PAPER

Impeller: A Path-based Heterogeneous Graph Learning Method for Spatial Transcriptomic Data Imputation

Ziheng Duan,[1] Dylan Riffle,[1] Ren Li,[2] Junhao Liu,[1] Martin Renqiang Min[3] and Jing Zhang[1,*]

[1]Department of Computer Science, University of California, Irvine, 92697, CA, USA, [2]Mathematical, Computational, and Systems Biology, University of California, Irvine, 92697, CA, USA and [3]NEC Labs America, Princeton, 08540, NJ, USA

*Corresponding author. zhang.jing@uci.edu

FOR PUBLISHER ONLY Received on Date Month Year; revised on Date Month Year; accepted on Date Month Year

Abstract

Motivation: Recent advances in spatial transcriptomics allow spatially resolved gene expression measurements with cellular or even sub-cellular resolution, directly characterizing the complex spatiotemporal gene expression landscape and cell-to-cell interactions in their native microenvironments. Due to technology limitations, most spatial transcriptomic technologies still yield incomplete expression measurements with excessive missing values. Therefore, gene imputation is critical to filling in missing data, enhancing resolution, and improving overall interpretability. However, existing methods either require additional matched single-cell RNA-seq data, which is rarely available, or ignore spatial proximity or expression similarity information.

Results: To address these issues, we introduce Impeller, a path-based heterogeneous graph learning method for spatial transcriptomic data imputation. Impeller has two unique characteristics distinct from existing approaches. First, it builds a heterogeneous graph with two types of edges representing spatial proximity and expression similarity. Therefore, Impeller can simultaneously model smooth gene expression changes across spatial dimensions and capture similar gene expression signatures of faraway cells from the same type. Moreover, Impeller incorporates both short- and long-range cell-to-cell interactions (e.g., via paracrine and endocrine) by stacking multiple GNN layers. We use a learnable path operator in Impeller to avoid the over-smoothing issue of the traditional Laplacian matrices. Extensive experiments on diverse datasets from three popular platforms and two species demonstrate the superiority of Impeller over various state-of-the-art imputation methods.

Availability and Implementation: The code and preprocessed data used in this study are available at https://github.com/aicb-ZhangLabs/Impeller and https://zenodo.org/records/11212604.

Contact: zhang.jing@uci.edu.

Supplementary Information: Additional information is shown in the supplementary file.

Key words: Spatial Transcriptomic, Gene Imputation, Graph Learning

Introduction

The orchestration of cellular life hinges on the precise control of when and where genes are activated or silenced. Characterizing such spatiotemporal gene expression patterns is crucial for a better understanding of life, from development to disease to adaptation (Mantri et al. 2021). While single-cell RNA sequencing (scRNA-seq) is a revolutionary and widely-available technology that enables simultaneous gene expression profiling over thousands of cells, it usually needs to dissociate cells from their native tissue and thus loses the spatial context (Lähnemann et al. 2020). Recent advances in spatial transcriptomics (Ståhl et al. 2016) allow spatially resolved gene expression measurements at a single-cell or even sub-cellular resolution, providing unprecedented opportunities to characterize the complex landscape of spatiotemporal gene expression and understand the intricate interplay between cells in their native microenvironments (Strell et al. 2019). However, due to technical and biological limitations, most spatial transcriptomic profiling technologies still yield incomplete datasets with excessive missing gene expression values, hindering our biological interpretation of such valuable datasets (Choe et al. 2023). Therefore, gene imputation is a critical task to enrich spatial transcriptomics by filling in missing data, enhancing resolution, and improving the overall quality and interpretability of the datasets.

Several methods have been successfully developed for gene imputation in spatial transcriptomics, which can be broadly summarized into two categories - reference-based and reference-free approaches. Since scRNA-seq data usually offers a deeper dive into transcriptome profiling, reference-based methods integrated spatial transcriptomic data with matched scRNA-seq data from the same sample for accurate imputation. While promising, these referenced-based methods usually suffer from two limitations. First, most studies do not always have matched scRNA-seq data, especially those using valuable and rare samples. Second, even with matched data, there can be significant gene expression distribution shifts due to sequencing protocol differences (e.g., single nuclei RNA-seq vs. whole cell spatial transcriptomics) (Zeng et al. 2022).

Researchers also employed reference-free methods for direct gene expression imputation. For instance, traditional gene imputation methods designed for scRNA-seq data, such as scVI (Lopez et al. 2018), ALRA (Linderman et al. 2018), Magic (van Dijk et al. 2018) and scGNN (Wang et al. 2021), have been adapted for spatial transcriptomic data imputation. While effectively capturing cell-type-specific gene expression signatures, these methods completely ignored the rich spatial information, resulting in suboptimal results. Later, scientists emphasized the importance of spatial context for cell-to-cell interaction (CCI) in modulating expression changes in response to external stimuli (Armingol et al. 2021). Therefore, Graph Neural Network (GNN) based methods have been developed to mimic CCIs for imputation tasks with improved performance. However, different types of CCI involve distinct cell signaling mechanisms with varying interaction ranges. Existing GNN-based methods employed very shallow convolutional layers for computational convenience, successfully modeling short-range CCI (e.g., via autocrine and juxtacrine) but ignoring long-range interactions (e.g., via paracrine and endocrine). As a result, they cannot fully exploit the spatial information for gene expression imputation.

To address the abovementioned issues, we propose Impeller, a path-based heterogeneous graph learning method for accurate spatial transcriptomic data imputation. Impeller contains two unique components to exploit both transcriptomic and spatial information. First, it builds a heterogeneous graph with nodes representing cells and two types of edges describing expression similarity and spatial proximity. Therefore, the expression-based edges allow it to capture cell-type-specific expression signatures of faraway cells from the same type, and the proximity-based edges incorporate CCI effects in the spatial context. Second, Impeller models long-range CCI by stacking multiple GNN layers and uses a learnable path operator instead of the traditional Laplacian matrices to avoid the over-smoothing problem. Extensive experiments on diverse datasets from three popular platforms and two species demonstrate the superiority of Impeller over various state-of-the-art imputation methods.

Our main contributions are summarized below:

- We propose a graph neural network, Impeller, for reference-free spatial transcriptomic data imputation. Impeller incorporates cell-type-specific expression signatures and CCI via a heterogeneous graph with edges representing transcriptomic similarity and spatial proximity.
- Impeller stacks multiple GNN layers to include both short- and long-range cell-to-cell interactions in the spatial context. Moreover, it uses a learnable path-based operator to avoid over-smoothing.
- To the best of our knowledge, this is the first paper to combine cell-type-specific expression signatures with spatial short- and long-range CCI for gene expression imputation.
- We extensively evaluate Impeller alongside state-of-the-art competitive methods on datasets from three sequencing platforms and two species. The results demonstrate that Impeller outperforms all of the baselines.

Related Work

Imputation Methods Ignoring Spatial Information

Earlier spatial transcriptomic data imputation methods adapted the computational strategies originally developed for scRNA-seq data, overlooking the spatial coordinate information of each spot. For instance, eKNN (expression-based K nearest neighbor) and eSNN (expression-based Shared nearest neighbor) are methods implemented using the Seurat R-package that rely on gene expressions of nearest neighbors. MAGIC adopted data diffusion across similar cells to impute missing transcriptomic data. ALARA used low-rank approximation to distinguish genuine non-expression from technical dropouts, thus preserving true gene absence in samples. scVI used a deep variational autoencoder for gene imputation by assuming the read counts per gene follow a zero-inflated negative binomial distribution. However, these methods completely ignored the rich spatial information, resulting in sub-optimal performance.

