# CONTRASTIVE POINCARÉ MAPS FOR SINGLE-CELL DATA ANALYSIS

**Nithya Bhasker[1], Hattie Chung[2], Louis Boucherie[3], Vladislav Kim[4], Stefanie Speidel[1] & Melanie Weber[5]**

[1] Department of Translational Surgical Oncology, National Center for Tumor Diseases (NCT/UCC Dresden), Dresden, Germany: German Cancer Research Center (DKFZ), Heidelberg, Germany; Faculty of Medicine and University Hospital Carl Gustav Carus, TUD Dresden University of Technology, Dresden, Germany; Helmholtz-Zentrum Dresden-Rossendorf (HZDR), Dresden, Germany
{firstname.lastname}@nct-dresden.de

[2] Department of Medicine, Cardiovascular Research Center, Yale School of Medicine
hattie.chung@yale.edu

[3] Technical University of Denmark
louibo@dtu.dk

[4] Machine Learning Research, Bayer AG, Berlin, Germany
vladislav.kim@bayer.com

[5] Harvard University
mweber@seas.harvard.edu

## ABSTRACT

Complex hierarchies and branching patterns underlie numerous biological processes, from organismic development to signal divergence across individual cells. Single-cell RNA sequencing has enabled the study of such complex biological processes at high resolution. However, classical analysis methods represent single-cell data in low-dimensional Euclidean space, distorting the complex hierarchies inherent in such data. A recent line of work proposes to represent hierarchical data in hyperbolic space instead, which mitigates much of the distortion effects observed in the Euclidean setting. However, existing approaches for hyperbolic representation learning emphasise the preservation of local structure at the cost of decreased global representation accuracy and are computationally inefficient. To overcome these limitations, we develop *Contrastive Poincaré Maps*, a novel self-supervised approach for learning hyperbolic representations of tabular data. Through a series of experiments on synthetic and real, single-cell data, we show that Contrastive Poincaré Maps accurately represent global structure in complex hierarchical data in a computationally efficient manner.

## 1 INTRODUCTION

Rapid advances in sequencing technology over the past decade have enabled the measurement of transcriptome-wide gene expressions at the level of single cells (*Single-cell RNA sequencing*), providing insight into complex biological processes, such as cellular differentiation. Computational data analysis tools are employed to discover and study the complex branching patterns exhibited by these processes. Notable examples include UMAP (McInnes et al., 2018) and t-SNE (Van der Maaten & Hinton, 2008), which are standard tools employed for data analysis across diverse domains, as well as specialised tools such as PHATE (Moon et al., 2019). A common feature of these tools is that they represent data in a Euclidean space. However, there are fundamental limitations in the accuracy with which hierarchical (or tree-like) data can be represented in low-dimensional Euclidean space. A remedy comes in the form of hyperbolic spaces, negatively curved manifolds, which are known to represent hierarchical data with high accuracy in low dimensions. Motivated by this realisation, Klimovskaia et al. (2020) recently proposed to analyse single-cell data in hyperbolic spaces. They propose *Poincaré Maps*, a method for learning representations of tabular data in hyperbolic space. Their work builds on a growing body of literature, which studies hyper-

bolic graph embeddings (Nickel & Kiela, 2017; Chamberlain et al., 2017; Chami et al., 2019) and their utility in downstream machine learning applications (Ganea et al., 2018; Weber et al., 2020). Poincaré Maps (PM) have been shown to learn high-quality representations for several single-cell data sets (Klimovskaia et al., 2020). However, there are also some crucial caveats that impact the method's utility in practice: While PM perform well at learning representations of *shallow* trees, their representation accuracy decreases, as the tree's depth increases. Importantly, single-cell data often exhibits complex branching patterns that form deep trees. This limits the applicability of PM to complex, large-scale single-cell data sets. In addition, PM often requires feature engineering to learn good representations, have a memory-intensive training process and cannot be updated efficiently.

In this work, we introduce an alternate architecture for learning hyperbolic representations of tabular data. Our approach, *Contrastive Poincaré Maps* (CPM), employs contrastive learning (Bahri et al., 2022) to train an encoder, which allows for efficiently learning hyperbolic representations. We systematically test the accuracy and computational complexity of CPM against PM on synthetic data, demonstrating that CPM significantly improves over PM along the axes identified above. We further compare CPM and PM on a single-cell data set, demonstrating that the observed improvements also translate to real data.

