Structure-Preserving and Batch-Correcting Visualization Using Deep Manifold Transformation for Single-cell RNA-Seq Profiles

Yongjie Xu[1,2,3], Zelin Zang[1,2,3], Jun Xia[1,2,3], Cheng Tan[1,2,3], Yulan Geng[1,2], and Stan Z. Li[1,2,*]

[1]AI Lab, School of Engineering, Westlake University, Hangzhou, 310024, China
[2]Institute of Advanced Technology, Westlake Institute for Advanced Study, Hangzhou, 310024, China
[3]Zhejiang University, Hangzhou, 310058, China
[*]stan.zq.li@westlake.edu.cn

ABSTRACT

Dimensionality reduction and visualization play an important role in biological data analysis, such as data interpretation of single-cell RNA sequence (scRNA-seq). It is desired to have a visualization method that can not only be applicable to various application scenarios, including cell clustering and trajectory inference, but also satisfy a variety of technical requirements, especially the ability to preserve inherent structure of data and handle with batch effects. However, no existing methods can accommodate these requirements in a unified framework. In this paper, we propose a general visualization method, deep visualization (DV), that possesses the ability to preserve inherent structure of data and handle batch effects and is applicable to a variety of datasets from different application domains and dataset scales. The method embeds a given dataset into a 2- or 3-dimensional visualization space, with either a Euclidean or hyperbolic metric depending on a specified task type and data type "time-fixed" and "time-evolution" scRNA-seq data, respectively. Specifically, DV learns a semantic graph to describe the relationships between data samples, transforms the data into visualization space while preserving the geometric structure of the data and correcting batch effects in an end-to-end manner. The experimental results on nine datasets in complex tissue from human patients or animal development demonstrate the competitiveness of DV in discovering complex cellular relations, uncovering temporal trajectories, and addressing complex batch factors. We also provide a preliminary attempt to pre-train a DV model for visualization of new incoming data.

Introduction

The advent of technologies that interrogate genome-scale molecular information at single-cell resolution, such as single-cell RNA sequencing and mass cytometry, provides significant insight into the comprehensive analysis of cellular differentiation and the relationship between cells[1]. Although single-cell RNA sequence (scRNA-seq) data have high dimensionality, their intrinsic dimensionality is typically low because many genes are co-expressed and the droplet-based scRNA-seq is very sparse (>90% genes with zero counts in a typical cell profile). Therefore, dimensionality reduction and visualization methods play an important role in interpreting scRNA-seq datasets, such as extracting effective information, intuitively understanding data distribution, and interpreting the relationship between cells[2,3].

At present, it is desired to develop a general dimensionality reduction and visualization method that can not only be applicable to various application scenarios but also satisfy a variety of technical requirements. For example, (1) learning low-dimensional embedding of "time-fixed" scRNA-seq data is to explore the relationship between different cell types[4–13], i.e., cells belonging to the same type move close to each other and cells belonging to different types move away from each other. (2) learning low-dimensional embedding of "time-evolution" scRNA-seq data is to uncover temporal trajectories[13–17], i.e., characterizing the transition process of immature cells into mature cells with specialized types. (3) performing batch correction is to build low-dimensional representations of the biological contents of cells disentangled from the technical variations[13]. (4) building a "batch invariant" model is to embed new incoming data impacted by diverse factors[13]. (5) building a pre-trained model is to embed new incoming heterogeneous data. However, current approaches are inflexible and generally follow uniform assumptions when faced with different application scenarios and can not accommodate these requirements in a unified framework.

Traditional linear/nonlinear dimensionality reduction methods have grown explosively during the last decades, including local geometric structure preservation and global geometric structure preservation methods. The former aims to find a subspace by preserving the local geometry structure such as locally linear embedding (LLE)[4], Laplacian eigenmaps (LE)[5] and stochastic neighbor embedding (SNE)[6]. The latter tries to preserve the global characteristics of raw data in a low dimensional subspace,

such as principal component analysis (PCA)[7], isometric mapping (ISOMAP)[14], diffusion map (DM)[15] and PHATE[18]. These methods are often insufficient to mine underlying biological information, as they only consider global or local structure preservation. In addition, t-distributed stochastic neighbor embedding (t-SNE)[16] and uniform manifold approximation and projection (UMAP)[8] based on manifold learning have demonstrated excellent performance in capturing complex local and global geometry structures of biological data. However, both of them suffer from several limitations. Firstly, t-SNE is not robust in the presence of technical noise and tends to form spurious clusters from randomly distributed data points, producing misleading results that may hinder biological interpretation[13]. Meanwhile, t-SNE preserves the local clustering structures, but the global structures such as inter-cluster relationships and distances cannot be reliably preserved. Secondly, the addition of new data points to existing embeddings is infeasible due to the non-parametric nature of t-SNE and UMAP. Instead, they need to be rerun on the combined dataset, which is computationally expensive and not scalable. Thirdly, the "cell-crowding" problem (e.g., t-SNE), "cell-mixing" (e.g., UMAP) problem, and incompatible batch correction capability also afflict general purpose data visualization when handling large-scale datasets with hundreds of thousands of cells[19].

In recent years, deep neural networks (DNNs)[20] have been utilized as effective non-linear dimensionality reduction tools for processing large datasets, incorporating different factors, and improving the scalable ability of models. There are two mainstream directions, deep manifold learning methods including parametric UMAP[9], Markov-Lipschitz deep learning (MLDL)[10], deep manifold transformation (DMT)[11] and IVIS[12], and deep reconstruction learning methods including various (variational) autoencoders[13,21,22]. Current deep reconstruction learning methods seek to reconstruct the raw data distribution and often ignore the importance of intrinsic geometry structure in raw data. In contrast, deep manifold learning methods preserve the geometry structure of raw data as much as possible, which is beneficial for mining the underlying information of biological data but is not compatible with batch correction ability simultaneously. Therefore, the above methods usually suffer three fundamental issues: (1) High distortion embedding problem. Most methods assume the embedding space is Euclidean, which is not enough for modeling and analyzing "time-evolution" scRNA-seq data because the Euclidean geometry is not optimal for representing the hierarchical and branched developmental trajectories. As shown in Bourgain's theorem, Euclidean space is unable to obtain comparably low distortion for tree data, even when using an unbounded number of dimensions[23]. Poincaré maps (Poin_maps)[17] was proposed recently to harness the power of hyperbolic geometry into the realm of "time-evolution" scRNA-seq data analysis. (2) Deep manifold learning methods can not preserve the geometric structure of the data and correct batch effects in an end-to-end manner. Most methods require multiple separate steps, each with its own method, including batch correction (e.g., Seurat3 CCA[2], Harmony[24], LIGER[25], fastMNN[26], Scanorama[27], SAUCIE[28], scVI[22] and Conos[29]), dimensionality reduction, and visualization. Ding et al.[30] proposed a scalable deep generative model scPhere based on variational autoencoders to embed cells into low-dimensional hyperspherical or hyperbolic spaces to better capture the inherent properties of scRNA-seq data. ScPhere addresses multi-level, complex batch factors, facilitates the interactive visualization of large datasets, resolves "cell-crowding", and uncovers temporal trajectories. (3) Poor flexibility for various application scenarios. Most methods tend to follow a uniform assumption for different application scenarios (e.g., "time-fixed" or "time-evolution" data with or without batch effects) without considering the inherent characteristics of biological data, and the pre-trained reference model can only map new incoming homogeneous data and lacks suitable preprocessing steps to accommodate new incoming heterogeneous data.

To address the above challenges, we propose a general visualization model, deep visualization (DV), that preserves the inherent structure of scRNA-seq data and handles complex batch effects. Specifically, DV learns semantic graphs based on local scale contraction to describe the relationships between cells more accurately, transforms the data into 2- or 3-dimensional embedding space while preserving the geometric structure of the data, and constructs batch graphs to correct batch effects in an end-to-end manner. For "time-fixed" scRNA-seq data, we minimize the structure distortion between semantic graph and visualization graph in Euclidean space (DV_Eu). For "time-evolution" cells, to better represent and infer underlying hierarchical and branched developmental trajectories, we embed cells to the hyperbolic space with Poincaré (DV_Poin) or Lorentz (DV_Lor) model and visualize the embedding in a Poincaré disk. We demonstrate the superior performance of DV on critical existing cases based on nine diverse datasets from human, mouse, and model organisms, including processing large "time-fixed" and "time-evolution" scRNA-seq datasets with/without complex multilevel batch effects, visualizing cell profiles from highly complex tissues and developmental processes. We also make a preliminary attempt to build a pre-trained reference model for visualization of new incoming homogeneous and heterogeneous data. Overall, our method serves as a unified solution for enhanced representation, complex batch correction, visualization, and an interpretation tool for single-cell genomics research.

Results

In order to meet the needs of "time-fixed" and "time-evolution" scRNA-seq data visualization, DV embeds them in latent spaces with different curvatures, respectively, including Euclidean or hyperbolic spaces, which is implied by the end of the DNNs (Fig. 1a). Euclidean space with zero curvature is commonly adopted by most visualization methods (e.g., t-SNE, UMAP, PHATE, and

IVIS) due to its flatness and intuitive class boundaries, which is enough for "time-fixed" data to explore the relationship between different cell types. Hyperbolic space with constant negative curvature has been proposed to learn hierarchical representations from textual and graph-structured data[23]. It is suitable for "time-evolution" data to uncover temporal trajectories because the exponential growth of the hyperbolic surface area with respect to its radius is analogous to the exponential growth of the number of leaves in a tree with respect to its depth, rather than polynomially as in the Euclidean case.

DV model assumes that a good embedding should preserve the geometry structure of scRNA-seq data as much as possible. According to manifold assumption, the observed data are low dimensional manifolds uniformly sampled in a high dimensional Euclidean space. In practice, the droplet-based scRNA-seq data usually has a large number of zeros or near-zero values. The relationship between cells is difficult to be defined by vector similarity (e.g., Euclidean distance) directly for such high-dimensional sparse data. Therefore, DV learns a reliable semantic graph based on local scale contraction between cells and their corresponding augmented cells (linear interpolation between cells and their k neighbor points) to describe the relationship between cells more accurately. Specifically, DV estimates the underlying manifold structure in three main steps (Fig. 1a): Firstly, constructing a fully connected undirected semantic graph $G_{semantic}$ for cells and their corresponding augmentation cells based on semantic embedding learned by semantic module, where each node corresponds to an individual cell and each edge has a weight (Euclidean distance between the semantic embeddings of the two connected cells). The purpose is to estimate the local geometries of the underlying topological manifold (locally Euclidean). Secondly, DV learns a low-dimensional Euclidean or hyperbolic embedding per cell and constructs a fully connected undirected visualization graph $G_{visualization}$ for cells and their corresponding augmentation cells based on visualization embedding learned by the visualization module. In detail, for Euclidean latent spaces, DV learns 2-dimensional embeddings and adopts Euclidean distance to describe the relationship between cells. For hyperbolic latent spaces, DV learns 2 or 3-dimensional embeddings with Poincaré or Lorentz model and adopts hyperbolic distance to describe the relationship between cells. Thirdly, DV converts $G_{semantic}$ and $G_{visualization}$ edge weight from distance to similarity based on the student's t-distribution. The purpose is to highlight similar pairwise nodes and weaken dissimilar pairwise nodes. These steps are commonly used in manifold learning (e.g., t-SNE, UMAP) to approximate the structure of an unknown manifold from similarities in the feature space. Finally, to preserve the semantic geometry structure of scRNA-seq data, DV adopts the geometric structure preservation loss function to train DNNs, which minimizes the distribution discrepancy between $G_{semantic}$ and $G_{visualization}$. To make DV compatible with the batch correction ability simultaneously, we integrate the manually designed batch graph G_{batch} into the $G_{visualization}$ to be learned in the training process to learn the $G_{visualization}$ with removing batch effects. As a deep learning model trained by mini-batch stochastic gradient descent, DV is especially suited to process large scRNA-seq datasets with *complex multilevel batch effects and facilitates emerging applications. We provide complete details in the "Methods" section.