Imputation Methods Utilizing Spatial Information

Later on, several methods were developed to exploit the spatial coordinate information to improve imputation accuracy. Since scRNA-seq data is usually sequenced deeper to provide more accurate expression measurements, several methods incorporated additional scRNA-seq data during the imputation process. For instance, gimVI used a low-rank approximation and includes scRNA reference (Lopez et al. 2019). Tangram mapped scRNA-seq data onto spatial transcriptomics data to facilitate imputation by fitting expression values on the shared genes (Biancalani et al. 2021). STLearn employed gene expression data, spatial distance, and tissue morphology data for imputing absent gene reads (Pham et al. 2020). However, additional scRNA-seq data is not always available and there can be large gene expression distribution shifts between these datasets due to differences in sequencing protocols (e.g., single-cell vs. single-nuclei), resulting in limited applications for reference-based methods.

On the other hand, several reference-free methods have been developed for more generalized settings. For example, the seKNN (spatial-expression-based K nearest neighbor) and seSNN (spatial-expression-based shared nearest neighbor) models (Butler et al. 2018; Hao et al. 2021; Satija et al. 2015; Stuart et al. 2019) incorporated cell-to-cell distance when defining the KNN for imputation tasks. Recently, STAGATE (Dong and Zhang 2022) is a graph attention auto-encoder framework that effectively imputes genes by integrating spatial data and cell type labels. Overall, these methods did not deeply integrate and exploit the full potential of combining expression and spatial data.

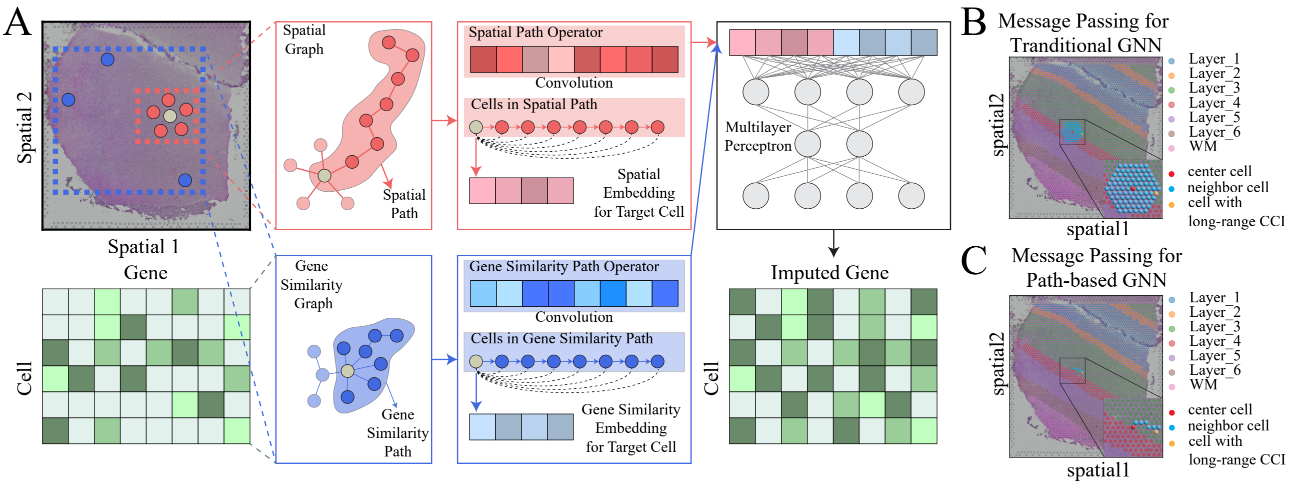

Fig. 1. The overview of Impeller. (A) Given observed matrix $\mathbf{X_{obs}} \in \mathbb{R}^{n\times m}$ of n cells and m genes, and cells' spatial coordinates $\mathbf{C} \in \mathbb{R}^{n\times 2}$, we build the spatial graph $\mathbf{G_s}$ and the gene similarity graph $\mathbf{G_g}$. The learned spatial and gene similarity path operators $\mathbf{op_s}$ and $\mathbf{op_g}$ are obtained through $\mathbf{path_s}$ and $\mathbf{path_g}$, respectively. Convoluting cell features with path operators yields spatial/gene similarity embeddings, which are concatenated and fed into a multilayer perceptron for final gene imputation. (B)-(C) Comparison of neighbor aggregation methods in GNNs. B: Traditional GNN stacks multiple layers to gather information from distant nodes. C: The path-based GNN, Impeller samples a path to the target node

Method

Problem Definition

Here, we aim to impute the excessive missing gene expression values in spatial transcriptomics data without matched reference scRNA-seq data. Formally, given a sparse cell-by-gene count matrix $\mathbf{X_{obs}} \in \mathbb{R}^{n\times m}$ which represents observations for n cells across m genes, and the spatial coordinates $\mathbf{C} \in \mathbb{R}^{n\times 2}$ of these cells, our goal is to impute the gene expression matrix $\hat{\mathbf{X}} \in \mathbb{R}^{n\times m}$. $\mathbf{X_{obs}}$ is derived from the ground truth matrix $\mathbf{X_{gt}} \in \mathbb{R}^{n\times m}$, which contains the observed non-zero entries pre-masking. To simulate real-world data conditions, 10% of the non-zero entries in $\mathbf{X_{gt}}$ are masked to form a test set and another 10% for validation, thus creating $\mathbf{X_{obs}}$. This matrix serves as the input for our imputation model. The major challenge is to generate $\hat{\mathbf{X}}$ that is as close as possible to the ground truth gene expression $\mathbf{X_{gt}}$, using both the observed gene expressions in $\mathbf{X_{obs}}$ and the spatial information in $\mathbf{C}$.

Heterogeneous Graph Construction

As shown in **Fig. 1**, we build our Impeller model based on two widely-accepted biological insights - 1) gene expression can be modulated by surrounding cells via CCI; 2) faraway cells of the same cell type may share stable gene expression signatures. Therefore, Impeller first builds a heterogeneous graph $\mathbf{G}$ to fully exploit both spatial and cell-type information, with nodes and edges representing cells and their relationships.

Specifically, $\mathbf{G}$ contains two complementary graphs: a spatial graph ($\mathbf{G_s}$) and a gene similarity graph ($\mathbf{G_g}$). Edges in $\mathbf{G_s}$ represent the cell's spatial proximity to model CCI, while edges in $\mathbf{G_g}$ denote the cell's transcriptomic similarity to capture the cell-type-specific expression signatures.

Spatial Graph Construction

The spatial graph $\mathbf{G_s}(\mathbf{V_s}, \mathbf{E_s})$ is created based on the spatial distance between cells, with nodes $\mathbf{V_s}$ representing the cells and edges in $\mathbf{E_s}$ connecting nearby cells. Specifically, an edge $e_{s,\{ij\}}$ in $\mathbf{G_s}$ is established between $v_i, v_j \in \mathbf{V_s}$ if and only if their Euclidean distance $d_{i,j}$ is less than a predefined threshold d_{thr}, which can be represented as:

$$e_{s,\{ij\}} = \text{if } ||\mathbf{C_i} - \mathbf{C_j}||_2 \leq d_{\text{thr}} \text{ else } 0, \quad (1)$$

where $\mathbf{C_i} = [C_{i,0}, C_{i,1}]$ and $\mathbf{C_j} = [C_{j,0}, C_{j,1}]$ are two dimensional spatial coordinates of cell i and j, respectively.

Gene Similarity Graph Construction

Impeller also builds a gene expression similarity graph $\mathbf{G_g}$ similar to that in scRNA-seq analysis. Specifically, we first extract the highly variable genes (default 3100). Then, for each target cell, we select its top K most similar cells. Mathematically,

$$e_{g,\{ij\}} = 1 \text{ if } j \in \mathbf{K_g}(\mathbf{X_i^h}) \text{ else } 0, \quad (2)$$

where $\mathbf{X_i^h}$ is the expression vector of highly variable genes in cell i, $\mathbf{K_g}(\mathbf{X_i^h})$ returns the top k_g cells most similar to cell i (e.g., using the Euclidean distance as the similarity metric), and $e_{g,\{ij\}}$ is the edge between cells i and j in $\mathbf{G_g}$.

