gaussian likelihood assumption (Neyman and Pearson, 1933). The proof of this claim as well as derivations of the test statistic, its distribution, and satisfaction of the Neyman-Pearson lemma assumptions can be found in Appendix A.4.

Using $\mathcal{L}(\mu, \sigma)$ to denote the Normal likelihood with mean μ and scale σ, we can create the following likelihood ratio statistic leveraging our predictions:

$$\Lambda = \frac{\mathcal{L}\left(f_{r=0}(X_{C,L}, X_{C,R}, M_C), \sigma\right)}{\mathcal{L}\left(f_{r=r^*}(X_{C,L}, X_{C,R}, M_C), \sigma\right)}.$$

Methods preceding ours could also be used to perform such a test, but we will show that their handling of spatial information is often less effective, potentially leading to inflated false positive rates or incorrectly ordered spatial gene rankings. In contrast, DeepST leverages GCNs to better capture spatial dependencies, resulting in improved predictive accuracy and a more reliable evaluation of spatial effects. Providing a statistically rigorous way to rank genes helps determine genes to prioritize when allocating assets for gene research.

2.5 Model Comparison with Spatial Transcriptomics Data

We assessed DeepST's performance on two large-scale, real, spatial transcriptomics datasets. The first is the MERFISH hypothalamus dataset (Moffitt et al., 2018). This dataset provides high-resolution single-cell spatial data, including approximately 1 million cells profiled across the hypothalamic preoptic region of 36 different mice. These cells are collected across several tissue slices across animals and across different locations within an animal (Figure 2.6A). For each cell, the data provides its (i) slice, (ii) position, (iii) cell type, and (iv) gene expression for 161 genes (71 ligands and receptors; 84 responses; 6 controls).

The second dataset we analyze is the Fresh Frozen Mouse Brain Replicates dataset provided by the Xenium platform (10x Genomics, 2023a). In contrast to earlier spatial transcriptomics platforms, Xenium data has both high spatial resolution and measurements of more genes. The Fresh Frozen Mouse Brain dataset contains 248 relevant genes, 100 of which are ligands and/or receptors. This dataset does not include ligand and receptor annotations, so to classify genes as ligands, receptors, or response genes, we query each gene in each of the following databases to determine if a gene is a known ligand or receptor: OmniPath Intercellular Roles and OmniPath Ligand-Receptor Interaction Database (Türei et al., 2021), ConnectomeDB2020 (Hou et al., 2020), and the Jin 2021 Mouse Ligand-Receptor Compendium (Jin et al., 2021). These curated sources provide broad and complementary coverage of ligand and receptor annotations. Furthermore, in contrast to technologies like MERFISH, which image tissue one thin section at a time and often space those sections tens of microns apart, Xenium captures multiple adjacent sections with continuous spatial coordinates in all three dimensions. This third spatial dimension allows us to understand DeepST's performance in settings with additional directional information that is not planar. Signals from neighboring cells can come from any direction, so we assess DeepST's ability to consider all plausible cell-cell communication (CCC) interactions.

Genes in the real spatial transcriptomics datasets we consider are categorized into four groups: ligand, receptor, response, and control/blank. Control genes are associated with blank barcodes that are not assigned to any RNA and do not hold any relevant gene expression information. Ligand and receptor genes facilitate cell communication events. Response genes are defined as any other gene in the dataset that is not a control, ligand, or receptor gene. Because control genes lack biological relevance, we exclude them, using only ligand and receptor genes to predict downstream response gene expression.

Furthermore, we analyze how methods perform when a response gene follows a well-defined CCC regime. These semi-synthetic datasets preserve the locations provided in the

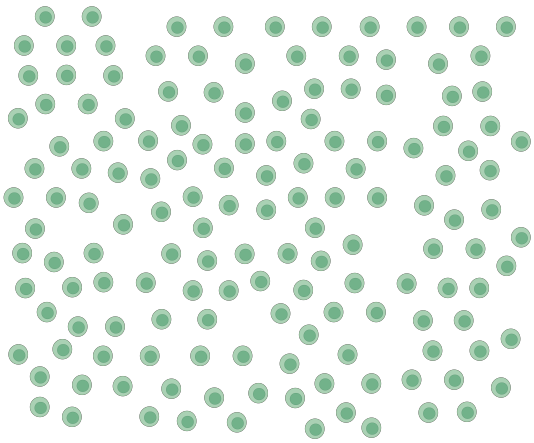


Figure 2.7: Cells Dispersed in Tissue (Before)

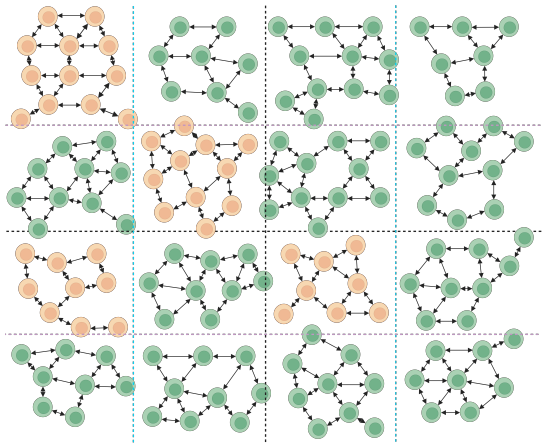


Figure 2.8: Partitioned Tissue Graphs (After)

Figure 2.9: Spatial graph construction and partitioning. In this example, tissue is divided into 16 non-overlapping spatial subgraphs by recursively splitting the tissue spatially at the midpoint (median coordinate) of each axis. Each subgraph is an observation in the batched dataset. These disjoint graphs are split into a training and validation sets (both in green) and a testing set (orange).

MERFISH hypothalamus dataset but we simulate the node attributes. This allows for a controlled evaluation of modeling spatial dependencies.

To showcase the necessity of strong spatially varying models, we also compare the predictive power of the DeepST model with prediction schemes that use fixed-dimensional encodings, including LightGBM, MESSI, and several linear models. In these fixed-dimensional approaches, the response gene in each target cell is predicted from (i) gene expression in the target cell, (ii) average neighborhood gene expression, and (iii) average neighborhood metadata (e.g., the relative abundance of different cell types in the neighborhood). We compare the predictive performance of all models in real and semi-synthetic data. Our results demonstrate that DeepST effectively incorporates spatial context to improve response gene expression predictions, especially in settings where spatial dependencies are pronounced.

All models are evaluated across a range of tissue graph radii to assess their ability to capture spatial dependencies. Comparison of model performance for various r values not only performs hyperparameter tuning, but can also identify genes whose expressions are influenced by CCCs over various distances. This distinction matters because CCC mechanisms function across varying spatial scales. For instance, the targets of paracrine and juxtacrine signals are within a close proximity. However, endocrine signals can be sent over long distances through the bloodstream before reaching a target. Any combination of these signals can be

responsible for an observed change in response gene expression; evaluating at several spatial scales ensures consideration of all potential influences. By evaluating the model across a range of radii, we can identify which neighborhood radius most likely reflect the true range of CCC influence and stress test which models can maintain performance when given too little or too many spatial signals.

2.5.1 MERFISH Hypothalamus Data

We applied our method and alternative methods to the mouse hypothalamus spatial transcriptomics dataset introduced in (Moffitt et al., 2018). This dataset consists of 36 animals which together contribute 181 tissue samples. Tissue samples were collected by slicing cross sectional mouse hypothalamus regions at various bregma values. The various slices were about 50μm apart and each slice is approximately of size 2000μm wide and long. For this application, we decided to take 29 animals (157 tissues) as training examples and 6 animals (24 tissues) as testing examples. One animal was held out of the training set for validation purposes.

We used the expression levels of ligand and receptor genes as covariates; these are known to facilitate intercellular communication. A common practice is to log transform the standardized count data for normalization purposes. This data transformation has been shown to produce models that have the lowest Type I error out of the typical data transformations (Ives, 2015). Additionally, due to the zero-inflated nature of gene expression data, the exact transformation utilized is $\log(1 + X)$, otherwise known as the log1p transformation. This manipulation maintains the zero counts while keeping the benefit of logarithmic normalization.

We trained each method to predict the expression levels of all other genes. Notably, several existing methods cannot output multi-gene outputs and have models tailored to predict each gene individually, providing an advantage to these models over DeepST—a one size fits all model that learns all response genes at once. Each cell in the MERFISH hypothalamus dataset is associated with one of 16 cell types. These cell types were identified by the authors and manually annotated for downstream tasks. Given that these cell type annotations help characterize specific behaviors via gene activity, they should be included as inputs for modeling. However, these annotations are not ubiquitous in ST datasets and could have significantly more than just 16 cell types. Excluding cell type annotations allows models to generalize to large-scale ST datasets that lack thorough manual annotations, but comes at the expense of excluding features that may be crucial for inferring CCC dependencies. In this work, we evaluate all models in both settings to assess their performance.

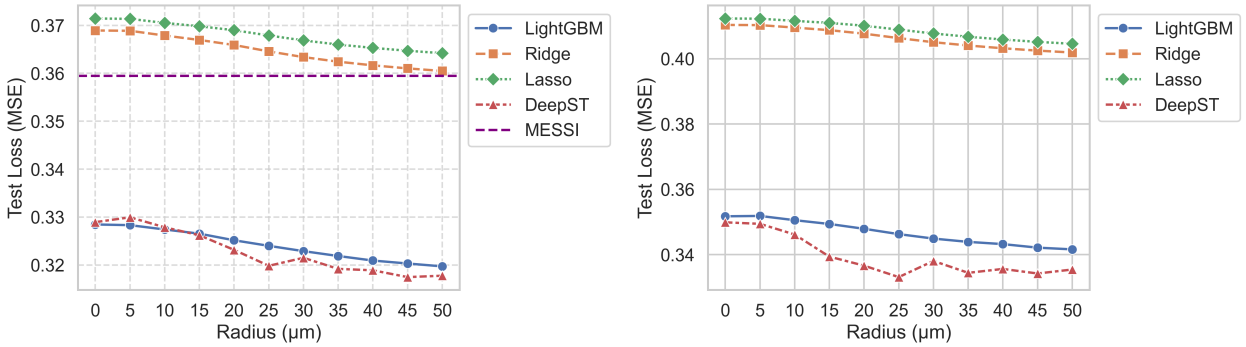
2.5.1.1 Results

We next examine predictive model performance on the MERFISH dataset in two settings: with and without cell-type metadata. Cell-type annotations for each cell of the MERFISH hypothalamus dataset were provided by the authors. On the one hand, cell-type annotations can help characterize specific behaviors via gene activity. On the other hand, cell-type annotations are inconsistent across spatial transcriptomics datasets and may be a source of data leakage, as the response gene may have been used to infer cell types. They vary in how they draw boundaries between types and in the level of granularity used to define the taxonomy. We do not take a firm stance on whether cell-type annotations should be included as predictors, and instead report results for both settings while acknowledging the limitations and advantages of each one. Figures 2.10a and 2.10b showcase performance on six animals in our test set, evaluated across increasing neighborhood radii. Figure 2.10a displays the test results from models trained with cell type metadata, while Figure 2.10b shows results from models trained without it. Since MESSI requires filtering by cell type, we report its performance only in the setting where cell types are included. For MESSI, we selected the 'Excitatory' cell type, with Delaunay triangulation as the cell neighborhood. Our results indicate that our model can achieve a 4.79% reduction in loss when cell types are absent and a 3.00% reduction when cell types are included when including spatial information. When cell types are included, LightGBM performs nearly identically to DeepST, which is expected since cell type annotations are strong indicators for CCC influences by capturing shared signaling environments and constraining plausible interaction partners (Cable et al., 2022). This is because inferred CCC interactions often rely on known ligand-receptor pairs, and these interactions are typically enriched within and between specific cell types. This is further evidenced by the fact that, when cell types are excluded, all models—including the best-performing ones—have a higher MSE than the spatially ignorant LightGBM or DeepST baselines. However, in the setting where cell types are omitted, DeepST manages to stabilize quickly and achieve a larger reduction than all competing methods.

2.5.1.2 Gene Ranking

Inferences about the spatial dependence of gene expressions has relevant applications in understanding cellular heterogeneity within tissues and understanding the multicellular environment of tumors (Satija et al., 2015a; Tirosh et al., 2016).

In this section, we explore using DeepST to reveal genes that are conditionally independent of neighboring genes. A possible way to identify conditional independence structure between cells can be assessed by pairing spatially aware models with their spatially ignorant



(a) Test MSE of DeepST and Classic Competitors **including** cell type information.

(b) Test MSE of DeepST and Classic Competitors **excluding** cell type information.

Figure 2.10: Comparison of test MSE for DeepST and classic competitors with and without cell type information on the MERFISH hypothalamus dataset.

counterparts. To identify instances of spatially dependent genes, we rank genes based on the contrast between DeepST's predictions and those of a baseline prediction method that bases its predictions only on the attributes of the cell (and not its neighbors). We compare the MSE loss between the baseline models which have no neighbors and the model with the lowest MSE across all r values for each response gene. Target expressions that are independent of neighboring expressions would yield similar losses in the spatially aware and spatially absent models. Target expressions that are spatially dependent however should see a decrease in loss in spatially aware models that include all relevant cell-cell communication events.

For each response gene, we compared DeepST's predictive performance with the performance of the baseline estimator. Figure 2.12 shows the differences. NNAT and MBP are the genes that stand out the most, with DeepST achieving 39.59% and 24.81% reduction in mean squared error, respectively, by considering neighboring signals. MBP has been corroborated to be spatially variable in the brains of mice (Oxford Nanopore Technologies, 2023). NNAT has been demonstrated to have spatial and temporal patterns during mouse eye development (Sel et al., 2017). Across all the response genes, 91.6% of them experienced an improvement from leveraging spatial information. Figure 2.11 presents MSE improvements for all response genes in the dataset. The median loss reduction was 2.77% (mean: 4.56%). These consistent MSE reductions allow us to apply the likelihood ratio test as a formal decision rule for identifying spatially variable genes, separating truly spatially variable genes from those whose improved performance with spatial information may not be distinguishable from random chance.

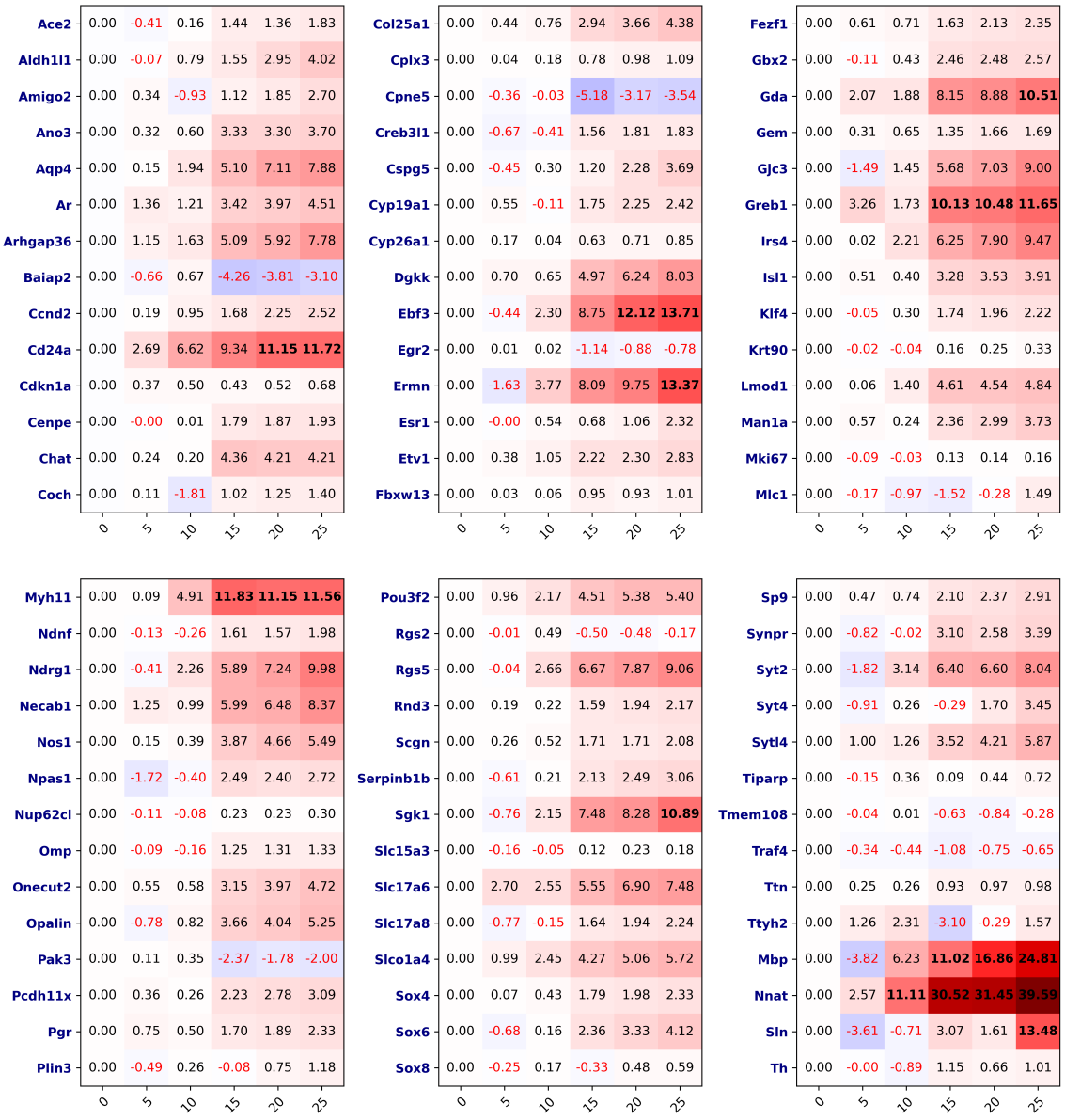


Figure 2.11: Relative improvements in gene prediction accuracy from DeepST trained on graphs with radius of consideration $r = 0\,\mu$m up to $r = 25\,\mu$m. The heat map highlights genes that benefit from spatial information, with warmer colors indicating greater improvement. Positive values (black) represent a reduction in MSE relative to the model trained without spatial information ($r = 0\,\mu$m), while negative values (red) indicate an increase in error.

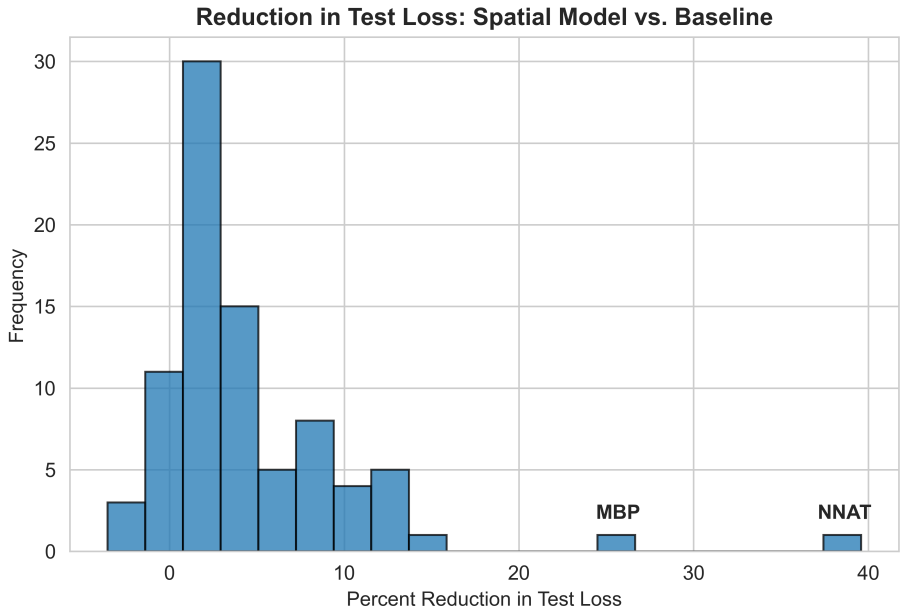


Figure 2.12: Histogram of test loss reductions moving from a graph neural network with radius 0 μm to one with 25 μm.

2.5.2 Xenium Fresh Frozen Mouse Brain Data

Having demonstrated DeepST's performance on Visium data, we now evaluate it on Xenium, a more recent platform offering higher resolution and depth-resolved spatial transcriptomics. In contrast to technologies like MERFISH, which image tissue one thin section at a time and often space those sections tens of microns apart, Xenium captures multiple adjacent sections with continuous spatial coordinates in all three dimensions. This third spatial dimension allows us to understand DeepST's performance in settings with additional directional information that is not planar. Signals from neighboring cells can come from any direction, so we assess DeepST's ability to consider all plausible CCC interactions.

We evaluate each model family's ability to leverage spatial information by varying the radius of consideration used to define neighboring context. Model performance across radii is presented in Table 2.2.

Just as with our evaluation on the MERFISH dataset, we include several linear models as baseline references but LightGBM serves as the primary non-graph-based benchmark due to its competitive performance and scalability. Despite both LightGBM and DeepST achieving reductions in test error with additional spatial information, DeepST's baseline is roughly the same as LightGBM's performance with 30μm of neighboring signal. This is concerning because practitioners often rely on performance gaps between models with and without spatial context to infer spatial dependence. DeepST's capacity to model nonlinear

Model / r	**0**	**5**	**10**	**15**	**20**	**25**	**30**
Ridge	0.258±0.010	0.258±0.010	0.256±0.011	0.250±0.011	0.241±0.011	0.235±0.011	0.231±0.011
Lasso	0.505±0.006	0.505±0.006	0.505±0.006	0.505±0.006	0.505±0.006	0.505±0.006	0.505±0.006
Elastic Net	0.275±0.008	0.275±0.008	0.274±0.008	0.271±0.008	0.266±0.008	0.264±0.008	0.262±0.008
LightGBM	0.194±0.007	0.194±0.007	0.192±0.008	0.188±0.008	0.183±0.008	0.181±0.008	0.179±0.008
DeepST	**0.176±0.011**	**0.175±0.009**	**0.175±0.011**	**0.172±0.010**	**0.168±0.009**	**0.166±0.009**	**0.166±0.009**

Table 2.2: Test MSE for various methods and neighborhood radii on the Xenium Fresh Frozen Mouse Brain dataset.

and hierarchical relationships allows it to make informative predictions based solely on a cell's intrinsic features. Therefore, when DeepST exhibits improved performance at higher radii, we can be more confident that these gains reflect genuine spatial effects, rather than compensating for a model's inability to accurately leverage a cell's intrinsic features.

We emphasize that in the evaluation of methods on both the MERFISH and Xenium Datasets, LightGBM is predicting each response gene individually. In contrast, DeepST predicts all response genes simultaneously in a single forward pass, learning shared representations that generalize across genes and reducing the computational burden associated with training separate models. We expect that a gene-specific variant of DeepST would manage to yield better predictions but at the cost of scalability and shared parameters. DeepST's joint prediction of all response genes in a single forward pass offers a computational advantage over LightGBM, which trains a separate decision tree model per response gene. We can assess which response genes from the Xenium dataset exhibit spatial dependence by comparing a DeepST with $r = 30\mu$m to a spatially ignorant DeepST as a baseline $r = 0\mu$m. Roughly 80% of response genes are better predicted with a spatial model than the baseline.

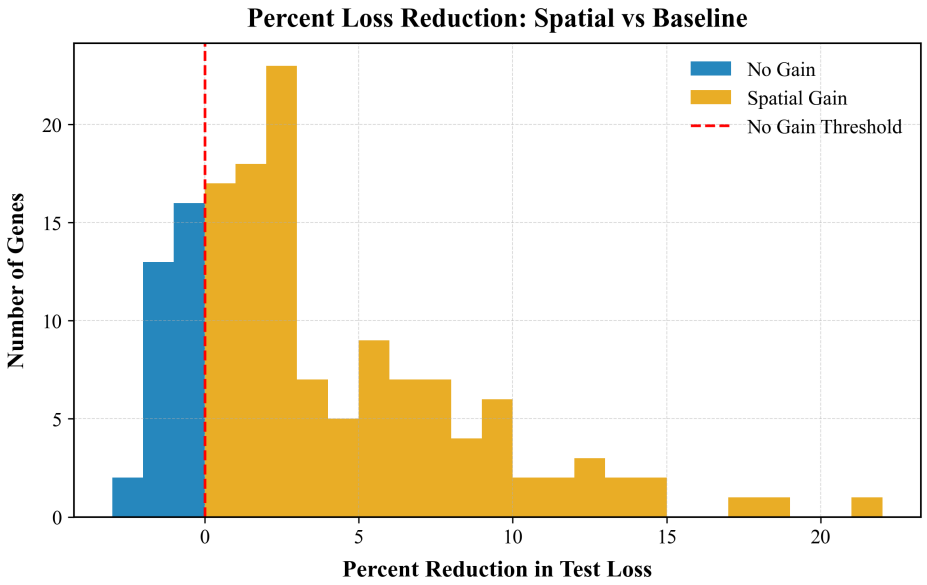


Figure 2.13: Histogram of test loss reductions moving from DeepST trained with $r = 0$ μm tissues to DeepST trained with $r = 30$ μm on the Xenium fresh frozen mouse brain dataset.

Gene	Change
CABP7	21.00%
GFAP	18.16%
ARC	17.22%

Table 2.3: Top response genes and their percentage decrease in test MSE from incorporating spatial information in DeepST.

Figure 2.13 demonstrates that most responses are better predicted with spatial information and highlights three top candidate genes exhibiting spatial dependence. These genes and their exact performance improvements are outlined in Table 2.3. The full decomposition is visualized in Figure 2.14.

The CABP7 gene is responsible for enabling calcium ion binding and has been shown to be expressed highly in subregions of the mouse hippocampus (Shi et al., 2023). The GFAP gene has been shown to be distinctly affected in Huntington's disease and appear with spatially distinct expression profiles (Brown et al., 2023). Lastly, the ARC gene has been shown to be primarily concentrated in the CA1 region of the hippocampus, more than any other region of the brain (Shi et al., 2023). This aligns with existing research suggesting the upregulation of the ARC gene is essential for spatial learning and supports long-term memory storage (Gao et al., 2018).

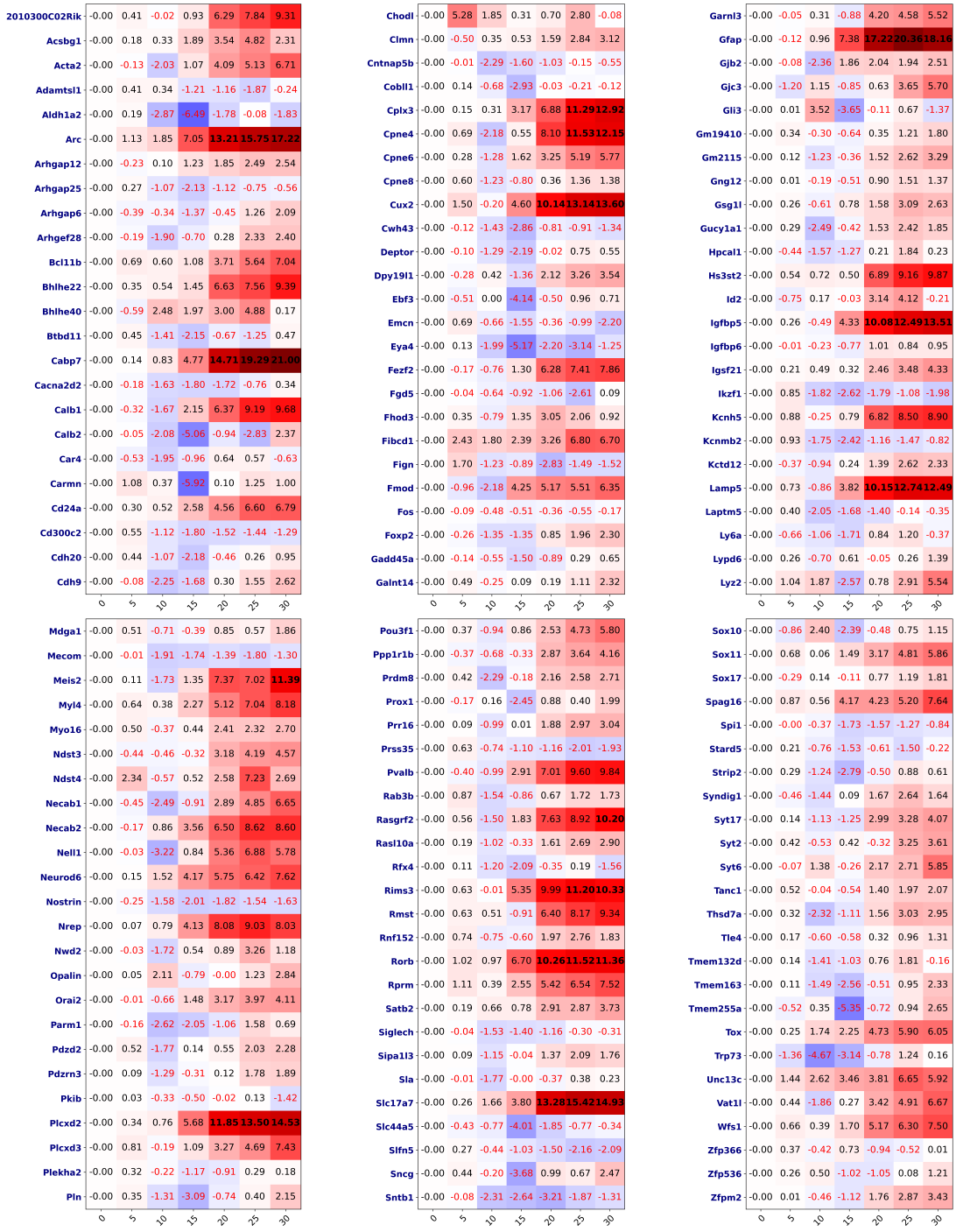


Figure 2.14: Relative improvements in gene prediction accuracy from DeepST trained on graphs with radius of consideration $r = 0\,\mu$m up to $r = 30\,\mu$m for the Xenium dataset. The heat map highlights genes that benefit from spatial information, with warmer colors indicating greater improvement. Positive values (black text) represent a reduction in MSE relative to the model trained without spatial information ($r = 0\,\mu$m), while negative values (red text) indicate an increase in error.

These results validate DeepST's effectiveness on high-resolution spatial transcriptomics data. This indicates DeepST is a scalable approach capable of assessing many response genes accurately in parallel for large, densely sampled tissues. Furthermore, we reinforce that DeepST has value in datasets that lack curated cell type information but preserve rich spatial structure.

2.5.3 Semi-Synthetic Experiments

We have demonstrated so far that DeepST and LightGBM yield superior response gene expression predictions relative to MESSI and classical regression techniques on modern ST datasets. In this section, we develop experiments on semi-synthetic data to explore how our model can recover response gene expressions in several regimes of cell-cell interactions.

We use semi-synthetic data to analyze how methods perform when genes follow a well-defined CCC pattern. These semi-synthetic datasets preserve the locations provided in the MERFISH hypothalamus dataset, but we simulate the gene expression. This allows for a controlled evaluation of different modeling strategies. We start by generating i.i.d. expressions for all genes for each cell. Then, the data is perturbed to reflect a predefined correlation structure between neighboring signal cells and their targets.

For each experiment, we assume the task is to predict a single response gene. The true response gene expression is calculated as a function of a selected ligand's expression from all neighbors of the target cell. For this set of experiments, we assumed a true neighborhood structure of ($r = 30$) and treat gene 0 ($g = 0$) as the response gene, while all other genes are considered ligands or receptors. Ideally, a model that accurately recovers the response expression would be able to do so when given all of the necessary predictors. We demonstrate that—when the appropriate neighborhood is considered—there exist settings where DeepST recovers the true relationship but other methods do not.

We start with a deterministic setting (#0) designed to test if models can recover a known, noiseless signal. The response gene ($g = 0$) expression is set to the sum of gene 1 ($g = 1$) expression of all neighboring cells if the total exceeds 1; otherwise, it is 0 (eq. (2.6)). This relationship is important for any model to detect because in practice ligand-receptor binding is not a deterministic process; the probability of the communication transpiring is often related to the expression of a ligand in neighboring cells (Cera, 2020). So, if a model cannot learn this deterministic relationship, it will likely be unable to make valuable predictions when the response expression is naturally noisy. In our notation, this relationship is given by

$$
\begin{aligned}
X_{c,g} &\sim \mathrm{NB}(1, 0.5)/5 \\
Y_{c,0} &= \mathbb{1}\left(\sum_{c' \in \mathcal{N}(X_c)} X_{c',1} > 1\right) * \sum_{c' \in \mathcal{N}(X_c)} X_{c',1}.
\end{aligned} \tag{2.6}
$$

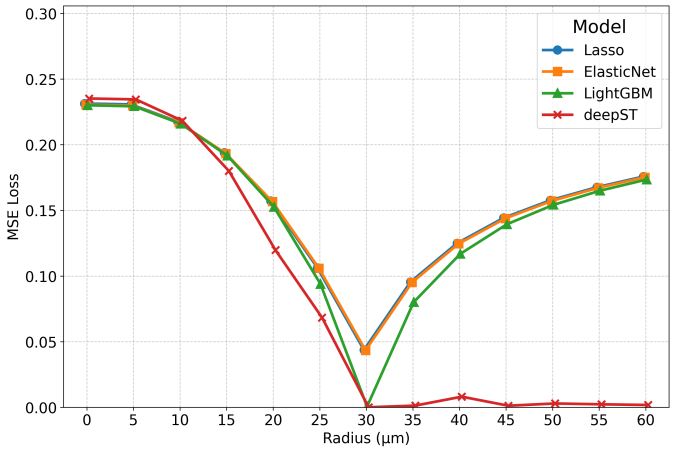


Figure 2.15: Synthetic setting #0 test losses. The true data generating process has a neighborhood radius of $r^* = 30$ µm.

For our first non-deterministic synthetic setting (#1), we sample all gene expressions (except the response) i.i.d. from an exponential distribution. The response gene expression is set to the sum of neighboring $g = 1$ expressions if it exceeds 1; otherwise, it is 0. Exponential noise is then added to finalize the values (eq. (2.7)). In our notation, this relationship is given by

$$
\begin{aligned}
X_{c,g} &\sim \exp(10), \;\; g = 1, 2, \ldots, G, \\
Y_{c,0} &= \left(\mathbb{1}\Big(\sum_{c' \in \mathcal{N}(X_c)} X_{c',1} > 1\Big) \sum_{c' \in \mathcal{N}(X_c)} X_{c',1}\right) + \varepsilon_c, \\
\varepsilon_c &\sim \exp(10)\,.
\end{aligned} \tag{2.7}
$$

The first two synthetic relationships are trivial as evidenced by the results displayed in Figures 2.15 and 2.16. GCNs and boosting techniques almost perfectly recover the true relationship between neighboring and target expressions. All models achieve their best performance at $r = 30\mu$m and gain performance advantages on unseen data as r increases from

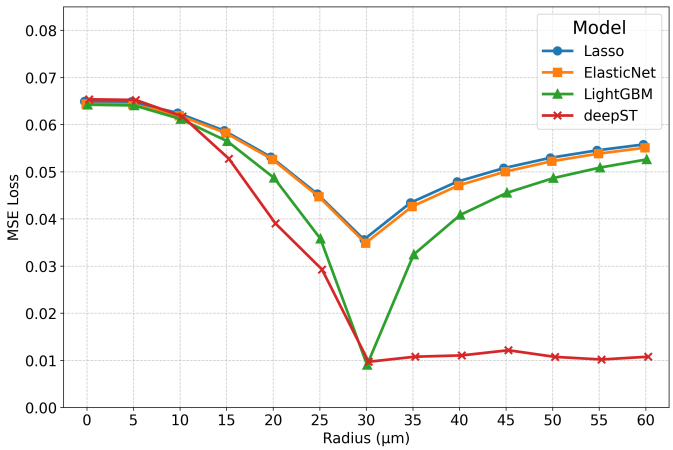


Figure 2.16: Synthetic setting #1 test losses. The true data generating process has a neighborhood radius of $r^* = 30$ µm.

0μm to 30μm. However, only DeepST is able to achieve optimal performance when given a surplus of neighboring information ($r > 30$). Including neighboring information that is as little as 5μm above the true radius that generated the synthetic data results is nearly as detrimental to model performance as excluding all data between 25μm and 30μm for all competing methods.

