Bi-channel Masked Graph Autoencoders for Spatially Resolved Single-cell Transcriptomics Data Imputation

Anonymous Author(s)
Affiliation
Address
email

Abstract

Spatially resolved transcriptomics bring exciting breakthroughs to single-cell analysis by providing physical locations along with gene expression. However, as a cost of the extremely high resolution, the technology also results in much more missing values in the data, i.e. dropouts. While a common solution is to perform imputation on the missing values, existing imputation methods majorly focus on transcriptomics data and tend to yield sub-optimal performance on spatial transcriptomics data. To advance spatial transcriptomics imputation, we propose a new technique to adaptively exploit the spatial information of cells and the heterogeneity among different cell types. Furthermore, we adopt a mask-then-predict paradigm to explicitly model the recovery of dropouts and enhance the denoising effect. Compared to existing studies, our work focuses on new large-scale cell-level data instead of spots or beads. Preliminary results have demonstrated that our method outperforms existing methods for removing dropouts in high-resolution spatially resolved transcriptomics data.

1 Introduction

The rapid advance of single-cell technologies has paved the way to investigate biological processes at an unprecedentedly high resolution. However, most single-cell technologies such as scRNA-seq [36] and scATAC-seq [6] are unable to capture spatial information of the cells, which is often coupled with cell identities. To spatially resolve transcriptomics profiles, spatial transcriptomic technologies have been recently developed where researchers are able to identify the spatial context of cells and cell clusters within a tissue [7]. Similar to standard single-cell RNA-sequencing, spatially resolved profiling technology may fail to capture a significant number of the expressed genes due to the low RNA capture rate. This leads to an artificially large proportion of false zero counts, i.e., the so-called "dropout" [34, 19]. A popular way to address this issue is to perform imputation, which corrects false zeros by estimating realistic values for those missing values. Recently, there has been an explosive growth of imputation methods for single-cell transcriptomics data, focusing on generative probability models, matrix factorization [38, 18, 37, 23] and deep learning models [35, 11, 8, 42, 29].

Although promising results have been achieved on single-cell RNA-sequencing data, it remains challenging to impute spatially resolved single-cell transcriptomics data. Due to the relatively low sequencing depth in each unit and the extra experimental processes to obtain spatial locations, spatial transcriptomics technology suffers from a higher noise level and drop-out rate (zero reads for many genes). Furthermore, most existing imputation models do not take advantage of spatial information and it is still unclear how to advance imputation with spatial information. To leverage the spatial

Submitted to NeurIPS 2021 AI for Science Workshop.

information, very recently graph-based methods [22, 43] have been designed. They first build a cell-cell graph based on the spatial position and then apply graph neural networks to the built graph. However, these methods do not consider heterogeneity among different types of cells. For example, tumor cells usually show strong aggregation, and neighboring tumor cells with similar micro-environment in terms of ligands tend to have higher gene expression similarity than distant cells [5, 21, 26]. By contrast, Treg cells are scarce spatially in many tissues but still tend to have similar gene expression profiles [31]. Hence, it is desired to rely on spatial information to impute cells like tumors, while leveraging information from the same type of cells to impute cells like Treg cells. To capture such heterogeneity while exploiting spatial information, we propose a novel bi-channel graph neural network (GNN). Specifically, we first construct two channels: a spatial neighboring graph capturing the spatial information, and a dynamic k-nearest neighbor (kNN) graph capturing the similarity of cells. Then we adopt a gating unit to adaptively fuse information from the two graphs, which adjusts the exploitation of spatial information and heterogeneity among different types of cells.

Furthermore, inspired by the denoising effect of masked graph autoencoders [16] which recovers the manually masked node features through a GNN, we introduce a novel mask-then-predict paradigm for training the proposed imputation method. Specifically, we randomly mask 60% of genes with zero in each cell and train our bi-channel graph neural network to recover the input expression. This training paradigm explicitly models the recovery of dropouts with graph neural networks. We evaluate our method on an unpublished dataset produced by CosMx Spatial Molecular Imager (SMI) [15]. Unlike existing spot-level datasets (e.g. 10x Visium [33]) or tiny cell-level datasets (e.g. MERFISH [44], Fishseq+ [10]), CosMx platform provides unprecedentedly large-scale single-cell spatial transcriptomics datasets, with more than 100,000 cells in each slice. Our proposed method achieves state-of-the-art performance on this dataset and significantly outperforms other baseline models. Our contribution can be summarized as follows:

- We propose a novel bi-channel graph neural network to leverage information from both spatial neighboring cells and similar cells to advance spatial transcriptomics data imputation.
- We introduce a mask-then-predict paradigm as the training objective to explicitly model the recovery of dropouts in spatial transcriptomics data, which shows outstanding performance.
- We make the first attempt to evaluate imputation methods on large-scale single-cell spatial transcriptomics datasets. Our method improves the data quality and makes downstream tasks more viable while providing a strong baseline for future spatial imputation models.

The rest of the paper is organized as follows. In Section 2, we review related work. We detail our proposed framework in Section 3 and discuss our experimental results in Section 4. Finally, we conclude the work with future directions in Section 5.

