OXFORD

Briefings in Bioinformatics, 2025, **26(2)**, bbaf162

https://doi.org/10.1093/bib/bbaf162

Problem Solving Protocol

Multi-Manifolds fusing hyperbolic graph network balanced by pareto optimization for identifying spatial domains of spatial transcriptomics

Ying Li[1], Qifeng Hu[1], Siyu Han[2], Rui Wang-Sattler[3,*], Wei Du[1,*]

[1]Key Laboratory of Symbol Computation and Knowledge Engineering of Ministry of Education, College of Computer Science and Technology, Jilin University, Qianjin Street 2699, Changchun 130012, Jilin, China

[2]TUM School of Medicine, Technical University of Munich, Ismaninger Straße 22, D-81675 Munich, Bavaria, Germany

[3]Institute of Translational Genomics, Helmholtz Zentrum Munchen, Ingolstadter Landstraße 1, D-85764 Neuherberg, Bavaria, Germany

*Corresponding authors. Wei Du, Key Laboratory of Symbol Computation and Knowledge Engineering of Ministry of Education, College of Computer Science and Technology, Jilin University, Qianjin Street 2699, Changchun 130012, Jilin, China. E-mail: weidu@jlu.edu.cn; Rui Wang-Sattler, Institute of Translational Genomics, Helmholtz Zentrum Munchen, Ingolstadter Landstrase 1, D-85764 Neuherberg, Bavaria, Germany. E-mail: rui.wang.sattler@helmholtz-munich.de

Abstract

Identifying spatial domains for spatial transcriptomics is crucial for achieving comprehensive insights into the pathogenesis of gene expression. Increasingly, computational methods based on graph neural networks are being developed for spatial transcriptomics. However, previous methods have solely focused on the Euclidean manifold. To effectively exploit and explore the informative and deeper topological structures of inherent manifolds, we presented a Multi-Manifolds fusing hyperbolic graph network, balanced by Pareto optimization, for identifying spatial domains in Spatial Transcriptomics (MManiST). First, we developed multi-manifolds encoders for distinct manifolds using the hyperbolic neural network. Features from different manifolds were then combined using an attention mechanism, with multiple reconstruction losses balanced by Pareto optimization. Extensive experiments on commonly used benchmark datasets show that our method consistently outperforms seven state-of-the-art methods. Additionally, we investigated the validity of each component and the impact of fusion methods in ablation experiments.

Keywords: spatial transcriptomics; spatial domain identification; hyperbolic space; graph neural network

Introduction

Spatial transcriptomics (ST) captures gene expression with spatial location information [1–3]. The accessibility of spatial information offers significant potential to depict the spatial patterns and activities of cells. It provides deeper insights into the underlying mechanisms and pathology of both healthy and diseased tissues. Identifying spatial domains within ST analysis is of great significance and essence. Numerous computational methods have been developed, which can be categorized into three groups: (1) **Statistical/probabilistic model-based methods**: Giotto [4] employs a hidden Markov random field to model nodes' gene expression. Drawing inspiration from achieving super-resolution images in computer vision, BayesSpace [5] constructs a fully Bayesian statistical model with a Markov random field. (2) **Deep learning-based methods**: stLearn [6] normalizes ST by using neighbor-based smoothing and morphological adjustment and identifies spatial domains by utilizing a graph-based clustering method. (3) **Graph neural network (GNN)-based methods**: SEDR [7] simultaneously trains a deep auto-encoder and a variational graph auto-encoder to learn low-dimensional spatial embeddings of ST. SpaGCN [8] integrates ST, spatial location, and histology to identify the spatial domain of ST. DeepST [9] integrates node features and location using a denoising autoencoder and GNN. STAGATE [10] adopts an adaptive graph attention auto-encoder to decipher spatial domains by integrating ST with location information. CCST [11] introduces the unsupervised deep graph infomax model [12] for identifying spatial domains. SpaceFlow [13] introduces spatially regularized deep graph networks to generate spatially consistent embeddings. GraphST [14] fully integrates spatial information and gene expression ST by using graph self-supervised contrastive learning. For reference, a survey [15] benchmarks these state-of-the-art methods.

Most existing methods focus on exploring features within the Euclidean manifold and fail to provide in-depth insights into the complex structures inherent in ST. Unlike the Euclidean manifold, hyperbolic geometry, as a Riemannian manifold with constant negative curvature, has demonstrated remarkable success due to its ability to model complex structured and hierarchical data. Hyperbolic graph neural networks (HGNNs) based on hyperbolic geometry were introduced by [16] and [17]. The Lorentz graph convolutional network (GCN) [18] focuses on the Lorentz model. HGCL [19] introduces contrastive learning to further enhance hyperbolic graph embedding. GraphZoo [20] provides a convenient toolkit for utilizing GNNs based on hyperbolic geometries. HGNN

Received: October 28, 2024. **Revised:** February 28, 2025. **Accepted:** March 19, 2025

has been applied in various fields, including recommendation systems [21], computer vision [22], and bioinformatics [23].

Driven by a hyperbolic graph network, we introduce Lorentz and Poincaré manifolds for ST to address the limitations of the Euclidean manifold. By introducing multi-manifolds, we aim to better capture the complex and deeper topological structures inherent in ST. To fully exploit and fuse the most informative and discriminative features from different manifolds, we formulate the solution as a multi-objective optimization problem. Balancing different objectives due to potential conflicts remains a fundamental and challenging issue in multi-objective optimization. Typically, hyperparameter search or heuristic algorithms are employed to solve multi-objective optimization, but these methods are highly time-consuming and do not guarantee a satisfying solution. Therefore, based on the multiple-gradients descent algorithm (MGDA) [24], the Pareto optimal solution is considered an alternative approach. To reduce the time consumption caused by multiple backpropagation processes in MGDA, an upper bound-based approach [25] is proposed. Additionally, MGDA has been adopted to reconcile classic self-supervised learning objectives in GNNs [26]. Inspired by these works on Pareto optimality, we use Pareto optimal methods to handle multi-object optimization. We aim to achieve an effective multi-manifolds fusing hyperbolic graph network.

Based on the discussion above, we propose a multi-manifolds fusing hyperbolic graph network, balanced by Pareto optimization, for identifying spatial domains in ST, named MManiST. To our knowledge, this is the first work introducing manifold-based hyperbolic geometry for ST analysis. Our main contributions are summarized as follows:

(i) **Multi-manifolds encoders:** we design multi-manifolds encoders, incorporating both Euclidean and hyperbolic geometry, to explore the latent complex structures inherent in ST.
(ii) **Pareto optimization for reconstruction objectives:** we address the optimization problem of multiple reconstruction objectives by achieving a Pareto optimal solution using the MGDA algorithm.
(iii) **Experimental validation:** extensive experiments on commonly used benchmark datasets show that MManiST consistently outperforms seven state-of-the-art methods. Additionally, we validate the effectiveness of MManiST's components and study the influence of different spatial graph construction and fusion methods in ablation experiments.

Methodology

Symbols and abbreviations

The achieved graph is denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the vertex set and $\mathcal{E}$ is the edge set. Euclidean and Hyperbolic space are denoted as $\mathcal{E}$ and $\mathcal{H}$, respectively. In practice, we often use projection models to indirectly represent Hyperbolic geometry with the Poincaré Ball and the Lorentz model. Specifically, two Hyperbolic models, Poincaré Ball and Lorentz, are denoted as $\mathcal{B}$ and $\mathcal{L}$, respectively. The curvature of the model is denoted as c.

Model architecture

As shown in Fig. 1, the overall architecture of MManiST is composed of five modules: (1) **Spatial Graph Construction of ST:** the spatial graph is constructed using the k-nearest neighbor (KNN) algorithm on the position matrix for ST data. For the case where the number of samples is <20 000, we set each sample as a spot, and for some large-scale datasets with subcellular, such as Stereo-Seq [27], we would use cell position information provided by official split results. (2) **Multi-manifolds Encoders:** we design graph encoders in three manifolds: Euclidean, Poincaré, and Lorentz, to explore unique geometric characteristics. (3) **Multi-manifolds Fusion by Attention Mechanism:** the attention mechanism fuses node features from different manifolds. (4) **Pareto Optimization:** Pareto optimization automatically adjusts the relative weight of each reconstruction task or the relative importance of each manifold. (5) **Identifying Spatial Domain:** the fused embeddings are applied to identify the spatial domain using the Gaussian mixture clustering algorithm.

Multi-manifolds encoders

Due to the variable metric tensor in hyperbolic geometry, performing operations such as matrix-vector addition, matrix-vector multiplication, or nonlinear activation is not straightforward. The hyperbolic graph network [16] is designed to introduce multi-manifolds for ST.