The advantage of the GCNs becomes more profound when the relationships between target responses and neighboring ligands are more complicated and nonlinear. We consider a hierarchical model where each gene has a negative binomial distribution centered at a mean drawn from a normal distribution. The function in equation (2.8) represents the influence of ligands from neighboring cells as a nonlinear, inverse function of their distance from the target cell, decaying from 1 to 0 over the interval $[0\mu\text{m}, 30\mu\text{m}]$. Additionally, the contribution of each neighboring cell is the square root of its ligand expression, emphasizing the impact of weaker signals:

$$\begin{aligned} \mu_1, \ldots, \mu_G &\sim N(20, 4), \\ X_{c,g} &\sim \frac{\text{NB}(\mu_g, 0.5)}{60}, \ g = 1, 2, \ldots, G, \\ Y_{c,0} &= \sum_{X_{c'} \in \mathcal{N}(X_c)} \sqrt{X_{c',1}} \left(1 - \frac{\sinh^{-1}\left(5.863\, d(X_c, X_{c'})\right)}{5.863}\right). \end{aligned} \tag{2.8}$$

Figure 2.17 demonstrates that in these settings, fixed-dimensional models not only have higher loss values than our GCN for $r > 15$, but their minimum losses occur at a radius differing from the actual interaction radius (20μm). In contrast, DeepST prediction error

monotonically decreases on the interval [0μm, 30μm] and then plateaus afterwards. This indicates that the network was able to tune out irrelevant spatial communications without overfitting. For the previous methods, the mean aggregation of neighboring signals is not enough to differentiate weak signals from strong ones. We hypothesize that the elevated error observed in fixed-dimensional models primarily results from their reliance on summarizing neighborhood gene expression by taking the *average* of neighboring expressions. Such averaging fails to capture the critical spatial relationships described by eq. (2.8), where gene expression is driven by a *sum* of neighboring values. Although it might be possible to construct carefully designed, fixed-dimensional summaries that better represent spatial dependencies, DeepST captures a wide variety of relationships easily, thus eliminating the need for manual summary design. In our analysis, while other feature sets and summaries were considered for competing methods, the mean tended to have the best performance in practice.

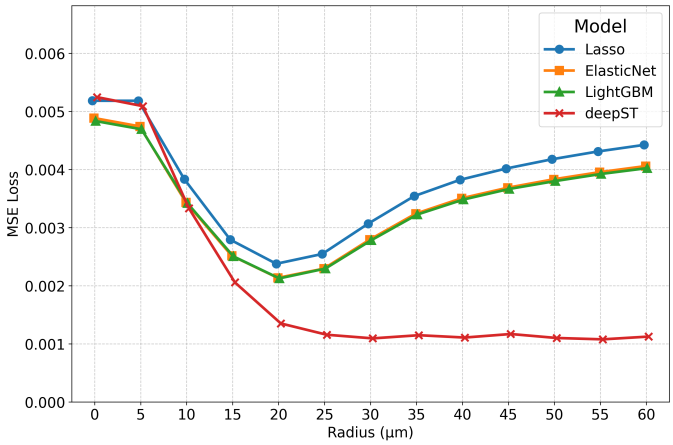


Figure 2.17: Synthetic setting #2 test losses. The true data generating process has a neighborhood radius of $r^* = 30$ µm.

In both experiments, the performance of DeepST remains stable after reaching a sufficiently high radius, but fixed-dimensional methods require more careful tuning of the radius parameter. In practice, the true radius of influence is unknown and may vary across cells and genes, exhibit multi-modal effects, or depend on complex biological interactions. This stability is desirable in practice, where the true radius of consideration is often unknown or variable across genes. Furthermore, the performance of fixed-dimensional methods is non-monotonic as a function of r. This makes it more difficult to interpret the results from the model selection perspective.

As we outlined in the Hypothesis Testing section, the decreases in MSE between baseline and spatially aware versions of the same model form the basis for a statistical test for

identifying spatially dependent genes. This means that if the spatial summary is incorrect or inflexible, performing a LRT with an incorrect r could lead to an incorrect conclusion. For instance, comparing LightGBM's baseline model against its $r = 30\mu$m or $r = 60\mu$m counterpart might yield conflicting conclusions due to the model's reliance on predefined spatial features rather than learned representations, even though spatial relationships clearly exist. However, with any $r > 30$, the conclusions yielded by DeepST are consistent and reliable. This means that a single trained DeepST instance is sufficient for robust inference as long as the chosen r is sufficiently large to capture relevant spatial dependencies.

2.6 Discussion and Further Applications

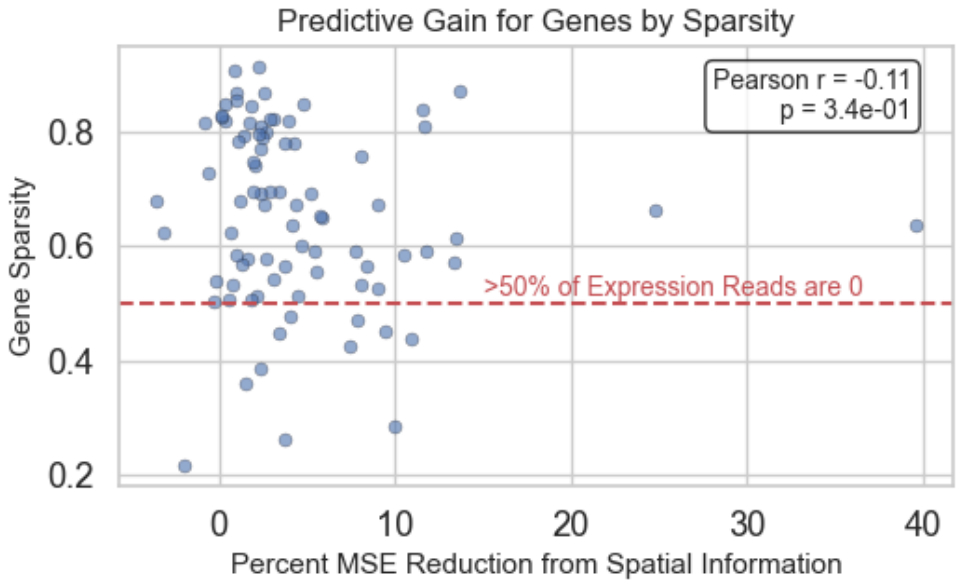
2.6.1 Data Sparsity

In this chapter, we designed DeepST to work out of the box for any dataset containing spatial information. However, in spatial transcriptomics, it is common for response genes to be extremely sparse. Even when DeepST includes a final ReLU layer, predicting exact zeros remains non-trivial for GCNs, whereas tree-based models like LightGBM more easily output values at or near zero. The capacity to output exact zeros as predictions may help DeepST generalize well to zero-inflated response distributions.

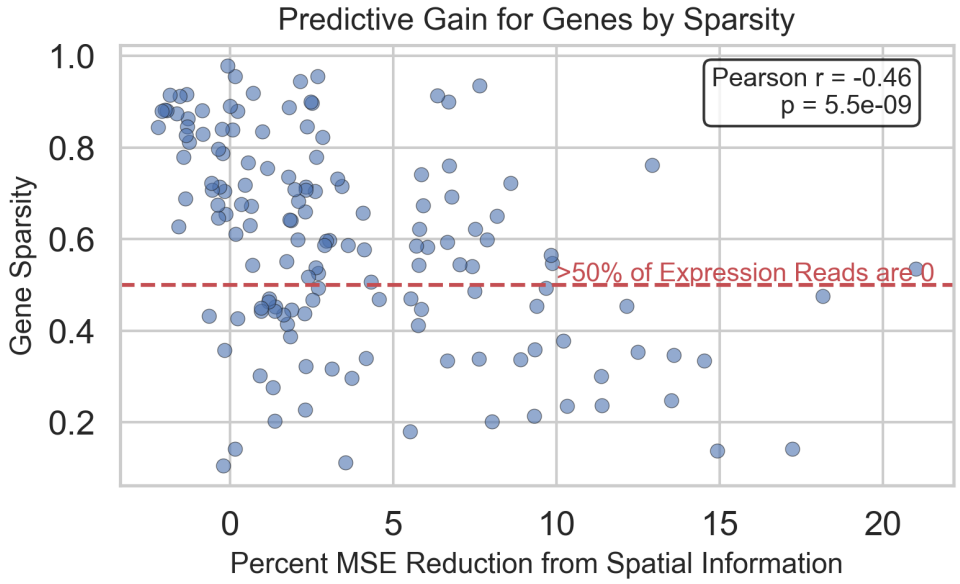
Figures 2.18a and 2.18b indicate that predictive performance gains from spatial modeling might decrease with sparsity of gene expression. This trend is less apparent in the MERFISH dataset because all but 11 response genes had a sparsity of greater than 50%. However, the Xenium dataset used in this chapter has more low-sparsity response genes available to evaluate this pattern. These findings motivate extending DeepST to explicitly account for zero inflation in gene expression.

One way to handle modeling sparsity is with a two-head network architecture. This strategy separates the prediction problem into two sub-tasks: (i) estimating whether a gene is expressed at all and (ii) estimating the magnitude of the expression. A two-head approach for DeepST would involve one head predicting the response expressions (as DeepST already does), and another head predicting a binary indicator of whether the expression is zero or not.

Suppose we denote predictions from the regression head as $\hat{\mu}$ and predictions from the exact zero classification head as $\hat{p}_0$. If we assume each expression follows a zero-inflated



(a) MERFISH hypothalamus dataset. The Pearson correlation between the loss reduction and sparsity is -0.11 which is not a statistically significant p-value in the two-sided Pearson correlation test.



(b) Xenium fresh frozen mouse brain dataset. The Pearson correlation between the loss reduction and sparsity is -0.46 and yields a statistically significant p-value in the two-sided Pearson correlation test, providing statistical support for the inverse relationship between gene sparsity and benefit from spatial modeling.

Figure 2.18: Percentage improvement in test loss from spatial modeling versus gene sparsity.

model

$$y_{c,g} \sim \begin{cases} 0 & \text{with probability } p_0(X_{C,L},\, X_{C,R},\, M_C) \\ \text{some } f(X_{C,L},\, X_{C,R},\, M_C) > 0 & \text{with probability } 1 - p_0(X_{C,L},\, X_{C,R},\, M_C) \end{cases}$$

then we can train a model with a sparsity aware loss

$$\mathcal{L}(X_{C,L}, X_{C,R}, M_C, y_{c,g}) = \begin{cases} -\log \hat{p}_0(X_{C,L}, X_{C,R}, M_C) & y_{c,g} = 0 \\ -\log\left(1 - \hat{p}_0(X_{C,L}, X_{C,R}, M_C)\right) & \\ +\left(\hat{\mu}(X_{C,L}, X_{C,R}, M_C) - y_{c,g}\right)^2 & y_{c,g} > 0 \end{cases}.$$

We can generate sparsity-aware predictions by multiplying the outputs of both heads:

$$\hat{y}_{c,g} = \left(1 - \hat{p}_0\left(X_{C,L},\, X_{C,R},\, M_C\right)\right) * \hat{\mu}\left(X_{C,L},\, X_{C,R},\, M_C\right).$$

Because the $\hat{p}_0$ is unlikely to yield a prediction that $P(y_{c,g} = 0 \mid X_{C,L},\, X_{C,R}\, M_C) = 1$, we can introduce a hard threshold at inference time that maps expressions associated with confident $\hat{p}_0$ outputs to zero:

$$y_{c,g} \sim \begin{cases} 0 & \text{if } p_0(X_{C,L},\, X_{C,R},\, M_C) > \tau \\ \hat{\mu}\left(X_{C,L},\, X_{C,R},\, M_C\right) & \text{o.w.} \end{cases}.$$

This two-head formulation, among other plausible strategies, provides a principled starting point for DeepST to handle sparsity in ST datasets.

2.6.2 3D-Aware Cell-Cell Communication Models

Much of the currently available data in ST is not able to capture the full extent of three-dimensional tissue structure. For example, the MERFISH hypothalamus dataset collects tissue samples as cross sections taken along the bregma coordinates (z-axis), where each cross section is exactly 0.05mm (50μm) apart. This makes measuring communication events between cells with centroids (x, y, z) that satisfy $\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \leq r$ such that $z_i \neq z_j$ a limiting task. This limitation in available data means that a small subset of all available close proximity cell-cell relationships actually get considered, namely ones where $z_i = z_j$. The Xenium dataset we used did have depth-resolved z-coordinates, but this was limited to 30μm. Should future spatial transcriptomics data collection techniques be able to lower the distance between cross sections, DeepST would naturally apply to three-dimensional ST datasets with anticipated improvements in practice and minimal design

modifications. The results presented in the semi-synthetic experiment validate using as many neighboring cells as feasible for inference. We showed that as long as the true CCC channels are a subset of the input graph, DeepST maintains stable prediction error, suggesting it should generalize effectively to three-dimensional settings with more neighborhood signals.

2.6.3 Transfer Learning

The work of Su et. al has found success in utilizing transfer learning for learning spatial dependencies (Su et al., 2022). Their framework suggests that appropriate spatial context can be incorporated in the form of a spatial prior. This removes the need for convolution operations during training. However, this comes at the cost of assuming the spatial dependencies a priori. Yet, Smoother demonstrates robust performance suggesting that fine-tuning non-spatial methods using spatially aware objectives can achieve a similar performance improvement in response gene prediction.

Though DeepST demonstrates that leveraging spatial information directly has merit, our results also corroborate the conclusion presented by Smoother. In many cases, the baseline model of DeepST outperforms boosting and regression methods, indicating that the improvement in prediction can be attributable to both better model selection and increasing neighborhood knowledge (in our case through the use of r).

While this approach could remove causal predictors from the model, the approach significantly decreases training time and memory required to train the model. DeepST's training time increases by 20.65% to train on graphs with a neighborhood radius of 60μm as opposed to its non-spatial counterpart (graph with $r = 0\,\mu$m). Identifying and leveraging a family of spatial objectives to enhance existing model families reduces computational resources and provides a promising avenue for integrating spatial context effectively into machine learning frameworks.

2.6.4 Hyperparameter Tuning

Part of validating a predictive model such as DeepST or LightGBM is performing proper hyperparameter tuning. However, DeepST makes predictions in one shot while LightGBM requires a separate model for each response gene. Furthermore, to ensure we setup models to make statistically grounded claims about spatial dependence, it is critical that spatially ignorant and spatially aware models are trained with identical hyperparameters—even though the optimal configuration may differ between the two cases. Both LightGBM and DeepST were evaluated over a large set of hyperparameters. All considered hyperparameter values for both LightGBM and DeepST can be found in Appendix A.2.

2.6.5 Causal Setting

One limitation of our approach is that DeepST lacks the immediate interpretability of simpler classical methods. While we statistically formalize the identification of spatially dependent genes, our approach makes it more challenging to infer causal relationships. Ideally, we would like to determine which ligand-receptor pairs drive changes in response gene expression. This could be accomplished by using graph neural networks that employ attention mechanisms, with HoloNet serving as a prominent example (Li et al., 2023a). HoloNet identifies functional communication events (FCEs) by examining the learned attention weights in a graph attention network. HoloNet takes a multi-view network as input, consisting of multiple graphs that share the same nodes V, where each graph $G^P = (V, E^P)$ represents the cell-cell communication events mediated by a specific ligand-receptor pair P. Together, the edges across all views characterize the full set of cell-cell communication events. HoloNet returns ligand-receptor pairs that influence a particular gene's expression via attention mechanisms. However, for computational purposes, they filter out any gene that is not present in at least 30% of the cells, which may lead to overlooking certain spatial dependency conclusions. Additionally, HoloNet trains this procedure separately for each response gene, limiting opportunities for joint inference.

An extension of DeepST that leverages learned attention scores could unify predictive accuracy and biological interpretability, offering a path toward identifying causal relationships between ligand-receptor interactions and response gene expression.

2.7 Conclusion

In this work, we demonstrate the suitability of GCNs for the response gene prediction task. This effectiveness leads to improved statistical inference for spatial dependence detection. Relative to classical methods that primarily use summary statistics of neighboring gene expressions as new features in tabular dataset, we use the gene expressions as node attributes for a tissue graph.

Other works have leveraged graph neural networks for spatial transcriptomics prior to us. However, they either tailored the method for clustering in downstream tasks or fail to outperform classical models. Many contemporary approaches, including MESSI, evaluated their performance against XGBoost, yet LightGBM consistently outperforms several of these methods due to its leaf-wise splitting strategy, which allows for more efficient tree construction. Moreover, LightGBM has been shown to be both more scalable and more accurate than XGBoost, making it a stronger baseline for comparison.

Our results indicate that leveraging graph neural networks with mixture model convolutions can lead to better prediction of response genes on real data. In settings where cell type information is known a priori and used as a predictor, we find that the gain obtained from the additional flexibility of GCNs is not necessarily worth the additional computational overhead. However, in settings where the cell type information is not available, using a GCN for predictive modeling can improve predictions by directly capturing spatial dependencies without relying on inferred or pre-labeled cluster assignments.

In semi-synthetic cases where the underlying relationship is known, our method is able to perform optimally as long as the true edges are a subset of the tissue graph. This eliminates the need to tune the neighborhood construction of the cell or construct finite-dimensional summaries that explain the relevant incoming signals. We demonstrate that in complex biological systems where CCCs follow nonlinear, context-dependent signaling mechanisms, the graph neural network serves as a universal approximator and can recover response genes accurately, even when the input graph has superfluous edges.

A strong spatially aware model such as DeepST allows us to identify spatially dependent genes. For each gene, we attempt to predict its expression using spatially aware and spatially ignorant models, and measure the difference in predictive performance. By ranking genes based on these differences, we can identify the most promising candidates for future research.

The prioritization of spatially dependent genes has far-reaching applications across biomedical research. Our framework provides a way to systematically contrast the spatial behavior of genes between healthy and disease tissue, potentially uncovering spatial patterns that contribute to disease pathology. This insight can be particularly valuable for assisting drug discovery teams with possible therapeutic targets to prioritize. Gene rankings generated by our approach enable experimental biologists to prioritize follow-up studies more efficiently in terms of cost and time.

CHAPTER 3

Spatial Bayesian Clustering with Stochastic Variational Inference

3.1 Introduction

Recent advancements in spatially resolved transcriptomics have opened new opportunities to map tissue regions into biologically meaningful clusters. These clusters are crucial for applications such as biomarker discovery and sub-population detection. While spatial transcriptomics data may sometimes include reference labels, such labels often have limitations, and there is a growing need to improve methods in reference-free settings. To address this challenge, we introduce BayXenSmooth, a stochastic variational inference (SVI) method designed to learn posterior spot cluster distributions that are both spatially coherent and biologically interpretable. BayXenSmooth enhances clustering accuracy by incorporating spatial relationships through carefully designed empirical prior distributions, allowing it to balance the trade-off between smoothness and expression differences. Furthermore, the method is scalable and effective across data resolutions. Spot data often scales quadratically with finer resolution of two-dimensional tissue samples. BayXenSmooth's use of SVI makes it more computationally efficient than previous methods that rely on posterior sampling techniques, such as Markov Chain Monte Carlo (MCMC), which can be prohibitively expensive to retrain. Our results demonstrate that BayXenSmooth effectively groups tissues into smoother regions compared to previous methods while preserving expression heterogeneity consistent with earlier studies, offering a competitive alternative to existing approaches.

We demonstrate on both synthetic and real tissue data that BayXenSmooth manages to produce spatially contiguous clusters and correctly removes isolated cluster assignments. In the synthetic data case, we construct a dataset where each location is given a cluster assignment and expressions for each location are sampled as a weighted average of Gaussians conditioned on the neighbors' cluster assignments. We show that similar to previously

proposed Bayesian methods, we can reconstruct the data generating process with high accuracy. In real human tissue data, we demonstrate that the total distance between clusters is reduced in comparison to competing methods regardless of the resolution of the data while retaining a biologically meaningful structure.

3.2 Xenium Data

The Xenium platform by 10x Genomics is a high-resolution spatial transcriptomics technology that combines single-molecule RNA detection with spatial context, enabling precise localization of gene expression within tissue sections (Janesick et al., 2023b). Its novelty lies in the integration of spatial data with transcriptomics, facilitating spatial analysis and allowing for more detailed inference of cellular micro-environments and interactions than popular predecessors such as MERFISH, Slide-seq, and Visium (Spatial). Xenium data provides cellular-level resolution with a large selection of genes—attributes that previously involved a compromise. This data opens opportunities for advanced spatial modeling and clustering techniques to uncover biological insights.

The spatial organization of a tissue into regions plays a role in many applications. Identifying these spatial segments and patterns between them within a tissue can reveal insights about expression heterogeneity, a feature of interest in applications such as tumor detection and disease control (Marx, 2021). The Xenium platform itself claims that it relies on spatial information "to study tumor subtype heterogeneity and the micro-environment surrounding tumor epithelial cells" (10x Genomics, 2023b). The integration of spatial information with transcriptomics facilitates a detailed analysis of the tissue architecture, providing insights into how gene expression varies across distinct regions. This allows for improved resolution of the spatial relationships between cells and cell groups, which is critical for understanding complex biological processes. By leveraging this technology, researchers can explore how gene expression differences relate to structural features and functional diversity, further advancing the field of spatial biology. Spatially bounded clusters with quantifiable uncertainty have been shown to have applications in providing clear visual boundaries between different cell types or states, revealing functional regions within tissues, therapeutic targeting (Arora et al., 2023a; Lopez et al., 2022; Walker et al., 2022).

It is common for previous methods to have been benchmarked with Visium data where advanced resolutions were achieved by subdividing spots into smaller subspots and treating expressions at these subspots as latent variables. Recent advances in the Xenium platform allow for details at higher resolutions than Visium data alongside an increase in sample size and number of genes analyzed. Data collected on the Visium platform commonly has spot

sizes of size 55 μm x 55 μm and with 100 μm between them (10x Genomics, ndb,n). Datasets limited to spots of low resolution can occlude cells with pertinent expressions and spatial arrangements within a spot (Figure 3.1). Conversely, the Visium HD Spatial Gene Expression slides offer higher resolution spots of 2μm x 2μm with no gaps between them (10x Genomics, nda). This tool is helpful when studying local regions of tissue, but can be challenging to scale to large tissue samples. The ability to cluster tissue regions into communities and the optimal spot size to measure for this task remains an open challenge. Clustering exclusively with expression data loses spatial information, but an over-reliance on spatial information can obfuscate crucial information from expression data. Xenium provides extremely high-resolution gene expression information, but practitioners are still using software designed for an earlier generation of spatial transcriptomics platforms. Previously established methods may require Xenium data to be heavily downsampled to get reasonable results with clustering algorithms designed for earlier technologies, negating some of the benefit of the Xenium platform. The high resolution of Xenium data allows us to transform the data into spots of any size, providing an opportunity to stress test spot-level clustering methods. This flexibility supports a more precise exploration of spatial relationships within tissue regions.

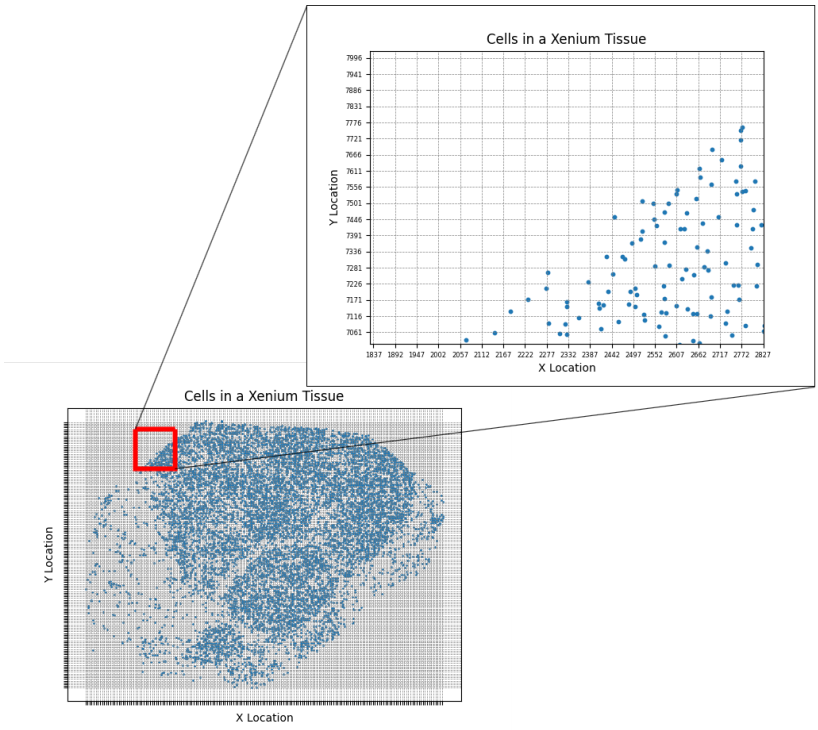
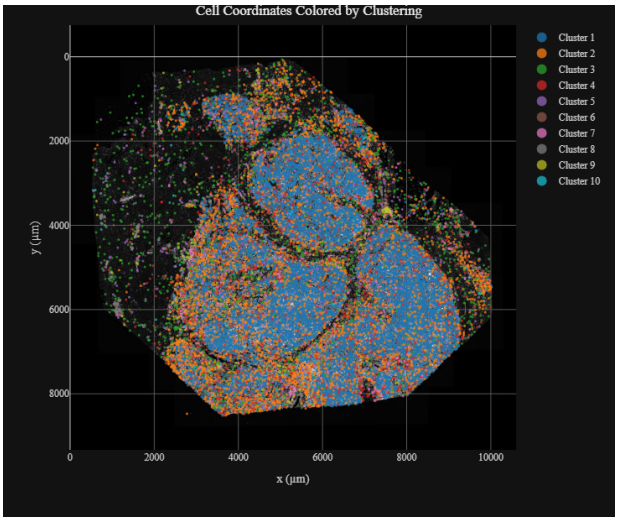
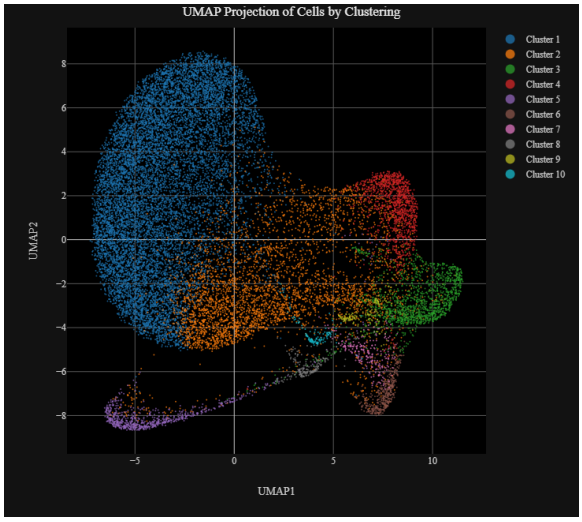


Figure 3.1: Organization of cell locations across a human breast tissue sample with 55 μm x 55 μm spots overlaid. Several of the spots have many cells even along the tissue border. Only 3% of all measured cells are displayed, meaning true densities are significantly higher.

Clustering regions of tissue offers an unsupervised approach to aggregate cells with similar profiles, but traditional methods can fail at evaluating the efficacy of these groupings. Moreover, integrating these techniques with established spatial and biological paradigms remains an ongoing challenge. Clustering methodologies that operate directly in ambient dimensions often focus on minimizing distances in a reduced dimensional space. The discordance of cluster assignments across the spatial and latent dimensions is visible in analysis output files from the Xenium Onboard Analysis Platform itself (Figure 3.2).



(a) K-means with K=10 output from the hBreast dataset.



(b) UMAP embeddings of cells and their associated cluster labels.

Figure 3.2: Cluster analysis results from the Xenium Onboard Analysis platform. (**Left**) Cluster labels mapped across tissue regions, highlighting spatially distinct areas. (**Right**) Clusters in a reduced dimensional space. These assignments are separable patterns, but not observable, limiting their practical spatial interpretation. (Image from Xenium Onboard Analysis Platform (Janesick et al., 2023a).)

However, in practice, the resulting clusters may not be useful for applications requiring clear separation of groups within the observable spatial dimensions. This challenge intensifies as the ambient dimension far exceeds the spatial dimensions, making it harder to produce spatially coherent groupings. The Xenium Onboard Analysis platform itself currently only touts K-means and graph-based clustering as its cluster analyses provided by default. This prompted a rush from the bioinformatics and computational biology community to provide additional clustering inference procedures that serve an applicable purpose. We discuss a variety of them in the next subsection.

3.3 Previous Work

While K-means and graph-based clustering methods can be useful for Xenium cluster analysis, their limitations have prompted the development of more rigorous approaches that more effectively harness the high-dimensional data generated by the Xenium platform. Platforms such as Seurat, scimap, and Voyager provide additional flexibility to existing graph-based clustering methods, allowing for a more customizable analysis experience (Hao et al., 2023; Moses et al., 2023; Nirmal and Sorger, 2024). Extending beyond the flexibility offered by these platforms, MERINGUE enhances graph-based clustering methods by making them spatially aware (Miller et al., 2021). It creates a neighborhood graph based on transcriptional similarities, with edges weighted by spatial proximity. This weighting emphasizes interactions between closely situated spots, effectively adapting a non-spatial method into one that considers spatial context. An alternative approach is to enhance the input data rather than modifying the clustering algorithm itself. stLearn leverages existing clustering techniques to produce spatially aware clusters by integrating three types of data—spatial location, tissue morphology, and gene expression—through its Spatial-Morphological-Expression (SME) framework (Pham et al., 2023).

While graph-based clustering methods are effective, they often rely on a predefined graphical structure that may not be statistically justified. Giotto offers a way to perform spatial clustering by leveraging Hidden Markov Random Fields (HMRFs) (Dries et al., 2021). HMRFs allow the model to incorporate spatial dependencies by using a hidden layer that regularizes clusters based on the spatial relationships between spots, enhancing spatial coherence in the identified clusters.

By combining graph-based methods with deep learning, two dominant approaches have emerged in spatial transcriptomics analysis: SpaGCN and STAGATE. SpaGCN is a graph convolutional network that learns latent variables for iterative clustering, initialized using the Louvain algorithm. The network's loss function is designed to emphasize high-confidence spot assignments based on this initialization. While effective, the model may be suboptimal in regions with high uncertainty, where robust inference and uncertainty quantification are arguably most critical. STAGATE introduces a graph attention network tailored for spatial transcriptomics, dynamically weighting neighboring spots based on spatial and transcriptional similarity, which enhances clustering accuracy in complex tissue regions (Dong and Zhang, 2022). However, a potential shortcoming of STAGATE is its complexity; the use of graph attention mechanisms can be computationally expensive, making it less scalable for very large datasets.

Bayesian approaches represent another important class of methods, with BayesSpace

standing out as a prominent framework in this area. BayesSpace has demonstrated improved clustering performance over many of the aforementioned works, making it a strong benchmark for spatial domain clustering. The framework constructs clusters by applying PCA to expression data and promoting spatial coherence through a Potts prior, which encourages adjacent spots to cluster together (Zhao et al., 2021). Through posterior sampling, BayesSpace generates both spot-level cluster assignments and latent subspot clusters, establishing itself as a strong benchmark framework in Bayesian spatial domain clustering. Furthermore, the strength of the spatial clustering similarity can be customized via a smoothing parameter. The primary challenge for BayesSpace is the need to incorporate a Markov random field in the form of a Potts prior:

$$\pi\left(z_i\right)=\exp\left(\frac{\gamma}{|\mathcal{N}(i,j)|}\times 2\sum_{\mathcal{N}(i,j)} I\left(z_i=z_j\right)\right)$$

into the posterior sampling procedure, which can be computationally intensive as the number of spots considered increases. Generating many posterior samples requires repeated message passing of the underlying graph across for every target spot. Furthermore, separate inference procedures must be conducted to evaluate multiple potential graphs. Therefore, the computational slowdown of an exact Bayesian procedure is amplified by the number of graphs we want to try out. This computational burden motivates our approximate Bayesian approach.

By developing an approximate approach that enforces spatial coherence between proximal spots, we demonstrate that BayXenSmooth generalizes appropriately as spot size decreases down to the sub-cellular level. This adaptability is critical given the continuous advancements in tissue profiling technology, enabling increasingly high-resolution data capture. The assignments output by BayXenSmooth are spatially contiguous while still preserving biological interpretations and probabilistic distributions over cluster memberships. The computation time and posterior sampling procedure rivals strong Bayesian competitors while matching performance in marker gene identification under spatial coherence constraints. Our proposed method lies at the unique intersection of being spatially aware, Bayesian, supportive of soft assignments, and computationally scalable with increasing resolutions.

3.4 Method

We train a forward model to learn approximate posteriors via SVI (Hoffman et al., 2013). The principal approximate posterior we attempt to learn is a soft cluster assignment for each spot. Typically, hard assignments are used in clustering due to their simplified inter-

pretability. However, hard assignments enforce discrete boundaries that might not always be applicable.For instance, when ST data is of coarser resolution, many cell types or spatial domains could coexist in the same spot. Additionally, transitions between regions is usually gradual rather than abrupt, motivating some uncertainty in where the boundary point actually occurs. Soft assignment affords us uncertainty quantification for these scenarios.

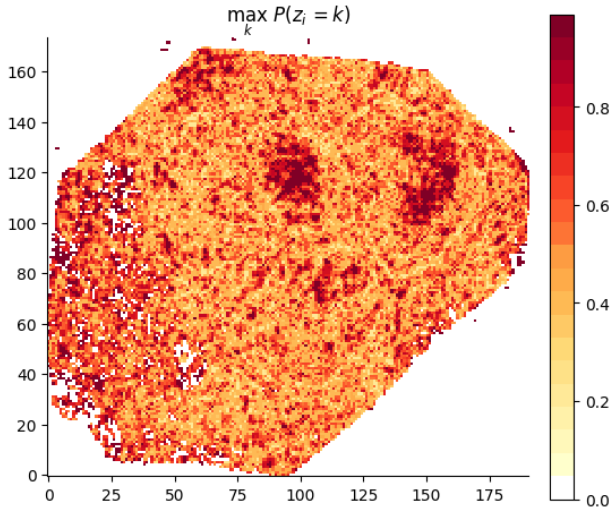
We assume that observed data is modeled as a mixture of Gaussians, with each component representing an underlying category that contributes to the observed gene expressions (e.g., cell types, experimental conditions, disease patterns, etc.). To ensure computational feasibility, dimensionality reduction is performed on measured gene transcripts. Prior knowledge regarding the spatial relationships between tissue spots serve as the construction of the empirical prior component weights for each spot. Each cluster has an independent and identically distributed (i.i.d.) prior distribution for the means and scales, which applies uniformly across all spots. This results in a global approximate posterior for each cluster's mean and scale applied to all spots. However, in contrast to a traditional Gaussian mixture model, which models the likelihood of the data using shared component weights, we assume that each spot has its own distinct weighting profile. This individualized weighting allows us to express uncertainty in cluster assignments and capture local heterogeneity in tissue composition. Prior knowledge regarding the spatial relationships between tissue spots serve as the construction of the empirical prior component weights for each spot. Learning the posterior component weights for individual spots can lead to uncertainty quantification of class assignment at any resolution (see Figure 3.3). Modeling with stronger, more informative prior distributions implies a higher degree of spatial influence in the assignments, whereas weaker priors allow for neighboring regions of tissue that were initially clustered together to break apart with smaller differentials in the expression data. This effectively introduces an interplay between ensuring spatial coherence and preserving biological significance in cluster assignments.

3.4.1 Gene Expression Data Pre-Processing

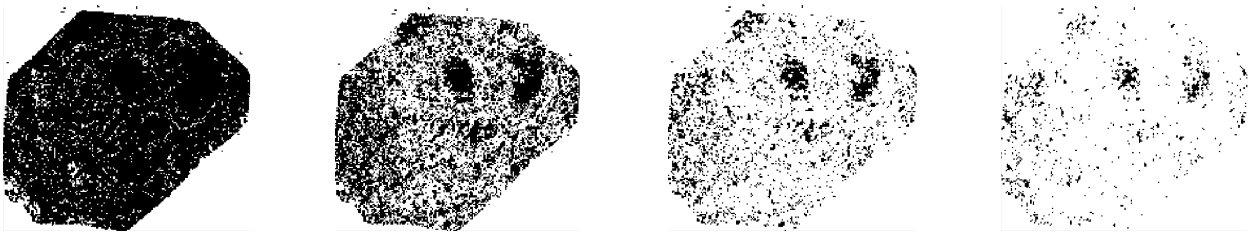
To prepare the data for modeling, we optionally remove all transcripts with low confidence. The quality value (q) can be converted into a probability of error ($E = 1$) via the following formula:

$$P(E = 1) = 10^{-\frac{q}{10}}. \tag{3.1}$$

The transcripts removed would have a probability of error above some threshold as defined in equation 3.1: $P(E = 1) > \delta$. Out of precaution we also remove all transcripts with missing



Top) Spot Cluster Confidence over Tissue



(a) $\max_k P(z_i = k) > 0.25$

(b) $\max_k P(z_i = k) > 0.5$

(c) $\max_k P(z_i = k) > 0.75$

(d) $\max_k P(z_i = k) > 0.9$

Figure 3.3: Posterior cluster uncertainty visualization. Posterior component weights provide uncertainty quantification of the hard cluster assignment. Moving from 3.3a to 3.3d, the plots display progressively higher confidence regions within the tissue. Results are taken from the KRT6B marker gene run presented in Figure 3.12.

