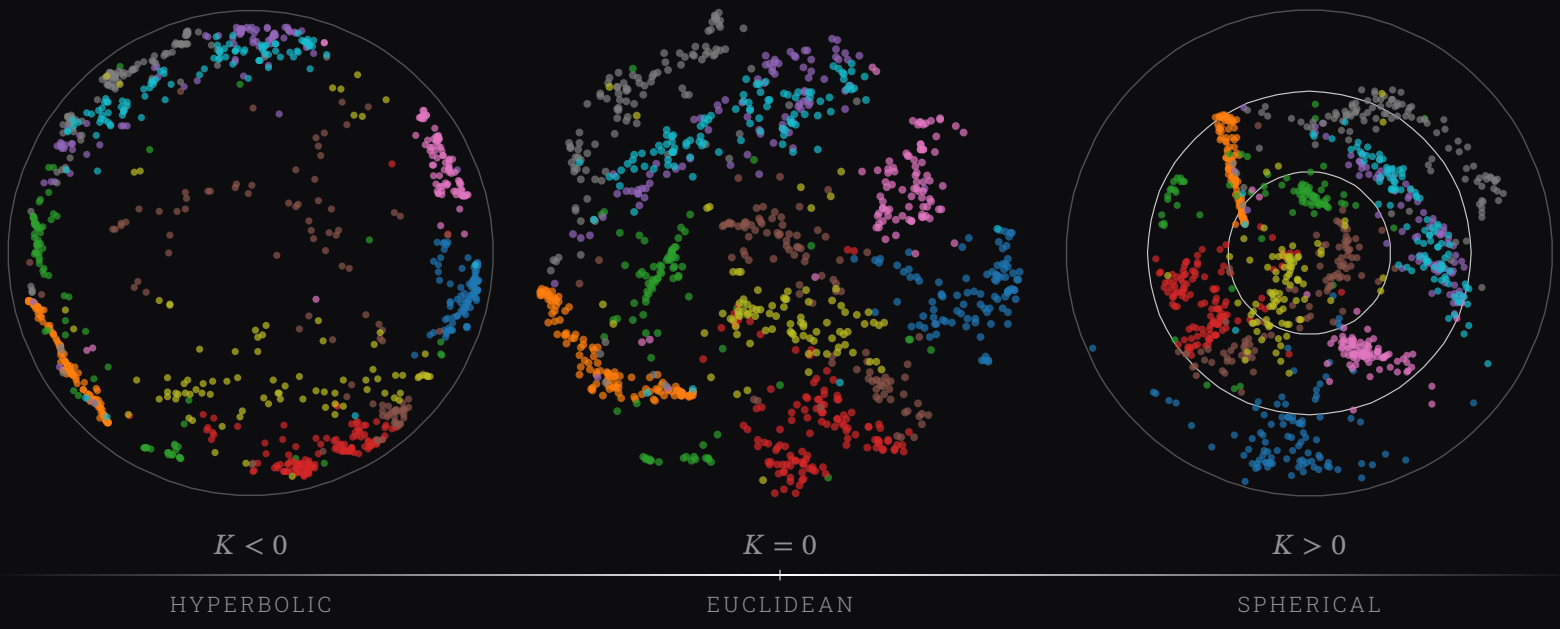
Visualising High-Dimensional Data using Non-Euclidean t-SNE

Ties Bloemen



TU Delft

Visualising High-Dimensional Data using Non-Euclidean t-SNE

by

Ties Bloemen

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Friday October 09, 2026 at 10:00

Student number:	5240298	
Project duration:	November 2025 – October 2026	
Daily supervisor:	Martin Skrodzki	
Thesis committee:	Klaus Hildebrandt	TU Delft, Supervisor
	Jasmijn Baaijens	TU Delft
	Martin Skrodzki	TU Delft.

TU Delft

Abstract

Dimensionality reduction methods such as t-SNE are standard for visualising high-dimensional data, yet almost universally embed into Euclidean space. Representation learning has shown that hyperbolic and spherical spaces fit hierarchical and cyclic data with lower distortion, but those gains are measured on downstream tasks, not on what an analyst reads from a two-dimensional scatter plot. This thesis asks how the curvature of the embedding space influences the quality of such visualisations.

To compare the visualisations, we generalise t-SNE to the Euclidean, hyperbolic and spherical geometries by using the geodesic distance in the low-dimensional kernel, and optimise it with Riemannian stochastic gradient descent. Because visualisation quality is multidimensional and t-SNE is sensitive to its hyperparameters, geometries are compared through visualisation corpora: Pareto fronts over six established quality metrics for local structure, distance preservation and label-aware structure, produced by multi-objective Bayesian optimisation over the hyperparameters, curvature included. The comparison covers synthetic datasets sampled from known constant-curvature manifolds and four real datasets.