Hyperbolic initialization layer

Initially, node representations on hyperbolic geometry are needed. The exponential function maps features from Euclidean space to a specific hyperbolic model. The logarithmic function performs the inverse operation. Specifically, for the Poincaré ball model, the exponential and logarithmic functions are summarized as follows:

$$\begin{aligned}\exp_{\mathbf{x}}^{c}(\mathbf{v}) &= \mathbf{x} \oplus_c \left(\tanh\left(\sqrt{|c|}\frac{\lambda_{\mathbf{x}}^{2}\|\mathbf{v}\|_2}{2}\right)\frac{\mathbf{v}}{\sqrt{|c|}\|\mathbf{v}\|_2}\right)\\ \log_{\mathbf{x}}^{c}(\mathbf{y}) &= \frac{2}{\sqrt{|c|}\lambda_{\mathbf{x}}^{c}}\tanh^{-1}\left(\sqrt{|c|}\left\|-\mathbf{x}\oplus_c\mathbf{y}\right\|_2\right)\frac{-\mathbf{x}\oplus_c\mathbf{y}}{\left\|-\mathbf{x}\oplus_c\mathbf{y}\right\|_2}\end{aligned} \quad (1)$$

For operations on the Lorentz model, the exponential and logarithmic functions are formulated as follows:

$$\begin{aligned}\exp_{\mathbf{x}}^{c}(\mathbf{v}) &= \cosh\left(\sqrt{|c|}\|\mathbf{v}\|_{\mathcal{L}}\right)\mathbf{x} + \mathbf{v}\frac{\sinh\left(\sqrt{|c|}\|\mathbf{v}\|_{\mathcal{L}}\right)}{\sqrt{|c|}\|\mathbf{v}\|_{\mathcal{L}}}\\ \log_{\mathbf{x}}^{c}(\mathbf{y}) &= \frac{\cosh^{-1}\left(c\langle\mathbf{x},\mathbf{y}\rangle_{\mathcal{L}}\right)}{\sinh\left(\cosh^{-1}\left(c\langle\mathbf{x},\mathbf{y}\rangle_{\mathcal{L}}\right)\right)}\left(\mathbf{y} - c\langle\mathbf{x},\mathbf{y}\rangle_{\mathcal{L}}\mathbf{x}\right)\end{aligned} \quad (2)$$

Representations on two models are calculated as follows:

$$\begin{aligned}\mathbf{x}^{\mathcal{B}} &= \exp_{0}^{c}(\mathbf{x}^{E})\\ \mathbf{x}^{\mathcal{L}} &= \exp_{\mathbf{o}}^{c}((0,\mathbf{x}^{E}))\end{aligned} \quad (3)$$

To adapt the hierarchical features of different types of data, the curvature c is set as a trainable parameter.

Hyperbolic feature transformation

Instead of directly using Möbius multiplication, we adopt an alternative approach driven by the model [16] to implement matrix-vector multiplication. Specifically, we first use a logarithmic mapping function to project hyperbolic features into their tangent space, a vector space isomorphic to Euclidean space. Thus, Euclidean multiplication can be applied in the tangent space. Subsequently, an exponential mapping function maps

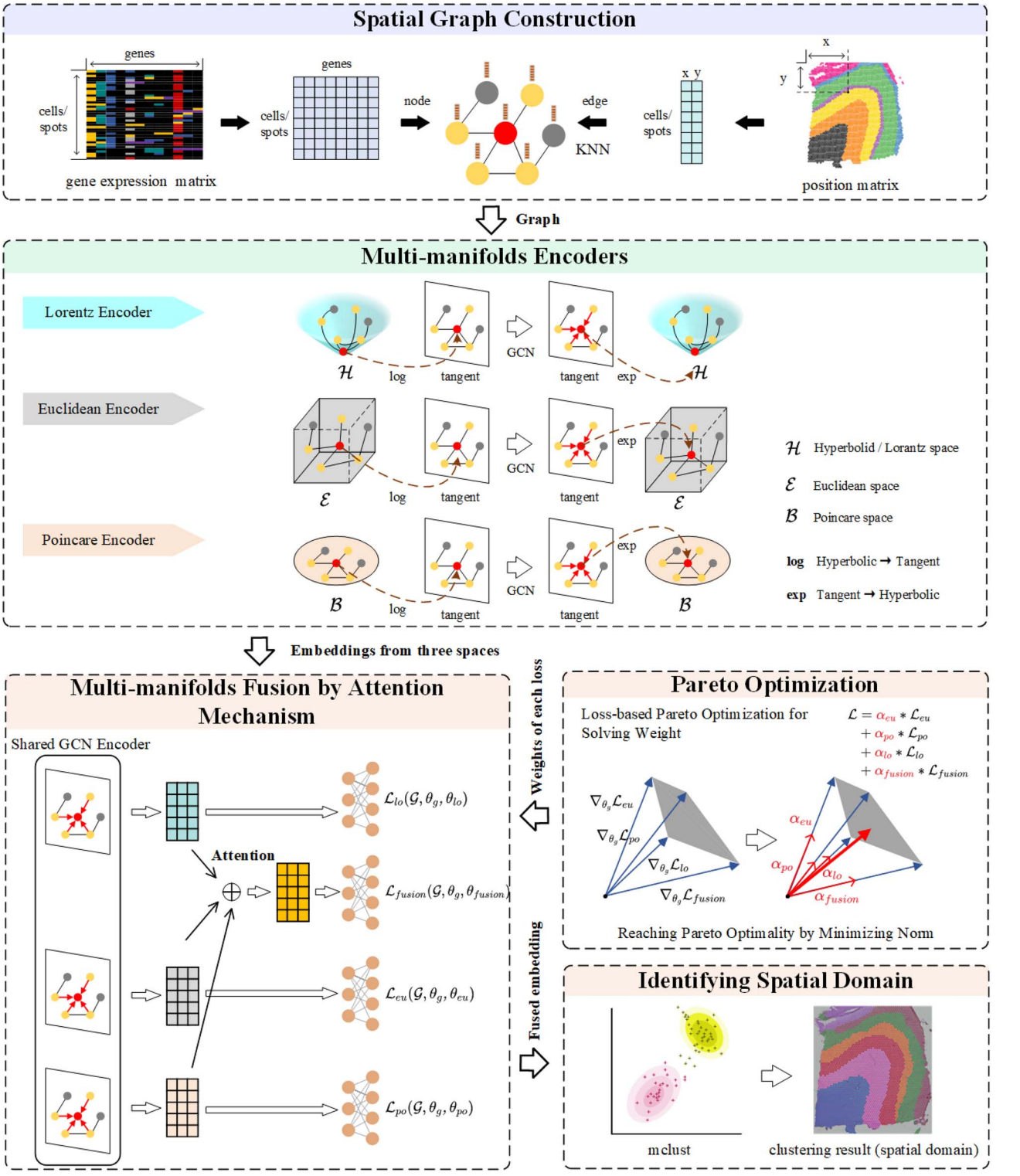

Figure 1. The workflow of MManiST: Spatial Graph Construction, Multi-manifolds Encoder, Multi-manifolds Fusion by Attention Mechanism, Pareto Optimization and Identifying spatial domain.

the transformed features back to the corresponding hyperbolic space. The multiplication process of the two models is shown in Eq. (4).

$$\begin{aligned} \mathbf{M} \otimes_c^{\mathcal{B}} \mathbf{x}^{\mathcal{B}} &= \exp_o^c\left(\mathbf{M} \log_o^c\left(\mathbf{x}^{\mathcal{B}}\right)\right) \\ \mathbf{M} \otimes_c^{\mathcal{L}} \mathbf{x}^{\mathcal{L}} &= \exp_o^c\left(0, \mathbf{M} \log_o^c\left(\mathbf{x}^{\mathcal{L}}\right)_{[1:n]}\right) \end{aligned} \tag{4}$$

Similarly, matrix-vector addition is implemented by parallel transport on the tangent space as follows:

$$\mathbf{x}^{\mathcal{H}} \oplus_c^{\mathcal{H}} \mathbf{b}^{\mathcal{H}} = \exp_{\mathbf{x}^{\mathcal{H}}}^c\left(PT_{0 \to \mathbf{x}^{\mathcal{H}}}^c\left(\log_o^c\left(\mathbf{b}^{\mathcal{H}}\right)\right)\right), \tag{5}$$

where PT means parallel transport.

