Spatial Information Based Imputation for Spatially Resolved Transcriptomics Data

By

Fahim Hafiz and 0122320008

Submitted in partial fulfillment of the requirements
of the degree of Master of Science in Computer Science and Engineering

October 14, 2025



Department of Computer Science and Engineering
United International University

Abstract

New technologies in Spatially Resolved Transcriptomics (SRT) Data have achieved near single-cell resolution while consisting of spatial data that has an immense impact in discovering various biological insights. High-resolution SRT data are sparse and contain dropouts, which may hinder the accurate interpretation of spatial domains. Hence, computationally available imputation methods are often necessary to impute the dropouts from such datasets. Numerous state-of-the-art (SOTA) imputation methods already exist that are used in general tabular data, dedicated single-cell RNA data, as well as the SRT datasets. However, there is no benchmarking of these imputation methods on these newer SRT technologies' datasets. In this work, we select seven SOTA imputation methods and provide detailed benchmarking results on five SRT technologies consisting of 23 datasets. We show that there are no single imputation methods that consistently outperform others, demonstrating poor imputation capability in most cases, and also poor performance in identifying valid dropouts. Thus, analyzing these SOTA methods' performance, we propose a new imputation method, `'SpaMean-Impute'`, designed for SRT datasets that utilize spatial information while mitigating dropouts. Furthermore, the proposed method can detect valid dropouts, unlike the SOTA benchmarked methods. Due to integration of spatial information-based dropout location detection, `'SpaMean-Impute'` only imputes a valid location while preserving the inherent biological relations of the datasets, unlike SOTA methods. Our proposed method outperforms the SOTA imputation methods across evaluation metrics such as: Adjusted Rand Index(ARI), Normalized Mutual Information (NMI), Adjusted Mutual Information(AMI), and Homogeneity (HOMO). In case of ARI, the proposed method outperforms the SOTA methods on average 16.15%, whereas 18.45% improvement in NMI, 18.96% improvement in AMI, and 13.98% improvement in the case of HOMO. Furthermore, the proposed method is computationally efficient compared to other SOTA methods. For example, compared to the SOTA deep-learning-based imputation methods, the proposed method is approximately 33 times faster and requires, on average, 1500 MB less memory during imputation.

Acknowledgements

This work would have not been possible without the input and support of many people over the last two trimesters. I would like to express my gratitude to everyone who contributed to it in some way or other.

First, I would like to thank my academic advisors. I would like to express my sincere gratitude to my advisors, **Dr. Riasat Azim** and **Dr. Swakkhar Shatabda**, for their invaluable guidance and support throughout my master's program. Their expertise and encouragement helped me to complete this research and write this thesis. Coming from an EEE background from an undergraduate program with little introduction to Machine Learning and AI, both my advisers supported me in every way possible. They advised me on what to read, what to learn, which workshop/seminar to attend, and in which direction to push my research. They always encouraged me, appreciated my efforts, and introduced me to various networks for my own growth in the research world. In the future, wherever I go, I will deeply remember both of them to the core of my heart.

My sincere gratitude goes to **Dr. Mohammad Nurul Huda** for guiding me at the beginning of my MSc and for his valuable suggestions that have shaped my academic career. I am also thankful to **Dr. Khondaker Abdullah-Al-Mamun** for his tremendous motivation and teachings in the Research & Methodology course in my MSc. I learned the basics of research from his course and utilized it to conduct my thesis.

I would like to express my deepest gratitude to my beautiful wife, **Mrs. Subhana Nowshin**, who has stood by me through thick and thin over the past two years of my MSc journey. Embarking on this path was a new challenge in a different domain, and her unwavering support gave me the strength to face my fears and endure the restless nights. Her motivation and constant encouragement were instrumental in helping me achieve this dream. This thesis would not have been possible without her. Thank you.

Last but not least, I owe to our family, including my **parents**, **my sister**, for their unconditional love and immense emotional support. They always understood my goals and supported me each minute, each second of my life. **Gracias, Mom, Dad, and Sister!**

Publication List

Journal Articles

1. Hafiz, F., Azim, R., & Shatabda, S. 'Spatial Information Matters: Are Traditional Imputation Methods Effective for Spatial Transcriptomics Data?' [Submitted to Briefings in Bioinformatics, Oxford University Press]

Table of Contents

Table of Contents **v**

List of Figures **vi**

List of Tables **viii**

1 Introduction **1**
1.1 Problem Overview 1
1.2 Motivation 2
1.3 Objectives 2
1.4 Methodology 3
1.5 Research Outcome 3
1.6 Organization of the Report 4

2 Background **5**
2.1 Background of SRT Technologies 5
2.2 Literature Review and Gap Analysis 8
2.3 Summary 12

3 Methodology **13**
3.1 Proposed Methodology for Benchmarking 13
3.1.1 Datasets 14
3.1.2 Dataset Preprocessing 17
3.1.3 Imputation Methods included in the Pipeline 17
3.1.4 Metrics to evaluate the Imputation Methods 24
3.2 Proposed Imputation Framework-SpaMean-Impute 26
3.3 Summary 31

4 Results **32**
4.1 Benchmark Results 32
4.1.1 Evaluation of Imputation Methods Using Clustering-Based Metrics 32

4.1.2 Evaluation of Dropout Reduction Capability Across Imputation Methods . . . 37
4.1.3 Computational Metrics among the imputation methods . . . 39
4.2 Performance of the SpaMean-Impute . . . 40
4.2.1 Comparison of clustering metrics of SpaMean-Impute with SOTA methods . . . 41
4.2.2 Average Zero Sparsity percentage comparison of SpaMean-Impute . 44
4.2.3 Computational Complexity of SpaMean-Impute . . . 45
4.3 Summary . . . 45

5 Discussion and Conclusion **48**
5.1 Summary . . . 48
5.2 Limitation . . . 50
5.3 Future Work . . . 50

References **57**

A Supplementary Materials **58**
A.1 Data and Code Availability . . . 58
A.2 Detailed Performance on sci-Space . . . 58
A.3 Sparsity in Slide-seqV2 before and after applying Imputation . . . 59
A.4 Sparsity in XYZeq before and after applying Imputation . . . 60
A.5 Sparsity in sci-Space before and after applying Imputation . . . 61
A.6 Computational Performance of 'SpaMean-Impute' on SRT Technologies (excluding 10x Genomics Visium) . . . 62

List of Figures

2.1 Example diagram (for illustration purposes) demonstrating the difference between Bulk RNA-seq, single-cell RNA-seq (scRNA-seq), and Spatial Transcriptomics in the Human Brain[1]. Bulk RNA-seq profiles the brain into average gene expression, while scRNA-seq technology can profile the brain tissue at the single-cell level. However, scRNA-seq loses the spatial information of the cell within the tissue, while spatial transcriptomics can retain the spatial information and profile the tissue [1]. 6

3.1 Overview of the computational pipeline for evaluating imputation methods in SRT data. 14

3.2 This diagram outlines the steps used in SpaMean-Impute to impute zero gene expression values in SRTs. Starting with a filtered expression matrix and spatial coordinates, the method identifies zero entries per spot and locates the k nearest neighbors. For each zero, a sub-matrix of neighbor values is extracted, and the zero ratio is computed. If the ratio is below a set threshold, the value is imputed using the average of non-zero neighbor values. This approach recovers likely technical dropouts while preserving true biological sparsity. 27

4.1 Clustering benchmark results for the 10x Genomics Visium dataset across three gene selection levels (2000 genes, 5000 genes, and all genes). Eight methods—including the base (no imputation) and seven imputation approaches—are evaluated using (a) ARI, (b) NMI, (c) AMI, and (d) HOMO scores. `gimVI` performs well in most cases. However, the base method performs well when all genes are selected. 33

4.2 Clustering benchmark results for the Stereo-seq dataset across three gene selection levels (2000 genes, 5000 genes, and all genes). Eight methods—including the base (no imputation) and seven imputation approaches—are evaluated using (a) ARI, (b) NMI, (c) AMI, and (d) HOMO scores. MAGIC method consistently performs well across all metrics. 34

4.3 Clustering performance on the Slide-seqV2 dataset across three gene selection levels (2000 genes, 5000 genes, and all genes), evaluated using (a) ARI, (b) NMI, (c) AMI, and (d) HOMO scores. The imputation method `tangram` consistently outperforms all other methods across most metrics and gene selections. 35
4.4 Clustering evaluation results on the XYZeq dataset across three gene selection levels (2000 genes, 5000 genes, and all genes), assessed using (a) ARI, (b) NMI, (c) AMI, and (d) HOMO scores. The boxplot distributions appear skewed in most cases. Deep learning-based methods such as `scVI`, `gimVI`, and `tangram` demonstrate competitive clustering performance across all metrics and gene subsets. In contrast, simpler imputation methods consistently underperform, highlighting the importance of model complexity in handling the characteristics of XYZeq. 36
4.5 Computational Complexity of the Imputation Methods for varying number of spots (a) Memory consumption and (b) Runtime comparison for the imputation methods for `200 genes, 5000 genes or all genes`. Deep learning-based methods, particularly `gimVI` and `scVI`, exhibit the highest runtimes due to their iterative training and complex model architectures, whereas `tangram` has the highest memory consumption. 40
4.6 Computational Complexity comparison between SpaMean-Impute and other SOTA methods for 10x Genomics Visiumdatasets (a) Memory consumption and (b) Runtime considering all genes. SpaMean-Impute shows one of the lowest runtimes and memory consumption compared to other methods, especially the deep learning-based models. 46

A.1 Memory consumption comparison of 'SpaMean-Impute' with others for Stereo-seq, Slide-seqV2, sci-Space. 'tangram' consumes the highest memory as well, and 'SpaMean-Impute' consumes the lowest. 62
A.2 Runtime comparison of 'SpaMean-Impute' with others for Stereo-seq, Slide-seqV2, sci-Space. 'gimVI' takes the highest runtime here as well, and 'SpaMean-Impute' takes the lowest. 63

List of Tables

3.1 SRT Datasets for 5 technologies used in our benchmarking. Dataset size indicates (Spots x Genes), and the table demonstrates both the size before preprocessing (i.e., Raw) and after preprocessing. 16

4.1 Percentage of Zero Sparsity in the 10x Genomics Visium datasets before (Raw) and after applying imputation methods. 38
4.2 Percentage of Zero Sparsity in the Stereo-seq datasets before (Raw) and after applying imputation methods. 38
4.3 ARI scores for SOTA imputation methods compared to 'SpaMean-Impute' The highest score for each technology's tissue slide corresponding to the respective imputation method is bolded. 41
4.4 Comparison of NMI Scores with 'SpaMean-Impute'. The highest score for each technology's tissue slide corresponding to the respective imputation method is bolded. 42
4.5 AMI scores for SOTA imputation methods compared to 'SpaMean-Impute' The highest score for each technology's tissue slide corresponding to the respective imputation method is bolded. 43
4.6 Comparison of HOMO Scores with 'SpaMean-Impute'. The highest score for each technology's tissue slide corresponding to the respective imputation method is bolded. 44
4.7 Average Zero Sparsity Percentage comparison of 'SpaMean-Impute' with other SOTA imputation methods. 45

A.1 Performance metrics for different imputation methods applied to sci-Space across varying numbers of top genes 58
A.2 Percentage of Zero Sparsity in the Slide-seqV2 datasets before (RAW) and after applying imputation methods. 59
A.3 Percentage of Zero Sparsity in the XYZeq datasets before (RAW) and after applying imputation methods. 60
A.4 Percentage of Zero Sparsity in the sci-Space datasets before (RAW) and after applying imputation methods. 61

Chapter 1

Introduction

In this chapter, the background of the thesis is discussed, as well as the motivation. In addition, the objectives are stated along with a summary of the methodology and the results of the project. Lastly, the structured organization of the thesis is presented.

1.1 Problem Overview

Spatially Resolved Transcriptomics (SRT) is a state-of-the-art (SOTA) technology for sequencing tissues and is currently the highest applied sequencing technology in the field of computational biology [2]. This technology is rapidly evolving and becoming more high-resolution (i.e., single-cell resolution), such as 10x Genomics Visium, Stereo-seq, Slide-seqV2, etc [3]. As a result, newer technologies in SRT are becoming more sparse, where technical biases are being introduced, which are termed as 'dropouts' (zero values due to technical artifacts) [1, 3, 4, 5]. Furthermore, there exists very few methods on SRT that can detect such valid dropouts and impute them[6]. However, there exist various types of imputation methods in tabular data, in single-cell data in the literature [7]. As the imputation methods in the field of SRT have not grown yet, it is necessary to benchmark existing SOTA imputation methods on how they perform on these new SRT technologies. Several research questions (RQs) have been proposed for benchmarking in this work:

- Are there any imputation method that consistently performs well across all SRT technologies?
- Is there any method that performs well in a particular technology?
- Do top gene selections impact the performance of the methods?.

Furthermore, after investigating the benchmarking results, several limitations of the SOTA methods have been identified and a new imputation method has been proposed, 'SpaMean-Impute'.

1.2 Motivation

SRT technologies have emerged as powerful tools for profiling gene expression while preserving the spatial context within tissues. As these platforms advance toward higher resolutions in platforms such as 10x Genomics Visium, Stereo-seq, and Slide-seqV2, the resulting datasets have become increasingly sparse [8]. A significant portion of this sparsity arises from zero values, commonly referred to as *dropouts*, which may reflect technical artifacts [6]. Furthermore, such datasets are sparse due to high-throughput sequencing, where biologically unexpressed genes are also zeros. Hence, mitigating these technical artifacts is necessary for such high-sparse data, as they directly affect key downstream analyses such as spatial clustering, domain detection, and the identification of spatially variable genes (SVGs) [6]. Inaccurate imputation can lead to artificial clusters or missed biological patterns. These analyses are fundamental for understanding tissue organization, cellular heterogeneity, and region-specific gene expression. Therefore, effective dropout handling ensures the preservation of meaningful spatial patterns and supports reliable biological interpretation. However, current imputation methods for SRT data are still developing. Many approaches are borrowed from single-cell RNA-seq or traditional tabular data domains and may fail to capture the spatial structure and technology-specific biases inherent in SRT datasets [6]. This creates an urgent need to systematically benchmark existing imputation methods across various SRT technologies to understand their robustness, limitations, and adaptability.

From a computational perspective, the imputation problem in SRT data poses several machine learning challenges. The data is high-dimensional and sparse [2, 5, 8]. The structure of the data—where each observation includes both gene expression features and spatial coordinates, introducing opportunities for leveraging spatial graphs and representation learning [9, 10, 11]. Furthermore, it involves developing methods that are robust to domain shifts, capable of capturing complex spatial correlations, and generalizable across varying levels of resolution, sparsity, and technologies. By evaluating a wide range of SOTA techniques, this work not only advances tools for spatial genomics but also contributes to the growing field of structured missing data imputation. The insights gained here may influence how missing data is treated in other complex, structured, and domain-constrained settings.

1.3 Objectives

The key objectives of this study are as follows:

1. To evaluate the performance of SOTA imputation methods across five SRT technologies with varying sizes and sparsity.

2. To identify whether any imputation method performs consistently well across all SRT platforms or exhibits technology-specific strengths.

3. To investigate the impact of feature selections on the performance of different imputation methods in the SRT context.

4. To analyze how dropout imputation affects core downstream tasks, such as spatial clustering based on several key evaluation metrics, along with the computational complexity of the selected SOTA imputation methods.

5. How the imputation methods perform when identifying the dropouts in such SRT datasets.

6. To propose a novel imputation method, SpaMean-Impute, designed to address the limitations found in current methods by better leveraging both gene expression patterns and spatial information.

1.4 Methodology

The methodology is divided in two parts: i) In the first part, the benchmarking method is discussed, where the details of the dataset preprocessing of the respective SRT technology is presented along with how the imputation methods are applied. The benchmark method allows for selecting a different number of genes, i.e., feature selections, while performing imputation. Finally, the first part of the methodology ends with discussing the evaluation metrics that can be used to analyze the performance of each imputation method. The benchmark pipeline is scalable, allowing for the easy integration of additional imputation methods.
ii) The second part of the methodology discusses the details of the proposed imputation method, 'SpaMean-Impute'. How spatial information is processed for enhanced imputation and how the method can identify valid dropouts is discussed.

1.5 Research Outcome

The possible outcomes of this study include:

- A comprehensive benchmarking of existing imputation methods, highlighting their strengths and weaknesses across various SRT technologies.

- As feature selection in high-dimensional data is a crucial step alongside imputation. The benchmark pipeline provides insights into the role of feature selection in improving or deteriorating the performance of imputation methods in SRT analysis.

- Identification of the limitations in current state-of-the-art approaches.

- The development and validation of a novel imputation method, SpaMean-Impute, that demonstrates improved performance in preserving spatial patterns and enhancing downstream analyses such as spatial clustering.

- Practical guidelines for researchers selecting feature selection approaches, imputation methods based on SRT technologies, and intended downstream applications.

1.6 Organization of the Report

The thesis is organized as follows: Chapter 1 discusses a short overview of the project alongside motivation, objectives, and outcomes. Chapter 2 discusses the background of the provided research, a detailed literature review, along with a gap analysis of the present research. Chapter 3 discussed the details of the methodology. Chapter 4 presents the results of the proposed methodology and comparison with SOTA methods. Finally, Chapter 5 provides a short discussion on the results and limitations of the projects.

Chapter 2

Background

In this chapter, SRT technologies are discussed in detail that are relevant to our project. Furthermore, the problems existing in such technology and how computational methods can solve them are discussed. Finally, a detailed summary of the similar works is presented along with a gap analysis.

2.1 Background of SRT Technologies

Transcriptomics is the study of RNA molecules (i.e., transcripts) expressed in cells, tissues, or organisms, unveiling gene activity as well as its role in biological processes and disease progression in the living body [2, 6]. Figure 2.1 illustrates the difference between three sequencing technologies: bulk RNA-seq, single-cell RNA-seq (scRNA-seq), and spatial transcriptomics. Tools like RNA sequencing (RNA-seq) analyze gene expression, recognizing patterns as well as cellular states in the tissue. In transcriptomic studies, bulk RNA-seq provides averaged gene expression from the tissue sample while elucidating global gene expression patterns and disease-specific markers [6]. However, bulk RNA-seq fails to provide accurate cell-specific functions, cellular heterogeneity, as well as spatial context among the cells [6, 12]. Conversely, scRNA-seq isolates single cells, providing intercellular heterogeneity and functionalities at the single-cell resolution, but loses overall spatial context among cells, which results in limited insights on intercellular interactions [13]. Spatially Resolved Transcriptomics (SRT) technology resolves such limitations of bulk RNA-seq and scRNA-seq by preserving the spatial relationship among the cells while providing gene expression, making it ideal for understanding how cells interact, tissue structures, disease mechanisms, and treatment strategies for the affected organisms [2, 6, 14]. SRT even allows certain reconstruction of three-dimensional tissue architecture directly from two-dimensional slices [15]. As indicated by Figure 2.1, scRNA-seq usually has matrix-shaped data with rows as genes and columns as cells, where the matrix value indicates the expression value of a particular gene in the respective cell. However, the SRT has both such matrix-shaped data along with the respective cells/spots' physical location (Figure 2.1). In this way, SRT retains crucial details about cellular heterogeneity and the spatial organization of tissues,

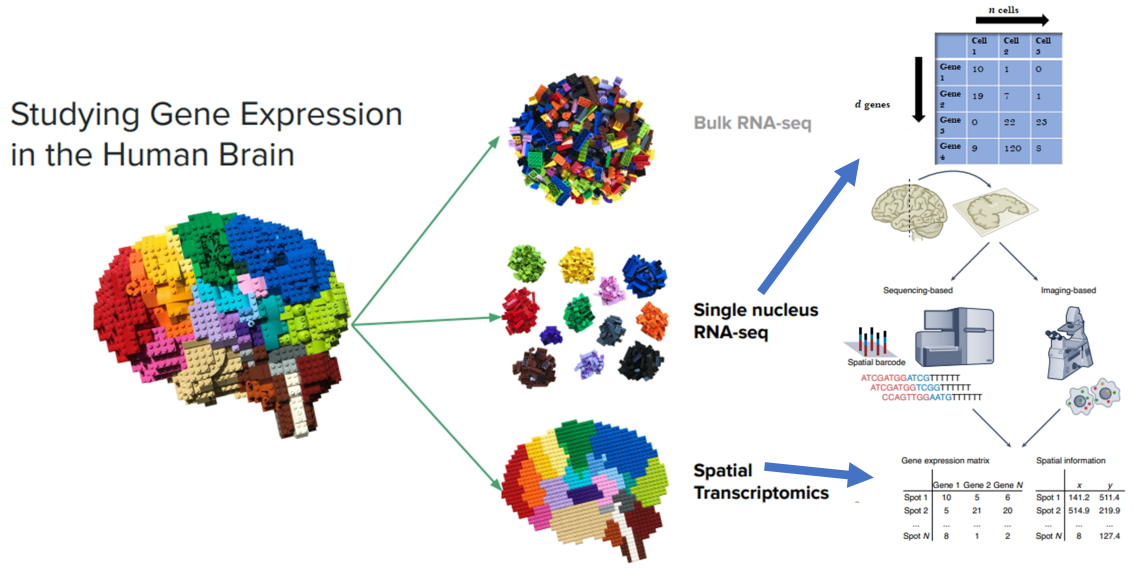


Figure 2.1: Example diagram (for illustration purposes) demonstrating the difference between Bulk RNA-seq, single-cell RNA-seq (scRNA-seq), and Spatial Transcriptomics in the Human Brain[2]. Bulk RNA-seq profiles the brain into average gene expression, while scRNA-seq technology can profile the brain tissue at the single-cell level. However, scRNA-seq loses the spatial information of the cell within the tissue, while spatial transcriptomics can retain the spatial information and profile the tissue [1].

enabling a deeper understanding of cellular interactions, functional states, and microenvironmental interactions [4, 5, 6, 8]. For example, in developmental biology, SRT allows the investigation of cellular interactions and symmetry breaking during tissue development [4], while in clinical applications, it helps uncover abnormal spatial organization in disease states such as cancer, which is critical for diagnosis and therapy selection [6]. Moreover, the power of SRT technologies in deciphering tissue complexity has enabled the generation of atlases of critical biological processes, such as tissue development or organoid formation [9, 16], and has provided insights into human disease pathogenesis [10, 11]. Despite the rapid technological progress, the field remains in its early developmental stages, with ongoing efforts like the SpaceTX consortium aiming to standardize and benchmark spatial transcriptomics methods [5, 8].