DV visualizes large datasets with multiple cell types and preserves the structure of scRNA-seq data in very low-dimensional spaces.

Applying DV to scRNA-seq data, we systematically assess the visualization performance of DV embedding in a latent space with few (2, 3) dimensions. We compare the geometric structure preservation performance (Q_{global} and Q_{local} scores, interpreted as a scRNA-seq dataset comprises a smooth manifold and a good dimensionality reduction method would preserve local and global distances on this manifold) of DV, which embeds cells in Euclidean or hyperbolic spaces, as well as to PCA[7], t-SNE[16], UMAP[8], IVIS[12], PHATE[18], Poin_maps[17] and hyperbolic scPhere (scPhere_wn)[30]. Following scPhere[30], we apply DV to seven scRNA-seq datasets from human and mouse, spanning from small (thousands) to very large number (hundreds of thousands) of cells from one or multiple tissues, and with a small (two) to very large (dozens) of expected cell types. The "small" datasets are: (1) a blood cell dataset[31] with only 10 erythroid cell profiles and 2293 CD14+ monocytes, (2) 3314 human lung cells[32], (3) 1378 mouse white adipose tissue stromal cells[33], and (4) 1755 human splenic nature killer cells spanning four subtypes[34]. The "large" datasets are: (1) 35,699 retinal ganglion cells (RGC) in 45 cell subsets[35], (2) 599,926 cells spanning 102 subsets across 59 human tissues in the Human Cell Landscape (HCL)[36], and (3) 86,024 C. elegans embryonic cells (ELEGAN) collected along a time course from <100 min to >650 min after each embryo's first cleavage[37].

DV obtains more competitive performance compared with baseline methods on "small" datasets (Supplementary Fig. 1 and Fig. 2). It is worth noting that DV_Poin and DV_Lor with hyperbolic latent spaces even perform better than DV_Eu in some datasets, although there are discrete cell types in these datasets. Overall, these methods achieve good visualizations for these smaller datasets without batch effects but with minor challenges. For example, (1) in adipose tissue stromal cells, all methods mixe pericyte and APC except DV_Lor (Supplementary Fig. 1m). (2) PHATE (Supplementary Figs. 2f, p) and Poin_maps (Supplementary Figs. 1i, s) — are designed for development trajectories—connecting cells inaccurately when only discrete cell types are present. (3) due to the lack of nonlinear ability, PCA cannot effectively capture complex data structures, resulting in the distortion issue with the increased number of cell types (Supplementary Fig. 2d). (4) compared with the dimensionality reduction methods based on geometric structure preservation (e.g., DV, UMAP, and t-SNE), scPhere_wn (Supplementary Figs. 2j, t) based on variational auto-encoder does not have enough power to aggregate similar samples (clusters) and push away

dissimilar samples (clusters).

DV obtains significant advantages over baseline methods on datasets with a larger number of cells and clusters, such as mouse RGC cells and human HCL cells. As shown in Fig. 2, although DV (with either Euclidean or hyperbolic latent space), Poin_maps, UMAP and t-SNE can distinguish individual cell types among RGC (Figs. 2a-h, Supplementary Figs. 3a-h) and HCL (Figs. 2i-q, Supplementary Figs. 3i-q) well, DV achieves superior local and global structure preservation performance (Figs. 2r, s). In contrast, DV_Poin and DV_Lor are based on hyperbolic latent spaces and can obtain better global structure preservation performance than DBMC_Eu based on Euclidean latent space in RGC cells, especially DV_Poin based on the Poincaré ball model. Specifically, Cartpt-RGC clusters are close together in DV_Poin hyperbolic latent space (Fig. 2j), but in different locations in the latent space of other methods except for Poin_maps (Fig. 2q), this shows that hyperbolic latent space can better preserve the hierarchical global structure of cells. In HCL cells, there are six major cell groups (fetal stromal cells, fetal epithelial cells, adult endothelial cells, endothelial cells, adult stromal cells, and immune cells), and their respective clusters are close to each other in DV_Eu (Fig. 2a), but are more dispersed in t-SNE embedding (Fig. 2g). HCL cells contain two cell sources (adult cells and fetal cells), their respective clusters are close to each other and have better differentiation between both sources in DBMC_Eu (Fig. 2a) but are more mixed in UMAP embedding (Fig. 2h). In contrast, DV_Poin (Fig. 2b) and DV_Lor (Fig. 2c) have larger volume for structure storage, thus gaining more power to push away dissimilar clusters, which will help in data analysis without labels (cell type), such as visualization partition analysis (e.g., DV_Poin hyperbolic latent space contains six major regions). Compared with DV_Eu, some clusters (enterocyte cells have an immune function) belonging to adult endothelial cells are close to immune cells (mostly B cells). Meanwhile, UMAP and t-SNE are often unable to effectively visualize datasets with a larger number of cells as reflected both visually. For example, in the "cell-crowding" problem in t-SNE (Fig. 2g), different clusters are uniform spread across the embedded space, leading to the inability to efficiently recognize the major clusters and effectively mine the distinct cell types. The "cell-mixing" problem in UMAP (Fig. 2h), different clusters are twisted and mixed. DV overcome the above issues because it learns a more reliable semantic graph $G_{semantic}$ based on nonlinear DNNs and is trained using mini-batches, while t-SNE and UMAP are learned using all the data and their hyperparameters (e.g., "perplexity" in t-SNE) have to be adapted to this larger number of cells, but increasing the "perplexity" makes t-SNE computationally expensive. DV is natural to process a large number of cells, with a time complexity that is linear with the number of input cells. As expected, PCA (Figs. 2d, l), IVIS (Figs. 2e, m) and PHATE (Figs. 2f, n) did not perform well for these large datasets of mostly discrete cell types.

DV obtains more intuitive visualization on "time-evolution" cells, which are expected to show developmental trajectories, such as from stem cells to mature cells (Figs. 3j-r), and explain the relationships between clusters (Figs. 3a-i). As described above, DBMC_Poin (Fig. 3b) and DBMC_Lor (Fig. 3c) have larger volumes for hierarchical structure storage, thus the dataset can be divided into multiple regions to facilitate the analysis of cell differentiation for each major cell type individually. For example, there are four major regions (muscle cells, excretory cells, pharyngeal cells, neuron cells, and other individual clusters) in DBMC_Poin latent space if not considering NA cells. Compared with DV, data in the central region of baseline methods are faced with a severe "cell-crowding" issue (Figs. 3m-r), which is not conducive to a more accurate recognition of cell origin. However, this issue is well solved in DV, where the data in the central region can be clearly identified and assigned to different differentiation branches (Figs. 3 j-l). Moreover, we can position the expected root cells of the developmental process at the center of a Poincaré disk, then the distance of each cell from the center can be thought of as a pseudo time. For a specific cell type, we can see cells progress with distance continuously in the Poincaré disk at almost a fixed angle. Fortunately, compared with scPhere_wn (Fig. 3r), DV (Figs. 3j-l) can automatically place root cells near the center of the Poincaré disk without prior knowledge of the root cells, which is convenient for data analysis when unknown time labels. Furthermore, DBMC_Poin (Fig. 3b) and DBMC_Lor (Fig. 3c) alleviate the "cell-mixing" problem (e.g., seam cells and hypodermis cells are entangled) existing in DV_Eu (Fig. 3a) and UMAP (Fig. 3h). Therefore, DV embeds "time-evolution" cells into a hyperbolic space with Lorentz or Poincaré model, which is suitable for differentiated data analysis, and optionally converts the coordinates in the Lorentz model to the Poincaré disk for 2D visualization. Specifically, in ELEGAN cells, we find that cells are ordered neatly in the latent space by both time and lineage, from a clearly discernible root at time 100–130 at the center of the Poincaré disk (cells from <100 were mostly unfertilized germline cells) to cells from time >650 near the border of the Poincaré disk or away from the origin in the Poincaré and Lorentz model (Figs. 3k, l, Supplementary Fig. 12). Within the same cell type, cells are ordered by embryo time in the Poincaré disk (Figs. 3b, c). After first appearing along a developmental trajectory, cells of the same type progress with embryo time, forming a continuous trajectory occupying a range of angles. Moreover, different cell types (e.g., ciliated amphid neurons, ciliated nonamphid neurons, hypodermis, seam cells, and body wall muscle) that appear at slightly different embryonic time points, have their origins around the same region and progress with embryonic time in a similar way, forming a continuous trajectory but at a different angle and/or distance ranges from the center. These patterns are harder to discern in IVIS (Fig. 3n) and PHATE (Fig. 3o), t-SNE (Fig. 3p) and UMAP (Fig. 3q), where cells from consecutive time points are compacted, cells that appear early are relatively distant from each other in the embeddings, and temporal progression is not in the same direction. Thus, the DV model with a hyperbolic latent space learns smooth (in time) and interpretable cell trajectories.

DV effectively models complex, multilevel batch, and other variables.

In realistic biological datasets, scRNA-seq profiles are typically impacted by diverse factors, including technical batch effects in separate experiments and different lab protocols, as well as biological factors, such as inter-individual variation, sex, disease, or tissue location. However, most batch-correction methods can handle only one batch variable and may not be well-suited to the increasing complexity of current datasets. Applying DV to scRNA-seq data with multiple known confounding factors (e.g., batches and conditions), we systematically assess the performance of DV embedding in a latent space with few (2, 3) or low (5, 10, 20) dimensions by comparing the geometric structure preservation performance (Q_{global} and Q_{local} scores, interpreted as a complex multi-batch scRNA-seq dataset comprises multiple smooth manifold and a good dimensionality reduction method would preserve local and global distances on each manifold after removing batch effect) and classification performance (ACC_{mvo} score, interpreted as a good batch correction method would integrate different manifolds) of DV, which embeds cells in Euclidean space for "time-fixed" cells and hyperbolic space for "time-evolution" cells , as well as to Euclidean scPhere (scPhere_normal), hyperspherical scPhere (scPhere_vmf), scPhere_wn and other visualization methods, including t-SNE, UMAP, IVIS and PHATE (with 50 principal components, batch-corrected by Harmony). Following scPhere[30], we apply DV to a dataset of 301,749 cells profiled in a complex experimental design from the colon mucosa of 18 patients with ulcerative colitis (UC), a major type of inflammatory bowel disease (IBD), and 12 healthy individuals[38]. The "time-fixed" datasets are: (1) 26,678 stromal cells and glia (12 cell types), and (2) 210,614 immune cells (23 cell types). The "time-evolution" dataset are: (1) 64,457 epithelial cells (12 cell types), and (2) 86,024 ELEGAN cells (12 time states).