## 2 METHODS

### 2.1 REPRESENTING DATA IN HYPERBOLIC SPACE

Hyperbolic spaces are smooth Riemannian manifolds with constant negative curvature. In this work, we consider the *Poincaré ball model* of hyperbolic space, one of several known parametrizations. It defines a hyperbolic space within the Euclidean unit ball, i.e.,

$$\mathbb{P}^d = \{x \in \mathbb{R}^{d+1} : \ \|x\| < 1\} \tag{1}$$

$$d_{\mathbb{P}}(x, y) = \operatorname{acosh}\left(1 + 2\frac{\|x - y\|^2}{(1 - \|x\|^2)(1 - \|y\|^2)}\right) , \tag{2}$$

where $\|\cdot\|$ denotes the usual Euclidean norm. The exponential map $\mathrm{Exp}_x : \ T_x\mathbb{P}^d \to \mathbb{P}^d$, an important geometric tool, which, e.g., characterises geodesics in the space, is given by

$$\mathrm{Exp}_x(v) = \cosh(\|v\|_M)x + \sinh(\|v\|_M)\frac{v}{\|v\|_M} ,$$

where $\|v\|_M = \sqrt{\langle v, v\rangle_M}$ is computed with respect to the Minkowski product $\langle u, v\rangle_M = -u_0v_0 + \sum_{i=1}^d u_iv_i$.

In this work, we are concerned with representing data in hyperbolic space, i.e., we want to learn a map $\phi : \ \mathcal{X} \to \mathbb{P}^d$ between a data space $\mathcal{X}$ and hyperbolic representation space $\mathbb{P}^d$, which maps a data point $x \in \mathcal{X}$ to $\phi(x) \in \mathbb{P}^d$. Classical results on metric embeddings provide theoretical guarantees on the representation accuracy achieved when embedding trees into low-dimensional Euclidean or hyperbolic space. Here, representation error is measured with respect to how well pairwise distances between data points are preserved from data space to representation space. Gupta (1999) shows that a tree with $L$ leaves (i.e., number of nodes in the last level of the tree) can be represented in $d$-dimensional Euclidean space with representation error not better than a multiplicative factor $O(L^{1/(d-1)})$. In contrast, Sarkar (2011) shows that a tree of *any* size may be represented in $d$-dimensional hyperbolic space while preserving distances almost perfectly, i.e., with representation error at most a multiplicative factor of $O(1 + \epsilon)$.

Hierarchical or tree-like data is ubiquitous in machine learning. The complexity of machine learning algorithms usually depends on the dimension of the data space. Hence, the fact that hyperbolic spaces allow for representing tree-like data accurately in low dimensions, holds great promise for the design of efficient data analysis. A growing body of literature studies methods for computing hyperbolic representations of data (e.g., Nickel & Kiela (2017); Chamberlain et al. (2017); Chami et al. (2019)). We briefly comment on PM (Klimovskaia et al., 2020), a state-of-the-art hyperbolic embedding approach for single-cell data, which will serve as a baseline in subsequent sections. PM is a shallow embedding technique that maps the similarities in feature space (inferred geodesic distances) to a Poincaré disk. The computation of similarities in feature space is memory intensive

and often requires the input to be low-dimensional. When applied to single-cell data, the feature representations correspond to gene expressions and the samples to cells. In contrast to PM, our proposed approach is a deep embedding technique that learns to map the rich information in feature space to a Poincaré disk in an end-to-end manner without the need for additional pre-processing steps. Consequently, the proposed approach scales well with the increasing number of samples or cells.

### 2.2 Contrastive Poincaré Maps

Concurrent studies have demonstrated success in obtaining good hyperbolic representations using contrastive learning approach for image analysis tasks (Ge et al., 2023; Yue et al., 2023). Nonetheless, the application of hyperbolic contrastive learning for tabular data, especially for tree-like biological data remains unexplored. Our proposed approach, *Contrastive Poincaré Maps* (CPM), builds on SCARF Bahri et al. (2022), a contrastive learning approach for tabular data. In contrastive learning, similarity structure in the data is inferred from samples of positive (similar) and negative (dissimilar) instances. CPM utilises the same sampling strategy as SCARF; it samples positive instances by generating an additional corrupted view of the data for each minibatch, by replacing a randomly selected fixed percentage of feature observations from the minibatch with feature observations from other data points, randomly selected from the training data. The encoder is composed of a projection block and a hyperbolic block (see Figure 3). The projection block consists of linear layers with ReLU activation. The hyperbolic block is composed of hyperbolic linear layers with ReLU activation. A similar setup for integrating hyperbolic geometry into deep architectures was previously considered in Chami et al. (2019). The following equations summarise one forward pass of the model, $H = \sigma(W_e X^e + b_1)$, where $H$ is the latent representation learnt by the projection block, $W_e$ are the parameters of the projection block, $X^e$ is the input data, $b_1$ is the bias, and $\sigma(.)$ is the non-linear ReLU activation unit. These representations are projected to the Poincaré ball using the exponential map $Exp_o(.)$ and consequently fed to the hyperbolic block, as seen in the following equation:

$$Z^h = \sigma^{\otimes}(W_h \otimes \mathrm{Exp}_o(H) \oplus \mathrm{Exp}_o(b_2)) \ .$$