GNN Model on Heterogeneous Graph

With the heterogeneous graph built, Impeller uses a path-based heterogeneous GNN to synthesize the impacts of spatial CCI ($\mathbf{G_s}$) and cell-type-specific expression signatures ($\mathbf{G_g}$) for the imputation task. We introduce the problem of traditional GNN, our learnable path operator, and the overall architecture of Impeller as follows.

Problem of Traditional GNN

We aim to impute the missing gene expression values in spatial transcriptomics data by incorporating its physical and transcriptional neighbors via a heterogeneous graph. By treating expression profiles as initial cell embeddings ($\mathbf{f^{(0)}} = \mathbf{X_{obs}}$), the l-th ($l \in \{1, 2, ..., L-1\}$) GNN layer follows a message passing form (Duan et al. 2024, 2023; Duan et al., 2022a,b,c; Wang et al. 2022, 2019; Xu et al. 2020, 2021) to generate cell i's embedding in layer l as follows:

$$\mathbf{f_i^{(l)}} = \gamma_\Theta(\mathbf{f_i^{(l-1)}} \bigoplus_{j \in \boldsymbol{\mathcal{N}}_s(i)} \phi_\Theta(\mathbf{f_i^{(l-1)}}, \mathbf{f_j^{(l-1)}}, e_{s,\{ij\}}) \bigoplus_{j \in \boldsymbol{\mathcal{N}}_g(i)} \psi_\Theta(\mathbf{f_i^{(l-1)}}, \mathbf{f_j^{(l-1)}}, e_{g,\{ij\}})), \qquad (3)$$

where $\mathbf{f_i^{(l)}} \in \mathbb{R}^{d_{emb}^{(l)}}$ is the embedding of cell i at l-th layer, $d_{emb}^{(l)}$ is the embedding dimension at l-th layer, and $\boldsymbol{\mathcal{N}_s}(i)$ and $\boldsymbol{\mathcal{N}_g}(i)$ are neighboring cell i in $\boldsymbol{G_s}$ and $\boldsymbol{G_g}$. $\bigoplus$ denotes a differentiable, permutation invariant function, e.g., sum, mean, and γ_Θ, ϕ_Θ, and ψ_Θ denote differentiable functions such as MLPs. After L layers, we obtain the imputed gene expressions, denoted as $\hat{\mathbf{X}} = \mathbf{f^{(L)}} \in \mathbb{R}^{n \times m}$.

In order to capture long-range CCI interaction, we have to include relatively far away cells by stacking multiple GNN layers via a larger L. Traditional Laplacian matrices-based GNN suffers from over-smoothing, resulting in deteriorated performance as L increases (Eliasof et al. 2022). Therefore, we introduce a learnable path operator to overcome this issue and better capture the long-range CCI.

Learnable Path Operator

We first define path $\mathbf{P_s} = (s_1, s_2, ..., s_{k_s})$ on $\mathbf{G_s}$ of length k_s and path $\mathbf{P_g} = (g_1, g_2, ..., g_{k_g})$ on $\mathbf{G_g}$ of length k_g, where s_i and g_i are node (cell) indexes. Node embeddings at l-th layer are denoted by $\mathbf{f_{s_i}^{(l)}} \in \mathbb{R}^{d_{emb}^{(l)}}$ and $\mathbf{f_{g_i}^{(l)}} \in \mathbb{R}^{d_{emb}^{(l)}}$. Then, $\mathbf{op_s^{(l)}} \in \mathbb{R}^{k_s \times d_{emb}^{(l)}}$ and $\mathbf{op_g^{(l)}} \in \mathbb{R}^{k_g \times d_{emb}^{(l)}}$ are two learnable path operators which allow us to convolve node embeddings along paths:

$$\mathbf{op_s^{(l)}}(\mathbf{P_s}) * \mathbf{f^{(l)}} = \sum_{i=1}^{k_s} \mathbf{op_{s,i}^{(l)}} * \mathbf{f_{s_i}^{(l)}} = \sum_{i=1}^{k_s} \sum_{j=1}^{d_{emb}^{(l)}} op_{s,i}^{(l)}[j] \cdot f_{s_i}^{(l)}[j],$$

$$\mathbf{op_g^{(l)}}(\mathbf{P_g}) * \mathbf{f^{(l)}} = \sum_{i=1}^{k_g} \mathbf{op_{g,i}^{(l)}} * \mathbf{f_{g_i}^{(l)}} = \sum_{i=1}^{k_g} \sum_{j=1}^{d_{emb}^{(l)}} op_{g,i}^{(l)}[j] \cdot f_{g_i}^{(l)}[j], \qquad (4)$$

where $'*'$ denotes the convolution operation, and $'\cdot'$ symbol is the multiplication operation between two scalars. Here $op_{s,i}^{(l)}[j]$, $op_{g,i}^{(l)}[j]$, $f_{s_i}^{(l)}[j]$ and $f_{g_i}^{(l)}[j]$ represent the j-th scalars of the $d_{emb}^{(l)}$-dimensional vector $\mathbf{op_{s,i}^{(l)}}$, $\mathbf{op_{g,i}^{(l)}}$, $\mathbf{f_{s_i}^{(l)}}$ and $\mathbf{f_{g_i}^{(l)}}$, respectively. Starting from each node, we generate multiple paths on $\mathbf{G_s}$ and $\mathbf{G_g}$ and aggregate results for a more expressive representation:

$$\mathbf{op_s^{(l)}}(\mathcal{P}_\mathbf{s}) * \mathbf{f^{(l)}} = \frac{1}{T_s} \sum_{\mathbf{P_s} \in \mathcal{P}_\mathbf{s}} \mathbf{op_s^{(l)}}(\mathbf{P_s}) * \mathbf{f^{(l)}},$$

$$\mathbf{op_g^{(l)}}(\mathcal{P}_\mathbf{g}) * \mathbf{f^{(l)}} = \frac{1}{T_g} \sum_{\mathbf{P_g} \in \mathcal{P}_\mathbf{g}} \mathbf{op_g^{(l)}}(\mathbf{P_g}) * \mathbf{f^{(l)}}, \qquad (5)$$

where $\mathcal{P}_\mathbf{s}$ and $\mathcal{P}_\mathbf{g}$ are sets of paths sampled from the $\mathbf{G_s}$ and $\mathbf{G_g}$, each containing T_s and T_g paths. Each path $\mathbf{P_s} \in \mathcal{P}_\mathbf{s}$ and $\mathbf{P_g} \in \mathcal{P}_\mathbf{g}$ are separately convolved using $\mathbf{op_s^{(l)}}$ or $\mathbf{op_g^{(l)}}$, and the results are averaged to acquire the node embeddings.

The Overall Architecture of Impeller

After convolving both spatial and gene similarity paths, we concatenate their embeddings to form the overall node embeddings, as in

$$\mathbf{f^{(l+1)}} = \sigma\Big(\mathbf{W_1^{(l)}}[\mathbf{op_s^{(l)}}(\mathcal{P}_\mathbf{s}) * \mathbf{f^{(l)}}, \mathbf{op_g^{(l)}}(\mathcal{P}_\mathbf{g}) * \mathbf{f^{(l)}}]\Big), \qquad (6)$$

Table 1. Summary of datasets.