DV_Eu obtains more competitive performance compared with baseline methods on "time-fixed" datasets with a larger number of cells and multiple confounding factors, such as stromal (30 patients with patient origin and disease status factors) and immune (30 patients with patient origin, disease status, and location factors). Analyzing cells with the patient origin and disease status (healthy, uninflamed, and inflamed) as the batch vector, not only recapitulate the main cell groups, but also allow us to better visually explore cellular relations (Supplementary Figs. 4a-i) and find cell groups related to disease directly (Supplementary Fig. 4k). For example, in the stromal dataset, the postcapillary venules cells, endothelial cells, and microvascular cells are close to each other, and adjacent to the pericyte (Supplementary Fig. 4a). Conversely, these distinctions can barely be discerned in a UMAP (Supplementary Fig. 4f) and Poin_maps (Supplementary Fig. 4g) plot of the same data, where endothelial and microvascular cells are very close. Among fibroblasts, cells are arranged in a manner that mirrors their position along the crypt–villus axis, from RSPO3+ cells, to WNT2B+ cells, to WNT5B+ cells. Strikingly, the inflammatory fibroblasts, which are unique to UC patients (Supplementary Fig. 4k) and are independent of the patient origin (Supplementary Fig. 4j), are readily visible (Supplementary Fig. 4a, pale green) and are both distinct from the other fibroblasts, while spanning the range of the "crypt–villus axis". Considering the ability to integrate disease status, DV_Eu merges part inflammatory fibroblasts with WNT2B+ fibroblasts (Supplementary Fig. 4a). When learning a DV_Eu model that includes patient origin, disease status, and anatomical region as the batch vectors, immune cells grouped visually by type (Fig. 4a), and the influence of patient origin (Fig. 4i), disease status (Fig. 4j) and region (Fig. 4k) is largely removed. For example, the CD8+IL17+ T cells are nestled between CD8+ T cells and activated CD4+ T cells in a manner that was intriguing and consistent with the mixed features of those cells (Fig. 4a). On the one hand, DV_Eu (batch correction on 2-dimensional embedding) obtains competitive local (Q_{local} score) and global (Q_{global} score) geometric structure preservation performance compared with UMAP (batch correction on 50 principal components) and outperform scPhere_normal and t-SNE methods on the stromal dataset (Supplementary Figs. 4l, m) and immune dataset (Figs. 4l, m). Moreover, DV_Eu obtains better performance when batch correction on low (5, 10, or 20) dimensional embedding. On the other hand, DV_Eu obtains competitive batch correction visualization performance (ACC_{mvo} score on 2-dimensional embedding) on the stromal dataset (Supplementary Fig. 4n) and immune dataset (Fig. 4n) compared with IVIS, PHATE, t-SNE, UMAP, Poin_maps and scPhere visualization methods with Harmony batch correction method based on 5-nearest neighbors (5-NN) classification accuracy of cell types. Moreover, DV_Eu obtains a significant batch correction advantage (ACC_{mvo} score on 5, 10, or 20-dimensional embedding) over scPhere method on low dimensional embedding (Fig. 4o, Supplementary Fig. 4o).

DV_Poin and DV_Lor obtain more intuitive visualization and competitive performance compared with baseline methods on "time-evolution" datasets, such as epithelial (30 patients with patient origin, disease status, and location factors) cells and ELEGAN cells (7 batches). In epithelial cells, we readily discern developmental ordering from intestinal stem cells to terminally differentiated cells in the Poincaré disk (Fig. 5b, c), with stem cells at the center of the disk for intuitive interpretation: one trajectory is from stem cells to secretory TA cells, to immature goblet cells, to goblet cells; the other trajectory is from stem cells to TA2 cells, to immature enterocyte cells, to enterocyte cells. In contrast, developmental trajectories are less apparent when we embed cells in Euclidean space (Fig. 5f). 2-dimensional visualization with t-SNE (Fig. 5g), UMAP (Fig. 5h), DV_Eu (Fig. 5a), IVIS (Fig. 5d), PHATE (Fig. 5e), scPhere_wn (Fig. 5j) and scPhere_vmf (Fig. 5i) is reasonable, although the t-SNE had some small spurious clusters, goblet cells had one spurious cluster close to enterocytes in DV_Eu, IVIS, scPhere_wn and scPhere_vmf, several cell types are merged (M-cells and TA2 cells, tuft and enteroendocrine cells) in PHATE, and developmental trajectories are less apparent when the cell types are missing in DV_Eu, scPhere_vmf and scPhere_wn.

In addition, DV_Poin and DV_Lor obtain a significant local (Fig. 5n) and global (Fig. 5o) geometric structure preservation advantage compared with baseline methods in Euclidean space, and this advantage increases significantly when the dimension of embedding space is increased. In ELEGAN cells, the cellular relations and developmental trajectories in DV and scPhere_wn have minor changes when compared Supplementary Fig. 5 (with batch correction) with Fig. 3 (without batch correction), which indicates that the batch effect has little influence on this dataset, but the former still mitigate the less apparent batch effect problem existing in some cell types (e.g., barpaaa lineage, ciliated nonamphid neurons, and ciliated non nonamphid neurons). Moreover, the advantages in this dataset described in the previous section of DV_Poin and DV_Lor (Supplementary Fig. 13) still remain when compared with t-SNE, UMAP, and scPhere (Supplementary Fig. 14).

DV_Eu achieves impressive results on all stromal, epithelial, and immune cells simultaneously (Supplementary Fig. 6a), demonstrating its capacity to embed large numbers of cells of diverse types, states, and proportions. However, using t-SNE (Supplementary Fig. 6g) or UMAP (Supplementary Fig. 6h) with Harmony batch-corrected results of raw data as input leads to an unsuccessful visualization, while DV_Eu (Supplementary Fig. 6f) can achieve competitive performance compared with scPhere_wn (Supplementary Fig. 6e) and scPhere_vmf (Supplementary Fig. 6i). For example, there are a lot of noise points existing in t-SNE embedding, which illustrates that t-SNE can not distinguish different cell types, and UMAP suffers from a severe confusion issue among different clusters. Moreover, this also verifies that the Harmony method can not effectively remove the batch effect problem in sparse data. For example, Plasma cells are separated in UMAP embedding. Overall, these results demonstrate the superior performance of DV_Eu compared to the combination of Harmony's batch correction and t-SNE or UMAP's visualization through multiple experiments on large datasets with a large number of cells and cell types, multilevel batch effects, and complex structures.

Pre-trained reference DV builds atlases for visualization and annotation of new incoming data.

As a parametric model, we can train DV to co-embed new incoming (test) data to a latent space learned from training data only. We use DV to map cells from new incoming patients, one critical homogeneous application case as multiple studies need to be integrated by training a "batch-invariant" DV model, the other critical heterogeneous application case as a new study can be explored by a pre-trained DV model. Then, DV takes the gene expression vectors or PCA principal components of cells as inputs and maps them to a 2-dimensional Euclidean latent space to achieve data visualization and annotation. Therefore, we conduct preliminary experiments to explore the scalable ability of DV to train general models based on large-scale datasets.

For homogeneous case (training data and test data share the same gene name and number), we learn a "batch-invariant" DV_Eu model for stromal, epithelial, and immune cells from 18 patients training data of the UC dataset and use it to visualize the cells from 12 patients test data directly. Then we train k-nearest neighbor (k-NN) classifiers ($k = 5$) on 2-dimensional representations of the training data and apply the learned k-NN classifiers to 2-dimensional representations of the test data. The experiment results demonstrate that DV's embedding of test data with high quality. For the stromal dataset (Fig. 6), DV_Eu (72.84%) obtains better classification accuracy compared with scPhere_normal (71.25%), scPhere_wn (70.25%) and scPhere_vmf (70.24%), it significantly improves the precision of most categories, especially on inflammatory fibroblasts cells, while the precision of RSPO3+ is unsatisfactory. For the immune dataset (Supplementary Fig. 7), DV_Eu (82.17%) outperform scPhere_normal (78.23%), scPhere_wn (79.37%), and scPhere_vmf (78.58%) in terms of classification accuracy, especially on DC1 cells, DC2 cells, macrophages and cycling monocytes, while the precision decrease in inflammatory monocytes, ILC cells, and DC8+IEL cells. For epithelial cells (Supplementary Fig. 8), DV and scPhere achieve similar classification accuracy, but DV possesses a better visualization ability to show two main developmental trajectories.

For heterogeneous case (training data and test data share different gene names and numbers), we design a series of critical preprocessing methods (Fig. 1c) to overcome the heterogeneous problem, including a heterogeneous correction module (the same genes in training data and test data are selected, for test data, the same genes are maintained as original value, missing genes are set to 0, and redundant genes are removed), normalization, log scaling, standardization module (mean and standard deviation learned on training data is used to scale test data), PCA module (PCA model learned on training data is used to map test data as 50 principal components). We learn a pre-trained DV_Eu model for HCL cells from 43 tissues training data, then use it to map the HCL cells from 28 tissues and mouse cell atlas cells (MCA) cells test data. The experiment results demonstrate that DV's embedding of test data with high quality even in heterogeneous cases. For example, the underlying biological information analysis in the HCL dataset mentioned above still remains, DV_Eu obtains 72.82% classification accuracy (Supplementary Fig. 9c, only assesses the cell types presented in the training set) when data is collected in the same species and profiled by the same platform (e.g., Microwell-Seq platform). Moreover, the pre-trained DV can still locate some important clusters (e.g., Erythroid cells, Macrophage, Monocyte, B cells, and Fasciculata cells, Supplementary Fig. 9d) corresponding to the training dataset when conducting cross-species experiments (e.g., training on HCL and testing on MCA). This verifies the analysis of the original study[36] that the major cell types in mammalian organs are similar.

Discussion

We propose the DV model to embed "time-fixed" and "time-evolution" scRNA-seq cells in low-dimensional Euclidean and hyperbolic spaces to enhance exploratory data analysis and visualization of cells from single-cell studies, especially with complex multilevel batch factors. DV provides more readily interpretable representations and avoids "cell-crowding" and "cell-mixing" problems. When embedding "time-evolution" cells in hyperbolic spaces, it helps to study developmental trajectories. In this case, DV_Poin and DV_Lor can place root cells near the center of a Poincaré disk automatically (distance to the center can be used as a natural definition for pseudo time). They can divide cells into multiple regions to facilitate the analysis of cell differentiation for each major cell type individually (the cells of specified type progress continuously with distance and angle in the Poincaré disk).

The main advantages of DV are in achieving geometric structure preservation of scRNA-seq data and accounting for multilevel complex batch effects simultaneously, which disentangles cell types from patients, diseases, and location variables. This is an important advantage over other DNNs methods, which fail to combine geometric structure preservation with batch correction capability. We can harness these abilities in several ways to meet different task requirements based on the inherent characteristics of biological data: to visualize "time-fixed" or "time-evolution" cells directly when dataset without batch effects, to visualize "time-fixed" or "time-evolution" cells considering one factor or combination of them for dataset with multilevel complex batch effects, to investigate which cell types are most affected by a factor or to generate universal reference embeddings, which can map new incoming homogeneous or heterogeneous data to an existing embedding and annotate cell types. DV's ability to handle complex batch factors is an advantage over previous methods for batch correction, which handle only one batch vector. Indeed, in our benchmarking with IBD cells with 30 patients with three disease statuses, DV performs better than state-of-the-art batch correction methods such as Harmony and scPhere. In the future, we can leverage supervised information to construct a more reliable batch graph. In addition, as a parametric model, DV can naturally co-embed homogeneous or heterogeneous new incoming (test) data to a latent space learned from training data only and denoise expression data successfully.

DV is especially suitable for analyzing large scRNA-seq datasets: its running time scales linearly with the number of input cells, it alleviates "cell-crowding" and "cell-mixing" issues when handling with large numbers of input cells, and it can preserve local and hierarchical global geometric structures in data better than baseline methods. Finally, by learning a "batch-invariant" model that takes gene expressions or principal components as inputs to learn latent embeddings, it forms a reference to visualize and annotate new profiled cells from future studies. This is an important advantage over nonparametric methods such as t-SNE, UMAP, PHATE, and Poincaré maps, which do not have the ability to embed new data, especially in the presence of batch effects.