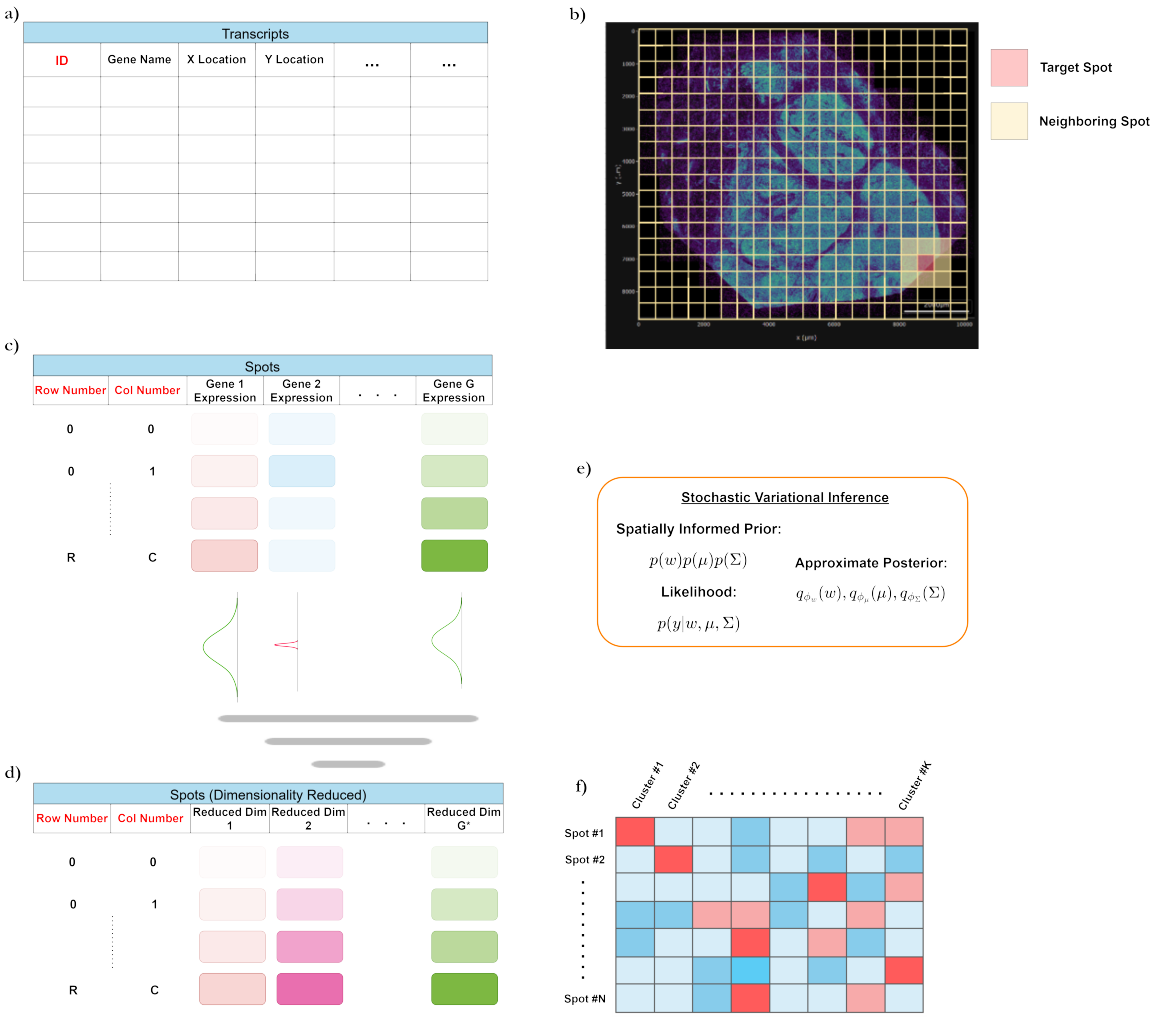


Figure 3.4: Schematic of BayXenSmooth: **A**) Collection of transcript IDs alongside gene names and spatial locations. **B**) Organization of transcript information into predetermined spatial bins, highlighting each target spot (red) and its neighboring spots (yellow). **C**) Compilation of gene expression data for each spot. **D**) Application of dimensionality reduction on the compiled spot data. **E**) Implementation of stochastic variational inference, utilizing a spatially informed prior and a likelihood model derived from the spot data. The prior distribution utilizes neighboring information (as defined in **B**) for initial estimates. **F**) Iterative updates in the variational inference process are visualized through the posterior weights, which significantly influence the model's output.

cell IDs. This is because aggregating transcripts into spots or cells Additionally, we remove any reads associated with blank or negative controls. These reads assist with generating accurate reads for gene expressions but are not gene expressions themselves. Optionally, a subsample of highly variable genes are filtered from the Next, the tissue is divided into tiles of a pre-defined spot size. These tiles can be two- or three-dimensional. The smaller the spot size, the closer each spot represents a cell or sub-cellular resolution snapshot. Larger spot sizes represent several cells that act as a collective in a region of the tissue. Each spot in the spot grid is assigned a row, column, and height index (three-dimensional case only). Going forward, we will focus on the two-dimensional case without loss of generality. Per the industry standard, all transcriptional counts are log1p ($\log(1 + X)$) normalized to reduce variance and transform multiplicative differences in expression to additive differences (Booeshaghi and Pachter, 2021).

The observed data used for modeling ($y_{i}{}_{i=1}^{N}$) can either be principal components, a subset of highly variable genes, or all available expression data. In the event we choose to model principal components or highly variable genes, we must perform dimension reduction. For principal components modeling, a predefined number of components are chosen and PCA is calculated for all gene expression values. For highly variable gene modeling, we identify a proportion of highly variable genes utilizing the dispersion-based method introduced by Seurat (Satija et al., 2015b). For every gene, the dispersion is calculated as shown in eq. 3.2. Then, these dispersion metrics are standard normalized across all genes (eq. 3.3):

$$D_g = \frac{\sum_{i=1}^{N} \left(X_{i,g} - \frac{1}{N} \sum_{i=1}^{N} X_{i,g} \right)^2}{\frac{1}{N} \sum_{i=1}^{N} X_{i,g}}, \; g = 1, 2, ..., G \tag{3.2}$$

$$D_g^* = \frac{D_g - \frac{1}{G} \sum_{g=1}^{G} D_g}{\sum_{i=1}^{N} \left(D_g - \frac{1}{G} \sum_{g=1}^{G} D_g \right)^2}, \; g = 1, 2, ..., G. \tag{3.3}$$

We set as a hyperparameter a desired proportion α of genes that we want to keep. A gene is considered highly variable if its normalized dispersion has a high-enough z-score: $D_g^* \geq \Phi^{-1}(\alpha)$. All other genes are dropped from the data.

While BayXenSmooth can still model the expression data without dimensionality reduction, performing this reduction avoids the curse of dimensionality allowing for better comparison against competing methods, faster training, more efficient sampling, and stronger convergence (Jia et al., 2022).

3.4.2 Empirical Prior Construction

Established prior Bayesian methods typically encode a network of relationships probabilistically is via a Markov Random Field (MRF). An MRF models spatial dependencies through a probabilistic graphical structure where each node is conditionally dependent only on its neighbors. MRFs require the construction of an adjacency matrix or neighborhood structure to define these dependencies explicitly. The spatial relationships are thus encoded in the graph structure, which defines how the values at neighboring nodes influence each other, often based on local smoothness or consistency constraints.

In contrast, our approach bypasses the need for an adjacency matrix during training or sampling by using an empirical, one-shot calculation based on an initial cluster labeling and directly embedding spatial dependency into the prior for the weights. This construction allows for efficient encoding of spatial relationships without the computational burden of storing or updating a large adjacency matrix, thereby streamlining inference while preserving spatial coherence.

To obtain cluster assignments that are spatially organized, we assume a set of spatial relationships between spots to be represented by our prior distributions. This requires a formal definition of proximity. Provided a neighborhood size r, for all pairs of spots i and j with row, column labels $p_i := x_i, y_i$ and $p_j := x_j, y_j$, respectively, an adjacency matrix can be constructed by following the rule:

$$A(i,j) = 1 - \mathbb{I}(||p_i - p_j||_\infty > r). \tag{3.4}$$

Prior distributions are created for the weights, means, and scales of the data generating process. The means and scales of each component have i.i.d. Normal and Standard Log-Normal prior distributions over the dimensions of the data. The cluster distributions themselves are also independent of each other. This method can be transformed to an empirical Bayes approach by using the empirical means and scales of the dimensionality-reduced expression data at each cluster to center the prior distributions.

$$\begin{gathered}
\mu_k \in \mathbb{R}^D \sim N(\mathbf{0}, \boldsymbol{I}) \quad \forall k = 1, 2, \dots K \\
\boldsymbol{\mu} := (\mu_1, \cdots, \mu_K) \\
\log(\sigma_k) \in \mathbb{R}^D \sim N(\mathbf{0}, \boldsymbol{I}) \quad \forall k = 1, 2, \dots K \\
\boldsymbol{\sigma} := (\sigma_1, \cdots, \sigma_K).
\end{gathered}$$

The prior distribution for the logits is where the spatial information is encoded. We

identify an empirical mean for the logits using the following procedure.

Let X be a dataset, INIT-METHOD(X, K) return the initial cluster indices of X for K clusters without spatial context, and ONEHOT(k) return a one-hot encoding of dimension K with the 1 assigned to the k^{th} index. Let each individual observation be denoted as y_i. Define $z_{i,k}$ for $k = 1, \ldots, K$ such that $z_{i,k} = 1$ if INIT-METHOD$(X, K) = k$ and $z_{i,k} = 0$ otherwise. Then our prior for component weights for each spot i can be written out as:

$$
\begin{aligned}
z_i = (z_{i,1}, z_{i,2}, \ldots, z_{i,K}) &= \text{ONEHOT}(\text{INIT-METHOD}(X, K))_i \in \Delta^{K-1}, \\
\Delta^{K-1} &= \left\{ z_i \in \mathbb{R}^K : \sum_{k=1}^{K} z_{i,k} = 1 \text{ and } z_{i,k} \in \{0, 1\} \ \forall k \right\}, \\
w_{i,k} &= \max\left(\left(\sum_{j:A(i,j)=1} \frac{z_j}{|\mathcal{N}(i)|} \right)_k, \epsilon \right), \quad \epsilon = 0.001 \text{ (hyperparameter)} \\
l_i &\sim N(\log(w_i), \lambda \boldsymbol{I}) \quad \lambda = 1.0, \text{ (hyperparameter)}.
\end{aligned}
$$

This approach provides an empirical, one-shot method for embedding spatial information directly into the prior distribution, enabling approximate posterior sampling without reliance on a stored adjacency matrix. This empirical prior construction on the weights allows us to capture spatial dependencies without requiring a Markov random field or propagation network, as used in methods like BayesSpace. By integrating spatial structure exclusively at the prior level, we create an informed, approximate Bayesian model with a spatial prior.

The motivation for this approach stems from the ability of existing clustering methods to effectively group ST regions based on gene expression data. Many clustering methods successfully partition expression profiles into biologically meaningful groups and exhibit—though perhaps not perfectly—a spatial structure between groups. Therefore, we try to enhance the spatial organization by constructing an empirical prior that uses neighboring information. Figure 3.5 exemplifies this scenario. By designing the prior this way, we ensure that its mode aligns with spatially coherent cluster assignments. Posterior deviations from the prior reflect an implicit trade-off between spatially proximal cluster assignments and an expression-based structure. Effectively, each target spot's prior weight distribution is peaked at the most frequently occurring neighboring cluster assignments. Performing (approximate) Bayesian inference with this prior ensures that we can correct instances where our empirical spatial assumption does not align with the underlying biological structure. An example of where these assumptions could be erroneous is for ST data with low resolution. Each spot could represent a large region of tissue with several communities within that differ from

similarly coarse neighboring spots.

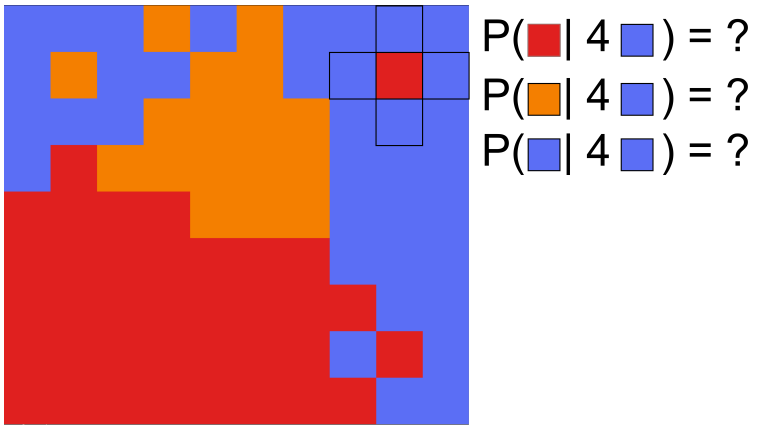


Figure 3.5: Illustration of empirical spatial prior motivation. The target spot labeled red at the center of the cross has neighboring spots belonging to a shared, different cluster, suggesting that the target spot may be misclassified in a spatial context. This may indicate that the target cell belongs to a different cluster than its original label. This motivates the construction of an empirical prior that integrates neighborhood information to refine cluster assignments.

To facilitate gradient-based updates, it is preferable to parameterize models in a way that avoids explicit constraints, such as those imposed by the simplex on the component weights. Therefore, we model the logits with a multivariate normal distribution, which yields a logistic-normal distribution on the soft assignments. An alternative approach is to parameterize the weights w with a Dirichlet distribution. While Dirichlet distributions can be used for inference, in practice, the sampled simplexes tend to be too noisy to differentiate posterior clusters or require such high concentration that posterior collapse occurs. This motivates approximating the Dirichlet with a multivariate logistic normal. This approach allows for lower variance gradient estimates via path-wise derivatives using the reparameterization trick (Xu et al., 2018b). Applying a softmax transformation to the samples from this approximation provides valid soft cluster assignments. The expectation and variances of these softmax outputs minimize the Kullback-Liebler (KL) divergence between the Dirichlet and logistic-normal distributions (Aitchison and Shen, 1980).

3.4.3 Likelihood Model

We assume that the data in each spot $y_i \in \mathbb{R}^d$ comes from an isotropic Gaussian mixture model with K components, where $\Sigma_k = \text{diag}(\sigma_k^2)$ and $\sigma_k \in \mathbb{R}^d$ is a vector of standard deviations for each data dimension, and the number of mixture components K is a hyperparameter:

$$(y_i|\boldsymbol{\mu}, \boldsymbol{\sigma}, w_i) \sim \sum_{k=1}^{K} w_{i,k} N(\mu_k, \Sigma_k). \tag{3.5}$$

The model is composed of cluster distributions and cluster weights. The weights act as a soft cluster assignment for each spot. We highlight that μ_k, Σ_k are treated as global parameters whereas the cluster weight $w_i := \text{softmax}(l_i)$ are treated as local parameters. The number of mixture components K is a hyperparameter. A hard assignment to a specific cluster in any spot can be accomplished by selecting the component with the highest weight:

$$z_i = \arg\max_k w_{i,k}.$$

This model can be applied such that y_i is a spot's gene expression profiles, principal components, or highly variable genes. In practice, we found that the model achieved optimal performance when modeling between 3 and 25 principal components of the hBreast data.

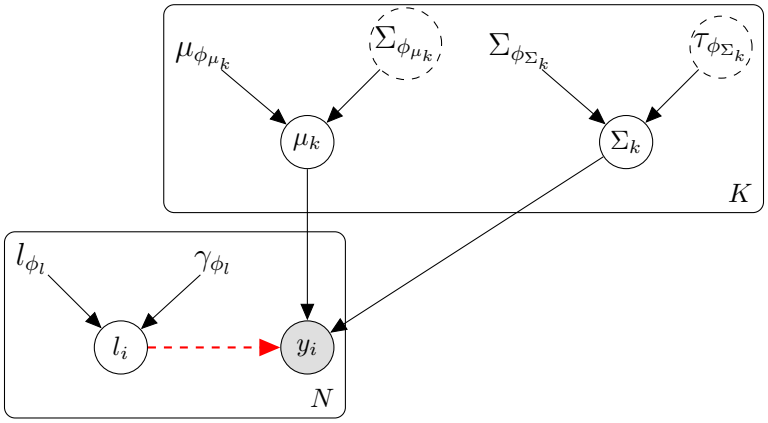


Figure 3.6: BayXenSmooth Variational Sampling Graph for ELBO Computation: Gray nodes represent observable features, white nodes represent latent parameters (both global and local), while learned variational parameters are represented without bounded circles. The dashed circles represent variational parameters that are learned but can be dropped to learn MAP estimates for global variational distributions. Edges from l_i to y_i are shown in red, dashed edges to emphasize that l_i is first mapped to cluster weights via a deterministic softmax transformation before influencing the likelihood of y_i.

3.4.4 Approximate Posterior Inference

To perform clustering while accounting for spatial dependencies, we aim to infer the posterior distributions of the latent variables in our model. The generative model assumes that the observed data $y_{i}{}_{i=1}^{N}$ are generated from a mixture of Gaussians, where each component

corresponds to a latent cluster with parameters (μ_k, Σ_k), and the mixture weights w_i are specific to each data point (i.e., spot) i. If we allow θ to be a catchall for all model parameters and $\phi := \{\phi_{l_i}, \phi_{\mu_k}, \phi_{\sigma_k}\}_{i=1..N,k=1..K}$ as the catchall for all variational parameters, we can formalize the entire posterior inference procedure.

The generative process involves drawing global cluster means and scales $\mu_k \sim p_\theta(\mu_k), \sigma_k \sim p_\theta(\sigma_k)$ and local mixture logits for each spot $l_i \sim p_\theta(l_i \mid f(y_i))$ where f is the initial cluster assignment function as outlined in the Prior section. Because all of these draws are independent, the product of these priors can be represented as a singular joint distribution $p_\theta(l, \mu, \sigma)$. The observed data are generated according to the likelihood:

$$p_\theta(y_i \mid l_i, \boldsymbol{\mu}, \boldsymbol{\sigma}) = \sum_{k=1}^{K} \text{softmax}(l_i)_k N(y_i \mid \mu_k, \Sigma_k) = \sum_{k=1}^{K} w_{i,k} N(y_i \mid \mu_k, \Sigma_k).$$

To approximate the intractable posterior $p_\theta(l_i, \boldsymbol{\mu}, \boldsymbol{\sigma} \mid y)$, we employ SVI with variational distributions $q_\phi(l_i, \boldsymbol{\mu}, \boldsymbol{\sigma}) = q_\phi(l_i) q_\phi(\boldsymbol{\mu}) q_\phi(\boldsymbol{\sigma})$, assuming a mean-field factorization for computational tractability. The exact structure of variational approximation and sampling path used for ELBO computation (excluding adjustments made for the reparameterization trick) is depicted in Figure 3.6. We choose variational distributions that match the family of the prior distributions for computational simplicity and to emulate conjugacy. For each cluster $k = 1, \ldots, K$, means and scales are sampled via the following procedure:

$$q_{\phi_{\mu_k}}(\mu_k) = N\left(\mu_k \mid \mu_{\phi_{\mu_k}}, \Sigma_{\phi_{\mu_k}}\right)$$
$$q_{\phi_{\sigma_k}}(\sigma_k) = \text{LogNormal}\left(\sigma_k \mid \sigma_{\phi_{\sigma_k}}, \tau_{\phi_{\sigma_k}}\right).$$

For each spot, the approximate posterior mixture logits are Gaussian. Consequently, applying the softmax transformation maps w_i to the simplex, resulting in a Logistic-Normal distribution:

$$q_{\phi_{l_i}}(l_i) = N\left(l_i \mid l_{\phi_{l_i}}, \gamma_{\phi_{l_i}}\right) \rightarrow q_{\phi_{l_i}}(w_i) = \text{LogisticNormal}\left(w_i \mid l_{\phi_{l_i}}, \gamma_{\phi_{l_i}}\right).$$

Retrieving the component weights is accomplished by taking the softmax of the learned mixture logits. Note that while we frequently represent each distribution with a covariance matrix, they are all constrained to be diagonal. BayXenSmooth attempts to maximize the ELBO, the explicit form of which is

$$\text{ELBO} := \mathbb{E}_{q_\phi(l_i, \boldsymbol{\mu}, \boldsymbol{\sigma})}\left[\log p_\theta(y_i | l_i, \boldsymbol{\mu}, \boldsymbol{\sigma}) + \log p_\theta(l_i, \boldsymbol{\mu}, \boldsymbol{\sigma}) - \log q_\phi(l_i, \boldsymbol{\mu}, \boldsymbol{\sigma})\right] \tag{3.6}$$

with a derivation and training algorithm provided in Appendix B.3.

Note that as the variational parameters for each cluster k—specifically, the covariance $\Sigma_{\phi_{\mu_k}}$ and the precision $\tau_{\phi_{\sigma_k}}$—approach zero, the posterior distributions learned for μ_k and σ_k converge in distribution to delta distributions centered at their respective MAP estimates: $\mu_k \xrightarrow{d} \delta\left(\mu_k - \mu_{\phi_{\mu_k}}\right)$ and $\sigma_k \xrightarrow{d} \delta\left(\sigma_k - \sigma_{\phi_{\sigma_k}}\right)$. Therefore, we treat the variational parameters $\mu_{\phi_{\mu_k}}$ and $\sigma_{\phi_{\sigma_k}}$ as fixed point estimates, effectively achieving MAP estimation by dropping the entropy term $\mathbb{E}_{q_\phi(l_i,\boldsymbol{\mu},\boldsymbol{\sigma})}\left[\log q_\phi\left(l_i, \boldsymbol{\mu}, \boldsymbol{\sigma}\right)\right]$ in the Evidence Lower Bound (ELBO). In this setup, the ELBO reduces to maximizing the log joint likelihood, aligning with MAP objectives. However, BayXenSmooth retains the flexibility to learn the scale parameters adaptively, allowing them to capture underlying data heterogeneity rather than forcing deterministic estimates.

We optimize the variational parameters using the Adam optimizer with an initial learning rate of 0.001 and hyperparameters $(\beta_1, \beta_2) = (0.9, 0.999)$ (Kingma and Ba, 2017). Gradient updates are performed until convergence of the ELBO, employing an early stopping mechanism that halts training when the ELBO shows no significant improvement over a predetermined number of iterations. Every distribution is constructed to be compatible with the reparameterization trick, enabling efficient gradient-based optimization (Kingma and Welling, 2014). The viability of the reparameterization trick is demonstrated in Appendix B.4.

3.5 Synthetic Experiment

To establish a baseline performance for our method, we generate data that adheres to the rook neighborhood structure, which also forms the basis of our empirical prior construction. The synthetic data assumes that the number of clusters is known a priori ($K = 5$). The ground truth is generated by placing 4 random circular masses of varying sizes on a grid and allowing any unmarked space to represent its own cluster. Then, we perform Gaussian smoothing over the grid to simulate spatial continuity and smoother boundaries. However, due to the heavy overlap of cluster masses, defining a single, definitive clustering becomes challenging. This overlap introduces regions where multiple clusters may exert influence on a single spot, resulting in blurred boundaries between clusters. Consequently, the ground truth is inherently fuzzy in these areas. This setup provides a realistic test scenario where methods need to handle uncertainty and spatial overlap effectively. Simulated expression data are generated by assigning a Gaussian distribution to each cluster class, with each spot's expression modeled as a mixture of Gaussian samples drawn from its own and neighboring components, weighted by the proportion of neighboring spots (including the target) that belong to each component.

3.5.1 Data Generating Process

The synthetic data we generated was for a sample grid where the number of rows and columns were both set to 50. We simulated 5 variables designed to represent principal components by assigning spatial clusters with randomized centers, radii, and Gaussian-smoothing to mimic realistic, gradual tissue heterogeneity. Given these hard labels, expression data was simulated via a Gaussian mixture model with mixture weights proportional to the frequency of cluster labels in the local neighborhood and the component samples drawn from cluster-specific means and diagonal covariance matrices. The exact procedure is presented in Algorithm 1.

3.5.2 Clustering Performance Evaluation

Having established how BayXenSmooth performs posterior inference and constructed a data generating process with known ground truth soft assignments, we evaluate its clustering performance alongside relevant competing models. Our synthetic dataset is a controlled benchmark that has known soft assignments, from which hard assignments can be obtained by applying an argmax over the soft cluster probabilities. Furthermore, the data generating process leverages a neighborhood structure that matches the spatial dependencies used to construct the empirical priors. Demonstrating that our method accurately recovers cluster labels in this setting is critical for validating its viability.

We evaluate clustering performance of all considered methods with the adjusted Rand index (ARI) (Rand, 1971). The Rand index is an established metric used to assess the similarity between two cluster assignments. The Rand index evaluates the proportion of pairs of elements that share the same cluster assignment across both partitions being compared. However, because the Rand index does not adjust for agreements that occur at random, it is difficult to determine whether a given score reflects the model's merit or is the result of chance. To address this, the adjusted Rand index is a corrected-for-chance version of the Rand index, accounting for the fact that random clustering assignments can yield nonzero similarity scores in expectation. When one partition is the ground truth and the other is a model's clustering output then the ARI compares how a learned cluster assignment does relative to the ground truth while adjusting for random agreements. For our purposes, we know a prior that both partitions have the same number of clusters (K).

Assuming that we have 2 datasets X and Y both partitioned into K groups, then the ARI has a formula derived from the contingency table below:

Algorithm 1 Simulation of Spatial Clusters and Expression Data

Require: grid size $N = 50$, number of clusters $K = 5$, data dimension $D = 5$, error noise parameter $\sigma_n = 0.5$, smoothing parameter $\sigma_s = 3$, neighborhood radius $r = 1$

Ensure: Expression data $\mathbf{y}_{i,j}$ for each cell (i, j)

Initialize Cluster Assignments: $C_{i,j} \leftarrow 0$ for all (i, j)

Assign Spatial Clusters:

for $k \leftarrow 1$ to K **do**

  Randomly select center: $(c_x^{(k)}, c_y^{(k)}) \sim \text{UniformDiscrete}(1, N)$

  Randomly select radius: $R^{(k)} \sim \text{UniformDiscrete}(10, 30)$

  Assign cells to cluster k:

  for all (i, j) such that $\left(i - c_x^{(k)}\right)^2 + \left(j - c_y^{(k)}\right)^2 < \left(R^{(k)}\right)^2$ **do**

    $C_{i,j} \leftarrow k$

  end for

end for

Add Noise: $M_{i,j} \leftarrow C_{i,j} + \epsilon_{i,j}, \quad \epsilon_{i,j} \sim N(0, \sigma_n^2)$

Gaussian Smoothing:

Define kernel: $G_{a,b} = \frac{1}{2\pi\sigma_s^2} \exp\left(-\frac{a^2+b^2}{2\sigma_s^2}\right)$

Smoothed label: $S_{i,j} \leftarrow \text{clip}\left(\text{round}\left(\sum_{(u,v)\in\mathcal{N}(i,j)} G_{i-u,j-v} M_{u,v}\right), 0, K\right)$

Construct Prior Weights:

$\mathcal{N}(i, j) = \{(u, v) \mid |i-u| + |j-v| \leq r\}$

$w_{i,j,k} \leftarrow \frac{1}{|\mathcal{N}(i,j)|} \sum_{(u,v)\in\mathcal{N}(i,j)} \delta(S_{u,v} = k)$

Sample Cluster Parameters:

for $k \leftarrow 1$ to K **do**

  $\mu_d^{(k)} \sim \text{Uniform}(-5, 5), \quad \sigma_d^{(k)} \sim \text{Uniform}(0, 2.5)$

  $\Sigma^{(k)} = \text{diag}((\sigma_1^{(k)})^2, \ldots, (\sigma_D^{(k)})^2)$

end for

Simulate Expression Data:

for all (i, j) **do**

  for $k \leftarrow 1$ to K **do**

    $x_{i,j,k} \sim N(\mu^{(k)}, \Sigma^{(k)})$

  end for

  $\mathbf{y}_{i,j} \leftarrow \sum_{k=1}^{K} w_{i,j,k} x_{i,j,k}$

end for

$X \backslash Y$	Y_1	Y_2	$\cdots$	Y_K	row sums
X_1	n_{11}	n_{12}	$\cdots$	n_{1K}	a_1
X_2	n_{21}	n_{22}	$\cdots$	n_{2K}	a_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
X_K	n_{K1}	n_{K2}	$\cdots$	n_{KK}	a_K
col sums	b_1	b_2	$\cdots$	b_K	

.

If we let n_{ij} be the number of points belonging to cluster i in partition a and cluster j in partition b, a_i the number of points in cluster i and b_j the number of points in cluster j, then

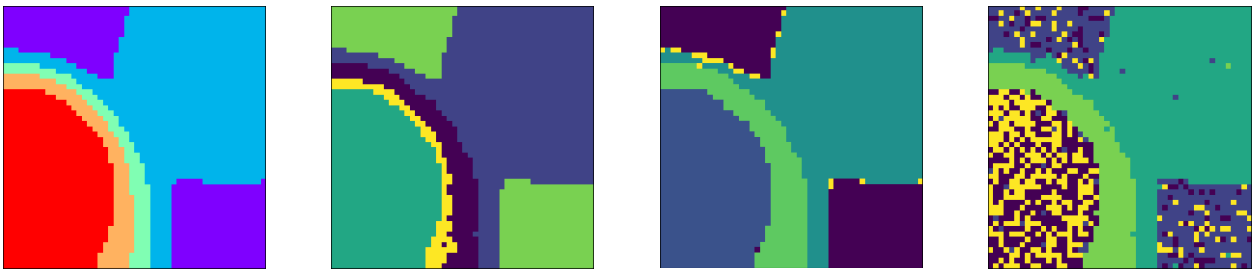
$$\text{ARI} = \frac{\sum_{ij} \binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] / \binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] / \binom{n}{2}}.$$

When the clusters perfectly agree, then collecting pairs from the contingency table is the same as taking pairs from either cluster partition: $\sum_{ij} \binom{n_{ij}}{2} = \sum_i \binom{a_i}{2} = \sum_j \binom{b_j}{2}$ and ARI $= 1$. If one partition represents the ground truth and the other represents a computed clustering solution, then the closer the computed ARI value is to 1, the more accurate the inferred partitioning is relative to the ground truth. ARI values near 0 indicate random labeling while negative ones correspond with rare cases of extreme label mismatching between clusters.

When evaluating several clustering method, we expect BayesSpace and BayXenSmooth to perform well on this problem given that the neighborhood construction for both methods matches the one used in the synthetic data generating process. Figure 3.7 shows the clusters learned by various cluster regimes. Several methods had hyperparameters that could be tuned; for these models, the ARI and cluster assignments presented for each method were the top performing configurations within each model family. Predictably, BayXenSmooth and BayesSpace were the top performers; both were initialized with mclust and generally recover the structure of the data with minimal boundary overlap. BayesSpace initializes the hard cluster assignments with mclust whereas BayXenSmooth constructs an empirical prior for the cluster means, scales, and logits of component weights. These results validate the constructed spatial priors, which effectively guide the models toward recovering cluster structures consistent with the ground truth.

The performance of competing methods highlights the effectiveness of crafting empirical priors around certain clustering approaches. For this synthetic dataset, mclust (Scrucca et al., 2023) and Hierarchical clustering (Johnson, 1967) seem to generally recover the general

spatial patterns on their own. Exceptions to the rule could be corrected with posterior inference as demonstrated by BayXenSmooth. However, if the initial cluster assignments used to construct the empirical prior is highly misaligned, lacks spatial coherence, or is excessively noisy (as observed with Leiden (Traag et al., 2019) or Louvain (Blondel et al., 2008)), then building sharp empirical priors around them can constrain posterior inference.

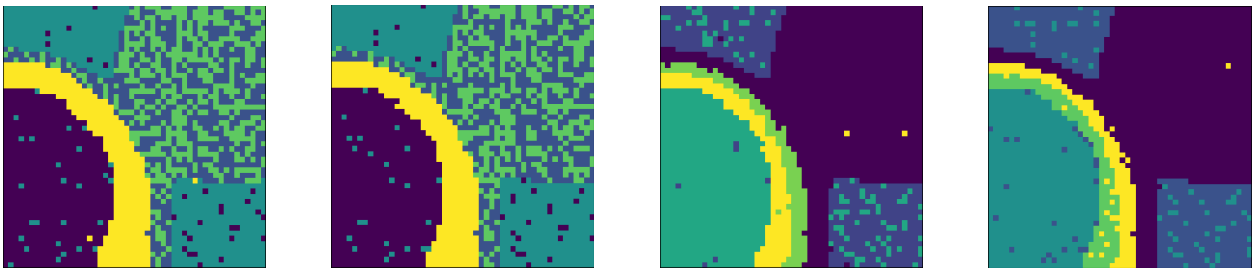


(a) Ground truth assignments

(b) BXS (Ours) (ARI=0.980)

(c) BayesSpace (ARI=0.957)

(d) K-means (ARI=0.781)



(e) Leiden (ARI=0.686)

(f) Louvain (ARI=0.691)

(g) Hierarchical (ARI=0.897)

(h) mclust (ARI=0.917)

Figure 3.7: Hard cluster assignments from various methods on synthetic data. The ground truth is shown in (a), with each subplot displaying cluster assignments and adjusted Rand index (ARI) relative to the ground truth. For Leiden and Louvain, the resolution parameter is $\lambda = 0.35$ and for BayesSpace the smoothing parameter is $\gamma = 3.0$.

3.5.3 Posterior Soft Assignments

When we presented results for BayXenSmooth for comparison purposes, we primarily focused on the discrete hard assignments achieved by taking the maximum index of the soft assignments. However, in order to ensure our method works on various resolutions of data where each spot may be a composition of multiple distinct cell types, we must verify that our approach can recover true soft assignments when they exist. Therefore, we assess whether the approximate posterior learned by BayXenSmooth meaningfully departs from the initial mode assignment rather than simply reinforcing it. However, BayXenSmooth utilizes an empirical prior from an informed initial assignment. When the prior is non-informative,

prior collapse is undesirable because it suggests that the model has collapsed and is simply copying the prior rather than integrating meaningful signal from the data. In this case, prior collapse would indicate that performance gain is simply achieved by aggregating neighboring information without meaningful posterior refinement. Ideally, we would like to show that approximate variational inference refines the initial assignment. We demonstrate that the approximate posterior improves upon the empirical prior distribution. BayXenSmooth avoids prior collapse, indicating that our approach infers something more intricate than a simple mode correction despite being initialized at the mode-centered empirical distribution.

We generated our synthetic data with spatially structured soft assignments. Because we explicitly define the data generating process, we have access to the true soft assignments ($w_{i,k}^{\text{true}}$) and we can evaluate the posterior's performance using the forward KL divergence and Monte Carlo samples from the approximate posterior:

$$\begin{aligned}\frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_{\hat{w}_i\sim q_{\phi_{l_i}}(w_i)}\left[\mathrm{KL}\left(\hat{w}_i\|w_i^{\text{true}}\right)\right] &= \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_{\hat{w}_i\sim q_{\phi_{l_i}}(w_i)}\left[\sum_{k=1}^{K}\hat{w}_{i,k}\log\frac{\hat{w}_{i,k}}{w_{i,k}^{\text{true}}}\right] \\ &\approx \frac{1}{N}\sum_{i=1}^{N}\frac{1}{S}\sum_{s=1}^{S}\sum_{k=1}^{K}\hat{w}_{i,k}^{(s)}\log\frac{\hat{w}_{i,k}^{(s)}}{w_{i,k}^{\text{true}}}, \quad \hat{w}_i^{(s)}\sim q_{\phi_{l_i}}(w_i).\end{aligned}$$

Figure 3.8 showcases the KL divergence between the true soft assignments from our synthetic experiments and the learned approximate soft assignments at every epoch of our SVI training procedure. The consistently observed decrease in KL divergence indicates that while the empirical prior serves as a worthy starting point for approximate inference, executing variational inference results in a posterior distribution that better reflects the true cluster weight distribution.