Parallel transport operations in two models are formulated as follows:

$$PT^{c}_{\mathbf{x}\to\mathbf{y}}(\mathbf{v}) = \frac{\lambda^{c}_{\mathbf{x}}}{\lambda^{c}_{\mathbf{y}}}\operatorname{gyr}[\mathbf{y},-\mathbf{x}]\,\mathbf{v}$$

$$PT^{c}_{\mathbf{x}\to\mathbf{y}}(\mathbf{v}) = \mathbf{v} - \frac{c\langle\mathbf{y},\mathbf{v}\rangle_{\mathcal{L}}}{1+c\langle\mathbf{x},\mathbf{y}\rangle_{\mathcal{L}}}(\mathbf{x}+\mathbf{y}) \tag{6}$$

Hyperbolic neighborhood aggregation

In GNNs, each adjacent node is considered equally important. To eliminate the impact of the node degree, we normalize the weight by the degree of the two endpoints. Additionally, a highly weighted self-loop is added to avoid the risk of over-smoothing, which is formulated as follows:

$$\begin{cases} \mathrm{A}_{ij} = 1, & i = j \\ \mathrm{A}_{ij} = 1/\sqrt{d_i d_j}, & i \neq j \end{cases} \tag{7}$$

For message aggregation, the sum aggregation strategy is adopted in the following:

$$\mathrm{AGG}(\mathbf{x}^{\mathcal{H}}_i) = \exp^{c}_{\mathrm{o}}\left(\sum_{j\in\mathcal{N}_i}\mathrm{A}_{ij}\left(\log^{c}_{\mathrm{o}}(\mathbf{x}^{\mathcal{H}}_i)\right)\right), \tag{8}$$

where $\mathbf{x}^{\mathcal{H}}_i$ represents the embedding of the ith node and $\mathcal{N}_i$ denotes the neighbors of node i.

Nonlinear activation

Nonlinear activation operation still needs to be projected into tangent space first, activated, and then projected back, as follows:

$$\begin{aligned} \sigma^{\otimes^{c_{l-1},c_l}}(\mathbf{x}^{\mathcal{B}}) &= \exp^{c_l}_{\mathrm{o}}\left(\sigma\left(\log^{c_{l-1}}_{\mathrm{o}}(\mathbf{x}^{\mathcal{B}})\right)\right) \\ \sigma^{\otimes^{c_{l-1},c_l}}(\mathbf{x}^{\mathcal{L}}) &= \exp^{c_l}_{\mathrm{o}}\left(0,\sigma\left(\log^{c_{l-1}}_{\mathrm{o}}\left(\mathbf{x}^{\mathcal{L}}_{[1:n]}\right)\right)\right), \end{aligned} \tag{9}$$

where σ denotes the nonlinear activation function. It is worth noting that the activation process is in the middle of two layers, or in other words, in the middle of two manifolds, which might have different curvatures.

Based on the above operations, the total process of the hyperbolic graph network layer is represented as follows:

$$\begin{aligned} \mathbf{h}^{l,\mathcal{H}}_i &= \left(\mathbf{W}^{l}\otimes^{c_{l-1}}\mathbf{x}^{l-1,\mathcal{H}}_i\right)\oplus^{c_{l-1}}\mathbf{b}^{l} \\ \mathbf{y}^{l,\mathcal{H}}_i &= \mathrm{AGG}^{c_{l-1}}\left(\mathbf{h}^{l,\mathcal{H}}\right)_i \\ \mathbf{x}^{l,\mathcal{H}}_i &= \sigma^{\otimes^{c_{l-1},c_l}}\left(\mathbf{y}^{l,\mathcal{H}}_i\right), \end{aligned} \tag{10}$$

where $\mathbf{W}^{l}$ is trainable weight matrix of l-layer.

Multi-manifolds fusion by attention mechanism

To obtain a more informative representation, we enhance and extend the architecture of heterogeneous attention graph networks [28] for multi-manifolds fusion. Specifically, the attention score matrix $\mathbf{B}$ is computed for adjacent nodes by taking the dot product between the trainable attention vector $\mathbf{v}$ and the concatenation of embeddings from each manifold. Then, this score matrix is normalized. The process of calculating attention score matrix is formulated as Eq. (11) and Eq. (12):

$$\mathrm{B}_{ij} = \begin{cases} \sigma\left(\mathbf{v}^{\top}\cdot\left[\mathbf{x}^{\mathcal{E}}_i\|\mathbf{x}^{\mathcal{B}}_j\|\mathbf{x}^{\mathcal{L}}_i\right]\right), & \mathrm{A}_{ij}\neq 0 \\ 0, & \mathrm{A}_{ij}=0 \end{cases} \tag{11}$$

$$\tilde{\mathrm{B}}_{ij} = \frac{\exp(\mathrm{B}_{ij})}{\sum_{j'\in\mathcal{N}_i}\exp(\mathrm{B}_{ij'})} \tag{12}$$

To alleviate overfitting, a damping factor λ is used to update the embeddings:

$$\mathrm{X}' = \left((1-\lambda)\,\mathrm{A}+\lambda\tilde{\mathrm{B}}\right)\mathrm{X}^{\mathcal{B}} \tag{13}$$

Pareto optimization for multi-reconstruction tasks

Balancing multiple objective functions is a perennial challenge. In this study, we employ the Multiple Gradient Descent Algorithm (MGDA) [24] to determine the weights of each objective function for Pareto optimality. Specifically, we design a shared GCN encoder that serves two roles. Firstly, input embeddings are encoded to a lower dimension in the forward propagation, and the information from neighboring nodes is further aggregated. The gradients produced from each objective function are recorded during the backward propagation. These gradients form the basis for calculating weight parameters.

Considering the gradients related to each loss function as a vector in the parameter space, these four vectors constitute a convex hull. The process of reaching Pareto optimality is equivalent to persistently taking steps in a direction that possesses the minimal common vector norm [24].

For the simple scenario of balancing two objectives, assuming that the sum of all weights is equal to 1, the goal can be expressed as follows:

$$\min_{\gamma\in[0,1]}\left\|\gamma\nabla_{\theta^g}\hat{\mathcal{L}}^1(\theta^g,\theta^1)+(1-\gamma)\nabla_{\theta^g}\hat{\mathcal{L}}^2(\theta^g,\theta^2)\right\|^2_2, \tag{14}$$

where θ^g represents the global weight parameter in the shared GCN encoder, and θ^1 and θ^2 represent task-specific weight parameters.

We consider three conditions as depicted in Fig. 2 to achieve the minimal common vector norm. Based on this, γ is calculated by Eq. (15).

$$\gamma = \left[\frac{\left(\nabla_{\theta^g}\hat{\mathcal{L}}^2(\theta^g,\theta^2)-\nabla_{\theta^g}\hat{\mathcal{L}}^1(\theta^g,\theta^1)^{\top}\right)\nabla_{\theta^g}\hat{\mathcal{L}}^2(\theta^g,\theta^2)}{\left\|\nabla_{\theta^g}\hat{\mathcal{L}}^1(\theta^g,\theta^1)-\nabla_{\theta^g}\hat{\mathcal{L}}^2(\theta^g,\theta^2)\right\|^2_2}\right]_{+,\frac{1}{\top}}, \tag{15}$$

where $[\cdot]_{+,\frac{1}{\top}}$ represents clipping to the interval $[0,1]$.

For the case of more than two objectives, we adopt the Frank–Wolfe algorithm [29] to iteratively solve for the weight vector $\boldsymbol{\alpha}\in\mathbb{R}^K$ with an initial value as $1/K$, where K is a total number of objectives. We denote current common descent direction as $\hat{\nabla}_{\theta^g}=\sum_{k=1}^{K}\boldsymbol{\alpha}_k\cdot\nabla_{\theta^g}\mathcal{L}_k(\mathcal{G};\theta^g,\theta^t)$. In each iteration, we update the objective whose descent direction correlates least with current common descent direction. Therefore, we choose t that satisfies $\mathrm{t}=\arg\min_r\sum_{i=1}^{K}\boldsymbol{\alpha}_i\cdot\nabla_{\theta^g}\mathcal{L}_i(\mathcal{G};\theta^g,\theta^r)^{\top}$ as the objective to be updated.

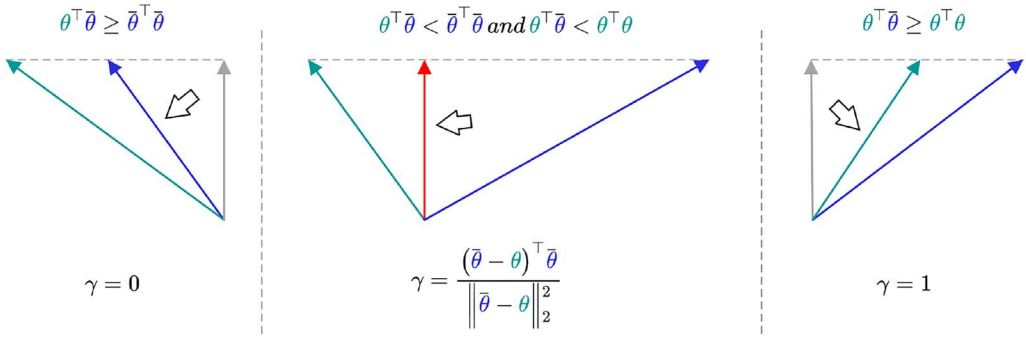

Figure 2. The case of two objectives, arrow points to common vector with minimal norm.