Platform	Organism	Sample ID	Raw Matrix (Cell, Gene)	Raw Density	Filter Matrix (Cell, Gene)	Filter Density	# Imputed Entries
10xVisium	Human Dorsolateral Prefrontal Cortex (DLPFC)	151507	4226, 33538	0.042	4117, 4028	0.261	437240
		151508	4384, 33538	0.036	4148, 3342	0.258	358184
		151509	4789, 33538	0.043	4700, 4188	0.258	508186
		151510	4643, 33538	0.041	4547, 3908	0.259	461112
		151669	3661, 33538	0.054	3617, 5246	0.277	525930
		151670	3498, 33538	0.050	3433, 4909	0.272	457770
		151671	4110, 33538	0.055	3988, 5539	0.278	615111
		151672	4015, 33538	0.052	3809, 5273	0.279	561166
		151673	3639, 33538	0.066	3628, 6538	0.286	677473
		151674	3673, 33538	0.080	3668, 7796	0.305	871032
		151675	3592, 33538	0.054	3565, 5454	0.267	518515
		151676	3460, 33538	0.058	3449, 5784	0.274	545920
Stereoseq	Mouse	/	19109, 14376	0.024	4036, 1581	0.193	123444
SlideseqV2	Mouse	/	20139, 11750	0.031	5161, 2611	0.217	292418

where $\sigma(\cdot)$ denotes the ReLU activation function, $\mathbf{W^{(l)}} \in \mathbb{R}^{d_{emb}^{(l+1)} \times 2 * d_{emb}^{(l)}}$ is the learnable weight matrix, $d_{emb}^{(l)}$ is the embedding dimension at l-th layer, and $[\cdot, \cdot]$ denotes concatenation operation. Then, Impeller tries to minimize the Mean Squared Error (MSE) between $\hat{\mathbf{X}}$ and $\mathbf{X_{gt}}$:

$$\mathcal{L} = \frac{\sum_{i=1}^{n} \sum_{j=1}^{m} \mathbb{1}[X_{gt,(i,j)} \neq 0](\hat{X}_{i,j} - X_{gt,(i,j)})^2}{\sum_{i=1}^{n} \sum_{j=1}^{m} \mathbb{1}[X_{gt,(i,j)} \neq 0]}, \qquad (7)$$

where $\mathbb{1}[\cdot]$ is an indicator function that equals 1 if the condition inside brackets is met ($X_{gt,(i,j)} \neq 0$), and 0 otherwise. The loss is computed only over non-zero entries of $\mathbf{X_{gt}}$.

Computational Complexity Analysis

k-hop Complexity Analysis

Traditional GNNs need to gather information from k-hop neighbor nodes after stacking of k layers. Given the complexity of each layer as $O(n \times d_t)$, where n is the number of nodes and d_t is the average node degree, the overall complexity becomes $O(n \times d_t \times k)$. In contrast, Impeller can directly access neighbors up to k-hop distance via a single layer by setting $k_s = k_g = k$. The computational complexity per layer for Impeller is $O(n \times (T_s \times k_s + T_g \times k_g))$, with T_s and T_g representing the number of paths in $\mathbf{G_s}$ and $\mathbf{G_g}$, k_s and k_g denoting path lengths. As a result, when $T_s < d_t$ (a condition satisfied in our task), Impeller offers superior computational efficiency.

The Number of Parameters

Traditional GNNs have $O(d_{emb}^{(l)} \times d_{emb}^{(l+1)})$ parameters per layer while the path operator of Impeller adds $(k_s + k_g) * d_{emb}^{(l)}$ parameters. Since $k_s + k_g$ is typically much smaller than $d_{emb}^{(l+1)}$, Impeller's number of parameters remains on par with traditional GNNs.

Experiments

Detailed Experimental Setup

Data Sources and Preprocessing

In our study, we tested Impeller using diverse datasets from three popular sequencing platforms and two organisms. Specifically, we included 10X Visium datasets from human dorsolateral prefrontal cortex (DLPFC), (Maynard et al. 2021), Steroseq datasets from mouse olfactory bulb (Chen et al. 2021), and Slide-seqV2 from mouse olfactory bulb (Stickels et al. 2021) in our analyses. Detailed attributes of these datasets are summarized in **Table 1** (filter details and visualizations see the appendix). After standard pre-processing and normalization procedures, we downsampled the data according to scGNN, where 10% of non-zero entries in the dataset were used as a

Table 2. Gene imputation benchmark. The best results are bolded. Results marked ’NA’ for stLearn indicate unavailable HE stained images required by the method.