Here, $W_h$ and $b_2$ are the trainable parameters of the hyperbolic block and $Z^h$ is the hyperbolic representation learnt by the model; $\sigma^{\otimes}(.) = Exp_o(\sigma(Log_o(.)))$ and $Log_o(.)$ is the logarithmic map (Chami et al., 2019).

Hyperbolic representations are learnt for both views of the data by passing them through the encoder blocks with shared weights. The similarities between the two views, which we term *Poincaré similarities*, are calculated based on Poincaré distances (Eq. 1) as $1/(1 + d_{\mathbb{P}}(x, y))$. These similarities are subsequently used in the classical InfoNCE contrastive loss for training our model. We use RiemannianAdam Chami et al. (2019) for optimising the model parameters. The parameters for the projection and hyperbolic blocks are optimised using different learning rates.

## 3 Experimental Results

We describe our experimental setup and present experimental results for synthetic and real single-cell data. This allows for a systematic comparison of CPM and PM in terms of representation accuracy, computational efficiency and utility in downstream tasks.

### 3.1 Experimental setup

**Data.** The synthetic data for the experiments in this section was generated using PHATE Moon et al. (2019), a method based on diffusion limited aggregation. The ground truth hierarchies (trees) have two branches at every depth level. To increase the heterogeneity of the trees, one of the branches at every depth level was set to contain one third of the total samples per level. The total number of samples in the tree is equal to $n_l$ (root level) $+(d \times n_l)$, where $n_l$ is the number of samples per level and $d$ is the depth of the tree. The features of the synthetic data were standardised to have zero mean and unit variance.

**Training.** The following parameters were used for CPM, unless otherwise specified. The rate of corruption was set to 0.2 for generating the corrupted view of the data. For the projection block,

Table 1: Quantitative metrics for the performance of CPM and PM across different depths of the synthetic data tree.

| Depth | Method | $Q_{local}$ ↑ | $Q_{global}$ ↑ | MAP ↑ | $D_{wc}$ ↓ |
|---|---|---|---|---|---|
| 5 | PM | 0.82 | 0.76 | 0.91 | 125.15 |
| | CPM | 0.78 | 0.88 | 0.90 | 1.20 |
| 10 | PM | 0.85 | 0.75 | 0.91 | 175.74 |
| | CPM | 0.82 | 0.90 | 0.84 | 1.37 |
| 15 | PM | 0.85 | 0.74 | 0.92 | 93.42 |
| | CPM | 0.81 | 0.93 | 0.82 | 1.28 |
| 20 | PM | 0.83 | 0.67 | 0.88 | 252.28 |
| | CPM | 0.79 | 0.89 | 0.82 | 1.67 |
| 25 | PM | 0.84 | 0.66 | 0.89 | 498.07 |
| | CPM | 0.81 | 0.91 | 0.80 | 1.00 |
| 30 | PM | 0.84 | 0.63 | 0.88 | 693.48 |
| | CPM | 0.81 | 0.91 | 0.81 | 1.00 |

the number of linear layers was set to 1, the learning rate was 1e-3 and the embedding dimension was 128. For the hyperbolic block, the embedding dimension was set to 2, and the number of hyperbolic linear layers and learning rate were varied based on the depth of the tree. The model was trained for 1000 epochs in batches of 128 samples with an early stopping criterion. Poincaré Maps (PM) Klimovskaia et al. (2020) was used as the baseline. The experiments were conducted on one NVIDIA V100 GPU and 8 threads of Intel Xeon Gold 5120 processor.

**Evaluation.** Both the baseline and our proposed method were evaluated by visualisation and via Q-scores ($Q_{local}$ and $Q_{global}$) (Lee & Verleysen, 2010). In addition, we compare the methods' worst-case distortion ($D_{wc}$) and $k$-MAP score. The value of $k$ for MAP score was set to 150. Definitions for all quality metrics and details of visualisation methods can be found in the appendix.