3.6 Human Breast Data Experiment

The data received from the Xenium Onboard Analysis software provides a complete list of transcripts reads organized with their cell id, nucleus overlap, location, and confidence metric of accuracy via a quality value score. To benchmark BayXenSmooth on data from real tissue, we specifically worked with formalin-fixed paraffin-embedded (FFPE) human breast sample with custom add-on panels. Detailed specifications about the data are provided in Table 3.1.

The intent of BayXenSmooth is to identify spatially contiguous domains in a tissue that retains biological significance. Having an isolated spot in a tissue that belongs to a different domain than all of its neighbors can be challenging to interpret in practice. As a starting

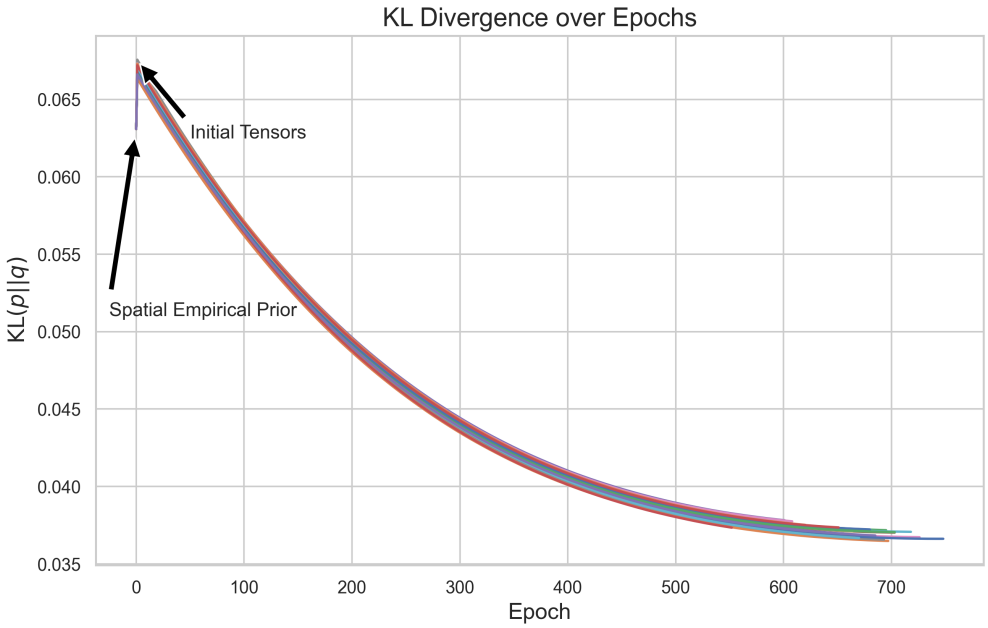


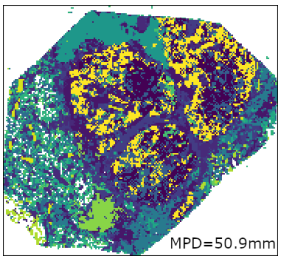
Figure 3.8: Kullback-Leibler divergence values between true cluster memberships and the approximate posterior of cluster memberships learned by BayXenSmooth at each epoch across 25 runs. For these 25 runs, we train BayXenSmooth for a maximum of 2500 epochs with early stopping patience of 5 epochs without ELBO improvement.

Metric	**Tissue Sample**
Median transcripts per cell	94
Cells detected	356,746
Decoded transcripts per 100 μm^2	75.3
Total transcripts detected	36,519,833
Scanned tissue area (cm^2)	0.72

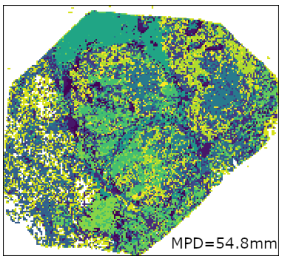
Table 3.1: Formalin-Fixed Paraffin-Embedded (FFPE) Human Breast with pre-designed panel (hBreast) key metrics.

point, we cluster the hBreast dataset at a spot size of 50μm x 50μm and calculated the sum of mean pairwise distances over the various clusters.

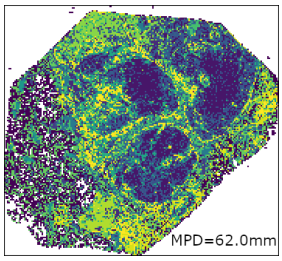
Using $K = 17$ clusters—a choice motivated by an early annotation of the hBreast data—Figure 3.9 visualizes the cluster assignment for the hBreast dataset at a spot size of 50μm x 50μm. We observe that regardless of the spot size, BayXenSmooth is able to generate clusters that are closer together than any of the methods we compare against. While this proximity suggests spatial continuity, it alone does not imply superior biological relevance; we must further demonstrate that the biological significance of these clustered regions is preserved. This will be explored in a subsequent section.



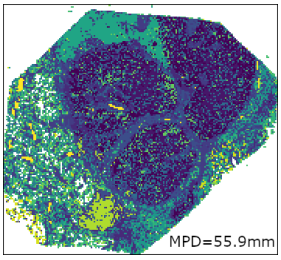
(a) BXS (ours)



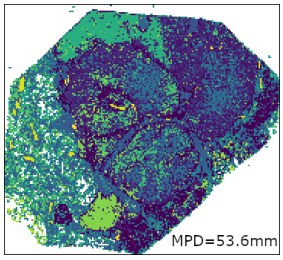
(b) BayesSpace



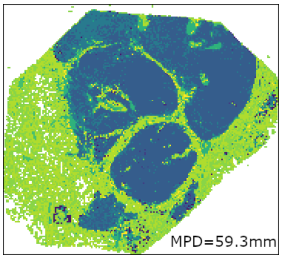
(c) K-means



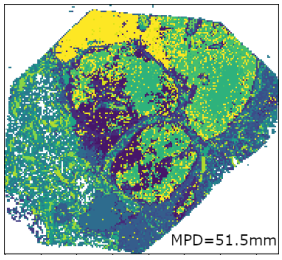
(d) Leiden



(e) Louvain



(f) Hierarchical



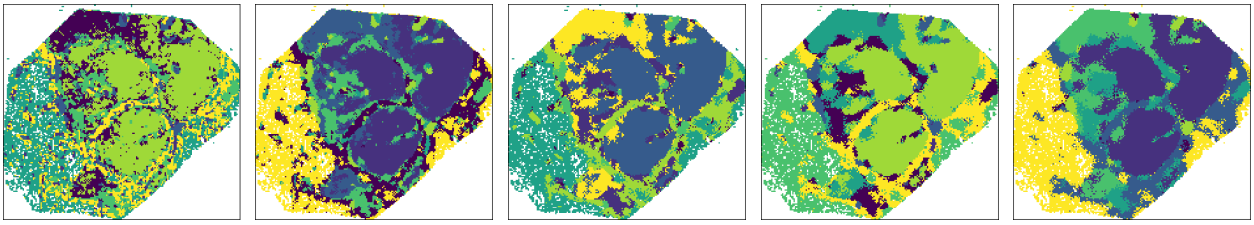
(g) mclust

Figure 3.9: Spatial cluster assignments detected by BayXenSmooth and competing methods. In this setting, $K = 17$ and the run chosen for each method was the best version for identifying the spatial autocorrelation of the marker gene POSTN. The sum of the mean pairwise distances (MPD) of all clusters is included in mm in the bottom right of each plot.

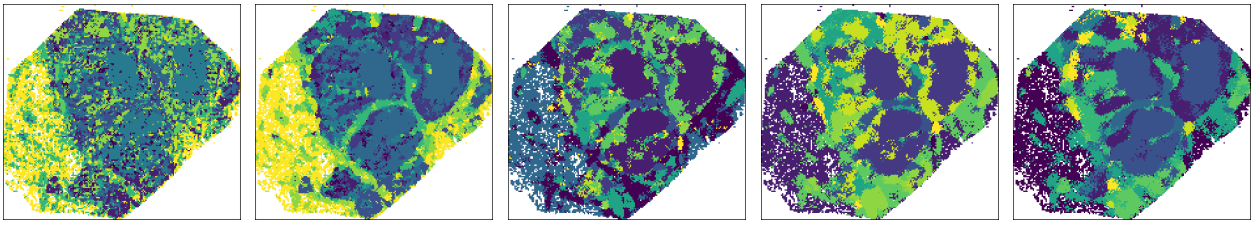
3.6.1 Sensitivity Analysis

While the prior distributions for each spot's cluster assignments have explicit distributions, they are constructed empirically based on a pre-defined neighborhood structure—a non-trivial design choice. For BayXenSmooth, we considered rook and queen criteria to define each spot's neighborhood. Figure 3.4b demonstrates an example with $r = 1$. Increasing the value of r polynomially increases the number of adjacent signals that inform the prior distribution. This effect is clearly demonstrated in Figure 3.10, which depicts how varying

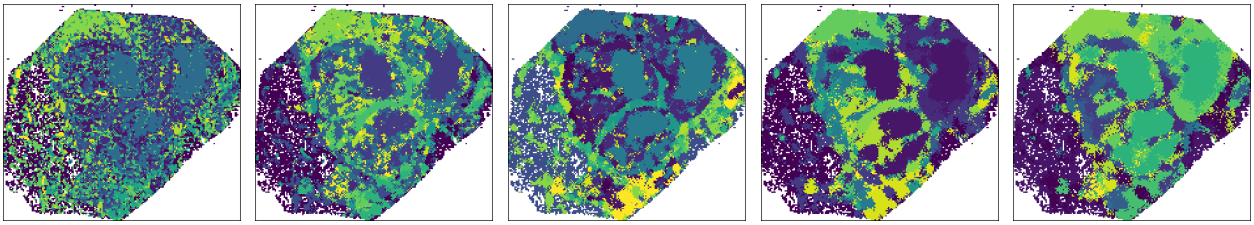
the neighborhood size influences the clustering outcomes for selected values for the number of clusters K. Selecting the optimal neighborhood size presents a balance: incorporating too many neighboring spots can dominate the analysis, potentially overshadowing smaller, biologically meaningful micro-communities. Conversely, selecting fewer neighbors may fail to detect spatial signaling that emerges only in broader contexts, especially in subcellular spots. Performing sensitivity analysis by varying r allows us to see how our model could detect fine-grained spatial phenomena such as subtypes or micro-community structure at smaller neighborhood sizes, as well as differential expression between major anatomical regions at larger neighborhood sizes. This analysis confirms that biologically relevant communities, whether large or small, are detected consistently across resolutions.



(a) K = 7, r = 1 (b) K = 7, r = 2 (c) K = 7, r = 3 (d) K = 7, r = 4 (e) K = 7, r = 5



(f) K = 12, r = 1 (g) K = 12, r = 2 (h) K = 12, r = 3 (i) K = 12, r = 4 (j) K = 12, r = 5

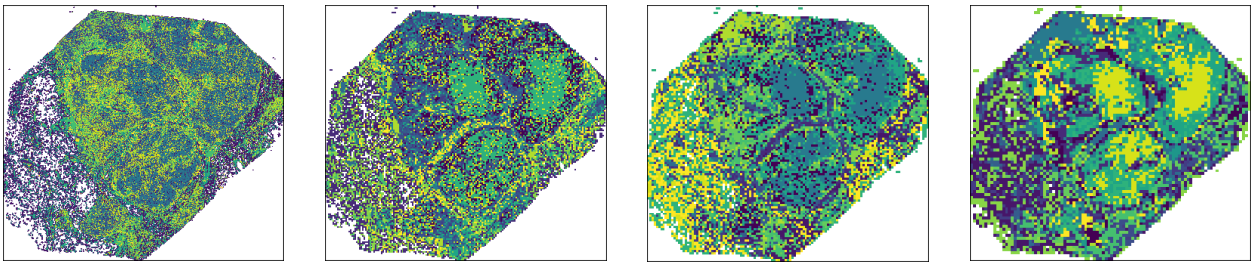


(k) K = 17, r = 1 (l) K = 17, r = 2 (m) K = 17, r = 3 (n) K = 17, r = 4 (o) K = 17, r = 5

Figure 3.10: Cluster assignments across varying configurations of the number of clusters, K, and the radius of neighboring spots, r. Moving from left to right, the diagrams show the influence of a strengthened spatial prior, enforcing a target spot to be similar to more neighboring spots. From top to bottom, the visualizations reveal the division and expansion of cluster groups as BayXenSmooth refines its estimation of higher-dimensional component weight posteriors at a spot size of 50 μm x 50 μm.

3.6.2 Resolution Analysis

Advancements in spatial transcriptomics have introduced expression data available at assorted spot sizes. Platforms like Xenium provide subcellular transcriptional reads, while others, such as the original spatial transcriptomics method or GeoMX, utilize larger spot sizes, sometimes exceeding 100μm x 100μm (Smith et al., 2024). BayXenSmooth is designed to parameterize spatial priors and posteriors across this diversity of resolutions. Figure 3.11 illustrates the spatial communities identified by BayXenSmooth across several resolutions. Groups of cells can send signals collectively, but the size of these groupings is frequently unknown a priori. Our method's ability to identify local communities in tissue samples at any resolution enables the detection of biologically relevant structures that may vary in scale. This facilitates insights into both fine-grained cellular interactions and broader spatial patterns, which are critical for understanding tissue organization and function. Similar to increases in neighborhood size, increases in effective spot size create more contiguous regions with fewer small communities. This effect is akin to a pooling operation in a convolutional neural network, taking inputs from several signals and summarizing the region with a single value. As spot size increases, a larger proportion of the tissue is included in each spot and in the union of its neighbors.

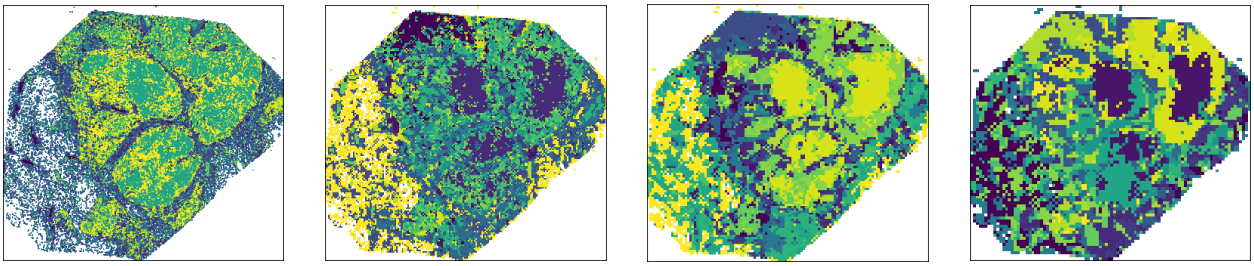


(a): **Method**: K-means, **Spot Size**: 25μm x 25μm

(b): **Method**: K-means, **Spot Size**: 50μm x 50μm

(c): **Method**: K-means, **Spot Size**: 75μm x 75μm

(d): **Method**: K-means, **Spot Size**: 100μm x 100μm



(e): **Method**: BayXenSmooth, **Spot Size**: 25μm x 25μm

(f): **Method**: BayXenSmooth, **Spot Size**: 50μm x 50μm

(g): **Method**: BayXenSmooth, **Spot Size**: 75μm x 75μm

(h): **Method**: BayXenSmooth, **Spot Size**: 100μm x 100μm

Figure 3.11: Learned spatial communities at spot sizes of 25 μm x 25 μm, 50 μm x 50 μm, 75 μm x 75 μm, and 100 μm x 100 μm. Runs are replicated on the same setting presented in Figure 3.12 for the KRT6B gene. (**Top**) K-means initialization at varying spot sizes. (**Bottom**) BayXenSmooth cluster outputs at varying spot sizes.

3.6.3 Marker Gene Analysis

Having demonstrated that BayXenSmooth identifies smooth, spatial posterior cluster assignments, we now show that the expression profiles of these regions carry biological significance. Among the known breast cancer marker genes are: CEACAM6, FASN, KRT6B, POSTN and TCIM. CEACAM6 plays a role in cell adhesion and is widely used as a tumor marker in determinations of carcinoma, which can include breast cancer (Wu et al., 2024). FASN is associated with fatty acid synthesis which can create poor prognosis in breast cancer patients; the expression of FASN can change the levels of specific fatty acids that promote tumor cell migration. (Fu et al., 2024; Xu et al., 2020). KRT6B is a prominent marker gene for basal-like breast cancer as opposed to breast cancers of other molecular subtypes. (Charafe-Jauffret et al., 2005). POSTN has been implicated in multiple processes of tumor development, including angiogenesis, invasion, cell survival, and metastasis (Labrèche et al., 2021). TCIM encodes a protein that functions as a positive regulator in the Wnt/β-catenin

signaling pathway in human breast cancer (Su et al., 2013). The BANK1 gene, while not usually attributed to breast tissue has genetic variants related to several autoimmune diseases, especially lupus erythematosus (Gómez Hernández et al., 2021). The relevance of these markers is further validated by comparing BayXenSmooth's results with those from Integrative and Reference-Informed Tissue Segmentation (IRIS) (Ma and Zhou, 2024). IRIS recently showed that their reference-based procedure identified similar marker genes that were differentially expressed across their learned spatial domains. The mean expressions of these identified marker genes across the posterior cluster assignments learned by BayXenSmooth closely resembles those calculated in the referenced-based method (Figure 3.12). This outcome represents that smoother spatial regions identified by BayXenSmooth can still capture relevant biological differential expressions across spatial domains.

To quantify marker gene differential expression, we use the spatial autocorrelation metric Global Moran's I. The Global Moran's I quantifies spatial autocorrelation by measuring the covariance in a given feature between neighboring spatial locations, weighted according to a predefined spatial proximity matrix. Positive Moran's I values indicate clustering of similar values, whereas negative values imply a spatially dispersed distribution.

The formula for Moran's I spatial autocorrelation is

$$I := \frac{N}{W} \frac{\sum_{i=1}^{N} \sum_{j=1}^{N} w_{ij} \left(x_i - \bar{x}\right) \left(x_j - \bar{x}\right)}{\sum_{i=1}^{N} \left(x_i - \bar{x}\right)^2}$$

where x_i represents the feature value for observation i, $\bar{x}$ represents the global average of the feature, w_{ij} represents a spatial weight between observations i and j, N is the sample size, and $W := \sum_{i=1}^{N} \sum_{j=1}^{N} w_{ij}$. The choice of the spatial weights $\{w_{ij}\}_{i,j=1,\ldots,N}$ is a critical design decision for spatial autocorrelation evaluation that shapes the analysis by influencing which spatial relationships are most accentuated. For the marker gene analysis, w_{ij} is designed to ensure w_{ij} is higher when i and j are in close proximity and belong to the same cluster, lower if they belong to the same cluster but are far apart, and zero if they belong to different clusters. In this analysis, w_{ij} is constructed based on distances in a low-dimensional embedding space generated by the UMAP algorithm. Additional details about the proximity matrices and the Moran's I statistic are provided in Appendix B.2.

If we define the UMAP distances as d_{ij}, then

$$w_{ij} := \mathbb{I}(z_i = z_j) \cdot d_{ij}^{-1}. \tag{3.7}$$

If we let $x_{i,g}$ to represent the expression of gene g at spot i, and use the weights as defined in (3.7), we can denote the Moran's I value as

$$I_g := \frac{N}{W} \frac{\sum_{i=1}^{N} \sum_{j=1}^{N} w_{ij} \left(x_{i,g} - \bar{x}_g\right) \left(x_{j,g} - \bar{x}_g\right)}{\sum_{i=1}^{N} \left(x_{i,g} - \bar{x}_g\right)^2}.$$

The Global Moran's I values for selected marker genes can be found in Table 3.2. For our purposes, a higher value implies stronger spatial autocorrelation of the gene expression feature while a negative value implies features with similar values are distant and spread out. The aim of our clustering exercise is to preserve biologically relevant groupings while considering spatial proximity. However, methods that explicitly promote spatial clustering——such as those that minimize distances between spots within the same cluster——tend to have lower spatial autocorrelation of gene expression. While this may produce clearer spatial segmentation, it can also come at the cost of over-smoothing, potentially masking biologically significant expression heterogeneity. In contrast, methods that do not prioritize minimizing spatial distances may yield larger autocorrelation values, preserving more expression variation at the expense of spatial coherence. We demonstrate that there exist marker genes for which our approach finds an optimal clustering that balances this trade-off.

At high sample sizes, the Global Moran's I has a normal distribution and can be transformed to a z-statistic that we could use for hypothesis testing. However, due to the excessively large sample sizes that occur at higher resolutions, these tests are frequently statistically significant even with the most generous multiple testing correction (Lin et al., 2013). Therefore, we demonstrate that the ordinal rank of notable marker genes remains mostly intact compared to previous works despite spatial constraints introduced by a strong prior. The ordinal ranks of marker genes hold clinical relevance, as they allow for prioritized examination of genes with notable spatial expression patterns, aiding in the efficient identification of potential biomarkers or therapeutic targets. In Table 3.2 we also present each gene's Moran's I ordinal rank out of 280 candidate genes. These results align closely with those produced by BayesSpace, as well as with groupings based solely on transcriptomic measurements.

To assess the spatial proximity of the cluster labels, we can again apply the Global Moran's I but to a different variable. If we allow the selected feature to be a binary indicator of class assignment

$$x_i := I(z_i = k),$$

we can measure the spatial coherence of each cluster by calculating the Moran's I for each class separately. Because the cluster assignment indicators are now the feature being considered, the weights are exclusively defined by the UMAP distances, ignoring cluster labels:

$$w_{ij} = d_{ij}^{-1}.$$

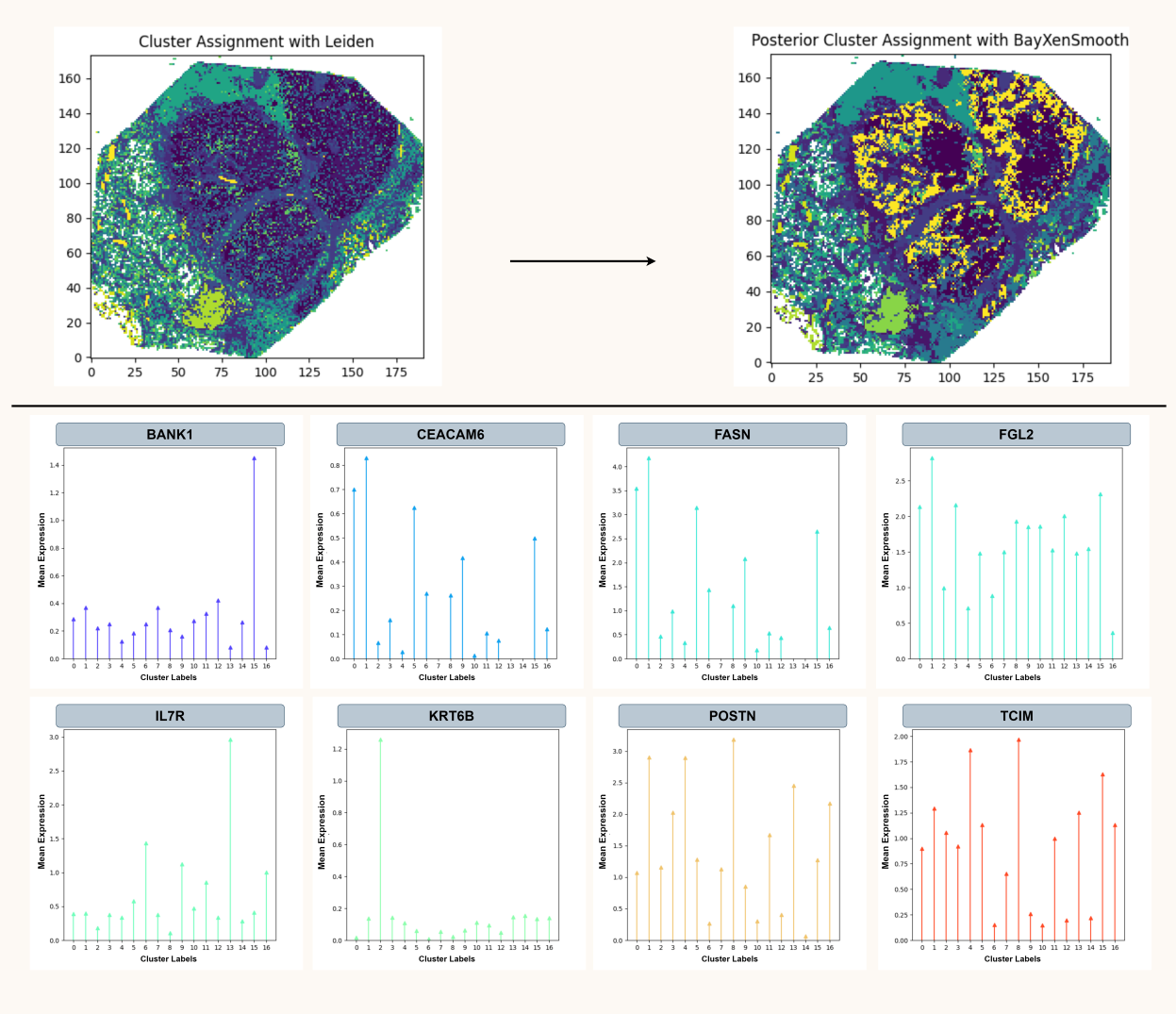


Figure 3.12: Marker gene analysis. (**Top**) Example transformation of clusters: Leiden (initialization) on the left, BayXenSmooth posterior assignments on the right (specific plot taken for the case of TCIM). (**Bottom**) Mean log1p expressions of selected marker genes across clusters. The expression variation across spatial regions aligns with spatial variation identified by a state-of-the-art reference-based method (Ma and Zhou, 2024).

Formally, for each cluster k we compute

$$I_k := \frac{N}{W} \frac{\sum_{i=1}^{N} \sum_{j=1}^{N} w_{ij} (x_i - \bar{x}) (x_j - \bar{x})}{\sum_{i=1}^{N} (x_i - \bar{x})^2}$$

and obtain a global Moran's I of

$$I = \frac{1}{K} \sum_{k=1}^{K} I_k.$$

High positive Moran's I values in these binary indicators suggest that spots assigned to the same cluster are spatially contiguous. Analyzing these results alongside the spatial autocorrelation of marker gene expression reveals that BayXenSmooth uniquely preserves biologically meaningful domains while minimizing fragmented or isolated cluster assignments. Table 3.3 presents the Moran's I values for cluster indicators, with a visualization of the trade-off in Figure 3.13. Although BayXenSmooth records slightly lower nominal spatial autocorrelation for marker gene expressions compared to BayesSpace and other non-spatial methods, it compensates by maintaining clear spatial coherence in cluster assignments. The preservation of ordinal rank across clusters validates our model's effectiveness in retaining biological relevance in clusters informed by spatial constraints.

3.7 Discussion

In this chapter, we introduced BayXenSmooth——a stochastic variational inference model designed to identify spatial clusters in high-throughput Xenium tissue data. By constructing a prior distribution based on the initialization, we learn posterior distributions for global means and variances describing the cluster groups and local cluster weights. Weaker priors allow for posteriors to have stronger biological interpretations while stronger priors fixate on spatial contiguity. We demonstrate in both real and synthetic data cases that incorporating an informative prior can be advantageous. This model effectively clusters tissue spots into spatially coherent regions while preserving biological significance despite variations in design choices, such as the number of clusters or neighborhood configuration. By consolidating transcriptional data into spots, BayXenSmooth enables comparisons across a range of resolutions, enhancing its applicability across diverse spatial omics technologies.

Relying solely on a neighborhood definition and an initial expression-based clustering, BayXenSmooth achieves spatial clustering without requiring external reference annotations. Ideally, the choice of the initialization should be a clustering procedure that outputs a realistic grouping of the gene expressions devoid of spatial context. In this work, we present results using K-means, Leiden, and mclust as candidates for the initialization. Methods such as K-

Gene\Model	BayXenSmooth	BayesSpace	K-means	mclust	Leiden	Louvain	Hierarchical
BANK1	1.154 (**#2**)	1.523 (**#2**)	1.449 (**#2**)	1.220 (#3)	1.403 (#3)	1.219 (#3)	0.656 (#54)
CEACAM6	0.406 (**#105**)	0.421 (#146)	0.385 (#194)	0.424 (#131)	0.343 (#206)	0.338 (#202)	0.441 (#135)
FASN	1.156 (**#1**)	1.217 (#2)	1.158 (#5)	1.276 (#2)	0.995 (#14)	0.929 (#21)	1.119 (#2)
FGL2	0.513 (#128)	0.603 (#130)	0.654 (**#121**)	0.485 (#131)	0.564 (#145)	0.497 (#150)	0.343 (#145)
IL7R	0.905 (#24)	1.278 (#5)	1.256 (**#4**)	0.970 (#47)	1.007 (#14)	0.904 (#20)	0.483 (#71)
KRT6B	0.751 (#76)	0.920 (#69)	0.963 (#27)	0.732 (#74)	0.908 (#30)	0.870 (**#26**)	0.251 (#112)
POSTN	0.547 (**#90**)	0.554 (#122)	0.478 (#166)	0.561 (#119)	0.615 (#123)	0.584 (#120)	0.343 (#142)
TCIM	0.701 (#72)	0.840 (**#42**)	0.472 (#168)	0.810 (#59)	0.834 (#71)	0.781 (#78)	0.375 (#157)

Table 3.2: Moran's I values for selected marker genes (Resolution: 50 μm x 50 μm). Each value includes its ordinal rank. Top-ranked performers are in bold and underlined.

Gene\Model	BayXenSmooth	BayesSpace	K-means	mclust	Leiden	Louvain	Hierarchical
BANK1	**0.343**	0.244	0.246	0.242	0.306	0.311	0.190
CEACAM6	0.375	**0.398**	0.246	0.224	0.306	0.311	0.190
FASN	**0.375**	0.301	0.246	0.224	0.306	0.311	0.190
FGL2	**0.411**	0.244	0.246	0.242	0.306	0.311	0.190
IL7R	**0.407**	0.244	0.246	0.242	0.306	0.311	0.190
KRT6B	**0.341**	0.305	0.246	0.321	0.306	0.311	0.190
POSTN	**0.455**	0.335	0.246	0.381	0.306	0.311	0.190
TCIM	**0.459**	0.377	0.246	0.381	0.306	0.311	0.190

Table 3.3: Moran's I values for cluster membership indicators across different methods (Resolution: 50 μm x 50 μm). For each marker gene, the model with the highest Moran's I value is highlighted in bold.

Figure 3.13: Side by side comparison of marker gene Moran's I values and cluster indicator Moran's I values.

means and Leiden allow for control over the number of clusters (the latter being contingent on identifying the optimal resolution parameter). In contrast, mclust determines the number of clusters using the Bayesian Information Criterion (BIC), providing an a priori approach to identifying the optimal number of clusters. Incorporating a spatial prior helps to refine uncertain regions, improving overall spatial coherence. Our approach can further extend to scenarios requiring reference-based annotation, where label estimates may be incomplete or noisy. In such cases, spatial priors informed by neighborhood structure can bolster initial estimates, adding robustness in complex tissue environments where cell-type markers may be sparse or unreliable.

Furthermore, BayXenSmooth has computational advantages over several preceding approaches. Similar to prior work, BayXenSmooth relies on a neighborhood structure to learn a spatial clustering. However, this structure is not needed to sample from a learned posterior. This is beneficial in cases where the adjacency matrix is not sparse or is expensive to store and propagate through. The posterior soft assignment vectors are learned independently, allowing BayXenSmooth to perform parallel sampling for each spatial location. This parallelization is equally applicable to the training procedure, yielding over a 10-fold speedup compared to BayesSpace when taking 1000 posterior samples per spot when using 3 principal components. The advantage becomes more pronounced as the dimension of the data grows (Figure 3.14).

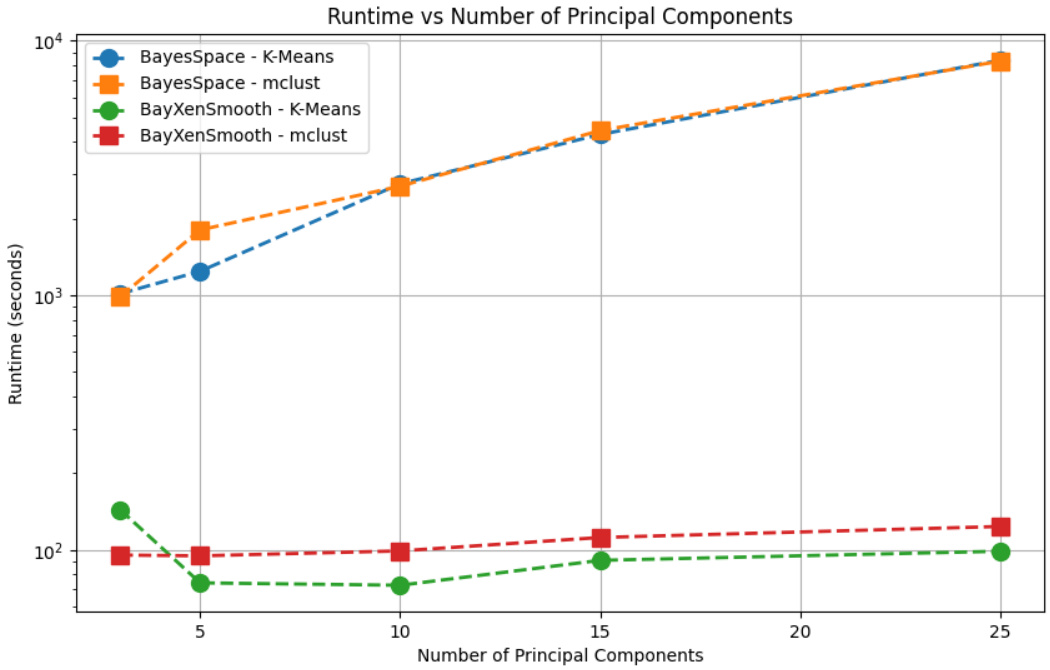


Figure 3.14: Runtime comparison between BayXenSmooth and BayesSpace across initializations and number of principal components considered analyzing the hBreast dataset.

BayXenSmooth takes existing cluster assignments generated for spatial transcriptomics and transforms them to be more spatially organized beyond a mere post-processing method. The posterior learned by BayXenSmooth enables downstream analyses to integrate uncertainty estimates, which are crucial when examining complex tissue architectures. Additionally, by leveraging the posterior distribution, BayXenSmooth facilitates adaptive resolution across tissue regions, making it applicable in scenarios where regions may vary in structural complexity.

The most obvious limitation of BayXenSmooth is the reliance on a trustworthy initial cluster assignment. This assumes that the data has spatial clusters worth learning and there exists a clustering method that learns the spatial clusters effectively enough to warrant an empirical, approximate Bayesian smoothing procedure. While in many settings, we justify that both simulated and applicable datasets meet these assumptions, there is no theoretical guarantee for it. However, given the vast array of clustering methods available BayXenSmooth provides a way to refine any of them with spatial informed probabilistic inference.

Another limitation is that BayXenSmooth is designed to learn a posterior distribution that treats the cluster of every spot independently. The mean-field assumption we employ for the approximate posterior distribution of cluster logits, while common in practice for

computational purposes, may not be a true assumption from a biological perspective. While the empirical prior employed by BayXenSmooth encourages nearby spots to share similar cluster compositions, the variational posterior itself by design cannot represent such dependencies, creating a mismatch between the spatial, empirical prior and the approximate posterior. A variational family with learnable inter-spot correlations could allow for posterior cluster distributions that may better reflect biological dependence structures but at the cost of scalability afforded by the mean-field assumption.

We conclude with two promising directions for extensions of BayXenSmooth: refining neighborhood structures to better reflect biological phenomena and formalizing its use in reference-based settings.