Metric	Method		Platform & Dataset							
			10xVisium						**Stereoseq**	**SlideseqV2**
			DLPFC						Mouse	Mouse
			151507	151508	151509	151510	151669	151670	/	/
L1 Distance	wo	scVI	0.794±0.004	0.838±0.006	0.800±0.002	0.670±0.003	0.810±0.003	0.696±0.005	1.442±0.005	1.127±0.006
		ALRA	0.499±0.003	0.512±0.001	0.490±0.001	0.496±0.001	0.467±0.002	0.472±0.002	0.406±0.013	0.649±0.066
		eKNN	0.274±0.001	0.281±0.001	0.275±0.000	0.275±0.001	0.269±0.000	0.272±0.000	0.205±0.001	0.294±0.001
		eSNN	1.254±0.001	1.373±0.001	1.266±0.001	1.294±0.000	1.017±0.001	1.071±0.001	2.802±0.002	2.071±0.002
		Magic	0.779±0.001	0.825±0.001	0.787±0.000	0.664±0.001	0.795±0.001	0.692±0.000	1.324±0.001	1.080±0.000
		scGNN	0.583±0.011	0.665±0.085	0.589±0.011	0.584±0.004	0.550±0.006	0.532±0.009	0.819±0.240	0.664±0.018
	w	gimVI	0.838±0.003	0.890±0.003	0.835±0.001	0.737±0.002	0.863±0.003	0.765±0.001	1.325±0.001	1.153±0.002
		seKNN	0.306±0.000	0.309±0.001	0.307±0.000	0.307±0.000	0.281±0.000	0.289±0.000	0.263±0.001	0.876±0.001
		seSNN	1.254±0.001	1.371±0.001	1.266±0.000	1.294±0.000	1.017±0.001	1.072±0.001	2.775±0.002	1.998±0.001
		Tangram	1.691±0.001	1.811±0.001	1.689±0.000	1.420±0.000	1.728±0.001	1.474±0.000	2.899±0.001	2.185±0.000
		STLearn	1.333±0.001	1.423±0.001	1.332±0.001	1.148±0.001	1.369±0.002	1.206±0.001	NA	NA
		STAGATE	0.297±0.001	0.300±0.002	0.295±0.005	0.294±0.004	0.274±0.005	0.278±0.002	0.289±0.006	0.502±0.007
		Impeller	**0.248±0.001**	**0.252±0.003**	**0.247±0.003**	**0.254±0.003**	**0.242±0.004**	**0.237±0.001**	**0.190±0.005**	**0.292±0.005**
Cosine Similarity	wo	scVI	0.907±0.001	0.913±0.001	0.906±0.001	0.903±0.001	0.909±0.001	0.904±0.001	0.941±0.001	0.919±0.002
		ALRA	0.948±0.002	0.952±0.002	0.952±0.001	0.952±0.001	0.938±0.006	0.944±0.003	0.980±0.002	0.927±0.018
		eKNN	0.983±0.000	0.984±0.000	0.983±0.000	0.984±0.000	0.979±0.000	0.979±0.000	0.993±0.000	0.989±0.000
		eSNN	0.842±0.000	0.841±0.000	0.839±0.000	0.840±0.000	0.846±0.000	0.843±0.000	0.777±0.001	0.838±0.000
		Magic	0.915±0.000	0.920±0.000	0.914±0.000	0.909±0.000	0.916±0.000	0.910±0.000	0.968±0.002	0.936±0.000
		scGNN	0.933±0.004	0.927±0.016	0.932±0.002	0.936±0.000	0.917±0.002	0.929±0.002	0.948±0.035	0.953±0.002
	w	gimVI	0.957±0.000	0.965±0.001	0.955±0.001	0.947±0.001	0.962±0.001	0.948±0.002	0.964±0.000	0.936±0.001
		seKNN	0.982±0.000	0.985±0.000	0.982±0.000	0.983±0.000	0.979±0.000	0.980±0.000	0.995±0.000	0.982±0.000
		seSNN	0.843±0.000	0.841±0.000	0.840±0.000	0.841±0.000	0.851±0.000	0.847±0.000	0.768±0.000	0.817±0.000
		Tangram	0.713±0.001	0.725±0.001	0.717±0.001	0.716±0.001	0.717±0.001	0.715±0.000	0.772±0.001	0.763±0.001
		STLearn	0.718±0.000	0.718±0.000	0.715±0.001	0.724±0.000	0.715±0.001	0.717±0.000	NA	NA
		STAGATE	0.983±0.000	0.985±0.000	0.983±0.001	0.984±0.001	0.980±0.001	0.980±0.000	0.990±0.000	0.961±0.000
		Impeller	**0.987±0.000**	**0.988±0.000**	**0.987±0.000**	**0.987±0.000**	**0.983±0.001**	**0.985±0.000**	**0.997±0.000**	**0.990±0.000**
RMSE	wo	scVI	0.940±0.005	0.993±0.006	0.949±0.003	0.803±0.003	0.959±0.003	0.834±0.005	1.628±0.005	1.307±0.007
		ALRA	0.784±0.003	0.810±0.005	0.766±0.001	0.777±0.001	0.735±0.004	0.743±0.003	0.723±0.036	1.061±0.107
		eKNN	0.380±0.001	0.395±0.002	0.382±0.001	0.384±0.001	0.368±0.001	0.374±0.001	0.402±0.008	0.416±0.003
		eSNN	1.378±0.001	1.503±0.000	1.393±0.000	1.419±0.001	1.143±0.002	1.199±0.001	2.778±0.001	2.177±0.001
		Magic	0.917±0.001	0.972±0.001	0.929±0.000	0.792±0.000	0.936±0.001	0.824±0.000	1.453±0.001	1.238±0.001
		scGNN	0.755±0.016	0.850±0.096	0.762±0.011	0.755±0.002	0.717±0.007	0.686±0.010	1.051±0.307	0.842±0.021
	w	gimVI	0.955±0.002	1.002±0.001	0.957±0.001	0.858±0.001	0.970±0.002	0.890±0.002	1.448±0.001	1.217±0.004
		seKNN	0.392±0.001	0.395±0.000	0.392±0.000	0.392±0.000	0.361±0.001	0.370±0.000	0.361±0.001	0.523±0.000
		seSNN	1.354±0.001	1.474±0.000	1.370±0.000	1.395±0.001	1.119±0.001	1.175±0.001	2.770±0.002	2.087±0.001
		Tangram	1.768±0.001	1.889±0.001	1.767±0.000	1.503±0.000	1.804±0.001	1.557±0.000	2.970±0.001	2.284±0.000
		STLearn	1.516±0.001	1.629±0.001	1.521±0.001	1.300±0.001	1.556±0.002	1.362±0.001	NA	NA
		STAGATE	0.384±0.002	0.393±0.002	0.379±0.007	0.380±0.007	0.357±0.007	0.365±0.004	0.485±0.008	0.765±0.005
		Impeller	**0.337±0.001**	**0.341±0.000**	**0.336±0.000**	**0.340±0.004**	**0.327±0.007**	**0.323±0.000**	**0.277±0.002**	**0.391±0.006**

test set, and another 10% of non-zero entries were reserved for validation. For a fair comparison, we repeat ten times with different mask configurations.

Baseline Methods for Benchmarking

We conducted a comparative study utilizing 12 state-of-the-art methods, including reference-free and reference-based methods that originally require additional scRNA-seq data. However, in our analysis, we did not use any additional scRNA-seq data for a fair comparison.

First, we included methods directly adapted from scRNA-seq data imputation and completely ignored the rich spatial information, including a deep generative model scVI, a low-rank approximation model ALRA, nearest neighbors-based models eKNN and eSNN, a diffusion-based model MAGIC, and a GNN-based model scGNN. Furthermore, we employed several imputation methods specifically designed for spatial transcriptomic data, such as seKNN (spatial-expression-based K nearest neighbor), and seSNN (spatial-expression-based shared nearest neighbor). gimVI and Tangram need additional scRNA-seq from matched samples, so we used a reference-free implementation available through their website for a fair comparison. Lastly, we included STAGATE a graph attention auto-encoder framework by amalgamating spatial data and gene expression profiles. We use default parameters in most baseline methods (details see the appendix).

Evaluation Metrics

We first define a test mask $\mathbf{M} \in \mathbb{R}^{n\times m}$ where the entries to be imputed are marked as 1 and the others as 0. Then we extract the relevant entries from both the imputed matrix $\hat{\mathbf{X}}$ and the ground truth matrix $\mathbf{X_{gt}}$ to form two vectors: $\hat{\mathbf{x}}$ (from $\hat{\mathbf{X}}$) and $\mathbf{x_{gt}}$ (from $\mathbf{X_{gt}}$), each of length N, where N is the total number of entries to be imputed. Following scGNN settings, we use L1 Distance, Cosine Similarity, and Root-Mean-Square Error (RMSE) to compare imputed gene expressions $\hat{\mathbf{x}}$ with the

Table 3. Performance of different receptive fields (RMSE).

Receptive Field	GCN	GraphSAGE	GAT	GraphTransformer	Impeller
2	0.339 ± 0.000	0.352 ± 0.008	0.360 ± 0.005	1.058 ± 0.463	**0.310 ± 0.016**
4	0.348 ± 0.001	0.362 ± 0.013	0.372 ± 0.022	0.424 ± 0.031	**0.286 ± 0.009**
8	0.386 ± 0.012	0.496 ± 0.033	0.454 ± 0.024	0.351 ± 0.002	**0.279 ± 0.001**
16	0.403 ± 0.001	0.617 ± 0.054	0.506 ± 0.061	0.435 ± 0.017	**0.286 ± 0.010**
32	0.418 ± 0.015	1.466 ± 0.031	1.458 ± 0.037	0.420 ± 0.000	**0.277 ± 0.002**
64	0.430 ± 0.002	1.615 ± 0.010	1.621 ± 0.004	0.420 ± 0.001	**0.302 ± 0.028**
128	0.429 ± 0.001	1.629 ± 0.003	1.614 ± 0.022	0.420 ± 0.001	**0.357 ± 0.001**

ground truth $\mathbf{x_{gt}}$. Mathematically:

$$\text{L1 Distance} = |\hat{\mathbf{x}} - \mathbf{x_{gt}}|, \tag{8}$$

$$\text{Cosine Similarity}(\hat{\mathbf{x}}, \mathbf{x_{gt}}) = \frac{\hat{\mathbf{x}}\mathbf{x_{gt}}^T}{||\hat{\mathbf{x}}|| * ||\mathbf{x_{gt}}||}, \tag{9}$$

$$\text{RMSE}(\hat{\mathbf{x}}, \mathbf{x_{gt}}) = \sqrt{\frac{\sum_{i=1}^{N} (\hat{\mathbf{x}}_i - \mathbf{x_{gt}}_i)^2}{N}}. \tag{10}$$

Experimental Results

Improved Imputation Accuracy

We benchmarked our performance against 12 leading methods by assessing imputation accuracy across 14 datasets. These datasets span three prominent sequencing platforms (10x Visium, Stereoseq, and Slideseq) and two species (human and mouse). **Table 2** summarizes the performance of Impellers and other baselines (for results of the other six samples of the DLPFC dataset, please see the appendix). For a fair comparison, we didn't include any additional scRNA-seq data to facilitate the imputation task. Overall, Impeller consistently outperforms others in all datasets using L1 distance, Cosine Similarity, and RMSE, indicating the effectiveness and robustness of our strategy.