One iteration of the updating process is formulated as follows:

$$\alpha := (1-\eta)\cdot\alpha + \eta\cdot \mathbf{e}_t$$
$$\eta = \left[\frac{\hat{\nabla}_{\theta^g}\cdot\left(\hat{\nabla}_{\theta^g} - \nabla_{\theta^g}\mathcal{L}_t\left(\mathcal{G};\theta^g,\theta^t\right)\right)^\top}{\left\|\hat{\nabla}_{\theta^g} - \nabla_{\theta^g}\mathcal{L}_t\left(\mathcal{G};\theta^g,\theta^t\right)\right\|_2^2}\right]_{+,\frac{1}{T}}, \quad (16)$$

where η is step size, and e_t is a one-hot vector with tth element equal to 1.

The iterations repeat until η is smaller than a predefined threshold or the number of iterations exceeds the maximum value allowed. After α is computed, the training loss is set as follows:

$$\mathcal{L}_{total} = \sum_{i=1}^{K}\alpha_i\cdot\mathcal{L}_i\left(\mathcal{G};\theta^g,\theta^t\right), \quad (17)$$

where K is number of tasks and in this study $\mathcal{L}$ means MSE loss for reconstruction.

Identifying spatial domain

We employed a Gaussian mixture model-based clustering algorithm, specifically mclust [30], to identify spatial domains by clustering the fused embeddings. The number of spatial domains is set to match the ground truth. For those dataset without ground truth, we choose Leiden algorithm to get the clustering results. We varied the resolution parameter from 0.2 to 1.0 and selected the resolution corresponding to the highest silhouette score. Additionally, to achieve higher consistency, we offer an optional refinement operation that aligns each node with the majority in its community within the selected radius.

Results

MManiST well identified spatial domains of dorsolateral prefrontal cortex data

As a common dataset, the LIBD human dorsolateral prefrontal cortex (DLPFC) is often used as a benchmark (Fig. 3(A)). We explored the performance of MManiST on this dataset. Compared with other state-of-the-art methods, MManiST could more accurately identify spatial domains. As shown in Fig. 3(B), MManiST outperformed previous methods across multiple evaluation metrics. Figure 3(C) demonstrates the visualized spatial domain identification results on section 151509. Regarding the ground truth labels in Fig. 3(B), we found that MManiST identified the complete Layer 1 and Layer 2 regions, while other methods tended to separate these regions easily. Additionally, the spatial domains identified by our method exhibited better spatial continuity. Figure 3(D) depicts the 2D UMAP visualization of the low-dimensional representations obtained by each encoder-based method. stLearn [6], SEDR [7], and SpaceFlow [13] displayed dispersed label distribution, corresponding to a lack of spatial continuity in the spatial domain identification results. The label distribution of STAGATE [10], GraphST [14], and MManiST is more compact. However, STAGATE [10] could not clearly classify Layer 5 and Layer 6, and a portion of Layer 2 was mixed into Layer 3. GraphST [14] also struggled to build clear boundaries between Layers 2 and 5. In contrast, our method produced distinct boundaries for different labels. We attempted to use the PAGA graph to reveal the potential developmental trajectory of tissue types, as shown in Fig. 3(E). We found that SpaceFlow [13], STAGATE [10], and MManiST exhibited a linear developmental trajectory, consistent with our prior knowledge of DLPFC.

MManiST demonstrates excellent noise reduction capabilities. In Fig. 4(A), we selected seven layer marker genes—GFAP, HPCAL1, CARTPT, NEFH, S100A11, FN1, and PLP1—and showed their spatial distribution before (original gene expression) and after (reconstructed gene expression) denoising. After denoising, the distribution of these genes became more concentrated in one or several layers, exhibiting more distinct spatial distribution characteristics. Figure 4(B) illustrates violin plots showing the distribution of layer marker genes across layers before and after denoising. Post-denoising, the expression of these genes shifted from a relatively uniform distribution to a more concentrated distribution in one or several layers, with expression in the remaining layers approaching zero.

MManiST well identified spatial domains of osmFISH data

In Fig. 5(B), we present the spatial domain identification results of various methods on the mouse somatosensory cortex dataset by osmFISH. MManiST achieved the highest ARI value. While CCST [11], SpaceFlow [13], GraphST [14], and our method were all able to identify a relatively clear hierarchical structure, the spatial domains identified by CCST [11] and SpaceFlow [13] lacked distinct boundaries and contained many noise points. Despite good spatial continuity, GraphST [14] could not distinguish between the Layer 2-3 lateral and Layer 2-3 medial subregions. In contrast, MManiST achieved a more accurate and refined identification that other methods could not. Figure 5(C) illustrates the 2D UMAP visualization of the low-dimensional representations obtained by each method. We found that the representations obtained by MManiST exhibited the smallest intra-class distance and the largest inter-class distance. Because our method can clearly separate spots of different categories,

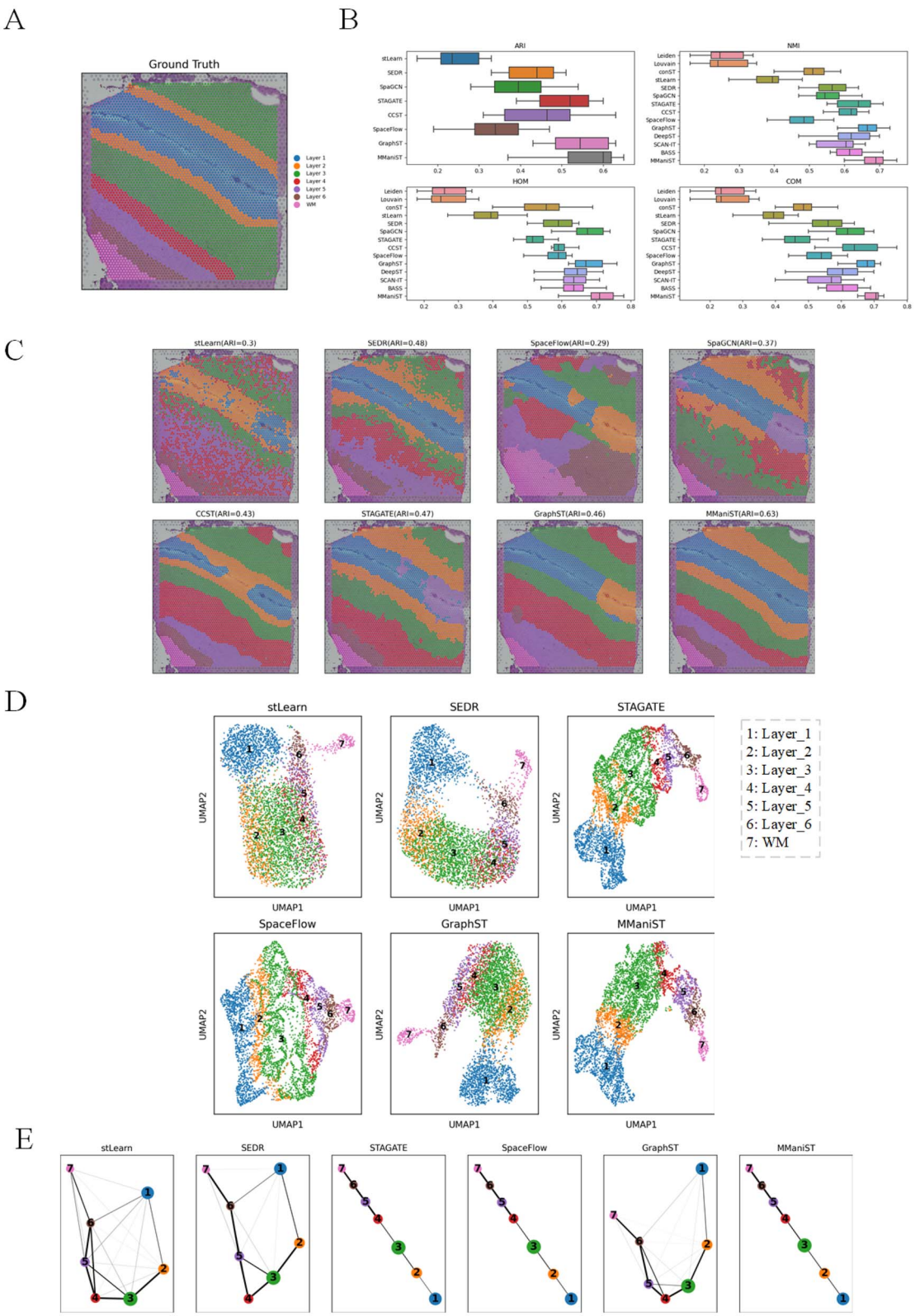

Figure 3. Spatial domain identification result of 10x Visium DLPFC data: (A) Ground truth, (B) Boxplot of clustering accuracy of the DLPFC dataset, (C) Spatial domain visualizations generated by stLearn, SEDR, SpaceFlow, SpaGCN, CCST, STAGATE, GraphST and MManiST on DLPFC section 151509, (D-E) UMAP visualizations and PAGA graphs generated by stLearn, SEDR, SpaceFlow, GraphST and MManiST.