DV converges rapidly and is robust to hyperparameters. For DV models with different latent spaces (Euclidean and hyperbolic), training is quite stable and converges rapidly. DV complete fitting with 100 epochs, and its morphology is consistent with that of 300 epochs (Supplementary Figs. 10a-j, k-t). Moreover, we adjust the hyperparameters according to the proposed typical value ranges. It can be observed that the influence of hyperparameters is limited (Supplementary Fig. 11a-f, g-m). Even if the visualization results change, it does not affect the underlying biological significance in "time-evolution" scRNA-seq dataset (Supplementary Fig. 12a-g, h-n).

DV can be extended in several other ways. When cell type annotations or cell type marker genes for some of the analyzed cells are available, we can include semi-supervised learning to annotate cell types. Although DV shows promising denoising results, further studies are required to explore its abilities to impute missing counts in scRNA-seq data and remove ambient RNA contamination. Given the rapid development of spatial transcriptomics, single-cell ATACseq, and other complementary measurements, DV can be extended for the integrative analysis of multimodal data. DV can also learn discrete hierarchical trees for better interpreting developmental trajectories using hyperbolic neural networks. Given its scope, flexibility, and extensibility, we foresee that DV will be a valuable tool for large-scale single-cell and spatial genomics studies.

References

1. Stegle, O., Teichmann, S. A. & Marioni, J. C. Computational and analytical challenges in single-cell transcriptomics. *Nat. Rev. Genet.* **16**, 133–145 (2015).

2. Butler, A., Hoffman, P., Smibert, P., Papalexi, E. & Satija, R. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nat. biotechnology* **36**, 411–420 (2018).

3. Luecken, M. D. & Theis, F. J. Current best practices in single-cell rna-seq analysis: a tutorial. *Mol. systems biology* **15**, e8746 (2019).

4. Roweis, S. T. & Saul, L. K. Nonlinear dimensionality reduction by locally linear embedding. *Science* **290**, 2323–2326 (2000).

5. Belkin, M. & Niyogi, P. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation* **15**, 1373–1396 (2003).
6. Hinton, G. & Roweis, S. T. Stochastic neighbor embedding. In *NIPS*, vol. 15, 833–840 (Citeseer, 2002).
7. Wold, S., Esbensen, K. & Geladi, P. Principal component analysis. *Chemom. intelligent laboratory systems* **2**, 37–52 (1987).
8. Becht, E. *et al.* Dimensionality reduction for visualizing single-cell data using umap. *Nat. biotechnology* **37**, 38–44 (2019).
9. Sainburg, T., McInnes, L. & Gentner, T. Q. Parametric umap embeddings for representation and semisupervised learning. *Neural Comput.* **33**, 2881–2907 (2021).
10. Li, S. Z., Zang, Z. & Wu, L. Markov-lipschitz deep learning. *arXiv preprint arXiv:2006.08256* (2020).
11. Li, S. Z., Zang, Z. & Wu, L. Deep manifold transformation for dimension reduction. *arXiv preprint arXiv:2010.14831* (2020).
12. Szubert, B., Cole, J. E., Monaco, C. & Drozdov, I. Structure-preserving visualisation of high dimensional single-cell datasets. *Sci. reports* **9**, 1–10 (2019).
13. Ding, J., Condon, A. & Shah, S. P. Interpretable dimensionality reduction of single cell transcriptome data with deep generative models. *Nat. communications* **9**, 1–13 (2018).
14. Tenenbaum, J. B., De Silva, V. & Langford, J. C. A global geometric framework for nonlinear dimensionality reduction. *Science* **290**, 2319–2323 (2000).
15. Coifman, R. R. & Lafon, S. Diffusion maps. *Appl. computational harmonic analysis* **21**, 5–30 (2006).
16. Van der Maaten, L. & Hinton, G. Visualizing data using t-sne. *J. machine learning research* **9** (2008).
17. Klimovskaia, A., Lopez-Paz, D., Bottou, L. & Nickel, M. Poincaré maps for analyzing complex hierarchies in single-cell data. *Nat. communications* **11**, 1–9 (2020).
18. Moon, K. R. *et al.* Visualizing structure and transitions in high-dimensional biological data. *Nat. biotechnology* **37**, 1482–1492 (2019).
19. Kobak, D. & Berens, P. The art of using t-sne for single-cell transcriptomics. *Nat. communications* **10**, 1–14 (2019).
20. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).
21. Wang, D. & Gu, J. Vasc: dimension reduction and visualization of single-cell rna-seq data by deep variational autoencoder. *Genomics, proteomics & bioinformatics* **16**, 320–331 (2018).
22. Lopez, R., Regier, J., Cole, M. B., Jordan, M. I. & Yosef, N. Deep generative modeling for single-cell transcriptomics. *Nat. methods* **15**, 1053–1058 (2018).
23. Peng, W., Varanka, T., Mostafa, A., Shi, H. & Zhao, G. Hyperbolic deep neural networks: A survey. *IEEE Transactions on Pattern Analysis Mach. Intell.* (2021).
24. Korsunsky, I. *et al.* Fast, sensitive and accurate integration of single-cell data with harmony. *Nat. methods* **16**, 1289–1296 (2019).
25. Welch, J. D. *et al.* Single-cell multi-omic integration compares and contrasts features of brain cell identity. *Cell* **177**, 1873–1887 (2019).
26. Haghverdi, L., Lun, A. T., Morgan, M. D. & Marioni, J. C. Batch effects in single-cell rna-sequencing data are corrected by matching mutual nearest neighbors. *Nat. biotechnology* **36**, 421–427 (2018).
27. Hie, B., Bryson, B. & Berger, B. Efficient integration of heterogeneous single-cell transcriptomes using scanorama. *Nat. biotechnology* **37**, 685–691 (2019).
28. Amodio, M. *et al.* Exploring single-cell data with deep multitasking neural networks. *Nat. methods* **16**, 1139–1145 (2019).
29. Barkas, N. *et al.* Joint analysis of heterogeneous single-cell rna-seq dataset collections. *Nat. methods* **16**, 695–698 (2019).
30. Ding, J. & Regev, A. Deep generative model embedding of single-cell rna-seq profiles on hyperspheres and hyperbolic spaces. *Nat. communications* **12**, 1–17 (2021).
31. Stoeckius, M. *et al.* Simultaneous epitope and transcriptome measurement in single cells. *Nat. methods* **14**, 865–868 (2017).
32. Vieira Braga, F. A. *et al.* A cellular census of human lungs identifies novel cell states in health and in asthma. *Nat. medicine* **25**, 1153–1163 (2019).

33. Hepler, C. *et al.* Identification of functionally distinct fibro-inflammatory and adipogenic stromal subpopulations in visceral adipose tissue of adult mice. *Elife* **7**, e39636 (2018).

34. Crinier, A. *et al.* High-dimensional single-cell analysis identifies organ-specific signatures and conserved nk cell subsets in humans and mice. *Immunity* **49**, 971–986 (2018).

35. Tran, N. M. *et al.* Single-cell profiles of retinal ganglion cells differing in resilience to injury reveal neuroprotective genes. *Neuron* **104**, 1039–1055 (2019).

36. Han, X. *et al.* Construction of a human cell landscape at single-cell level. *Nature* **581**, 303–309 (2020).

37. Packer, J. S. *et al.* A lineage-resolved molecular atlas of c. elegans embryogenesis at single-cell resolution. *Science* **365**, eaax1971 (2019).

38. Smillie, C. S. *et al.* Intra-and inter-cellular rewiring of the human colon during ulcerative colitis. *Cell* **178**, 714–730 (2019).

39. Pedregosa, F. *et al.* Scikit-learn: Machine learning in python. *J. machine Learn. research* **12**, 2825–2830 (2011).

40. Wolf, F. A., Angerer, P. & Theis, F. J. Scanpy: large-scale single-cell gene expression data analysis. *Genome biology* **19**, 1–5 (2018).

41. Lee, J. A. & Verleysen, M. Scale-independent quality criteria for dimensionality reduction. *Pattern Recognit. Lett.* **31**, 2248–2257 (2010).

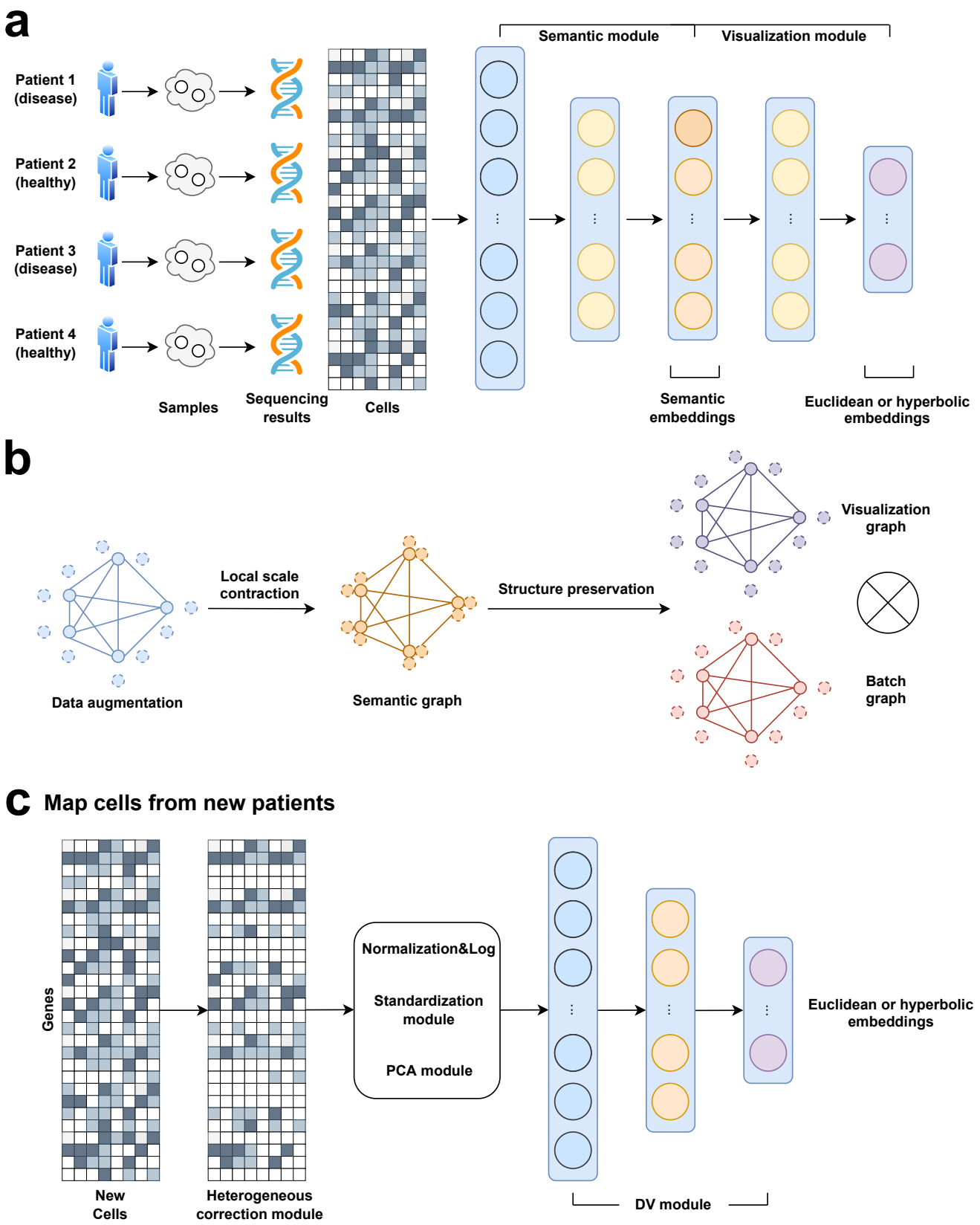

Figure 1. DV model. a Method framework. DV takes as input scRNA-seq measurements with multilevel technical or biological factors (e.g., replicate patient, disease) and learns the latent structure of cells accounting for batch effect problem. **b** DV learns semantic graph based on local scale contraction, merges batch graph with visualization graph, and preserves the geometric structure of scRNA-seq data. **c** The preprocessing modules for heterogeneous new datasets and the learned DV model are used for mapping new datasets to the reference.