Additionally, we found that most methods utilizing spatial information (w* group in **Table 2**) demonstrated higher imputation accuracy than those ignoring spatial information (wo* group in **Table 2**), validating the presence of rich information in the spatial context. Notably, Impeller surpasses even the best gene expression-only method, eKNN, with improvements of 11.32% on 10xVisium DLPFC, 31.09% on Stereoseq, and 6.01% on SlideseqV2 Mouse. Furthermore, compared to uniform averaging using KNN, GNN allows for more flexible neighbor information aggregation for better imputation accuracy, as reflected by the noticeably improved performance of Impeller and STAGATE.

Impact of Long-range CCI

To probe disparities between Impeller and traditional GNNs in capturing long-range cell dependencies, we examined several models—Impeller, GCN (Kipf and Welling 2016), GraphSAGE (Hamilton et al. 2017), GAT (Veličković et al. 2017), and GraphTransformer (Shi et al. 2020)—across varying receptive fields in the Stereoseq dataset.

In **Table 3**, GAT and GraphSAGE suffer from gradient vanishing/exploding issues as more layers are added to capture long-range CCI, resulting in quickly degraded performance. GCN works best initially, but its performance drops with more layers added. This could be because the number of neighbors grows fast as we increase the receptive field, leaving it difficult for the target cell to understand the influence of each neighbor. Furthermore, GraphTransformer starts with high errors at a receptive field of 2. It works best at a receptive field of 8, but the error goes up again at 32. This increase in error is similar to the problem of GCN, as all cells start to look too similar to make useful representations. On the other hand, Impeller effectively tackles these challenges by the path operator, as reflected by the consistently improved results until the receptive field of 32. As the receptive field continues to grow, Impeller's performance slightly declines, likely because distant information becomes less relevant for the target cell's gene imputation. An additional perturbation study, demonstrating the effectiveness of Impeller in capturing CCI, is shown in the appendix.

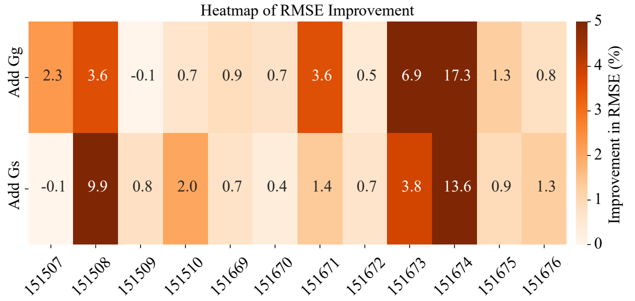

Fig. 2. RMSE improvement by adding different graph modalities.

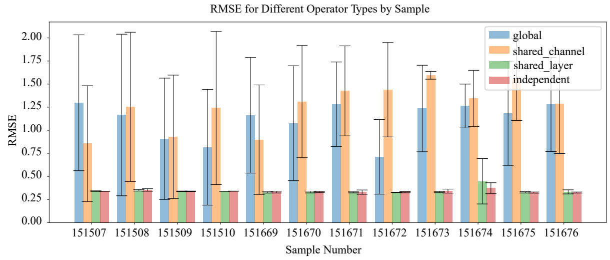

Fig. 3. RMSE w.r.t. different path operators.

Advantage of Heterogeneous Graph

In our study, we explored the influence of graph modalities on imputation accuracy by assessing three key variants: var_s, employing solely the spatial graph; var_g, utilizing only the gene similarity graph; var_h, integrating both graphs. We then calculated the performance improvement from adding $\mathbf{G_g}$ by comparing var_h with var_s, and the improvement from adding $\mathbf{G_s}$ by comparing var_h with var_g. As shown in **Fig. 2**, the majority of the cases (22 out of 24) exhibit positive improvements. Specifically, in the DLPFC sample 151674, the inclusion of the gene similarity graph yields a 17.3% improvement, and the 13.6% enhancement is achieved by adding the spatial graph alone. Similarly, in sample 151508, the gene similarity graph and the spatial graph contribute to improvements of 3.6% and 9.9%, respectively. These results underscore the efficacy of our approach, particularly in scenarios where the complex interaction between spatial and gene expression data is pivotal for enhancing gene imputation accuracy.

Ablation Study

We conducted an ablation study to evaluate the performance of four primary path operator variants of Impeller: $\mathbf{op_{glo}}$, where all Impeller layers and channels (each channel representing one dimension of $\mathbf{f}^{(1)}$) share one path operator; $\mathbf{op_{cha}}$, where channels share an operator but layers have distinct ones; $\mathbf{op_{lay}}$, where all layers share one, but channels have

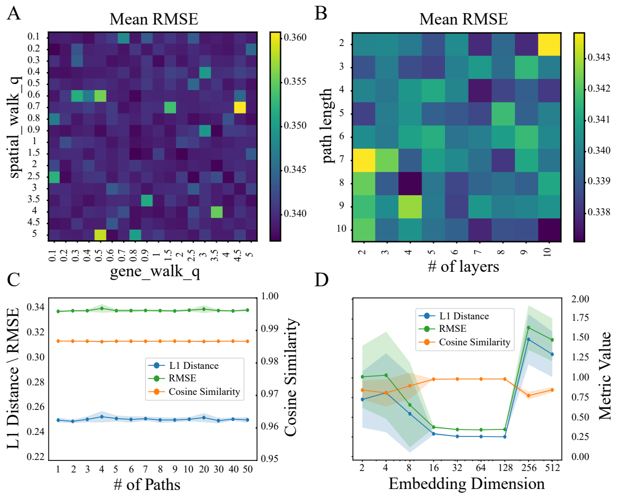

Fig. 4. Parameter analysis. (A) The mean RMSE w.r.t. different q_s and q_g for generating random walks in $\mathbf{G_s}$ and $\mathbf{G_g}$. (B) The mean RMSE w.r.t. different path lengths k_s and k_g, and the number of Impeller layers L. (C) The L1 Distance, Cosine Similarity, and RMSE w.r.t. different number of paths T_s and T_g. (D) The L1 Distance, Cosine Similarity and RMSE w.r.t. different embedding dimensions $d_{emb}^{(l)}$.

Table 4. Running time summary of graph-based models.

Model	# of Parameters	Path (ms)	Training (ms)	Inference (ms)	RMSE
GCN	519676	–	21.73±10.44	21.04±10.84	0.37±0.02
GAT	523768	–	23.84±13.25	24.99±11.41	0.36±0.02
GraphSAGE	1035260	–	18.77±11.06	16.97±12.16	0.38±0.01
Transformer	2078704	–	33.94±16.16	38.13±8.71	0.36±0.01
Impeller	538108	1.61±0.60	**6.35±0.30**	**8.43±0.25**	**0.34±0.00**

individual operators; and $\mathbf{op_{ind}}$ where every layer and channel possesses an independent path operator. As depicted in **Fig. 3**, both $\mathbf{op_{glo}}$ and $\mathbf{op_{cha}}$ performed poorly on the DLPFC dataset, indicating the importance of distinct operators for each channel. Notably, $\mathbf{op_{lay}}$ and $\mathbf{op_{ind}}$ showed comparable results, suggesting that layer-specific operators might be optional, depending on the specific application. Another ablation study regarding different path construction and graph construction methods is shown in the appendix.

Parameter Analysis

To investigate the influence of Impeller's various hyper-parameters, we conducted extensive experiments using the DLPFC dataset (Sample ID: 151507) and reported the mean and standard deviation of the imputation accuracy over ten repetitions.