2 Related Work

2.1 Spatial Resolved Transcriptomics

While single-cell technologies are able to capture transcriptomic information at the single-cell level, most are unable to capture spatial information of the cells, due to the isolation of single-cells. However, the relationship between cells and their relative locations within tissue is critical to understanding normal development and disease pathology. Spatial transcriptomic technologies are a more recent development in transcriptomics profiling, which are able to spatially resolve transcriptomics profiles. With spatial transcriptomics technologies, researchers are able to learn the spatial context of cells, and cell clusters within a tissue [7]. Most commercial spatially resolved profiling technologies are not at the single-cell resolution that has arisen the problem of cell-type deconvolution and segmentation [33, 27].

The major technologies/platforms for spatial transcriptomics are Visium by 10x [33], GeoMx Digital Spatial Profiler (DSP) [27] by NanoString and CosMx Spatial Molecular Imager (SMI) by NanoString, MERFISH, Vizgen, Resolve, Rebus, and molecular cartography. 10x Visium does not profile at single-cell resolution, and while GeoMx DSP is capable of single-cell resolution through user-drawn profiling

regions, the scalability is limited. The most recent platform, CosMx Spatial Molecular Imager (SMI) [15], is able to profile consistently at single cell, and even sub-cellular resolution. CosMx SMI follows much of the initial protocol as GeoMx DSP, with barcoding and ISH hybridization. However, the SMI instrument performs 16 cycles of automated cyclic readout, and in each cycle the set of barcodes (readouts) are UV-cleaved and removed. These cycles of hybridization and imaging yield spatial resolved profiling of RNA and protein at single-cell ($\sim 10\mu m$) and subcellular ($\sim 1\mu m$) resolution.

In this work, we use an unpublished dataset produced by the CosMx platform. In order to obtain the cell level gene expression, we utilize CellPose [32] software to conduct cell segmentation.

2.2 Single-cell Transcriptomics Data Imputation

The increased resolution of transcriptomics profiling methods comes at a cost of increasing data sparsity. The profiling technology may fail to capture a number of the expressed genes of an individual cell due to low RNA capture rate as well as the stochasticity of mRNA expression. This results in a large proportion of false zero counts, defined as "dropout" [34, 19]. A popular way to address dropout is to perform imputation, which aims to correct false zeros by estimating realistic values for those gene-cell pairs. A large number of methods have been developed for the task of scRNA-seq data imputation, mainly focusing on generative probability models or matrix factorization [38, 30, 14, 18]. Aside from these methods, deep learning models have gained immense popularity over recent years. A natural deep learning architecture for the imputation task is the autoencoder, due to its prevalence in data denoising and missing data applications [28, 12, 13, 3, 4]. For example, as one of the early autoencoder methods for scRNA imputation, AutoImpute [35] employs an overcomplete autoencoder model rather than the usual undercomplete one. Later, the deep count autoencoder (DCA) [11] is developed for single-cell transcriptomics data imputation. In contrast to the overcomplete model of AutoImpute, DCA takes a form similar to a standard variational autoencoder (VAE). Moreover, DeepImpute [2] builds multiple neural networks in parallel to impute target genes using a set of input genes.

2.3 Denoising Transciptomics Data via Graph Neural Networks

Despite the huge success achieved by deep generative models such as VAEs in single-cell data imputation, a new group of methods based on graph neural networks [25] have recently gained increasing attention. For instance, scGNN [42] first builds a cell-cell graph based on gene expression similarity and then utilizes a graph autoencoder together with a standard autoencoder to refine graph structures and cell representation. Lastly, an imputation autoencoder is trained with a graph Laplacian smoothing term added to the reconstruction loss. GraphSCI [29] constructs a graph from the data with genes taken as the nodes and the edges between them given by the correlation coefficient of the expression data. Given this graph, the autoencoder and graph autoencoder make use of the expression data and graph data to reconstruct its input.

More recently, with the development of new spatial transcriptomics technologies, some graph neural networks methods are introduced [17, 22, 9, 43] to address data noise in spatial transcriptomics data. SpaGCN [17] first constructs an undirected weighted graph of spots from the spatial transcriptomics data, where the edge weight is determined by the distance between spots. A popular GNN model called GCN [20] is then utilized to aggregate spot gene expression from neighborhoods and update spot gene expression. Similarly, CCST [22] constructs a spatial neighboring cell graph, and introduces a self-supervised graph neural network model, deep graph infomax (DGI) [40], to obtain cell representations from the graph. Moreover, STAGATE [9] also constructs a spatial cell graph but uses a different aggregation method, graph attention network [39], to learn low-dimensional latent embeddings with an autoencoder framework. Notably, all these methods leverage the denoising effect of graph neural networks but they are not specifically designed for imputation. Meanwhile, most of them are only demonstrated on spot-level spatial transcriptomics datasets. To leverage the spatial information, the latest method, Sprod [43] first projects transcriptomics features to a latent space, then connects