The central hypothesis, that an embedding geometry matching the sign of the data's curvature improves visualisation quality, is not supported on the high-dimensional synthetic datasets. A curvature fit on their pairwise distances does not distinguish the generating geometries, whereas on two-dimensional versions of the same datasets it does, and there the visualisations suggest that the matched geometry is the most faithful. The geometry changes distance preservation only slightly, and label-aware differences change sign with the dataset rather than following matching. In the hyperbolic geometry, beyond a moderate dimensionless curvature (the curvature normalised by the squared characteristic length of the embedding) the metrics decline and global structure is visibly lost. Overall, curvature improves a high-dimensional visualisation little and not through matching, but can degrade it considerably. We recommend starting from Euclidean t-SNE, trying a curved geometry mainly when the curvature of the data can be recovered from its distances, and reporting its dimensionless curvature next to a Euclidean visualisation.

Contents

Abstract iii

1 Introduction 1
- 1.1 Approach and contributions . . . 2
- 1.2 Thesis Structure . . . 3

2 Related work 4
- 2.1 Dimensionality Reduction Methods . . . 4
- 2.2 Non-Euclidean Embeddings for Dimensionality Reduction . . . 5
- 2.3 Evaluating Dimensionality Reduction . . . 5
- 2.4 Multi-Objective Bayesian Optimisation for Visualisation . . . 6

3 Background 7
- 3.1 Dimensionality Reduction and Visualisation . . . 7
  - 3.1.1 The Problem of High-Dimensional Data . . . 7
  - 3.1.2 t-SNE . . . 8
  - 3.1.3 Limitations of t-SNE . . . 8
  - 3.1.4 Tasks Suited to t-SNE . . . 9
- 3.2 Non-Euclidean Geometry . . . 9
  - 3.2.1 Riemannian Manifolds and Curvature . . . 9
  - 3.2.2 Euclidean Space . . . 9
  - 3.2.3 Hyperbolic Space . . . 10
  - 3.2.4 Spherical Space . . . 10
  - 3.2.5 Projections . . . 11
    - 3.2.5.1 Hyperbolic projection . . . 11
    - 3.2.5.2 Spherical projection . . . 11
  - 3.2.6 Scale Invariance and the Dimensionless Curvature . . . 11

4 Methods 13
- 4.1 Infrastructure . . . 14
  - 4.1.1 Generalised t-SNE on Constant-Curvature Manifolds . . . 14
    - 4.1.1.1 Centroid Alignment . . . 15
  - 4.1.2 Riemannian Optimisation . . . 15
    - 4.1.2.1 Tangent Space Projection . . . 15
    - 4.1.2.2 Exponential Map . . . 16
    - 4.1.2.3 Parallel Transport . . . 16
    - 4.1.2.4 Iteration . . . 16
  - 4.1.3 Auxiliary Loss Terms for Global Structure . . . 17
    - 4.1.3.1 Centering Loss . . . 17
    - 4.1.3.2 Global-Structure Loss . . . 17
    - 4.1.3.3 Depth-Norm Loss . . . 18
- 4.2 Evaluation Metrics . . . 18
  - 4.2.1 Local Structure Preservation . . . 19
  - 4.2.2 Distance Preservation . . . 20
  - 4.2.3 Label-Aware Structure . . . 22
- 4.3 Generating visualisation corpora . . . 25
  - 4.3.1 Hyperparameter Optimisation Procedure . . . 25
  - 4.3.2 Training Schedule . . . 25

4.3.3 Hyperparameter Search Space . . . 25
4.3.4 Front Quality Indicator . . . 26
4.3.5 Preference Regions and Recommended Configurations . . . 27

5 Results **28**
5.1 Datasets . . . 28
5.1.1 Synthetic Datasets . . . 28
5.1.2 Real-World Datasets . . . 29
5.2 Does Matching Dataset and Embedding Curvature Improve Quality? . . . 30
5.2.1 Overall Quality of the Synthetic Corpora . . . 30
5.2.2 Where the Front Points Lie . . . 31
5.2.3 Visual Evidence for Geometry Matching . . . 32
5.3 What Effect Does Curvature Have on Established Metrics? . . . 34
5.3.1 Metric-Level Comparisons Across Geometries . . . 34
5.3.2 Which Metrics Carry the Curvature Gain? . . . 36
5.3.3 Metric Dependence . . . 38
5.3.4 Metrics in the Embedding and in the Visualisation . . . 40
5.4 Can Curvature Be Tuned as Its Own Hyperparameter? . . . 42
5.4.1 Controlling the Curvature . . . 42
5.4.2 Qualitative Changes Across Curvature Magnitudes . . . 43
5.5 Can Auxiliary Global Loss Terms Improve the Visualisations? . . . 44
5.5.1 Comparison Settings . . . 44
5.5.2 Overall Gains from Each Auxiliary Loss . . . 44
5.5.3 Global Improvements and Local Costs . . . 46