First, we studied the impact of q_s and q_g on the RMSE of a random walk on $\mathbf{G_s}$ and $\mathbf{G_g}$ following the Node2Vec mechanism. Higher values of q (i.e., q_s and q_g) encourage the walk to sample more distant nodes, enhancing the exploration of the global graph structure, while lower values bias the walk towards neighboring nodes, facilitating local exploration. As shown in **Fig. 4A**, Impeller exhibits strong robustness with RMSE from 0.33 to 0.36 when q_s and q_g varied from 0.1 to 5. However, higher values of q_s and q_g tend to induce larger errors. For generality, we selected 1 as the default value for q_s and q_g.

We investigated the impact of random walk length (k_s and k_g) and layer number (L), shown in **Fig. 4B**. A path length of 2 with 10 layers results in maximum errors, reducing our model to a standard ten-layer GCN. This is because, at this path length, the model focuses on immediate neighbors, akin to how traditional GCNs operate. Such a setup, while deep, limits neighborhood exploration and increases over-smoothing risk. Conversely, a path length of 8 with 4 layers allows for capturing broader interactions (up to 28 hops), balancing extended reach and computational efficiency, thus avoiding over-smoothing and optimizing long-range CCI capture.

Next, we examined the impact of the number of random walks (T_s and T_g). As shown in **Fig. 4C**, T_s and T_g appeared to have a minimal effect on results, due to the robustness of Impeller which resamples at each epoch during training. We chose 8 as the default number of random walks.

Lastly, we evaluated how the embedding dimension $d_{emb}^{(l)}$ affects Impeller's performance. As shown in **Fig. 4D**, smaller $d_{emb}^{(l)}$ (such as 2, 4, 8) leads to limited expressive power and larger imputation errors. As $d_{emb}^{(l)}$ increases to 16, 32, 64, or 128, Impeller's expressive power improves and operations converge well in each run. Due to our early stopping criterion, we cease training if the validation RMSE doesn't improve for 50 consecutive epochs. When $d_{emb}^{(l)}$ was set to 256 or 512, it's hard for Impeller to converge quickly at these dimensions. To strike a balance between complexity and representational power, we opted for $d_{emb}^{(l)}$ of 64.

In summary, these comprehensive parameter analyses reveal that Impeller is robust across a wide range of parameter settings, while still providing tunable options for balancing computational efficiency and prediction accuracy. These results further substantiate the effectiveness and practicality of our proposed model for gene imputation tasks.

Neighbor Visualization

To better understand the differences between traditional GNNs and our path-based GNN, Impeller, we turned to a visual example (sample 151507 from the DLPFC dataset). **Fig. 1B** shows how the typical GNN gathers information from far-away neighbors. The center node (red sphere) stacks five GNN layers to gather information from distant nodes like the one shown in yellow. But this method sometimes pulls in extra information from different tissue layers that isn't needed. On the other hand, **Fig. 1C** shows our Impeller model. Instead of stacking GNN layers, Impeller samples a direct path from the center node to the target node. While using this direct path method, Impeller offers better gene imputation performance by capturing the relevant long-range CCI.

Running Time Analysis

As shown in **Table 4**, we conducted a comparative model parameter and runtime analysis with popular graph-based models (GCN, GAT, GraphSAGE, and Transformer) on the DLPFC dataset. As discussed in the 'The Number of Parameters' section, our model maintains a parameter count comparable to traditional GNNs, with the complexity per layer defined as $O(d_{emb}^{(l)} \times d_{emb}^{(l+1)})$. Specifically, our model introduces only a 3.5% increase in parameters for GCN and a 2.7% increase for GAT. In contrast, it achieves a 48.0% reduction in parameters for GraphSAGE and a 74.1% reduction for GraphTransformer (**Table 4**). Despite its additional path sampling step, Impeller remarkably outperformed the others in training and inference efficiency. This can be partially credited to leveraging the DGL library's optimized implementation for path sampling[1] and the inherently faster multiplication process used in path-based convolution compared to edge-wise information aggregation in traditional GNNs. Additionally, Impeller showed the lowest RMSE, indicating superior prediction accuracy. Hence, Impeller offers a balanced blend of efficiency and precision for spatial transcriptomic data imputation, outperforming other graph-based models.

Conclusion

In this study, we introduced Impeller, a path-based heterogeneous graph learning approach tailored for spatial transcriptomic data imputation. By constructing a heterogeneous graph capturing both spatial proximity and gene expression similarity, Impeller offers a refined representation of cellular landscapes. Further, its integration of multiple GNN layers, coupled with a learnable path operator, ensures comprehensive modeling of both short and long-range cellular interactions while effectively averting over-smoothing issues. Benchmark tests across diverse datasets spanning various platforms and species underscore Impeller's superior performance compared to state-of-the-art imputation methods. This work not only establishes Impeller's prowess in spatial transcriptomic imputation but also underscores its potential to model both short- and long-range cell-cell interactions.

Competing interests

No competing interest is declared.

Acknowledgments

The authors thank the anonymous reviewers for their valuable suggestions and the UCI OIT department for GPU resources.

Funding

This work was supported by the National Institutes of Health [R01HG012572, R01NS128523].

Data availability

The code and preprocessed data used in this study are available at https://github.com/aicb-ZhangLabs/Impeller and https://zenodo.org/records/11212604.

References

E. Armingol, A. Officer, O. Harismendy, and N. E. Lewis. Deciphering cell–cell interactions and communication from gene expression. *Nature Reviews Genetics*, 22(2):71–88, 2021.

T. Biancalani, G. Scalia, L. Buffoni, R. Avasthi, Z. Lu, A. Sanger, N. Tokcan, C. R. Vanderburg, Å. Segerstolpe, M. Zhang, I. Avraham-Davidi, S. Vickovic, M. Nitzan, S. Ma, A. Subramanian, M. Lipinski, J. Buenrostro, N. B. Brown, D. Fanelli, X. Zhuang, E. Z. Macosko, and A. Regev. Deep learning and alignment of spatially resolved single-cell transcriptomes with tangram. *Nature Methods*, 18 (11):1352–1362, Nov 2021. ISSN 1548-7105. doi: 10.1038/s41592-021-01264-7.

A. Butler, P. Hoffman, P. Smibert, E. Papalexi, and R. Satija. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nature Biotechnology*, 36:411–420, 2018. doi: 10.1038/nbt.4096.

A. Chen, S. Liao, M. Cheng, K. Ma, L. Wu, Y. Lai, J. Yang, W. Li, J. Xu, S. Hao, et al. Large field of view-spatially resolved transcriptomics at nanoscale resolution. *BioRxiv*, 2021, 2021.

K. Choe, U. Pak, Y. Pang, W. Hao, and X. Yang. Advances and challenges in spatial transcriptomics for developmental biology. *Biomolecules*, 13(1):156, 2023.

K. Dong and S. Zhang. Deciphering spatial domains from spatially resolved transcriptomics with an adaptive graph attention auto-encoder. *Nature communications*, 13(1): 1739, 2022.

Z. Duan, C. Lee, J. Zhang, et al. Exad-gnn: Explainable graph neural network for alzheimer's disease state prediction from single-cell data. *APSIPA Transactions on Signal and Information Processing*, 12(5).

Z. Duan, Y. Wang, W. Ye, Q. Fan, and X. Li. Connecting latent relationships over heterogeneous attributed network for recommendation. *Applied Intelligence*, 52(14):16214–16232, 2022a.

Z. Duan, H. Xu, Y. Huang, J. Feng, and Y. Wang. Multivariate time series forecasting with transfer entropy graph. *Tsinghua Science and Technology*, 28(1):141–149, 2022b.