SRT technologies are generally divided into two main categories: sequencing-based methods and imaging-based methods [1, 3, 5, 6, 8]. Sequencing-based methods, such as 10x Genomics Visium, Slide-seqV2, and Stereo-seq, use next-generation sequencing (NGS) technologies to capture polyadenylated RNA from tissue samples, offering varying resolutions from tissue-wide to near-single-cell scales [4]. These techniques provide unbiased whole-transcriptome analysis, ideal for discovering novel biological mechanisms [4]. In contrast, imaging-based methods, including MERFISH, seqFISH, and STARmap, rely on in situ hybridization and advanced imaging to capture transcript data with subcellular resolution, making them particularly useful for studying intracellular organiza-

[2]Image Credit: Bo Xia `https://twitter.com/boxia7/status/1261464021322137600?s=12`

tion and molecular interactions [3, 4, 6]. Another emerging category is image-guided spatially resolved single-cell sequencing, which combines high-resolution microscopy with sequencing technologies for enhanced cellular resolution [4]. An extended classification further distinguishes between imaging-based, sequencing-based (microdissection-based), in situ sequencing-based, and in situ spatial barcoding-based approaches [8]. Imaging-based methods (e.g., smFISH, RNAscope, seqFISH, MERFISH) use in situ hybridization to target RNA molecules in cells, tissue sections, or FFPE samples [8]. Sequencing-based methods (e.g., Geo-seq, tomo-seq) rely on microdissection for transcriptome-wide data acquisition from fresh-frozen or FFPE samples [8]. In situ sequencing-based methods (e.g., STARmap, STARmap PLUS, Image-seq, ISS) directly analyze RNA transcripts within cells and tissue sections [8]. In situ spatial barcoding-based methods (e.g., 10x Genomics Visium, Slide-seq, Slide-seqV2, DBiT-seq, XYZeq, sci-Space, Stereo-seq, Pixel-seq) use spatial barcodes to map RNAs or molecules in fresh-frozen or FFPE samples [8]. From a biological standpoint, sequencing-based SRT technologies are particularly useful for mapping tissue architectures, identifying cell types, and studying their interactions across large tissue sections. These methods have been instrumental in characterizing the spatial distribution of cell types in organs like the brain, kidney, and lung [1]. The biological differences between various sequencing-based SRT methods arise primarily from their spatial resolution, RNA capture efficiency, and how well they maintain the native tissue architecture during sample preparation. For example, 10x Genomics Visiumcaptures mRNA with spatial probes printed onto slides, offering a center-to-center distance of 100 μm and a spot diameter of 55 μm. Although this is smaller than earlier ST methods, it is still larger than most single cells, leading to transcript contamination from multiple cells [8]. In contrast, Slide-seq and Slide-seqV2 use beads with spatial barcodes laid on flat surfaces, achieving a spatial resolution of 10 μm, closely approaching single-cell level; HDST further pushes this to 2 μm [9]. While Slide-seq and HDST offer superior spatial resolution, they suffer from lower RNA capture efficiencies, limiting their ability to detect enough transcripts for reliable single-cell analyses [8]. Technical challenges specific to in situ spatial barcoding-based methods include constructing effective "capturing areas" to deliver barcodes, managing the number of barcode probes per spot (affecting capture efficiency), and balancing spatial resolution (determined by spot size and distance) against capture area and throughput [8]. Different SRT platforms yield datasets with varying biological insights. Technologies like Stereo-seq exhibit the highest capturing capabilities and provide dense transcriptomic data, enhancing the detection of tissue structures and facilitating downstream tasks such as clustering, region annotation, and cell–cell communication analysis [5]. Stereo-seq also generates more sequencing reads for the same tissue region compared to other platforms like Slide-seqV2 or DBiT-seq [5]. Probe-based approaches such as Visium show improved total UMI counts, possibly due to better read-capturing efficiency or over-quantification [5]. The diversity among SRT technologies is substantial. 10x Genomics Visiumachieves a spatial resolution of 100 μm center-to-center with a spot diameter of 55 μm, capturing approximately 15,377 UMIs per 55×55 μm area in the mouse olfactory bulb, offering

multi-cell capture with relatively high UMI counts. Slide-seq achieves a 10 μm resolution but with much lower capture efficiency, recording around 59 UMIs per 10×10 μm in E12.5 mouse embryos, although it nearly reaches single-cell resolution. Slide-seqV2 maintains the 10 μm resolution but improves capture efficiency significantly, with 550 UMIs per 10×10 μm area in E12.5 mouse embryos [8]. High-Definition Spatial Transcriptomics (HDST) achieves even finer resolution at 2 μm, approaching subcellular resolution but with low capture efficiency [8]. XYZeq operates at a coarser 500 μm spatial resolution but enables single-cell capture with around 1009 UMIs and 456 genes detected per cell in mouse liver and tumor tissues [8]. sci-Space offers a spatial resolution of 222 μm and captures approximately 2514 UMIs and 1231 genes per cell in E14.0 mouse embryos [8]. Pixel-seq improves spatial granularity to about 1 μm resolution, yielding around 977 UMIs per 10×10 μm area in the mouse olfactory bulb [8]. Seq-Scope achieves an even finer 0.6 μm resolution and captures around 1000 UMIs per 10×10 μm area in mouse liver tissues [8].

Although technologies like Stereo-seq, Slide-seq, Slide-seqV2, and HDST push spatial resolution to subcellular levels, challenges remain in achieving high RNA capture efficiencies, which are crucial for accurate single-cell analyses [8]. Overall, spatial transcriptomics is a rapidly evolving field with ongoing challenges. Although significant strides have been made, including the emergence of ultra-high-resolution methods and improved capture techniques, no single method excels across all metrics, such as spatial resolution, transcriptome coverage, and RNA capture efficiency [5, 8]. Method selection depends heavily on the biological question, tissue type, and required resolution [5]. Furthermore, the absence of comprehensive benchmarking studies for sequencing-based SRT platforms complicates method comparisons [5].

2.2 Literature Review and Gap Analysis

Despite offering significant advantages over scRNA-seq by incorporating spatial context, SRT still deals with issues such as technical biases (batch effects), inconsistent preprocessing methods, and data sparsity (dropouts where expressed genes go undetected). Various computational pipelines and algorithms have been proposed over the years, such as various data imputation techniques, to solve these issues. The computational pipeline for analyzing spatial transcriptomics data involves several intricate steps: generating the spatial matrix by decoding barcodes, image registration, and cell segmentation, preprocessing, clustering, spatial domain identification, cell type annotation, and cell communication inference [8]. Despite substantial advances in spatial resolution and gene-capture chemistry, low probe/molecule capture efficiency, limited sequencing depth, and RNA degradation remain significant challenges, as evident from the recent literature [17]. This is evident across multiple platforms. For example, the widely used Visium platform continues to show systematic gene-specific biases in a recent study, missing key marker genes such as Cdc42 and Cx3cl1 despite their inclusion in the probe set [18, 19]. Likewise, even highly

sensitive imaging-based methods detect fewer transcripts overall [20, 21]. In addition, current sequencing runs across multiple platforms still fail to reach saturation, indicating that under-sequencing remains a pervasive issue [18]. These lower library sizes can make it difficult to detect important genes, particularly those that are lowly expressed [18]. Stereo-seq V2, which represents one of the most advanced technologies currently available. Stereo-seq V2 uses random primers that improve capture efficiency over traditional poly(T) primers and perform better on fragmented RNA from FFPE samples [21]. However, even Stereo-seq V2 experiences inherent limitations: because of its relatively short read length and broad genome coverage, reads covering specific regions remain low [19]. High dropout rates and sparse count matrices are still described as "inherent" to SRT data and present computational challenges such as algorithm instability [22, 23]. Finally, while data quality has improved in terms of spatial resolution and sensitivity, it remains variable and influenced by factors such as molecular diffusion, noise, and artifacts. Molecular diffusion during sample preparation can blur the spatial location of transcripts, reducing the effective resolution of the technology [20]. SRT data is also susceptible to various sources of noise, including ambient RNA, signal spillover between spots, and transcripts from incomplete cells [22], and sample preparation can introduce artifacts such as blood contamination [18]. In addition, trade-offs between sensitivity and transcript detection persist across platforms [18]. Consequently, computational imputation remains a recognized and necessary step to recover signals for lowly expressed genes and stabilize downstream analyses [17, 23]. For example, SpaNorm and other approaches "borrow" information from surrounding regions to recover signals [17]. Even with the most recent SRT technologies, the data often contains missing values for genes that are known to be expressed, particularly for lowly expressed genes in regions with low library sizes [17].

Hence, the recent studies support that a major computational hurdle of SRT data lies in data sparsity, especially in high-resolution SRT datasets that suffer from low RNA capture per spot or bead, resulting in sparse expression matrices that hinder robust downstream analyses [8]. To address sparsity, gene expression imputation becomes critical. Current approaches leverage spatial smoothing, neighboring spot information, or graph-based inference to recover missing expression values, enhancing the biological interpretability of the data [8]. However, achieving accurate imputation without introducing artifacts remains challenging, especially in regions with genuine biological heterogeneity. Researchers have proposed various imputation methods over the years. These include methods applied to general high dimensional tabular data, specialized in scRNA-seq or SRT-based data. SOTA methods in general tabular data include KNN Impute, SoftImpute, SimpleImpute, etc, [24, 25, 26]. Conversely, there exist numerous imputation methods on scRNA-seq [7, 27, 28, 29, 30]. Imputation methods used in other biological platforms are often explored on different technologies to explore their effectiveness, limitations, and adaptability. Furthermore, comprehensive benchmarking on such imputation methods used in other domains gives the researchers valuable insights and provides guidance to model efficient methods. For example, Han et. al. benchmarked scRNA-seq-based imputation methods

on scHi-C, whereas Liu et. al. explored the effectiveness of scRNA-seq-based imputation methods in scATAC-seq data [31, 32]. Han et. al. found that, unlike scRNA-seq-based imputation methods, methods for scHi-C data are very few and underexplored [31]. However, zeros are observed in large proportions in single-cell Hi-C data. The authors explored 5 imputation methods from the scRNA-seq-based domain, among which only one tended to perform well in inferring the zeros and showed improvement for clustering cells of the same types together over clustering results before imputation. Similarly, Liu et. al. benchmarked several scRNA-seq-based imputation methods on the scATAC-seq data and found that only the 'MAGIC' imputation method worked well [32]. Another limitation identified by this benchmark is that both dropouts caused by experimental factors and genuine biological zero expressions are imputed indiscriminately, making it difficult to distinguish between true biological absences and technical artifacts. In our work, along with benchmarking several imputation methods on new SRT technologies, we explored the selected method's discriminative capability between true biological zeros and technical errors (i.e., 'dropouts').

Since SRT technologies are newer and reaching single-cell resolution, there exist only a few imputation methods on SRT datasets. Hence, Standard imputation methods are yet to be developed for gene-sequencing-based SRT methods. Imputation methods in SRT datasets can be either reference-free (only uses spatial gene expression) or reference-based (needs scRNA-seq expression along with spatial gene expression) [33, 34, 35, 36, 37, 38, 39]. There are some recent works that have adapted imputation approaches integrating multimodal data. For example, a recent framework, JAMIE, performs true cross-modal imputation by directly learning to predict one modality from another [40]. It takes advantage of paired and unpaired data to infer gene expression from open chromatin regions (OCRs) or vice versa, and even predicts electrophysiological features from transcriptomics data. By explicitly coupling the encoder of one modality to the decoder of the other and enforcing alignment loss, JAMIE ensures that both modalities share a unified latent space. This design allows it to robustly use one modality as a predictor of the other, making it well-suited for situations where a modality is missing or only partially available. Another method, inClust+, extends imputation beyond simple intra-modal completion by explicitly leveraging reference modalities to fill gaps in another dataset [41]. For example, it uses scRNA-seq to impute unmeasured genes in MERFISH and employs CITE-seq references to generate protein abundance profiles for monomodal scRNA-seq. The model incorporates input and output masks to specify which genes or modalities are learned during training, giving fine-grained control over which imputation tasks are prioritized. This masking strategy enables it to transfer information effectively across modalities, making it highly adaptable to multimodal reference datasets. Monae addresses both intra- and cross-modal imputation by learning integrated cell representations capable of predicting one modality from another [42]. It uses a combination of contrastive learning and a graph variational autoencoder to create embeddings that generalize even when paired measurements are missing. With these learned embeddings, Monae reconstructs RNA counts from ATAC-seq and

vice versa, treating the imputation as a matrix product between integrated cell embeddings and feature embeddings of the target modality. This strategy enables the model to handle diverse modalities without requiring strict one-to-one paired data, broadening its applicability. Furthermore, another method, scVAEIT, is explicitly designed for multi-feature and multi-modality imputation, including transfer learning scenarios [43]. It can infer missing genes, proteins, or chromatin accessibility features from observed modalities, making it ideal for datasets with incomplete modality coverage. A key feature of scVAEIT is its mask mechanism, which lets the model systematically learn to predict missing parts from present parts during training. Because it supports trimodal settings and generalizes well to unseen datasets without retraining, scVAEIT is particularly powerful for imputing new modalities in large-scale integrative studies. scMODAL, on the other hand, focuses on cross-modality feature imputation by sharing a latent space between modalities and using encoders/decoders to bridge them [44]. It predicts protein abundance from gene expression data and imputes gene expression for scATAC-seq profiles, demonstrating versatility in both directions. The model can also target specific genes in spatial proteomics datasets, such as imputing MKI67, to enhance downstream analysis. By learning the relationships between different molecular layers, scMODAL provides a flexible and interpretable framework for predicting missing modalities across single-cell and spatial omics platforms. Though these models focus on multi-modal data during imputation, they don't focus on SRT technologies. Due to such limited number of imputation methods developed on SRT datasets, especially on the recent technologies like Stereo-seq, Slide-seqV2, XYZeq, sci-Space, evaluation of the different SOTA imputation methods is necessary. However, to our knowledge, there are no benchmarking evaluations of imputation methods on recent SRT technologies. Conversely, imputation methods' performance on SRT datasets usually involves 10x Genomics Visiumand lacks the inclusion of newer technology. Furthermore, the evaluation of these recent methods does not include important metrics such as ARI, NMI, which are important from a biological perspective [35, 36].

In this work, we propose a pipeline to explore the effectiveness of different imputation methods, including those developed for scRNA-seq data, SRT data, and traditional methods on new SRT technology. We measure the effectiveness using clustering accuracy and other relevant metrics. Imputation methods in SRTs are limited and still evolving. Hence, we select the most SOTA imputation methods from the literature based on statistical approaches and deep-learning techniques. Consequently, we select different SRT technologies with ground truth and apply the imputation methods to measure their effectiveness. Furthermore, considering the importance of spatial information of SRT datasets in imputation, we propose a simple and scalable imputation method, SpaMean-Impute, that has the ability to determine the actual dropouts and imputes the necessary dropout locations with the aggregated spot-gene expression value. Our proposed method outperforms the benchmarked imputation methods consistently in different SRT technology datasets evaluated by the downstream analysis, which shows that consideration of spatial information is a must while imputing such datasets. The main goal of this work was to

explore the effectiveness of various imputation methods in diverse SRT datasets and propose a more efficient method. The benchmarking focuses on understanding whether any imputation method performs consistently well across different SRT technologies, whether certain methods are better suited for specific technologies, and how the selection of top genes affects the performance. Furthermore, from understanding the benchmark's results, we propose a new imputation method, 'SpaMean-Impute'.

2.3 Summary

As the SRT technology is still growing, the present literature lacks of any kind benchmark pipeline of imputation methods in newer SRT technologies. Such a benchmark is necessary as the newer technologies are achieving single-cell resolution, i.e., more sparse and high-dimensional. Several imputation methods have been proposed recently for SRT technologies. However, they do not consider the newer technologies and proper evaluation metrics, considering downstream analysis of such datasets. Furthermore, imputation methods from other domain can be explored on such datasets. However, the present literature lacks any kind of benchmarking analysis on the new SRT technology. In this work, I propose a scalable benchmark pipeline that considers several diverse SOTA imputation methods along with a feature selection strategy for high-dimensonal new SRT technology. From the benchmark analysis, the results are evaluated, and an efficient imputation method, 'SpaMean-Impute' is proposed.

Chapter 3

Methodology

In this chapter, we first discuss the benchmarking pipeline and the associated datasets of the SRT technology, dataset preprocessing, imputation methods, and evaluation metrics in Section 3.1. Next, we discuss the proposed method, 'SpaMean-Impute' in Section 3.2. We demonstrate how our method has the ability to detect valid dropouts utilizing the spatial information of SRT datasets.

3.1 Proposed Methodology for Benchmarking

We present a unified computational pipeline to evaluate the imputation techniques in SRT datasets. The workflow demonstrated in Figure 3.1, begins with the acquisition of raw SRT datasets from platforms such as 10x Genomics Visium, Stereo-seq, Slide-seqV2, XYZeq, and sci-Space. These datasets are structured as spatial gene expression matrices with spatial spots as rows and genes as columns. The raw datasets undergo a preprocessing stage, which is discussed in detail in Section 3.1.2.

The preprocessing stage involves quality filtering and normalization to ensure consistency and minimize technical noise. Following preprocessing, the feature selection stage using `Scanpy's` highly variable genes allows the retention of either the top 2000 or 5000 highly variable genes, or the number of total genes [45]. The selection of genes, i.e., features, is kept in our benchmarking pipeline to observe how the number of features affects the imputation characteristics and downstream analysis. However, the choice of 2000 or 5000 features is due to biological convention in selecting highly variable features. Once gene selection is finalized, an imputation method is applied to reconstruct the missing or sparse gene expression values. Available methods include BASE, which performs no imputation, and a range of SOTA imputation techniques such as MAGIC, KNN-Impute, Simple-Impute, Soft-Impute, scVI, gimVI, and tangram (discussed in Section 3.1.3). The pipeline is designed in such a way that additional imputation methods can be easily integrated into our pipeline. The imputed gene expression matrix is then subjected to dimensionality reduction using Uniform Manifold Approximation and Projection (UMAP), followed by clustering (the 'leiden' clustering algorithm is used in this method) to identify

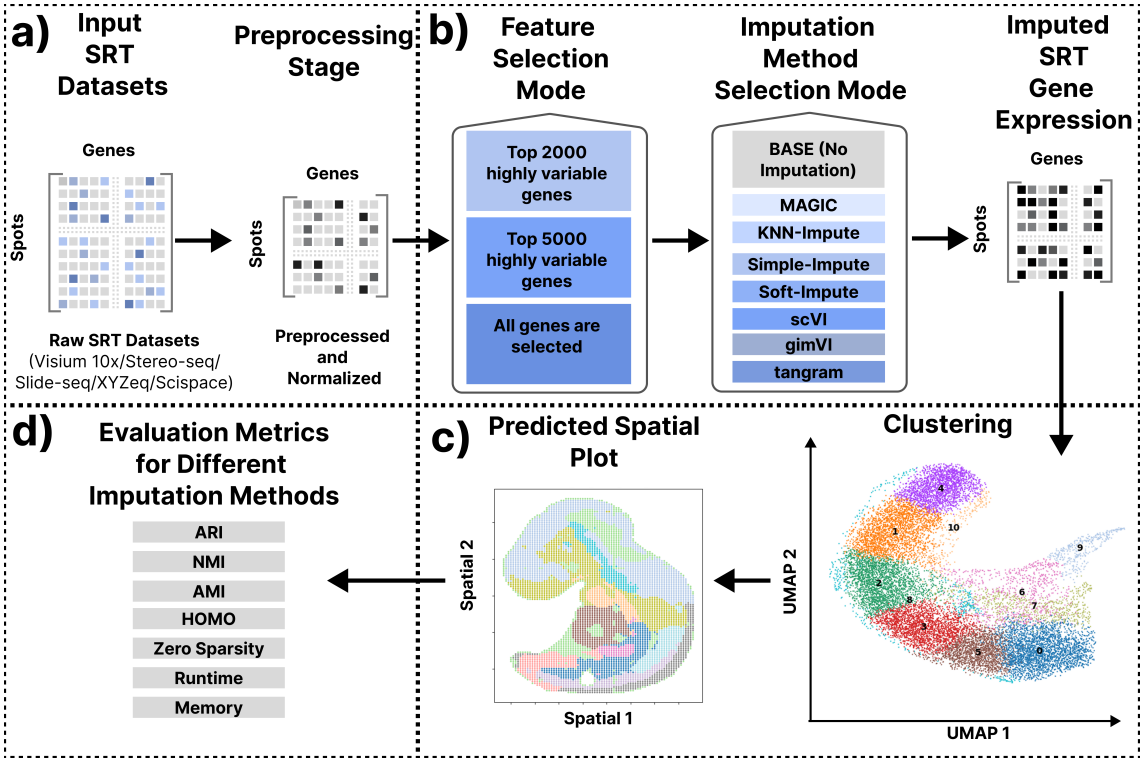


Figure 3.1: Overview of the computational pipeline for evaluating imputation methods in SRT data.

spatially coherent regions [46]. The resulting clusters are visualized both in the reduced UMAP space and in their original spatial coordinates to interpret the spatial context of biological structures and patterns. The predicted spatial plot provides spatial correspondence of cluster assignments back to the tissue sections.