6 Discussion **48**
6.1 Interpretation of the Results . . . 48
6.1.1 Curvature Matching . . . 49
6.1.2 Which Metrics Curvature Moves . . . 50
6.1.3 Curvature as a Hyperparameter . . . 51
6.1.4 Auxiliary Losses . . . 51
6.1.5 Answer to the Main Question . . . 52
6.2 Limitations . . . 52
6.3 Future Work . . . 53
6.4 Reflection on Research in the Broader Computer Science Field . . . 54
6.5 Reflection on Implications and Applicability . . . 55
6.6 Usage of AI . . . 56

7 Conclusion **57**

8 Exploratory research **59**
8.1 Defining the search bounds . . . 59
8.2 Does the Estimated Curvature Track the Data's Geometry? . . . 61

References **63**

A Unbounded Quality Metrics Across Curvature **66**

B Derivation of Parallel Transport on the Sphere **68**

C Centroid Alignment via Householder Reflection **70**

1 Introduction

A single RNA assay produces expression levels for tens of thousands of genes, a clinical record carries hundreds of variables per patient, and a document encoded by a contemporary language model is a vector of several hundred numbers. Each measurement is a separate dimension, and any interesting patterns, such as clusters of similar observations, isolated outliers, or gradual transitions between groups, live in a space of far more dimensions than humans can directly analyse.

The gap between data dimensionality and human visual capacity creates a basic difficulty for data exploration. A two-dimensional scatter plot shows two variables, and a three-dimensional plot with colour and point size can show five. Beyond that, direct inspection becomes impossible: there is no analogous chart for a hundred dimensions, let alone ten thousand. Yet many analyst questions, such as how many groups are present and which observations stand apart, are most naturally answered by visual inspection rather than by computing summary statistics.

Dimensionality reduction is the standard remedy: each high-dimensional point is mapped to a low-dimensional coordinate, typically in two dimensions, chosen so that points close in the original space remain close in the embedding and points far apart stay far apart. The resulting figure can then be inspected visually, while retaining some confidence that the relationships seen in the visualisation reflect those present in the data. Methods such as t-distributed stochastic neighbour embedding (t-SNE) [1] and uniform manifold approximation and projection (UMAP) [2] have become standard tools for this purpose, producing visualisations that reveal local geometry and cluster structure across a wide range of domains.

t-SNE and UMAP almost universally embed data into Euclidean space. Embedding into Euclidean space is rarely questioned, yet it carries an implicit assumption: that the structure of the data is best captured by a flat geometry. For real-world datasets, the flat-geometry assumption may not hold. Hierarchical data, such as taxonomies or organisational trees, exhibit exponential growth in the number of nodes with depth, a property that hyperbolic space, whose volume expands exponentially with radius, captures naturally but Euclidean space cannot [3]. On the other hand, data with cyclic or periodic structure, such as seasonal patterns or angular measurements, may be better represented on a sphere, where any two geodesics intersect and the angles of a triangle exceed 180 degrees. When the geometry of the embedding space does not match the intrinsic geometry of the data, the resulting projection can suffer distortion, loss of structure, and visual artefacts that mislead the analyst [4].

Recent work in representation learning has demonstrated significant benefits from moving beyond Euclidean geometry. Hyperbolic embeddings achieve lower distortion for hierarchical graphs [5,6], mixed-curvature product spaces improve embeddings for heterogeneous data [7], and there is a growing consensus that non-Euclidean geometries deserve a central role in machine learning [4,8]. These gains, though, are measured against downstream-task criteria such as link-prediction accuracy or classification loss, not against the visual criteria an analyst applies when reading a two-dimensional scatter plot.

An analyst using hierarchical, cyclic, or otherwise non-flat data faces a practical question: does a non-Euclidean embedding reveal more structure than the standard Euclidean t-SNE, and which curvature should they choose? Answering this question requires a side-by-side comparison of embedding geometries on the same data and against the same quality metrics.

This thesis addresses that question by generalising t-SNE to constant-curvature Riemannian manifolds, encompassing Euclidean, hyperbolic, and spherical geometry, and systematically evaluating how embedding curvature influences visualisation quality. We say that an embedding geometry matches a dataset when its curvature has the same sign as the intrinsic curvature of the data: hyperbolic for hierarchical data, spherical for cyclic data, and Euclidean for flat data. The central hypothesis is that a matching embedding geometry improves visualisation quality.