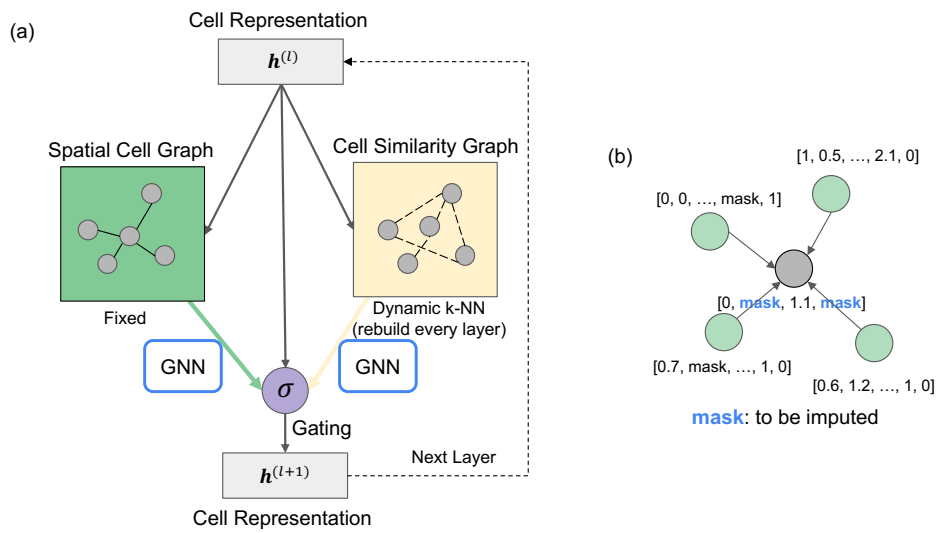

Figure 1: An illustration of the proposed method. (a) A Bi-channel graph neural network with a fixed spatial cell graph and a dynamic kNN cell similarity graph. (b) Masked graph autoencoders that explicitly takes imputation as the training objective.

neighboring cells in the latent space, and prunes it with physical distance. Then a denoised matrix is learned by minimizing the reconstruction error and a graph Laplacian smoothing term.

Different from the aforementioned methods, we propose to construct two separate graphs for spatial relations and cell similarity. In addition, we introduce a novel mask-then-predict paradigm as the training objective to explicitly model the recovery of dropouts in spatial transcriptomics data.

3 The Proposed Framework

We first introduce some notations and basic concepts. A spatial transcriptomics dataset typically consists of two parts: the gene expressions and the spatial positions. We denote the gene expression data as a matrix $\mathbf{X} \in \mathcal{R}^{N \times k}$, where N is the number of spots or cells (depending on the dataset) and k is the number of gene types measured in the dataset. Hereby, $\mathbf{X}_{i,j}$ represents the count of gene-j captured in corresponding spot-i or cell-i. We use another matrix $\mathbf{C} \in \mathcal{R}^{N \times 2}$ to denote the two-dimensional coordinates of each spot or cell. Note that in the case of single-cell level datasets, the source data are preprocessed with a cell segmentation method and the whole space are separated into individual cell regions. The gene expression matrix $\mathbf{X}$ is generated by counting RNA molecules in each cell region, and the coordinates $\mathbf{C}$ is the center position of each cell region.

In the spatial transcriptomics imputation problem, a part of values in $\mathbf{X}$ are missing, denoted as a mask matrix $\mathbf{M} \in \{0, 1\}^{N \times k}$, where the value of $\mathbf{X}_{i,j}$ can only be observed when $\mathbf{M}_{i,j} = 1$. A partially observed data matrix $\widetilde{\mathbf{X}}$ is hereby defined as:

$$\widetilde{\mathbf{X}}_{i,j} = \begin{cases} 0 & \mathbf{M}_{i,j} = 0 \\ \mathbf{X}_{i,j} & \mathbf{M}_{i,j} = 1 \end{cases} \tag{1}$$

Our goal is to predict the missing values $\mathbf{X}_{i,j}$ at $\mathbf{M}_{i,j} = 0$, given the partial gene expression data $\widetilde{\mathbf{X}}_{i,j}$ and the spatial positions $\mathbf{C}$. In the following, we present the details of our proposed framework.

3.1 Cell Graph Construction with Spatial Distance and Gene Expression Similarity

Based on gene spatial distance and expression similarity, there are multiple ways to construct a cell graph and correspondingly facilitate imputation. For instance, scGNN [42] constructed a k-nearest-neighbor graph using only gene expressions; CCST [22] and STAGATE [9] constructed spatial

neighboring graphs that only connect adjacent cells; and Sprod [43] first projects transcriptomics features to a latent space, then connects neighboring cells in the latent space and prune it with physical distance. In this work, we propose to construct two separate graphs for spatial relations and cell similarity respectively. This will enable our model to impute different types of cells by adaptively leveraging both spatial and similarity information.

A spatial cell graph is denoted as $\mathcal{G}_{\text{spa}} = (\mathcal{V}, \mathcal{E}_{\text{spa}})$, where $\mathcal{V}$ is the set of cells, and $\mathcal{E}_{\text{spa}}$ is the set of edges. Here, the edges describe the spatial relations between cells and can also be represented by an adjacency matrix $\mathbf{A}_{\text{spa}} \in \mathcal{R}^{N\times N}$. To construct $\mathcal{G}_{\text{spa}}$, we first calculate pairwise euclidean distances between cells. Concretely, a distance matrix $\mathbf{D} \in \mathcal{R}^{N\times N}$ is calculated as:

$$\mathbf{D}_{i,j} = ||\mathbf{C}_i - \mathbf{C}_j|| = \sqrt{(\mathbf{C}_{i,0} - \mathbf{C}_{j,0})^2 + (\mathbf{C}_{i,1} - \mathbf{C}_{j,1})^2} \tag{2}$$

We then build the adjacency matrix $\mathbf{A}_{\text{spa}}$ by setting a threshold p on the distance matrix:

$$\mathbf{A}_{\text{spa}} = \begin{cases} 0 & \mathbf{D}_{i,j} > p \\ 1 & \mathbf{D}_{i,j} < p \end{cases} \tag{3}$$