### 3.2 Results for synthetic data

**Hypothesis 1: CPM achieves higher global representation accuracy for non-shallow trees.** In contrast to PM, CPM makes use of the rich information available in observation space to learn a robust representation of the data, which is unaffected by the depth of the tree. To corroborate this hypothesis, we conducted experiments on synthetic data by varying the depth of the tree from 5 to 30. The dimension of the observation space was set to 1000. The number of samples per level was equal to 750. The number of hyperbolic linear layers in CPM was varied from 2 at $d = 5, 10$, to 3 at $d = 15, 20, 25, 30$. The learning rate for the optimisation of the hyperbolic block parameters was set to 1e-5. The results in Table 1 indicate a clear advantage of CPM over PM in terms of representation accuracy. Although PM shows higher local representation accuracy (reflected in $Q_{local}$ and MAP score), the global representation accuracy (reflected in $Q_{global}$ and $D_{wc}$) worsens with the depth of the tree. However, CPM maintains a high global representation accuracy across all depths, while maintaining a competitive local representation accuracy. Example embeddings for PM and CPM at $d = 25$ can be found in the appendix (Figure 4).

**Hypothesis 2: CPM is memory-efficient.** Single-cell RNA sequencing often results in large data sets due to the exponential growth of the cells, which leads to high memory consumption. This, in turn, requires complex infrastructure to run single-cell data analysis. From energy-efficiency and accessibility perspective there is an urgent need for memory-efficient tools for single-cell data analysis (Stein, 2010). As the total number of samples increases, we observe that CPM is more memory efficient than PM. To illustrate this claim, we conducted experiments on synthetic data by varying sample sizes and measured the performance of CPM and PM. The dimension of the observation space was set to $10^4$, and the depth of the tree was kept constant at $d = 5$. For the CPM's hyperbolic block, the number of hyperbolic linear layers was 1 and the learning rate was 1e-4. The dimension of the observation space was reduced to 20 principal components for the baseline. Figure 1 illustrates the results of the experiment; we observe that memory and computing time grow

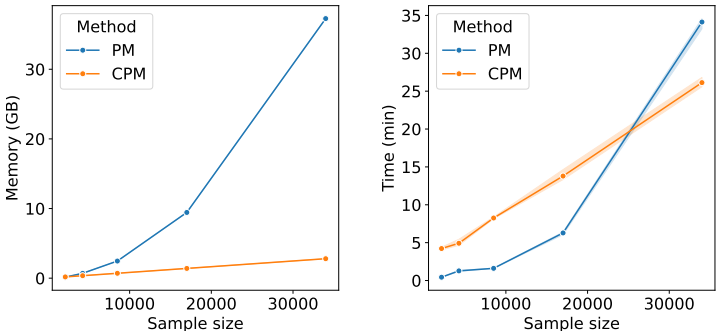


Figure 1: Comparison of CPM with PM in terms of memory and time consumption across different sample sizes.

Table 2: Performance of CPM and PM for chicken heart development dataset. Best results in bold font.

| Method | $Q_{local} \uparrow$ | $Q_{global} \uparrow$ | MAP $\uparrow$ | $D_{wc} \downarrow$ |
|---|---|---|---|---|
| CPM (Prj. block + Hyp. block) | **0.46** | 0.81 | 0.10 | **5.93** |
| CPM (PCA + Hyp. block) | 0.40 | 0.81 | 0.05 | 10.27 |
| CPM (PCA + Prj. block + Hyp. block) | 0.43 | **0.84** | 0.05 | 5.98 |
| PM w. PCA | 0.32 | 0.73 | **0.13** | 6.75 |
| PM w/o PCA | 0.31 | 0.74 | 0.11 | 12.30 |

Abbreviations: Prj. - projection, Hyp. - hyperbolic, w. - with, w/o - without.

linearly with the number of the samples in CPM as opposed to exponential growth in PM. Table 5 (see appendix) shows the evaluation results of the embeddings. The $Q_{gloabl}$ score for PM worsens with increasing sample size, while CPM is able to maintain both $Q_{global}$ and $Q_{local}$ across different sample sizes. The worst case distortion, $D_{wc}$ for both methods deplete with increasing sample size, but the decline is sharper for PM than CPM.

### 3.3 Application to Single-Cell Data Analysis

To validate the results observed on synthetic data, we conducted experiments on single-cell RNA sequencing data from Mantri et al. (2021). The data consists of 22,315 cells, which include progenitor and mature cell types from multiple lineages driving cardiogenesis in chicken embryos. The raw data was normalised to 10000 counts per cell, Log1p transformed and filtered to contain 2000 highly variable genes. For CPM's hyperbolic block, the number of hyperbolic linear layers was 2 and the learning rate was 1e-5. For the baseline, experiments were conducted both without dimensionality reduction, and with dimensionality reduction (number of principal components equal to 128). Table 3 compares the embedding quality of CPM with PM and also with other state of the art euclidean embedding methods. As observed in the case of synthetic data, CPM is able to achieve higher global representation accuracy than PM. CPM also respects and preserves local hierarchies within individual lineages, and relative hierarchies among different lineages. For example, CPM preserves hierarchies within the myocardial, epicardial and endocardial lineages in the dataset (Figure 2). It also separates and resolves structures within the non-cardiac erythrocyte, macrophages and dendritic cell types as seen in Figure 2. The lineages in Figure 2 were visualised by translating different parts of the same embedding to the origin (see Appendix C).