Finally, the performance of each imputation method is assessed using the evaluation metrics mentioned in section 3.1.4. Additionally, zero sparsity is used to quantify the reduction of zero entries post-imputation, while runtime and memory usage are monitored to evaluate the computational efficiency and resource demands of each method. This comprehensive methodology provides a robust framework for benchmarking imputation strategies in SRT data analysis. The detailed results of this benchmarking pipeline across all the SRT datasets are discussed in the Results section.

3.1.1 Datasets

The study employed datasets from five spatial transcriptomics technologies: **10x Genomics Visium, Stereo-seq, Slide-seqV2, XYZeq, sci-Space** to benchmark imputation. The **DLPFC** and **BRCA** dataset was produced using the *10x Genomics Visium* platform, which captures transcriptome-wide expression with a spatial resolution of approximately **55** μ**m** [47, 48]. **DLPFC** dataset profiled the human dorsolateral prefrontal cortex from twelve sections across neurotypical adult donors and provided high-quality data for identifying laminar-specific gene expression, making it a common benchmark for

spatial clustering methods [47]. Conversely, the **BRCA** dataset mapped gene expression across human breast cancer tissue over **3,798 spots and 36,601 genes** [48]. **Stereo-seq (Spatiotemporal Enhanced Resolution Omics-sequencing)** is a high-resolution spatial transcriptomics technology that captures genome-wide gene expression with spatial fidelity using DNA nanoball (DNB)-patterned arrays [49]. The datasets we utilized for Stereo-seq contained bin having 50×50 DNBs spaced 715 nm apart, providing a resolution of 36×36 μm per bin [49]. For the mouse liver atlas, 878,334 non-overlapping bins covering 11.22 cm^2 were profiled and integrated with single-cell RNA-seq data from 473,290 cells (435,413 newly generated), resulting in the identification of 8,127 zonated genes and 1,140 ligand-receptor interactions (686 zonated), thereby enabling construction of a high-definition spatial-temporal atlas of liver homeostasis and regeneration [49]. The **Slide-seqV2** dataset utilized a bead-based barcoding strategy that achieves **10 μm resolution**, enabling near-cellular transcriptome profiling with markedly higher sensitivity than its predecessor [50]. It has been used to explore spatial gene expression in mouse brain tissues, including the hippocampus and embryonic neocortex, recovering a high number of UMIs per feature and allowing integration with trajectory analysis tools [50]. **sci-Space(Spatial Combinatorial Indexing Sequencing)** is another method enabling single-cell resolution across large tissue areas [51]. It integrates combinatorial indexing with spatially patterned oligo hashing on glass slides. Srivatsan et. al [51] profiled 14 sagittal sections from E14.0 mouse embryos, generating 121,909 spatially resolved single-cell transcriptomes (mean 2514 UMIs and 1231 genes per cell), covering 15,102 spatial positions (8.1 nuclei/position) over an 18×18 mm area with 7056 unique hashing spots at ∼200 μm resolution. Spatial localization was validated through fluorescent waypoints, and data were integrated with existing atlases. The approach allows cell-type-specific spatial analysis, detection of spatially patterned gene expression (e.g., *Slc6a3*, *Cyp26b1*), and reveals spatially informed pseudotemporal trajectories, enabling high-resolution spatial atlases of development. Furthermore, **XYZeq** dataset is also integrated in our benchmarking, where this dataset leveraged a two-round split-pool indexing strategy to encode spatial metadata directly into single-cell RNA-seq libraries, achieving spatial resolution of approximately **500 μm** [52]. Applied to murine tumor models, XYZeq enabled spatially resolved transcriptome profiling at single-cell resolution. Together, these datasets offer complementary resolution scales and tissue types, providing a robust foundation for benchmarking and evaluating the performance of computational methods. The details of the dataset's size are summarized in Table 3.1. We selected these datasets because they have proper annotations, which are necessary to evaluate the imputation capability through downstream analyses.

We can observe in Table 3.1 that 13 data sets from the 10x Genomics Visium platform are used, with spot counts ranging from 1,102 to 4,184 and gene counts between 7,130 and 18,968. Stereo-seq datasets include DT2_D0, DX6_D2, and FB2_D1, with spot counts ranging from 14,455 to 23,259. Slide-seqV2 comprises three datasets—WT, diabetes, and mouse—featuring spot counts between 9,008 and 22,560. For XYZeq, three datasets were

Table 3.1: SRT Datasets for 5 technologies used in our benchmarking. Dataset size indicates (Spots x Genes), and the table demonstrates both the size before preprocessing (i.e., Raw) and after preprocessing.

SL No.	**Technology**	**Dataset Name**	**Dataset Size Before Preprocessing (Spots × Genes)**	**Dataset Size After Preprocessing (Spots × Genes)**
01.	10x Genomics Visium [47, 48]	151507	4221 × 33538	2890x15274
		151508	4381 × 33538	2549 × 14507
		151509	4788 × 33538	4184 × 15892
		151510	4595 × 33538	3722 × 15111
		151669	3636 × 33538	1333 × 14539
		151670	3484 × 33538	1102 × 13923
		151671	4093 × 33538	1808 × 14976
		151672	3888 × 33538	1848 × 14816
		151673	4221 × 33538	2890 × 15274
		151674	3635 × 33538	3197 × 7130
		151675	3566 × 33538	2081 × 15148
		151676	3431 × 33538	2220 × 15434
		BRCA1	3798 × 36601	3169 × 18968
02.	Stereo-seq [49]	DT2_D0	42741 × 29075	23259 × 16865
		DX6_D2	14929 × 29075	14455 × 17864
		FB2_D1	16264 × 29075	16211 × 18252
03.	Slide-seqV2 [50]	WT	31659 × 24450	9008 × 16598
		diabetes	27194 × 24420	9435 × 16884
		mouse	41786 × 4000	22560 × 3919
04.	sci-Space [51]	GSE166692	122278 × 52636	9517 × 24879
05.	XYZeq [52]	GSE164430	7505 × 52025	2914 × 25556
		GSM5009529	6447 × 121472	4293 × 19390
		GSM5009539	2703 × 9776	2501 × 8776

used, including GSE164430, with a maximum gene count of 25,556. Lastly, sci-Space includes one dataset (GSE166692) containing 9,517 spots and 24,879 genes. We can observe from the dataset size that Stereo-seq and Slide-seqV2 have the maximum number of spots detected in tissues, whereas 10x Genomics Visium and XYZeq have the lowest. This diverse selection ensures broad coverage across widely used SRT technologies.

3.1.2 Dataset Preprocessing

All SRT datasets undergo a standardized quality control (QC) and preprocessing workflow, with slight adjustments based on dataset-specific characteristics such as size and quality. The preprocessing was performed using the `scanpy` library [45].

First, mitochondrial genes were annotated by identifying gene names starting with "MT-". QC metrics per cell were calculated, including the total counts of UMI, the number of genes detected, and the percentage of counts attributed to mitochondrial genes. Spots with extremely low or high total counts (likely representing empty or doublet-containing barcodes) were filtered. The filtering thresholds varied across datasets: the `min_counts` parameter ranged from 50 to 500, depending on the dataset's overall quality and sequencing depth, while `max_counts` was typically set around 35,000. Cells with more than 20% mitochondrial gene expression were excluded to remove potentially unhealthy or dying cells. Next, genes detected in fewer than a threshold number of spots were removed to reduce noise; the `min_cells` threshold was set between 10 and 50, depending on the dataset. Following filtering, the datasets were normalized using total-count normalization and log-transformed after adding a pseudocount of 1. These preprocessing steps ensured that downstream analyses were performed on high-quality, comparable gene expression matrices across technologies. Finally, the pre-processed dataset from all raw versions of the datasets is saved in `'.h5ad'` AnnData format to facilitate easier handling in subsequent benchmarking analyses. The size of the preprocessed datasets after preprocessing can be observed in Table 3.1 with respect to the size of the dataset before preprocessing (i.e., raw versions). All preprocessed datasets can be found in the 'Data Availability Sections' and in the GitHub link [53].

3.1.3 Imputation Methods included in the Pipeline

We applied seven imputation methods: **MAGIC, KNN Impute, SoftImpute, SimpleImpute, scVI, gimVI, and tangram** in our study. Each method leverages distinct statistical, machine learning, or deep learning principles to recover missing gene expression signals and improve downstream analysis. These methods were chosen based on their strong performance in previous benchmarks, their broad adoption in the single-cell research community, and their ability to compute tabular data. These methods consist of generalized imputation characteristics (KNN Impute, SimpleImpute), specially designed for scRNA-seq datasets (such as MAGIC, SoftImpute, scVI) or designed SRT datasets

(gimVI, tangram). Such diverse methods are selected to assess their capability in imputing dropouts in SRT datasets.

MAGIC (Markov Affinity-based Graph Imputation of Cells) is a diffusion-based imputation method that constructs a graph where nodes represent cells, and edges reflect cell-to-cell similarity based on gene expression profiles [54]. The algorithm applies a Markov diffusion operator to spread gene expression information across the graph, effectively smoothing expression values and recovering gene–gene relationships lost due to dropout events. This method is particularly useful for visualizing continuous trajectories in cell differentiation processes.

MAGIC imputes missing transcriptomic data by diffusing observed gene expression across a graph of similar cells. It constructs a Markov transition matrix from spot-spot affinities and propagates information via matrix multiplication over a user-defined diffusion time t as demonstrated below, yielding smoothed and biologically plausible expression profiles.

Algorithm 3.1: Pseudo code of MAGIC

Input: Count matrix $D \in \mathbb{R}^{n \times m}$, diffusion time t

Output: Imputed and rescaled matrix D_{imputed}

$D \leftarrow \text{preprocess}(D)$

$\text{Dist} \leftarrow \text{compute_distance_matrix}(D)$

$A \leftarrow \text{compute_affinity_matrix}(\text{Dist})$

$M \leftarrow \text{compute_markov_affinity_matrix}(A)$

$D_{\text{imputed}} \leftarrow M^t \cdot D$

$D_{\text{rescaled}} \leftarrow \text{Rescale}(D_{\text{imputed}})$

return D_{rescaled}

The algorithm scales approximately linearly with the number of cells m due to the kNN-based graph construction, while matrix operations depend on the sparsity and dimensionality of the affinity matrix.

KNN Impute relies on the principle that similar cells exhibit similar gene expression patterns. For each cell with missing values, the algorithm identifies the K nearest neighbors based on Euclidean distance in gene expression space [24, 26]. Missing entries are imputed by averaging the corresponding values across these neighbors. While simple, this method provides a non-parametric and intuitive baseline for imputing sparse gene matrices, especially in datasets with a clear clustering structure. However, this method may fail in sparse tabular data due to sparsity among the K-nearest neighbors.The algorithm of KNN Impute is shown below:

The KNN Impute algorithm fills missing values in a dataset by identifying the k nearest neighbors of a sample with missing values and using their corresponding feature values (usually averaged) for imputation. It relies on distance computation across all samples and features. For each missing entry, distances to all other samples must be computed, leading to $O(n \cdot d)$ per query, where n is the number of samples and d is the number of

Algorithm 3.2: Pseudo code of KNN Impute

Input: Expression matrix X, number of neighbors k
Output: Imputed matrix X'
1. Replace missing values (zeros) with NaN.
2. For each missing entry x_{ij} in X: a. Find the k nearest neighbors of sample i using Euclidean distance.
3. b. Compute the average of the corresponding feature values among neighbors.
4. c. Replace x_{ij} with the computed average.
5. 3. Return the imputed matrix X'.

features. For all entries, this results in an overall complexity of approximately $O(n^2 d)$, which can become expensive for large datasets.

SoftImpute implements a matrix completion strategy via low-rank approximation. Specifically, it iteratively performs soft-threshold singular value decomposition (SVD) to minimize the reconstruction error over observed entries while shrinking singular values to enforce a low-rank structure [25]. This approach assumes the gene expression matrix is approximately low rank and is well-suited for datasets with random missingness and low technical noise. However, this approach assumes '`NaN`' value for the dropouts, i.e., requiring prior knowledge of the dropouts. We converted all the zero values in the SRT datasets to `'NaN'` and then applied SoftImpute.The SoftImpute algorithm performs matrix completion where missing values are iteratively filled. Missing values are first initialized, followed by repeated application of low-rank approximation until convergence. The pseudo code is shown below:

Algorithm 3.3: Pseudo code of SoftImpute

Input: Matrix X, max_rank, max_iters
Initialize missing entries with column means;
for iter = 1 to max_iters **do**
    Perform low-rank SVD on X with rank $\leq$ max_rank;
    Reconstruct X using soft-thresholded singular values;
    Replace only missing entries with reconstructed values;
Output: Imputed matrix X

Each iteration requires a truncated SVD computation, which is $O(ndr)$ where r is the chosen rank ($r \leq$ max_rank). For T iterations, the total complexity is $O(T \cdot ndr)$. While more costly than SimpleImpute, it is typically faster and more scalable than full matrix completion methods.

SimpleImpute from scikit-learn is also employed as a benchmark imputation method due to its simplistic nature in imputing tabular data. It fills missing values using basic statistical measures such as the mean, median, or most frequent value calculated independently for each gene across all cells [26]. This univariate approach does not incorporate any structure or dependencies within the data, such as gene–gene correlations or cell–cell similarities. Despite its simplicity, it provides a useful baseline for assessing the relative performance of more sophisticated imputation models. SimpleImpute similarly

requires prior knowledge of the dropouts and replacement of the dropouts with '`NaN`'. The SimpleImpute algorithm replaces missing values with a basic statistic such as the mean, median, or most frequent value of the column, as shown below. It is a straightforward approach requiring only a pass over the data to compute the chosen statistic.

Algorithm 3.4: Pseudo code of SimpleImpute

Input: Matrix X, strategy = "mean"

for each column j in X **do**

  Compute the mean of non-missing values in column j;

  Replace all missing values in column j with the computed mean;

Output: Imputed matrix X

The computation for SimpleImpute of mean requires $O(n)$ per column, where n is the number of samples. Since all d columns are processed, the total complexity is $O(nd)$, which makes SimpleImpute highly efficient even for large datasets.

scVI (single-cell Variational Inference) is a deep generative model based on a variational autoencoder framework [55]. It models gene expression counts as samples from a negative binomial distribution, parameterized by latent variables inferred from the data. scVI accounts for batch effects and overdispersion and captures complex, nonlinear gene dependencies through its learned latent space. This probabilistic framework allows for robust denoising, imputation, and latent representation of high-dimensional scRNA-seq data. The framework imputes scRNA-seq data by first encoding each cell's transcriptome into a low-dimensional latent representation, correcting for nuisance variation such as sequencing depth, and then decoding this representation through a probabilistic model to recover expected counts. Specifically, the method integrates spatially-aware zero-inflated negative binomial (ZINB) modeling to differentiate between technical zeros and biological expression absence. The imputed values are obtained by scaling the batch-corrected latent representation of each gene with the estimated library size of the cell, thereby generating normalized and biologically meaningful estimates. The proposed method, however, did not provide any pseudocode or any time complexity. Below, the process of scVI is discussed briefly in pseudo code format along with the details found from the work on computational performance:

Algorithm 3.5: Pseudo code of scVI Imputation

Input: Gene expression matrix X (cells $\times$ genes)

Output: Imputed matrix X'

1 1. Encode each cell n:

2 a. Map raw expression profile to latent vector z_n via encoder neural network.

3 b. Estimate nuisance variable ℓ_n representing sequencing depth/capture efficiency.

4 2. Decode latent representation:

5 a. Use decoder neural networks to map (z_n, ℓ_n) to gene-level distribution parameters.

6 b. Assume a Zero-Inflated Negative Binomial (ZINB) distribution for counts.

7 3. Estimate normalized transcript proportions ρ_{gn}.

8 4. Compute imputed counts: $X'_{gn} = \rho_{gn} \cdot \ell_n$.

9 5. Return imputed expression matrix X'.

The scVI model is computationally efficient due to its variational Bayesian inference framework combined with stochastic optimization. By operating on fixed-size mini-batches (e.g., $M = 128$ cells per iteration), scVI avoids loading the full dataset into memory, enabling scalability to datasets with over one million cells. Training can be completed in under one hour for very large datasets (e.g., BRAIN-LARGE with 1.3M cells) when early stopping is used. The model architecture is based on deep neural networks with configurable depth (one to three layers) and width (128–256 nodes), and requires tuning only a few dataset-specific hyperparameters (learning rate, number of layers, layer width). Dispersion parameters θ_g for each gene are globally estimated, while latent variables are cell-specific. Overall, the time complexity is dominated by the forward and backward passes of the encoder–decoder networks over mini-batches, making the runtime approximately linear in the number of cells and robust to scale, while the memory complexity remains bounded by the mini-batch size.

Conversely, **gimVI** extends scVI to jointly model paired or unpaired single-cell and SRT data [33]. It incorporates a multi-view Variational Autoencoder (VAE) structure where one encoder processes the spatial modality and another encoder handles the single-cell modality. A shared latent space is learned, enabling the model to predict unmeasured genes in spatial data by transferring information from the well-characterized single-cell reference. The model integrates spatial transcriptomics with scRNA-seq data by projecting both modalities into a shared latent space. Missing genes in spatial data are imputed by inferring the latent representation z from spatial expression and decoding it through a neural network trained on scRNA-seq, thereby recovering counterfactual expression values.

Algorithm 3.6: Pseudocode of gimVI Imputation

Input: scRNA-seq data X, spatial data X', latent dimension d, gene set G, observed spatial gene set G', max iterations T

Output: Imputed spatial expression matrix $\hat{X}'$

1. Initialize latent variables $z_n \sim \mathcal{N}(0, I_d)$ for each cell n;
2. Assign modality indicator $s_n = 0$ (scRNA-seq) or $s_n = 1$ (spatial);
3. Train encoder networks $q_\phi(z, \ell \mid x, s)$ and $q_\psi(z \mid x', s)$ using variational inference;
4. For each iteration $t = 1$ to T:
   1. Sample z from posterior $p_\Theta(z \mid x, s)$;
   2. Decode z via neural network $f_\eta(z, s)$ to obtain normalized gene frequencies ρ_n;
   3. For scRNA-seq ($s = 0$): model $x_{ng} \sim \text{ZINB}(\ell_n \rho_{ng}, \theta_g)$;
   4. For spatial ($s = 1$): model $x'_{ng'} \sim \text{Poisson/NB}(\ell'_n \rho'_{ng'}, \theta'_{g'})$;
5. To impute missing gene $g \in G - G'$:
   1. Sample $z \sim p_\Theta(z \mid x', s = 1)$;
   2. Predict $\hat{x}^*_g(z) = f_{\eta g}(z)$ as imputed expression;
6. Return imputed matrix $\hat{X}'$;

gimVI is a generative probabilistic model parameterized by $\Theta = \{\eta, \nu, \theta, \theta'\}$, where η and ν correspond to neural network weights, and θ, θ' are gene-specific inverse dispersion parameters. The latent space dimension d is typically set to 10, and a hyperparameter κ balances modality alignment with biological signal preservation. The model leverages stochastic variational inference for efficient optimization and uncertainty quantification. In practice, gimVI scales efficiently: fitting real datasets such as mSMS (3k scRNA-seq + 4.4k osmFISH cells) and mPFC (71k scRNA-seq + 3.7k starMAP cells) required only a few minutes. While no closed-form Big-O complexity is provided, the runtime grows approximately linearly with the number of cells due to mini-batch training, and parameter counts are dominated by the neural network size. Its generative formulation also provides confidence estimates for imputed values, with higher uncertainty on harder-to-predict genes. Compared with scVI, which only models scRNA-seq data, gimVI extends the framework to spatial transcriptomics, enabling cross-modality integration and imputation of unmeasured genes in spatial assays.

tangram is another mapping framework that aligns scRNA-seq profiles to SRT coordinates by solving an optimization problem that minimizes the difference between the observed spatial gene expression and a weighted sum of single-cell profiles [34]. It operates under the assumption that spatial tissue sections can be explained as mixtures of cell types or states observed in scRNA-seq data. 'tangram' produces a probabilistic assignment ma-

trix, enabling high-resolution spatial reconstruction of cellular composition across tissue sections. The framework aligns sc/snRNA-seq data with spatial transcriptomics by learning a mapping matrix that links cells to spatial voxels. Once the alignment is established, Tangram imputes unmeasured genes in the spatial data by transferring transcriptome-wide annotations from the single-cell profiles into space through matrix multiplication.