In addition, we construct a dynamic cell similarity graph, denoted as $\mathcal{G}^{(l)}_{\text{sim}} = (\mathcal{V}, \mathcal{E}^{(l)}_{\text{sim}})$, where we exploit the low-dimensional hidden states from our model. To be concrete, let $\mathbf{H}^{(l-1)} \in \mathcal{R}^{N\times d^{(l-1)}}$ and $\mathbf{H}^{(l)} \in \mathcal{R}^{N\times d^{(l)}}$ be the input and output cell representations of the l-th layer, where $d^{(l)}$ is pre-defined hidden size. $\mathbf{H}^{(0)}$ can be initialized by a low-dimensional representation generated from $\hat{\mathbf{X}}$ with unsupervised algorithms such as principle component analysis (PCA) [1]. We construct a distance matrix $\hat{\mathbf{D}}^{(l)}$ in each layer, using a formula similar to Eq 2, while $\mathbf{C}$ is replaced with $\mathbf{H}^{(l-1)}$. After that, a similarity-based adjacency matrix $\mathbf{A}^{(l)}_{\text{sim}} \in \mathcal{R}^{N\times N}$ is built with a k-nearest neighbor algorithm. Formally,

$$\mathbf{A}^{(l)}_{\text{sim}} = \begin{cases} 0 & j \notin \mathcal{N}^{(l)}_k(i) \\ 1 & j \in \mathcal{N}^{(l)}_k(i) \end{cases} \tag{4}$$

where $\mathcal{N}^{(l)}_k(i)$ denotes the set containing the k-nearest neighbors of cell i according to the distance matrix $\hat{\mathbf{D}}^{(l)}$. In practice, we separate cells into fields of view (FOVs), where we only calculate distances between cells within the same FOV. This allows our model to have the potential to be scalable to millions of cells.

3.2 Bi-channel Graph Neural Network

Our bi-channel graph neural network layer as shown in Figure 3 performs aggregation on both the spatial cell graph $\mathcal{G}_{\text{spa}}$ and the cell similarity graph $\mathcal{G}^{(l)}_{\text{sim}}$ and then updates the node features, i.e., cell representations. Essentially, it consists of three parts: a spatial graph attention layer, a similarity graph attention layer, and a gating unit. The two graph attention layers (GATs) [39] aggregate the information from $\mathcal{G}_{\text{spa}}$ and $\mathcal{G}^{(l)}_{\text{sim}}$ parallelly. Then a gating unit fuses information from the two GATs, and generates the output of the whole layer. Formally, a GAT layer can be formulated as:

$$\mathbf{z}^{(l+1)}_i = \mathbf{W}^{(l)}\sigma\left(\sum\nolimits_{j\in\mathcal{N}(i)} \alpha^{(l)}_{i,j}\mathbf{z}^{(l)}_j\right), \quad \text{with } \alpha^{(l)}_{i,j} = \frac{\exp\left(\sigma\left(\mathbf{a}^{(l)}\cdot\left[\mathbf{z}^{(l)}_i \| \mathbf{z}^{(l)}_j\right]\right)\right)}{\sum_{k\in\mathcal{N}(i)}\exp\left(\sigma\left(\mathbf{a}^{(l)}\cdot\left[\mathbf{z}^{(l)}_i \| \mathbf{z}^{(l)}_k\right]\right)\right)}, \tag{5}$$

where $\mathbf{z}^{(l)}_i$ denotes the output node features of node i of the l-th layer; σ is an arbitrary non-linear activation function; $\mathcal{N}(i)$ is the set of neighboring nodes of node i, $\mathbf{W}^{(l)}$ is a learnable weight matrix for linear transformation, $[\cdot\|\cdot]$ indicates the concatenation operation, $\mathbf{a}^{(l)}$ is a learnable weight vector for attention and $\alpha^{(l)}_{ij}$ is the attention score between node i and node j.

In a bi-channel graph neural network layer, we have two GAT layers for $\mathcal{G}_{\text{spa}}$ and $\mathcal{G}^{(l)}_{\text{sim}}$, respectively. One is parameterized by $\mathbf{W}^{(l)}_{\text{spa}}$ and $\mathbf{a}^{(l)}_{\text{spa}}$, denoted as $\text{GAT}^{(l)}_{\text{spa}}$. The other is parameterized by $\mathbf{W}^{(l)}_{\text{sim}}$ and

$\mathbf{a}_{\text{sim}}^{(l)}$, denoted as $\text{GAT}_{\text{sim}}^{(l)}$. Formally, the input and output of $\text{GAT}_{\text{spa}}^{(l)}$ and $\text{GAT}_{\text{sim}}^{(l)}$ can be written as:

$$\mathbf{H}_{\text{spa}}^{(l)} = \text{GAT}_{\text{spa}}^{(l)}\left(\mathbf{A}_{\text{spa}}, \mathbf{H}^{(l-1)}\right), \mathbf{H}_{\text{sim}}^{(l)} = \text{GAT}_{\text{sim}}^{(l)}\left(\mathbf{A}_{\text{sim}}^{(l)}, \mathbf{H}^{(l-1)}\right) \tag{6}$$

Note that $\mathcal{G}_{\text{spa}}$ and $\mathcal{G}_{\text{sim}}^{(l)}$ share the same set of nodes $\mathcal{V}$ (i.e. the same set of cells), and the two GAT layers share the same input node features (a.k.a $\mathbf{H}^{(l-1)}$). The difference between $\mathcal{G}_{\text{spa}}$ and $\mathcal{G}_{\text{sim}}^{(l)}$ lies in the adjacency matrix $\mathbf{A}_{\text{spa}}$ and $\mathbf{A}_{\text{sim}}^{(l)}$. Hence, $\mathcal{N}(i)$ in Eq 5 is different in $\text{GAT}_{\text{spa}}^{(l)}$ and $\text{GAT}_{\text{sim}}^{(l)}$.