**Hypothesis 3: CPM does not need additional feature processing steps.** CPM can resolve hierarchies without feature engineering and dimensionality reduction. PM requires feature processing steps like PCA to reduce distortion in the embeddings (Table 2).

**Hypothesis 4: CPM learns representations that can be efficiently updated, as new data becomes available.** The Contrastive Poincaré model can be used in an inductive setting to generate embeddings for data held out from the training process. To corroborate this hypothesis, experiments were conducted by dividing the data set into train and test data in a 3:1 ratio. The test data was held out from model training. The trained model was used in inference mode to generate embeddings of

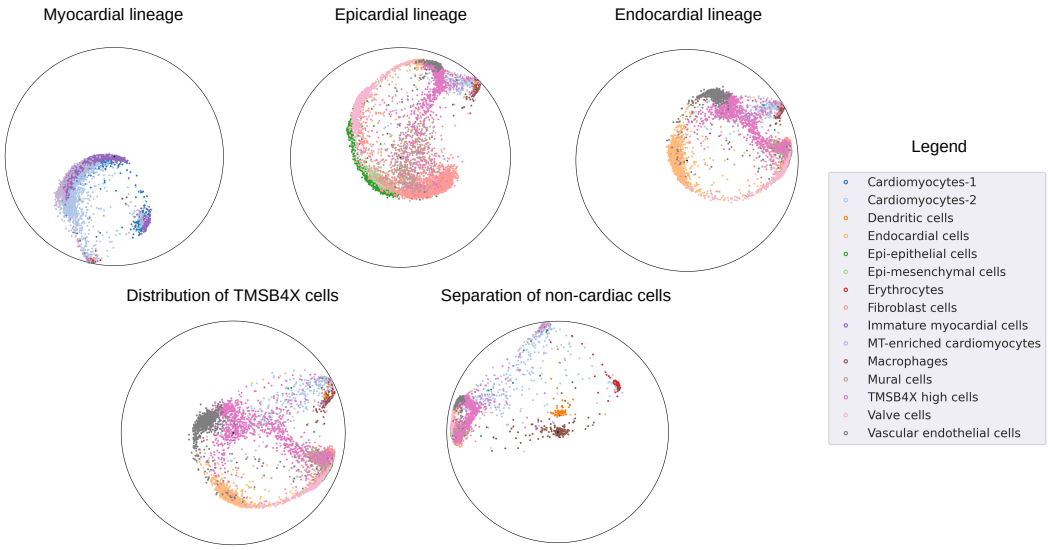


Figure 2: Illustration of CPM resolving multiple cardiac cell lineages in the chicken heart development dataset.

Table 3: Comparison of performance of CPM on chicken heart development dataset with state of the art embedding methods for single cell data analysis.

| Method | $Q_{local}$ ↑ | $Q_{global}$ ↑ | MAP ↑ | $D_{wc}$ ↓ |
|---|---|---|---|---|
| CPM | **0.46** | **0.81** | 0.10 | 5.93 |
| PM | 0.31 | 0.74 | **0.11** | 12.30 |
| t-SNE (Van der Maaten & Hinton, 2008) | 0.32 | 0.75 | 0.05 | 123.89 |
| UMAP (McInnes et al., 2018) | 0.36 | 0.77 | 0.06 | 111.47 |
| PHATE (Moon et al., 2019) | 0.45 | **0.81** | **0.11** | **3.00** |

the held out test data. Table 4 illustrates the generalisation power of CPM by comparing the quality of training data embedding with unseen test data embedding.