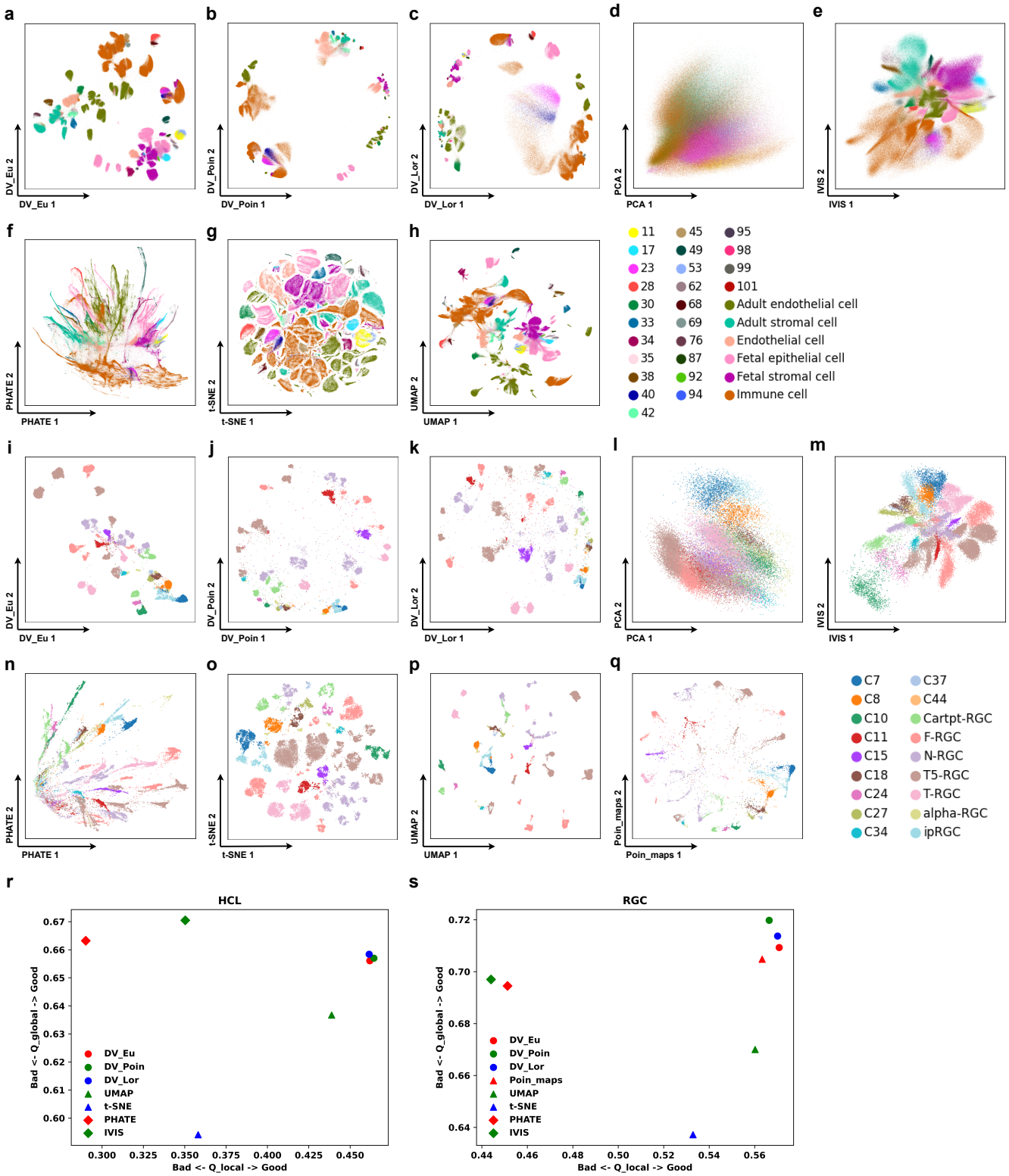

Figure 2. DV preserves local and global structures in visualizing large "time-fixed" scRNA-seq data. a-q DV learns latent representations that provide excellent visualization of local and global structure, even in very large datasets. DV_Eu learns representations in the Euclidean space (**a, i**). DV_Poin learns representations in the hyperbolic space with Poincaré model (**b, j**). DV_Lor learns representations in the hyperbolic space with Lorentz model (**c, k**). 2-dimensional PCA (**d, l**), IVIS (**e, m**), PHATE (**f, n**), t-SNE (**g, o**), UMAP (**h, p**) and Poin_maps (**q**) for human cell landscape (HCL) (**a-h**) and 45 mouse retinal ganglion cells (RGC) (**i-q**) with cells colored by major cell types. **r-s** Quantify global/hierarchical and local structure preservation performance of HCL and RGC datasets.

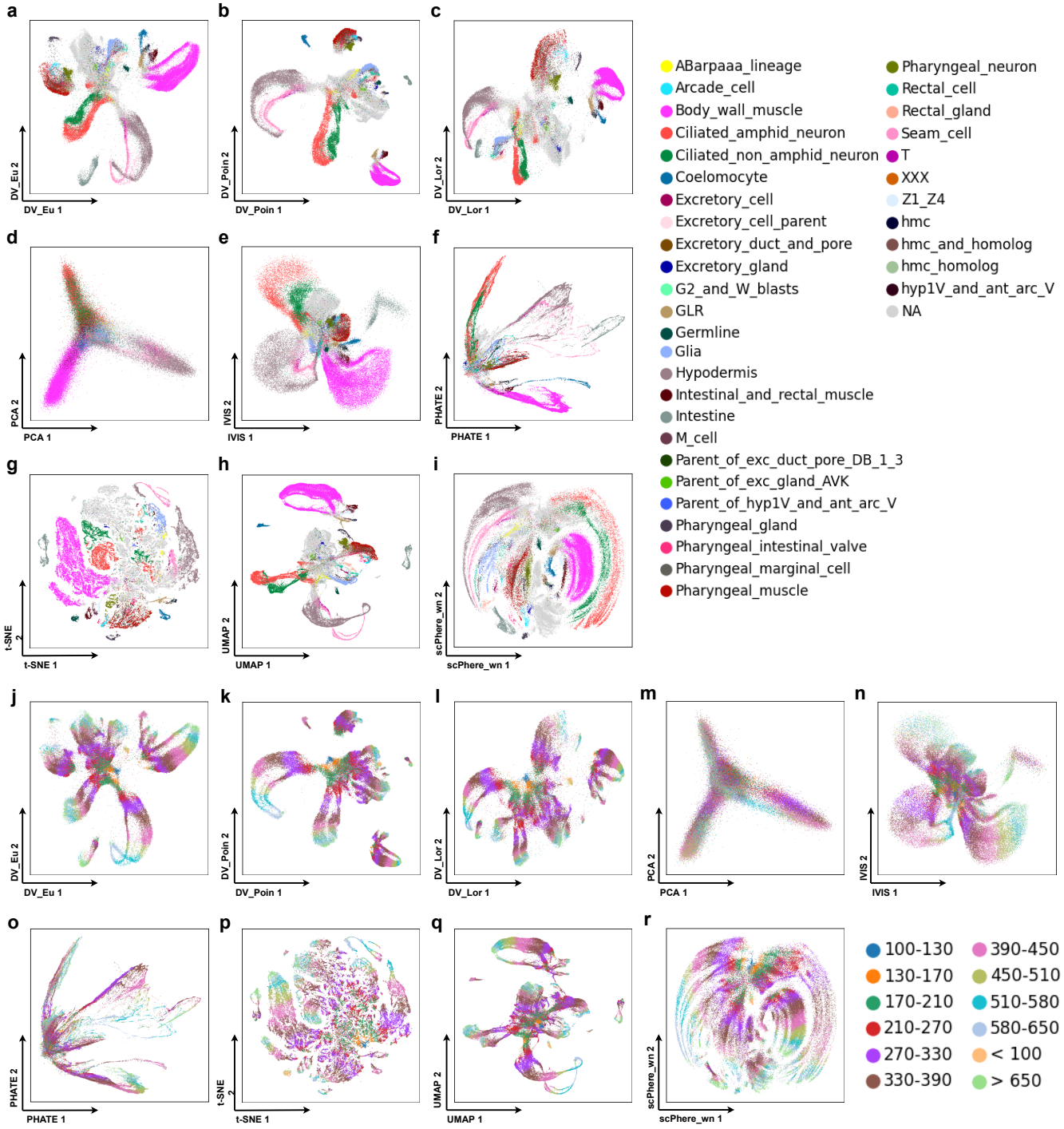

Figure 3. DV shows developmental trajectories in visualizing large scRNA-seq "time-evolution" data. DV_Eu learns representations in the Euclidean space (**a, j**). DV_Poin learns representations in the hyperbolic space with Poincaré model (**b, k**). DV_Lor learns representations in the hyperbolic space with Lorentz model (**c, l**). 2-dimensional PCA (**d, m**), IVIS (**e, n**), PHATE (**f, o**), t-SNE (**g, p**), UMAP (**h, q**) and scPhere_wn (**i, r**) for ELEGAN with cells colored by cell types (**a-i**) and time phases (**j-r**).