3.7.1 Biologically Informed Neighborhood Structures

The neighborhood structure used in this work took rook or queen neighbors of spots in the tissue to compare against previous works. The optimal neighborhood cardinality varies with spot size and the expressions we try to model due to the wide range of spatial variances and dependence structures among genes. Additionally, the way neighboring information is currently used only accommodates local relationships between tissue regions emulating paracrine and juxtacrine signaling. However, cells can communicate to each other via endocrine signaling. This involves signals being sent via the bloodstream which can span large distances within a tissue. Incorporating information such as proximity to bloodstream and other biological factors can provide a more accurate spatially aware prior distribution. As another example, the Human DLPFC 10x Visium dataset has references that have a higher spatial similarity in its primary axis than its secondary; even this morsel of information can prove useful for constructing neighborhoods for prior distributions. Unlike previous methods adapted to support a rigid neighborhood definition, our method is applicable to any definition because model inference and posterior samples only depend on a summary statistic of the neighborhood graph.

3.7.2 Extension to Reference-based Data

While BayXenSmooth was designed for applications in reference-free environments, further research could enhance its utility in confidently assigning references, even in complex spatial datasets. For example, spatially adjacent clusters often exhibit boundary regions with high uncertainty, especially in cases where cell types or tissue regions have gradual transitions rather than sharp separations. Proper analysis of these boundaries is essential for minimizing misclassification risks and improving interpretability. Leveraging BayXenSmooth as a

probabilistic correction technique allows for posterior annotations based on predefined labeling assumptions, such as spatial autocorrelation or prior knowledge of tissue structure. Mathematically, this is accomplished by setting each spot's initial cluster assignment z_i to be the one-hot encoded annotation. By learning clusters from the combined influence of the spatial prior, likelihood model, and approximate posterior, BayXenSmooth enables comparison between initial annotations and posterior-inferred labels, allowing us to pinpoint regions where the data-driven model refines or challenges the original annotations.

CHAPTER 4

Amortized and Generalizable Approximate Bayesian Spatial Clustering with Normalizing Flows

4.1 Introduction

The ability to segment tissue samples based on biological composition and spatial context has long been a central challenge in computational biology. Clustering regions of tissue plays a crucial role in identifying and understanding functional regions, types, or states within biological systems. In spatial transcriptomics, clustering can reveal patterns in gene expression linked to tissue function or pathology. Previous works have developed clustering procedures that accurately recovered tissue-specific expression patterns, tumor micro-environments, and developmental zones among many other tissue attributes (Arora et al., 2023b; Long et al., 2023; Stickels et al., 2020b). Most traditional clustering algorithms create separable groups in latent spaces with substantially fewer dimensions relative to the ambient space. However, these lower-dimensional representations are not directly interpretable or observable, and thus less useful for the aforementioned applications.

Bayesian modeling is a promising solution to this problem. Bayesian clustering models can encourage spatial coherence by defining a prior distribution that incentivizes spots within a neighborhood to share assignments. Spatial transcriptomics data is a prime example of a setting where such priors are valuable, as applications including but not limited to targeted therapy, drug screening, and tissue engineering work best when cluster regions are separable in the spatial dimensions (Habern, 2024; Krause et al., 2018; Wang et al., 2024). While in practice each region might be tied to a single group, there are benefits to measuring the uncertainty of this decision or understanding if spots exhibit features of more than one cluster. This motivates learning posterior distributions over soft assignment vectors rather than committing to hard labels. Ideally, with an appropriate model, we can identify

reasonable posterior cluster assignment distributions. However, as the resolution of spatial transcriptomics measurements increases, the inference approach needs to scale appropriately.

In this chapter, we investigate the effectiveness of replacing conjugate distributions in exact Bayesian procedures with normalizing flows as approximate posteriors for spatial clustering tasks in transcriptomics data. Motivated by methods that use exact Bayesian inference, we propose an approximate, variational framework that can explore a wider array of posterior families using normalizing flows. This method yields approximate posteriors capable of capturing the spatial dependencies inherent in the data while remaining tractable.

We demonstrate that normalizing flows successfully recover spatial structures in both synthetic and real datasets. These results provide greater confidence in mapping tissue samples where the ground truth is unavailable——a scenario frequently encountered in spatial transcriptomics.

4.2 Background

4.2.1 Previous Work

In spatial transcriptomics, gene expression is tightly coupled with spatial context, making clustering not just a statistical task, but a means of uncovering the functional architecture of tissue. Accordingly, a growing body of clustering methods has been proposed to make explicit use of spatial dependencies during model design and inference. One prominent model family is exact Bayesian inference models that learn latent variables corresponding to cluster assignment with applications in spatial domain detection and cell type clustering. These developments were prompted by the dominance of legacy methods, despite their limitations. For instance, the 10x Genomics Xenium platform only provides K-means and graph-based clusterings as defaults. These methods typically assume independent cluster assignments across spots, relying solely on local gene expression to determine labels. This is restrictive because without incorporating spatial information, clusters are learned disregarding the fact that neighboring spots often share biological function. To combat this, the hidden-Markov random field (HMRF), single-cell minimum enclosing ball (scMEB), BayesSpace, and Bayesian Analytics for Spatial Segmentation (BASS) have been proposed as state-of-the-art Bayesian clustering methods for spatial transcriptomics data (Li and Zhou, 2022; Yang et al., 2021; Zhang et al., 2001; Zhao et al., 2021). These methods typically model each spot's expression y_i as being drawn from a cluster-specific distribution, conditioned on an assignment z_i: $y_i \mid z_i \sim f(z_i)$. Unlike non-spatial and non-Bayesian approaches, these works explicitly account for spatial dependencies, often through hierarchical structures

and graphical models, thereby capturing both the local spatial coherence and the underlying cluster-specific expression patterns. This is most commonly accomplished with a Potts model (Wu, 1982), which leads to a hierarchical model:

$$\begin{aligned} \boldsymbol{z} &\sim \text{Potts}(\text{A}, \gamma) \\ y_i | z_i &\sim f(z_i), \end{aligned}$$

where A is the adjacency matrix encoding spatial relationships and γ controls the strength of spatial smoothing. Typically, $f(z_i)$ is represented by a Gaussian mixture model (GMM), either via a hierarchy or as a weighted sum of samples from independent component Gaussian distributions.

The Potts model is desirable in this setting because it represents a joint prior over all latent variables with a tractable spatial dependency structure adhering to an adjacency matrix and smoothing hyperparameter (A, γ). Moreover, the prior probability of assignments under the Potts model is trivial to compute. For instance, in the BayesSpace model, the Potts prior is represented as

$$p(z_i = k \mid \boldsymbol{z}_{-i}) \propto \exp\left(\frac{\gamma}{\mid \langle ij \rangle \mid} * 2 \sum_{\langle ij \rangle} I(z_i = z_j) \right),$$

where $\langle ij \rangle$ is the set of neighbors for the target spot i and $|\langle ij \rangle|$ is the total number of neighbors spot i has. This property allows for efficient inference using Gibbs sampling or Metropolis-Hastings when the model is designed with careful conjugacy.

Despite demonstrated predictive performance, exact Bayesian inference models can have limitations in these settings. Our work can be thought of as an extension of previous approaches by addressing potential limitations:

Flexible Posterior Modeling: Exact spatial Bayesian methods require parametric posteriors enforced by conjugacy—a strict constraint. Replacing these distributions with normalizing flows allows for the exploration of a broader class of models that are not limited by tractable conjugate priors.

Flexible Prior Constraints: The appeal of the Potts model in exact Bayesian inference is that it parametrically imposes an incentive that neighboring entities are clustered similarly. However, sometimes a more complicated graphical constraint can be more informative for cluster assignment predictions.

Gaussian Mixture Modeling: Our method assumes that a spot's gene expression can indicate membership in multiple domains or cell types, leveraging a Gaussian mixture formu-

lation. While previous work conditions gene expression on hard assignments sampled from soft assignment posteriors, we directly model the uncertainty without requiring intermediate hard assignments, making our approach more adaptable to complex, heterogeneous data.

Scalability: Spatial transcriptomics data has become very high resolution, which yields extremely large datasets. Furthermore, the model specification and exact spatial dependency between spatial tissue units is an open question, and we may want to explore multiple configurations. Approximate inference is better designed for environments where we want to explore a wide array of models.

Our work mitigates these limitations by shifting from an exact inference regime to an approximate one with normalizing flows. In the following subsections, we outline all of the prerequisite concepts necessary to understand our method.

4.2.2 Normalizing Flows

Normalizing flows are a family of methods for learning flexible probability distributions that retain a tractable density. These models transform a relatively simple base distribution into a more expressive one through a sequence of invertible transformations. Using the change of variables formula, for a single transformation $y = g(x)$ we denote the density of y as

$$p_Y(y) = p_X(g^{-1}(y)) \left| \det \left(\frac{dg^{-1}(y)}{dy} \right) \right|.$$

For computational convenience, this expression is reformulated as a sum in the log domain as:

$$\log p_Y(y) = \log p_X(g^{-1}(y)) + \log \left| \det \left(\frac{dg^{-1}(y)}{dy} \right) \right|.$$

Composing many of these transformations together $y = g_P \circ g_{P-1} \circ \cdots \circ g_1(x)$, we obtain a flexible universal density approximator. We define the intermediate flow outputs as $y_0 = x$, $y_1 = g_1(y_0)$, $y_2 = g_2(y_1)$, $\ldots$, $y_P = g_P(y_{P-1})$. y_0 represents a sample from the base distribution while y_P corresponds to the transformed sample from the target distribution. The full log density of this stacked transformation can be expressed as:

$$\log \left(p_Y(y) \right) = \log \left(p_X \left(\left(g_1^{-1} \circ g_2^{-1} \circ \cdots \circ g_P^{-1} \right)(y) \right) \right) + \sum_{p=1}^{P} \log \left(\left| \frac{dg_p^{-1}(y_p)}{dy_p} \right| \right).$$

The major computational bottleneck of normalizing flows is calculating the Jacobian de-

terminant: $\det\left(\frac{\partial g_p^{-1}}{\partial y_p}\right)$. For an $N \times N$ matrix, this operation scales as $O(N^3)$. To address this, several architectures have been designed to ensure that the Jacobian is diagonal, triangular, or otherwise structured in a way that facilitates more efficient computation. Among these approaches, the masked autoregressive flow (MAF) and continuous normalizing flow (CNF) have shown the best performance in our experiments. The MAF leverages autoregressive factorizations to model a target joint distribution, while the CNF defines continuous-time transformations of target distributions whose solutions yield invertible mappings. For the final implementation of XenNF, we chose to use CNFs because they allow a flow to be characterized by a single network and do not require tuning the flow depth. MAFs yielded comparable results and could serve as a reasonable alternative. We provide further computational details about these flow architectures in Appendix C.1.

4.2.3 Variational Inference with Normalizing Flows

Normalizing flows allow us to capture more intricate dependencies between the context variable (gene expression) and the approximate posterior distribution. Unlike traditional Bayesian inference or standard variational inference, where the relationship between the conditioning variable and the distribution is often defined by a fixed parametric form, normalizing flows enable flexible, data-driven transformations that can represent highly complex and non-linear relationships.

In latent variable models, we assume that the observable features x are conditionally dependent on some latent variables z. Fully Bayesian models require a prior $p(z)$, likelihood $p(x|z)$, and posterior $p(z|x)$ that all have some parametric form. Frequently, the posterior distribution is intractable due to the normalizing constant necessary to analytically compute the posterior: $p(z|x) = \frac{p(x|z)p(z)}{\int p(x|z)p(z)dz}$. To obtain samples from the true posterior, computational methods such as Markov Chain Monte Carlo (MCMC) or other sampling techniques are employed. While these methods are robust and unbiased, they often struggle to scale as posterior complexity increases or datasets grow larger. This is largely due to their sequential nature and the need for many samples to achieve convergence and support analysis, making them computationally expensive for modern large-scale applications.

Variational inference modernizes this by reframing the inference problem as an optimization task. Although the distribution learned by VI is not the exact posterior $p(z|x)$, it learns a tractable surrogate distribution $q(z|x)$ that maximizes the evidence lower bound (ELBO), equivalently minimizing the KL-divergence between itself and the true posterior $D_{KL}(q(z|x), p(z|x))$. Furthermore, the computation can scale well if we use stochastic variational inference (SVI) because we can apply stochastic gradient steps in batches to maximize

the ELBO. (We want to explore many different possible posteriors for the soft assignments and exact inference can get in the way of this exploration). Even then, the flexibility of SVI comes with trade-offs. A commonly used approach is to choose a simple variational family for $q(z|x)$, such as factorized distributions, to ensure computational tractability. Examples include mean-field approximations, fully factorized Gaussian variational distributions, and simpler constructions in models such as Variational Autoencoders (VAEs). While applicable for some applications, these approximations often impose restrictive assumptions, such as independence between latent variables or uni-modality, which can fail to capture the true posterior's structure. The work of Rezende and Mohamed allows for normalizing flows to serve as an effective approximate posterior (Rezende and Mohamed, 2016). This advancement enables the use of complex yet tractable distributions as posterior approximations, significantly enhancing the expressiveness of approximate Bayesian methods and narrowing the gap between exact and approximate posteriors.

In the case of attempting to create spatially contiguous cluster assignments for practical applications, the approximate posterior distributions yielded by normalizing flows may identify legitimate spatial patterns that exist near the prior belief. This approach allows for modeling complex posterior structures, such as multi-modal distributions or intricate spatial dependencies, which are often challenging to capture with simpler variational families.

4.3 Method

Our method assumes that some transformation of observed data is described by a Gaussian mixture model, where the cluster means, scales, and logits serve as latent variables:

$$z = (\boldsymbol{l}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \quad \text{where} \quad w_i = \text{softmax}(l_i).$$

We will demonstrate that modeling the logits is preferable in our case to modeling the component weights directly. Here we outline the explicit construction of the likelihood, prior, and approximate posterior we use for approximate posterior inference.

4.3.1 Model Prior

For XenNF, we define a prior distribution for each cluster mean μ_k, each cluster scale σ_k, and each spot's cluster logit vector l_i (from which the weights are derived via softmax). One way to construct the prior distributions over the means and scales is:

$$
\begin{aligned}
\mu_k \in \mathbb{R}^d \sim N(\mathbf{0}, \boldsymbol{I}_{d\times d}) \quad & \forall k \in (1, 2, ...K), \\
\boldsymbol{\mu} &:= (\mu_1, \cdots, \mu_K) \\
\log(\sigma_k) \in \mathbb{R}^d \sim N(\mathbf{0}, \boldsymbol{I}_{d\times d}) \quad & \forall k \in (1, 2, ...K), \text{ where } \Sigma_k = \text{diag}(\sigma_k^2) \\
\boldsymbol{\sigma} &:= (\sigma_1, \cdots, \sigma_K).
\end{aligned}
$$

However, this prior is non-informative. Ideally, we want it to reflect a spatial organization. Given a cluster assignment $z_i\{i = 1, 2, \ldots, N\}$, we can calculate empirical means and scales for each cluster. These empirical estimates then serve as the centers for our prior distributions, and have customizable scales controlled by hyperparameters λ_μ and λ_Σ:

$$
\begin{aligned}
N_k &= \sum_{i=1}^{N} \mathbb{I}(z_i = k) \\
s_k &= \frac{1}{N_k} \sum_{i:z_i=k} y_i \\
\mu_k \in \mathbb{R}^d &\sim N\left(s_k, \lambda_\mu \mathbf{I}\right), \quad \forall k \in (1, 2, ...K) \\
\log(\sigma_k) \in \mathbb{R}^d &\sim N\left(\frac{1}{N_k} \sum_{i:z_i=k} (y_i - s_k)^2, \lambda_\Sigma \mathbf{I}\right) \quad \forall k \in (1, 2, ...K).
\end{aligned}
$$

In previous works that developed exact Bayesian models, the Potts model was utilized to impose a Markov random field structure on the latent cluster variables. We achieve a similar spatially aware structure by modeling the logit dynamics using a shallow graph convolutional network (GCN). Mirroring the structure of the posterior, we define the prior over logits as a conditional normalizing flow. Using a GCN as the hypernetwork enables the prior to incorporate spatial dependencies by conditioning on neighborhood interactions encoded in the tissue graph. By leveraging the graph structure of the tissue, our approach approximates the spatial dependencies captured by the Potts model within a variational framework. Specifically, the dynamics of the logits are defined as:

$$
\frac{\partial l_i(t)}{\partial t} = f_\theta(l_i(t), t, G_r)
$$

where $G_r = (V, A)$ is a graph of the tissue with spots V and adjacency matrix A is an adjacency matrix that represents spatial dependencies between spots. Given a radius r, the

adjacency matrix entry between region i and region j is defined as:

$$A_{ij} = \mathbb{I}\left\{d(i,j) < r\right\}.$$

So, f_θ, parameterized by a GCN, encodes the spatial interactions from the tissue graph and integrates them into the dynamics of the continuous normalizing flow. For convenience, we represent the prior distribution as follows:

$$p_\theta(l_i) := \mathrm{CNF}_\theta(z_i, G_r),$$

where z_i is the base distribution (e.g. standard Gaussian) and CNF_θ transforms it conditioned on graph G_r.

4.3.2 Model Likelihood

Let $f(y_i)$ represent a transformation of the transcriptomic profile (y_i) of a spot (e.g. PCA for dimensionality reduction). In the event we work with raw expression data, we let f be the identity function. The conditional density of $p(f(y_i) \mid w_i, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is modeled as a Gaussian mixture:

$$p(f(y_i) \mid w_i, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \sim \sum_{k=1}^{K} w_{i,k} N(\mu_k, \Sigma_k)$$

where $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_K)$ and $\boldsymbol{\Sigma} = (\Sigma_1, \ldots, \Sigma_K)$. For computational efficiency, we assume diagonal covariance matrices. Therefore, we frequently represent the covariance matrix as $\Sigma_k = \mathrm{diag}(\sigma_{\mathrm{k}}^2)$ where σ_k represents the scales of each data dimension of cluster k.

Because we model the logits, we apply a softmax function over the logits to obtain a valid probability simplex over clusters. Then, the likelihood can be written as:

$$p(f(y_i) \mid \mathrm{softmax}(l_i), \boldsymbol{\mu}, \boldsymbol{\Sigma}) \sim \sum_{k=1}^{K} \mathrm{softmax}(l_i)_k N(\mu_k, \Sigma_k).$$

The GMM is a natural choice for modeling $f(y_i)$ as it allows for the identification of distinct subpopulations or clusters within the dataset, reflecting underlying biological heterogeneity. We let the means and scales be shared parameters over the samples in the dataset while the soft assignments are local to each sample. This is a crucial design choice because we want proximal spots to have similar assignments, but achieving similar assignments should require the spots to share a common distribution.

4.3.3 Approximate Posterior

The aim of an approximate Bayesian method is to choose surrogate distributions for our model parameters that are both tractable and as close as possible to the true posterior $p(\boldsymbol{\mu}, \boldsymbol{\Sigma}, l_i|f(y_i))$ within their distributional family. This can be achieved via stochastic variational inference with a mean-field factorization assumption.

The means and scales of each cluster are shared parameters over all samples of the tissue. These posteriors can be approximated with traditional distributions because the parameters are global and apply uniformly across the dataset. This uniformity allows them to be updated by aggregating evidence from all observations. These approximate posteriors can be written as follows:

$$q_{\phi_{\mu_k}}(\mu_k) = N\left(\mu_k \mid \mu_{\phi_{\mu_k}}, \Sigma_{\phi_{\mu_k}}\right)$$

$$q_{\phi_{\sigma_k}}(\sigma_k) = \text{LogNormal}\left(\sigma_k \mid \sigma_{\phi_{\sigma_k}}, \tau_{\phi_{\sigma_k}}\right).$$

However, each spot has a unique soft assignment that captures local spatial information. To infer the posterior for the soft assignments, the model combines spatial encodings from the prior with the representational power of the normalizing flow, enabling the capture of spatial dependencies from the transcriptomic measurements.

Normalizing flows are typically designed to learn transformations between continuous variables without domain constraints, operating over the entire real line. However, each soft assignment $w_i\{i = 1, 2, \ldots, N\}$ lives on the $(K-1)$-simplex. Therefore, we model the logits with normalizing flows, and perform a deterministic softmax over logit samples to sample the soft assignment probability vector. To ensure identifiability and numerical stability of the logits during training, we constrain their scale by enforcing a zero maximum value:

$$\max(l_{i,1}, l_{i,2}, \ldots, l_{i,K}) = 0.$$

If X represents our base distribution (standard Gaussian), y_i represents the gene data, and $f(y_i)$ represents the transformed gene expression vector we model via the GMM, then the conditional normalizing flow with P transforms $g_1, g_2, \ldots, g_{P-1}$ that yield an approximate log-posterior for the component logits $l_i|f(y_i) = g_P \circ g_{P-1} \circ \cdots \circ g_1(x|f(y_i))$ can be written as:

$$\log q_l(l_i \mid f(y_i)) = \log p_X\left(g_1^{-1} \circ g_2^{-1} \cdots \circ g_P^{-1}(l_i \mid f(y_i))\right) + \sum_{p=1}^{P} \log \left| \det \frac{\partial g_p^{-1}(l_p \mid f(y_i))}{\partial w_p} \right|.$$

Note that we can condition on y_i instead of $f(y_i)$ because f is injective.

Under a mean-field assumption, the variational distributions for the model parameters are defined as $q_{\phi_{\mu_k}}(\mu_k)\{k = 1, 2, \ldots, K\}$, $q_{\phi_{\sigma_k}}(\sigma_k)\{k = 1, 2, \ldots, K\}$, $q_{\phi_{l_i}}(l_i|f(y_i))\{i = 1, 2, \ldots, N\}$. Therefore, our mean-field variational posterior can be expressed as:

$$q_\phi(\boldsymbol{l}, \boldsymbol{\mu}, \boldsymbol{\Sigma} \mid \boldsymbol{y}) = \prod_{i=1}^{N} q_{\phi_{l_i}}(l_i \mid f(y_i)) \prod_{k=1}^{K} q_{\phi_{\mu_k}}(\mu_k) q_{\phi_{\sigma_k}}(\sigma_k)\,.$$

4.4 Experiments

To validate the efficacy of normalizing flows as approximate posterior distributions, we demonstrate their performance on synthetic data, as well as on the dorsolateral prefrontal cortex (DLPFC) tissue dataset.

4.4.1 Synthetic Data

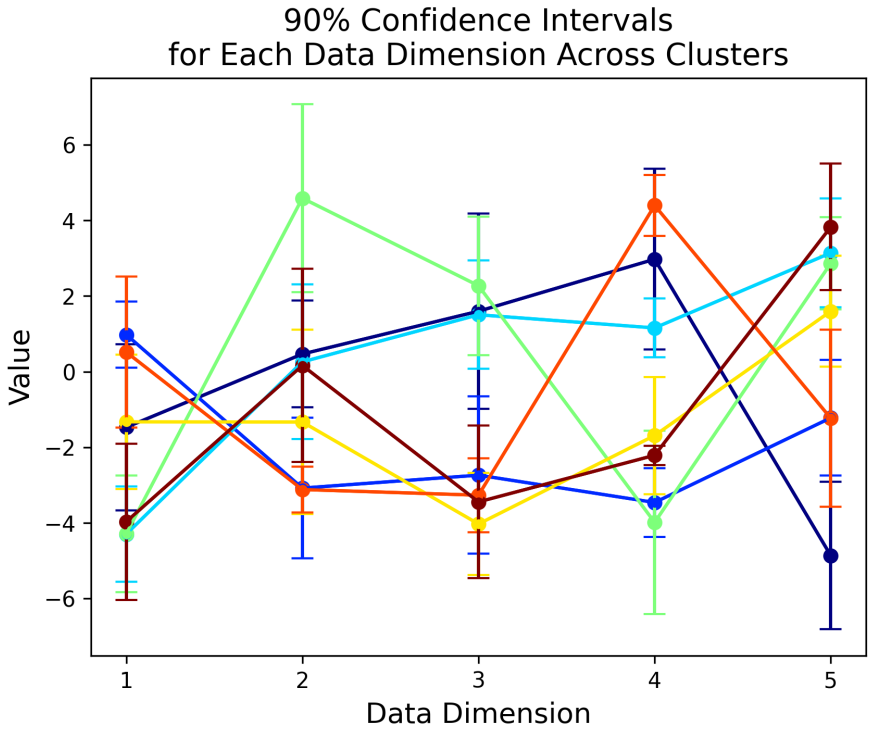


Figure 4.1: 90% confidence interval for each data dimension across clusters in synthetic data. The massive overlap between clusters across several dimensions means that the clustering problem is non-trivial and recovery should be challenging.

In many biological processes, only a small set of marker genes drives spatial differentiation, while most other genes show low variability across sample spots (Li et al., 2023b; Zhao et al., 2024). Even when signal is more diffuse, it is common practice to focus inference on highly variable genes. To evaluate the ability of normalizing flows to perform clustering in this context, we tested their performance on a synthetic dataset designed to emulate this phenomenon, containing 7 clusters and 5 features. The data generating process is the same as Algorithm 1 outlined in Chapter 3, but with a different random seed. We obtain hard labels by taking the argmax over samples from the approximate posterior. Figure 4.2 demonstrates averages of posterior cluster assignments across varying numbers of samples (100, 250, 500, 1000, 2500) and at neighborhood radii of 1, 2, and 3 grid units. These results demonstrate that when a prior encodes sufficient local spatial dependencies, XenNF accurately recovers cluster assignments that are spatially organized and similar to the ground truth.

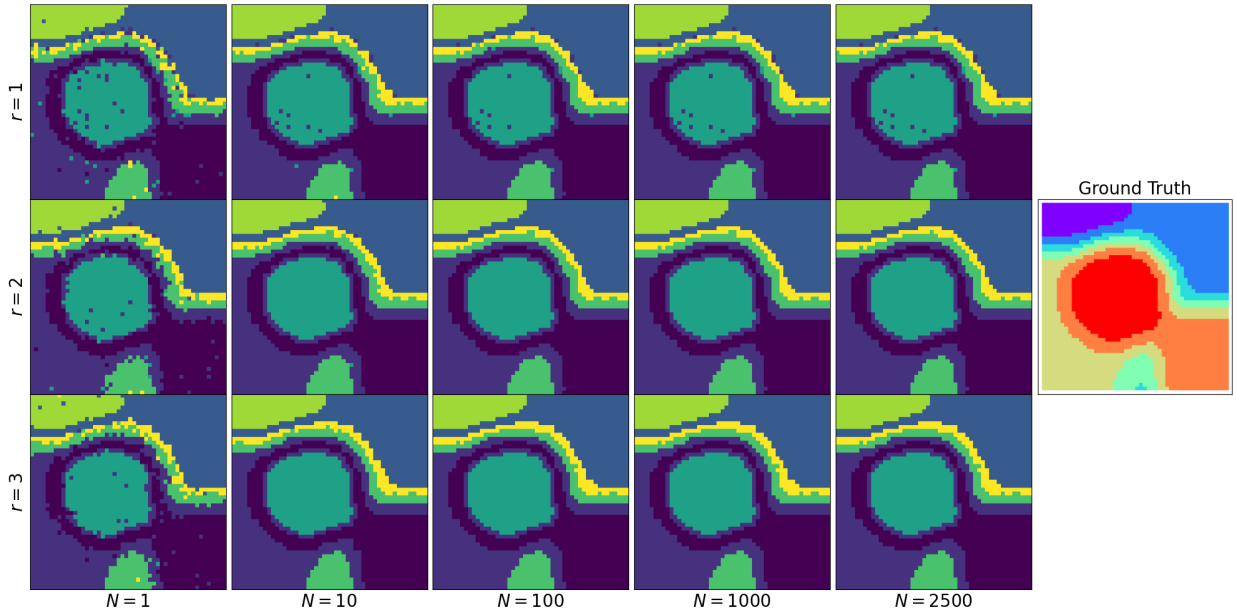
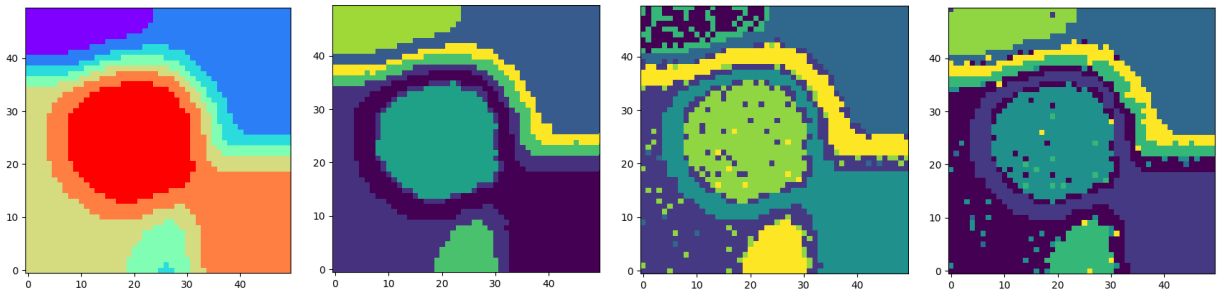


Figure 4.2: Posterior cluster assignments averaged over 1, 10, 100, 1000, and 25000 samples for each of three spatial priors with neighborhood radii of 1, 2, and 3.

We compare the assignments learned by XenNF against industry-standard methods. Just as in Chapter 3 we use the adjusted Rand index (ARI) to compare learned assignments against the ground truth. For additional signal, we also include the normalized mutual information score (NMI). Higher values for the NMI indicate information overlap between the two partitions. We want to ensure that beyond pairwise agreement, predicted clusters remain generally informative about the true labels. Figure 4.3 presents the resulting cluster assignments along with ARI and NMI scores for each method. The general spatial structure

of clusters is apparent in many of the non-spatial competitors, but the assignments are visibly noisy. Despite this, based on expression only, these methods tend to emit a coarse approximation of a spatial structure. Therefore, these clusters contain enough structure to justify constructing empirical priors around them. In practice, the goal is to delineate spatially coherent tissue regions as distinct clusters. XenNF is the only method that manages to have smooth, well-contoured regions that more closely align with ground truth structure, as reflected by its higher ARI and NMI scores.

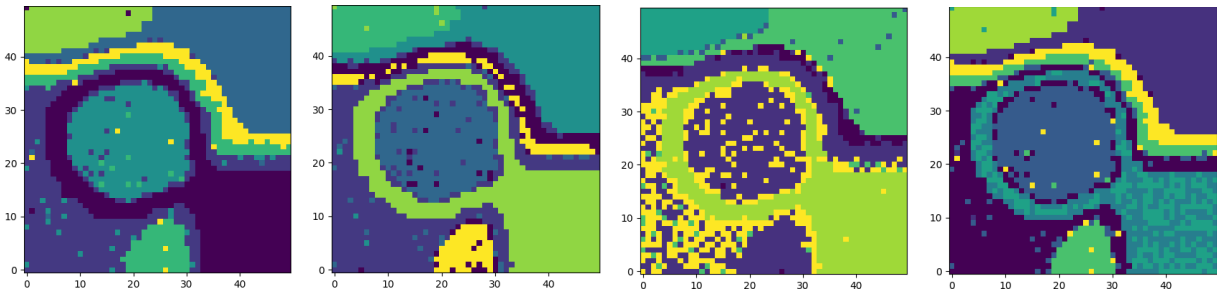


Ground truth
ARI = 1.000,
NMI = 1.000

XenNF (Ours)
ARI = **0.925**,
NMI = **0.927**

K-means
ARI = 0.811,
NMI = 0.815

Leiden
ARI = 0.828,
NMI = 0.834



Louvain
ARI = 0.858,
NMI = 0.834

mclust
ARI = 0.819,
NMI = 0.821

Hierarchical
ARI = 0.586,
NMI = 0.637

Phenograph
ARI = 0.740,
NMI = 0.806

Figure 4.3: DLPFC clustering results across methods. Adjusted Rand Index (ARI) and Normalized Mutual Information (NMI) scores are reported with the best performance in bold.

4.4.2 Dorsolateral Prefrontal Cortex Data

The human dorsolateral prefrontal cortex (DLPFC) data available on the 10x Visium platform (also known as SpatialLIBD) is often considered a keystone benchmark among spatial

clustering methods (Maynard et al., 2021). This dataset includes gene expression profiles for 33,538 genes, derived from two pairs of adjacent 10 μm tissue sections collected from three neurotypical adult donors. The second pair of sections is located 300 μm posterior to the first, resulting in a total of 12 tissue sections. The authors identified DLPFC layers and white matter (WM) through manual annotation, guided by morphological characteristics and key gene markers. Each region represents a spatial domain that uses visible tissue structure and known marker genes to assign each region to a brain layer (a horizontal slice of the cortex where cells are similar and do similar jobs) or white matter (the deeper part of the brain that carries signals between different regions, like wiring). These layers have been shown to have differential expression and spatial separation that is associated with schizophrenia and autism spectrum disorder (Maynard et al., 2021). Visualizing the spatial organization of these annotations explains why this dataset is a canonical benchmark (Figure 4.4). Each layer, despite being concentrically organized, also has unique gene expression profiles, serving as a good application of spatial clustering when we need spatially contiguous domains without compromising cluster assignments based on transcriptomic similarity. We evaluate if XenNF is capable of recovering these domains. Better performance on the DLPFC benchmark serves as evidence that a clustering procedure is spatially aware without compromising biological integrity.

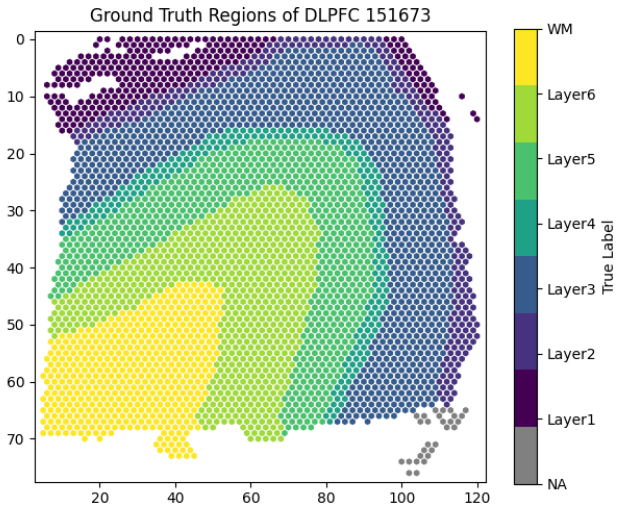
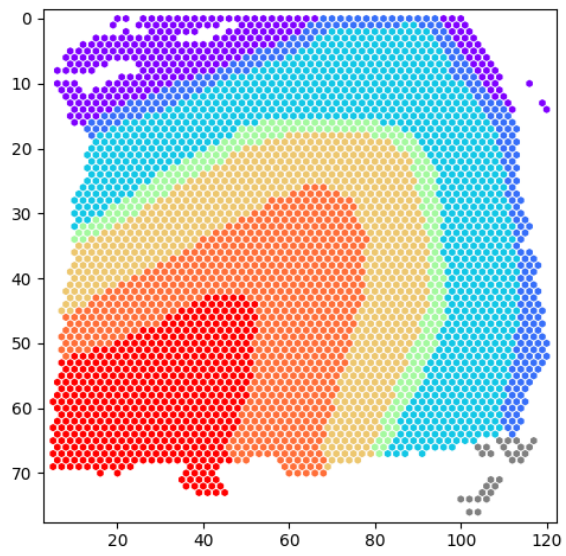


Figure 4.4: Manual annotation of DLPFC sample 151673, illustrating concentric cortical layers and white matter that share contiguous boundaries.

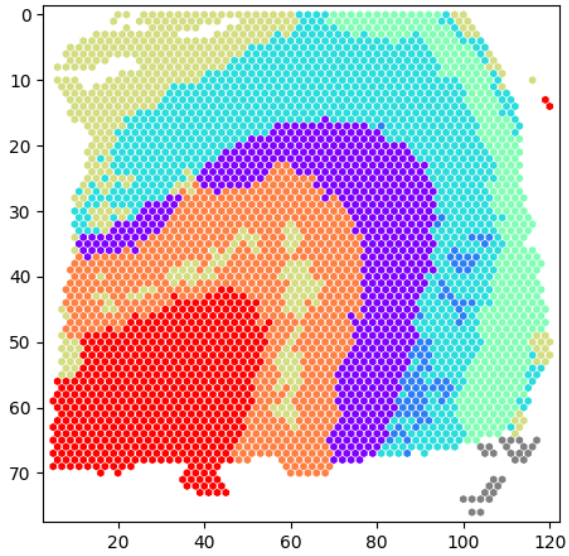
We craft empirical prior distributions for XenNF's cluster mean and scale parameters

based on K-means initialization and train XenNF to convergence. K-means captures some of the concentric nature of the cortical layers but has significant noise in the contiguous spatial structure. XenNF uses the K-means global parameters as a starting point to inform the local cluster structure. Figures 4.5 and 4.6 showcase the performances of several methods commonly used for spatial transcriptomics clustering. XenNF manages to improve the ARI and NMI above its starting point and create smoother assignments. K-means (MacQueen, 1967) and mclust (Scrucca et al., 2023) segment only based on expression; they manage to recover some spatial structure, but assignments remain scattered. Hierarchical clustering (Johnson, 1967) performs especially poorly, likely because the relationship between transcriptomic data and spatial domains is poorly described by agglomerative linkage. Louvain (Blondel et al., 2008), Leiden (Traag et al., 2019), and Phenograph (Levine et al., 2015) require graphical neighborhood structures and a resolution to determine the number of communities that do not at all reflect the underlying spatial structure. XenNF is the only one that effectively balances the spatial coherence and expression-based separation, achieving the highest ARI and NMI values among the methods evaluated.