Algorithm 3.7: Pseudocode of tangram Imputation

Input: sc/snRNA-seq matrix S (cells $\times$ genes), spatial matrix G (voxels $\times$ genes), density vector $\vec{d}$, number of epochs T

Output: Imputed spatial expression matrix A^{transf}

1. Initialize unconstrained mapping matrix $\tilde{M}$ (cells $\times$ voxels);
2. Compute mapping matrix $M = \text{softmax}(\tilde{M})$ such that $\sum_j M_{ij} = 1$;
3. Define objective function $\Phi(\tilde{M})$ with three terms:
   1. Density term: minimize KL divergence between predicted and expected voxel densities;
   2. Gene/voxel expression term: maximize cosine similarity between $M^T S$ and G;
   3. Voxel/gene expression term: maximize cosine similarity across voxels;
4. Optimize $\Phi(\tilde{M})$ using gradient-based optimization (PyTorch);
5. If applicable, learn filter vector $\vec{f}$ to select a subset of cells, minimizing $\Phi(\tilde{M}, \vec{f})$;
6. After convergence, compute imputation for unmeasured genes $g \in G - G'$: a. Form annotation matrix A with missing genes from sc/snRNA-seq;
   b. Compute imputed expression $A^{transf} = M^T A$;
   c. If filter used: $A^{f\ transf} = M^T(\text{diag}(\vec{f}) \cdot A)$;
7. Return imputed matrix A^{transf};

tangram does not rely on tunable hyperparameters but learns the mapping matrix M (and optionally a filter vector $\vec{f}$) through stochastic gradient descent. Its efficiency allows the mapping of ∼100,000 cells in a few minutes on a single P100 GPU. Convergence typically occurs within 300–6,000 epochs depending on the spatial technology (e.g., 300 for Visium, 1,200 for MERFISH/STARmap, 6,000 for deterministic Visium mapping). A subset of high-quality marker genes is used for training (e.g., ∼1,000 genes), after which genome-wide prediction is achieved via annotation transfer. Unlike scVI and gimVI, which are generative probabilistic models, Tangram focuses on deterministic alignment and transfer, enabling efficient expansion of measured spatial profiles to full transcriptomes without explicitly modeling uncertainty.

These diverse imputation methods are used to impute the five SRT technologies' 23 datasets and subsequently evaluate these methods' imputation capability.

3.1.4 Metrics to evaluate the Imputation Methods

To evaluate the performance of imputation methods on SRT datasets, we employ a set of well-established clustering evaluation metrics. These metrics compare predicted clusters, P, (e.g., using the Leiden method) obtained from imputed gene expression profiles against known biological annotations i.e., true spatial domain labels or manually annotated layers, T.

Adjusted Rand Index (ARI): ARI quantifies the agreement between predicted clustering assignments, P and true biological labels, T [56]. It corrects for chance grouping by comparing the number of cell/spot pairs that are consistently grouped or separated in both partitions. ARI ranges from -1 (no agreement) to 1 (perfect agreement), with 0 representing chance-level agreement. The equation of ARI is given as follows [57]:

$$ARI = \frac{\text{Index} - \text{Expected Index}}{\text{Max Index} - \text{Expected Index}} \tag{3.1}$$

where

$$\text{Index (Measures actual agreement between P and T)} = \sum_{ij} \binom{n_{ij}}{2}, \tag{3.2}$$

$$\text{Expected Index (Measures expected agreement between P and T by chance)} = \frac{\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}}{\binom{n}{2}}, \tag{3.3}$$

$$\text{Max Index (Measures best possible agreement between P and T)} = \frac{1}{2}\left(\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right) \tag{3.4}$$

Here, n_{ij} denotes the number of spots belonging to both ith true class, T, and jth predicted cluster, P. This value reflects how many matched pairs exist between the true and predicted labels. Similarly, a_i represents the number of samples in the ith true class, and b_j represents the number of samples in the jth predicted label. The total number of samples is denoted by n. The corresponding pairs among true and predicted labels are computed for expected agreement by random assignments. By calculating the actual agreement between true pairs, T, and predicted pairs, P, between spots over the random assignments, the numerator of ARI measures how much better the clustering is than random chance. The denominator captures the maximum number of pairwise agreements possible, given the distributions of true and predicted labels. By normalizing in this way, ARI provides a chance-corrected measure of clustering accuracy, where 1 indicates perfect agreement, and 0 indicates agreement no better than random. Although ARI is bounded above by 1, it can take negative values when the clustering result is worse than random assignment.

Normalized Mutual Information (NMI): NMI measures the mutual dependence between predicted clusters and ground truth labels. It is based on the Mutual Information (MI), which quantifies how much knowing the predicted label reduces the uncertainty of the true label [58]. NMI scales MI between 0 and 1, making scores comparable across datasets. The equation of NMI is as follows:

$$\text{NMI} = \frac{\text{MI}(T, P)}{\sqrt{H(T) \cdot H(P)}} \tag{3.5}$$

$$\text{MI}(T, P) = H(T) - H(T|P) \tag{3.6}$$

Where T and P are the true and predicted label sets, respectively, and $H(\cdot)$ denotes Shannon entropy. $H(T)$ is the entropy of the true labels, and $H(T|P)$ is the conditional entropy of true labels given predicted clusters. The entropy of true labels $H(T)$ quantifies the uncertainty or diversity in the distribution of ground truth classes—higher entropy means more evenly spread labels. Conditional entropy $H(T \mid P)$ measures the remaining uncertainty in the true labels after knowing the predicted clusters, reflecting how informative the predictions are. Their difference, the mutual information $\text{MI}(T, P)$, quantifies the reduction in uncertainty about T when P is known [58].

Adjusted Mutual Information (AMI): AMI improves upon NMI by correcting for the expected MI under random assignments [58]. This makes it especially robust for unbalanced or skewed class distributions, which are common in real SRT datasets. The equation for AMI is as follows:

$$\text{AMI} = \frac{\text{MI}(T, P) - \mathbb{E}[\text{MI}(T, P)]}{\text{avg}(H(T), H(P)) - \mathbb{E}[\text{MI}(T, P)]} \tag{3.7}$$

Here, $\mathbb{E}[\text{MI}(T, P)]$ is the expected mutual information between T and P under a random model.

Homogeneity (HOMO): Homogeneity assesses whether each predicted cluster contains only spots from a single ground truth label. A perfectly homogeneous clustering (score = 1) means that no cluster mixes spots from different biological classes. The equation for HOMO is shown below [59]:

$$\text{Homo} = 1 - \frac{H(T|P)}{H(T)} \tag{3.8}$$

Where $H(T|P)$ is the conditional entropy of predicted labels T given the true labels P. Lower conditional entropy (i.e., higher homogeneity) implies that each predicted cluster

is dominated by a single ground truth class.

Among these metrics, AMI is theoretically preferred over NMI, although both typically yield similar results. Compared to ARI, AMI is more suitable when cluster sizes are unbalanced or include small clusters [58]. Since SRT data often contain rare cell populations, AMI is generally more appropriate, whereas ARI is preferable when cluster sizes are nearly uniform [59]. Together, these metrics provide a comprehensive view of clustering quality, including label agreement, information preservation, and structural integrity of the spatial data after imputation.

3.2 Proposed Imputation Framework-SpaMean-Impute

In this work, we also propose a simple imputation framework, 'SpaMean-Impute' that incorporates the spatial information in the SRT datasets explicitly for imputation. The overall method can be divided into four parts: i) Finding the valid dropouts and true biological zeros, ii) similar spots determination of each spot using spatial coordinates, iii) Zero ratio determination across similar spots for differentiating between dropouts and biological zeros, and finally, iv) Imputing the valid dropouts. The proposed methodology is depicted in Figure 3.2.

Finding the Indices of Zeros in the Spot-Gene Matrix

The first task is to determine the indices of zeros in the datasets. As demonstrated in Figure 3.2, a spot-gene expression matrix $\mathbf{X} \in \mathbb{R}^{n \times g}$ is first processed, where each row represents a spatial location (or spot) and each column represents a gene. After applying the top genes selection mode, we get another matrix, $\mathbf{X'} \in \mathbb{R}^{n \times g'}$ (g' is the number of genes selected for the final imputation stage). Now, the goal is to identify all zero-valued entries. These zeros often correspond to dropouts and are targeted for potential imputation. We define a mapping from each spot $i \in \{1, 2, \ldots, n\}$ to the set of gene indices $j \in \{1, 2, \ldots, g\}$ for which the expression value $X'_{ij} = 0$. Let $\mathbb{1}_{\{X_{ij}=0\}}$ be the indicator function that returns 1 if $X_{ij} = 0$ and 0 otherwise. Then, for each spot i, the set of zero-valued gene indices is defined as:

$$\mathcal{Z}_i = \{j \in \{1, 2, \ldots, g\} \mid X_{ij} = 0\} = \begin{cases} 1 & \text{if } X_{ij} = 0 \\ 0 & \text{otherwise} \end{cases}$$

These sets $\mathcal{Z}_i$ are stored in a dictionary-like structure:

$$\text{zero_dict_spot}[i] = \mathcal{Z}_i$$

Additionally, to reduce computational complexity, any gene j with zero expression across all spots is first removed to ensure that only informative genes are retained.

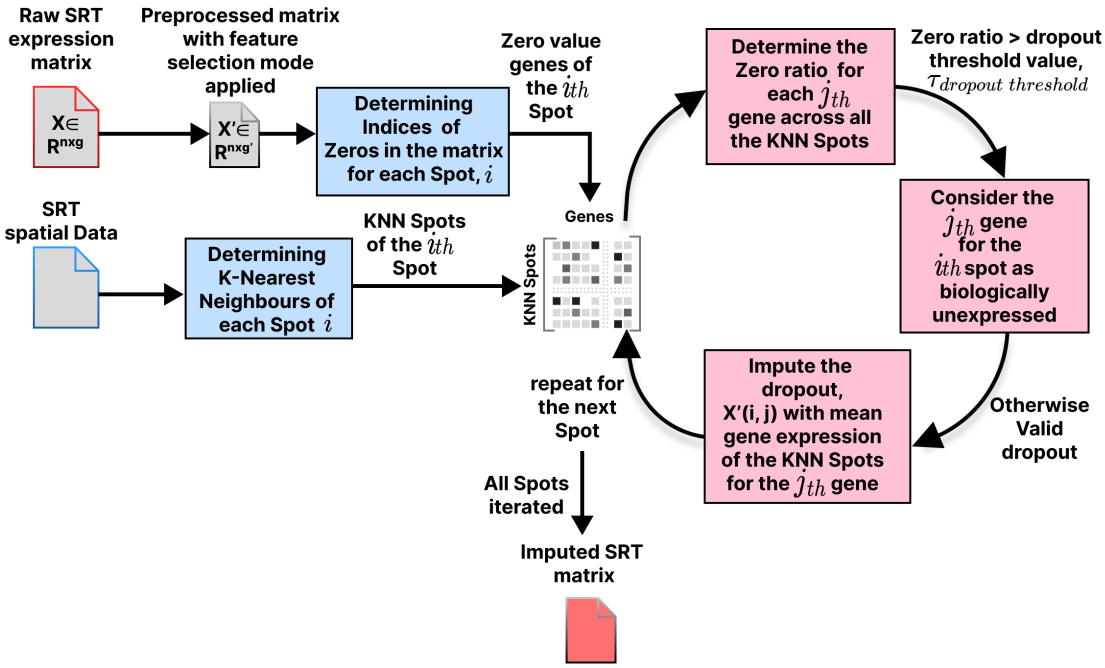


Figure 3.2: This diagram outlines the steps used in SpaMean-Impute to impute zero gene expression values in SRTs. Starting with a filtered expression matrix and spatial coordinates, the method identifies zero entries per spot and locates the k nearest neighbors. For each zero, a sub-matrix of neighbor values is extracted, and the zero ratio is computed. If the ratio is below a set threshold, the value is imputed using the average of non-zero neighbor values. This approach recovers likely technical dropouts while preserving true biological sparsity.

Determining the Nearest Spots Using Spatial Coordinates

The spot-gene matrix of the tissue, $\mathbf{X}$ has additional data, that is the spatial information. The next step, demonstrated in Figure 3.2, is to determine nearest neighbors using spatial coordinates in the SRT datasets, where each spot in SRT is associated with a 2D spatial coordinate representing its physical location on the tissue slide. Let the coordinate matrix be denoted by:

$$\mathbf{S} \in \mathbb{R}^{n\times 2}, \quad \text{where} \quad \mathbf{S}_i = (x_i, y_i)$$

for spot $i = 1, 2, \ldots, n$. To perform local neighborhood-based imputation, we compute the k nearest spatial neighbors for each spot using Euclidean distance. The Euclidean distance d_{ij} between any two spots i and j is given by:

$$d_{ij} = \|\mathbf{S}_i - \mathbf{S}_j\|_2 = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

For each spot i, we find a set of k nearest neighbors (excluding the spot itself), denoted by:

$$\mathcal{N}_i^{(k)} = \text{argsort}_{j\neq i}(d_{ij})[1:k]$$

This results in a neighbor dictionary:

$$\text{neighbors_dict}[i] = \mathcal{N}_i^{(k)}$$

This spatially-derived neighbor list serves as the foundation for imputing gene expression values from nearby spots, leveraging the spatial smoothness inherent in tissue structure.

Submatrix Extraction and Zero-Ratio Calculation Across KNN Spots

After determining the k nearest neighbors for each spot i, denoted as $\mathcal{N}_i^{(k)}$, and identifying the indices of genes that have zero expression in spot i, denoted as:

$$\mathcal{Z}_i = \{j \mid X_{ij} = 0\}$$

a submatrix $\mathbf{X}_i^{\text{sub}} \in \mathbb{R}^{k\times|\mathcal{Z}_i|}$ is extracted containing the expression values of those genes across k neighbors of spot i:

$$\mathbf{X}_i^{\text{sub}} = [X_{nj}] \quad \text{for} \quad n \in \mathcal{N}_i^{(k)},\ j \in \mathcal{Z}_i$$

To assess the reliability of imputing the particular spot's gene expression, we compute the `zero ratio` for each gene j across its neighborhood. The algorithm checks if a neighbor of ith spot has zero entry for jth gene. The total number of zeros found is divided by the total number of neighbor spots, k.:

$$\text{zero_ratio}_{ij} = \frac{\sum_{n \in \mathcal{N}_i^{(k)}} \mathbb{1}_{X_{nj}=0}}{k}$$

This ratio helps in deciding whether the gene j in spot i should be imputed. Only if $\text{zero_ratio}_{ij} \leq \delta$, where δ is the dropout threshold (e.g., $\delta = 0.4$), the imputation is allowed. This is from the biological perspective that higher zero ratios may indicate true zero expression, where lower zero ratios in a gene across KNN spots may indicate technical error in the data and need imputations. Hence, the set of valid genes to be imputed for spot i is:

$$\mathcal{Z}_i^{(\text{valid})} = \{j \in \mathcal{Z}_i \mid \text{zero_ratio}_{ij} \leq \delta\}$$

Mean-Based Imputation Using KNN Neighbors

Once the set of valid zero-expression genes for each spot i has been filtered using the zero-ratio criterion (i.e., $\mathcal{Z}_i^{(\text{valid})}$), the next step is to impute those missing values using the average expression values from the neighboring spots. $\mathcal{N}_i^{(k)}$ denote the k nearest neighbors of spot i and $\mathbf{X}' \in \mathbb{R}^{n \times g'}$ is the gene expression matrix. For each gene $j \in \mathcal{Z}_i^{(\text{valid})}$ of spot i, the imputed expression value is computed as the mean of non-zero expression values of gene j across the neighbors of spot i:

$$X_{ij}^{(\text{imputed})'} = \frac{\sum_{m \in \mathcal{N}_i^{(k)} \wedge X'_{mj} \neq 0} X'_{nj}}{|\mathcal{N'}_i^{(k)}|}$$

where,

$$\mathcal{N'}_i^{(k)} = \left\{ \mathcal{N}_i^{(k)} \middle| X'_{nj} \neq 0 \right\}$$

If all neighbors have zero values for gene j (i.e., the denominator is zero), then the imputation for that entry is skipped to avoid division by zero:

$$X_{ij}^{(\text{imputed})'} = \begin{cases} \text{mean of non-zero } X_{nj} & \text{if } \exists\, n \in \mathcal{N}_i^{(k)} \text{ such that } X_{nj} \neq 0 \\ 0 \text{ (unchanged)} & \text{otherwise} \end{cases}$$

After computing the imputed values for all $j \in \mathcal{Z}_i^{(\text{valid})}$, they are inserted back into the original gene expression matrix:

$$X'_{ij} \leftarrow X_{ij}^{(\text{imputed})'}, \quad \forall j \in \mathcal{Z}_i^{(\text{valid})}$$

This approach assumes that the non-zero expression values from neighboring spots can provide reliable estimates for the missing gene expression values at spot i, particularly when the surrounding cells show consistent expression for that gene. This process continues until it iterates over all the spots. The pseudo code of our proposed method is shown in Algorithm 3.8. This algorithm performs local imputation of missing gene expression values

by identifying the indices of genes with zero expression and the k nearest neighboring spots. These neighbors serve as a local reference for estimating missing values. For each zero-valued gene in a spot, the algorithm calculates the proportion of zero values (zero ratio) for that gene across the neighborhood. If the ratio is below a predefined threshold δ, the gene is considered eligible for imputation by averaging the non-zero values of the gene from the neighboring spots. This strategy prevents over-smoothing and preserves biologically meaningful sparsity, as it avoids imputing genes that are likely truly unexpressed in a local context. The proposed SpaMean-Impute first removes all-zero genes ($O(nd)$), then computes k-nearest neighbors for each of the n spots based on spatial coordinates, which dominates the complexity at $O(n^2)$ using pairwise distances. For each zero entry, averaging over neighbors adds $O(nk)$, yielding an overall complexity of approximately $O(n^2 + nk)$. This is higher than SimpleImpute ($O(nd)$) but more efficient than KNN Impute imputation ($O(n^2d)$) and SoftImpute($O(Tndr)$), while deep generative models like scVI and gimVI, or mapping-based methods like tangram, which achieve near-linear scaling with cells via mini-batch optimization or SGD.

Algorithm 3.8: Pseudo code of the proposed method

Input: Expression matrix $\mathbf{X} \in \mathbb{R}^{n \times g}$;
Spatial coordinate matrix $\mathbf{S} \in \mathbb{R}^{n \times 2}$;
Number of nearest neighbors k;
Dropout threshold $\delta \in [0, 1]$
Output: Imputed expression matrix $\mathbf{X}'$

Step 1: Remove all-zero genes
Remove column j from $\mathbf{X}$ if $\sum_{i=1}^{n} \mathbb{1}_{X_{ij} \neq 0} = 0$.
Step 2: Identify zero indices
foreach spot $i \in \{1, \ldots, n\}$ **do**
    $\mathcal{Z}_i \leftarrow \left\{ j \mid X'_{ij} = 0 \right\}$
Step 3: Find k nearest spatial neighbors
foreach spot $i \in \{1, \ldots, n\}$ **do**
    Compute $d_{ij} = \|\mathbf{S}_i - \mathbf{S}_j\|_2$ for all $j \neq i$
    $\mathcal{N}_i^{(k)} \leftarrow$ indices of k closest spots to i
Step 4: Impute eligible zero entries using neighbors
foreach spot i **do**
    foreach gene $j \in \mathcal{Z}_i$ **do**
        foreach neighbor $m \in N_i^k$ **do**
            Extract values $\{X'_{nj} \mid m \in \mathcal{N}_i^{(k)}\}$
            Compute zero ratio: $\text{zero_ratio}_{ij} = \frac{1}{k} \sum_{n \in \mathcal{N}_i^{(k)}} \mathbb{1}_{X'_{mj}=0}$
        if $\text{zero_ratio}_{ij} \leq \delta$ **then**
            Let $\mathcal{N}'^{(\mathrm{k})}_i = \{n \in \mathcal{N}_i^{(k)} \mid X'_{nj} \neq 0\}$, impute:
            $X_{ij}^{\text{imputed}} \leftarrow \frac{1}{|\mathcal{N}_{ij}^{\text{nonzero}}|} \sum_{n \in \mathcal{N}_{ij}^{\text{nonzero}}} X'_{nj}$
            Update $X'_{ij} \leftarrow X_{ij}^{\text{imputed}}$
return $\mathbf{X}'$

3.3 Summary

This chapter presents the benchmark methodology in detail, along with the datasets and their preprocessing. Furthermore, the metrics to evaluate the performance of the imputation methods are discussed. Lastly, the proposed imputation method, 'SpaMean-Impute' is explained with its architecture and algorithm.

Chapter 4

Results

This chapter discusses the results in two segments. First, the results from benchmarking the imputation methods are discussed in Section 4.1. Next, the performance of the proposed method, 'SpaMean-Impute' is discussed in detail in Section 4.2.

4.1 Benchmark Results

Performance evaluation of the SOTA imputation and proposed method is discussed in this section. The benchmarking is performed centering several research questions(RQs) such as:

- Are there any imputation method that consistently performs well across all SRT technologies?
- Is there any method that performs well in a particular technology?
- Do top gene selections impact the performance of the methods?

The following subsection 4.1.1 to 4.1.3 will discuss these RQs.