Next, we define a gating unit as follows:

$$\Theta^{(l)} = \text{SOFTMAX}\left(\mathbf{W}_{\text{gate}}^{(l)}\mathbf{h}^{(l-1)}\right) \tag{7}$$

$$\mathbf{h}^{l} = \theta_0^{(l)}\mathbf{h}_{\text{spa}}^{(l)} + \theta_1^{(l)}\mathbf{h}_{\text{sim}}^{(l)} + \theta_2^{(l)}\mathbf{h}^{(l-1)} \tag{8}$$

where $\mathbf{h}^{(l-1)}$ is the input node embedding of any node in layer l, $\mathbf{h}_{\text{spa}}^{(l)}$ is the output node embedding from $\text{GAT}_{\text{spa}}^{(l)}$, $\mathbf{h}_{\text{sim}}^{(l)}$ is the output node embedding from $\text{GAT}_{\text{sim}}^{(l)}$, $\mathbf{W}_{\text{gate}}^{(l)} \in \mathcal{R}^{3\times d^{(l-1)}}$ is a learnable matrix that project node embedding to a 3-dimensional vector $\Theta^{(l)}$, and $\theta_i^{(l)}$ is the i-th element of vector $\Theta^{(l)}$. As a result, the new embedding of each node is a weighted sum of the output of $\text{GAT}_{\text{spa}}^{(l)}$, the output of $\text{GAT}_{\text{sim}}^{(l)}$, and the old embedding. In this way, our bi-channel graph neural network layer fuses information from $\mathcal{G}_{\text{spa}}$ and $\mathcal{G}_{\text{sim}}^{(l)}$.

Overall, a bi-channel graph neural network layer $\phi^{(l)}$ can be summarized as:

$$\mathbf{H}^{(l)} = \phi^{(l)}\left(\mathbf{H}^{(l-1)}, \mathbf{A}_{\text{spa}}, \mathbf{A}_{\text{sim}}^{(l)}\right) \tag{9}$$

where $\mathbf{H}^{(l)}$ will be used to build $\mathbf{A}_{\text{sim}}^{(l+1)}$ and fed to layer l+1. The similarity graph $\mathcal{G}_{\text{sim}}^{(l)}$ is therefore gradually refined through layers.

3.3 Graph Masked Autoencoders

The idea of graph masked autoencoders is to train an encoder-decoder architecture to recover some artificially added dropouts in the graph. This serves as a self-supervised pretraining objective and the trained encoder can be applied to different downstream tasks. The overall structure is similar to a denoising autoencoder[41], while both the encoder and the decoder can be instantiated as a graph neural network, and both edges and nodes in the input graph can be modified to create noisy input. Here, we follow the framework of GraphMAE [16] to only mask node features as shown in Figure 3.

More specifically, we uniformly sample a mask $\mathbf{M}' \in \{0,1\}^{N\times k}$. To be consistent with previous sections, we denote the original input gene expression matrix as $\widetilde{\mathbf{X}}$. The node feature $\widetilde{\mathbf{X}}_{i,j}$ is masked when $\mathbf{M}'_{i,j} = 0$, resulting in a new node feature matrix $\widetilde{\mathbf{X}}' \in \mathcal{R}^{N\times k}$ written as:

$$\widetilde{\mathbf{X}}' = \widetilde{\mathbf{M}} \odot \widetilde{\mathbf{X}} \tag{10}$$

where $\odot$ indicates element-wise multiplication. Note that the generation process of $\widetilde{\mathbf{X}}'$ perfectly matches the generation of $\widetilde{\mathbf{X}}$, which means the optimization objective is consistent with our imputation goal. This is different from previous studies, as we discussed in Section 2.3. During training, masked features $\widetilde{\mathbf{X}}'$ are used as initial node features for $\phi^{(1)}$, i.e. $\mathbf{H}^{(0)} = \widetilde{\mathbf{X}}'$. As we mentioned in Section 3.1, we can also add dimension reduction methods, such as $\mathbf{H}^{(0)} = \text{PCA}\left(\widetilde{\mathbf{X}}'\right)$. For testing, the original input $\widetilde{\mathbf{X}}$ is enabled, which provides extra information for imputation.

Finally, let L be the total number of graph neural network layers, we consider all those L layers as an encoder. Therefore, $H^{(l)}$ is the latent code that is the output from the encoder. We then add a fully connected decoder to predict the unmasked features, the prediction is thus denoted as:

$$\mathbf{Z} = \sigma\left(\mathbf{H}^{(l)}\mathbf{W}_{\text{o}}\right) \tag{11}$$

where $\mathbf{W}_{\text{o}} \in \mathcal{R}^{d^{(l)}\times k}$ is the learnble weight matrix, and $\mathbf{Z} \in \mathcal{R}^{N\times k}$ is the recovered gene expression matrix.

3.4 The Objective Function

As a masked autoencoder, the objective function for the whole framework can be defined as:

$$\mathcal{L}_{\text{MSE}} = \left\| (1 - \mathbf{M}') \odot \left(\mathbf{Z} - \widetilde{\mathbf{X}} \right) \right\|_F^2 \tag{12}$$

where we only calculate mean squared error (MSE) for the prediction of masked values.