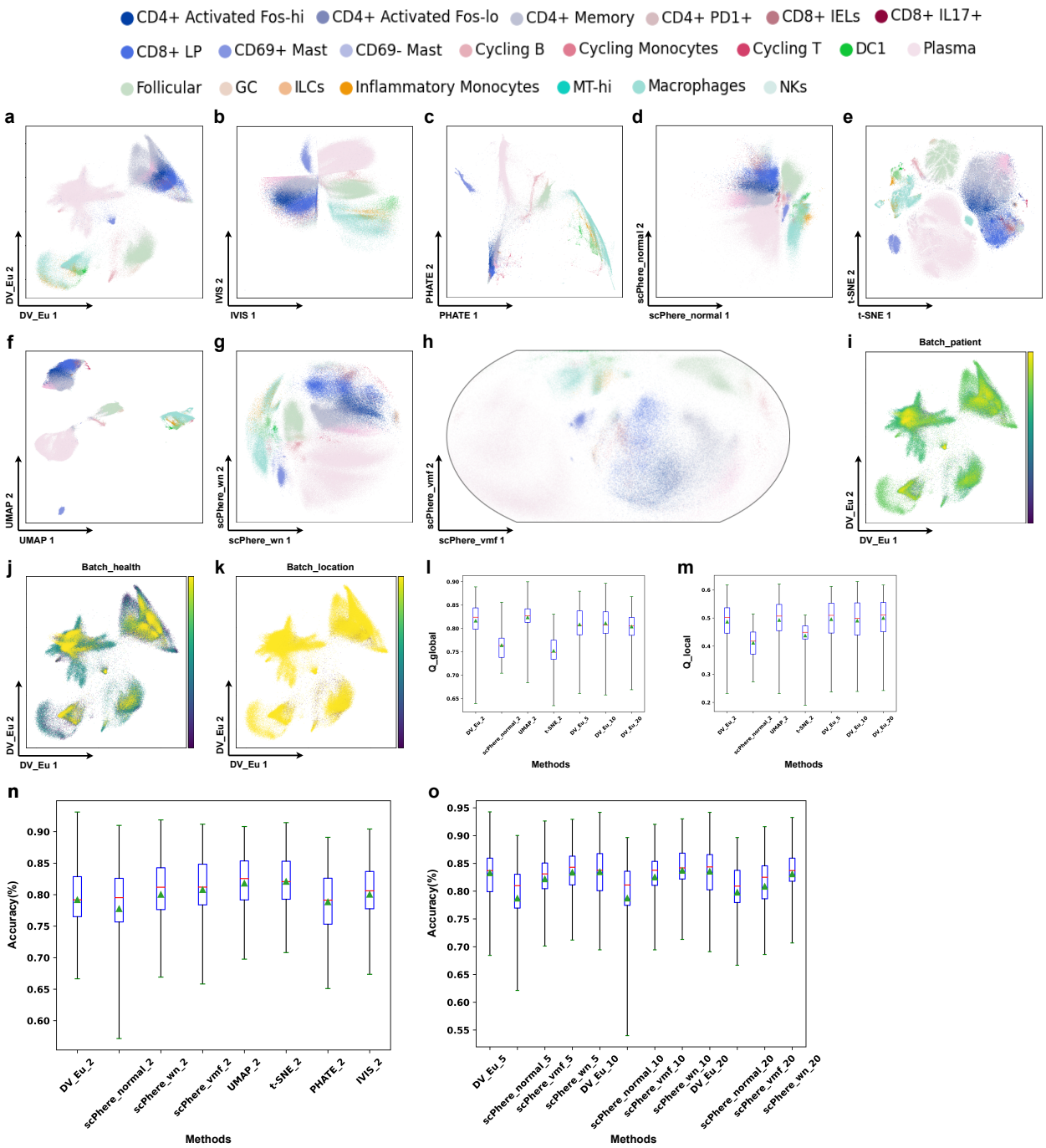

Figure 4. DV addresses complex technical and biological batch for visualization and analysis in colon biopsies from healthy individuals and UC immune patients. 2-dimensional DV_Eu (**a**), scPhere_normal (**d**), scPhere_wn (**g**) and scPhere_vmf (**h**) embedding accounting for the patient, location, and disease status. 2-dimensional IVIS (**b**), PHATE (**c**), t-SNE (**e**) and UMAP (**f**) embedding (batch-corrected by Harmony accounting for the patient status). DV_Eu embedding colored by batch_patient (**i**), batch_health (**j**) and batch_location (**k**) label. Successful batch correction visualization as reflected by local (**l**) and global (**m**) geometric structure preservation performance (y axis), and *k*-NN classification accuracies (y axis, *k*=5) in 2-dimensional (**n**) and low-dimensional(**o**) embeddings. The geometric structure preservation is tested on the cells from one patient between input of model and visualization embeddings. The *k*-NN classification accuracy is tested on the cells from one patient after training on the cells from all other patients. Boxplots denote means, medians and interquartile ranges (IQRs). The whiskers of a boxplot are the lowest datum still within 15 IQR of the lower quartile and the highest datum still within 15 IQR of the upper quartile.

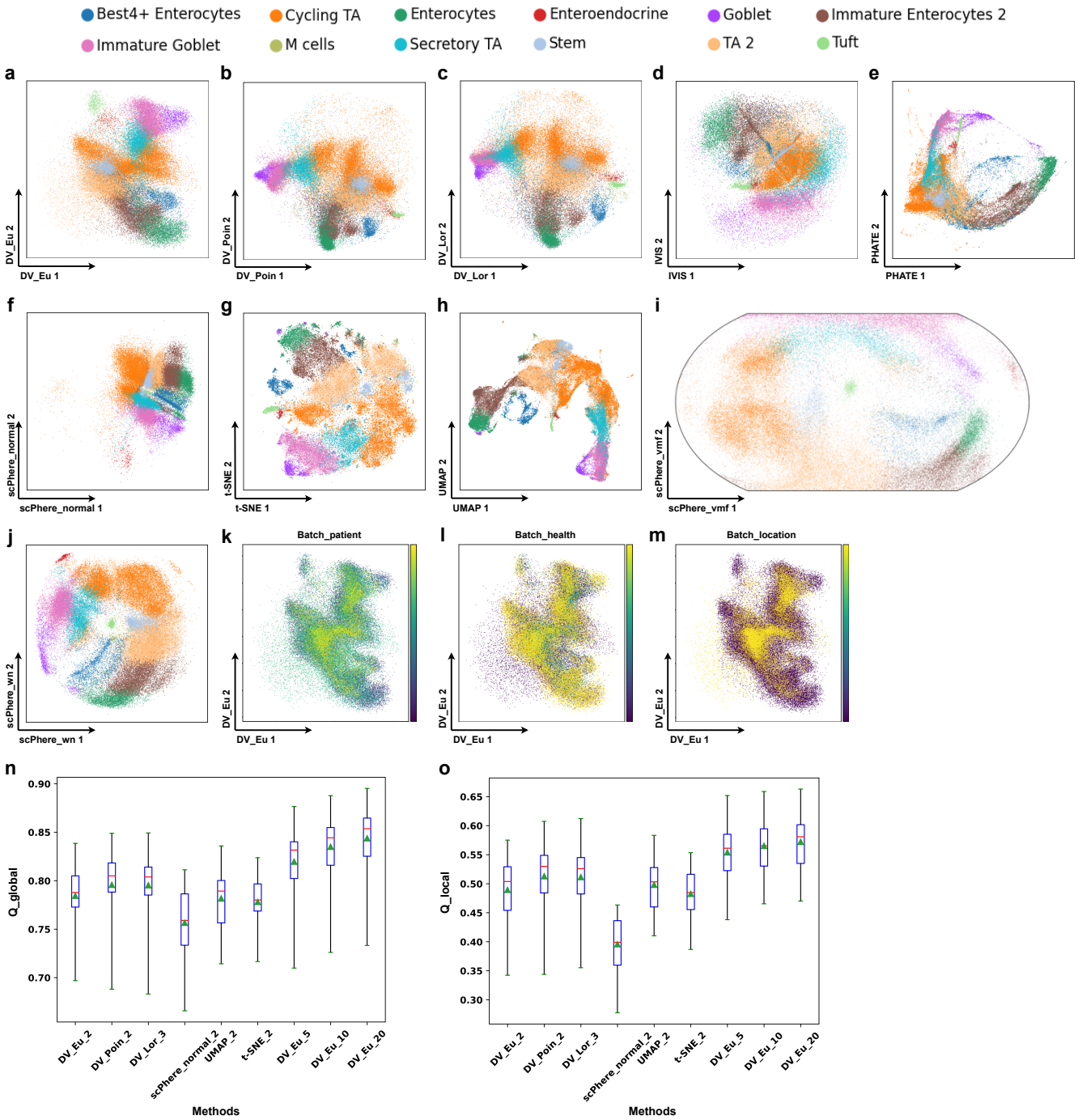

Figure 5. DV addresses complex technical and biological batch for visualization and analysis in colon biopsies from healthy individuals and UC epithelial patients. DV_Eu (**a**), DV_Poin (**b**), DV_Lor (**c**), scPhere_normal (**f**), scPhere_wn (**j**) and scPhere_vmf (**i**) embedding accounting for the patient, location and disease status. 2-dimensional IVIS (**d**), PHATE (**e**), t-SNE (**g**) and UMAP (**h**) embedding (batch-corrected by Harmony accounting for the patient status). DV_Eu embedding colored by batch_patient (**k**), batch_health (**l**) and batch_location (**m**) label based on DV_Eu visualization embeddings. Successful batch correction visualization as reflected by local (**n**) and global (**o**) geometric structure preservation performance in two dimensions and low dimensions.

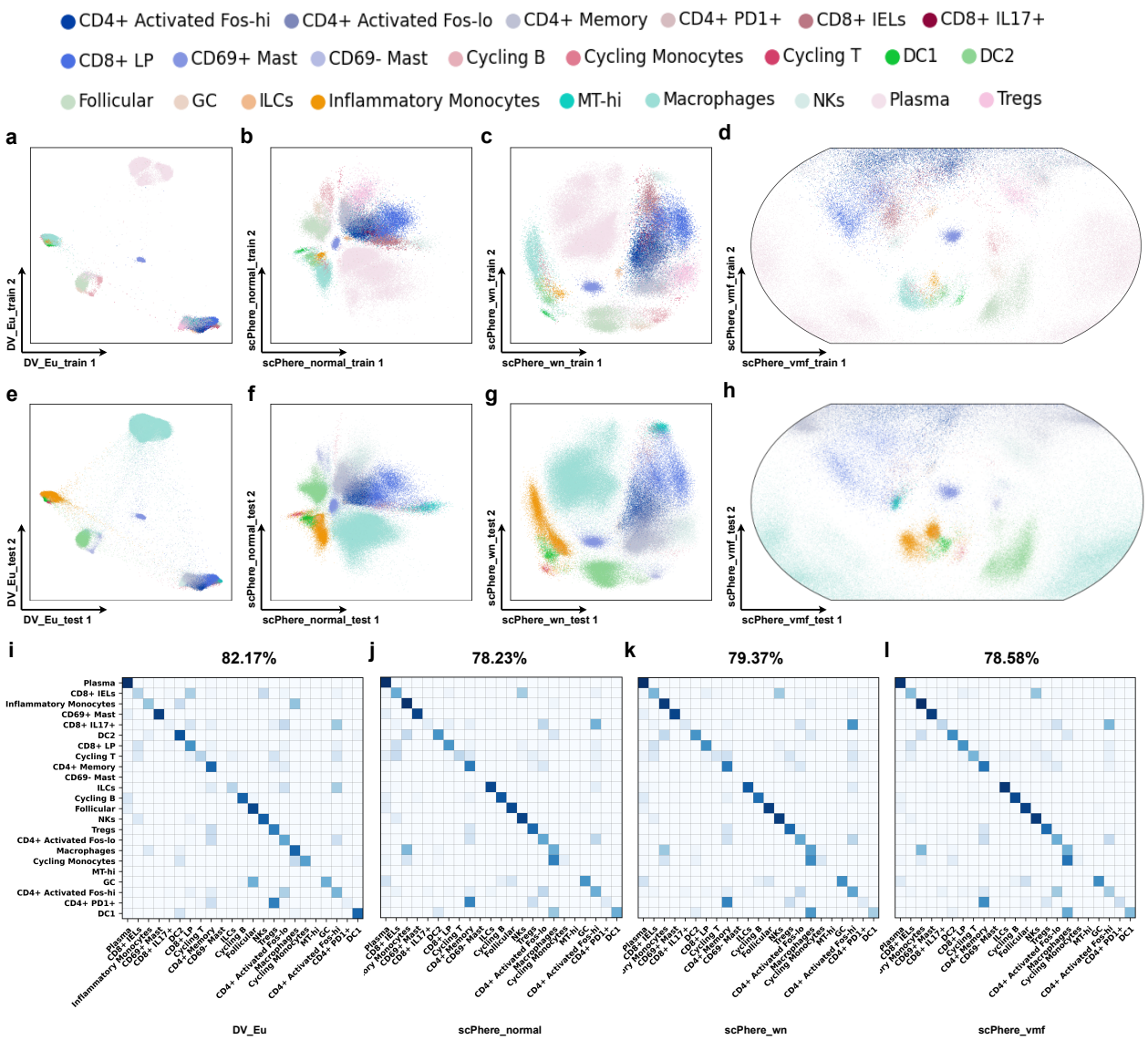

Figure 6. Using learned DV model to visualize cells, pinpoint cell types impacted by biological factors, and generate a batch-invariant reference atlas. Batch-invariant DV trained on 18 UC immune patients (**a-d**) and test on 12 UC immune patients (**e-h**). DV_Eu learned representations in the Euclidean space (**a, e**). 2-dimensional scPhere_normal (**b, f**), scPhere_wn (**c, g**) and scPhere_vmf (**d, h**). Confusion matrices (**i-l**) of the overlap in cells (row-centered and scaled Z-score, color bar) between "true" cell types from the original study (rows) and cell assignments by k-NN classifications (k=5) from batch-invariant DV trained only on training set cells for immune cells. CD69- mast cells and MThi cells are observed in the training data only.