4.1.1 Evaluation of Imputation Methods Using Clustering-Based Metrics

The benchmarking evaluates imputation methods against the base method (i.e., no imputation) across 5 technology datasets, using ARI, NMI, AMI, and HOMO as performance metrics. Figure 4.1 demonstrates different scores with base and seven imputation methods for visium datasets. ARI results reveal that most of the imputation method struggles with clustering accuracy compared to the base method. The gene selection also has an impact, where the method performs poorly when the top 2000 or 5000 genes are selected, compared to when all the genes are selected. Both NMI and AMI values highlight the superior performance of KNN Impute gimVI, and tangram when all genes are selected, whereas SimpleImpute performed better when 2000 or 5000 genes are selected. Lastly, HOMO scores indicate that gimVI offers the most homogeneous clustering, displaying stable and

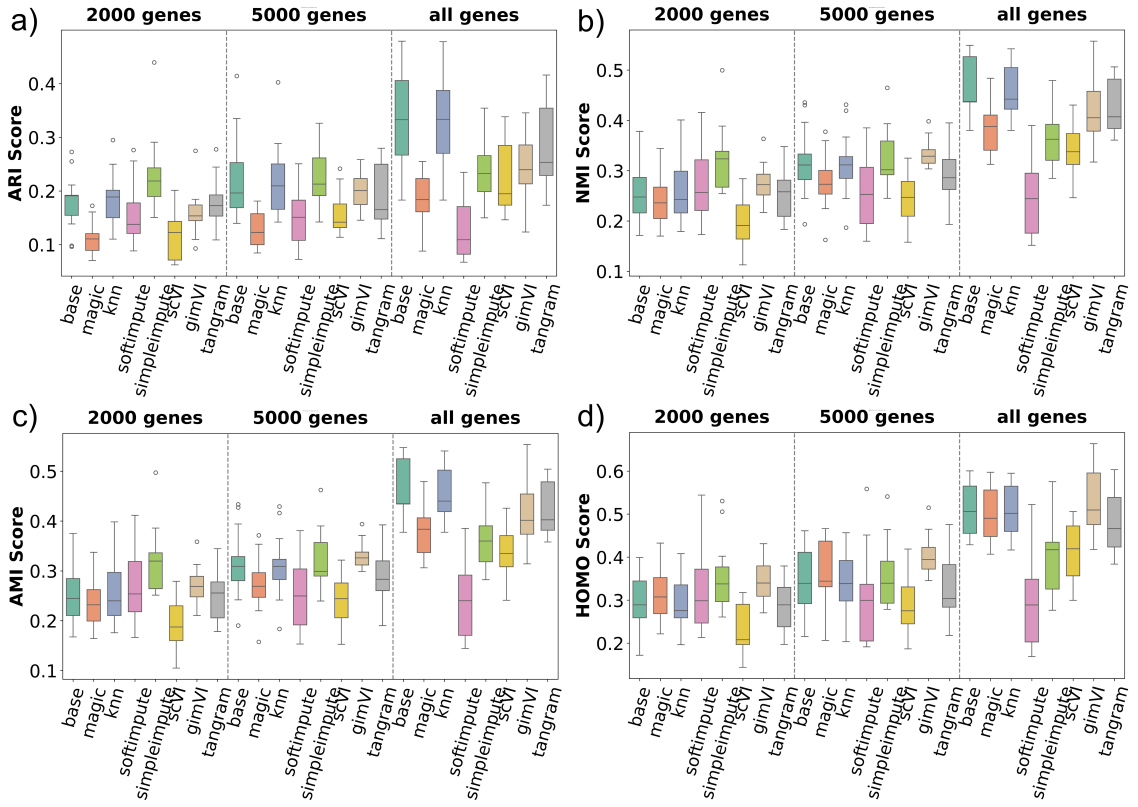


Figure 4.1: Clustering benchmark results for the 10x Genomics Visium dataset across three gene selection levels (2000 genes, 5000 genes, and all genes). Eight methods—including the base (no imputation) and seven imputation approaches—are evaluated using (a) ARI, (b) NMI, (c) AMI, and (d) HOMO scores. `gimVI` performs well in most cases. However, the base method performs well when all genes are selected.

high values across various gene subsets. Overall, gimVI performed well for most cases and remained consistent across all genes subset. It is also interesting to notice that the mean ARI, NMI, AMI, and HOMO scores are better for base and across all imputation methods when `'all genes'` are selected for the 10x Genomics Visium.

Figure 4.2 presents comparisons of the performance of clustering in the Stereo-seq technology for three gene selection strategies. The base method consistently achieves strong clustering performance, particularly with ARI, outperforming most imputation methods across all gene subsets except for MAGIC method. In contrast, more complex methods like `scVI`, `gimVI`, and `tangram` tend to show reduced performance in both clustering accuracy and stability. Although `KNN Impute`, `gimVI` provides relatively stable and homogeneous clustering in the HOMO score, it does not consistently outperform the base method. These results suggest that aggressive or deep imputation may distort the biological structure of the Stereo-seq data, and simpler methods may be more suitable. Unlike 10x Genomics Visium the mean ARI, NMI, AMI, and HOMO scores are better for most of the imputation methods (i.e., MAGIC) when `2000 or 5000 genes` are selected for the Stereo-seq. This suggests that imputation in Stereo-seq may improve when highly variable gene selection

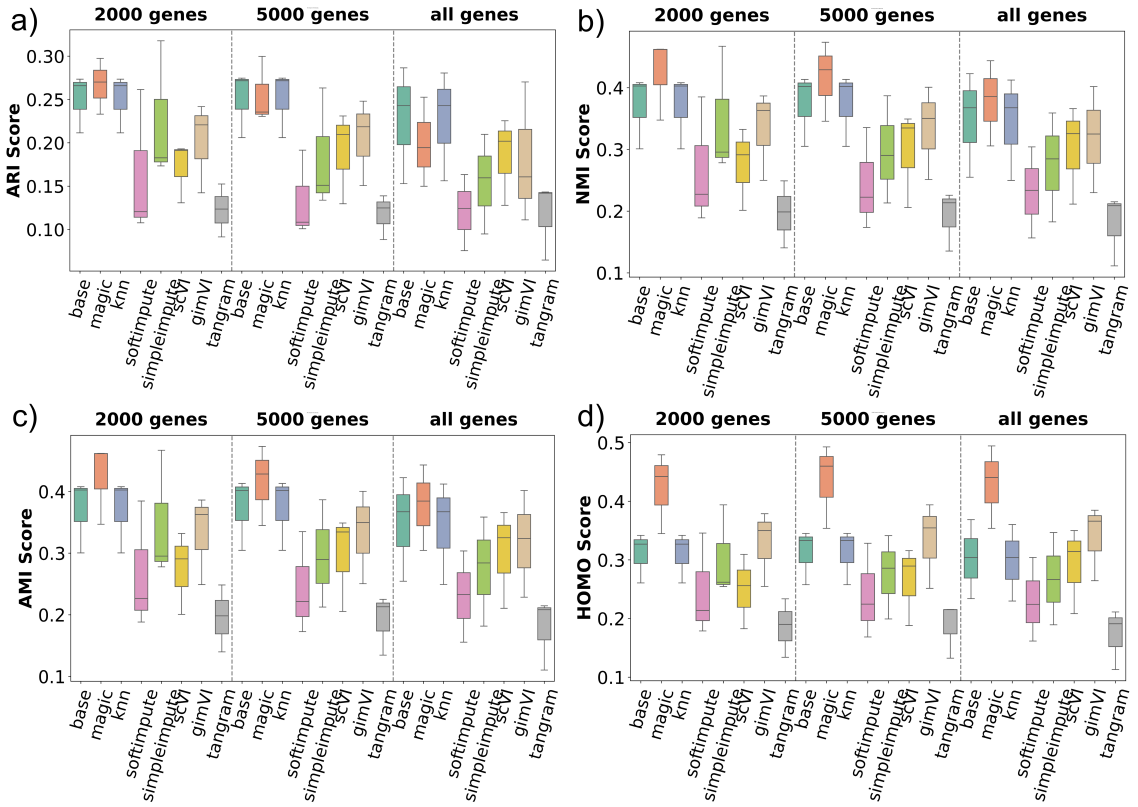


Figure 4.2: Clustering benchmark results for the Stereo-seq dataset across three gene selection levels (2000 genes, 5000 genes, and all genes). Eight methods—including the base (no imputation) and seven imputation approaches—are evaluated using (a) ARI, (b) NMI, (c) AMI, and (d) HOMO scores. MAGIC method consistently performs well across all metrics.

is applied. Stereo-seq is a much larger and sparser dataset compared to 10x Genomics VisiumṪhis is one of the reasons why gene selection results in better performance.

Figure 4.3 shows clustering performance on the Slide-seqV2Åcross all four metrics, tangram supersedes all other methods, including the base method. `gimVI` also performs well for AMI and HOMO scores. However, simpler methods like `magic` and `SimpleImpute` show poor performance in most scenarios, indicating that an imputation method performing better in one technology might not perform good for other technologies. In contrast, deep-learning-based methods such as `scVI`, `gimVI`, and `tangram` generally performed well. This may contribute to the fact that Slide-seqV2 is high-dimensional and consists of higher sparsity compared to 10x Genomics Visium. The HOMO scores highlight that `tangram` and `gimVI` can maintain better cluster homogeneity in some settings. KNN Impute also performed well. However, it was later discovered that KNN Impute fails to impute such high sparse data effectively. Furthermore, Figure 4.3 demonstrates that when `5000` genes are selected, the performance variability across different imputation methods is reduced, as evidenced by the narrower interquartile ranges in the box plots. This observation implies that the stability and consistency of imputation methods improve with the inclusion

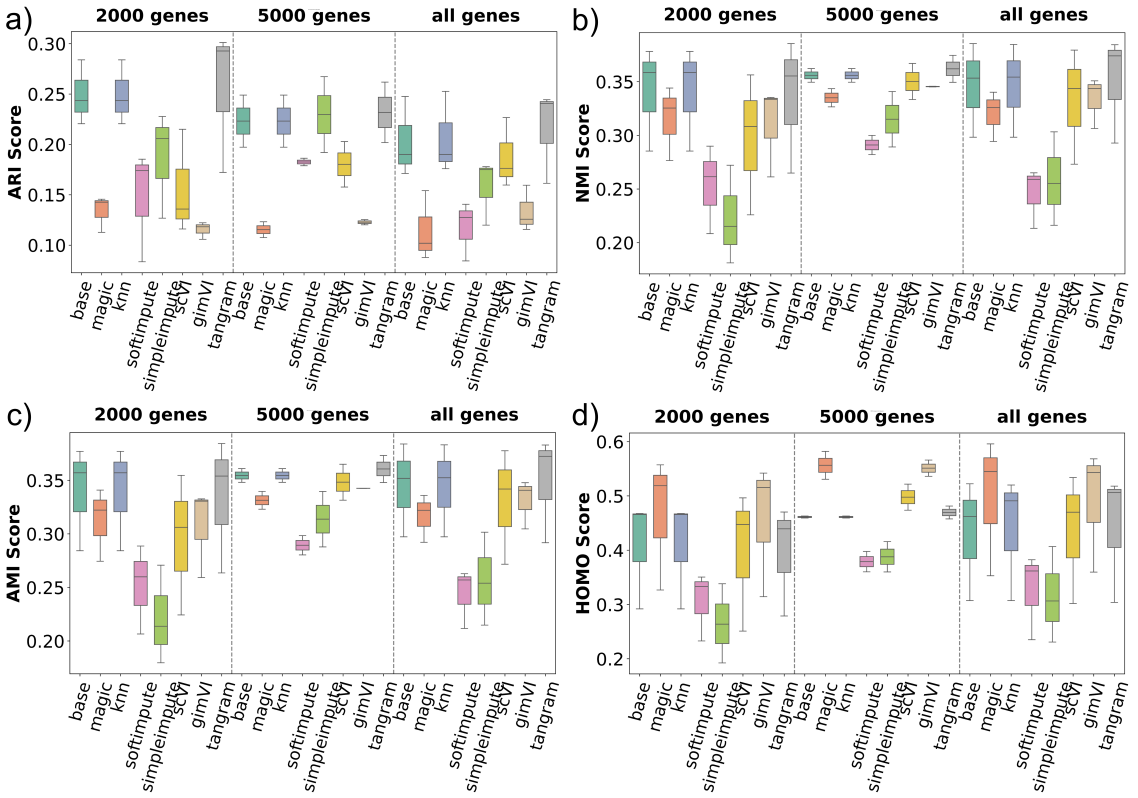


Figure 4.3: Clustering performance on the Slide-seqV2 dataset across three gene selection levels (2000 genes, 5000 genes, and all genes), evaluated using (a) ARI, (b) NMI, (c) AMI, and (d) HOMO scores. The imputation method `tangram` consistently outperforms all other methods across most metrics and gene selections.

of a larger set of gene features in the Slide-seqV2.

Figure 4.4 presents clustering evaluation results on the XYZeq where the boxplot distribution seems skewed for most cases. While the base method maintains moderate clustering quality, deep learning based imputation methods such as `scVI`, `gimVI`, and `tangram` show improved performance. However, other simpler methods consistently underperform across all metrics and gene settings.

We also evaluated these imputation methods on sci-Space dataset. However, we found only one sci-Space dataset having annotations. Hence, we do not generated a box plot distribution for this technology. However, the details of the clustering metrics results for this technology can be found in the Appendix (Table A.1), where MAGIC consistently outperformed across all metrics and gene selections. This indicates that some technology datasets (for e.g., sci-Space) may need simpler imputation methods, whereas others (for e.g., XYZeq, Stereo-seq, Slide-seqV2) may need more complex, deep learning-based architectures due to the inherently complex nature of the datasets.

The benchmarking results indicate that no single imputation method consistently outperforms others across all SRT technologies. However, some methods demonstrate superior performance for specific technologies; for example, `gimVI` performs well on the 10x

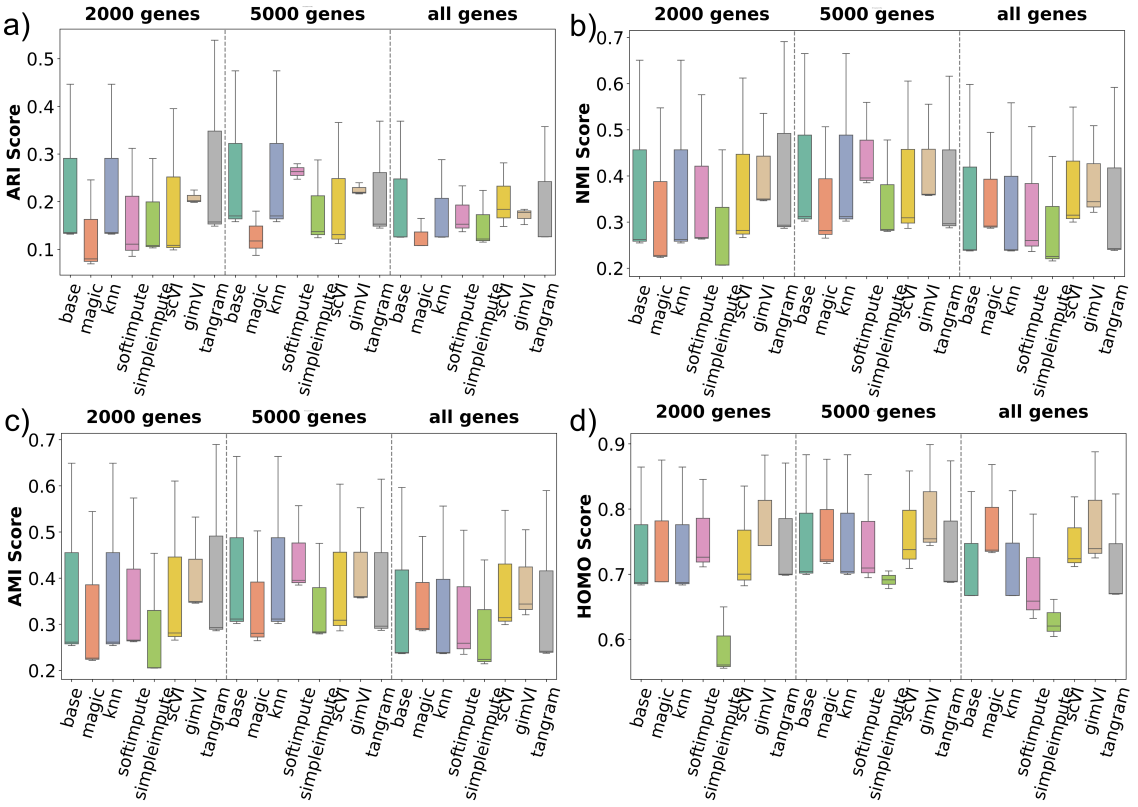


Figure 4.4: Clustering evaluation results on the XYZeq dataset across three gene selection levels (2000 genes, 5000 genes, and all genes), assessed using (a) ARI, (b) NMI, (c) AMI, and (d) HOMO scores. The boxplot distributions appear skewed in most cases. Deep learning-based methods such as `scVI`, `gimVI`, and `tangram` demonstrate competitive clustering performance across all metrics and gene subsets. In contrast, simpler imputation methods consistently underperform, highlighting the importance of model complexity in handling the characteristics of XYZeq.

Genomics Visium dataset, while `tangram` achieves strong results on Slide-seqV2. Furthermore, the results highlight the impact of gene selection on performance, as observed in Stereo-seq and Slide-seqV2, where selecting the top 2000 or 5000 genes leads to more stable and improved clustering outcomes. These findings emphasize the importance of aligning imputation strategies with both the characteristics of the technology and the gene selection criteria.

4.1.2 Evaluation of Dropout Reduction Capability Across Imputation Methods

Imputation methods must have the ability to differentiate between valid dropout and biological zero. Otherwise, it will impute the biological zeros, resulting in the distortion of the underlying biological nature of the datasets. Here, we discuss how these dropouts are removed by methods. Table 4.1 shows the percentage of zero values in the 10x Genomics Visium datasets before and after imputation across three different gene selection scenarios. The raw data exhibits extremely high sparsity, with most datasets having over 85-95% zeros. Imputation methods such as `SoftImpute`, `SimpleImpute`, `scVI`, `gimVI`, and `tangram` completely eliminate all zeros, leading to a zero sparsity of 0%. In contrast, `MAGIC` shows a partial but very high amount of imputation depending on the dataset. Notably, the `KNN Impute` method does not reduce sparsity at all, maintaining the same percentage of zeros as in the raw data. This indicates that `KNN Impute` do not have the ability to impute any dropouts. To ensure, the K-value was varied, and zero sparsity remained the same as the raw data. Conversely, `SoftImpute` and `SimpleImpute` need prior knowledge of dropouts. Hence, we set all the zeros in the Raw data to 'NaN' values, which are imputed by these methods. In this way, all the zeros are imputed, whether they are due to dropout or biological reasons. However, the deep learning-based methods do not need prior knowledge, yet they also impute all the zeros, i.e., fail to differentiate the actual dropouts among the zeros in SRTs.

We observe a similar trend in zero sparsity after imputing the Stereo-seq datasets as shown in Table 4.2. KNN Impute still fails to impute any data despite different K values, whereas other imputation methods impute all zeros in the datasets. This is a big concern as all the zeros cannot be due to technical reasons. Thus, imputation methods must have a detection capability that can differentiate between true dropout and actual biological expression in SRTs. For other technologies, we have found similar zero-sparsity trends and showed detailed results in the Appendix (Table A.2 to Table A.4).

Methods such as `gimVI` and `tangram` are designed with internal mechanisms to probabilistically distinguish between technical dropouts and biological zeros [33, 34]. For example, `gimVI` employs a generative model to estimate dropout probability, while `tangram` leverages dense reference transcriptomes for spatial mapping. However, they completely removed all zeros, including biological ones. This suggests that, despite their theoretical ability, they may not effectively differentiate valid dropouts from biological zeros in

Table 4.1: Percentage of Zero Sparsity in the 10x Genomics Visium datasets before (Raw) and after applying imputation methods.

SL No.	Top Genes	Dataset Name	RAW	MAGIC	KNN Impute	Soft Impute	Simple Impute	scVI	gimVI	tangram
01.	2000	151507	93.99	0.021	93.99	0	0	0	0	0
		151508	94.86	0	94.86	0	0	0	0	0
		151509	94.63	0.30	94.63	0	0	0	0	0
		151510	94.72	0.28	94.72	0	0	0	0	0
		151669	89.61	0	89.61	0	0	0	0	0
		151670	88.31	0	88.31	0	0	0	0	0
		151671	91.23	0	91.23	0	0	0	0	0
		151672	92.002	0	92.002	0	0	0	0	0
		151673	93.99	0.019	93.99	0	0	0	0	0
		151674	89.54	0.23	89.54	0	0	0	0	0
		151675	91.64	0.20	91.64	0	0	0	0	0
		151676	91.14	0.05	91.14	0	0	0	0	0
		BRCA1	80.38	0.37	80.38	0	0	0	0	0
02.	5000	151507	92.41	0.003	92.41	0	0	0	0	0
		151508	93.44	0	93.44	0	0	0	0	0
		151509	93.41	0.23	93.41	0	0	0	0	0
		151510	93.07	0.15	93.07	0	0	0	0	0
		151669	86.08	0	86.08	0	0	0	0	0
		151670	85.78	0	85.78	0	0	0	0	0
		151671	89.58	0	89.58	0	0	0	0	0
		151672	89.84	0	89.84	0	0	0	0	0
		151673	92.41	0.005	92.41	0	0	0	0	0
		151674	88.62	0.13	88.62	0	0	0	0	0
		151675	90.28	0.1	90.28	0	0	0	0	0
		151676	89.84	0.03	89.84	0	0	0	0	0
		BRCA1	79.83	0.45	79.83	0	0	0	0	0
03.	all	151507	89.69	0.01	89.69	0	0	0	0	0
		151508	90.88	0	90.88	0	0	0	0	0
		151509	90.48	0.20	90.48	0	0	0	0	0
		151510	90.52	0.16	90.52	0	0	0	0	0
		151669	84.28	0	84.28	0	0	0	0	0
		151670	84.28	0	84.28	0	0	0	0	0
		151671	86.45	0	86.45	0	0	0	0	0
		151672	86.66	0	86.66	0	0	0	0	0
		151673	89.69	0.02	89.69	0	0	0	0	0
		151674	83.94	0.15	83.94	0	0	0	0	0
		151675	87.35	0.12	87.35	0	0	0	0	0
		151676	86.43	0.08	86.43	0	0	0	0	0
		BRCA1	73.21	1.26	73.21	0	0	0	0	0

Table 4.2: Percentage of Zero Sparsity in the Stereo-seq datasets before (Raw) and after applying imputation methods.