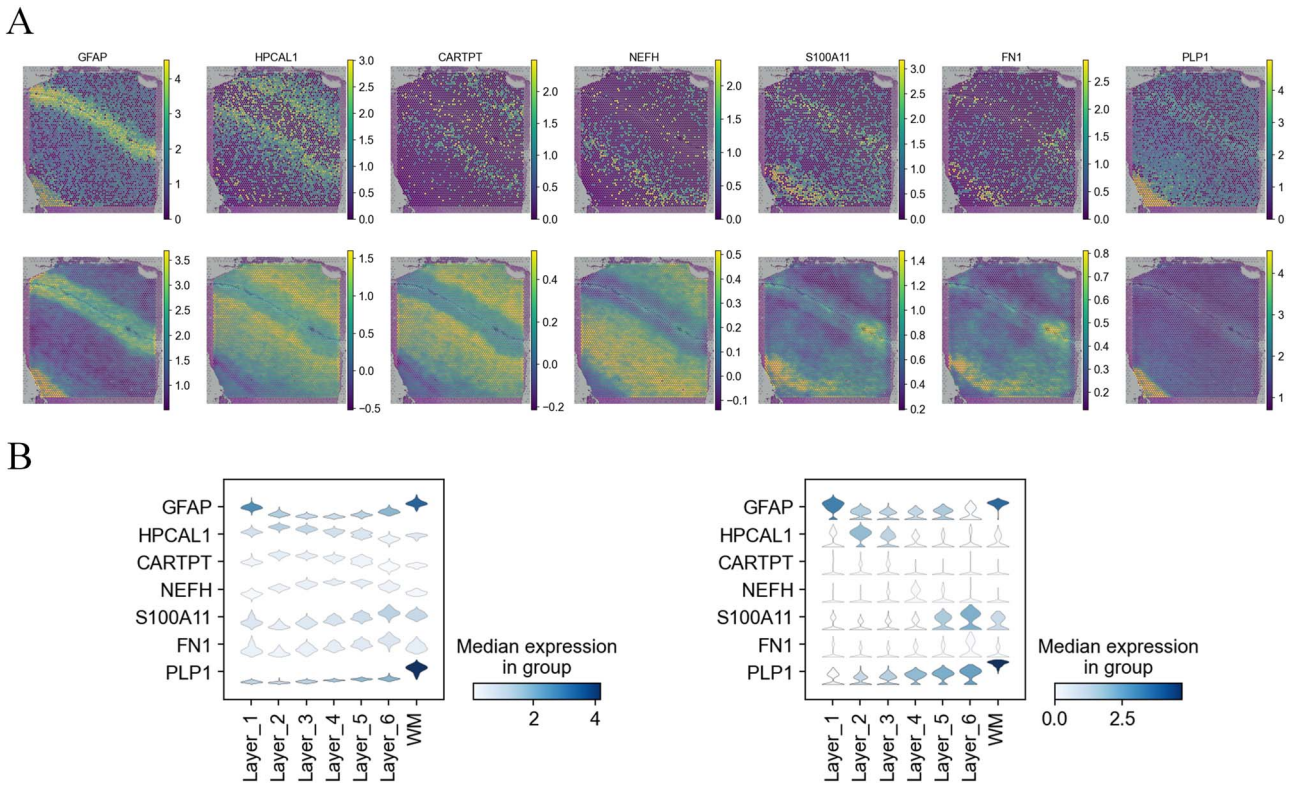

Figure 4. Denoising result of DLPFC data: (A) Layer-specific marker genes expression before (upper) and after (bottom) denoising, (B) Violin plot of layer-specific marker genes before (left) and after (right) denoising.

the predicted developmental trajectory shown in the PAGA graph in Fig. 5(D) had fewer edges and a more distinct developmental pathway than other methods. Both SpaceFlow [13] and our method predicted a clear developmental trajectory, with the spot types along this pathway being Pia Layer 1, Layer 2-3 medial, Layer 2-3 lateral, Layer 3-4, Layer 4, Layer 5, and Layer 6.

MManiST well identified spatial domains of starMAP data

MManiST achieved the best performance in spatial domain identification on the starMAP mouse medial prefrontal dataset. In Fig. 6, we selected the 20180417_BZ5_control section for demonstration. As shown in Fig. 6(A), this section is unevenly divided into four spatial domains, with Domain 1 being the smallest and Domain 3 the largest. In Fig. 6(C), we presented the spatial domain identification results of different methods for this section. We observed that stLearn [6], SEDR [7], SpaGCN [8], and STAGATE [10] did not identify distinct spatial domains, while CCST [11] incorrectly assigned the majority of spots to a single spatial domain. Although SpaceFlow [13] successfully divided the section into four spatial domains, it did not accurately distinguish the quantitative relationship between Domain 2 and Domain 3. GraphST [14] and MManiST successfully delineated these four spatial domains, demonstrating our method's ability to identify imbalanced spatial domain data. From the UMAP visualization shown in Fig. 6(D), we observed that the actual label distributions for SpaceFlow [13], GraphST [14], and MManiST were relatively concentrated. The PAGA graph shown in Fig. 6(E) indicated that only SpaceFlow [13] and MManiST exhibited a clear linear developmental trajectory, which aligns with the actual data.

MManiST well identified spatial domains of baristaseq and merFISH data

MManiST also demonstrated strong performance on data obtained through other techniques, highlighting the generalizability of hyperbolic embeddings across different data types. We presented the spatial domain identification results of various methods on the BaristaSeq mouse primary visual cortex and merFISH mouse hypothalamus datasets in Fig. 7. From Figures 7(B) and (E), we found that our method achieved the best performance across multiple evaluation metrics on both data sets. Figure 7(C) demonstrated the analysis results on the BaristaSeq Slice:2 section. stLearn [6], SEDR [7], and SpaGCN [8] failed to identify a clear hierarchical structure, and the spatial domains identified by STAGATE [10] lacked spatial continuity. GraphST [14] and CCST [11] did not effectively separate certain categories, resulting in some categories containing a larger number of spots. Although SpaceFlow [13] identified a detailed hierarchical structure, it did not effectively distinguish between the VISp_IV and VISp_V layers. We demonstrated the analysis results on the merFISH_0.19 section in Fig. 7(F). We found that STAGATE [10], SpaceFlow [13], GraphST [14], and our method all delineated accurate spatial domains. However, only our method identified the complete MPA region, while other methods partitioned this region in various ways.

Ablation study and parameter analysis

To thoroughly investigate the effectiveness and significance of each component in MManiST, we conducted comprehensive ablation experiments from four perspectives using the DLPFC benchmark dataset. Figures 8(A) to (D) illustrate the impact