In addition, due to the data sparsity, most values in $\widetilde{\mathbf{X}}$ are zero, therefore most predictions are also expected to be zero. To address this kind of unbalanced data, a weighted MSE loss is alternatively introduced, defined as:

$$\mathcal{L}_{\text{WMSE}} = \left\| (1 - \mathbf{M}') \odot \left(\left(\widetilde{\mathbf{X}} + \gamma \right) \cdot \left(\mathbf{Z} - \widetilde{\mathbf{X}} \right) \right) \right\|_F^2 \tag{13}$$

where we use the ground truth value in $\widetilde{\mathbf{X}}$ to assign a weight encouraging non-zero expressions, and γ is a scaling factor that controls this effect.

After the model is trained, we run our model on the original input matrix $\widetilde{\mathbf{X}}$, and the resulting recovered gene expression matrix $\mathbf{Z}$ will be the final imputation result.

4 Experiment

In this section, we evaluate the effectiveness of our methodon spatial transcriptomics data imputation. Before presenting our experimental results and observations, we first introduce the experimental settings.

4.1 Experimental settings

4.1.1 Datasets

We illustrate the effectiveness of our algorithm on a spatial transcriptomics dataset over human Non-Small Cell Lung Cancer (NSCLC) Formalin-Fixed Paraffin-Embedded (FFPE) slides using the CosMX platform from Nanostring [15]. In this data, 100,149 cells were profiled on 960 pre-selected marker genes and 20 negative probe controls with an average of 300 transcripts per cell and an average dropout rate of 87%.

4.1.2 Baselines

To evaluate the effectiveness of our method, we compare it with the state-of-the-art spatial and non-spatial imputation models:

- **scGNN** [42] first builds a cell-cell graph based on gene expression similarity and then utilizes a graph autoencoder together with a standard autoencoder to refine graph structures and cell representation. Lastly, an imputation autoencoder is trained with a graph Laplacian smoothing term added to the reconstruction loss.
- **scVI** [24] is based on a hierarchical Bayesian model with conditional distributions specified by neural networks. It models dropouts with a zero-inflated negative binomial distribution, and can estimate the distributional parameters of each gene in each cell.

We intended to run another state-of-the-art model Sprod [43] on our dataset. However, we failed to get results within 8 hours. We also lack comparisons with other methods, e.g. scImpute [23], SAVER [18], etc. This is still a work-in-progress, we will present a more comprehensive experiment when officially published.

4.1.3 Implementation Settings

For baseline models, we follow the default settings on authors' github. Regarding preprocessing, we remove cells with no gene expressions. For scVI, the input is raw read counts of genes. While for

	r=0.1		r=0.3	
	RMSE	Correlation	RMSE	Correlation
scVI	0.406	0.561	0.318	0.610
scGNN	**0.294**	0.668	0.407	0.531
Ours (only spatial)	0.329	0.706	0.290	0.684
Ours (only knn)	0.317	0.694	0.291	0.686
Ours	0.317	**0.710**	**0.284**	**0.690**

Table 1: The experiment results show that our method outperforms other baselines.

other models, we use log-normalized expression data. To create the training data, we randomly set a certain ratio r of gene expressions to be zero, to simulate different degrees of noise. The experiments are conducted under two different settings, $r = 0.3$ and $r = 0.1$. Lastly, all experiments are run 5 times, and the average performance is reported.

4.2 Imputation Performance

To evaluate the imputation performance, we use values that are dropped in training data as groundtruth labels and compare them with the imputed values from the imputation model. The metrics we use are root mean squared error (RMSE, the lower, the better) and Pearson correlation coefficient (Correlation, the higher the better). All evaluations are conducted based on log-normalized values.

In Table 1, we present the experiment results. It is shown that our method significantly outperforms other baselines by showing a higher correlation, and the ablation study shows that both spatial graph and kNN graph benefit the imputation performance.

5 Conclusion

In this work, we propose a new technique to adaptively exploit the spatial information of cells and the heterogeneity among different types of cells for imputation. Furthermore, a mask-then-predict paradigm is adopted to explicitly model the recovery of dropouts and enhance the denoising effect. Compared to existing studies, our work focus on new large-scale cell-level data instead of spots or beads. Preliminary results have demonstrated that our method outperforms existing methods for removing dropouts in high-resolution spatially resolved transcriptomics data. This is a work-in-progress and we will continue to add more comprehensive experiments, including clustering results, visualizations and model interpretability analysis.

References

[1] Hervé Abdi and Lynne J Williams. 2010. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics* 2, 4 (2010), 433–459.

[2] Cédric Arisdakessian, Olivier Poirion, Breck Yunits, Xun Zhu, and Lana X Garmire. 2019. DeepImpute: an accurate, fast, and scalable deep neural network method to impute single-cell RNA-seq data. *Genome biology* 20, 1 (2019), 1–14.

[3] Brett K Beaulieu-Jones, Jason H Moore, and POOLED RESOURCE OPEN-ACCESS ALS CLINICAL TRIALS CONSORTIUM. 2017. Missing data imputation in the electronic health record using deeply learned autoencoders. In *Pacific symposium on biocomputing 2017*. World Scientific, 207–218.