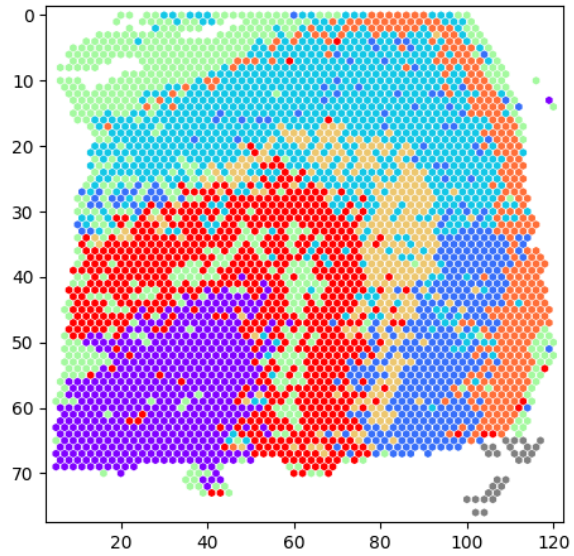
Ground Truth

ARI = 1.000, NMI = 1.000



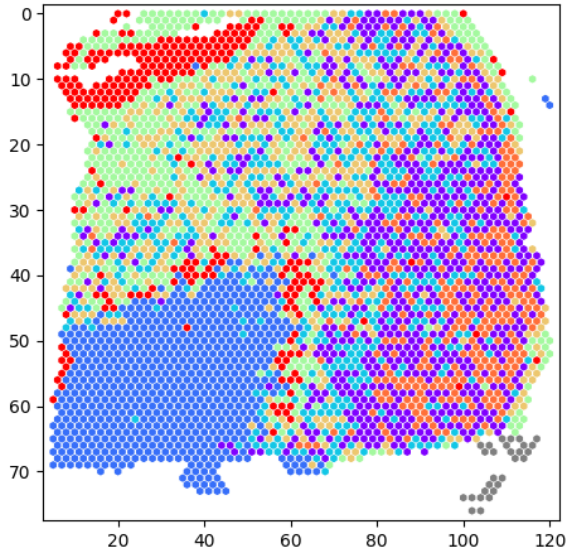
XenNF (ours)

ARI = 0.504, NMI = 0.619



K-means

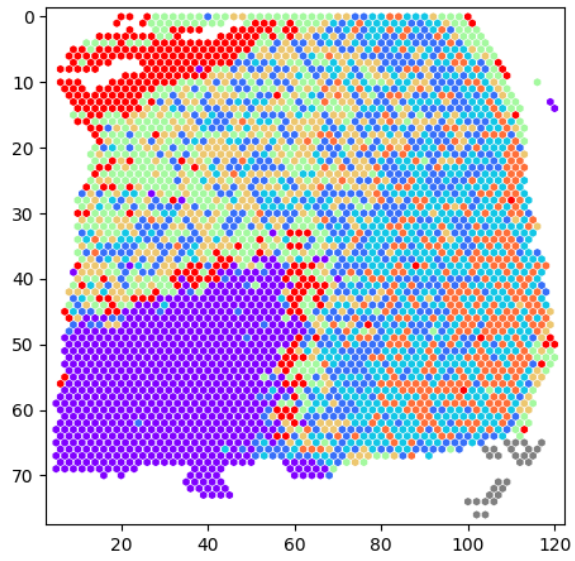
ARI = 0.399, NMI = 0.513



Leiden

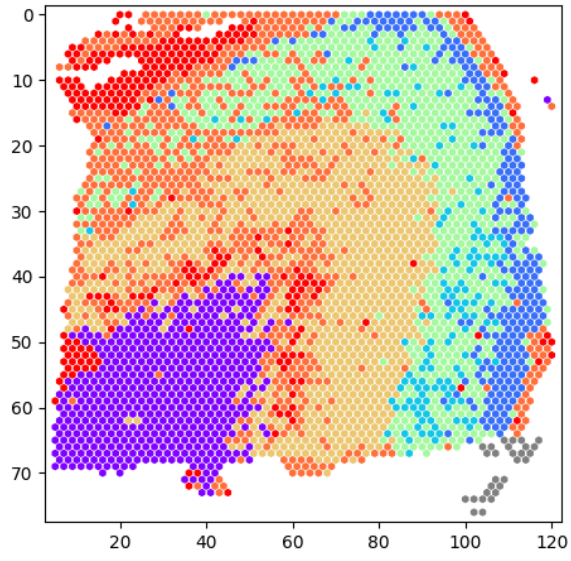
ARI = 0.260, NMI = 0.378

Figure 4.5: Clustering results (part 1) on sample 151673 of the DLPFC dataset. ARI and NMI are shown for each method.



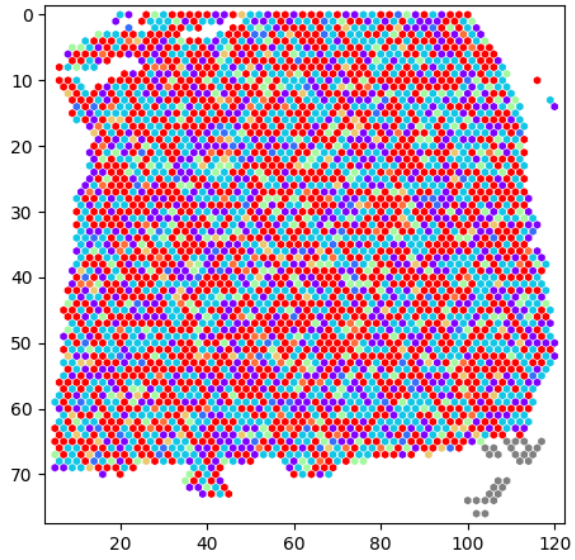
Louvain

ARI = 0.188, NMI = 0.280



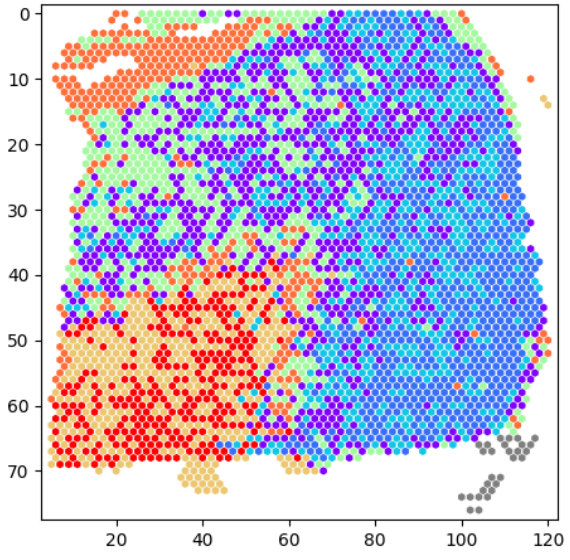
mclust

ARI = 0.401, NMI = 0.534



Hierarchical

ARI = 0.002, NMI = 0.005



Phenograph

ARI = 0.144, NMI = 0.276

Figure 4.6: Clustering results (part 2) on the DLPFC dataset. Remaining methods shown with corresponding ARI and NMI scores.

When comparing the results of XenNF to exact Bayesian methods, however, the results are substantially closer. We refer to the results of prolific Bayesian spatial methods from (Li and Zhou, 2022), (Yang et al., 2021), and (Xu et al., 2024). XenNF performs comparably to

these methods (Table 4.1) but loses the performance margin it had when compared against conventional baselines. We hypothesize that XenNF falls short of strictly outperforming all of these methods because the Potts assumption describes the DLPFC data well; every spot in the ground truth has at least one neighbor in the same class as itself. XenNF does not receive such a stringent modeling assumption, instead opting for learnable weights in a shallow flow to help inform what spatial relationships should define prior distributions. Even so, XenNF achieves the highest ARI on 7 of the 12 DLPFC tissue sections, indicating that approximate Bayesian inference with normalizing flows can match or exceed clustering performance of exact methods, even without enforcing hard spatial constraints like the Potts model, highlighting XenNF's adaptability.

4.5 Discussion

This chapter investigated the potential of normalizing flows as approximate posteriors and spatial priors for clustering spatial transcriptomics data. The true posterior landscape for spatial clustering can have complicated relationships with observed data, motivating the need for flexible conditional distributions. Normalizing flows address this challenge, transforming simple base distributions into flexible, tractable approximate posteriors. In our approach, we placed a continuous-time normalizing flow over the prior and approximate posterior distributions. Using a normalizing flow as an approximate posterior, we learn a more nuanced distribution of cluster logits conditioned on expression data. Furthermore, by modeling the prior over logits using a graph-conditioned CNF, we inherit the tractable spatial dependency that the Potts model allowed in the exact models.

Our experiments on real and synthetic datasets show that normalizing flows can recover true cluster assignments when data exhibits well-defined spatial organization. Furthermore, XenNF achieves significant computational speedups relative to exact posterior sampling approaches. These efficiencies position normalizing flows as scalable and powerful tools for large-scale spatial transcriptomics clustering and analysis.

Despite normalizing flows serving as a valuable avenue for spatial clustering in transcriptomics data, it comes with limitations. Training normalizing flows can be challenging, as their performance is sensitive to the selection of hyperparameters. Beyond selecting hyperparameters for each hypernetwork, the flow length and transformations need to be carefully chosen. These design choices lack definitive, correct values, necessitating brute-force or domain-specific tuning to ensure stable training and meaningful posterior estimates. Additionally, even when the right combination of settings is found, normalizing flows can take tens of thousands of iterations to converge. In practical applications where simpler distri-

Tissue	XenNF	scmeb	BayesSpace	Giotto
151507	**0.43**	0.42	0.33	0.33
151508	**0.46**	0.44	0.36	0.34
151509	0.49	**0.52**	0.44	0.35
151510	**0.55**	0.39	0.43	0.33
151669	0.33	0.32	**0.41**	0.25
151670	0.35	**0.43**	**0.43**	0.21
151671	**0.54**	0.42	0.38	0.40
151672	0.39	0.44	**0.77**	0.38
151673	0.50	0.49	**0.55**	0.37
151674	**0.45**	0.43	0.33	0.29
151675	**0.44**	0.31	0.41	0.32
151676	**0.47**	0.39	0.32	0.26

Table 4.1: ARI scores across DLPFC tissues for exact and approximate Bayesian methods.

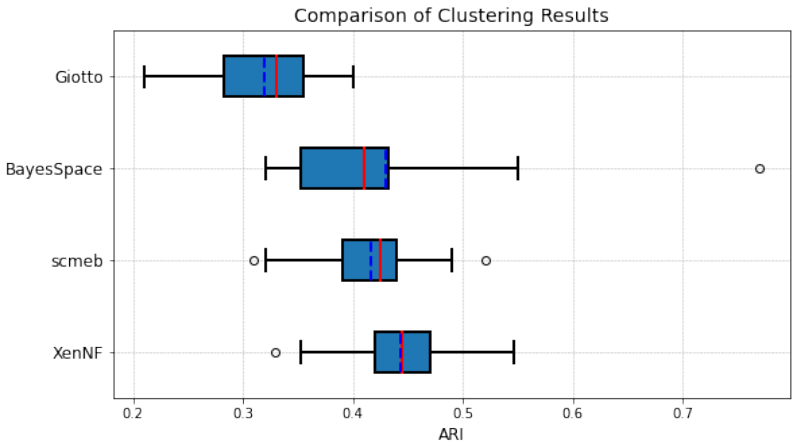


Figure 4.7: Boxplot summary of ARI scores.

Figure 4.8: Clustering performance comparison on DLPFC tissue sections. Mean performance indicated by dashed, blue line. Median performance indicated by solid, red line.

butional families suffice for posterior estimation, this computational overhead can outweigh the benefits. Longer flows with larger networks can be computationally expensive to sample from due to costly forward passes, but unlike Bayesian samplers, flow-based models allow for embarrassingly parallel sampling, enabling faster generation of posterior samples given sufficient computational resources.

Approximate Bayesian inference allows us to explore many model types with prior-informed spatial topologies at scale. For instance, if we believe regions of tissue connected by a bloodstream should also have similar assignments, XenNF can be quickly retrained with a prior using an adjacency matrix that reflects such a dependence.

Having demonstrated that XenNF improves upon fixed, baseline clustering by modeling principal components of transcriptomic data, we can extend XenNF to work on more sophisticated representations for $f(y_i)$. The SEDR model by (Xu et al., 2024) was able to achieve an ARI of 0.684 on the 151673 DLPFC tissue sample by using a joint autoencoder and variational graph autoencoder architecture to learn latents capable of reconstructing both the tissue graph adjacency and gene expression matrix. These learned latents alone were useful for downstream clustering tasks. After training, their clustering procedure does not incorporate any spatial prior. They use mclust on their spatially informed latents to identify the cortical layers. Introducing XenNF as the clustering procedure by using the learned latents as the $f(y_i)$ in the likelihood term could improve spatial organization in final cluster assignments.

XenNF is a successful example for how normalizing flows can support scalable, spatially aware clustering without conjugacy or likelihood assumptions, allowing for exploration of wider and richer model structures.

CHAPTER 5

Conclusions

Through three models, we have demonstrated how to incorporate the spatial context of spatial transcriptomics (ST) datasets—a feature absent in traditional omics datasets—into downstream inference. In Chapter 2, we have shown how using graphs to represent tissue data in place of tabular data yields better predictions of gene expressions and demonstrated why these improvements aid our understanding of cell-cell communication effects. In Chapter 3, we expand the framework of spatial Bayesian clustering for ST data by translating the task from an exact Bayesian inference procedure to an approximate one. This is accomplished by incorporating spatially structured empirical priors and learning approximate posterior distributions over the soft assignments of all spatial regions. In Chapter 4, we increase the flexibility of spatial clustering with approximate Bayesian inference by using normalizing flows to categorize spatial dependencies in the prior and cluster membership weights in the approximate posterior.

In Chapter 2, we introduced a regression model that utilizes graph convolutional neural networks for predicting response gene expressions from ligand-receptor communications. We convert tissue data into graphs by treating cells as nodes, cell-cell communication (CCC) channels as edges, and signaling gene expressions (ligands and receptors) as node attributes. Using these graphs, we more accurately predict gene expressions by propagating and transforming signals from all ligands and receptors across spatial neighborhoods. We also demonstrate that non-graph-based methods systematically fail on real datasets and synthetic settings with nonlinear and distance-weighted spatial influence from CCCs. Our model takes all ligands and receptors as inputs, avoiding the need for predefined ligand-receptor pair features. Notably, this design is both a contribution and a limitation. Not requiring specific ligand-receptor pair information ensures that all potential signals get considered for CCC effects. This is especially helpful in settings where the full list of ligand-receptor pairs is uncertain or we are interested in batch CCC effects. Simultaneously, this limits the causal conclusions we can make about which specific molecule interactions are predictive of a response gene expres-

sion. To this end, we discussed designing an architecture that accepts a multi-view graph and isolates the effects of cell–cell communication into specific ligand–receptor channels. This would allow for cause-and-effect interpretations to be made with our model pipeline, but current computational constraints require filtration of ligand-receptor pairs, which can lead to violation causal inference assumptions.

While Chapter 2 studies gene-level spatial dependence with regression, many practical applications such as domain identification and tissue segmentation require discrete assignments of spatial regions to distinct groups. Thus, in Chapter 3, we shifted to using spatial positions for clustering aimed at discovering coherent spatial domains. Bayesian inference is useful for this task because it allows for uncertainty quantification over probabilistic assignments in the posterior while incorporating a prior that accounts for spatial dependencies. This extends beyond basic smoothing of cluster assignments because posterior samples can correct inaccuracies in the prior when supported by adequate biological evidence. The prominent Bayesian models proposed for this classification task are computationally expensive and typically require sampling techniques, such as Markov chain Monte Carlo or Gibbs sampling, which rely on sufficient mixing to obtain reliable cluster assignments for individual samples. We achieve similar clinical results to these methods even if we switch to an approximate Bayesian inference technique anchored by an empirical spatial prior that encourages neighborhood coherence. The computational expense in training time between exact and approximate Bayesian methods is substantially reduced and decreases even further as data dimensionality increases. In this work, the empirical priors we used to encode spatial dependencies rely on simple spatial neighborhood definitions. Future work could explore biologically motivated adjacency structures informed by known cell-cell signaling or established spatial proximity patterns.

Extending the work in Chapter 3, in Chapter 4 we introduce XenNF, which replaces simple prior and approximate posterior distributions with normalizing flows to model more expressive and multi-modal distributions over cluster memberships. Conventional distributions are often too restrictive to capture the spatial dependencies present in tissue organization. By pairing a shallow spatial prior with a flexible posterior, XenNF enforces spatial coherence without sacrificing representational power. BayXenSmooth serves as an analog to fully Bayesian models that leverage Potts energies as cluster priors, encouraging neighboring spots to share cluster labels, but limited by their computational cost and inflexibility. The use of normalizing flows to parameterize the approximate posterior allows for an exploration of more model families that can be trained to produce approximate posterior samples through amortized inference. To make this approach computationally feasible, however, the posterior cluster distribution over all regions follows a mean-field assumption. Employing flows that

can model the cluster assignments jointly would be a natural next step to more faithfully capturing spatial domains that reflect the underlying biology, but this comes at the cost of a higher computational burden.

Formalizing statistical modeling with expressive models, like the ones we have proposed in this dissertation, advances our ability to statistically model transcriptomic information intertwined with spatial data. These methods contribute to a large set of existing models that aim to leverage the natural positional information present in spatial transcriptomics data. These contributions demonstrate we can use graphs rather than summary statistics to better infer cell-cell communication effects and use faster approximate Bayesian inference in place of exact Bayesian methods to learn spatially contiguous clusters. Although these proposed methods were motivated by biological underpinnings, they are applicable in environments of tiled or graph-based data with informative prior structures. The modeling foundations developed here can be extended or combined with other contributed approaches to yield more interpretable and accurate insights into spatial dependency and organization, a phenomenon central to the established and burgeoning world of spatial transcriptomics.

While we have discussed potential future directions for each chapter's contribution individually, we also posit what may be plausible future directions for statistical inference with spatial transcriptomics data.

For instance, in clinical applications, knowing that a cell-cell communication effect exists or using Bayesian clustering to identify spatial regions that violate prior spatial expectations should be accompanied by downstream analysis that investigates the source or significance of such deviations. In the case of CCC, we may want to know which signaling genes may be driving spatial dependencies in gene expressions. As we discussed in Chapter 2, graph attention networks on multi-view graph inputs would be a natural way to keep the model flexibility that DeepST offers. They would also provide clearer interpretability via attention weights that may identify specific ligand-receptor interactions that drive spatial dependencies in response gene expressions. Similarly, using a graph attention network in place of a graph convolutional network for prior normalizing flows can better describe the spatial dependency structure. Rather than use a broad assumption that neighboring regions should be characterized similarly, attention weights would allow us to learn which spatial neighbors are most informative.

Additionally, our work would benefit from the incorporation of multi-modal inputs. We were motivated in part by modern ST datasets providing additional confidence that we have measured—and therefore observed—the relevant genes that govern CCC influence or characterize spatial domains. However, ST datasets are still prone to measurement noise, low capture rates, dropout, and platform-specific biases. Multi-modal inputs such as histology

images, proximity to vasculature, or extensive metadata can potentially include relevant signals that are undetected from the transcriptome alone. Multi-modal data is also possibly aligned with causal inference motivations. A pivotal assumption of causal inference is no unobserved confounding; if other data modalities contain information related to gene expressions, it is important they are observed. Just as we used modern ST technologies with high resolution and gene coverage, future work can incorporate all available data to reduce the chance of not observing causal variables.

Extending this work to be adaptable in causal inference settings would not only widen the application of these methods, but separate correlation from mechanism, a shift consistent with the goal of making more direct and interpretable biological claims from spatial data.

APPENDIX A

DeepST Additional Model Details

A.1 DeepST Memory, Training and Hardware

As the radius of consideration increases, the number of edges increases at $O(r^2)$. The number of values needed to store a forward pass increases at $O(r^2kd)$ where r is the radius of consideration, k is the kernel size, and d is the pseudo-coordinate dimension. Running deeper and wider models may require splitting the model across multiple GPUs. DeepST models leveraged 3 NVIDIA TU102 cards in the cases that the input graphs were too large to be store on a single GPU card. This was accomplished using PyTorch Lightning's Distributed Data Parallel (DDP) strategy, which partitions each batch across devices and synchronizes gradient updates during training (Falcon et al., 2025).

A.2 Hyperparameter Tuning

DeepST was evaluated for a variety of hyperparameters to identify the best version for generalization. We selected the model with the lowest validation mean squared loss. The winning candidate has the hyperparameters displayed in the tree below.

```
optimizer
    Adam
        Learning Rate = 0.001
        Weight Decay = 1e-6
        Momentum = 0
training
    Epochs = 1200
    Dropout = True
    Early Stopping = True
    Log Transform = True
model
```

- Hidden Layers = (512, 512, 512)
- # of GMM Kernels = 10
- Skip Connections = True
- Batchnorm = False
- Dropout = 0

To ensure a fair comparison against LightGBM and MESSI, we tuned plausible hyperparameters for each model family and evaluated their best-performing versions.

Using LightGBM's gradient boosting type on the regression task evaluated with MSE, the hyperparameters we considered are listed in the tree below. Elements underlined in blue indicate the final hyperparameter chosen for test set evaluation.

- tuning grid
  - Learning Rate = [0.1, 0.01, 0.005, 0.001]
  - n_estimators = [100, 300, 500]
  - max_depth = [3, 5, 7]
  - min_data_in_leaf = [15, 20, 30]

MESSI performed a grid search of their own, and their published source code comes preconfigured with the best-performing hyperparameters for each dataset based on validation error. Each cell type is associated with a specific number of experts, and a weighting scheme (soft or hard) over them. We followed the authors' recommended settings.

A.3 Graph Splitting

The limitation of including additional neighbors for each node is the memory needed to store the entire graph. If cell positions are distributed uniformly at random across the tissue slide, then the number of neighbors increases at rate of $\mathcal{O}(r^d)$, where r is the radius of consideration and d is the number of spatial dimensions. To reduce the memory of each graph, we can divide a graph into smaller but representative subgraphs.

The graph splitting algorithm takes as input the graph generated from the original tissue source. We will refer to this as the source graph. The splitting procedure then outputs 4 graphs that each contain 25% of the cells of the original graph.

Algorithm 2 Graph Splitting (2D Case)

Require: $N \geq 0$ ▷ N: The number of rounds of splits
Require: Graph $\mathcal{G}$ ▷ The source graph
Ensure: List $\mathcal{Q}$ ▷ The split graphs to be used in models

$\mathcal{Q} \leftarrow [\mathcal{G}]$ ▷ $\mathcal{Q}$: Queue of graphs to be split
$\mathcal{Q}_{\text{new}} \leftarrow []$
while $N \neq 0$ **do**
  for all $g \in \mathcal{Q}$ **do**
    $X_0 \leftarrow g((x, y) : x < \text{MEDIAN}(g(x)))$
    $X_1 \leftarrow g((x, y) : x \geq \text{MEDIAN}(g(x)))$
    $X_{00} \leftarrow X_0((x, y) : y < \text{MEDIAN}(X_0(y)))$
    $X_{01} \leftarrow X_0((x, y) : y \geq \text{MEDIAN}(X_0(y)))$
    $X_{10} \leftarrow X_1((x, y) : y < \text{MEDIAN}(X_1(y)))$
    $X_{11} \leftarrow X_1((x, y) : y \geq \text{MEDIAN}(X_1(y)))$
    $\mathcal{Q}_{\text{new}} \leftarrow \mathcal{Q}_{\text{new}} \cup \{X_{00}, X_{10}, X_{01}, X_{11}\}$
  end for
  $\mathcal{Q} \leftarrow \mathcal{Q}_{\text{new}}$
  $\mathcal{Q}_{\text{new}} \leftarrow []$
  $N \leftarrow N - 1$
end while
return $\mathcal{Q}$ ▷ Final split graphs

In the case that there is limited input data, this would be a way to create more training examples for the model to use, though at the cost of some higher dependence between samples generated from the same tissue. Furthermore, the graph splitting ensures model generalization in cases where the data only includes a single tissue, like the Xenium Fresh Frozen Mouse Brain data.

While each graph has fewer communication events considered, the split is done so that the diameter of the split graph does not fall beneath realistic communication distances. Performing a grid search under various radii of consideration, it became clear that utilizing graphs with radii $> \approx 60$ μm leads to overfitting.

Algorithm 3 Graph Splitting (3D Case)

Require: $N \geq 0$ ▷ N: Number of rounds of splits
Require: Graph $\mathcal{G}$ ▷ The source graph
Ensure: List $\mathcal{Q}$ ▷ The split graphs to be used as model inputs

$\mathcal{Q} \leftarrow [\mathcal{G}]$ ▷ Queue of graphs to be split
$\mathcal{Q}_{\text{new}} \leftarrow []$
while $N \neq 0$ **do**
  for all $g \in \mathcal{Q}$ **do**
    $\mathcal{Q} \leftarrow \mathcal{Q}[1:]$
    $X_0 \leftarrow g\{(x, y, z) : x < \text{MEDIAN}(g\{x\})\}$
    $X_1 \leftarrow g\{(x, y, z) : x \geq \text{MEDIAN}(g\{x\})\}$
    $X_{00} \leftarrow X_0\{(x, y, z) : y < \text{MEDIAN}(X_0\{y\})\}$
    $X_{01} \leftarrow X_0\{(x, y, z) : y \geq \text{MEDIAN}(X_0\{y\})\}$
    $X_{10} \leftarrow X_1\{(x, y, z) : y < \text{MEDIAN}(X_1\{y\})\}$
    $X_{11} \leftarrow X_1\{(x, y, z) : y \geq \text{MEDIAN}(X_1\{y\})\}$
    $X_{000} \leftarrow X_{00}\{(x, y, z) : z < \text{MEDIAN}(X_{00}\{z\})\}$
    $X_{001} \leftarrow X_{00}\{(x, y, z) : z \geq \text{MEDIAN}(X_{00}\{z\})\}$
    $X_{010} \leftarrow X_{01}\{(x, y, z) : z < \text{MEDIAN}(X_{01}\{z\})\}$
    $X_{011} \leftarrow X_{01}\{(x, y, z) : z \geq \text{MEDIAN}(X_{01}\{z\})\}$
    $X_{100} \leftarrow X_{10}\{(x, y, z) : z < \text{MEDIAN}(X_{10}\{z\})\}$
    $X_{101} \leftarrow X_{10}\{(x, y, z) : z \geq \text{MEDIAN}(X_{10}\{z\})\}$
    $X_{110} \leftarrow X_{11}\{(x, y, z) : z < \text{MEDIAN}(X_{11}\{z\})\}$
    $X_{111} \leftarrow X_{11}\{(x, y, z) : z \geq \text{MEDIAN}(X_{11}\{z\})\}$
    $\mathcal{Q}_{\text{new}} \leftarrow \mathcal{Q}_{\text{new}} \cup \{X_{000}, X_{001}, X_{010}, X_{011}, X_{100}, X_{101}, X_{110}, X_{111}\}$
  end for
  $\mathcal{Q} \leftarrow \mathcal{Q}_{\text{new}}$
  $\mathcal{Q}_{\text{new}} \leftarrow []$
  $N \leftarrow N - 1$
end while
return $\mathcal{Q}$ ▷ Final split graphs

A.4 Hypothesis Testing Details

In section 2.4.5, we introduced a hypothesis test for identifying spatially dependent genes using DeepST predictions. This appendix section provides complete derivations for the

claims made in section 2.4.5.

A.4.1 Equivalence of MSE Minimization and Gaussian Likelihood Maximization

DeepST is trained by minimizing the mean squared error, defined as

$$\min_{\hat{Y}} \frac{1}{CG} \sum_{c=1}^{C} \sum_{g=1}^{G} (Y_{c,g} - \hat{Y}_{c,g})^2.$$

Assuming each $Y_{c,g}$ follows a Gaussian distribution with fixed variance and that DeepST provides the predicted mean $\hat{Y}_{c,g}$, we show that:

$$\begin{aligned}
\max_{\hat{Y}} p(\boldsymbol{Y} \mid \hat{\boldsymbol{Y}}) &= \max_{\hat{Y}} \prod_{c=1}^{C} \prod_{g=1}^{G} \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{ -\frac{(Y_{c,g} - \hat{Y}_{c,g})^2}{2\sigma^2} \right\} \\
&= \max_{\hat{Y}} \sum_{c=1}^{C} \sum_{g=1}^{G} \left[\log\left(\frac{1}{\sqrt{2\pi}\sigma} \right) - \frac{(Y_{c,g} - \hat{Y}_{c,g})^2}{2\sigma^2} \right] \\
&= \max_{\hat{Y}} \sum_{c=1}^{C} \sum_{g=1}^{G} -\frac{(Y_{c,g} - \hat{Y}_{c,g})^2}{2\sigma^2} \\
&= \max_{\hat{Y}} \frac{1}{CG} \sum_{c=1}^{C} \sum_{g=1}^{G} -(Y_{c,g} - \hat{Y}_{c,g})^2 \\
&= \min_{\hat{Y}} \frac{1}{CG} \sum_{c=1}^{C} \sum_{g=1}^{G} (Y_{c,g} - \hat{Y}_{c,g})^2.
\end{aligned}$$

A.4.2 Connecting LRT to Prediction Error

The test statistic $\Lambda(Y_{c,g})$ can be simplified to a ratio of exponentials:

$$\begin{aligned}
\Lambda_{c,g} &= \frac{(\sqrt{2\pi}\sigma_g)^{-1} \exp\left(-(2\sigma_g^2)^{-1}(Y_{c,g} - \mu_0)^2\right)}{(\sqrt{2\pi}\sigma_g)^{-1} \exp\left(-(2\sigma_g^2)^{-1}(Y_{c,g} - \mu_{r^*})^2\right)} \\
&= \frac{\exp\left(-(2\sigma_g^2)^{-1}(Y_{c,g} - \mu_0)^2\right)}{\exp\left(-(2\sigma_g^2)^{-1}(Y_{c,g} - \mu_{r^*})^2\right)}.
\end{aligned}$$

If we switch to a decision rule in the log space, we still have the decision rule $\log(\Lambda) < \log(c) = c^*$. We can express the log likelihood ratio test as follows:

$$
\begin{aligned}
\log(\Lambda_{c,g}) &= -\frac{1}{2\sigma_g^2}(Y_{c,g} - \mu_0)^2 - \left(-\frac{1}{2\sigma_g^2}(Y_{c,g} - \mu_{r^*})^2\right) \\
&= \frac{1}{2\sigma_g^2}((Y_{c,g} - \mu_{r^*})^2 - (Y_{c,g} - \mu_0)^2) \\
&\propto (Y_{c,g} - \mu_{r^*})^2 - (Y_{c,g} - \mu_0)^2 \\
&= (Y_{c,g}^2 - 2Y_{c,g}\mu_{r^*} + \mu_{r^*}^2 - Y_{c,g}^2 + 2Y_{c,g}\mu_0 - \mu_0^2) \\
&= (2Y_{c,g}(\mu_0 - \mu_{r^*}) + \mu_{r^*}^2 - \mu_0^2).
\end{aligned}
$$

Therefore, we have a likelihood ratio test statistic that can be the difference between spatially aware and spatially ignorant model performances.

A.4.3 Null Distribution of Test Statistic

The test statistic derived from the difference in test MSEs between spatially aware and spatially ignorant models emits a tractable distribution. If H_0 is true, then $\mu_{c,g} = \mu_0$ and $Y_{c,g} \sim N(\mu_0, \sigma_g^2)$. Because the log likelihood ratio test statistic is a composition of affine transformations being applied to $Y_{c,g}$ we can show that it too has a normal distribution:

$$
\begin{aligned}
Y_{c,g} &\sim N(\mu_0, \sigma_g^2) \\
2Y_{c,g} &\sim N(2\mu_0, 4\sigma_g^2) \\
2Y_{c,g}(\mu_0 - \mu_{r^*}) &\sim N\Big(2\mu_0(\mu_0 - \mu_{r^*}), 4\sigma_g^2(\mu_0 - \mu_{r^*})^2\Big) \\
2Y_{c,g}(\mu_0 - \mu_{r^*}) + \mu_{r^*}^2 - \mu_0^2 &\sim N\Big(2\mu_0(\mu_0 - \mu_{r^*}) + \mu_{r^*}^2 - \mu_0^2, 4\sigma_g^2(\mu_0 - \mu_{r^*})^2\Big).
\end{aligned}
$$

Since our test compares two simple hypotheses with fixed means under a shared, known variance, and our test statistic admits a closed-form distribution with a decision rule $\Lambda_{c,g} < c$, the assumptions of the Neyman-Pearson lemma are satisfied. Therefore, we have the most powerful α-level test for detecting spatial dependence in gene expression under the fixed-variance Gaussian likelihood model.

A.4.4 Gene-Level Testing

To assess spatial dependence at the gene level, we consider the average of the likelihood ratio test statistics across cells. Since we have shown that each $\log \Lambda_{c,g}$ is Gaussian, the average

test statistic over all cells C can be written as

$$\bar{T}_g = \frac{1}{C} \sum_{c=1}^{C} \log \Lambda_{c,g}$$

with

$$\bar{T}_g \sim N \left(2\mu_0 \left(\mu_0 - \mu_{r^*} \right) + \mu_{r^*}^2 - \mu_0^2, \frac{4\sigma_g^2 \left(\mu_0 - \mu_{r^*} \right)^2}{C} \right).$$

Under the null hypothesis, we assume each $\mu_{c,g}$ is accurately predicted by a spatially ignorant DeepST. This provides a valid basis for global testing of spatial dependence at the gene level.

A.4.5 Generalization to Unknown Variance

We have demonstrated the role of the MSE in testing differences between two means for a Gaussian distribution with a fixed scale. However, the fixed scale assumption may be erroneous. Even in this case, we can use the MLE estimate for the variance to show that the testing mean square errors of model pairs computes the test statistic. Because the MLE estimate for the variance of a Gaussian distribution is exactly the training mean squared error, if we define l_0 and l_{r^*} as the training MSE of DeepST with $r = 0$ and $r = r^*$, respectively, we can craft an alternate LRT:

$$\Lambda = \frac{(\sqrt{2\pi l_0})^{-1} \exp\left(-(2l_0)^{-1}(Y_{c,g} - \mu_0)^2\right)}{(\sqrt{2\pi l_{r^*}})^{-1} \exp\left(-(2l_{r^*})^{-1}(Y_{c,g} - \mu_{r^*})^2\right)}.$$

Moving to log space we obtain

$$\begin{aligned} \log(\Lambda) &= -\frac{1}{2}\log l_0 - \frac{1}{2l_0}(Y_{c,g} - \mu_0)^2 - \left(-\frac{1}{2}\log l_{r^*} - \frac{1}{2l_{r^*}}(Y_{c,g} - \mu_{r^*})^2 \right) \\ &= \frac{1}{2}(\log l_{r^*} - \log l_0) + \frac{1}{2}\left(\frac{1}{l_{r^*}}(Y_{c,g} - \mu_{r^*})^2 - \frac{1}{l_0}(Y_{c,g} - \mu_0)^2 \right). \end{aligned}$$

Unfortunately, this test statistic does not emit a simple or tractable distribution, but a permutation test can be used to construct an empirical null distribution for hypothesis testing. Still, conclusions about spatial dependence assume accurate values for the test MSE values, so DeepST proves to be a reliable framework for this task, offering improved robustness in spatial dependence inference.