SL No.	Top Genes	Dataset Name	RAW	MAGIC	KNN Impute	Soft Impute	Simple Impute	scVI	gimVI	tangram
01.	2000	DT2_D0	92.88	0.05	92.88	0	0	0	0	0
		DX6_D2	86.18	0	0.002	86.18	0	0	0	0
		FB2_D1	86.04	0.002	86.04	0	0	0	0	0
02.	5000	DT2_D0	92.5	0.019	92.5	0	0	0	0	0
		DX6_D2	84.92	0.003	84.92	0	0	0	0	0
		FB2_D1	84.62	0.006	84.62	0	0	0	0	0
03.	all	DT2_D0	89.67	0.06	89.67	0	0	0	0	0
		DX6_D2	84.41	0.017	84.41	0	0	0	0	0
		FB2_D1	83.44	0.079	83.44	0	0	0	0	0

practice. Consequently, this over-imputation may distort the biological integrity of the datasets. This is a significant limitation of these imputation methods as the dropouts caused by experimental factors and genuine biological zero expressions are imputed indiscriminately by these models. However, as imputation methods, these models should be able to distinguish between dropout events and true biological zeros, an essential characteristic for accurate imputation. The proposed method 'SpaMean-Impute' is inspired by these major findings.

4.1.3 Computational Metrics among the imputation methods

In this section, we discuss the computational complexity of the benchmarked imputation methods. We used the NVIDIA A100 GPU available in Colab, typically features 40 GB HBM2 memory, up to 312 TFLOPS FP16 performance, and over 1.5 TB/s memory bandwidth, making it highly suitable for large-scale deep learning and spatial omics workloads. Figure 4.5 demonstrates the memory and runtime during imputation time vs the number of spots for each imputation method for the different feature subsets. The total of 23 datasets from five technologies have various ranges of spots/cells, with 10x Genomics Visium having the lowest one (around 3 thousand spots) and Stereo-seq having the highest one (around 20 thousand spots). We can observe from Figure 4.5 that the memory and runtime increase with the number of spots. Figure 4.5a. demonstrates that `tangram` takes maximum memory during imputation, whereas `gimVI` takes the maximum runtime for imputation (Figure 4.5b). Surprisingly, `scVI` takes the lowest memory, even compared to other simple imputation methods, but has a very high runtime with an increasing number of spots. The observed computational demands of deep learning-based methods—particularly `scVI`, `gimVI`, and `tangram`—can be attributed to their architectural complexity and data handling strategies. Specifically, both `scVI` and `gimVI` utilize VAE frameworks involving iterative stochastic optimization over multiple epochs. `gimVI`, an extension of `scVI`, incorporates both spatial and gene expression modalities, adding further computational burden during training, which explains its higher runtime compared to `scVI`. Despite the high runtime, `scVI` exhibits surprisingly low memory usage. This can be attributed to its use of mini-batch training and efficient memory management practices in PyTorch, which allow it to process subsets of the data sequentially rather than loading the full dataset into memory. In contrast, `gimVI` requires access to both paired spatial and transcriptomic inputs, which increases the dimensionality and memory footprint of each training sample. Moreover, it often disables aggressive mini-batching due to spatial dependencies, leading to higher memory use. `tangram`, on the other hand, requires memory-intensive matrix operations and alignment steps between single-cell and SRT datasets. These operations are often dense and require holding large matrices in memory, especially when aligning high-dimensional single-cell reference data to tens of thousands of spatial locations. As a result, `tangram` shows high memory usage even though its runtime is not as extreme as `gimVI`. In comparison, traditional methods such as `magic`, `knn`, or

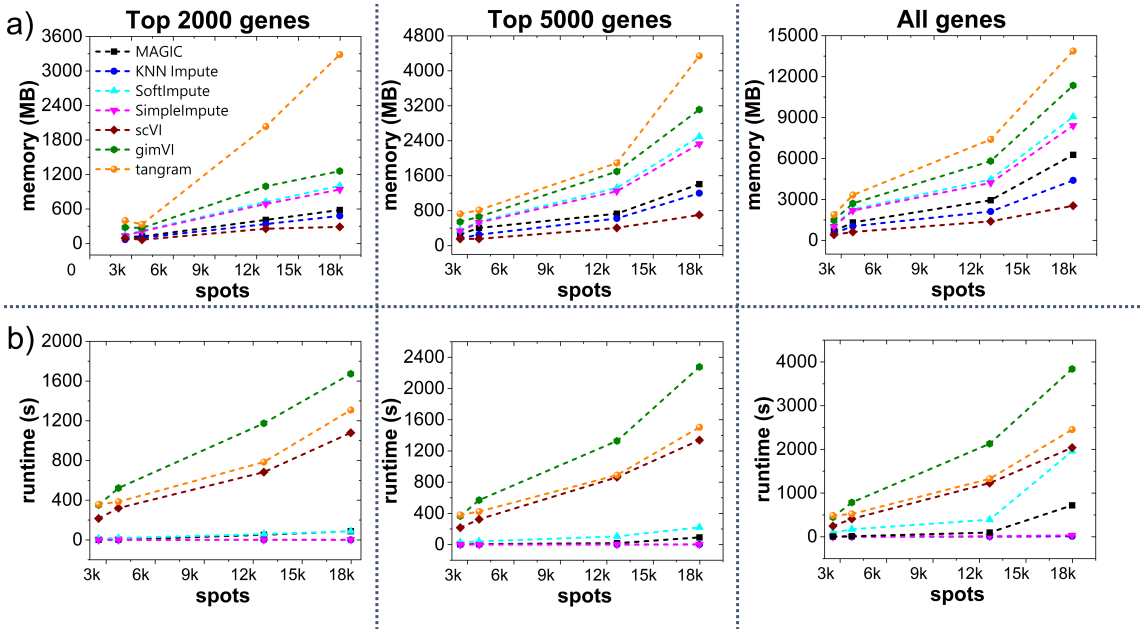


Figure 4.5: Computational Complexity of the Imputation Methods for varying number of spots (a) Memory consumption and (b) Runtime comparison for the imputation methods for `200 genes, 5000 genes or all genes`. Deep learning-based methods, particularly `gimVI` and `scVI`, exhibit the highest runtimes due to their iterative training and complex model architectures, whereas `tangram` has the highest memory consumption.

`simpleimpute` have simpler algorithmic steps, resulting in moderate increases in runtime and memory as the dataset size grows. However, their computational load is negligible compared to deep learning-based methods, which are designed for high-capacity learning but come at a significant computational cost. Other basic imputation methods also take higher runtime and memory with an increasing number of spots; however, they are negligible compared to the deep-learning methods.

4.2 Performance of the SpaMean-Impute

We proposed SpaMean-Impute considering two facts: i) Utilization of spatial information in SRT datasets while imputing, ii) detection of valid dropouts. Considering these criteria, we tested our proposed method on the different datasets and found that our method outperformed SOTA imputation methods, demonstrating stable performance. Although integration of spatial information in SRTs imputation is common [33, 34, 36, 37], valid dropout detection is missing in the literature. Our method can detect valid dropouts and hence preserve the actual zeros, unlike the SOTA methods' inability to distinguish between valid dropouts and actual zeros as discussed in Section 4.1.2. Hence, the enhancement of clustering metrics can be achieved with imputing valid dropouts. This also reflects in the zero sparsity before and after imputation using our method. In the following sections, the performance of the proposed method is discussed on the SRT technologies, excluding

XYZeq technology due to the lack of spatial location information in the XYZeq dataset we acquired.

4.2.1 Comparison of clustering metrics of SpaMean-Impute with SOTA methods

As we demonstrated in our benchmarking results (Section 4.1.1), highly variable genes have a significant impact on the imputation methods' performance and subsequently on the clustering results. For example, imputation methods perform well when `all genes` are selected for 10x Genomics Visium whereas `top 2000 or 5000` genes selection work well in the case of Stereo-seq, Slide-seqV2, and sci-Space. Hence, we considered `all genes` for 10x Genomics Visium, `top 2000` genes for Stereo-seq, and `top 5000` genes for Slide-seqV2 when comparing our proposed method's performance with SOTA methods.

Table 4.3 presents the ARI scores across SRTs compared to 'SpaMean-Impute'. Across the `10x Genomics Visium` datasets, `SpaMean-Impute` consistently achieves the highest ARI scores for all samples, significantly outperforming both baseline and existing methods. In the case of `Stereo-seq` data, although `MAGIC` occasionally performs competitively, `SpaMean-Impute` still provides superior performance across all samples. For `Slide-seqV2`, which typically presents more sparsity and complexity, `SpaMean-Impute` again outperforms other methods, including `scVI` and `tangram`, showing its adaptability across gene densities and platform characteristics. Overall, this table illustrates that `SpaMean-Impute` not only generalizes well across technologies and gene selection settings, but also consistently leads to better clustering fidelity.

Table 4.3: ARI scores for SOTA imputation methods compared to 'SpaMean-Impute'. The highest score for each technology's tissue slide corresponding to the respective imputation method is bolded.

SL No.	Technology	Top Genes	Dataset Name	base	MAGIC	KNN Impute	Soft Impute	Simple Impute	scVI	gimVI	tangram	SpaMean-Impute
01.	10x Genomics Visium	all	151507	0.26	0.18	0.27	0.06	0.19	0.18	0.22	0.24	**0.36**
			151508	0.25	0.17	0.25	0.08	0.14	0.17	0.19	0.20	**0.31**
			151509	0.21	0.10	0.21	0.12	0.23	0.17	0.16	0.20	**0.30**
			151510	0.27	0.12	0.27	0.09	0.17	0.16	0.24	0.22	**0.32**
			151669	0.40	0.17	0.38	0.09	0.23	0.19	0.23	0.34	**0.45**
			151670	0.36	0.16	0.36	0.07	0.22	0.19	0.26	0.26	**0.49**
			151671	0.47	0.24	0.47	0.17	0.30	0.31	0.29	0.41	**0.49**
			151672	0.45	0.22	0.44	0.10	0.28	0.33	0.28	0.37	**0.48**
			151673	0.26	0.19	0.27	0.07	0.19	0.18	0.21	0.24	**0.29**
			151674	0.33	0.19	0.33	0.17	0.24	0.28	0.23	0.25	**0.35**
			151675	0.44	0.25	0.42	0.23	0.35	0.28	0.34	0.40	**0.54**
			151676	0.37	0.23	0.38	0.23	0.26	0.28	0.29	0.35	**0.41**
			BRCA1	0.18	0.08	0.18	0.14	0.19	0.14	0.12	0.17	**0.19**
02.	Stereo-seq	2000	DT2_D0	0.26	0.29	0.26	0.26	0.31	0.19	0.24	0.12	**0.31**
			DX6_D2	0.27	0.27	0.27	0.12	0.18	0.19	0.22	0.15	**0.29**
			FB2_D1	0.21	0.23	0.21	0.10	0.17	0.13	0.14	0.09	**0.23**
03.	Slide-seqV2	5000	WT	0.24	0.12	0.24	0.18	0.26	0.20	0.12	0.26	**0.27**
			diabetes	0.19	0.10	0.19	0.17	0.19	0.15	0.12	0.20	**0.22**
			mouse	0.19	0.15	0.19	0.08	0.11	0.15	0.15	0.16	**0.20**
04.	sci-Space	5000	GSE 166692	0.45	0.41	0.45	0.24	0.34	0.43	0.37	0.44	**0.48**

Furthermore, Table 4.4 reports the comparison for NMI scores where `SpaMean-Impute` consistently achieves the highest NMI scores, with values reaching up

to 0.67 for dataset `151675` for 10x Genomics Visium. It outperforms methods such as `SoftImpute`, `SimpleImpute`, and deep learning models like `scVI`, `gimVI`, and `tangram`. Similar performance is observed for sci-Spacetechnology. The NMI scores for `SpaMean-Impute` remain notably superior across most samples, indicating its robustness in preserving mutual information between predicted and ground truth labels. In the `Stereo-seq` datasets, although `MAGIC` briefly matches the top performance in one instance (e.g., `DX6_D2`), `SpaMean-Impute` again generally shows stable performance, suggesting adaptability across varying spatial resolution and sparsity. Although Table 4.3 shows that `SpaMean-Impute` performs competitively on the `Stereo-seq` datasets in terms of ARI, its performance in terms of NMI is relatively lower. This discrepancy arises from the intrinsic differences between these two clustering evaluation metrics. The ARI metric evaluates clustering accuracy by comparing all pairs of samples and checking how consistently they are grouped in both the predicted and true labels. It is particularly sensitive to the precise grouping of sample pairs and penalizes random agreements. In contrast, NMI measures the amount of mutual information shared between predicted and true labels, normalized by their entropies. NMI tends to reward clusterings that align well in terms of label distributions and balanced entropy, even if some sample groupings are slightly off. In the case of `Stereoseq`, which exhibits high data sparsity, `SpaMean-Impute` appears effective at correctly assigning sample pairs to the same clusters, yielding high ARI values. However, the predicted cluster distributions might not align as well in entropy or size balance with the true labels, leading to comparatively lower NMI scores. Similarly, for `Slide-seqV2` data, which tends to be more sparse and noisy, `SpaMean-Impute` achieves the best NMI values. Overall, the results highlight the advantage of spatially aware, tailored imputation like `SpaMean-Impute` for enhancing downstream clustering outcomes.

Table 4.4: Comparison of NMI Scores with 'SpaMean-Impute'. The highest score for each technology's tissue slide corresponding to the respective imputation method is bolded.

SL No.	Technology	Top Genes	Dataset Name	base	MAGIC	KNN Impute	Soft Impute	Simple Impute	scVI	gimVI	tangram	SpaMean-Impute
01.	10x Genomics Visium	all	151507	0.43	0.38	0.44	0.161	0.32	0.32	0.40	0.39	**0.55**
			151508	0.40	0.35	0.40	0.19	0.28	0.31	0.36	0.36	**0.52**
			151509	0.40	0.34	0.39	0.25	0.39	0.36	0.39	0.376	**0.53**
			151510	0.43	0.34	0.43	0.21	0.28	0.29	0.43	0.38	**0.52**
			151669	0.43	0.31	0.43	0.17	0.31	0.24	0.36	0.39	**0.53**
			151670	0.43	0.31	0.42	0.15	0.33	0.26	0.37	0.40	**0.53**
			151671	0.52	0.39	0.52	0.29	0.36	0.36	0.45	0.50	**0.59**
			151672	0.54	0.41	0.54	0.24	0.41	0.39	0.47	0.50	**0.60**
			151673	0.43	0.40	0.44	0.15	0.32	0.33	0.40	0.40	**0.47**
			151674	0.46	0.42	0.46	0.29	0.38	0.37	0.43	0.40	**0.52**
			151675	0.53	0.48	0.50	0.38	0.47	0.43	0.55	0.49	**0.67**
			151676	0.52	0.45	0.53	0.34	0.41	0.41	0.49	0.48	**0.60**
			BRCA1	0.37	0.31	0.37	0.32	0.37	0.31	0.31	0.37	**0.39**
02.	Stereo-seq	2000	DT2_D0	0.40	0.46	0.40	0.38	0.46	0.33	0.46	0.33	**0.48**
			DX6_D2	0.40	**0.46**	0.40	0.22	0.29	0.29	0.36	0.24	0.43
			FB2_D1	0.30	**0.34**	0.30	0.18	0.27	0.20	0.24	0.14	0.32
03.	Slide-seqV2	5000	WT	0.36	0.34	0.36	0.29	0.34	0.36	0.34	0.37	**0.38**
			diabetes	0.34	0.32	0.34	0.28	0.28	0.33	0.34	0.34	**0.353**
			mouse	0.29	0.29	0.29	0.21	0.21	0.27	0.30	0.29	**0.30**
04.	sci-Space	5000	GSE 166692	0.67	0.68	0.67	0.5	0.58	0.65	0.63	0.67	**0.70**

Table 4.5 demonstrates the AMI comparative scores. Among the `10x Genomics Visium`

datasets, `SpaMean-Impute` consistently achieves the highest AMI scores, demonstrating significant improvements over baseline and existing imputation techniques. For `Stereo-seq`, while `MAGIC` performs competitively, achieving the highest score on DX6_D2 and FB2_D1, `SpaMean-Impute` leads on DT2_D0, indicating effectiveness depending on the sample. On the more sparse and noisy `Slide-seqV2` datasets, AMI scores are closely clustered among methods, with `SpaMean-Impute` providing marginal gains. Lastly, in the `sci-Space` dataset, `SpaMean-Impute` outperforms all baselines, achieving the top AMI score. These results reinforce that `SpaMean-Impute` not only generalizes across technological platforms but also offers reliable clustering accuracy as reflected in mutual information metrics.

Table 4.5: AMI scores for SOTA imputation methods compared to 'SpaMean-Impute'. The highest score for each technology's tissue slide corresponding to the respective imputation method is bolded.

SL No.	Technology	Top Genes	Dataset Name	base	MAGIC	KNN Impute	Soft Impute	Simple Impute	scVI	gimVI	tangram	SpaMean-Impute
01.	10x Genomics Visium	all	151507	0.43	0.38	0.43	0.15	0.31	0.32	0.4	0.39	**0.55**
			151508	0.4	0.34	0.4	0.18	0.28	0.3	0.35	0.35	**0.52**
			151509	0.39	0.33	0.39	0.24	0.39	0.35	0.39	0.37	**0.54**
			151510	0.43	0.33	0.42	0.21	0.28	0.29	0.43	0.38	**0.52**
			151669	0.43	0.3	0.43	0.17	0.31	0.24	0.35	0.39	**0.55**
			151670	0.43	0.3	0.41	0.14	0.33	0.25	0.37	0.4	**0.52**
			151671	0.52	0.39	0.52	0.28	0.35	0.36	0.45	0.50	**0.59**
			151672	0.54	0.40	0.54	0.24	0.41	0.39	0.47	0.50	**0.59**
			151673	0.43	0.39	0.43	0.15	0.31	0.33	0.39	0.40	**0.45**
			151674	0.46	0.42	0.46	0.29	0.38	0.37	0.42	0.40	**0.52**
			151675	0.53	0.47	0.50	0.38	0.47	0.42	0.55	0.49	**0.67**
			151676	0.52	0.44	0.52	0.34	0.40	0.41	0.48	0.47	**0.60**
			BRCA1	0.37	0.31	0.37	0.32	0.37	0.31	0.31	0.37	**0.38**
02.	Stereo-seq	2000	DT2_D0	0.40	0.46	0.40	0.38	0.46	0.33	0.38	0.19	**0.48**
			DX6_D2	0.40	**0.46**	0.40	0.22	0.29	0.29	0.36	0.24	0.36
			FB2_D1	0.30	**0.34**	0.30	0.18	0.27	0.20	0.24	0.13	0.3
03.	Slide-seqV2	5000	WT	0.36	0.33	0.36	0.29	0.33	0.36	0.34	**0.37**	0.34
			diabetes	0.34	0.32	0.34	0.28	0.28	0.33	0.34	0.34	**0.35**
			mouse	0.29	0.29	0.29	0.21	0.21	0.27	**0.3**	0.29	0.29
04.	sci-Space	5000	GSE 166692	0.67	0.67	0.67	0.5	0.57	0.64	0.62	0.66	**0.69**

Furthermore, across the `10x Genomics Visium` datasets, `SpaMean-Impute` consistently achieved the highest HOMO scores, surpassing both conventional imputation baselines and deep learning approaches, thus indicating a strong preservation of class-specific structure post-imputation (Table 4.6). For the `sci-Space` dataset, `SpaMean-Impute` again achieved the top HOMO score. In contrast, for datasets from the `Stereo-seq` and `Slide-seqV2` platforms, the HOMO scores for all methods—including `SpaMean-Impute`, are generally lower. This can be attributed to the higher sparsity and noisier gene expression profiles inherent to these technologies, which pose significant challenges for accurately reconstructing spot type boundaries. Moreover, due to the ultra-high spatial resolution in these platforms, spatial spots often capture only partial transcriptomic profiles or contain mixed cell signals, leading to degraded clustering homogeneity post-imputation. In such contexts, while `SpaMean-Impute` does not always achieve the best HOMO score, it remains competitive and still offers improvements over other methods.