The central question guiding this thesis is:

"How does the curvature of the embedding space influence the quality of two-dimensional visualisations?"

What is meant by "quality" here deserves care. No single metric captures the value of a low-dimensional visualisation: different metrics measure different aspects, such as local neighbourhood preservation, global distance preservation, and cluster separability, and an embedding that excels on one may fall short on another. We approach these contradictions by taking into account all metrics when making a comparison between visualisations.

The main research question is answered through four sub-questions.

1. **RQ1: Does matching the curvature of the dataset and embedding space improve visualisation quality?** This asks whether a geometry whose curvature sign matches that of the data gives a better corpus of visualisations than the other two geometries at all. We address the research question on synthetic datasets sampled from known constant-curvature manifolds, embedding each into all three geometries and comparing the resulting corpora of visualisations against each other.
2. **RQ2: What effect does curvature have on established metrics in visualisation literature?** Different metrics measure different properties of a visualisation, so an improvement in overall quality can conceal a loss on one of them. This question answers which geometry a visual analyst who cares particularly about a certain metric should choose, and which trade-offs that choice brings with respect to the intrinsic curvature of the dataset. As an additional experiment, the correlation between metrics is reported, since metrics that order visualisations alike carry redundant information and would otherwise count twice in the comparison.
3. **RQ3: Can curvature be tuned as its own hyperparameter?** Instead of examining metrics, in this question we inspect the qualitative properties of a visualisation. By comparing embeddings of the same dataset at low, medium and high curvature, we explore what changes in the visualisation as the curvature is increased, and we examine whether the curvature chosen for the embedding space determines how curved the embedding turns out, independently of the other hyperparameters.
4. **RQ4: Can auxiliary global loss terms improve the visualisations produced by non-Euclidean t-SNE?** Standard t-SNE optimises a single Kullback-Leibler divergence that penalises false neighbours far more strongly than missing neighbours, a choice widely held to distort global structure. Several auxiliary terms have been proposed to address this, including a global-structure KL term using a heavy-tailed kernel [9], a depth-norm regulariser from CO-SNE [10], and a centering regulariser of our own design. We ask which of these terms extend the Pareto front beyond the standard objective, and whether any such benefit depends on the embedding curvature.

1.1 Approach and contributions

The experimental core of this thesis is a systematic comparison across a curvature-geometry grid. The three embedding geometries, Euclidean, hyperbolic, and spherical, are applied to datasets with flat, hierarchical, or cyclic intrinsic structure, forming an experimental design that isolates curvature alignment as the primary variable. This thesis makes four contributions.

1. **A unified framework for curved t-SNE.** Existing non-Euclidean extensions of t-SNE-style methods target hyperbolic space only [9–12]; no prior work embeds into spherical space for visualisation. We rewrite the low-dimensional kernel with geodesic distance so a single formula spans Euclidean, hyperbolic, and spherical geometry through the sign of curvature, add a centroid-alignment correction needed to make the spherical case work, and derive the spherical parallel transport this requires (Appendix B). The resulting optimiser is built on Riemannian stochastic gradient descent [5,13] and released as an open-source implementation[1].

[1] https://github.com/tbloemen/rust-fitting-curvature

2. **A multi-objective formulation of visualisation quality.** Prior evaluations collapse a battery of quality metrics into a single averaged score [14] or a single recommended configuration [15], which implicitly fixes a weighting between metrics that measure different things. We instead treat the metrics as a multi-objective problem, using optimisation to trace out a Pareto front of visualisations per dataset and geometry, and comparing fronts under different analyst preferences rather than reducing them to one number. The correlations between the metrics support this choice: they vary more between datasets than between geometries, so no metric stands in for another in general (RQ2). This front-based comparison underlies every result in Chapter 5 and Chapter 6.
3. **Evidence that the benefit of matching curvature does not carry over to visualisations of high-dimensional data.** In representation learning, choosing an embedding space whose curvature matches the data consistently improves results, but those improvements are measured on downstream tasks such as link prediction [5,6], not on a two-dimensional picture an analyst reads. Embedding high-dimensional synthetic datasets of known curvature into all three geometries, we find that the matching geometry outperforms both alternatives on only one dataset, a set of concentric shells in hyperbolic space, and even there the other curved geometry recovers most of the improvement and most of both curved corpora is effectively flat, so the improvement comes from the dataset rather than from curvature. On two-dimensional versions of the same datasets, whose curvature can be recovered from their distances, the visualisations suggest that the matching geometry is the most faithful. The choice of geometry also changes how well the best visualisations preserve distances only slightly, and not in a way that follows matching, although strongly curved hyperbolic visualisations preserve them worse (RQ1-RQ2, Chapter 5, Chapter 6). Curved embeddings do not fail outright: raising the curvature within the spherical geometry costs no quality, even at the strongest curvature tested, and moderate hyperbolic curvature keeps hierarchies visually legible. What does not hold, on data whose distances are dominated by its dimensions rather than its curvature, is the benefit that representation learning attributes to matching curvature, once quality is judged on the visualisation itself.
4. **An evaluation of auxiliary global-structure loss terms across curvature, including a new centering regulariser.** CO-SNE's depth-norm term [10] and a heavy-tailed global-structure term [9] were both introduced for hyperbolic embeddings only; we adapt the depth-norm term to use hop distance directly, generalise both to arbitrary constant curvature, and add a centering regulariser of our own to counteract boundary degeneracy. Across the curvature grid, the depth-norm term is the only one that never loses in hyperbolic geometry, for which it was designed, while in Euclidean geometry it gains more on some datasets and loses on others, plausibly where its target describes no structure of the data (RQ4, Chapter 5, Chapter 6). The centering regulariser acts only in hyperbolic geometry, where its effect is dataset-specific: together with the global-structure term it gives the largest single gains, on the tree, but it loses on other datasets. Read together with contribution 3, this points to a broader pattern: benefits reported for a single geometry do not automatically generalise across curvature, so any such claim needs testing on the geometry it will actually be used in.