Table 4: Quantitative metrics for the performance of CPM in an inductive setting.

| Data | $Q_{local}$ | $Q_{global}$ | MAP | $D_{wc}$ |
|---|---|---|---|---|
| Train (75%) | 0.47 | 0.81 | 0.12 | 6.01 |
| Unseen test (25%) | 0.46 | 0.81 | 0.23 | 5.13 |

## 4 Discussion

We introduced Contrastive Poincaré Maps (CPM), a novel contrastive learning method for learning hyperbolic representation of single-cell data. In comparison with Poincaré Maps (PM) (Klimovskaia et al., 2020), an earlier, shallow hyperbolic representation method, CPM shows significant improvements along several axes: Through systematic experiments, we have demonstrated that CPM outperforms PM by (i) achieving high global representation accuracy for deep trees, (ii) reducing memory complexity, (iii) eliminating additional feature processing steps, and (iv) providing an inductive training setup, which facilitates downstream analyses (Table 4). In future work, we will expand the scale and breadth of the experiments, and apply the proposed method for other real world scenarios. In addition, we will perform a systematic comparison with Euclidean baselines on downstream tasks across different datasets. Euclidean tools, such as UMAP or t-SNE, are widely used visualisation tools for the qualitative assessment of single-cell data. However, representations in (Euclidean) UMAP space cannot be quantitatively interpreted due to representation error (see Section 2.1). In contrast, the preservation of global and local data geometry in hyperbolic space could enable the interpretability of latent features, e.g. by analysing feature variation along geodesics.

ACKNOWLEDGMENTS

Nithya Bhasker is funded by German Federal Ministry of Health (BMG) within the "Surgomics" project (Grant Number: BMG 2520DAT82) and partially by German Federal Ministry of Education and Research (BMBF) within the DAAD Konrad Zuse AI school SECAI (School of Embedded Composite AI, `https://secai.org/`) (Project number: 57616814). The authors would like to thank Alice Bizeul for her input during the initial phase of the project. The authors would also like to express their sincere gratitude to the anonymous reviewers for their valuable feedback.

## REFERENCES

Dara Bahri, Heinrich Jiang, Yi Tay, and Donald Metzler. Scarf: Self-supervised contrastive learning using random feature corruption. In *International Conference on Learning Representations*, 2022.

Benjamin Paul Chamberlain, James Clough, and Marc Peter Deisenroth. Neural embeddings of graphs in hyperbolic space. *arXiv preprint arXiv:1705.10359*, 2017.

Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks. *Advances in neural information processing systems*, 32, 2019.

Octavian Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic neural networks. *Advances in neural information processing systems*, 31, 2018.

Songwei Ge, Shlok Mishra, Simon Kornblith, Chun-Liang Li, and David Jacobs. Hyperbolic contrastive learning for visual representations beyond objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6840–6849, 2023.

Anupam Gupta. Embedding tree metrics into low dimensional euclidean spaces. In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing*, STOC '99, pp. 694–700, New York, NY, USA, 1999. Association for Computing Machinery. ISBN 1581130678.

Anna Klimovskaia, David Lopez-Paz, Léon Bottou, and Maximilian Nickel. Poincaré maps for analyzing complex hierarchies in single-cell data. *Nature communications*, 11(1):2966, 2020.

John A. Lee and Michel Verleysen. Scale-independent quality criteria for dimensionality reduction. *Pattern Recognition Letters*, 31(14):2248–2257, October 2010.

Madhav Mantri, Gaetano J Scuderi, Roozbeh Abedini-Nassab, Michael FZ Wang, David McKellar, Hao Shi, Benjamin Grodner, Jonathan T Butcher, and Iwijn De Vlaminck. Spatiotemporal single-cell rna sequencing of developing chicken hearts identifies interplay between cellular differentiation and morphogenesis. *Nature communications*, 12(1):1771, 2021.

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

Kevin R Moon, David van Dijk, Zheng Wang, Scott Gigante, Daniel B Burkhardt, William S Chen, Kristina Yim, Antonia van den Elzen, Matthew J Hirn, Ronald R Coifman, et al. Visualizing structure and transitions in high-dimensional biological data. *Nature biotechnology*, 37(12):1482–1492, 2019.

Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. *Advances in neural information processing systems*, 30, 2017.

Rik Sarkar. Low distortion delaunay embedding of trees in hyperbolic plane. In *International Conference on Graph Drawing*, pp. 355–366, 2011.

Lincoln D Stein. The case for cloud computing in genome informatics. *Genome biology*, 11:1–7, 2010.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Melanie Weber, Manzil Zaheer, Ankit Singh Rawat, Aditya K Menon, and Sanjiv Kumar. Robust large-margin learning in hyperbolic space. *Advances in Neural Information Processing Systems*, 33:17863–17873, 2020.

Yun Yue, Fangzhou Lin, Kazunori D Yamada, and Ziming Zhang. Hyperbolic contrastive learning. *arXiv preprint arXiv:2302.01409*, 2023.