Methods

Data preprocessing. For raw sequencing data, we strongly recommend preprocessing it with one of the preprocessing pipelines. The pipeline should be chosen according to the sequencing machine used to produce the data. Typically, preprocessing steps involve quality control, normalization, log scaling, and PCA. For complex multi-batch datasets, batch correction can be necessary.

The nearest-neighbors manifold learning methods (e.g., t-SNE, UMAP, IVIS, PHATE, and Poincaré maps) can suffer from the curse of dimensionality. Therefore, when the dimensionality of data exceeds 100 dimensions, preprocessing to its 50–100 principal components is a widely used strategy to address this problem.

For PCA, t-SNE, UMAP, and PHGAE, we used the code in the "scikit-learn" package[39]. For Poincaré maps, IVIS and scPhere, we follow its tutorial and released code. For these methods except scPhere, we use 50 principal components preprocessed as inputs by default. For scPhere, we use the raw data only preprocessed by log scaling as inputs. The proposed DV method is compatible with the above two preprocessing methods.

For the traditional dimensionality reduction methods, such as PCA, t-SNE, UMAP, IVIS, PHATE, and Poincaré maps, require multiple separate steps (batch correction, dimensionality reduction, and visualization) to achieve visualization, each with its own method or algorithm. For the batch-correction method Harmony, we use the code in "scanpy" package[40] and use the default parameter settings. Seurat3 CCA and LIGER methods can only handle one batch vector, and we thus remove them from experimental comparison. However, DV and scPhere provide an end-to-end, single process that can achieve visualization and batch correction simultaneously.

DV overview. DV receives a scRNA-seq dataset $\mathscr{D} = \{(\mathbf{x_i}, \mathbf{y_i})\}_{i=1}^{n}$ as input, where $\mathbf{x_i} \in \mathbb{R}^d$ is the gene expression vector of cell i, $\mathbf{y_i}$ is a categorical variable vector specifying the batch (multi-hot encoding) in which $\mathbf{x_i}$ is measured, d is the number of measured genes, and n is the number of cells.

For scRNA-seq data, the observed unique molecular identifier (UMI) counts of cells are sparse. Therefore, linear data augmentation (mixup) is adopted to improve the stability and generalization of the model, and it generates augmented data by a convex combination between cells and their k neighbors based on a raw graph G_{raw} (k-NN graph constructed on raw data):

$$\begin{aligned} \hat{\mathbf{x}}_\mathbf{i} &= (1 - r_u) \cdot \mathbf{x_i} + r_u \cdot \mathbf{x_j}, \qquad where \;\; \mathbf{x_j} \in N(\mathbf{x_i}) \\ \hat{\mathbf{y}}_\mathbf{i} &= (1 - r_u) \cdot \mathbf{y_i} + r_u \cdot \mathbf{y_j}, \qquad where \;\; \mathbf{y_j} \in \mathbf{y_i}(N(\mathbf{x_i})) \end{aligned} \tag{1}$$

where $N(\mathbf{x_i})$ denotes the k-NN list of the data point $\mathbf{x_i}$, $r_u \sim U(0, p_u)$ denotes the linear combination parameter r_u is sampled from the uniform distribution $U(0, p_u)$ and p_u is a hyperparameter. Now, we obtain the updated dataset $\widetilde{\mathscr{D}} = \{(\tilde{\mathbf{x}}_\mathbf{i}, \tilde{\mathbf{y}}_\mathbf{i})\}_{i=1}^{a \times b} = \{(\mathbf{x_i}, \mathbf{y_i})\}_{i=1}^{b} \cup \{(\hat{\mathbf{x}}_\mathbf{i}, \hat{\mathbf{y}}_\mathbf{i})\}_{i=b+1}^{(a+1)\times b}$ combined original dataset $\mathscr{D}$ with augmented dataset $\widehat{\mathscr{D}} = \{(\hat{\mathbf{x}}_\mathbf{i}, \hat{\mathbf{y}}_\mathbf{i})\}_{i=1}^{a \times b}$ as input, where a is the data augmentation number of each cell, $\hat{\mathbf{x}}_\mathbf{i}$ and $\hat{\mathbf{y}}_\mathbf{i}$ are the augmented gene expression vector of cell i and the corresponding batch categorical variable vector, respectively, and here $b = n$ due to DV is trained by mini-batch stochastic gradient descent.

Although $\mathbf{x}_i$ is high-dimensional, its intrinsic dimensionality is typically much lower. Manifold learning assumes that a decent embedding should preserve the geometry structure of raw data as much as possible. Therefore, we adopt the geometric structure preservation loss function to optimize DV, which minimizes the distribution discrepancy between $G_{semantic}$ and $G_{visualization}$ in the form of fuzzy sets cross entropy (two-way divergency):

$$L_{GSP} = \sum_{i,j=1}^{(a+1)\times b} \mathbf{u}_\mathbf{ij}^\mathbf{se} \log \frac{\mathbf{u}_\mathbf{ij}^\mathbf{se}}{\mathbf{u}_\mathbf{ij}^\mathbf{vi}} + \left(1 - \mathbf{u}_\mathbf{ij}^\mathbf{vi}\right) \log \frac{\left(1 - \mathbf{u}_\mathbf{ij}^\mathbf{vi}\right)}{\left(1 - \mathbf{u}_\mathbf{ij}^\mathbf{se}\right)} \tag{2}$$

where b is the number of batch size, $\mathbf{u}_\mathbf{ij}^\mathbf{se}$ is the undirectional similarity between semantic embedding $\mathbf{z}_\mathbf{i}^\mathbf{se}$ and $\mathbf{z}_\mathbf{j}^\mathbf{se}$ learned by semantic module, and $\mathbf{u}_\mathbf{ij}^\mathbf{vi}$ is the undirectional similarity between visualization embedding $\mathbf{z}_\mathbf{i}^\mathbf{vi}$ and $\mathbf{z}_\mathbf{j}^\mathbf{vi}$ learned by visualization module. The undirectional similarity $\mathbf{u}_\mathbf{ij}$ is defined as following:

$$\mathbf{u}_\mathbf{ij} = \mathbf{u}_{\mathbf{i}|\mathbf{j}} + \mathbf{u}_{\mathbf{j}|\mathbf{i}} - 2\mathbf{u}_{\mathbf{i}|\mathbf{j}}\mathbf{u}_{\mathbf{j}|\mathbf{i}} \tag{3}$$

where $\mathbf{u}_{\mathbf{j}|\mathbf{i}}$ is a directional similarity (edge weigt of graph) converted from the Euclidean or hyperbolic distance $D\left(\tilde{\mathbf{z}}_\mathbf{i}, \tilde{\mathbf{z}}_\mathbf{j}\right)$ between embedding $\tilde{\mathbf{z}}_\mathbf{i}$ and embedding $\tilde{\mathbf{z}}_\mathbf{j}$ and adopts normalized squared t-distribution:

$$\mathbf{u}_{\mathbf{j}|\mathbf{i}}(\nu) = g\left(D\left(\tilde{\mathbf{z}}_\mathbf{i}, \tilde{\mathbf{z}}_\mathbf{j}\right) \mid \nu\right) = C_\nu \left(1 + \frac{D\left(\tilde{\mathbf{z}}_\mathbf{i}, \tilde{\mathbf{z}}_\mathbf{j}\right)}{\nu}\right)^{-(\nu+1)} \tag{4}$$

where ν is the degrees of freedom in the t-distribution, ν^{se} in $\mathbf{u}^{\mathbf{se}}_{\mathbf{j}|\mathbf{i}}$ is set to 100, ν^{vi} in $\mathbf{u}^{\mathbf{vi}}_{\mathbf{j}|\mathbf{i}}$ is a hyperparameter, and

$$C_\nu = 2\pi \left(\frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\nu\pi}\Gamma\left(\frac{\nu}{2}\right)} \right)^2 \tag{5}$$

is the normalizing function of ν.

Especially, given that the observed UMI counts of cells are sparse, the relationship between cells is difficult to be defined by vector similarity (e.g., Euclidean distance) directly for such high-dimensional sparse data. Therefore, to estimate the local geometries of the underlying topological manifold and construct more reliable $G_{semantic}$ to describe the relationship between cells more accurately, we make a local scale contraction for Euclidean distance between cells, and their corresponding augmented cells for $G_{semantic}$, and $\mathbf{u}_{\mathbf{j}|\mathbf{i}}$ is redefined as:

$$\mathbf{u}^{\mathbf{se}}_{\mathbf{j}|\mathbf{i}}(\nu) = g\left(D\left(\mathbf{z}^{\mathbf{se}}_{\mathbf{i}}, \hat{\mathbf{z}}^{\mathbf{se}}_{\mathbf{j}}\right) \mid \nu\right) = C_\nu \left(1 + \frac{D\left(\mathbf{z}^{\mathbf{se}}_{\mathbf{i}}, \hat{\mathbf{z}}^{\mathbf{se}}_{\mathbf{j}}\right)/\gamma}{\nu}\right)^{-(\nu+1)} \tag{6}$$

where γ is the local scale contraction coefficient.

In addition, to remove batch effect problem simultaneously, we introduce batch graph G_{batch} constructed on batch categorical variable vector based on Euclidean distance $D_{\mathbb{E}}\left(\hat{\mathbf{y}}_{\mathbf{i}}, \hat{\mathbf{y}}_{\mathbf{j}}\right)$ merged with $G_{visualization}$ in the training process. Therefore, the $\mathbf{u}_{\mathbf{j}|\mathbf{i}}$ is redefined as:

$$\mathbf{u}^{\mathbf{vi}}_{\mathbf{j}|\mathbf{i}}(\nu) = g\left(D\left(\hat{\mathbf{z}}^{\mathbf{vi}}_{\mathbf{i}}, \hat{\mathbf{z}}^{\mathbf{vi}}_{\mathbf{j}}\right) \mid \nu\right) = C_\nu \left(1 + \frac{D\left(\hat{\mathbf{z}}^{\mathbf{vi}}_{\mathbf{i}}, \hat{\mathbf{z}}^{\mathbf{vi}}_{\mathbf{j}}\right) + \beta \cdot D_{\mathbb{E}}\left(\hat{\mathbf{y}}_{\mathbf{i}}, \hat{\mathbf{y}}_{\mathbf{j}}\right)}{\nu}\right)^{-(\nu+1)} \tag{7}$$

where β represents the importance of G_{batch}.