1.2 Thesis Structure

Chapter 2 positions this work among existing dimensionality reduction research, Chapter 3 develops the geometric and metric tools used throughout, Chapter 4 presents the generalised t-SNE algorithm and the experimental design, Chapter 5 reports the empirical findings, Chapter 6 interprets them and reflects on their implications, Chapter 7 answers the research questions, and Chapter 8 presents exploratory work on estimating the intrinsic curvature of a dataset.

2 Related work

Four strands of prior work inform this thesis: dimensionality reduction methods themselves, the body of work on embedding into non-Euclidean spaces, the literature on assessing visualisation quality, and multi-objective Bayesian optimisation as a tool for the hyperparameter search used in our experiments.

2.1 Dimensionality Reduction Methods

Dimensionality reduction methods differ chiefly in which structural property they preserve, and consequently in the visualisation tasks they support. We group prior work by the aspect each method emphasises: global pairwise distances, local manifold neighbourhoods, or neighbour probabilities.

Global-distance methods. Principal component analysis (PCA) [16] and multidimensional scaling (MDS) [17] both preserve global structure: PCA produces an embedding in which each output coordinate is a fixed linear combination of the input features, while MDS places points in a low-dimensional space so as to minimise a stress over all pairwise distances. Both are cheap and interpretable but cannot represent structure that lies on a curved manifold; a circle in three dimensions, for instance, collapses to a line segment under any linear projection to two. MDS additionally scales as $O(N^2)$ and compresses local neighbourhoods when the data lies on a non-linear manifold [18]. PCA is therefore most useful as a preprocessing step before a non-linear method [19], and MDS is most useful when global distances are themselves the object of interest.

Manifold-learning methods. ISOMAP [18], Locally Linear Embedding (LLE) [20], Adaptive Locally Linear Embedding (ALLE) [21], and LAMP [22] each model the data as lying on a low-dimensional manifold and reconstruct local neighbourhoods or local geodesic distances rather than global ones. ISOMAP substitutes geodesic distances along a k-nearest-neighbour graph for the straight-line Euclidean distances of MDS, recovering global structure when the manifold is connected, though it is sensitive to the choice of k [23]. LLE reconstructs each point as a linear combination of its neighbours and is light and robust, but biased under uneven sampling or noisy data [21]. ALLE replaces the fixed Euclidean metric with a learned Mahalanobis metric $d_M(x_i, x_j) = \sqrt{(x_i - x_j)^T M (x_i - x_j)}$ optimised jointly with the reconstruction weights, improving trustworthiness and continuity on datasets such as MNIST and the Olivetti faces. LAMP projects through user-selectable control points using local affine transformations and is well suited to interactive workflows. ALLE in particular responds to the same observation that motivates this thesis, that Euclidean distance is not always the right similarity measure, but it adapts the input-space metric while still embedding into Euclidean space, whereas we fix the input metric and adapt the embedding-space geometry.