Z. Duan, H. Xu, Y. Wang, Y. Huang, A. Ren, Z. Xu, Y. Sun, and W. Wang. Multivariate time-series classification with hierarchical variational graph pooling. *Neural Networks*, 154:481–490, 2022c.

Z. Duan, Y. Dai, A. Hwang, C. Lee, K. Xie, C. Xiao, M. Xu, M. J. Girgenti, and J. Zhang. iherd: an i ntegrative hie rarchical graph r epresentation learning framework to quantify network changes and prioritize risk genes in d isease. *PLOS Computational Biology*, 19(9):e1011444, 2023.

Z. Duan, S. Xu, S. Sai Srinivasan, A. Hwang, C. Y. Lee, F. Yue, M. Gerstein, Y. Luan, M. Girgenti, and J. Zhang. scencore: leveraging single-cell epigenetic data to predict chromatin conformation using graph embedding. *Briefings in Bioinformatics*, 25(2):bbae096, 2024.

M. Eliasof, E. Haber, and E. Treister. pathgcn: Learning general graph spatial operators from paths. In *International Conference on Machine Learning*, pages 5878–5891. PMLR, 2022.

W. Hamilton, Z. Ying, and J. Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

Y. Hao, S. Hao, E. Andersen-Nissen, W. M. M. III, S. Zheng, A. Butler, M. J. Lee, A. J. Wilk, C. Darby, M. Zagar, P. Hoffman, M. Stoeckius, E. Papalexi, E. P. Mimitou, J. Jain, A. Srivastava, T. Stuart, L. B. Fleming, B. Yeung, A. J. Rogers, J. M. McElrath, C. A. Blish, R. Gottardo, P. Smibert, and R. Satija. Integrated analysis of multimodal single-cell data. *Cell*, 2021. doi: 10.1016/j.cell.2021.04.048.

T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

D. Lähnemann, J. Köster, E. Szczurek, D. J. McCarthy, S. C. Hicks, M. D. Robinson, C. A. Vallejos, K. R. Campbell, N. Beerenwinkel, A. Mahfouz, et al. Eleven grand challenges in single-cell data science. *Genome biology*, 21(1):1–35, 2020.

[1] https://docs.dgl.ai/en/0.8.x/api/python/dgl.sampling.html

G. C. Linderman, J. Zhao, and Y. Kluger. Zero-preserving imputation of scrna-seq data using low-rank approximation. *bioRxiv*, 2018. doi: 10.1101/397588.

R. Lopez, J. Regier, M. B. Cole, M. I. Jordan, and N. Yosef. Deep generative modeling for single-cell transcriptomics. *Nature Methods*, 15(12):1053–1058, Dec 2018. ISSN 1548-7105. doi: 10.1038/s41592-018-0229-2.

R. Lopez, A. Nazaret, M. Langevin, J. Samaran, J. Regier, M. I. Jordan, and N. Yosef. A joint model of unpaired data from scrna-seq and spatial transcriptomics for imputing missing gene expression measurements. *CoRR*, abs/1905.02269, 2019.

M. Mantri, G. J. Scuderi, R. Abedini-Nassab, M. F. Z. Wang, D. McKellar, H. Shi, B. Grodner, J. T. Butcher, and I. De Vlaminck. Spatiotemporal single-cell rna sequencing of developing chicken hearts identifies interplay between cellular differentiation and morphogenesis. *Nature Communications*, 12(1):1771, Mar 2021. ISSN 2041-1723. doi: 10.1038/s41467-021-21892-z.

K. R. Maynard, L. Collado-Torres, L. M. Weber, C. Uytingco, B. K. Barry, S. R. Williams, J. L. Catallini, M. N. Tran, Z. Besich, M. Tippani, et al. Transcriptome-scale spatial gene expression in the human dorsolateral prefrontal cortex. *Nature neuroscience*, 24(3):425–436, 2021.

D. Pham, X. Tan, J. Xu, L. F. Grice, P. Y. Lam, A. Raghubar, J. Vukovic, M. J. Ruitenberg, and Q. Nguyen. stlearn: integrating spatial location, tissue morphology and gene expression to find cell types, cell-cell interactions and spatial trajectories within undissociated tissues. *bioRxiv*, 2020. doi: 10.1101/2020.05.31.125658.

R. Satija, J. A. Farrell, D. Gennert, A. F. Schier, and A. Regev. Spatial reconstruction of single-cell gene expression data. *Nature Biotechnology*, 33:495–502, 2015. doi: 10.1038/nbt.3192.

Y. Shi, Z. Huang, S. Feng, H. Zhong, W. Wang, and Y. Sun. Masked label prediction: Unified message passing model for semi-supervised classification. *arXiv preprint arXiv:2009.03509*, 2020.

P. L. Ståhl, F. Salmén, S. Vickovic, A. Lundmark, J. F. Navarro, J. Magnusson, S. Giacomello, M. Asp, J. O. Westholm, M. Huss, et al. Visualization and analysis of gene expression in tissue sections by spatial transcriptomics. *Science*, 353(6294):78–82, 2016.

R. R. Stickels, E. Murray, P. Kumar, J. Li, J. L. Marshall, D. J. Di Bella, P. Arlotta, E. Z. Macosko, and F. Chen. Highly sensitive spatial transcriptomics at near-cellular resolution with slide-seqv2. *Nature biotechnology*, 39(3):313–319, 2021.

C. Strell, M. M. Hilscher, N. Laxman, J. Svedlund, C. Wu, C. Yokota, and M. Nilsson. Placing rna in context and space–methods for spatially resolved transcriptomics. *The FEBS journal*, 286(8):1468–1481, 2019.

T. Stuart, A. Butler, P. Hoffman, C. Hafemeister, E. Papalexi, W. M. M. III, Y. Hao, M. Stoeckius, P. Smibert, and R. Satija. Comprehensive integration of single-cell data. *Cell*, 177:1888–1902, 2019. doi: 10.1016/j.cell.2019.05.031.

D. van Dijk, R. Sharma, J. Nainys, K. Yim, P. Kathail, A. J. Carr, C. Burdziak, K. R. Moon, C. L. Chaffer, D. Pattabiraman, B. Bierie, L. Mazutis, G. Wolf, S. Krishnaswamy, and D. Pe'er. Recovering gene interactions from single-cell data using data diffusion. *Cell*, 174(3):716–729.e27, 2018. ISSN 0092-8674. doi: https://doi.org/10.1016/j.cell.2018.05.061.

P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

J. Wang, A. Ma, Y. Chang, J. Gong, Y. Jiang, R. Qi, C. Wang, H. Fu, Q. Ma, and D. Xu. scgnn is a novel graph neural network framework for single-cell rna-seq analyses. *Nature communications*, 12(1):1882, 2021.

Y. Wang, Z. Duan, B. Liao, F. Wu, and Y. Zhuang. Heterogeneous attributed network embedding with graph convolutional networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 10061–10062, 2019.

Y. Wang, Z. Duan, Y. Huang, H. Xu, J. Feng, and A. Ren. Mthetgnn: A heterogeneous graph embedding framework for multivariate time series forecasting. *Pattern Recognition Letters*, 153:151–158, 2022.

H. Xu, R. Chen, Y. Wang, Z. Duan, and J. Feng. Cosimgnn: Towards large-scale graph similarity computation. *arXiv preprint arXiv:2005.07115*, 2020.

H. Xu, Z. Duan, Y. Wang, J. Feng, R. Chen, Q. Zhang, and Z. Xu. Graph partitioning and graph neural network based hierarchical graph matching for graph similarity computation. *Neurocomputing*, 439:348–362, 2021.

Z. Zeng, Y. Li, Y. Li, and Y. Luo. Statistical and machine learning methods for spatially resolved transcriptomics data analysis. *Genome biology*, 23(1):1–23, 2022.