A.5 Code Availability

All implementation of DeepST and accompanying scripts for reproducing real and synthetic experiments are available at https://github.com/prob-ml/spatial.

APPENDIX B

BayXenSmooth Additional Model Details

B.1 BayXenSmooth Model Details

BayXenSmooth has several hyperparameters that can be varied to adjust for larger neighborhoods, cluster classes, spot resolution, data dimensionality, and training stability. We outline the various hyperparameter values that were used. Shared hyperparameters applicable to competing methods were consistently applied to each method.

- Hyperparameters
  - Number of Clusters (K)
    - $K = 5,\ 7,\ 9,\ 17$
  - Radius of Neighbors (r)
    - $r = 1,\ 2,\ 3,\ 4,\ 5$
  - Spot Size (in μm)
    - $25,\ 50,\ 75,\ 100$
  - Data Dimension (d)
    - $d = 3,\ 5,\ 10,\ 15,\ 25$
  - Training
    - Learning Rate
      - lr $= 0.01,\ 0.001,\ 0.0005,\ 0.0001$
    - Epochs = 500, 1000
    - Early Stopping Patience
      - $3,\ 5,\ 10$
    - Prior Variances
      - Mean Prior Variance
        - sd$(\mu_{\theta_\mu}) = 0.1I,\ I,\ 2.5I,\ 5I,\ 10I$
      - Scale Prior Variance
        - sd$(\Sigma_{\theta_\Sigma}) = 0.25I,\ I$
      - Logit Prior Variance (λ)
        - $\lambda = 0.1,\ 1.0,\ 5.0,\ 10.0,\ 25.0$
    - Prior Weight Clamping (ϵ)
      - $\epsilon = 0.001$

B.2 Moran's I Details

To analyze the spatial correlation of genetic expressions and cluster assignments, we used the Moran's I statistic. The following sections outline several weighting strategies that were considered, concluding with the rationale for selecting UMAP-based embeddings as the optimal approach for this study.

B.2.1 Standard

As a trivial baseline, we construct an unweighted adjacency matrix where entries indicate whether pairs of spots belong to the same cluster:

$$w_{ij} = \mathbb{I}(z_i = z_j).$$

This weighting identifies whether spots within the same cluster share similar expression profiles, but it disregards spatial proximity information entirely.

B.2.2 Power-Transformed Inverse Distance

To incorporate some spatial proximity information, we introduce a distance-based decay factor that weights connections between spots inversely to their spatial separation:

$$w_{ij} = \mathbb{I}(z_i = z_j) * \frac{1}{(1 + d(i,j))^{\frac{1}{p}}}.$$

The choice of p controls the influence of distance, with higher values decreasing the decay effect, causing weights for distant neighbors to grow closer to those of nearby neighbors. As $p \to \infty$, all neighbors contribute equally to the network. However, tuning p adds an extra layer of hyperparameter optimization with a challenging interpretative basis.

B.2.3 Gaussian

A commonly used spatial weighting matrix for the Moran's I is the Gaussian kernel:

$$w_{ij} = \mathbb{I}(z_i = z_j) * \exp\left(-\frac{d(i,j)^2}{2\sigma^2}\right).$$

With a globally defined σ^2, the kernel down-weights connections between spots as their distance increases. However, choosing an appropriate σ^2 remains a challenging aspect of model tuning, as it determines the scale of spatial influence for all pairwise spot relationships.

B.2.4 UMAP

The weighting mechanism employed in our analysis is adapted from the similarity graph construction used in the UMAP algorithm (McInnes et al., 2020).

$$w_{ij} = \mathbb{I}(z_i = z_j) w_{ij}^{\text{sym}}$$

where the symmetric weight w_{ij} is calculated via the UMAP weighting formulas. They can be described as follows:

1. $\rho_i := \min(d(i,j) \mid d(i,j) > 0)$
2. σ_i is defined as the solution to $\log_2(k) = \sum_j \exp\left(-\frac{d(i,j)-\rho_i}{\sigma_i}\right)$.
3. $\kappa(i,j) = \exp\left(-\frac{d(i,j)-\rho_i}{\sigma_i}\right)$
4. $w_{ij}^{\text{sym}} = \kappa(i,j) + \kappa(k,i) - \kappa(i,j) * \kappa(j,i)$.

This method has an advantage over the Gaussian weighting due to having a local σ_i instead of a global σ. This localization allows for distances to in spots to be weighted based on a local definition of proximity instead of a global one. This adaptation enables weights to adjust based on neighborhood-specific density, rather than applying a uniform scaling across all distances.

B.3 BayXenSmooth ELBO Derivation and Optimization

Given the ELBO definition:

$$\text{ELBO} := \mathbb{E}_{q_\phi(l,\mu,\sigma)} \left[\log p_\theta(y|l,\mu,\sigma) + \log p_\theta(l,\mu,\sigma) - \log q_\phi(l,\mu,\sigma)\right]$$

we can use in conjunction the linearity of expectation and the fact that all model and variational parameters are independent of one another to write

$$
\begin{aligned}
\text{ELBO} &:= \mathbb{E}_{q_\phi(l,\mu,\sigma)}\left[\log p_\theta(y|l,\mu,\sigma) + \log p_\theta(l,\mu,\sigma) - \log q_\phi(l,\mu,\sigma)\right] \\
&= \mathbb{E}_{q_{\phi_l}(l)q_{\phi_\mu}(\mu)q_{\phi_\sigma}(\sigma)}\big[\log p_\theta(y|l,\mu,\sigma) + \log p_\theta(l) + \log p_\theta(\mu) + \log p_\theta(\sigma) \\
&\quad - \log q_{\phi_l}(l) - \log q_{\phi_\mu}(\mu) - \log q_{\phi_\sigma}(\sigma)\big] \\
&= E_q\left[\log p_\theta(y|l,\mu,\sigma)\right] + E_q\left[\log p_\theta(l)\right] + E_q\left[\log p_\theta(\mu)\right] + E_q\left[\log p_\theta(\sigma)\right] \\
&\quad - E_q\left[\log q_{\phi_l}(l)\right] - E_q\left[\log q_{\phi_\mu}(\mu)\right] - E_q\left[\log q_{\phi_\sigma}(\sigma)\right] \\
&= E_q\left[\log p_\theta(y|l,\mu,\sigma)\right] + E_q\left[\log p_\theta(l)\right] + \sum_{k=1}^{K} E_q\left[\log p_\theta(\mu_k)\right] + \sum_{k=1}^{K} E_q\left[\log p_\theta(\sigma_k)\right] \\
&\quad - E_q\left[\log q_{\phi_l}(l)\right] - \sum_{k=1}^{K} E_q\left[\log q_{\phi_\mu}(\mu_k)\right] - \sum_{k=1}^{K} E_q\left[\log q_{\phi_\sigma}(\sigma_k)\right].
\end{aligned}
$$

For simplicity of notation, we define $q := q_\phi(l,\mu,\sigma) = q_{\phi_l}(l)q_{\phi_\mu}(\mu)q_{\phi_\sigma}(\sigma)$. Plugging in the densities for each term in the summation, we achieve the following simplifications. Note that the final R.H.S. will treat as constant any term not dependent on $\{\theta,\phi\}$. Any term denoted with the superscript $*$ means it is a deterministic hyperparameter.

$$
\begin{aligned}
E_q\left[\log p_\theta(y|l,\boldsymbol{\mu},\boldsymbol{\sigma})\right] &= E_q\Bigg[\log\Bigg(\sum_{k=1}^{K} w_k(2\pi)^{-d/2}(\det\Sigma_k)^{-1/2} \\
&\qquad \times \exp\left(-\frac{1}{2}(y-\mu_k)^\top\Sigma_k^{-1}(y-\mu_k)\right)\Bigg)\Bigg] \\
&= \log\left(\sum_{k=1}^{K} w_k(2\pi)^{-d/2}(\det\Sigma_k)^{-1/2}\exp\left(-\frac{1}{2}(y-\mu_k)^\top{\Sigma_k}^{-1}(y-\mu_k)\right)\right)
\end{aligned}
$$

$$
\begin{aligned}
E_q\left[\log p_\theta(\mu_k)\right] &= E_q\Big[\log\Big((2\pi)^{-d/2}(\det\Sigma^*_{\theta_{\mu_k}})^{-1/2}\exp\big(-\tfrac{1}{2}(\mu_k-\mu_{\theta_{\mu_k}})^\top{\Sigma^*_{\theta_{\mu_k}}}^{-1}(\mu_k-\mu_{\theta_{\mu_k}})\big)\Big)\Big] \\
&= E_q\left[\log(2\pi)^{-d/2}\right] + E_q\left[\log(\det\Sigma^*_{\theta_{\mu_k}})^{-1/2}\right] \\
&\quad - E_q\left[\tfrac{1}{2}(\mu_k-\mu_{\theta_{\mu_k}})^\top\Sigma^{*-1}_{\theta_{\mu_k}}(\mu_k-\mu_{\theta_{\mu_k}})\right] \\
&= -\tfrac{1}{2}(\mu_k-\mu_{\theta_{\mu_k}})^\top\Sigma^{*-1}_{\theta_{\mu_k}}(\mu_k-\mu_{\theta_{\mu_k}}) + C_{p_\theta(\mu_k)}
\end{aligned}
$$

$$
\begin{aligned}
E_q\left[\log p_\theta(\sigma_k)\right] &= E_q\left[\log\left((2\pi)^{-\frac{d}{2}}(\det\tau^*_{\theta_{\Sigma_k}})^{-\frac{1}{2}}\prod_{j=1}^{d}\sigma_{k,j}^{-1}\right.\right.\\
&\qquad\left.\left.\times\exp\left(-\frac{1}{2}(\log(\sigma_k)-\sigma_{\theta_{\Sigma_k}})^T{\tau^*_{\theta_{\Sigma_k}}}^{-1}(\log(\sigma_k)-\sigma_{\theta_{\Sigma_k}})\right)\right)\right]\\
&= E_q\left[\log(2\pi)^{-d/2}\right]+E_q\left[\log(\det\tau^*_{\theta_{\Sigma_k}})^{-1/2}\right]\\
&\qquad+E_q\left[\sum_{j=1}^{d}\log\sigma_{k,j}^{-1}\right]-E_q\left[\frac{1}{2}(\log(\sigma_k)-\sigma_{\theta_{\Sigma_k}})^T{\tau^*_{\theta_{\Sigma_k}}}^{-1}(\log(\sigma_k)-\sigma_{\theta_{\Sigma_k}})\right]\\
&= -\frac{1}{2}(\log(\sigma_k)-\sigma_{\theta_{\Sigma_k}})^T{\tau^*_{\theta_{\Sigma_k}}}^{-1}(\log(\sigma_k)-\sigma_{\theta_{\Sigma_k}})+\sum_{j=1}^{d}\log\sigma_{k,j}^{-1}+C_{p_\theta(\Sigma_k)}
\end{aligned}
$$

$$
\begin{aligned}
E_q\left[\log p_\theta(l)\right] &= E_q\left[\log\left((2\pi)^{-d/2}(\det(\lambda\mathbf{I}))^{-1/2}\exp\left(-\frac{1}{2}(l-l_{\theta_l})^T(\lambda\mathbf{I})^{-1}(l-l_{\theta_l})\right)\right)\right]\\
&= E_q\left[\log(2\pi)^{-d/2}\right]+E_q\left[\log(\det(\lambda\mathbf{I}))^{-1/2})\right]-E_q\left[\frac{1}{2}(l-l_{\theta_l})^T(\lambda\mathbf{I})^{-1}(l-l_{\theta_l})\right]\\
&= -\frac{1}{2}(l-l_{\theta_l})^T(\lambda\mathbf{I})^{-1}(l-l_{\theta_l})+C_{p_\theta(l)}
\end{aligned}
$$

$$
\begin{aligned}
E_q\left[\log q_\phi(\mu_k)\right] &= E_q\left[\log\left((2\pi)^{-d/2}(\det\Sigma_{\phi_{\mu_k}})^{-1/2}\exp(-\frac{1}{2}(\mu_k-\mu_{\phi_{\mu_k}})^T{\Sigma_{\phi_{\mu_k}}}^{-1}(\mu_k-\mu_{\phi_{\mu_k}})\right)\right]\\
&= E_q\left[\log(2\pi)^{-d/2}\right]+E_q\left[\log(\det\Sigma_{\phi_{\mu_k}})^{-1/2}\right]\\
&\qquad-E_q\left[\frac{1}{2}(\mu_k-\mu_{\phi_{\mu_k}})^T{\Sigma_{\phi_{\mu_k}}}^{-1}(\mu_k-\mu_{\phi_{\mu_k}})\right]\\
&= E_q\left[-\frac{1}{2}\log(\det\Sigma_{\phi_{\mu_k}})-\frac{1}{2}(\mu_k-\mu_{\phi_{\mu_k}})^T{\Sigma_{\phi_{\mu_k}}}^{-1}(\mu_k-\mu_{\phi_{\mu_k}})\right]+C_{q_\phi(\mu_k)}
\end{aligned}
$$

$$
\begin{aligned}
E_q\left[\log q_\phi(\sigma_k)\right] &= E_q\Bigg[\log\Bigg((2\pi)^{-d/2}(\det\tau_{\phi_{\Sigma_k}})^{-1/2}\prod_{j=1}^{d}\sigma_{k,j}^{-1} \\
&\qquad \times \exp\left(-\frac{1}{2}(\log(\sigma_k)-\sigma_{\phi_{\Sigma_k}})^\top\tau_{\phi_{\Sigma_k}}^{-1}(\log(\sigma_k)-\sigma_{\phi_{\Sigma_k}})\right)\Bigg)\Bigg] \\
&= E_q\left[\log(2\pi)^{-d/2}\right] + E_q\left[\log(\det\tau_{\phi_{\Sigma_k}})^{-1/2}\right] \\
&\qquad + E_q\left[\sum_{j=1}^{d}\log\sigma_{k,j}^{-1}\right] - E_q\left[\frac{1}{2}(\log(\sigma_k)-\sigma_{\phi_{\Sigma_k}})^\top\tau_{\phi_{\Sigma_k}}^{-1}(\log(\sigma_k)-\sigma_{\phi_{\Sigma_k}})\right] \\
&= E_q\Bigg[-\frac{1}{2}\log(\det\tau_{\phi_{\Sigma_k}}) + \sum_{j=1}^{d}\log\sigma_{k,j}^{-1} \\
&\qquad -\frac{1}{2}(\log(\sigma_k)-\sigma_{\phi_{\Sigma_k}})^\top\tau_{\phi_{\Sigma_k}}^{-1}(\log(\sigma_k)-\sigma_{\phi_{\Sigma_k}})\Bigg] + C_{q_\phi(\Sigma_k)}
\end{aligned}
$$

$$
\begin{aligned}
E_q\left[\log q_\phi(l)\right] &= E_q\left[\log\left((2\pi)^{-d/2}(\det\gamma_{\phi_l})^{-1/2}\exp\left(-\frac{1}{2}(l-l_{\phi_l})^T\gamma_{\phi_l}^{-1}(l-l_{\phi_l})\right)\right)\right] \\
&= E_q\left[\log(2\pi)^{-d/2}\right] + E_q\left[\log(\det\gamma_{\phi_l})^{-1/2})\right] - E_q\left[\frac{1}{2}(l-l_{\phi_l})^T\gamma_{\phi_l}^{-1}(l-l_{\phi_l})\right] \\
&= E_q\left[-\frac{1}{2}\log(\det\gamma_{\phi_l}) - \frac{1}{2}(l-l_{\phi_l})^T\gamma_{\phi_l}^{-1}(l-l_{\phi_l})\right] + C_{q_\phi(l)}
\end{aligned}
$$

B.4 Proof of BayXenSmooth Reparameterization Trick Viability

To maximize the ELBO (eq. 3.6), we apply gradient steps to model and variational parameters. Because model parameters are independent of variational parameters, we can pass gradients through the expectations.

$$
\nabla_{\mu_{\theta_{\mu_k}}}\text{ELBO} = \nabla_{\mu_{\theta_{\mu_k}}}\left[-\frac{1}{2}(\mu_k-\mu_{\theta_{\mu_k}})^T{\Sigma^*_{\theta_{\mu_k}}}^{-1}(\mu_k-\mu_{\theta_{\mu_k}})\right] = {\Sigma^*_{\theta_{\mu_k}}}^{-1}(\mu_k-\mu_{\theta_{\mu_k}})
$$

$$\nabla_{\Sigma_{\theta_{\Sigma_k}}} \text{ELBO} = \nabla_{\Sigma_{\theta_{\Sigma_k}}} \left[-\frac{1}{2} (\log(\sigma_k) - \Sigma_{\theta_{\Sigma_k}})^T {\tau^*_{\theta_{\Sigma_k}}}^{-1} (\log(\sigma_k) - \Sigma_{\theta_{\Sigma_k}}) \right]$$
$$= {\tau^*_{\theta_{\Sigma_k}}}^{-1} (\log(\sigma_k) - \Sigma_{\theta_{\Sigma_k}})$$

$$\nabla_{l_{\theta_l}} \text{ELBO} = \nabla_{l_{\theta_l}} \left[-\frac{1}{2} (l - l_{\theta_l})^T {\gamma^*_{\theta_l}}^{-1} (l - l_{\theta_l}) \right] = {\gamma^*_{\theta_l}}^{-1} (l - l_{\theta_l})$$

Unlike the model parameters, the variational parameters parameterize both the distribution and the function under the expectation. In these cases we make use of the reparameterization trick. Allowing $\epsilon \in \mathbb{R}^d \sim N(0, I)$, we can rewrite parameters as

$$\mu_k = \mu_{\phi_{\mu_k}} + \Sigma^{1/2}_{\phi_{\mu_k}} \epsilon$$

$$\sigma_k = \sigma_{\phi_{\Sigma_k}} + \tau^{1/2}_{\phi_{\Sigma_k}} \epsilon$$

$$l = l_{\phi_l} + \gamma^{1/2}_{\phi_l} \epsilon.$$

Note that because we treat each dimension in the likelihood independently, we can write $\Sigma_k = \text{diag}(\sigma_k)$ where σ_k is a vector of the diagonal elements of Σ_k. We denote the reparameterized distributions as q_ϵ. Gradient estimates are then computed by averaging Monte Carlo samples of ϵ plugged into gradient formulas. In the setting where we learn MAP estimates of global parameters, the variational scales $(\Sigma_{\phi_{\mu_k}}, \tau_{\phi_{\Sigma_k}})$ collapse to $\mathbf{0}$, simplifying computation. We utilize vector and matrix derivatives without proof using definitions provided in (Petersen and Pedersen, 2012).

To optimize the ELBO, we take gradient steps for all variational parameters and employ early stopping based on the ELBO value. The algorithm returns the optimized model and variational parameters θ^*, ϕ^*. The full training algorithm is provided in Algorithm 4.

Algorithm 4 BayXenSmooth ELBO Optimization

Require: Observed data $\mathbf{y}$, model params θ, variational params ϕ, learning rate η, patience T

Ensure: Optimized params θ^*, ϕ^*

Initialize $\theta^{(0)}, \phi^{(0)}, \mathcal{L}^* \leftarrow -\infty$

Define the ELBO:

$$\mathcal{L}(\theta, \phi) = \mathbb{E}_{q_\phi(w,\mu,\Sigma)} \left[\log p_\theta(y \mid w, \mu, \Sigma) + \log p_\theta(w, \mu, \Sigma) - \log q_\phi(w, \mu, \Sigma)\right]$$

while $T > 0$ **do**

  for $i = 1 \to n$ **do**

    for $k = 1 \to K$ **do**

      Sample $\mu_k^{(i)} \sim \mathcal{N}(\mu_{\phi_{\mu_k}}, \Sigma_{\phi_{\mu_k}})$, $\log \operatorname{diag}(\Sigma_k)^{(i)} \sim \mathcal{N}(\sigma_{\phi_{\Sigma_k}}, \tau_{\phi_{\Sigma_k}})$, $l^{(i)} \sim \mathcal{N}(l_{\phi_l}, \gamma_{\phi_l})$

    end for

  end for

Compute estimated ELBO:

$$\hat{\mathcal{L}} = \tfrac{1}{n} \textstyle\sum_{i=1}^{n} \big[\log p_\theta(y \mid \operatorname{softmax}(l^{(i)}), \mu^{(i)}, \Sigma^{(i)}) + \log p_\theta(\operatorname{softmax}(l^{(i)}), \mu^{(i)}, \Sigma^{(i)}) - \log q_\phi(\operatorname{softmax}(l^{(i)}), \mu^{(i)}, \Sigma^{(i)})\big]$$

  if $\hat{\mathcal{L}} > \mathcal{L}^*$ **then**

    $\mathcal{L}^* \leftarrow \hat{\mathcal{L}}$

  else

    $T \leftarrow T - 1$

  end if

  for $k = 1 \to K$ **do**

    $\mu_{\theta_{\mu_k}} \leftarrow \mu_{\theta_{\mu_k}} + \eta \nabla_{\mu_{\theta_{\mu_k}}} \hat{\mathcal{L}}$

    $\Sigma_{\theta_{\Sigma_k}} \leftarrow \Sigma_{\theta_{\Sigma_k}} + \eta \nabla_{\Sigma_{\theta_{\Sigma_k}}} \hat{\mathcal{L}}$

    $\mu_{\phi_{\mu_k}} \leftarrow \mu_{\phi_{\mu_k}} + \eta \nabla_{\mu_{\phi_{\mu_k}}} \hat{\mathcal{L}}$

    $\Sigma_{\phi_{\mu_k}} \leftarrow \Sigma_{\phi_{\mu_k}} + \eta \nabla_{\Sigma_{\phi_{\mu_k}}} \hat{\mathcal{L}}$

    $\Sigma_{\phi_{\Sigma_k}} \leftarrow \Sigma_{\phi_{\Sigma_k}} + \eta \nabla_{\Sigma_{\phi_{\Sigma_k}}} \hat{\mathcal{L}}$

    $\tau_{\phi_{\Sigma_k}} \leftarrow \tau_{\phi_{\Sigma_k}} + \eta \nabla_{\tau_{\phi_{\Sigma_k}}} \hat{\mathcal{L}}$

  end for

  $l_{\theta_l} \leftarrow l_{\theta_l} + \eta \nabla_{l_{\theta_l}} \hat{\mathcal{L}}$

  $l_{\phi_l} \leftarrow l_{\phi_l} + \eta \nabla_{l_{\phi_l}} \hat{\mathcal{L}}$

  $\gamma_{\phi_l} \leftarrow \gamma_{\phi_l} + \eta \nabla_{\gamma_{\phi_l}} \hat{\mathcal{L}}$

end while

return θ^*, ϕ^*

B.5 Additional Posterior Inference Results

B.5.1 Posterior Means and Scales

BayXenSmooth primarily assigns clusters, but assessing the posterior means and scales ensures they align with inferred memberships. The posterior soft assignments represent a membership each spot has to each cluster class whether it corresponds to a cell type, tissue domain, or other biologically relevant structure. Therefore, we would hope that the means and scales reflect the inferred membership probabilities.

For this evaluation, we use the synthetic dataset from Chapter 2 and initialize with cluster assignments from mclust, adopting its inferred means and scales. BayXenSmooth is sensitive to which cluster method is used for initialization because the empirical priors used for variational inference can be distant from the true means, but VI penalizes the approximate posterior for diverging too far from the prior. If the empirical prior is poorly specified, it can degrade model performance.

We assume the priors are informative, meaning they should be close to the initialization estimates. If the initialization method yields means and scales that significantly deviate from the true values, constructing priors around it may provide little benefit and could hinder posterior inference.

Figures B.1 and B.2 represent the learned posterior means and scales for each cluster distribution. As expected with a sharp, informed prior, the posterior means tend to stay close to the empirical prior. This verifies that when the empirical prior serves as a well-posed initial estimate, the posterior maintains it rather than overwriting it. Preserving accurate means and scales is critical because, in the next section, we show that BayXenSmooth's posterior soft assignments provide a meaningful refinement of cluster memberships, improving upon the empirical prior's soft assignments.

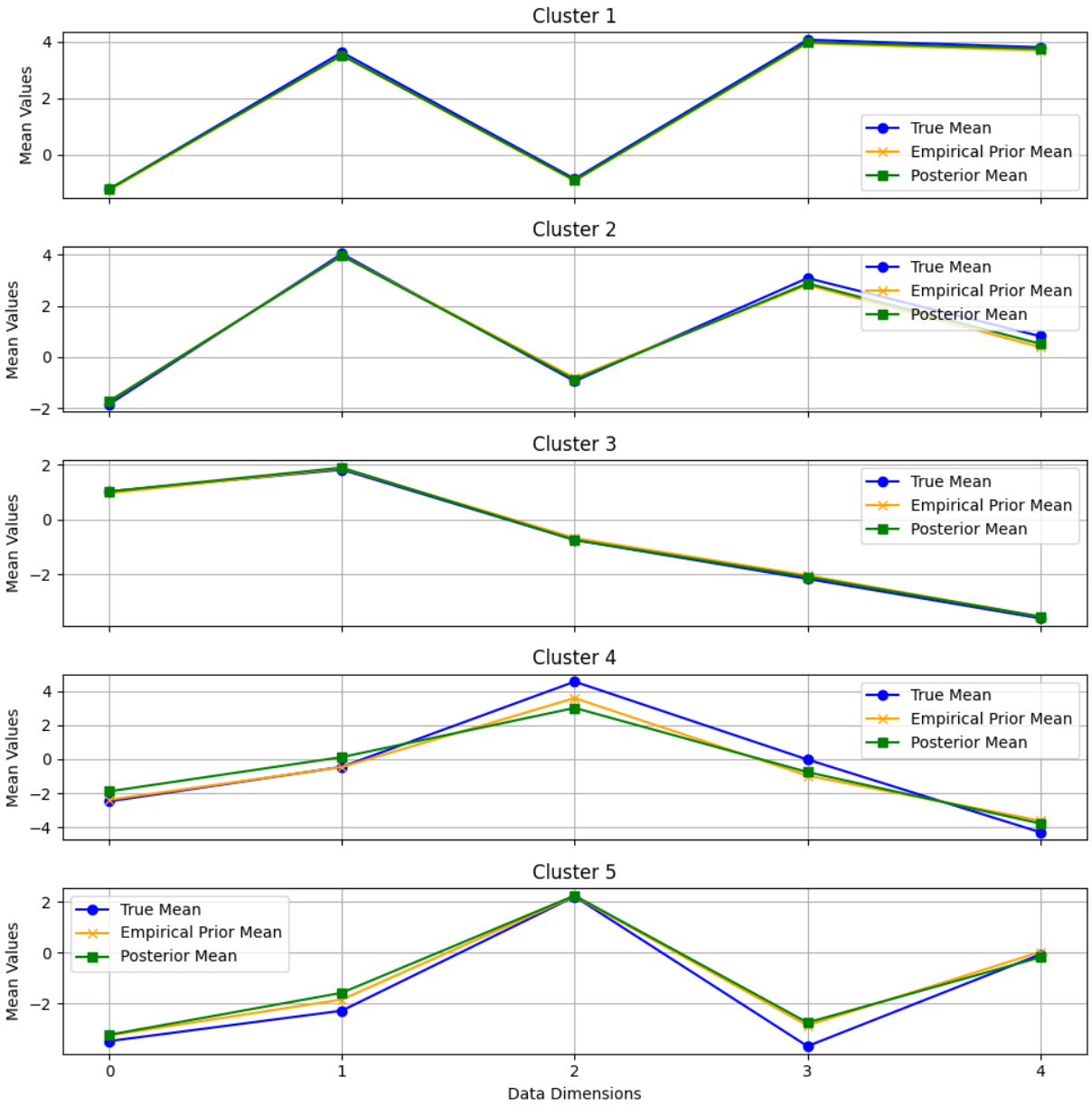


Figure B.1: True, empirical prior, and approximate posterior mean parameters for each cluster. The Hungarian algorithm was used to align clusters and avoid label switching.

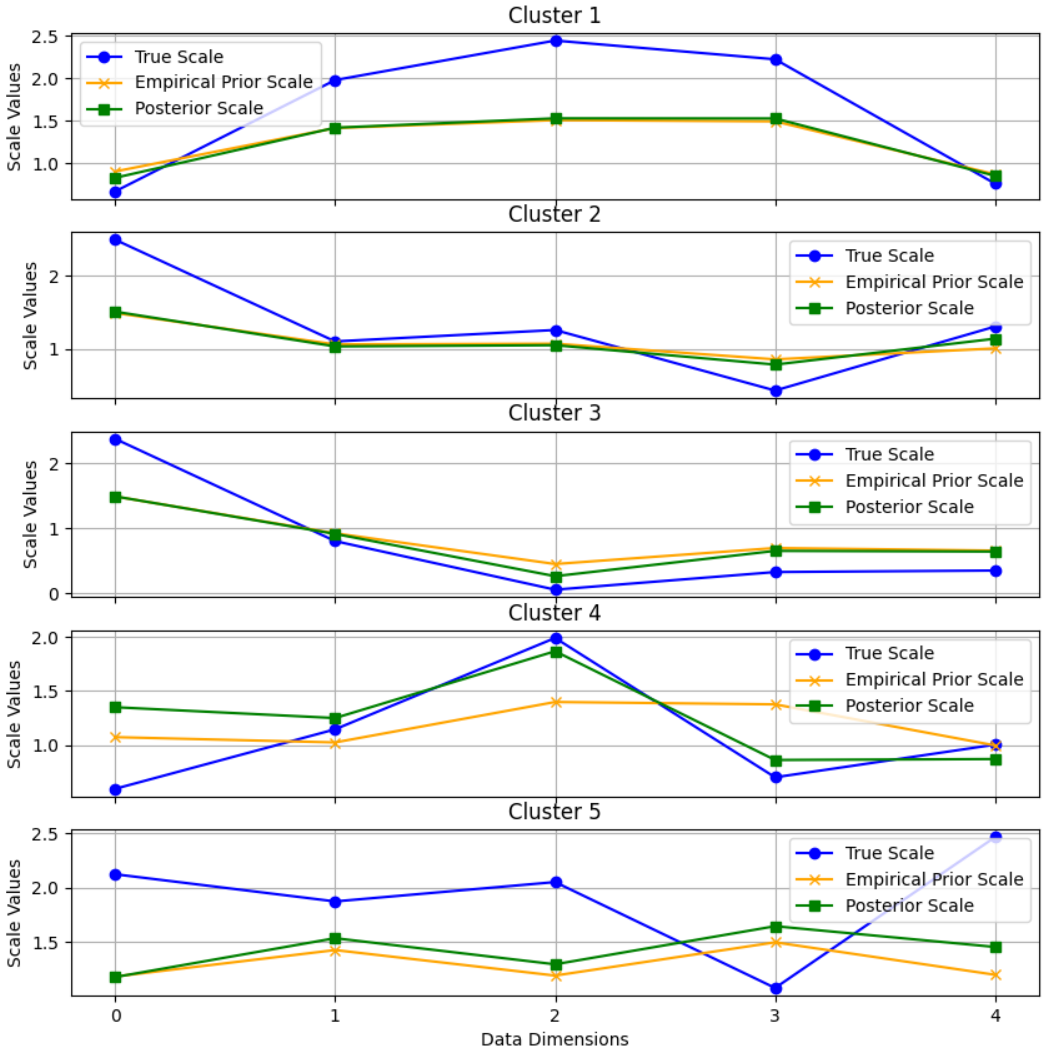


Figure B.2: True, empirical prior, and approximate posterior scale parameters for each cluster. The Hungarian algorithm was used to align clusters and avoid label switching.

B.6 Code Availability

All implementation of BayXenSmooth and accompanying scripts for reproducing real and synthetic experiments are available at https://github.com/prob-ml/XeniumCluster.

APPENDIX C

XenNF Additional Model Details

C.1 Normalizing Flow Architecture Details

C.1.1 Masked Autoregressive Flows

The Masked Autoregressive Flow (MAF) is a prominent example of an expressive flow that emits a tractable distribution. The MAF models conditionals parameterized as Gaussians

$$p\left(x_i \mid \mathbf{x}_{1:i-1}\right) = \mathcal{N}\left(x_i \mid \mu_i, \left(\exp \alpha_i\right)^2\right)$$

where each Gaussian mean μ_i and log standard deviation α_i is given by the flows

$$\mu_i = f_{\mu_i}\left(\mathbf{x}_{1:i-1}\right), \ \alpha_i = f_{\alpha_i}\left(\mathbf{x}_{1:i-1}\right).$$

The autoregressive nature of this flow ensures that the Jacobian is triangular and therefore its determinant can be calculated via a summation:

$$\left|\det\left(\frac{\partial f^{-1}}{\partial \mathbf{x}}\right)\right| = \exp\left(-\sum_i \alpha_i\right).$$

The product of these conditionals yields the target joint distribution being modeled: $p(\boldsymbol{x}) = p(x_1, x_2, \ldots, x_D)$. The combination of the structure's simple inverse and easily computable Jacobian determinant makes it a computationally scalable transform. Indeed, using neural networks as hypernetworks to generate the parameters of each conditional transformation is an effective design choice because they are universal approximators that when used with autoregressive masking define transformations with triangular Jacobians. Furthermore, these transformations can be stacked together to learn multi-modal conditionals. In implementation, the MAF is a stacking of Masked Autoencoders for Distribution Estimation (MADE). Its forward passes are computationally efficient due to a binary masking procedure that allows us to use a fully connected layer to represent a sequential transformation

calculation. By creating an autoregressive binary mask, we can get all dimensional outputs that satisfy the autoregressive property. If we write out the autoregressive mask M as

$$M = \begin{array}{c} \begin{matrix} x_1 & x_2 & x_3 & \cdots & x_d \end{matrix} \\ \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 0 & \cdots & 0 \\ 1 & 1 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \cdots & 1 \end{bmatrix} \end{array} \begin{matrix} f_1 \\ f_2 \\ f_3 \\ \vdots \\ f_d \end{matrix}$$

and allow W_μ and W_α to describe the weights of a layer of the underlying hypernetwork that emits the values used in each conditional transformation, then a layer of the MADE can be written as the equation pair

$$\mu = \sigma((M \odot W_\mu)x)$$

$$\alpha = \sigma((M \odot W_\alpha)x).$$

The stacking of these layers composes a MADE and the composition of MADE blocks defines the MAF. Forward passes through these layers represent efficient sampling and the model is trained by standard backpropagation, making MAFs a great candidate for XenNF. Despite the efficiency MAFs present for learning target distributions, we find continuous normalizing flows a more interpretable fit for XenNF.

C.1.2 Continuous Normalizing Flows

Continuous normalizing flows (CNFs) are functions that approximate the limit of the set of discrete transformation:

$$z(t_{k+1}) = z(t_k) + f(z(t_k), \theta), \qquad k = 1, 2, \ldots, K$$

where f is a transformation parameterized by θ. CNFs model transformations as a continuous-time process governed by the ordinary differential equation (ODE)

$$\frac{dz(t)}{dt} = NN(z(t), t; \theta),$$

where NN is a neural network with parameters θ that learns these dynamics. Using an ODE solver, the continuous transformation is represented as

$$z\left(t_1\right) = z\left(t_0\right) + \int_{t_0}^{t_1} \frac{dz(t)}{dt} dt = z\left(t_0\right) + \int_{t_0}^{t_1} \mathrm{NN}_\theta(z(t), t) dt.$$

However, backpropagating through a continuous-depth network can prove challenging since many intermediate values need to be stored. This memory overhead can be circumvented using the adjoint sensitivity method (Pontryagin et al., 1962). By introducing a variable called the adjoint, $a(t) = \frac{\partial L}{\partial z(t)}$, we can model the sensitivity of the loss to the system state. As derived in Pontryagin et al. (1962), the adjoint satisfies its own ODE

$$\frac{da(t)}{dt} = -a(t)^\top \frac{\partial f(z(t), t, \theta)}{\partial z}.$$

Therefore, we can retrieve the gradient of the loss with respect to the state $z(t)$ on the fly by solving another ODE. Some clever algebra yields that gradients of the loss w.r.t. the network parameters θ can be achieved via the definite integral:

$$-\int_{t_1}^{t^0} a(t)^\top \frac{\partial f(z(t), t, \theta)}{\partial \theta} dt.$$

The distribution over the final state $z(t_1)$ defines the transformed density which represents the approximate posterior. For CNFs, the instantaneous change of variables for the log probability boils down to a trace calculation of the Jacobian under assumptions of f being uniformly Lipschitz continuous in z and continuous in t.