Neighbour-probability methods. t-SNE [1] and UMAP [2] dominate current visualisation practice. Both convert pairwise similarities into probability distributions over neighbour relations and optimise an embedding that matches the two distributions. They excel at cluster separation and local neighbourhood preservation, which makes them the methods of choice for cluster identification, outlier detection, and neighbourhood inspection, but are correspondingly poor at preserving inter-cluster distances and cluster densities [24]. The objective is non-convex with many local minima, so random initialisation causes repeated runs on the same data to produce visually different layouts; deterministic initialisation such as PCA mitigates this [19]. UMAP is generally faster than t-SNE and is argued to better preserve global structure [25], although the comparison depends on the metric used to assess it and on the initialisation of both methods [26]. We extend t-SNE rather than UMAP because its low-dimensional kernel is an explicit function of the pairwise distances in the embedding [1], so a geodesic distance can take the place of the Euclidean one, and because the existing non-Euclidean extensions that this thesis builds on are all based on t-SNE [9,10,12].

A common thread runs through all three groups: they embed into Euclidean space. This thesis asks instead whether matching the embedding geometry to the intrinsic curvature of the data gives better visualisations.

2.2 Non-Euclidean Embeddings for Dimensionality Reduction

A small but growing body of work has extended dimensionality reduction beyond Euclidean space, motivated above all by the observation that hyperbolic geometry represents hierarchical and tree-like structure with substantially lower distortion than its flat counterpart.

CO-SNE [10] confronts the problem of visualising high-dimensional hyperbolic embeddings in two dimensions. It learns a low-dimensional hyperbolic representation that preserves both local neighbourhood structure and the geometry of the original high-dimensional embedding, so that hierarchical relationships survive the projection onto a Poincaré disk. At the heart of CO-SNE is a depth-norm regulariser, weighted by a coefficient λ_2, that pulls each embedded point's hyperbolic radius towards a target derived from its distance to a designated root. We adopt the same regulariser as one of the auxiliary loss terms in Subsection 4.1.3 and tune λ_2 jointly with the other t-SNE hyperparameters.

A complementary global-structure loss for t-SNE [9] is built around the heavy-tailed kernel $1 + d^2$ in place of the standard $\left(1 + d^2\right)^{-1}$. Inverting the kernel shifts most of its mass to large distances, so the resulting Kullback-Leibler term penalises configurations that fail to preserve global pairwise distances. The original formulation was introduced for hyperbolic embeddings of gene-expression data; we adopt it as a general-purpose auxiliary loss term applicable to embeddings of any constant-curvature geometry. A follow-up adaptive variant [11] selects the embedding curvature jointly with optimisation, complementing the search-based curvature optimisation pursued in this thesis.

A kernelised variant of t-SNE [27] replaces the Euclidean pairwise distances in the input space with distances computed via a kernel function. Kernel t-SNE encodes non-Euclidean input geometry without modifying the embedding space, admits domain-specific similarity measures, and demonstrates improved cluster separation on several benchmark datasets. Like ALLE, it adapts the input metric while keeping the embedding Euclidean, the complementary axis to the one varied in this thesis.

Non-Euclidean graph and network visualisation has been surveyed across spherical, hyperbolic, and toroidal geometries [28], with contributions categorised by geometry, type, and graph class, and open questions raised about the systematic evaluation of non-Euclidean visualisation quality. This thesis takes up one of those open questions by providing a quantitative comparison of embedding geometries for general high-dimensional data visualisation.

A corrected formulation of hyperbolic t-SNE [12] addresses an error shared by several prior works that adapted t-SNE to the Poincaré disk model. The corrected formulation applies the proper hyperbolic distance metric consistently throughout the gradient computation, resolving inconsistencies that had led to suboptimal embeddings of tree-like and network-structured data. Our hyperbolic t-SNE implementation follows this corrected formulation.

2.3 Evaluating Dimensionality Reduction

A large-scale quantitative survey of projection techniques [14] compares over forty methods across a diverse collection of datasets using a battery of quality metrics. The survey framework organises metrics by the aspect of quality they assess, including neighbourhood preservation, distance faithfulness, and cluster separability, and provides reference values that situate the relative quality scores of different methods. We adopt its grouping of metrics into local structure, distance preservation and label-aware structure, and take trustworthiness, continuity, the neighbourhood hit and Shepard goodness from it (Section 4.2).

A study of t-SNE and UMAP misuse in visual analytics documents systematic errors, in particular the temptation to read inter-cluster distances in a t-SNE projection as if they reflected true distances in the original space [24]. Drawing on a review of 136 papers and on interviews with practitioners and experts, the study traces these misuses largely to limited dimensionality reduction literacy and finds that existing guidance has not been enough to prevent them. The findings reinforce the importance of explicit evaluation metrics that surface the limitations of a given projection to its users.

New t-SNE hyperparameter guidelines [15] are derived from a large grid search across many datasets, and a neural network is trained to predict near-optimal hyperparameters from graph-theoretic features of the dataset. The empirically derived defaults match or exceed existing guidelines across a broad collection of datasets, and we adopt them as the starting configuration for our experiments.