## A Model Architecture: Contrastive Poincaré Maps

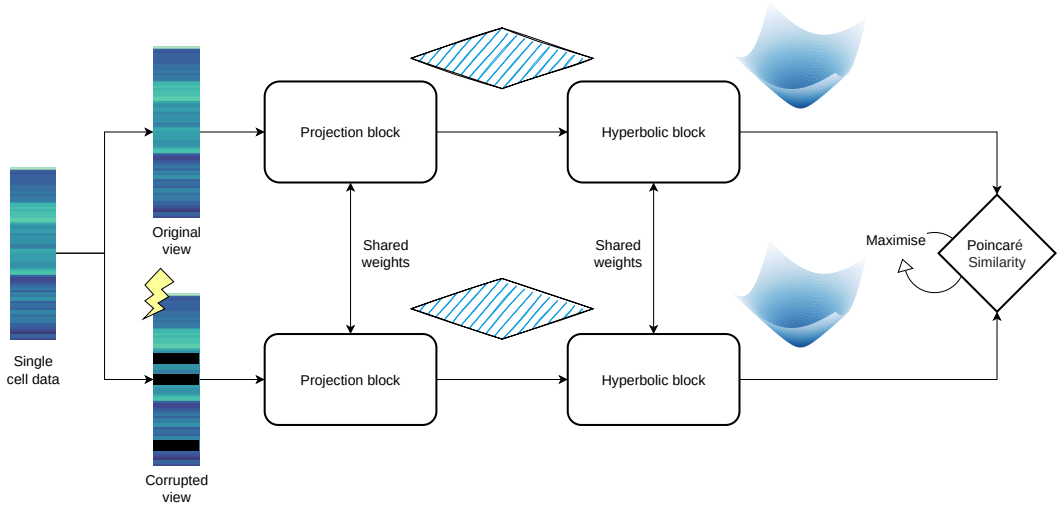


Figure 3: Contrastive Poincaré Maps

## B Embedding quality metrics

In the following, let $\phi : \mathcal{X} \to \mathbb{P}^d$ denote an embedding of a data set $\mathcal{X}$. We define the quality metrics reported in the main text.

**Distortion.** Following conventions in the classical metric embedding literature, we can measure the representation error of $\phi$ with respect to *distortion*:

$$\frac{1}{c_M} d_{\mathbb{P}}(\phi(x), \phi(y)) \leq d_{\mathcal{X}}(x, y) \leq c_M(\phi) d_{\mathbb{P}}(\phi(x), \phi(y)) \quad \forall x, y \in \mathcal{X} \ .$$

If $\phi$ is an isometric embedding, then $c_M(\phi) = 1$. Notice that $c_M$ is computed via a worst-case analysis. In our analysis, we compute *worst-case distortion* as follows:

$$D_{wc}(\phi) = \sup_{x,y \in \mathcal{X}} \left| \left( \frac{d_{\mathbb{P}}(x, y)}{d_{\mathcal{X}}(x, y)} \right)^2 - 1 \right| \ . \tag{3}$$

**$k$-MAP score.** The MAP score is a ranking loss, which evaluates how well proximity is preserved locally. We construct the similarity graph of $\mathcal{X}$ and evaluate how well the relationship of a point $x \in \mathcal{X}$ with its $k$-hop neighbours is preserved by a hyperbolic embedding $\phi$:

$$\mathrm{MAP}(\phi) = \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} \frac{1}{\deg(x)} \sum_{i=1}^{|\mathcal{N}_k(x)|} \frac{|\mathcal{N}_1(x) \cap R_{x,i}|}{|R_{x,i}|} \in [0, 1] \ . \tag{4}$$

Here, $R_{x,i}$ denotes the smallest set of nearest neighbours required to retrieve the $i^{th}$ neighbour of $\phi(x)$ in $\mathbb{P}^d$. If $\phi$ is an isometric (i.e., distance-preserving), then $\mathrm{D_{wc}} = 0$ and $\mathrm{MAP} = 1$.

**Q-scores.** Q-scores are standard metrics in the manifold embedding literature introduced by Lee & Verleysen (2010). Let $Q = \{q_{kl}\}_{k,l \in [N]}$ denote the *co-ranking matrix* defined by

$$q_{kl} := |\{(k, l) : \ \mu_{ij} = k, \nu_{ij} = l\}|$$
$$\mu_{ij} := |\{k : \ d_{\mathcal{X}}(x_i, x_k) < d_{\mathcal{X}}(x_i, x_j) \text{ and } \{d_{\mathcal{X}}(x_i, x_k) = d_{\mathcal{X}}(x_i, x_j) \text{ if } 1 \leq k < j \leq N\}\}|$$
$$\nu_{ij} := |\{k : \ d_{\mathbb{P}}(x_i, x_k) < d_{\mathbb{P}}(\phi(x_i), \phi(x_j)) \text{ and } \{d_{\mathbb{P}}(\phi(x_i), \phi(x_k)) = d_{\mathbb{P}}(\phi(x_i), \phi(x_j)) \text{ if } 1 \leq k < j \leq N\}\}|$$