According to the overall clustering results, it is evident that '`SpaMean-Impute`' demonstrates consistently competitive performance across all evaluation metrics. Specifically, it achieves relatively high values in ARI, NMI, and AMI, indicating its effectiveness in align-

Table 4.6: Comparison of HOMO Scores with 'SpaMean-Impute'. The highest score for each technology's tissue slide corresponding to the respective imputation method is bolded.

SL No.	Technology	Top Genes	Dataset Name	base	MAGIC	KNN Impute	Soft Impute	Simple Impute	scVI	gimVI	tangram	SpaMean-Impute
01.	10x Genomics Visium	all	151507	0.45	0.46	0.45	0.17	0.32	0.36	0.48	0.42	**0.61**
			151508	0.42	0.42	0.41	0.2	0.27	0.35	0.41	0.38	**0.57**
			151509	0.44	0.46	0.43	0.28	0.42	0.45	0.5	0.41	**0.62**
			151510	0.4	0.44	0.46	0.24	0.28	0.35	0.52	0.43	**0.57**
			151669	0.46	0.4	0.5	0.2	0.38	0.29	0.43	0.45	**0.59**
			151670	0.5	0.43	0.48	0.19	0.41	0.32	0.47	0.51	**0.6**
			151671	0.56	0.52	0.56	0.34	0.4	0.43	0.59	0.58	**0.64**
			151672	0.6	0.55	0.59	0.3	0.5	0.47	0.64	0.6	**0.67**
			151673	0.45	0.49	0.45	0.16	0.32	0.36	0.46	0.42	**0.49**
			151674	0.5	0.55	0.5	0.34	0.43	0.41	0.52	0.46	**0.58**
			151675	0.53	0.59	0.54	0.46	0.49	0.5	0.66	0.52	**0.74**
			151676	0.57	0.57	0.58	0.4	0.43	0.47	0.59	0.53	**0.66**
			BRCA1	0.57	0.55	0.57	0.52	0.57	0.47	0.5	0.57	**0.61**
02.	Stereo-seq	2000	DT2_D0	0.32	**0.47**	0.32	0.34	0.39	0.30	0.37	0.19	0.38
			DX6_D2	0.34	**0.44**	0.34	0.21	0.25	0.25	0.34	0.23	0.31
			FB2_D1	0.26	**0.34**	0.26	0.17	0.26	0.18	0.25	0.13	0.27
03.	Slide-seqV2	5000	WT	0.46	**0.58**	0.46	0.39	0.41	0.52	0.56	0.48	0.44
			diabetes	0.45	**0.53**	0.45	0.36	0.35	0.47	**0.53**	0.45	0.47
			mouse	0.3	**0.35**	0.3	0.23	0.23	0.3	**0.35**	0.3	0.3
04.	sci-Space	5000	GSE 166692	0.63	0.69	0.63	0.44	0.51	0.6	0.58	0.62	**0.74**

ing predicted clusters with true labels. This suggests that SpaMean-Impute preserves biological structure while maintaining cluster purity and label consistency after imputation compared to other SOTA methods.

4.2.2 Average Zero Sparsity percentage comparison of SpaMean-Impute

SpaMean-Impute consider the zero values to be dropout based on the nearest neighbor and their corresponding values of the respective genes. Hence, this method doesn't impute the biological zeros as demonstrated in Table 4.7. As discussed earlier, it can be observed again that the average zero values become completely imputed after using the seven imputation methods. However, when SpaMean-Impute is applied to the RAW data, the average zero values after the imputation process still remain pretty high, on average 85.6% zeros as valid zeros, indicating 0.85% to be the dropout and imputed them. In the previous section 4.2.1, it was demonstrated how this proposed method outperformed significantly considering this small proportion to be dropped out. SpaMean-Imputefound a higher dropout in Stereo-seq whereas on average 0.5%-0.54% in case of Slide-seqV2 and sci-Space. However, the parameters nearest-neighbor, k, and δ (dropout threshold) play crucial roles in controlling the balance between sensitivity and specificity of dropout recovery. A higher k value increases the spatial context by considering more neighbors during imputation. While this may improve robustness to noise, it may also introduce averaging effects that dilute local expression patterns. Conversely, a smaller k may better preserve local heterogeneity but risks being more sensitive to local outliers or noise. The dropout threshold δ governs whether a zero entry should be considered a dropout or a true biological zero. A lower δ value makes the method conservative, imputing fewer zeros—thus reducing the risk of falsely imputing genuine non-expression values. In contrast, a higher δ increases the number of zeros that qualify for imputation, which can enhance data recovery but may

also introduce bias if truly non-expressed genes are inappropriately filled. Therefore, fine-tuning k and δ is essential for optimal imputation performance and should be guided by the sparsity level and biological complexity of the dataset. In our hyperparameter tuning, we observed that the model performs well when k is in the range of 3,5,7, and δ in the range of 5% to 20%.

Table 4.7: Average Zero Sparsity Percentage comparison of 'SpaMean-Impute' with other SOTA imputation methods.

SL No.	Technology	Top Genes	Zero Sparsity in RAW Data (%)	After MAGIC (%)	After KNN Impute (%)	After SoftImpute (%)	After SimpleImpute (%)	After scVI (%)	After gimVI (%)	After tangram (%)	After SpaMeanImpute (%)	Dropout Detected by SpaMeanImpute (%)
01.	10x Genomics Visium	all	86.45	0.154	86.45	0	0	0	0	0	85.6	0.85
02.	Stereo-seq	2000	88.37	0.02	88.37	0	0	0	0	0	85.94	2.43
03.	Slide-seqV2	5000	94.51	1.05	94.51	0	0	0	0	0	94.01	0.5
04.	sci-Space	5000	87.65	4.64	87.65	0	0	0	0	0	87.11	0.54

4.2.3 Computational Complexity of SpaMean-Impute

We also evaluated the memory consumption and runtime of 'SpaMean-Impute'. As illustrated in Figure 4.6, SpaMean-Impute demonstrates notable computational efficiency across the 10x Genomics Visium datasets. Specifically, in terms of memory usage (Figure 4.6a), SpaMean-Impute maintains consistently low consumption, performing significantly better than deep learning-based models like gimVI and tangram, and comparably or better than simpler methods such as MAGIC and SimpleImpute. Additionally, the runtime comparison (Figure 4.6b) shows that SpaMean-Impute is among the fastest methods across all datasets, achieving one of the lowest execution times. Similar observations were found in the case of Stereo-seq, Slide-seqV2, and sci-Space. The details on these technologies are provided in the Appendix (Figure A.1 and Figure A.2). This highlights the method's suitability for large-scale SRT data where both memory and time efficiency are crucial.

The overall performance of SpaMean-Impute indicates that it outperforms all imputation methods utilizing the spatial information in the SRT datasets while showing significantly lower memory consumption and runtime. However, the proposed method has limitations that include its lower performance when other gene selection settings are selected for different datasets. Furthermore, deep learning-based dropout detection may be explored to achieve even more enhanced performance.

4.3 Summary

This chapter concludes the summary of the benchmark method and the proposed method. The benchmark method revealed that no imputation method perform showed consistency in terms of their imputation performance in terms of clustering downstream analysis.

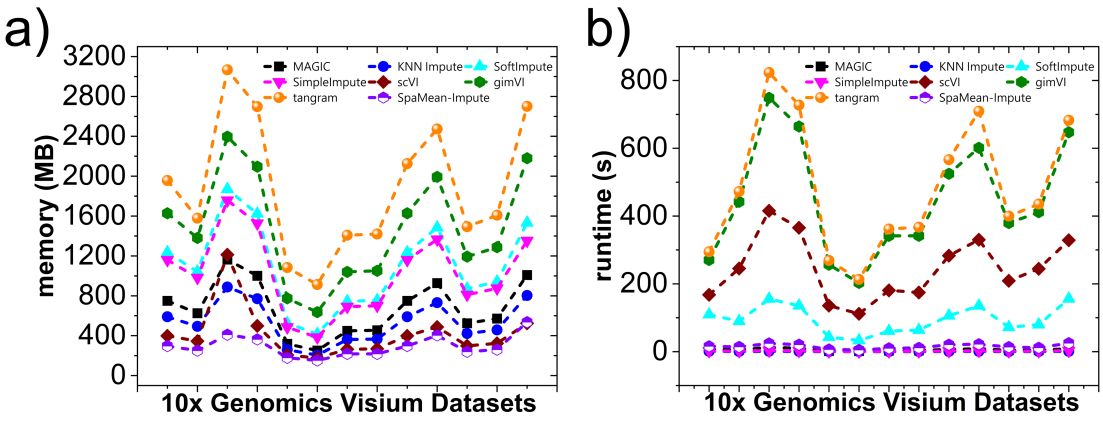


Figure 4.6: Computational Complexity comparison between SpaMean-Impute and other SOTA methods for 10x Genomics Visiumdatasets (a) Memory consumption and (b) Runtime considering all genes. SpaMean-Impute shows one of the lowest runtimes and memory consumption compared to other methods, especially the deep learning-based models.

The proposed method demonstrates significant superiority over several SOTA imputation algorithms across multiple performance metrics, including the ARI, NMI, AMI, and HOMO. Quantitatively, our approach achieves an average improvement of **16.15% in ARI**, **18.45% in NMI**, **18.96% in AMI**, and **13.98% in HOMO** compared to existing SOTA methods. These gains indicate that the proposed framework not only enhances cluster consistency and agreement with ground truth labels but also better preserves biological homogeneity across spatial domains. Unlike deep learning-based methods such as `scVI`, `gimVI`, and `tangram`, which rely on computationally intensive optimization and large neural architectures, our method maintains high accuracy while remaining computationally lightweight. The algorithm achieves this efficiency through spatially informed local averaging and adaptive zero-filtering, which eliminate redundant computations and reduce the number of required iterations. As a result, the proposed model is approximately **33 times faster** and requires on average **1500 MB less memory** during the imputation process than existing deep learning-based SOTA approaches. This combination of accuracy and scalability makes the proposed method highly suitable for large-scale spatial transcriptomics datasets, where computational resources and runtime often impose major constraints. Furthermore, the method's efficiency enables seamless integration into downstream analyses such as clustering, spatial domain identification, and differential gene expression studies without compromising performance or interpretability. One of the major findings in the benchmark pipeline was that each SOTA imputation method highly over-imputes the SRT datasets and fails to differentiate between the biologically unexpressed and valid dropouts. Even the deep learning-based methods also fail. Additionally, the number of feature selection, i.e., the number of top genes selected, has a significant impact on the downstream analysis for SRT datasets. From such interesting findings, we proposed our method that showed consistent performance over different SRT technologies. Our method doesn't over-impute; rather, it can identify valid dropouts utilizing the pro-

posed statistical algorithm. Furthermore, the proposed method has the lowest runtime and memory compared to the other models, demonstrating its significance in terms of computational efficiency.

Chapter 5

Discussion and Conclusion

5.1 Summary

SRT technologies have emerged as powerful tools that bridge spatial context and transcriptomic profiling, enabling in-depth understanding of tissue architecture and cellular heterogeneity. However, the near single-cell resolution achieved by modern SRT technologies often comes at the cost of high sparsity and dropout events in the gene expression matrices, posing significant challenges for downstream analyses such as clustering and spatial domain identification. Imputation methods are usually developed to tackle such high dropouts in genomics data. However, imputation methods developed for SRT technologies are still in the early stages. Here is the summary of this work:

- In this study, we addressed this critical issue through a comprehensive benchmarking of SOTA imputation methods across 23 datasets from five prominent SRT platforms, including 10x Genomics Visium, Stereo-seq, Slide-seqV2, XYZeq, and sci-Space.
- Our benchmarking reveals that no single existing method consistently excels across all datasets and evaluation metrics. Simpler methods (e.g., `MAGIC`, `SimpleImpute`) show limited improvements and often degrade performance under high-sparsity conditions such as for Stereo-seq, Slide-seqV2.
- Deep learning-based models like `scVI`, `gimVI`, and `tangram` demonstrate better performance in certain settings, particularly for highly sparse datasets such as Stereo-seq, Slide-seqV2; however, they incur significantly higher runtime and memory usage due to their complex model architectures. For `10x Genomics Visium`, `gimVI` consistently achieved strong clustering results across all gene subsets, whereas in `Stereo-seq`, `MAGIC` demonstrated better performance. `tangram` showed superior performance for `Slide-seqV2`, while for `sci-Space`, `MAGIC` was most effective. In the case of `XYZeq`, both `gimVI` and `tangram` demonstrated improved clustering outcomes. Notably, `scVI` exhibits the longest runtime while consuming the least memory, whereas `gimVI` and `tangram` are both memory-intensive, reflecting different

trade-offs in model design and spatial integration strategies.

- Additionally, KNN Impute fails to impute properly, while other methods impute all the zeros. Another key finding from the benchmarking pipeline is that most imputation methods, including deep learning-based approaches, tend to over-impute spatial transcriptomics data and struggle to distinguish between biologically unexpressed genes and true dropouts. For example, the sparsity found in RAW data in different technologies varies from 80% to 90%. However, when methods like Soft-Imputeor scVI, gimVI, and tangram are applied, the sparsity in the imputed data becomes completely zero.

- This over-imputation indicates that these methods impute the biologically unexpressed zeros too, distorting the inherent nature of the datasets.

- The number of genes selected for imputation was also varied, and the box-distribution plot indicates in some cases, that the number of feature selection has a strong influence on the downstream metrics, such as in the case of 10x Genomics Visium, the ARI, NMI, AMI, or HOMO scores become higher when all the genes are selected for imputation.

- Motivated by these findings, we introduced `SpaMean-Impute`, a novel imputation approach tailored for SRT data that explicitly incorporates spatial information and includes dropout detection. Unlike other methods, `SpaMean-Impute` distinguishes between technical and biological zeros, reducing the risk of over-imputation.

- Our method achieves superior performance in most scenarios across all evaluation metrics while maintaining low computational overhead. Particularly, in datasets from 10x Genomics Visium, Stereo-seqand sci-Space, `SpaMean-Impute` consistently outperforms all other imputation methods, including the base data, thereby enhancing clustering fidelity and spatial homogeneity. In the case of all slides of 10x Genomics Visium, SpaMean-Impute surpassed all the methods' performance, including the base, in terms of ARI, NMI, AMI, and HOMO.

- In other SRT technologies, SpaMean-Impute outperformed in most cases too. However, these results are for particular gene selection settings, as the benchmark analysis already provided us with insights that the performance of the imputation methods is also correlated to different numbers of features for different SRT technologies.

- Furthermore, SpaMean-Impute can detect valid dropouts and doesn't over-impute. In case of 10x Genomics Visium, SpaMean-Impute detected on average 0.85% dropouts, whereas it detected around 2.43% dropouts in Stereo-seq technology while achieving better performance. This indicates that while imputing valid dropouts, the accuracy increases significantly.

In conclusion, this study underscores the importance of leveraging spatial context in the imputation of SRT data and highlights the limitations of applying generic or scRNA-seq-specific methods in this domain. `SpaMean-Impute` offers a balanced, accurate, and efficient solution for robust SRT data analysis. Future work could explore integrating deep-learning-based techniques to differentiate between dropouts and biological zeros with proper reasoning in spatial genomics.

5.2 Limitation

The proposed imputation method has several advantages compared to the SOTA methods. However, it has several limitations that need to be addressed. Here are some of the limitations of the proposed method:

- The performance of the proposed method, 'SpaMean-Impute' performed significantly well in 10x Genomics Visium technology. In contrast, it did not show a significant enhancement in the case of complex, high-dimensional datasets like Stereo-seq, Slide-seqV2.
- This may be due to the simple architecture of the proposed method. Since the method only imputes using the mean value of its neighbors, the method remains a simple statistical one. However, learning the imputation value may enhance the performance further.
- More complex architecture can be developed in the future to accurately detect the neighbors as well as the valid dropout locations. However, it must be kept in mind that complex architecture can significantly increase the memory consumption and runtime of the model.

5.3 Future Work

There can be several improvements in future works based on the limitations discussed above, such as:

- Future developments could aim to design more adaptive and intelligent imputation strategies that go beyond fixed neighborhood averaging.
- Another promising direction is the integration of SOTA deep learning models along with the statistical approach proposed in this study, which can capture complex nonlinear relationships and spatial dependencies inherent in spatial transcriptomics data. For example, Graph-based deep learning frameworks, such as Graph Neural Networks (GNNs), offer the potential to learn both the spatial topology and gene

expression dynamics in a data-driven manner. Additionally, transformer architectures—particularly those enhanced with attention mechanisms—can be employed to dynamically prioritize informative spatial neighbors or contextual patterns during the imputation process.

- Probabilistic models, such as variational autoencoders (VAEs) and Bayesian neural networks, could also be explored to explicitly model uncertainty in gene dropout events and better infer biologically meaningful missing values. Combining these techniques with contrastive or self-supervised learning could enable robust performance even in low-data or noisy scenarios. To ensure scalability, future work may consider hybrid models that balance expressiveness with computational efficiency or incorporate dimensionality reduction techniques that preserve spatial structure.

Overall, the incorporation of modern deep learning tools along with the present approach presents a promising avenue to improve the accuracy while having the capability of proper dropout detection and imputation. Furthermore, they can also offer generalizability, and interpretability of imputation in spatial transcriptomics.

References

[1] Luyi Tian, Fei Chen, and Evan Z Macosko. The expanding vistas of spatial transcriptomics. Nature Biotechnology, 41(6):773–782, 2023.

[2] Vivien Marx. Method of the year: spatially resolved transcriptomics. Nature methods, 18(1):9–14, 2021.

[3] Ye Wang, Bin Liu, Gexin Zhao, YooJin Lee, Anton Buzdin, Xiaofeng Mu, Joseph Zhao, Hong Chen, and Xinmin Li. Spatial transcriptomics: Technologies, applications and experimental considerations. Genomics, 115(5):110671, 2023.

[4] Tsai-Ying Chen, Li You, Jose Angelito U Hardillo, and Miao-Ping Chien. Spatial transcriptomic technologies. Cells, 12(16):2042, 2023.

[5] Yue You, Yuting Fu, Lanxiang Li, Zhongmin Zhang, Shikai Jia, Shihong Lu, Wenle Ren, Yifang Liu, Yang Xu, Xiaojing Liu, et al. Systematic comparison of sequencing-based spatial transcriptomic methods. Nature Methods, 21(9):1743–1754, 2024.

[6] Ricfrid van der Marel. Navigating the complexity of data imputation in spatial transcriptomics: Strategies, challenges, and future directions. 2024.

[7] Wenpin Hou, Zhicheng Ji, Hongkai Ji, and Stephanie C. Hicks. A systematic evaluation of single-cell rna-sequencing imputation methods. Genome Biology, 21(1):218, Aug 2020.

[8] Mengnan Cheng, Yujia Jiang, Jiangshan Xu, Alexios-Fotios A Mentis, Shuai Wang, Huiwen Zheng, Sunil Kumar Sahu, Longqi Liu, and Xun Xu. Spatially resolved transcriptomics: a comprehensive review of their technological advances, applications, and challenges. Journal of Genetics and Genomics, 50(9):625–640, 2023.

[9] Peng He, Kyungtae Lim, Dawei Sun, Jan Patrick Pett, Quitz Jeng, Krzysztof Polanski, Ziqi Dong, Liam Bolt, Laura Richardson, Lira Mamanova, et al. A human fetal lung cell atlas uncovers proximal-distal gradients of differentiation and key regulators of epithelial fates. Cell, 185(25):4841–4860, 2022.

[10] Sara M Parigi, Ludvig Larsson, Srustidhar Das, Ricardo O Ramirez Flores, Annika Frede, Kumar P Tripathi, Oscar E Diaz, Katja Selin, Rodrigo A Morales, Xinxin Luo,

et al. The spatial transcriptomic landscape of the healing mouse intestine following damage. Nature communications, 13(1):828, 2022.

[11] Amy R Cross, Carlos E de Andrea, María Villalba-Esparza, Manuel F Landecho, Lucia Cerundolo, Praveen Weeratunga, Rachel E Etherington, Laura Denney, Graham Ogg, Ling-Pei Ho, et al. Spatial transcriptomic characterization of covid-19 pneumonitis identifies immune circuits related to tissue injury. JCI insight, 8(2):e157837, 2023.

[12] Jana-Charlotte Hegenbarth, Giuliana Lezzoche, Leon J De Windt, and Monika Stoll. Perspectives on bulk-tissue rna sequencing and single-cell rna sequencing for cardiac transcriptomics. Frontiers in Molecular Medicine, 2:839338, 2022.

[13] Xiangyu Wu, Xin Yang, Yunhan Dai, Zihan Zhao, Junmeng Zhu, Hongqian Guo, and Rong Yang. Single-cell sequencing to multi-omics: technologies and applications. Biomarker Research, 12(1):110, 2024.

[14] Kangning Dong and Shihua Zhang. Deciphering spatial domains from spatially resolved transcriptomics with an adaptive graph attention auto-encoder. Nature communications, 13(1):1739, 2022.

[15] Yunfei Hu, Manfei Xie, Yikang Li, Mingxing Rao, Wenjun Shen, Can Luo, Haoran Qin, Jihoon Baek, and Xin Maizie Zhou. Benchmarking clustering, alignment, and integration methods for spatial transcriptomics. Genome Biology, 25(1):212, 2024.