The authors of the CNF also proved that by using continuous flows in place of discrete ones, the change in log probability no longer requires a determinant, but instead only a simple trace:

$$\frac{\partial \log p(z(t))}{\partial t} = -tr\left(\frac{df}{dz(t)}\right).$$

This reduces the computational bottleneck from $O(N^3)$ to $O(N^2)$.

Furthermore, the computational burden of sampling from a CNF can be reduced using Hutchinson's trick (Hutchinson, 1990). Hutchinson's trick approximates the trace of the Jacobian with a Monte Carlo estimate:

$$\frac{d \log p(z(t))}{dt} = -tr\left(\frac{df}{dz(t)}\right) = E_{\epsilon \sim N(0,I)}\left[\epsilon^\top \frac{df}{dz(t)}\epsilon\right]$$
$$\approx \frac{1}{M}\sum_{m=1}^{M} \epsilon_m^\top \frac{df}{dz(t)}\epsilon_m.$$

This trick works because components of ϵ are independent of each other and the square of a standard normal random variable follows a χ_1^2 distribution. Therefore, $E(\epsilon_i \epsilon_j) = 0$ and $E(\epsilon_i^2) = 1$. Using this, we see that

$$E_{p(\epsilon)}\left[\epsilon^\top \frac{df}{dz(t)}\epsilon\right] = E_{p(\epsilon)}\left[\sum_{i=1}^{d}\sum_{j=1}^{d} \epsilon_i^\top \frac{df}{dz(t)}_{ij} \epsilon_j\right]$$
$$= \sum_{i=1}^{d}\sum_{j=1}^{d} \frac{df}{dz(t)}_{ij} E_{p(\epsilon)}\left[\epsilon_i^\top \epsilon_j\right]$$
$$= \sum_{i=1}^{d} \frac{df}{dz(t)}_{ii} = tr\left(\frac{df}{dz(t)}\right).$$

Through a reparameterization of the outputs as $f_{\text{weighted}} = \epsilon_1 f_1 + \epsilon_2 f_2 + \cdots + \epsilon_N f_n$. The gradient of this new function is precisely the dot product between our Jacobian and the noise vector:

$$\nabla f_{\text{weighted}} = \frac{df}{dz(t)} \cdot \epsilon.$$

Computing the unbiased estimate of the Jacobian trace is as simple as performing a dot product with the weighted output gradients: $\epsilon^\top \nabla f_{\text{weighted}}$. Avoiding explicit Jacobian calculation reduces the complexity to $O(N)$.

CNFs enable flexible yet tractable transformations of distributions by jointly transforming inputs over a continuous time instead of an iterative fashion. Our distributions over the logits, therefore, are represented by more gradual transformations rather then the potentially jagged transformations given by MAFs. We find that continuous time transformations with a shallow GCN as the hypernetwork encourages proximal spots to have similar logits without enforcing a hard neighborhood constraint.

C.2 XenNF Model Details

Training normalizing flows requires careful selection of many hyperparameters and architectures. We show many of the ones we evaluated here.

We start with the optimizer hyperparameters. XenNF was trained using the ClippedAdam optimizer—an extension of the Adam optimizer with built-in learning rate decay and gradient clipping—to mitigate the risk of numerical instability brought on by updating parameters into invalid regions due to large gradient steps. The tree below lists the hyperparameters explored with the final selected values underlined in blue.

- lr
  - Cluster Mean Parameters
    - betas
      - [0.9, 0.999]
    - lr: 0.001, 0.0001
    - lrd: 1.0, $0.1^{1/1000}$, $0.1^{1/10000}$
  - Cluster Scale Parameters
    - betas
      - [0.9, 0.999]
    - lr: 0.001, 0.0001
    - lrd: 1.0, $0.1^{1/1000}$, $0.1^{1/10000}$
  - Logit Parameters
    - betas
      - [0.9, 0.999]
    - lr: 0.001, 0.0001
    - lrd: 1.0, $0.1^{1/1000}$, $0.1^{1/10000}$

With the above established optimizer, Table C.1 represents the various hyperparameters that were tried for prior and posterior normalizing flows. For CNFs, we plug in a GCN as the hypernetwork to parameterize the continuous transformation dynamics. In contrast, MAFs consist of stacking multiple transformations. So, for MAF-based flows, we substitute the hypernetwork of the first transformation with a GCN. Just as before, final selected values are underline in blue.

Block	Hyperparameter	Value
Prior Flow (MAF)	Flow Length	1, 2, 3, 4
	Hidden Layers	[128, 128], [256, 256], [512, 512], [128, 128, 128], [256, 256, 256], [512, 512, 512]
	Graph Width	128, 256, 512, 1024
	Graph Depth	1, 2, 3
Prior Flow (CNF)	Graph Width	128, 256, 512, 1024
	Graph Depth	1, 2, 3
Posterior Flow (MAF)	Flow Length	4, 8, 12, 16
	Hidden Layers	[128, 128], [256, 256], [512, 512], [128, 128, 128], [256, 256, 256], [512, 512, 512]
Posterior Flow (CNF)	Relative Integration Tolerance	10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}
	Absolute Integration Tolerance	10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}
	Hidden Layers	[512, 512, 512], [1024, 1024, 1024] [2048, 1024, 512, 256, 128, 64, 32, 16] [512, 256, 128, 64, 32, 16]
	Activation	ELU, ReLU, Sigmoid, SoftPlus, Tanh
Training	KL Annealing Steps	0, 50, 100, 250, 1000
	Early Stopping Patience	3, 5, 10, 25, 50, 100
	Max Epochs	1000, 10000
	Empirical Prior Initial Clustering	K-Means, mclust
Data	Data Dimension (PCs)	5, 8 , 12, 15, 30
	Number of Clusters (K)	7

Table C.1: XenNF hyperparameter settings for posterior inference on DLPFC data.

C.3 Code Availability

All implementation of XenNF and accompanying scripts for reproducing real and synthetic experiments are available at https://github.com/prob-ml/xennf/tree/main/nf.

Bibliography

10x Genomics (2023a). Fresh frozen mouse brain replicates - 1 standard, in situ gene expression dataset analyzed using xenium onboard analysis 1.0.2, 10x genomics, (2023, january 22).

10x Genomics (2023b). Human breast dataset explorer. Accessed: 7 June 2024.

10x Genomics (n.d.a). Visium hd spatial gene expression. Accessed: 2024-06-06.

10x Genomics (n.d.b). What is the size of the spots on the visium gene expression slide? Accessed: 2024-06-06.

10x Genomics (n.d.c). What is the spatial resolution and configuration of the capture area of the visium v1 gene expression slide? Accessed: 2024-06-06.

Aitchison, J. and Shen, S. M. (1980). Logistic-normal distributions: Some properties and uses. *Biometrika*, 67(2):261.

Arora, R., Cao, C., Kumar, M., Sinha, S., Chanda, A., McNeil, R., Samuel, D., Arora, R. K., Matthews, T. W., Chandarana, S., Hart, R., Dort, J. C., Biernaskie, J., Neri, P., Hyrcza, M. D., and Bose, P. (2023a). Spatial transcriptomics reveals distinct and conserved tumor core and edge architectures that predict survival and targeted therapy response. *Nature Communications*, 14(1).

Arora, R., Cao, C., Kumar, M., Sinha, S., Chanda, A., McNeil, R., Samuel, D., Arora, R. K., Matthews, T. W., Chandarana, S., Hart, R., Dort, J. C., Biernaskie, J., Neri, P., Hyrcza, M. D., and Bose, P. (2023b). Spatial transcriptomics reveals distinct and conserved tumor core and edge architectures that predict survival and targeted therapy response. *Nature Communications*, 14(1).

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008.

Booeshaghi, A. S. and Pachter, L. (2021). Normalization of single-cell RNA-seq counts by log(x + 1)† or log(1 + x)†. *Bioinformatics*, 37(15):2223–2224.

Brown, T. G., Thayer, M. N., VanTreeck, J. G., Zarate, N., Hart, D. W., Heilbronner, S., and Gomez-Pastor, R. (2023). Striatal spatial heterogeneity, clustering, and white matter association of gfap+ astrocytes in a mouse model of huntington's disease. *Frontiers in Cellular Neuroscience*, 17.

Buxbaum, A. R., Haimovich, G., and Singer, R. H. (2014). In the right place at the right time: visualizing and understanding mrna localization. *Nature Reviews Molecular Cell Biology*, 16(2):95–109.

Cable, D. M., Murray, E., Shanmugam, V., Zhang, S., Zou, L. S., Diao, M., Chen, H., Macosko, E. Z., Irizarry, R. A., and Chen, F. (2022). Cell type-specific inference of differential expression in spatial transcriptomics. *Nature Methods*, 19(9):1076–1087.

Cera, E. D. (2020). Mechanisms of ligand binding. *Biophysics Reviews*, 1(1).

Chami, I., Abu-El-Haija, S., Perozzi, B., Ré, C., and Murphy, K. (2021). Machine learning on graphs: A model and comprehensive taxonomy.

Charafe-Jauffret, E., Ginestier, C., Monville, F., Finetti, P., Adélaïde, J., Cervera, N., Fekairi, S., Xerri, L., Jacquemier, J., Birnbaum, D., and Bertucci, F. (2005). Gene expression profiling of breast cell lines identifies potential new basal markers. *Oncogene*, 25(15):2273–2284.

Chen, A., Liao, S., Cheng, M., Ma, K., Wu, L., Lai, Y., Qiu, X., Yang, J., Xu, J., Hao, S., Wang, X., Lu, H., Chen, X., Liu, X., Huang, X., Li, Z., Hong, Y., Jiang, Y., Peng, J., Liu, S., Shen, M., Liu, C., Li, Q., Yuan, Y., Wei, X., Zheng, H., Feng, W., Wang, Z., Liu, Y., Wang, Z., Yang, Y., Xiang, H., Han, L., Qin, B., Guo, P., Lai, G., Muñoz-Cánoves, P., Maxwell, P. H., Thiery, J. P., Wu, Q.-F., Zhao, F., Chen, B., Li, M., Dai, X., Wang, S., Kuang, H., Hui, J., Wang, L., Fei, J.-F., Wang, O., Wei, X., Lu, H., Wang, B., Liu, S., Gu, Y., Ni, M., Zhang, W., Mu, F., Yin, Y., Yang, H., Lisby, M., Cornall, R. J., Mulder, J., Uhlén, M., Esteban, M. A., Li, Y., Liu, L., Xu, X., and Wang, J. (2022). Spatiotemporal transcriptomic atlas of mouse organogenesis using dna nanoball-patterned arrays. *Cell*, 185(10):1777–1792.e21.

Chen, K. H., Boettiger, A. N., Moffitt, J. R., Wang, S., and Zhuang, X. (2015). Spatially resolved, highly multiplexed rna profiling in single cells. *Science*, 348(6233).

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, 2(4):303–314.

Delaunay, B. (1934). Sur la sphère vide. *Izvestiya Akademii Nauk SSSR. Otdelenie Matematicheskikh i Estestvennykh Nauk*, 7:793–800.

Dong, K. and Zhang, S. (2022). Deciphering spatial domains from spatially resolved transcriptomics with an adaptive graph attention auto-encoder. *Nature Communications*, 13(1).

Dries, R., Zhu, Q., Dong, R., Eng, C.-H. L., Li, H., Liu, K., Fu, Y., Zhao, T., Sarkar, A., Bao, F., George, R. E., Pierson, N., Cai, L., and Yuan, G.-C. (2021). Giotto: a toolbox for integrative analysis and visualization of spatial expression data. *Genome Biology*, 22(1).

Emmert-Buck, M. R., Bonner, R. F., Smith, P. D., Chuaqui, R. F., Zhuang, Z., Goldstein, S. R., Weiss, R. A., and Liotta, L. A. (1996). Laser capture microdissection. *Science*, 274(5289):998–1001.

Eng, C.-H. L., Lawson, M., Zhu, Q., Dries, R., Koulena, N., Takei, Y., Yun, J., Cronin, C., Karp, C., Yuan, G.-C., and Cai, L. (2019). Transcriptome-scale super-resolved imaging in tissues by rna seqfish+. *Nature*, 568(7751):235–239.

Falcon, W., Borovec, J., Wälchli, A., Eggert, N., Schock, J., Jordan, J., Skafte, N., Ir1dXD, Bereznyuk, V., Harris, E., Tullie Murrell, Yu, P., Præsius, S., Addair, T., Zhong, J., Lipin, D., Uchida, S., Shreyas Bapat, Schröter, H., Dayma, B., Karnachev, A., Akshay Kulkarni, Shunta Komatsu, Martin.B, Jean-Baptiste SCHIRATTI, Mary, H., Byrne, D., Cristobal Eyzaguirre, Cinjon, and Bakhtin, A. (2025). Pytorchlightning/pytorch-lightning: 2.5.1 release.

Femino, A. M., Fay, F. S., Fogarty, K., and Singer, R. H. (1998). Visualization of single rna transcripts in situ. *Science*, 280(5363):585–590.

Fischer, D. S., Schaar, A. C., and Theis, F. J. (2023). Modeling intercellular communication in tissues using spatial graphs of cells. *Nature Biotechnology*, 41(3):332–336.

Fix, E. and Hodges, J. (1951). Pattern classification and scene analysis. *Proceedings of the Western Joint Computer Conference*, 8:125–134.

Foster, D., Frost-LaPlante, B., Victor, C., and Restrepo, J. M. (2021). Gradient sensing via cell communication. *Physical Review E*, 103(2).

Fu, X., Li, X., Wang, W., and Li, J. (2024). Dpp3 promotes breast cancer tumorigenesis by stabilizing fasn and promoting lipid synthesis. *Acta Biochimica et Biophysica Sinica*, 56(5):805–818.

Gao, X., Castro-Gomez, S., Grendel, J., Graf, S., Süsens, U., Binkle, L., Mensching, D., Isbrandt, D., Kuhl, D., and Ohana, O. (2018). Arc/arg3.1 mediates a critical period for spatial learning and hippocampal networks. *Proceedings of the National Academy of Sciences*, 115(49):12531–12536.

Giladi, A., Cohen, M., Medaglia, C., Baran, Y., Li, B., Zada, M., Bost, P., Blecher-Gonen, R., Salame, T.-M., Mayer, J. U., David, E., Ronchese, F., Tanay, A., and Amit, I. (2020). Dissecting cellular crosstalk by sequencing physically interacting cells. *Nature Biotechnology*, 38(5):629–637.

Gómez Hernández, G., Morell, M., and Alarcón-Riquelme, M. E. (2021). The role of bank1 in b cell signaling and disease. *Cells*, 10(5):1184.

Habern, O. (2024). How mapping the spatial biology of the tumor microenvironment can benefit cancer drug discovery and development - 10x Genomics — 10xgenomics.com. [Accessed 11-19-2024].

Hamilton, W., Ying, Z., and Leskovec, J. (2017). Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pages 1024–1034.

Hao, Y., Stuart, T., Kowalski, M. H., Choudhary, S., Hoffman, P., Hartman, A., Srivastava, A., Molla, G., Madad, S., Fernandez-Granda, C., and Satija, R. (2023). Dictionary learning for integrative, multimodal and scalable single-cell analysis. *Nature Biotechnology*.

He, S., Bhatt, R., Brown, C., Brown, E. A., Buhr, D. L., Chantranuvatana, K., Danaher, P., Dunaway, D., Garrison, R. G., Geiss, G., Gregory, M. T., Hoang, M. L., Khafizov, R., Killingbeck, E. E., Kim, D., Kim, T. K., Kim, Y., Klock, A., Korukonda, M., Kutchma, A., Lewis, Z. R., Liang, Y., Nelson, J. S., Ong, G. T., Perillo, E. P., Phan, J. C., Phan-Everson, T., Piazza, E., Rane, T., Reitz, Z., Rhodes, M., Rosenbloom, A., Ross, D., Sato, H., Wardhani, A. W., Williams-Wietzikoski, C. A., Wu, L., and Beechem, J. M. (2022). High-plex imaging of rna and proteins at subcellular resolution in fixed tissue by spatial molecular imaging. *Nature Biotechnology*, 40(12):1794–1806.

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. (2013). Stochastic variational inference. *Journal of Machine Learning Research*, 14(40):1303–1347.

Hou, R., Denisenko, E., Ong, H. T., Ramilowski, J. A., and Forrest, A. R. R. (2020). Predicting cell-to-cell communication networks using natmi. *Nature Communications*, 11(1):5011.

Hu, J., Li, X., Coleman, K., Schroeder, A., Ma, N., Irwin, D. J., Lee, E. B., Shinohara, R. T., and Li, M. (2021). SpaGCN: Integrating gene expression, spatial location and histology to identify spatial domains and spatially variable genes by graph convolutional network. *Nature Methods*, 18(11):1342–1351.

Hutchinson, M. (1990). A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics - Simulation and Computation*, 19(2):433–450.

Ives, A. R. (2015). For testing the significance of regression coefficients, go ahead and log-transform count data. *Methods in Ecology and Evolution*, 6(7):828–835.

Janesick, A., Shelansky, R., Gottscho, A. D., Wagner, F., Williams, S. R., Rouault, M., Beliakoff, G., Morrison, C. A., Oliveira, M. F., Sicherman, J. T., Kohlway, A., Abousoud, J., Drennon, T. Y., Mohabbat, S. H., and Taylor, S. E. B. (2023a). High resolution mapping of the tumor microenvironment using integrated single-cell, spatial and in situ analysis. *Nature Communications*, 14(1).

Janesick, A., Shelansky, R., Gottscho, A. D., Wagner, F., Williams, S. R., Rouault, M., Beliakoff, G., Morrison, C. A., Oliveira, M. F., Sicherman, J. T., Kohlway, A., Abousoud, J., Drennon, T. Y., Mohabbat, S. H., and Taylor, S. E. B. (2023b). High resolution

mapping of the tumor microenvironment using integrated single-cell, spatial and in situ analysis. *Nature Communications*, 14(1).

Jia, W., Sun, M., Lian, J., and Hou, S. (2022). Feature dimensionality reduction: a review. *Complex & Intelligent Systems*, 8(3):2663–2693.

Jin, S., Guerrero-Juarez, Z., Zhang, X., Chang, Z., Shao, B., Yang, C.-L., Choi, Y., McGeer, T. A., Plikus, W. E., and Nie, Q. (2021). Inferences and analysis of cell-cell communication using cellchat. *Nature Communications*, 12(1):1088.

Johnson, S. C. (1967). Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254.

Kim, J. and Cho, J. (2019). Delaunay triangulation-based spatial clustering technique for enhanced adjacent boundary detection and segmentation of LiDAR 3d point clouds. *Sensors*, 19(18):3926.

Kingma, D. P. and Ba, J. (2017). Adam: A method for stochastic optimization.

Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR)*.

Kipf, T. N. and Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907.

Krause, A. L., Beliaev, D., Van Gorder, R. A., and Waters, S. L. (2018). Lattice and continuum modelling of a bioactive porous tissue scaffold. *Mathematical Medicine and Biology: A Journal of the IMA*, 36(3):325–360.

Labrèche, C., Cook, D. P., Abou-Hamad, J., Pascoal, J., Pryce, B. R., Al-Zahrani, K. N., and Sabourin, L. A. (2021). Periostin gene expression in neu-positive breast cancer cells is regulated by a fgfr signaling cross talk with tgfβ/pi3k/akt pathways. *Breast Cancer Research*, 23(1).

Lei, J., G'Sell, M., Rinaldo, A., Tibshirani, R. J., and Wasserman, L. (2018). Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113(523):1094–1111.

Levine, J. H., Simonds, E. F., Bendall, S. C., Davis, K. L., Amir, E.-a. D., Tadmor, M. D., Litvin, O., Fienberg, H. G., Jager, A., Zunder, E. R., Finck, R., Gedman, A. L., Radtke, I., Downing, J. R., Pe'er, D., and Nolan, G. P. (2015). Data-driven phenotypic dissection of aml reveals progenitor-like cells that correlate with prognosis. *Cell*, 162(1):184–197.

Li, D., Ding, J., and Bar-Joseph, Z. (2020). Identifying signaling genes in spatial single-cell expression data. *Bioinformatics*, 37(7):968–975.

Li, H., Ma, T., Hao, M., Guo, W., Gu, J., Zhang, X., and Wei, L. (2023a). Decoding functional cell–cell communication events by multi-view graph learning on spatial transcriptomics. *Briefings in Bioinformatics*, 24(6).

Li, X. and Wang, C.-Y. (2021). From bulk, single-cell to spatial rna sequencing. *International Journal of Oral Science*, 13(1).

Li, Y., Wu, M., Ma, S., and Wu, M. (2023b). Zinbmm: a general mixture model for simultaneous clustering and gene selection using single-cell transcriptomic data. *Genome Biology*, 24(1).

Li, Z. and Zhou, X. (2022). Bass: multi-scale and multi-sample analysis enables accurate cell type clustering and spatial domain detection in spatial transcriptomic studies. *Genome Biology*, 23(1).

Lin, M., Lucas, H. C., and Shmueli, G. (2013). Research commentary—too big to fail: Large samples and the p-value problem. *Information Systems Research*, 24(4):906–917.

Long, Y., Ang, K. S., Li, M., Chong, K. L. K., Sethi, R., Zhong, C., Xu, H., Ong, Z., Sachaphibulkij, K., Chen, A., Zeng, L., Fu, H., Wu, M., Lim, L. H. K., Liu, L., and Chen, J. (2023). Spatially informed clustering, integration, and deconvolution of spatial transcriptomics with graphst. *Nature Communications*, 14(1).

Lopez, R., Li, B., Keren-Shaul, H., Boyeau, P., Kedmi, M., Pilzer, D., Jelinski, A., Yofe, I., David, E., Wagner, A., Ergen, C., Addadi, Y., Golani, O., Ronchese, F., Jordan, M. I., Amit, I., and Yosef, N. (2022). Destvi identifies continuums of cell types in spatial transcriptomics data. *Nature Biotechnology*, 40(9):1360–1369.

Luo, W., Lin, G. N., Song, W., Zhang, Y., Lai, H., Zhang, M., Miao, J., Cheng, X., Wang, Y., Li, W., Wei, W., Gao, W.-Q., Yang, R., and Wang, J. (2021). Single-cell spatial transcriptomic analysis reveals common and divergent features of developing postnatal granule cerebellar cells and medulloblastoma. *BMC Biology*, 19(1).

Ma, Y. and Zhou, X. (2024). Accurate and efficient integrative reference-informed spatial domain detection for spatial transcriptomics. *Nature Methods*, 21(7):1231–1244.

Mac, N. A. and Nguyen, H. S. (2021). Rotation invariance in graph convolutional networks. In *Annals of Computer Science and Information Systems*. IEEE.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, pages 281–297, Berkeley, CA. University of California Press.

Marx, V. (2021). Method of the year: spatially resolved transcriptomics. *Nature methods*, 18(1):9–14.

Maynard, K. R., Collado-Torres, L., Weber, L. M., Uytingco, C., Barry, B. K., Williams, S. R., Catallini, J. L., Tran, M. N., Besich, Z., Tippani, M., Chew, J., Yin, Y., Kleinman, J. E., Hyde, T. M., Rao, N., Hicks, S. C., Martinowich, K., and Jaffe, A. E. (2021). Transcriptome-scale spatial gene expression in the human dorsolateral prefrontal cortex. *Nature Neuroscience*, 24(3):425–436.

McInnes, L., Healy, J., and Melville, J. (2020). Umap: Uniform manifold approximation and projection for dimension reduction.

Miller, B. F., Bambah-Mukku, D., Dulac, C., Zhuang, X., and Fan, J. (2021). Characterizing spatial gene expression heterogeneity in spatially resolved single-cell transcriptomic data with nonuniform cellular densities. *Genome Research*, 31(10):1843–1855.

Moffitt, J. R., Bambah-Mukku, D., Eichhorn, S. W., Vaughn, E., Shekhar, K., Perez, J. D., Rubinstein, N. D., Hao, J., Regev, A., Dulac, C., et al. (2018). Molecular, spatial, and functional single-cell profiling of the hypothalamic preoptic region. *Science*, 362(6416):eaau5324.

Monti, F., Boscaini, D., Masci, J., Rodolà, E., Svoboda, J., and Bronstein, M. M. (2016). Geometric deep learning on graphs and manifolds using mixture model cnns.

Moses, L., Einarsson, P. H., Jackson, K., Luebbert, L., Booeshaghi, A. S., Antonsson, S., Bray, N., Melsted, P., and Pachter, L. (2023). Voyager: exploratory single-cell genomics data analysis with geospatial statistics.

Neyman, J. and Pearson, E. S. (1933). On the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 231:289–337.

Nirmal, A. J. and Sorger, P. K. (2024). Scimap: A python toolkit for integrated spatial analysis of multiplexed imaging data. *Journal of Open Source Software*, 9(97):6604.

Oxford Nanopore Technologies (2023). Single-cell and spatial transcriptomics help to unlock our understanding of the subtleties of cellular diversity.

Palla, G., Spitzer, H., Klein, M., Fischer, D., Schaar, A. C., Kuemmerle, L. B., Rybakov, S., Ibarra, I. L., Holmberg, O., Virshup, I., Lotfollahi, M., Richter, S., and Theis, F. J. (2022). Squidpy: a scalable framework for spatial omics analysis. *Nature Methods*, 19(2):171–178.

Petersen, K. B. and Pedersen, M. S. (2012). The matrix cookbook. Version 20121115.

Pham, D., Tan, X., Balderson, B., Xu, J., Grice, L. F., Yoon, S., Willis, E. F., Tran, M., Lam, P. Y., Raghubar, A., Kalita-de Croft, P., Lakhani, S., Vukovic, J., Ruitenberg, M. J., and Nguyen, Q. H. (2023). Robust mapping of spatiotemporal trajectories and cell–cell interactions in healthy and diseased tissues. *Nature Communications*, 14(1).

Pichon, X., Lagha, M., Mueller, F., and Bertrand, E. (2018). A growing toolbox to image gene expression in single cells: Sensitive approaches for demanding challenges. *Molecular Cell*, 71(3):468–480.

Pontryagin, L., Boltyanskii, V., Gamkrelidze, R., and Neustadt, L. (1962). *The Mathematical Theory of Optimal Processes*. Wiley-Interscience.

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850.

Rezende, D. J. and Mohamed, S. (2016). Variational inference with normalizing flows.

Rodriques, S. G., Stickels, R. R., Goeva, A., Martin, C. A., Murray, E., Vanderburg, C. R., Welch, J., Chen, L. M., Chen, F., and Macosko, E. Z. (2019). Slide-seq: A scalable technology for measuring genome-wide expression at high spatial resolution. *Science*, 363(6434):1463–1467.

Satija, R., Farrell, J. A., Gennert, D., Schier, A. F., and Regev, A. (2015a). Spatial reconstruction of single-cell gene expression data. *Nature Biotechnology*, 33(5):495–502.

Satija, R., Farrell, J. A., Gennert, D., Schier, A. F., and Regev, A. (2015b). Spatial reconstruction of single-cell gene expression data. *Nature Biotechnology*, 33(5):495–502.

Scrucca, L., Fraley, C., Murphy, T. B., and Raftery, A. E. (2023). *Model-Based Clustering, Classification, and Density Estimation Using mclust in R*. Chapman and Hall/CRC.

Sel, S., Patzel, E., Poggi, L., Kaiser, D., Kalinski, T., Schicht, M., Paulsen, F., and Nass, N. (2017). Temporal and spatial expression pattern of nnat during mouse eye development. *Gene Expression Patterns*, 23-24:7–12.

Shi, H., He, Y., Zhou, Y., Huang, J., Maher, K., Wang, B., Tang, Z., Luo, S., Tan, P., Wu, M., Lin, Z., Ren, J., Thapa, Y., Tang, X., Chan, K. Y., Deverman, B. E., Shen, H., Liu, A., Liu, J., and Wang, X. (2023). Spatial atlas of the mouse central nervous system at molecular resolution. *Nature*, 622(7983):552–561.

Smith, K. D., Prince, D. K., MacDonald, J. W., Bammler, T. K., and Akilesh, S. (2024). Challenges and opportunities for the clinical translation of spatial transcriptomics technologies. *Glomerular Dis.*, 4(1):49–63.

Ståhl, P. L., Salmén, F., Vickovic, S., Lundmark, A., Navarro, J. F., Magnusson, J., Giacomello, S., Asp, M., Westholm, J. O., Huss, M., Mollbrink, A., Linnarsson, S., Codeluppi, S., Borg, Å., Pontén, F., Costea, P. I., Sahlén, P., Mulder, J., Bergmann, O., Lundeberg, J., and Friszén, J. (2016). Visualization and analysis of gene expression in tissue sections by spatial transcriptomics. *Science*, 353(6294):78–82.

Stickels, R. R., Murray, E., Kumar, P., Li, J., Marshall, J. L., Di Bella, D. J., Arlotta, P., Macosko, E. Z., and Chen, F. (2020a). Highly sensitive spatial transcriptomics at near-cellular resolution with slide-seqv2. *Nature Biotechnology*, 39(3):313–319.

Stickels, R. R., Murray, E., Kumar, P., Li, J., Marshall, J. L., Di Bella, D. J., Arlotta, P., Macosko, E. Z., and Chen, F. (2020b). Highly sensitive spatial transcriptomics at near-cellular resolution with slide-seqv2. *Nature Biotechnology*, 39(3):313–319.

Stockmann, H., Todorovic, V., Richardson, P. L., Marin, V., Scott, V., Gerstein, C., Lake, M., Wang, L., Sadhukhan, R., and Vasudevan, A. (2017). Cell-surface receptor–ligand interaction analysis with homogeneous time-resolved FRET and metabolic glycan engineering: Application to transmembrane and GPI-anchored receptors. *Journal of the American Chemical Society*, 139(46):16822–16829.

Su, J., Reynier, J.-B., Fu, X., Zhong, G., Jiang, J., Escalante, R. S., Wang, Y., Izar, B., Knowles, D. A., and Rabadan, R. (2022). A unified modular framework to incorporate structural dependency in spatial omics data. *bioRxiv*.

Su, K., Huang, L., Li, W., Yan, X., Li, X., Zhang, Z., Jin, F., Lei, J., Ba, G., Liu, B., Wang, X., and Wang, Y. (2013). Tc-1 (c8orf4) enhances aggressive biologic behavior in lung cancer through the wnt/β-catenin pathway. *Journal of Surgical Research*, 185(1):255–263.

Tirosh, I., Izar, B., Prakadan, S. M., Wadsworth, M. H., Treacy, D., Trombetta, J. J., Rotem, A., Rodman, C., Lian, C., Murphy, G., Fallahi-Sichani, M., Dutton-Regester, K., Lin, J.-R., Cohen, O., Shah, P., Lu, D., Genshaft, A. S., Hughes, T. K., Ziegler, C. G. K., Kazer, S. W., Gaillard, A., Kolb, K. E., Villani, A.-C., Johannessen, C. M., Andreev, A. Y., Allen, E. M. V., Bertagnolli, M., Sorger, P. K., Sullivan, R. J., Flaherty, K. T., Frederick, D. T., Jané-Valbuena, J., Yoon, C. H., Rozenblatt-Rosen, O., Shalek, A. K., Regev, A., and Garraway, L. A. (2016). Dissecting the multicellular ecosystem of metastatic melanoma by single-cell RNA-seq. *Science*, 352(6282):189–196.

Traag, V. A., Waltman, L., and van Eck, N. J. (2019). From louvain to leiden: guaranteeing well-connected communities. *Scientific Reports*, 9(1).

Türei, D., Valdeolivas, A., Gul, L., Palacio-Escat, N., Klein, M., Ivanova, O., Olbei, M., Gabor, A., Theis, F. J., Módos, D., Korcsmáros, T., and Saez-Rodriguez, J. (2021). Integrated intra- and intercellular signaling knowledge for multicellular omics analysis. *Molecular Systems Biology*, 17(3):e9923.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. (2017). Graph attention networks. In *International Conference on Learning Representations*.

Walker, B. L., Cang, Z., Ren, H., Bourgain-Chang, E., and Nie, Q. (2022). Deciphering tissue structure and function using spatial transcriptomics. *Communications Biology*, 5(1).

Wang, N., Song, Y., Hong, W., Mo, H., Song, Z., Dai, W., Wang, L., Zhang, H., Zhang, Y., Zhang, Q., Zhang, H., Zhang, T., Wang, Y., Li, Y., Ma, J., Shao, C., Yu, M., Qian, H., Ma, F., and Ding, Z. (2024). Spatial single-cell transcriptomic analysis in breast cancer reveals potential biomarkers for pd1 blockade therapy.

Weber, L. M., Saha, A., Datta, A., Hansen, K. D., and Hicks, S. C. (2023). nnsvg for the scalable identification of spatially variable genes using nearest-neighbor gaussian processes. *Nature Communications*, 14(1).

Wei, R., He, S., Bai, S., Sei, E., Hu, M., Thompson, A., Chen, K., Krishnamurthy, S., and Navin, N. E. (2022). Spatial charting of single-cell transcriptomes in tissues. *Nature Biotechnology*, 40(8):1190–1199.

Wilczynski, B., Liu, Y.-H., Yeo, Z. X., and FSUPURLong, E. E. M. (2012). Predicting spatial and temporal gene expression using an integrative model of transcription factor occupancy and chromatin state. *PLoS Computational Biology*, 8(12):e1002798.

Wolf, F. A., Angerer, P., and Theis, F. J. (2018). Scanpy: large-scale single-cell gene expression data analysis. *Genome Biology*, 19(1).

Wolpert, L. (1969). Positional information and the spatial pattern of cellular differentiation. *Journal of theoretical biology*, 25(1):1–47.

Wu, F. Y. (1982). The potts model. *Reviews of Modern Physics*, 54(1):235–268.

Wu, G., Wang, D., Xiong, F., Wang, Q., Liu, W., Chen, J., and Chen, Y. (2024). The emerging roles of ceacam6 in human cancer (review). *International Journal of Oncology*, 64(3).

Xu, H., Fu, H., Long, Y., Ang, K. S., Sethi, R., Chong, K., Li, M., Uddamvathanak, R., Lee, H. K., Ling, J., Chen, A., Shao, L., Liu, L., and Chen, J. (2024). Unsupervised spatially embedded deep representation of spatial transcriptomics. *Genome Medicine*, 16(1).

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. (2018a). How powerful are graph neural networks? In *International Conference on Learning Representations*.

Xu, M., Quiroz, M., Kohn, R., and Sisson, S. A. (2018b). Variance reduction properties of the reparameterization trick.

Xu, S., Chen, T., Dong, L., Li, T., Xue, H., Gao, B., Ding, X., Wang, H., and Li, H. (2020). Fatty acid synthase promotes breast cancer metastasis by mediating changes in fatty acid metabolism. *Oncology Letters*, 21(1):1–1.

Yang, Y., Hong, Y., Zhao, K., Huang, M., Li, W., Zhang, K., and Zhao, N. (2024). Spatial transcriptomics analysis identifies therapeutic targets in diffuse high-grade gliomas. *Frontiers in Molecular Neuroscience*, 17.

Yang, Y., Shi, X., Liu, W., Zhou, Q., Lau, M. C., Lim, J. C. T., Sun, L., Ng, C. C. Y., Yeong, J., and Liu, J. (2021). SC-MEB: spatial clustering with hidden markov random field using empirical bayes. *Briefings in Bioinformatics*, 23(1).

Yuan, Y. and Bar-Joseph, Z. (2020). GCNG: graph convolutional networks for inferring gene interaction from spatial transcriptomics data. *Genome Biology*, 21(1).

Zhang, Y., Brady, M., and Smith, S. (2001). Segmentation of brain mr images through a hidden markov random field model and the expectation-maximization algorithm. *IEEE Transactions on Medical Imaging*, 20(1):45–57.

Zhao, E., Stone, M. R., Ren, X., Guenthoer, J., Smythe, K. S., Pulliam, T., Williams, S. R., Uytingco, C. R., Taylor, S. E. B., Nghiem, P., Bielas, J. H., and Gottardo, R. (2021). Spatial transcriptomics at subspot resolution with BayesSpace. *Nature Biotechnology*, 39(11):1375–1384.

Zhao, Y., Kohl, C., Rosebrock, D., Hu, Q., Hu, Y., and Vingron, M. (2024). Cabinet: joint clustering and visualization of cells and genes for single-cell transcriptomics. *Nucleic Acids Research*, 52(13):e57–e57.