Poincaré ball and Lorentz model of the hyperbolic space. A Riemannian manifold $(\mathbb{M}, g)$ is a real and smooth manifold equipped with an inner product $g_{\mathbf{x}} : \mathbb{T}_{\mathbf{x}}\mathbb{M} \times \mathbb{T}_{\mathbf{x}}\mathbb{M} \to \mathbb{R}$ at each point $\mathbf{x} \in \mathbb{M}$, which is called a Riemannian metric and allows us to define the geometric properties of a space such as angles and the length of a curve. We introduce three commonly used hyperbolic manifolds compared with Euclidean space:

The Euclidean manifold is a manifold with zero curvature. The metric tensor is defined as $g^{\mathbb{E}} = \mathrm{diag}([1, 1, \ldots, 1])$, The closed-form distance, i.e., the length of the geodesic, which is a straight line in Euclidean space, between two points is given as:

$$D_{\mathbb{E}}(\mathbf{x_i}, \mathbf{x_j}) = \sqrt{\left(\mathbf{x_i} - \mathbf{x_j}\right)^T \left(\mathbf{x_i} - \mathbf{x_j}\right)} \tag{8}$$

The exponential map of the Euclidean manifold is defined as:

$$\exp_{\mathbf{x}}(\mathbf{v}) = \mathbf{x} + \mathbf{v} \tag{9}$$

The Poincaré ball model with constant negative curvature $-K(K > 0)$ corresponding to the Riemannian manifold $(\mathbb{P}, g^{\mathbb{P}}_{\mathbf{x}})$, where $\mathbb{P} = \left\{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\| < \frac{1}{K}\right\}$ is an open ball. Its metric tensor is $g^{\mathbb{P}}_{\mathbf{x}} = (\lambda^K_{\mathbf{x}})^2 g^{\mathbb{E}}$, where $\lambda^K_{\mathbf{x}} = \frac{2}{1+K\|\mathbf{x}\|^2}$ is the conformal factor and g^E is the Euclidean metric tensor. The distance between two points $\mathbf{x_i}, \mathbf{x_j} \in \mathbb{P}$ is given as:

$$D_{\mathbb{P}}(\mathbf{x_i}, \mathbf{x_j}) = \frac{1}{\sqrt{K}} \operatorname{arcosh}\left(1 + 2K \frac{\|\mathbf{x_i} - \mathbf{x_j}\|^2}{\left(1 - K\|\mathbf{x_i}\|^2\right)\left(1 - K\|\mathbf{x_j}\|^2\right)}\right) \tag{10}$$

For any point $\mathbf{x_i} \in \mathbb{P}$, the exponential map $\exp_{\mathrm{x}} : \mathbb{T}_{\mathrm{x}}\mathbb{P} \to \mathbb{P}$ is defined for the tangent vector $\mathbf{v} \neq 0$ and the point $\mathbf{x_j} \neq 0$ as:

$$\exp^K_{\mathbf{x}}(\mathbf{v}) = \mathbf{x} \oplus_K \left(\tanh\left(\frac{\sqrt{K}\lambda^K_{\mathbf{x}}\|\mathbf{v}\|}{2}\right) \frac{\mathbf{v}}{\sqrt{K}\|\mathbf{v}\|}\right) \tag{11}$$

where $\oplus_K$ is the **Möbius addition** for any $\mathbf{x_i}, \mathbf{x_j} \in \mathbb{P}$:

$$\mathbf{x_i} \oplus_K \mathbf{x_j} = \frac{\left(1 + 2K\langle \mathbf{x_i}, \mathbf{x_j}\rangle - K\|\mathbf{x_j}\|^2\right)\mathbf{x_i} + \left(1 - K\|\mathbf{x_i}\|^2\right)\mathbf{x_j}}{1 + 2K\langle \mathbf{x_i}, \mathbf{x_j}\rangle + K^2\|\mathbf{x_i}\|^2\|\mathbf{x_j}\|^2} \tag{12}$$

The Lorentz model avoids numerical instabilities that may arise with the Poincaré distance (mostly due to the division). Let $\mathbf{x_i}, \mathbf{x_j} \in \mathbb{R}^{d+1}$, then the Lorentzian scalar product is defined as:

$$\langle \mathbf{x_i}, \mathbf{x_j} \rangle_{\mathbb{L}} = -(\mathbf{x_i})_0 (\mathbf{x_j})_0 + \sum_{i=1}^{d} (\mathbf{x_i})_d (\mathbf{x_j})_d \tag{13}$$

The Lorentz model of n-dimensional hyperbolic space with constant negative curvature $-K(K>0)$ is defined as the Riemannian manifold $\left(\mathbb{L}, g_{\mathbf{x}}^{\mathbb{L}}\right)$, where $\mathbb{L} = \left\{\mathbf{x} \in \mathbb{R}^{n+1} : \langle \mathbf{x}, \mathbf{x} \rangle_{\mathbb{L}} = -1, \mathbf{x_0} > 0\right\}$ and where $g^{\mathbb{L}} = \mathrm{diag}([-1, 1, \ldots, 1])$. The induced distance function is given as:

$$D_{\mathbb{L}}(\mathbf{x_i}, \mathbf{x_j}) = \sqrt{K} \operatorname{arcosh}\left(-\langle \mathbf{x_i}, \mathbf{x_j} \rangle_{\mathbb{L}} / K\right) \tag{14}$$

The exponential map $\exp_{\mathbf{x}} : \mathbb{T}_{\mathbf{x}}\mathbb{L} \to \mathbb{L}$ is defined as:

$$\exp_{\mathbf{x}}^{K}(\mathbf{v}) = \cosh\left(\frac{\|\mathbf{v}\|_{\mathbb{L}}}{\sqrt{K}}\right)\mathbf{x} + \sqrt{K}\sinh\left(\frac{\|\mathbf{v}\|_{\mathbb{L}}}{\sqrt{K}}\right)\frac{\mathbf{v}}{\|\mathbf{v}\|_{\mathbb{L}}} \tag{15}$$

where $\|\mathbf{v}\|_{\mathbb{L}} = \sqrt{\langle \mathbf{v}, \mathbf{v} \rangle_{\mathbb{L}}}$. The origin, i.e., the zero vector in Euclidean space and the Poincaré ball, is equivalent to $(1, 0, ..., 0)$ in the Lorentz model. The point $(\mathbf{x_0}, \mathbf{x_1}, \ldots, \mathbf{x_n})^T$ between Poincaré ball and Lorentz model can be conveniently converted:

$$p_{\mathbb{L}\to\mathbb{P}}(\mathbf{x_0}, \mathbf{x_1}, \ldots, \mathbf{x_n}) = \frac{(\mathbf{x_1}, \ldots, \mathbf{x_n})}{\sqrt{K}\mathbf{x_0} + 1} \tag{16}$$

Model structure We use the Leaky rectified linear units (LeakyReLU) activation functions for hidden layers. The gradient can also be calculated where the LeakyReLU activation function input is less than zero during the backpropagation process, avoiding the jagged problem in the direction of the gradient. Meanwhile, we use batch normalization (BN) for hidden layers. In the training process of deep neural networks, the input of each layer of neural networks can keep the same distribution.

For all experiments, we use a six-layered neural network, including a semantic module (d/50-500-300-100) and a visualization module (100-300-100-3/2). The dimensionality $\mathbf{z}$ of the visualization embedding layer was typically 2/3 for visualization purposes. When comparing DV with different latent spaces (5, 10, and 20), we keep all other factors the same. We use the Adam stochastic optimization algorithm and train model for 300 epochs. For our current implementation, we do not introduce an early stopping but train DV for a given number of epochs. We can obtain a good embedding when we only train the model for 50 epochs. We run all the experiments using a Ubuntu server and a single V100 GPU with 32GB memory.

Choices of hyperparameters. To ensure that each method achieves its optimal performance, we use the grid search method to find the optimal hyperparameters. For t-SNE, the search space of "perplexity" is $\{5, 10, 15, 20, 30, 40\}$. For UMAP, the search space of "min_dist" is $\{0.1, 0.3, 0.5, 0.7, 0.9\}$. For IVIS, the search space of "k" is $\{2, 4, 8, 16, 32, 64\}$. For Poincaré maps, the search space of "knn" is $\{15, 20, 30\}$, "sigma" is $\{1, 2\}$ and "gamma" is $\{1, 2\}$. For PHATE, we use the default hyperparameter settings. For scPhere, we use the default hyperparameter setting according to the released code.

In the following, we discuss the function of different hyperparameters in DV and propose typical value ranges. The "learning rate" adjusts the objective function converge to the local minimum in the proper time and is typically set to $\{1e^{-3}, 5e^{-3}\}$, the "batch size" improves the memory utilization and the accuracy of gradient descent direction, and is typically set to $\{500, 1000, 2000\}$, the "v" controls the edge weight size in $G_{visualization}$ and is typically set to $\{1e^{-3}, 5e^{-3}, 1e^{-2}\}$, the "$\gamma$" controls the local scale contraction level between cells and their corresponding augmentation cells and is typically set to $\{10, 1000, 1e^{5}\}$, the "β" controls how much batch information need to remove in loss function and is typically set to $\{1e^{-2}, 1, 100\}$.

Quantifying global/hierarchical and local structure preservation. To quantitatively compare the performance of different embedding approaches, we use the scale-independent quality criteria proposed by Lee and Verleysen[41] following Poin_maps[17]. The main idea of this approach is that a good dimensionality reduction approach should have good preservation of local and global distances on the manifold, e.g., close neighbors should be placed close to each other while maintaining large distances between distant points. Therefore, they proposed to use two scalar quality criteria Q_{local} and Q_{global} focusing separately on low and high-dimensional qualities of the embedding. The quantities of Q_{local} and Q_{global} range from 0 (bad) to 1 (good) and reflect how well are local and global properties of the dataset are preserved in the embedding. To estimate distances in the

high-dimensional space, we use Euclidean distances estimated as the length in a full-connected graph. For Poincaré maps, we follow its released code and use the geodesic distances estimated as the length of a shortest-path in a k-nearest neighbors graph. For distances in the low-dimensional space, we use Euclidean distances for all the embeddings except DV_Poin, DV_Lor, and Poincaré maps, for which we use hyperbolic distances. Furthermore, for datasets with batch effect problem, we evaluate the embedding performance of each batch separately, calculate the Q_{local} and Q_{global} between input data before batch correction and visualization embedding, and use the boxplots for statistics.

To learn a DV model that is invariant to the batch vectors and can be used to map cells from completely new batches, when training the DV model, we use DV model to map a gene expression vector to the low-dimensional representation directly without using the batch vector as an input to the model. The batch vector is only used in the objective function that takes both the latent representation of a cell and its cell batch vector to construct the latent geometric structure to preserve the semantic geometric structure during training DV. We call this modality of DV with no batch vectors for the "batch-invariant" DV, as it learns latent representations that are invariant to the batch vectors.

Data availability

We use publicly available datasets in this study (GEO: GSE126954[31], GSE119562[34], GSE130148[32], GSE111588[33], GSE137400[35], GSE134355[36], GSE126954[37]; Single Cell Portal: SCP259, SCP551. To make the results presented in this study reproducible, all processed data are available in Single Cell Portal SCP1873 (we will remove the access permission after the manuscript is accepted).

Code availability

The DV software package, implemented in Pyotrch, is available free from https://github.com/Westlake-AI/DV (we will remove the access permission after the manuscript is accepted), and as a Supplementary Software 1 accompanying this manuscript.

Acknowledgements

This work is supported in part by the Science and Technology Innovation 2030 - Major Project (No. 2021ZD0150100) and National Natural Science Foundation of China (No. U21A20427).

Author contributions statement

Yongjie Xu, Zelin Zang, and Stan Z. Li developed the method, Yongjie Xu conceived the experiment with guidance from Stan Z. Li, and Yongjie Xu conducted the experiment, analyzed the results, and wrote the manuscript. All authors collected the datasets and reviewed the manuscript.