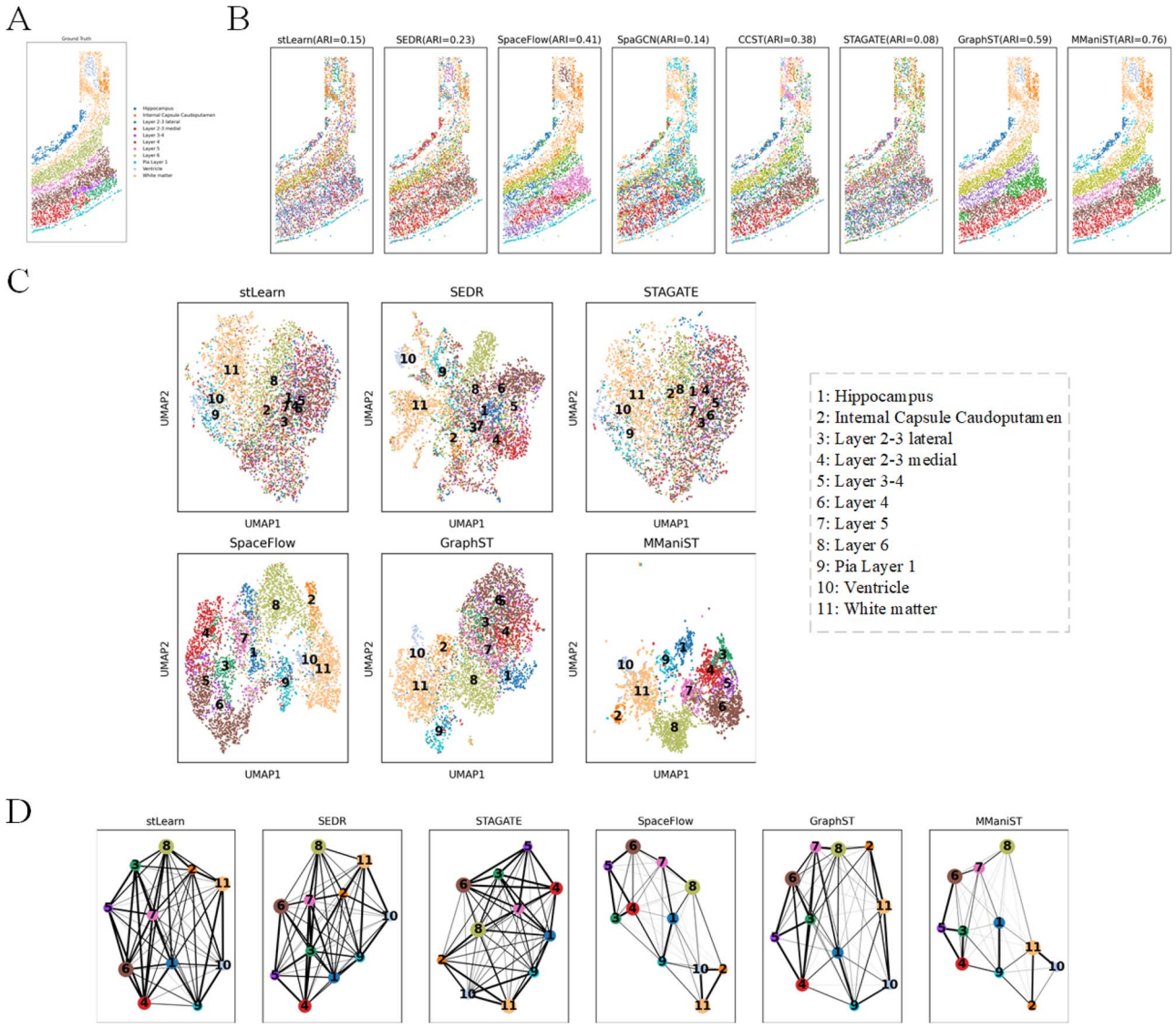

Figure 5. Spatial domain identification result of osmFISH mouse somatosensory cortex data: (A) Ground Truth, (B) Spatial domain visualizations generated by stLearn, SEDR, SpaceFlow, SpaGCN, CCST, STAGATE, GraphST and MManiST, (C-D) UMAP visualizations and PAGA graphs generated by stLearn, SEDR, SpaceFlow, GraphST and MManiST.

of composition methods, embedding manifolds, feature fusion approaches, and the use of Pareto optimization on spatial domain identification results. In Fig. 8(C), we tested various traditional fusion methods, including linear attention [31], self-attention and its multi-head version [32], channel attention [33], inter-view attention (proposed in this study), co-guided fusion [34], and simple concatenation. We reached the following conclusions: firstly, we compared different graph construction methods with highlight the superiority of a location-based graph. However, the KNN and spatial radius algorithms showed no significant performance difference. Additionally, comparing three manifolds demonstrated that graph auto-encoders in hyperbolic space achieved remarkably better results. Investigating different fusion methods showed that an appropriate fusion approach could effectively improve performance, with our inter-view attention achieving the best results. Furthermore, comparing results with and without Pareto optimization highlighted its validity in balancing multiple objectives, though the improvement was marginal.

In the training process, two main parameters were considered: the number of neighbors k and training epochs (*epoch*). To investigate the impact of each parameter, we computed average ARI and NMI values across 12 DLPFC sections under different conditions, illustrated in Fig. 8(E) and (F). Overall, MManiST demonstrated robust performance with respect to parameter variations. With the increase of k, spatial domain identification performance initially improved and then deteriorated, especially when k reached 15. The model achieved its best result when *epoch* reached 3500.

Conclusion

In this study, we propose a multi-manifolds fusing hyperbolic graph network, balanced by Pareto optimization, for identifying ST spatial domains, named MManiST. To explore the complex structure inherent in ST, we introduce the hyperbolic space, including the Lorentz and Poincaré manifolds. To effectively fuse multi-manifolds, we solve the problem within a multi-objective optimization framework. Additionally, to balance multiple reconstruction losses, we utilize Pareto optimization. Experiments demonstrate the advancements of MManiST over state-of-the-art

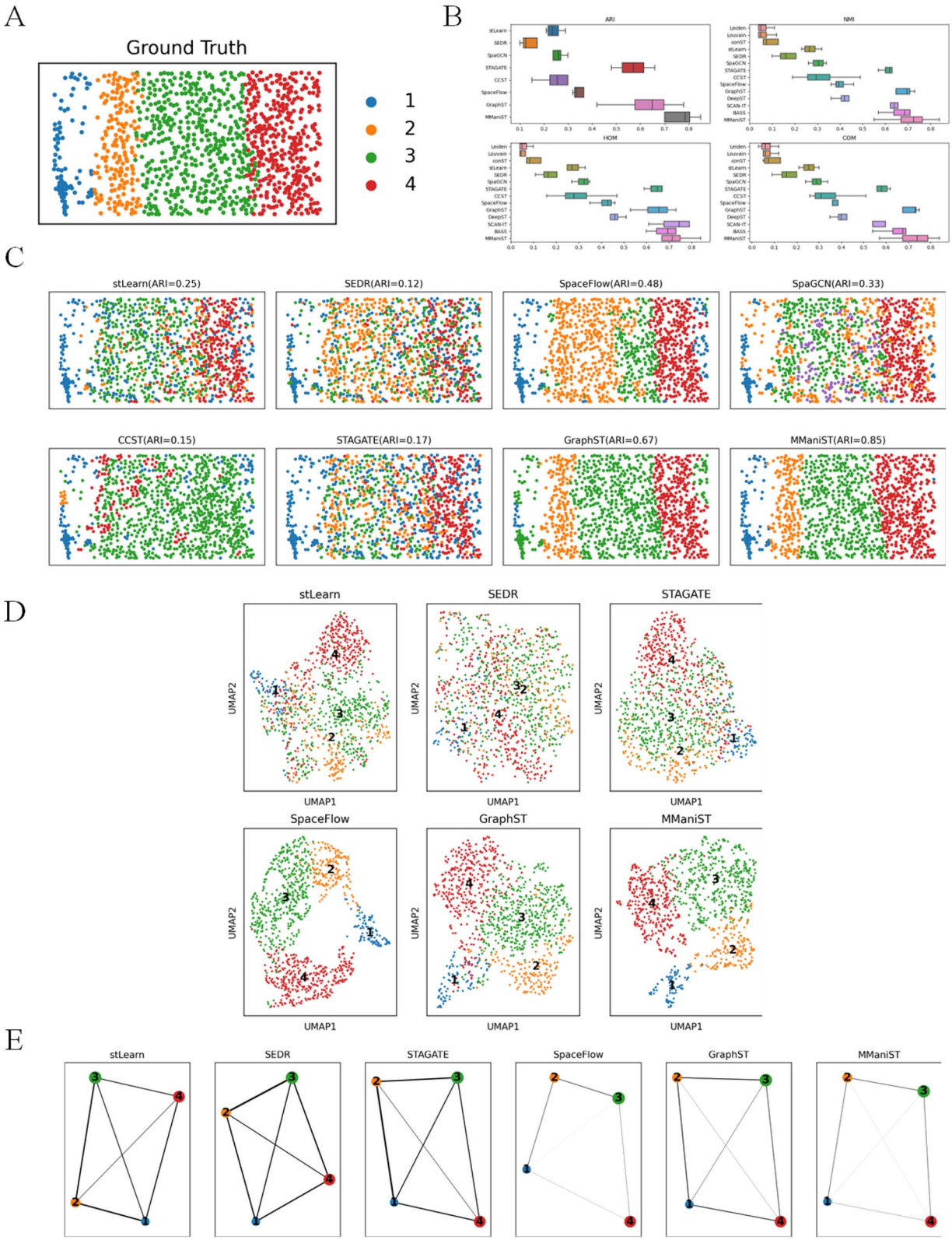

Figure 6. Spatial domain identification result of starMAP mouse mddial prefrontal dataset: (A) Ground Truth, (B) Boxplot of clustering accuracy of the starMAP dataset, (C) Spatial domain visualizations generated by stLearn, SEDR, SpaceFlow, SpaGCN, CCST, STAGATE, GraphST and MManiST on section 20180417_BZ5_control, (D-E) UMAP visualizations and PAGA graphs generated by stLearn, SEDR, SpaceFlow, GraphST and MManiST.