A different route to the same problem, t-viSNE [29], is an interactive visual analytics tool that surfaces information about hyperparameter effects, neighbourhood and distance preservation, local densities, and the contribution of individual dimensions to visual patterns, helping practitioners spot when a projection is likely to mislead. Rather than reducing quality to a score, t-viSNE gives the analyst additional views with which to judge a projection, a complementary route to the metric-based comparison of this thesis. The quality measures underlying t-viSNE overlap substantially with those in our evaluation framework, which we adopt and complement with non-Euclidean variants.

2.4 Multi-Objective Bayesian Optimisation for Visualisation

Hyperparameter selection for dimensionality reduction is naturally multi-objective: trustworthiness, continuity, distance preservation, and cluster separability are not perfectly correlated, and any single scalar score, like in [14], implicitly weights them. An analyst who cares about both local neighbourhoods and global distances therefore wants the Pareto front, not a single winner picked under a hidden scalarisation.

Bayesian optimisation is a sample-efficient approach to expensive black-box objectives [30], of which a single embedding run is a representative example. ParEGO [31] lifts single-objective Bayesian optimisation into the Pareto setting by scalarising objectives with a randomly weighted Chebyshev aggregation. More recent methods, including qEHVI [32], qNEHVI [33], and MORBO [34], approximate the Pareto front directly and scale to higher-dimensional search spaces. We use qParEGO [32] because the seven-dimensional search of Subsection 4.3.3 sits well within the regime where Chebyshev scalarisation is competitive with direct hypervolume methods and substantially cheaper to fit.

Where the grid search of [15] returns one recommended configuration per dataset, we return a set of configurations that each trade the quality metrics off differently, so that an analyst can choose the one that suits their task (Subsection 4.3.1).

Background 3

This chapter develops the technical background for two theoretical concepts: t-SNE and non-Euclidean geometries. To that end, first we discuss dimensionality reduction, t-SNE and the limitations of t-SNE. After, we discuss the non-Euclidean geometries used as embedding spaces, and the scale dependence of curvature and its implications for visualisation.

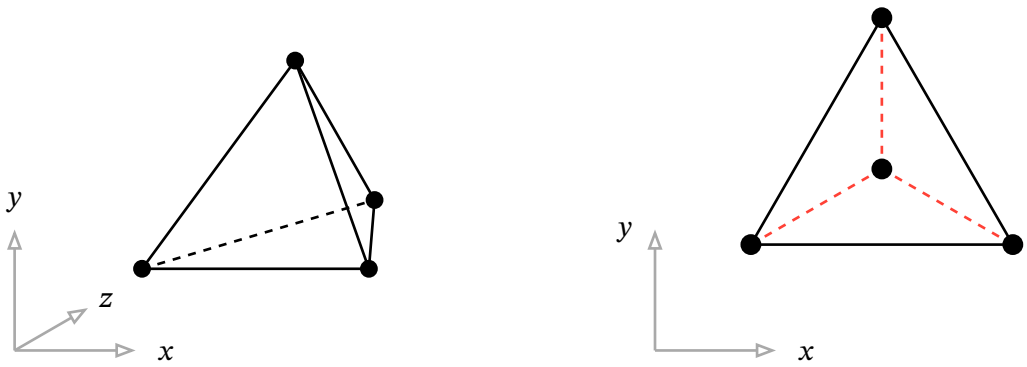
3.1 Dimensionality Reduction and Visualisation

3.1.1 The Problem of High-Dimensional Data

Many datasets in science and engineering are intrinsically high-dimensional. A single MNIST image has 784 pixel values, each a dimension of its own; a gene expression profile can run to tens of thousands of features; a typical word embedding lives in 100 to 300 dimensions. These datasets are inaccessible to human visual inspection, which is limited to two or three dimensions.

Dimensionality reduction closes this gap by embedding the data of the *original space* into a low-dimensional *embedding space* that retains the relevant structure of the original.

Three distinct objects are involved once the embedding space is allowed to be non-Euclidean, and this thesis keeps them apart throughout. The *embedding* is the point configuration itself: the set of low-dimensional points the algorithm produces, living in the embedding space. The *projection* is the map that takes an embedding to the two-dimensional page. The *visualisation* is the resulting two-dimensional figure, the object an analyst actually inspects. When the embedding space is the Euclidean plane the projection is the identity and the three coincide, which is why the distinction is normally left implicit. For a non-Euclidean embedding space they do not coincide, and the projection contributes distortion of its own (Subsection 3.2.5).