Here, $|I|$, denotes the cardinality of a set $I$. We define Q-curves as

$$Q(\phi, K) := \frac{1}{KN} \sum_{(k,l) \in I_K} q_{kl} \ , \tag{5}$$

which are computed over blocks $I_K := \{1, \ldots, K\} \times \{1, \ldots, K\}$ of the co-ranking matrix, where we only consider the upper half of $I_K$, due to symmetry. We then compute local and global Q-scores of a hyperbolic embedding $\phi$ via

$$K_{\max} := \arg\max_K \left( Q(\phi, K) - \frac{K}{N-1} \right) \tag{6}$$

$$Q_{local}(\phi) := \frac{1}{K_{\max}} \sum_{K=1}^{K_{\max}} Q(\phi, K) \tag{7}$$

$$Q_{global}(\phi) := \frac{1}{N - K_{\max}} \sum_{K=K_{\max}}^{N-1} Q(\phi, K) \ . \tag{8}$$

Here, the local and global regimes are distinguished by a split of the Q-curve at $K_{\max}$. $Q_{local}$ and $Q_{global}$ attain values from 0 (worst) to 1 (best).

## C VISUALISATION

Contrastive Poincaré Maps are visualised using three methods, as done in Klimovskaia et al. (2020). All methods provide 2D visualisations on a circle of unit radius. The first method is a naïve scatter plot of the embedding. The second method provides a zoomed in view of the scatter plot by linearly scaling the embedding. Both of these methods place a sample with shortest distances to all other samples at the origin. The third method translates the entire embedding to place a sample of interest at the origin, while still preserving all distances between the samples. Because of the amplified spatial resolution at the origin in hyperbolic representations, the third method is useful to study relative hierarchical structures when the root node is known, or to study individual cell lineages when the dataset contains multiple lineages. The translation of the entire embedding $x$ to place $r$ at the origin is defined as:

$$\tau_\phi(r, x) = \frac{(1 + 2\langle r, x \rangle + \|x\|^2) r + (1 - \|r\|^2) x}{1 + 2\langle r, x \rangle + \|r\|^2 \|x\|^2}$$

## D ADDITIONAL EXPERIMENTAL RESULTS

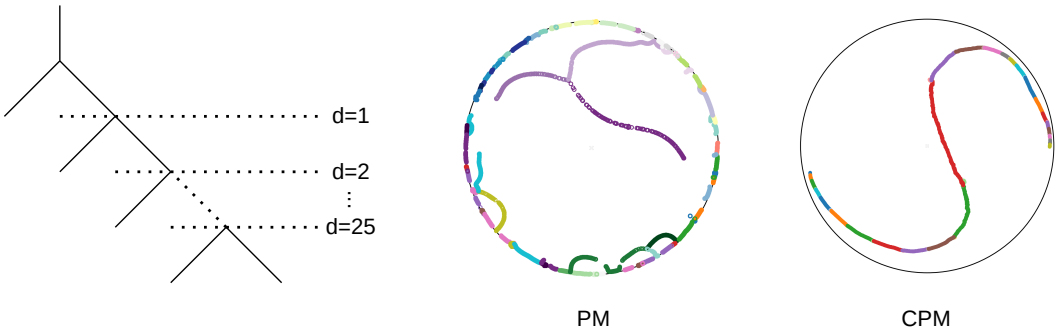


Figure 4: Comparison of CPM with PM for depth, $d = 25$.

Table 5: Quantitative metrics for the performance of CPM and PM for various sample sizes.

| Samples | Method | $Q_{local}$ ↑ | $Q_{global}$ ↑ | MAP ↑ | $D_{wc}$ ↓ |
|---|---|---|---|---|---|
| 2125 | PM | 0.88 | 0.91 | 0.93 | 8.84 |
| | CPM | 0.79 | 0.91 | 0.88 | 1.81 |
| 4250 | PM | 0.88 | 0.79 | 0.91 | 210.93 |
| | CPM | 0.81 | 0.92 | 0.91 | 2.02 |
| 8500 | PM | 0.90 | 0.75 | 0.92 | 342.73 |
| | CPM | 0.84 | 0.90 | 0.93 | 2.06 |
| 17000 | PM | 0.89 | 0.69 | 0.92 | 1173.97 |
| | CPM | 0.84 | 0.93 | 0.92 | 2.16 |
| 34000 | PM | 0.85 | 0.61 | 0.87 | 1559.02 |
| | CPM | 0.87 | 0.91 | 0.94 | 8.46 |