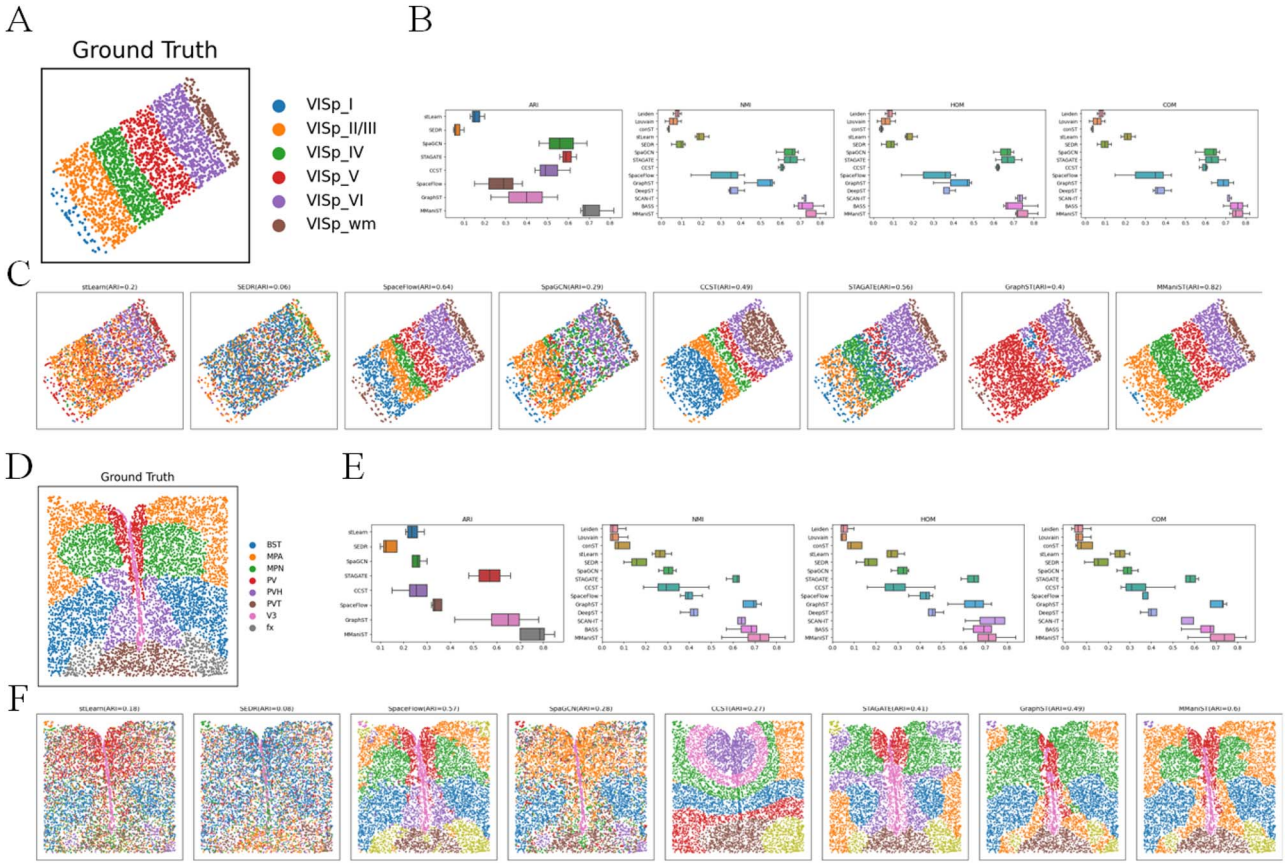

Figure 7. Spatial domain identification result of BaristaSeq and merFISH dataset: (A) Ground Truth of BaristaSeq mouse primary visual cortex dataset, (B) Boxplot of clustering accuracy of the BaristaSeq dataset, (C) Spatial domain visualizations of BaristaSeq dataset, (D) Ground Truth of merFISH mouse hypothalamus datasets, (E) Boxplot of clustering accuracy of the merFISH dataset, (F) Spatial domain visualizations of merFISH dataset.

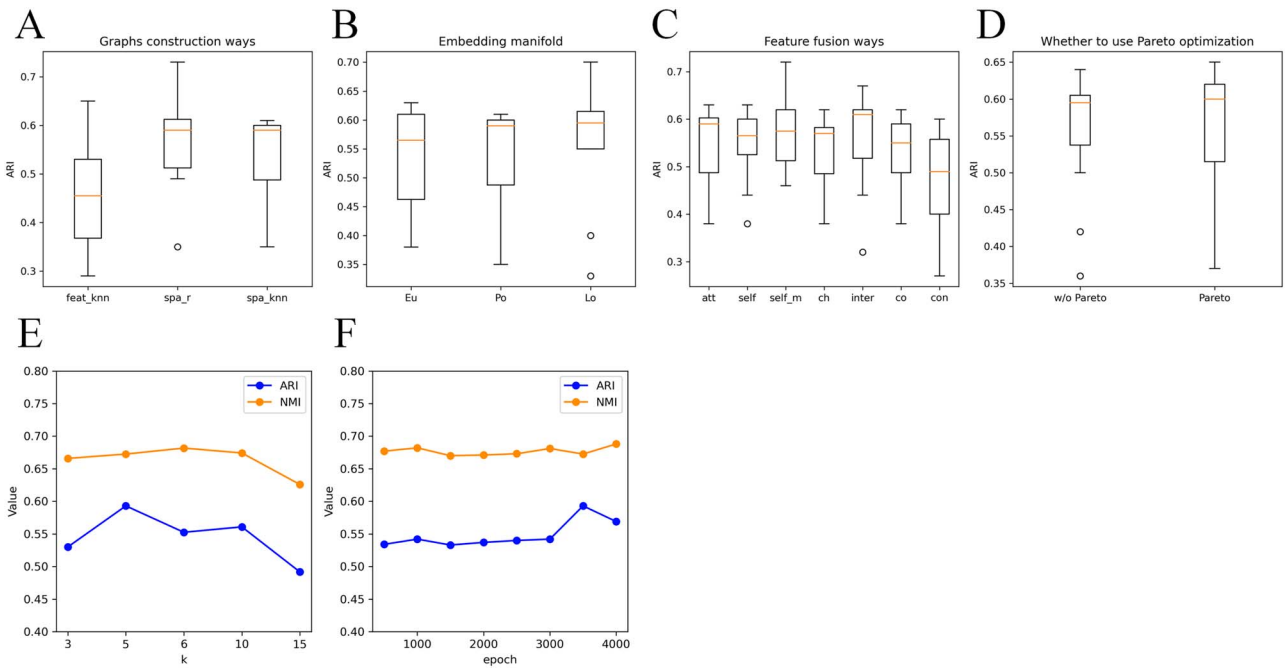

Figure 8. Ablation study and parameter analysis: (A-D) Ablation study of graph construction ways, embedding manifold, feature fusion ways and Pareto optimization, (E-F) Parameter analysis with respect to *k* and *epoch*. **feat_knn:** knn graph by feature, **spa_r:** radius graph by spatial location, **spa_knn:** knn graph by spatial location, **Eu:** Euclidean, **Po:** Poincaré Ball, **Lo:** Lorentz, **att:** linear attention, **self:** self-attention, **self_m:** self-attention multi-head version, **ch:** channel attention, **inter:** inter-view attention, **co:** co-guided attention, **con:** concatenation.

and basic methods. Extensive ablation experiments validate the effectiveness of each component and further investigate the potential application of other technologies in this field.

Future studies will explore several directions for improving the spatial domain of ST. First, the MGDA algorithm's multiple gradient backward propagation process significantly increases time complexity. The approach proposed by Sener [25] offers a theoretical foundation for reducing time expenditure, which is expected to address this issue. Furthermore, a solid theoretical framework for the fusion of different manifolds still needs to be developed, requiring more effort.

Key Points

- The study addresses the importance of identifying spatial domains for spatial transcriptomics to understand gene expression's pathogenesis.
- A new method called MManiST is introduced, which encodes gene expression data into both Euclidean and Hyperbolic manifolds by hyperbolic graph network.
- MManiST combines features from different manifolds using an attention mechanism and balances multiple reconstruction losses using Pareto optimization.
- The method has been tested on benchmark datasets, showing consistent outperformance compared with seven state-of-the-art methods.
- Ablation experiments were conducted to assess the validity of each component and the impact of different fusion methods.

Supplementary data

Supplementary data is available at *Briefings in Bioinformatics* online.

Conflict of interest

None declared.

Funding

The authors acknowledge financial support from the National Natural Science Foundation of China (62372494) and Natural Science Foundation of Jilin Province (20240302086GX, 20220101117JC).