[16] Tim Lohoff, Shila Ghazanfar, A Missarova, Noushin Koulena, Nico Pierson, Jonathan A Griffiths, Evan S Bardot, C-HL Eng, Richard CV Tyser, Ricard Argelaguet, et al. Integration of spatial and single-cell transcriptomic data elucidates mouse organogenesis. Nature biotechnology, 40(1):74–85, 2022.

[17] Agus Salim, Dharmesh D Bhuva, Carissa Chen, Chin Wee Tan, Pengyi Yang, Melissa J Davis, and Jean YH Yang. Spanorm: spatially-aware normalization for spatial transcriptomics data. Genome Biology, 26(1):109, 2025.

[18] Yue You, Yuting Fu, Lanxiang Li, Zhongmin Zhang, Shikai Jia, Shihong Lu, Wenle Ren, Yifang Liu, Yang Xu, Xiaojing Liu, et al. Systematic comparison of sequencing-based spatial transcriptomic methods. Nature methods, 21(9):1743–1754, 2024.

[19] Yu Zhao, Young Li, Ying He, Junqi Wu, Yi Liu, Xinxing Li, Zhaoxun Li, Qiaomei Yuan, Jialuo Li, Xinya Zhang, et al. Stereo-seq v2: Spatial mapping of total rna on ffpe sections with high resolution. Cell, 2025.

[20] Haoyu Wang, Peng Cheng, Juan Wang, Hongzhi Lv, Jie Han, Zhiyong Hou, Ren Xu, and Wei Chen. Advances in spatial transcriptomics and its application in the musculoskeletal system. Bone Research, 13(1):54, 2025.

[21] Stathis Megas, Anna Wilbrey-Clark, Aidan Maartens, Sarah A Teichmann, and Kerstin B Meyer. Spatial transcriptomics of the respiratory system. Annual Review of Physiology, 87, 2024.

[22] Nuray Sogunmez Erdogan and Deniz Eroglu. Sparse deconvolution of cell type medleys in spatial transcriptomics. PLOS Computational Biology, 21(6):e1013169, 2025.

[23] Sikta Das Adhikari, Jiaxin Yang, Jianrong Wang, and Yuehua Cui. Recent advances in spatially variable gene detection in spatial transcriptomics. Computational and Structural Biotechnology Journal, 23:883–891, 2024.

[24] Shichao Zhang. Nearest neighbor selection for iteratively knn imputation. Journal of Systems and Software, 85(11):2541–2552, 2012.

[25] Rahul Mazumder, Trevor Hastie, and Robert Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. The Journal of Machine Learning Research, 11:2287–2322, 2010.

[26] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12:2825–2830, 2011.

[27] Hongyu Zhang, Weining Li, and Jinting Guan. sctsi: an effective two-stage imputation method for single-cell rna-seq data. Briefings in Bioinformatics, 26(3):bbaf298, 06 2025.

[28] Riasat Azim, Shulin Wang, and Shoaib Ahmed Dipu. Cdsimpute: An ensemble similarity imputation method for single-cell rna sequence dropouts. Computers in Biology and Medicine, 146:105658, 2022.

[29] Lejun Gong, Xiong Cui, Yang Liu, Cai Lin, and Zhihong Gao. Sincwim: An imputation method for single-cell rna sequence dropouts using weighted alternating least squares. Computers in Biology and Medicine, 171:108225, 2024.

[30] Yueying Wang, Kewei Li, Ruochi Zhang, Yusi Fan, Lan Huang, and Fengfeng Zhou. Graceimpute: A novel graph clustering autoencoder approach for imputation of single-cell rna-seq data. Computers in Biology and Medicine, 184:109400, 2025.

[31] Chenggong Han, Qing Xie, and Shili Lin. Are dropout imputation methods for scrna-seq effective for schi-c data? Briefings in Bioinformatics, 22(4):bbaa289, 2021.

[32] Yue Liu, Junfeng Zhang, Shulin Wang, Xiangxiang Zeng, and Wei Zhang. Are dropout imputation methods for scrna-seq effective for scatac-seq data? Briefings in Bioinformatics, 23(1):bbab442, 2022.

[33] Romain Lopez, Achille Nazaret, Maxime Langevin, Jules Samaran, Jeffrey Regier, Michael I Jordan, and Nir Yosef. A joint model of unpaired data from scrna-seq and spatial transcriptomics for imputing missing gene expression measurements. arXiv preprint arXiv:1905.02269, 2019.

[34] Tommaso Biancalani, Gabriele Scalia, Lorenzo Buffoni, Raghav Avasthi, Ziqing Lu, Aman Sanger, Neriman Tokcan, Charles R Vanderburg, Åsa Segerstolpe, Meng Zhang, et al. Deep learning and alignment of spatially resolved single-cell transcriptomes with tangram. Nature methods, 18(11):1352–1362, 2021.

[35] Tianyi Chen, Yunfei Zhang, Lianxin Xie, Wenjun Shen, Si Wu, and Hau-San Wong. Spotdiff: Spatial gene expression imputation diffusion with single-cell rna sequencing data integration. Proceedings of the AAAI Conference on Artificial Intelligence, 39(15):15848–15856, Apr. 2025.

[36] Kongming Li, Jiahao Li, Yuhao Tao, and Fei Wang. stdiff: a diffusion model for imputing spatial transcriptomics through single-cell transcriptomics. Briefings in Bioinformatics, 25(3):bbae171, 04 2024.

[37] Xiaoyu Li, Wenwen Min, Shunfang Wang, Changmiao Wang, and Taosheng Xu. stmcdi: Masked conditional diffusion model with graph neural network for spatial transcriptomics data imputation, 2024.

[38] Ziheng Duan, Dylan Riffle, Ren Li, Junhao Liu, Martin Renqiang Min, and Jing Zhang. Impeller: a path-based heterogeneous graph learning method for spatial transcriptomic data imputation. Bioinformatics, 40(6):btae339, 05 2024.

[39] Yunfei Hu, Yuying Zhao, Curtis T. Schunk, Yingxiang Ma, Tyler Derr, and Xin Maizie Zhou. Adept: Autoencoder with differentially expressed genes and imputation for robust spatial transcriptomics clustering. iScience, 26(6), Jun 2023.

[40] Noah Cohen Kalafut, Xiang Huang, and Daifeng Wang. Joint variational autoencoders for multimodal imputation and embedding. Nature machine intelligence, 5(6):631–642, 2023.

[41] Lifei Wang, Rui Nie, Xuexia Miao, Yankai Cai, Anqi Wang, Hanwen Zhang, Jiang Zhang, and Jun Cai. Inclust+: the deep generative framework with mask modules for multimodal data integration, imputation, and cross-modal generation. BMC bioinformatics, 25(1):41, 2024.

[42] Zhenchao Tang, Guanxing Chen, Shouzhi Chen, Jianhua Yao, Linlin You, and Calvin Yu-Chian Chen. Modal-nexus auto-encoder for multi-modality cellular data integration and imputation. Nature Communications, 15(1):9021, 2024.

[43] Jin-Hong Du, Zhanrui Cai, and Kathryn Roeder. Robust probabilistic modeling for single-cell multimodal mosaic integration and imputation via scvaeit. Proceedings of the National Academy of Sciences, 119(49):e2214414119, 2022.

[44] Gefei Wang, Jia Zhao, Yingxin Lin, Tianyu Liu, Yize Zhao, and Hongyu Zhao. scmodal: a general deep learning framework for comprehensive single-cell multi-omics data alignment with feature links. Nature Communications, 16(1):4994, 2025.

[45] F Alexander Wolf, Philipp Angerer, and Fabian J Theis. Scanpy: large-scale single-cell gene expression data analysis. Genome biology, 19:1–5, 2018.

[46] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426, 2018.

[47] Kristen R Maynard, Leonardo Collado-Torres, Lukas M Weber, Cedric Uytingco, Brianna K Barry, Stephen R Williams, Joseph L Catallini, Matthew N Tran, Zachary Besich, Madhavi Tippani, et al. Transcriptome-scale spatial gene expression in the human dorsolateral prefrontal cortex. Nature neuroscience, 24(3):425–436, 2021.

[48] Hang Xu, Huazhu Fu, Yahui Long, Kok Siong Ang, Raman Sethi, Kelvin Chong, Mengwei Li, Rom Uddamvathanak, Hong Kai Lee, Jingjing Ling, et al. Unsupervised spatially embedded deep representation of spatial transcriptomics. Genome Medicine, 16(1):12, 2024.

[49] Jiangshan Xu, Pengcheng Guo, Shijie Hao, Shuncheng Shangguan, Quan Shi, Giacomo Volpe, Keke Huang, Jing Zuo, Juan An, Yue Yuan, et al. A spatiotemporal atlas of mouse liver homeostasis and regeneration. Nature genetics, 56(5):953–969, 2024.

[50] Robert R Stickels, Evan Murray, Pawan Kumar, Jilong Li, Jamie L Marshall, Daniela J Di Bella, Paola Arlotta, Evan Z Macosko, and Fei Chen. Highly sensitive spatial transcriptomics at near-cellular resolution with slide-seqv2. Nature biotechnology, 39(3):313–319, 2021.

[51] Sanjay R Srivatsan, Mary C Regier, Eliza Barkan, Jennifer M Franks, Jonathan S Packer, Parker Grosjean, Madeleine Duran, Sarah Saxton, Jon J Ladd, Malte Spielmann, et al. Embryo-scale, single-cell spatial transcriptomics. Science, 373(6550):111–117, 2021.

[52] Youjin Lee, Derek Bogdanoff, Yutong Wang, George C Hartoularos, Jonathan M Woo, Cody T Mowery, Hunter M Nisonoff, David S Lee, Yang Sun, James Lee, et al. Xyzeq: Spatially resolved single-cell rna sequencing reveals expression heterogeneity in the tumor microenvironment. Science advances, 7(17):eabg4755, 2021.

[53] Fahim Hafiz. Spatial information matters: Are traditional imputation methods effective for spatial transcriptomics data? https://github.com/fahimhafiz/spameanimpute.

[54] David van Dijk, Juozas Nainys, Roshan Sharma, Pooja Kaithail, Ambrose J Carr, Kevin R Moon, Linas Mazutis, Guy Wolf, Smita Krishnaswamy, and Dana Pe'er. Magic: A diffusion-based imputation method reveals gene-gene interactions in single-cell rna-sequencing data. BioRxiv, page 111591, 2017.

[55] Romain Lopez, Jeffrey Regier, Michael B Cole, Michael I Jordan, and Nir Yosef. Deep generative modeling for single-cell transcriptomics. Nature methods, 15(12):1053–1058, 2018.

[56] Reuben Moncada, Dalia Barkley, Florian Wagner, Marta Chiodin, Joseph C Devlin, Maayan Baron, Cristina H Hajdu, Diane M Simeone, and Itai Yanai. Integrating microarray-based spatial transcriptomics and single-cell RNA-seq reveals tissue architecture in pancreatic ductal adenocarcinomas. Nature Biotechnology, 38(3):333–342, March 2020.

[57] Jiachen Li, Siheng Chen, Xiaoyong Pan, Ye Yuan, and Hong-Bin Shen. Cell clustering for spatial transcriptomics data with graph neural networks. Nature Computational Science, 2(6):399–408, June 2022.

[58] Simone Romano, Nguyen Xuan Vinh, James Bailey, and Karin Verspoor. Adjusting for chance clustering comparison measures. Journal of Machine Learning Research, 17(134):1–32, 2016.

[59] Chen Shengquan, Zhang Boheng, Chen Xiaoyang, Zhang Xuegong, and Jiang Rui. stplus: a reference-based method for the accurate enhancement of spatial transcriptomics. Bioinformatics, 37(Supplement_1):i299–i307, 2021.

Appendix A

Supplementary Materials

A.1 Data and Code Availability

Source codes and datasets for our proposed framework and benchmarking codes can be obtained from: `https://github.com/FahimHafiz/SpaMean-Impute`

A.2 Detailed Performance on sci-Space

The benchmark performance of the imputation methods based on ARI, NMI, AMI, and HOMO scores is presented in Table A.1. We can observe that the MAGICoutperformed all the methods, especially for the 5000 or all genes case.

Table A.1: Performance metrics for different imputation methods applied to sci-Space across varying numbers of top genes

Dataset Name	Metric	Top Genes	base	MAGIC	KNN Im-pute	Soft Im-pute	Simple Im-pute	scVI	gimVI	tan gram
GSE 166692	ARI	2000	0.39	**0.44**	0.39	0.24	0.26	0.39	0.38	0.39
		5000	**0.45**	0.41	**0.45**	0.24	0.35	0.43	0.38	0.44
		all	0.35	**0.47**	0.35	0.22	0.22	0.40	0.41	0.32
	NMI	2000	0.66	**0.68**	0.66	0.49	0.52	0.61	0.62	0.65
		5000	0.67	**0.68**	0.671	0.51	0.58	0.65	0.63	0.672
		all	0.60	**0.70**	0.60	0.42	0.44	0.65	0.65	0.59
	AMI	2000	0.65	**0.67**	0.65	0.49	0.51	0.60	0.61	0.64
		5000	0.66	**0.68**	0.67	0.50	0.57	0.64	0.62	0.66
		all	0.59	**0.69**	0.59	0.41	0.43	0.64	0.63	0.58
	HOMO	2000	0.59	**0.67**	0.59	0.43	0.44	0.57	0.58	0.59
		5000	0.63	**0.69**	0.63	0.44	0.51	0.60	0.58	0.62
		all	0.53	**0.69**	0.53	0.35	0.37	0.60	0.60	0.52

A.3 Sparsity in Slide-seqV2 before and after applying Imputation

Table A.2: Percentage of Zero Sparsity in the Slide-seqV2 datasets before (RAW) and after applying imputation methods.

SL No.	**Top Genes**	**Dataset Name**	**RAW**	**MAGIC**	**KNN Impute**	**Soft Impute**	**Simple Impute**	**scVI**	**gimVI**	**tangram**
01.	2000	WT	93.32	0.29	93.32	0	0	0	0	0
		diabetes	93.53	0.45	93.53	0	0	0	0	0
		mouse	96.52	1.14	96.52	0	0	0	0	0
02.	5000	WT	93.90	0.52	93.90	0	0	0	0	0
		diabetes	93.85	0.60	93.85	0	0	0	0	0
03.	all	WT	94.02	0.76	94.02	0	0	0	0	0
		diabetes	93.73	0.96	93.73	0	0	0	0	0
		mouse	95.78	2.03	95.78	0	0	0	0	0

A.4 Sparsity in XYZeq before and after applying Imputation

Table A.3: Percentage of Zero Sparsity in the XYZeq datasets before (RAW) and after applying imputation methods.

SL No.	**Top Genes**	**Dataset Name**	**RAW**	**MAGIC**	**KNN Impute**	**Soft Impute**	**Simple Impute**	**scVI**	**gimVI**	**tangram**
03.	2000	GSE 164430	97.8	14.3	97.8	0	0	0	0	0
		GSM 5009529	98.6	7.6	98.6	0	0	0	0	0
		GSM 5009539	98.6	6.8	98.6	0	0	0	0	0
03.	5000	GSE 164430	97.17	18.02	97.17	0	0	0	0	0
		GSM 5009529	98.26	8.18	98.26	0	0	0	0	0
		GSM 5009539	98.24	8.46	98.24	0	0	0	0	0
03.	all	GSE 164430	96.62	16.77	96.62	0	0	0	0	0
		GSM 5009529	97.87	8.86	97.87	0	0	0	0	0
		GSM 5009539	97.91	9.29	97.91	0	0	0	0	0

A.5 Sparsity in sci-Space before and after applying Imputation

Table A.4: Percentage of Zero Sparsity in the sci-Space datasets before (RAW) and after applying imputation methods.

SL No.	**Top Genes**	**Dataset Name**	**RAW**	**MAGIC**	**KNN Impute**	**Soft Impute**	**Simple Impute**	**scVI**	**gimVI**	**tangram**
01.	2000	GSE 166692	86.61	3.99	86.61	0	0	0	0	0
02.	5000	GSE 166692	87.66	4.64	87.66	0	0	0	0	0
03.	all	GSE 166692	85.86	5.22	85.86	0	0	0	0	0

A.6 Computational Performance of 'SpaMean-Impute' on SRT Technologies (excluding 10x Genomics Visium)

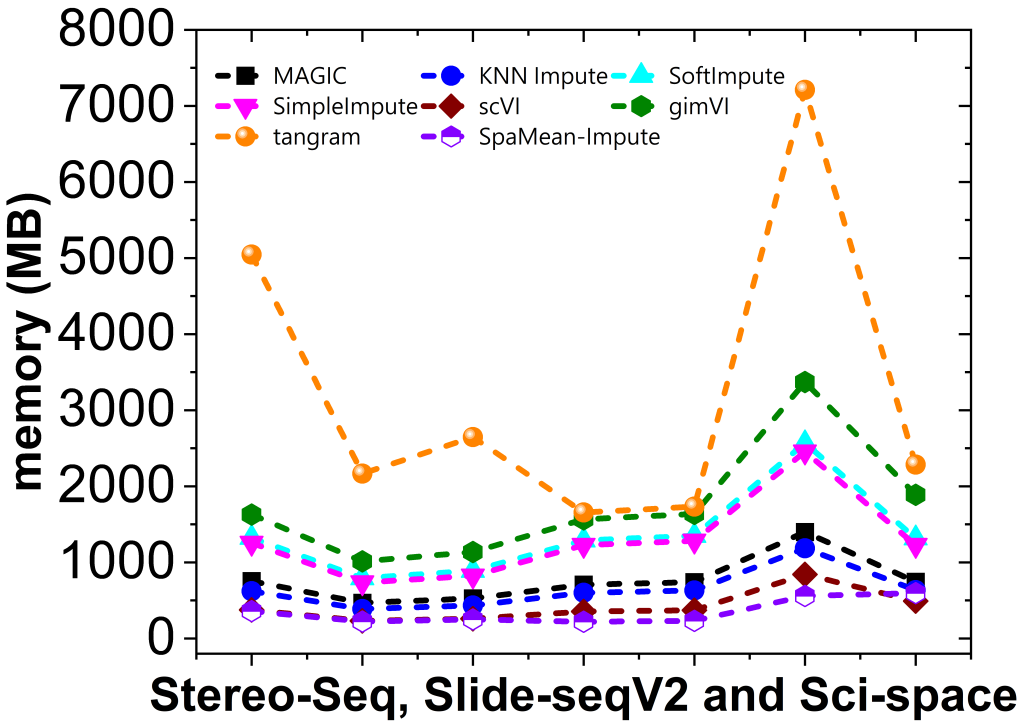


Figure A.1: Memory consumption comparison of 'SpaMean-Impute' with others for Stereo-seq, Slide-seqV2, sci-Space. 'tangram' consumes the highest memory as well, and 'SpaMean-Impute' consumes the lowest.

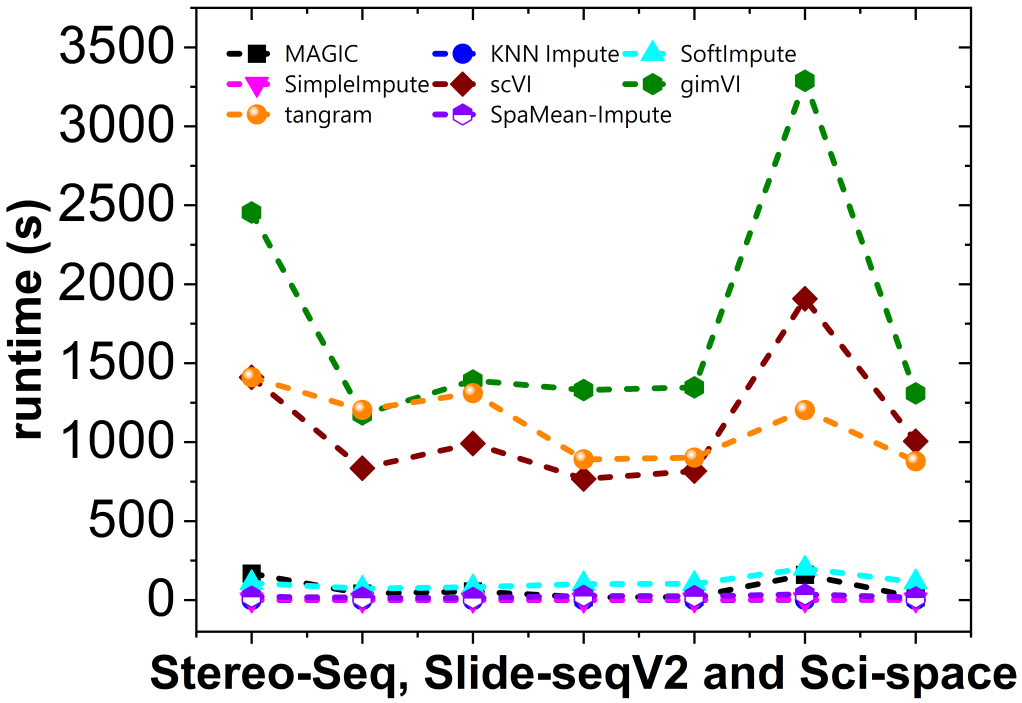


Figure A.2: Runtime comparison of 'SpaMean-Impute' with others for Stereo-seq, Slide-seqV2, sci-Space. 'gimVI' takes the highest runtime here as well, and 'SpaMean-Impute' takes the lowest.