[4] Guillem Boquet, Jose Lopez Vicario, Antoni Morell, and Javier Serrano. 2019. Missing data in traffic estimation: A variational autoencoder imputation method. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2882–2886.

[5] Robin Browaeys, Wouter Saelens, and Yvan Saeys. 2020. NicheNet: modeling intercellular communication by linking ligands to target genes. *Nature methods* 17, 2 (2020), 159–162.

[6] Jason D. Buenrostro, Beijing Wu, Ulrike M. Litzenburger, Dave Ruff, Michael L. Gonzales, Michael P. Snyder, Howard Y. Chang, and William J. Greenleaf. 2016. Single-cell chromatin accessibility reveals principles of regulatory variation. *Nature* 523, 7561 (2016), 486—490.

[7] Darren J Burgess. 2019. Spatial transcriptomics coming of age. *Nature Reviews Genetics* 20, 6 (2019), 317–317.

[8] Yue Deng, Feng Bao, Qionghai Dai, Lani F Wu, and Steven J Altschuler. 2018. Massive single-cell RNA-seq analysis and imputation via deep learning. *BioRxiv* (2018), 315556.

[9] Kangning Dong and Shihua Zhang. 2022. Deciphering spatial domains from spatially resolved transcriptomics with an adaptive graph attention auto-encoder. *Nature communications* 13, 1 (2022), 1–12.

[10] Chee-Huat Linus Eng, Michael Lawson, Qian Zhu, Ruben Dries, Noushin Koulena, Yodai Takei, Jina Yun, Christopher Cronin, Christoph Karp, Guo-Cheng Yuan, et al. 2019. Transcriptome-scale super-resolved imaging in tissues by RNA seqFISH+. *Nature* 568, 7751 (2019), 235–239.

[11] Gökcen Eraslan, Lukas M Simon, Maria Mircea, Nikola S Mueller, and Fabian J Theis. 2019. Single-cell RNA-seq denoising using a deep count autoencoder. *Nature communications* 10, 1 (2019), 1–14.

[12] Lovedeep Gondara and Ke Wang. 2017. Multiple imputation using deep denoising autoencoders. *arXiv preprint arXiv:1705.02737* 280 (2017).

[13] Lovedeep Gondara and Ke Wang. 2018. Mida: Multiple imputation using denoising autoencoders. In *Pacific-Asia conference on knowledge discovery and data mining*. Springer, 260–272.

[14] Wuming Gong, Il-Youp Kwak, Pruthvi Pota, Naoko Koyano-Nakagawa, and Daniel J Garry. 2018. DrImpute: imputing dropout events in single cell RNA sequencing data. *BMC bioinformatics* 19, 1 (2018), 1–10.

[15] Shanshan He, Ruchir Bhatt, Carl Brown, Emily A. Brown, Derek L. Buhr, Kan Chantranuvatana, Patrick Danaher, Dwayne Dunaway, Ryan G. Garrison, Gary Geiss, Mark T. Gregory, Margaret L. Hoang, Rustem Khafizov, Emily E. Killingbeck, Dae Kim, Tae Kyung Kim, Youngmi Kim, Andrew Klock, Mithra Korukonda, Alecksandr Kutchma, Zachary R. Lewis, Yan Liang, Jeffrey S. Nelson, Giang T. Ong, Evan P. Perillo, Joseph C. Phan, Tien Phan-Everson, Erin Piazza, Tushar Rane, Zachary Reitz, Michael Rhodes, Alyssa Rosenbloom, David Ross, Hiromi Sato, Aster W. Wardhani, Corey A. Williams-Wietzikoski, Lidan Wu, and Joseph M. Beechem. 2022. High-plex Multiomic Analysis in FFPE at Subcellular Level by Spatial Molecular Imaging. *bioRxiv* (2022). `https://doi.org/10.1101/2021.11.03.467020`

[16] Zhenyu Hou, Xiao Liu, Yukuo Cen, Yuxiao Dong, Hongxia Yang, Chunjie Wang, and Jie Tang. 2022. GraphMAE: Self-Supervised Masked Graph Autoencoders. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Washington DC, USA) *(KDD '22)*. Association for Computing Machinery, New York, NY, USA, 594–604. `https://doi.org/10.1145/3534678.3539321`

[17] Jian Hu, Xiangjie Li, Kyle Coleman, Amelia Schroeder, Nan Ma, David J Irwin, Edward B Lee, Russell T Shinohara, and Mingyao Li. 2021. SpaGCN: Integrating gene expression, spatial location and histology to identify spatial domains and spatially variable genes by graph convolutional network. *Nature methods* 18, 11 (2021), 1342–1351.

[18] Mo Huang, Jingshu Wang, Eduardo Torre, Hannah Dueck, Sydney Shaffer, Roberto Bonasio, John I Murray, Arjun Raj, Mingyao Li, and Nancy R Zhang. 2018. SAVER: gene expression recovery for single-cell RNA sequencing. *Nature methods* 15, 7 (2018), 539–542.

[19] Peter V. Kharchenko, Lev Silberstein, and David T. Scadden. 2014. Bayesian approach to single-cell differential expression analysis. *Nature Methods* 11, 7 (2014), 740—-742.

[20] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[21] Dongshunyi Li, Jun Ding, and Ziv Bar-Joseph. 2021. Identifying signaling genes in spatial single-cell expression data. *Bioinformatics* 37, 7 (2021), 968–975.

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

[23] Wei Vivian Li and Jingyi Jessica Li. 2018. An accurate and robust imputation method scImpute for single-cell RNA-seq data. *Nature communications* 9, 1 (2018), 1–9.