Data availability

DLPFC dataset is from ref. [35], which can be downloaded from http://spatial.libd.org/spatialLIBD. OsmFISH, starMAP, merFISH, and BaristaSeq dataset can be downloaded from http://sdmbench.drai.cn/ with paper [15].

Code availability

All source codes used in our experiments have been deposited at https://github.com/xiao-kong-long/MManiST/tree/master.

Training Details

We trained MManiST on one NVIDIA RTX 3090 for 3500 epochs, taking ∼5 min. On the training process, ∼8GB of GPU memory being used. We employ an Adam optimizer with a learning rate 1e-4.

References

1. Crosetto N, Bienko M, Van Oudenaarden A. Spatially resolved transcriptomics and beyond. *Nat Rev Genet* 2015;**16**:57–66. https://doi.org/10.1038/nrg3832
2. Rao A, Barkley D, França GS. *et al.* Exploring tissue architecture using spatial transcriptomics. *Nature* 2021;**596**:211–20. https://doi.org/10.1038/s41586-021-03634-9
3. Moses L, Pachter L. Museum of spatial transcriptomics. *Nat Methods* 2022;**19**:534–46. https://doi.org/10.1038/s41592-022-01409-2
4. Dries R, Zhu Q, Dong R. *et al.* Giotto: A toolbox for integrative analysis and visualization of spatial expression data. *Genome Biol* 2021;**22**:1–31.
5. Zhao E, Stone MR, Ren X. *et al.* Spatial transcriptomics at subspot resolution with bayesspace. *Nat Biotechnol* 2021;**39**:1375–84. https://doi.org/10.1038/s41587-021-00935-2
6. Pham D, Tan X, Balderson B. *et al.* Robust mapping of spatiotemporal trajectories and cell–cell interactions in healthy and diseased tissues. *Nat Commun* 2023;**14**:7739. https://doi.org/10.1038/s41467-023-43120-6
7. Xu H, Fu H, Long Y. *et al.* Unsupervised spatially embedded deep representation of spatial transcriptomics. *Genome Medicine* 2024;**16**:12.
8. Jian H, Li X, Coleman K. *et al.* Spagcn: Integrating gene expression, spatial location and histology to identify spatial domains and spatially variable genes by graph convolutional network. *Nat Methods* 2021;**18**:1342–51.
9. Chang X, Jin X, Wei S. *et al.* Deepst: Identifying spatial domains in spatial transcriptomics by deep learning. *Nucleic Acids Res* 2022;**50**:e131–1.
10. Dong K, Zhang S. Deciphering spatial domains from spatially resolved transcriptomics with an adaptive graph attention auto-encoder. *Nat Commun* 2022;**13**:1739. https://doi.org/10.1038/s41467-022-29439-6
11. Li J, Chen S, Pan X. *et al.* Cell clustering for spatial transcriptomics data with graph neural networks. *Nat Comput Sci* 2022;**2**:399–408. https://doi.org/10.1038/s43588-022-00266-5
12. Veličković P, Fedus W, Hamilton WL. *et al.* Deep graph infomax. *ICLR (poster)* 2019;**2**:4.
13. Ren H, Walker BL, Cang Z. *et al.* Identifying multicellular spatiotemporal organization of cells with spaceflow. *Nat Commun* 2022;**13**:4076. https://doi.org/10.1038/s41467-022-31739-w
14. Long Y, Ang KS, Li M. *et al.* Spatially informed clustering, integration, and deconvolution of spatial transcriptomics with graphst. *Nat Commun* 2023;**14**:1155. https://doi.org/10.1038/s41467-023-36796-3
15. Yuan Z, Zhao F, Senlin Lin Y. *et al.* Benchmarking spatial clustering methods with spatially resolved transcriptomics data. *Nat Methods* 2024;**21**:712–22. https://doi.org/10.1038/s41592-024-02215-8
16. Chami I, Ying Z, Ré C. *et al.* Hyperbolic graph convolutional neural networks. *Advances in neural information processing systems* 2019;**32**:4869–80.
17. Liu Q, Nickel M, Kiela D. Hyperbolic graph neural networks. *Advances in neural information processing systems* 2019;**32**:8228–39.

18. Zhang Y, Wang X, Shi C. *et al.* Lorentzian graph convolutional networks. In: Ladisch F, Anand A, Baeza-Yates R. et al. (eds.), *Proceedings of the Web Conference 2021 (WWW'21)*. New York, NY, USA: Association for Computing Machinery; 2021, 249–61.
19. Liu J, Yang M, Zhou M. *et al.* Enhancing hyperbolic graph embeddings via contrastive learning arXiv preprint arXiv:2201.08554. 2022.
20. Vyas A, Choudhary N, Khatir M. *et al.* Graphzoo: A development toolkit for graph neural networks with hyperbolic geometries. In: Ladisch F, Anand A, Baeza-Yates R. et al. (eds.), *Proceedings of the Web Conference 2022(WWW'22)*. New York, NY, USA: Association for Computing Machinery; 2022, 184–8.
21. Yang M, Li Z, Zhou M. *et al.* HiCF: Hyperbolic informative collaborative filtering. In: Reddy CK, Aggarwal CC, Ding Y. et al. (eds.) *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'22)*. New York, NY, USA: Association for Computing Machinery; 2022, 2212–21.
22. Ermolov A, Mirvakhabova L, Khrulkov V. *et al.* Hyperbolic Vision Transformers: Combining improvements in metric learning. In: Khan FS, Fowlkes CC, Maire M. et al. (eds.), *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2022)*. Los Alamitos, CA, USA: IEEE Computer Society; 2022, 7409–19.
23. Klimovskaia A, Lopez-Paz D, Bottou L. *et al.* Poincaré maps for analyzing complex hierarchies in single-cell data. *Nat Commun* 2020;**11**:2966. https://doi.org/10.1038/s41467-020-16822-4
24. Désidéri J-A. Multiple-gradient descent algorithm (mgda) for multiobjective optimization. *Comptes Rendus Mathematique* 2012;**350**:313–8. https://doi.org/10.1016/j.crma.2012.03.014
25. Sener O, Koltun V. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems* 2018;**31**:525–36.
26. Mingxuan J, Zhao T, Wen Q. *et al.* Multi-task self-supervised graph neural networks enable stronger task generalization arXiv preprint arXiv:2210.02016. 2022.
27. Chen A, Liao S, Cheng M. *et al.* Spatiotemporal transcriptomic atlas of mouse organogenesis using dna nanoball-patterned arrays. *Cell* 2022;**185**:1777–1792.e21. https://doi.org/10.1016/j.cell.2022.04.003
28. Yang T, Linmei H, Shi C. *et al.* Hgat: Heterogeneous graph attention networks for semi-supervised short text classification. *ACM Trans Inf Syst* 2021;**39**:1–29. https://doi.org/10.1145/3450352
29. Jaggi M. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In: Dasgupta S, McAllester D. (eds.), *Proceedings of the 30th International Conference on Machine Learning (ICML 2013)*. Brookline, MA, USA: Proceedings of Machine Learning Research; 2013, 427–35.
30. Scrucca L, Michael Fop T, Murphy B. *et al.* Mclust 5: Clustering, classification and density estimation using gaussian finite mixture models. *The R journal* 2016;**8**:289–317.
31. Wang X, Zhu M, Bo D. *et al.* AM-GCN: Adaptive multi-channel graph convolutional networks. In: McAuley J, Krishnapuram B, Ooi BC, Xu A. (eds.), *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD'20)*. New York, NY, USA: Association for Computing Machinery; 2020, 1243–53.
32. Vaswani A, Shazeer N, Parmar N. *et al.* Attention is all you need. *Advances in neural information processing systems* 2017;**30**: 5998–6008.
33. Hu J, Shen L, Sun G. Squeeze-and-Excitation Networks. In: Ferrari V, Hebert M, Sminchisescu C, Weiss Y. (eds.) *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2018)*. Los Alamitos, CA, USA: IEEE Computer Society; 2018, 7132–41.
34. Zhang X, Xu B, Liang Y. *et al.* Price Does Matter! Modeling price and interest preferences in session-based recommendation. In: Murdock V, Nie J-Y, Weikum G. (eds.) *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'22)*. New York, NY, USA: Association for Computing Machinery; 2022, 1684–93.
35. Maynard KR, Collado-Torres L, Weber LM. *et al.* Transcriptome-scale spatial gene expression in the human dorsolateral prefrontal cortex. *Nat Neurosci* 2021;**24**:425–36. https://doi.org/10.1038/s41593-020-00787-0

Briefings in Bioinformatics, 2025, **26(2)**, bbaf162
https://doi.org/10.1093/bib/bbaf162
Problem Solving Protocol