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

[25] Yao Ma and Jiliang Tang. 2021. *Deep learning on graphs*. Cambridge University Press.

[26] Ying Ma and Xiang Zhou. 2022. Spatially informed cell-type deconvolution for spatial transcriptomics. *Nature Biotechnology* (2022), 1–11.

[27] Christopher R Merritt, Giang T Ong, Sarah E Church, Kristi Barker, Patrick Danaher, Gary Geiss, Margaret Hoang, Jaemyeong Jung, Yan Liang, Jill McKay-Fleisch, et al. 2020. Multiplex digital spatial profiling of proteins and RNA in fixed tissue. *Nature Biotechnology* (05 2020). `https://doi.org/10.1038/s41587-020-0472-9`

[28] Ricardo Cardoso Pereira, Miriam Seoane Santos, Pedro Pereira Rodrigues, and Pedro Henriques Abreu. 2020. Reviewing autoencoders for missing data imputation: Technical trends, applications and outcomes. *Journal of Artificial Intelligence Research* 69 (2020), 1255–1285.

[29] Jiahua Rao, Xiang Zhou, Yutong Lu, Huiying Zhao, and Yuedong Yang. 2021. Imputing single-cell RNA-seq data by combining graph convolution and autoencoder neural networks. *Iscience* 24, 5 (2021), 102393.

[30] Jonathan Ronen and Altuna Akalin. 2018. netSmooth: Network-smoothing based imputation for single cell RNA-seq. *F1000Research* 7 (2018).

[31] Alexander Y Rudensky. 2011. Regulatory T cells and Foxp3. *Immunological reviews* 241, 1 (2011), 260–268.

[32] Carsen Stringer, Tim Wang, Michalis Michaelos, and Marius Pachitariu. 2021. Cellpose: a generalist algorithm for cellular segmentation. *Nature methods* 18, 1 (2021), 100–106.

[33] Patrik L. Ståhl, Fredrik Salmén, Sanja Vickovic, Anna Lundmark, José Fernández Navarro, Jens Magnusson, Stefania Giacomello, Michaela Asp, Jakub O. Westholm, Mikael Huss, Annelie Mollbrink, Sten Linnarsson, Simone Codeluppi, Åke Borg, Fredrik Pontén, Paul Igor Costea, Pelin Sahlén, Jan Mulder, Olaf Bergmann, Joakim Lundeberg, and Jonas Frisén. 2016. Visualization and analysis of gene expression in tissue sections by spatial transcriptomics. *Science* 353, 6294 (2016), 78–82. `https://doi.org/10.1126/science.aaf2403`

[34] Valentine Svensson, Kedar Nath Natarajan, Lam-Ha Ly, Ricardo J Miragaia, Charlotte Labalette, Iain C Macaulay, Ana Cvejic, and Sarah Teichmann. 2017. Power analysis of single-cell RNA-sequencing experiments. *Nature Methods* 14, 4 (2017), 381–387.

[35] Divyanshu Talwar, Aanchal Mongia, Debarka Sengupta, and Angshul Majumdar. 2018. AutoImpute: Autoencoder based imputation of single-cell RNA-seq data. *Scientific reports* 8, 1 (2018), 1–11.

[36] Fuchou Tang, Catalin Barbacioru, Yangzhou Wang, Ellen Nordman, Clarence Lee, Nanlan Xu, Xiaohui Wang, John Bodeau, Brian B Tuch, Asim Siddiqui, Kaiqin Lao, and M Azim Surani. 2009. mRNA-Seq whole-transcriptome analysis of a single cell. *Nature Methods* 6, 5 (2009), 377–382.

[37] Wenhao Tang, François Bertaux, Philipp Thomas, Claire Stefanelli, Malika Saint, Samuel Marguerat, and Vahid Shahrezaei. 2020. bayNorm: Bayesian gene expression recovery, imputation and normalization for single-cell RNA-sequencing data. *Bioinformatics* 36, 4 (2020), 1174–1181.

[38] 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. 2017. MAGIC: A diffusion-based imputation method reveals gene-gene interactions in single-cell RNA-sequencing data. *BioRxiv* (2017), 111591.

[39] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *stat* 1050 (2017), 20.

[40] Petar Veličković, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. 2018. Deep graph infomax. *arXiv preprint arXiv:1809.10341* (2018).

[41] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*. 1096–1103.

[42] Juexin Wang, Anjun Ma, Yuzhou Chang, Jianting Gong, Yuexu Jiang, Ren Qi, Cankun Wang, Hongjun Fu, Qin Ma, and Dong Xu. 2021. scGNN is a novel graph neural network framework for single-cell RNA-Seq analyses. *Nature communications* 12, 1 (2021), 1–11.

[43] Yunguan Wang, Bing Song, Shidan Wang, Mingyi Chen, Yang Xie, Guanghua Xiao, Li Wang, and Tao Wang. 2022. Sprod for de-noising spatially resolved transcriptomics data based on position and image information. *Nature methods* 19, 8 (2022), 950–958.

[44] Chenglong Xia, Jean Fan, George Emanuel, Junjie Hao, and Xiaowei Zhuang. 2019. Spatial transcriptome profiling by MERFISH reveals subcellular RNA compartmentalization and cell cycle-dependent gene expression. *Proceedings of the National Academy of Sciences* 116, 39 (2019), 19490–19499.