(a) Regular tetrahedron in $\mathbb{R}^3$ **(b)** Most-symmetric 2D embedding

Figure 3.1: The four vertices of a regular tetrahedron in $\mathbb{R}^3$ (left) are pairwise equidistant, so all six edges have the same length. No placement of four points in $\mathbb{R}^2$ can reproduce this: in the most-symmetric 2D arrangement (right), three vertices form an equilateral triangle and the fourth sits at the centroid, but the three centroid edges (red, dashed) are shorter than the outer edges by a factor of $\sqrt{3}$. The figure illustrates that any 2D visualisation of an intrinsically higher-dimensional configuration must distort some pairwise distances; dimensionality reduction therefore unavoidably trades faithfulness on one structural property against another, and the choice of which to preserve is what distinguishes one method from another.

There is not always a low-dimensional embedding which perfectly preserves every property of the original data at once. As a minimal example, consider the four vertices of a regular tetrahedron in three-dimensional Euclidean space: every edge has the same length. There is no placement of four points in a two-dimensional plane such that the edge lengths are equal, so any two-dimensional embedding of the regular tetrahedron must include some distortion in the edge lengths (Figure 3.1). This mismatch between the original space and

the embedding space means that any dimensionality reduction method must therefore make a choice about which aspects of the original structure to preserve and which to sacrifice.

3.1.2 t-SNE

t-distributed Stochastic Neighbour Embedding (t-SNE) [1] is among the most widely used visualisation techniques for high-dimensional data. It works by defining probability distributions over pairs of points in the original and embedding spaces and minimising the divergence between them. We focus on t-SNE because the algorithm depends on the embedding geometry only through pairwise distances, and the formulation, built on distance-dependent kernels and a Kullback-Leibler objective, therefore generalises naturally to non-Euclidean geometries: it is enough to replace Euclidean distances with geodesic distances in the embedding space.

In the original space, the affinity between points x_i and x_j is defined using a Gaussian kernel with a per-point bandwidth σ_i calibrated to achieve a target perplexity:

$$p_{j|i} = \frac{\exp\left(-\|x_i - x_j\|^2 / 2\sigma_i^2\right)}{\sum_{k \neq i} \exp\left(-\|x_i - x_k\|^2 / 2\sigma_i^2\right)}. \tag{3.1}$$

The conditional probabilities are symmetrised to obtain a joint distribution:

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N}. \tag{3.2}$$

In the embedding space, a Student-t distribution with one degree of freedom, equivalent to a Cauchy distribution, is used as the kernel:

$$q_{ij} = \frac{\left(1 + \|y_i - y_j\|^2\right)^{-1}}{\sum_{k \neq l} \left(1 + \|y_k - y_l\|^2\right)^{-1}}, \tag{3.3}$$

where $y_i \in \mathbb{R}^2$ is the embedding of the high-dimensional point x_i.

Using the heavy-tailed Student-t distribution in the embedding space, rather than a Gaussian, allows moderately distant points in the original space to be mapped further apart in the embedding. This reduces the crowding problem that arises when many points must share a limited embedding space.

The embedding is found by minimising the Kullback-Leibler divergence between the two distributions $P = \{p_{ij}\}$ and $Q = \{q_{ij}\}$:

$$\mathcal{L} = \mathrm{KL}(P \parallel Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}. \tag{3.4}$$

The KL divergence has an important asymmetry. The cost is high when p_{ij} is large but q_{ij} is small: true neighbours that the embedding has mapped far apart. The cost is low when p_{ij} is small but q_{ij} is large: non-neighbours that the embedding has placed close together. The asymmetry means t-SNE strongly preserves local neighbourhood structure but may distort global relationships.

The optimisation is performed using gradient descent. In practice, a two-phase training schedule is used: an early exaggeration phase where the high-dimensional affinities p_{ij} are multiplied by a factor, typically 4, to encourage the formation of tight clusters, followed by a main phase with the true affinities.

3.1.3 Limitations of t-SNE

For all its popularity, t-SNE has well-documented limitations [24,35]. Convergence is slow and there is no reliable way to tell when it has been reached. The perplexity, learning rate, and iteration count each influence the result strongly, so different settings can produce qualitatively different embeddings of the same data. Relative cluster sizes and densities are not preserved either: a dense and a sparse cluster can end up looking comparable in size, and two clusters that are far apart in the original space can sit at the same separation in

the embedding as two that are close. Apparent structure can be an artifact, rather than a true cluster present in the data, while genuine structure such as parallel manifolds or topological features can be flattened away.

Slow convergence, hyperparameter sensitivity, and global-distance distortion motivate the search for improvements. This thesis investigates whether the loss and distortion of global structure in particular can be mitigated by choosing an embedding geometry that better matches the data.

3.1.4 Tasks Suited to t-SNE

The local-neighbourhood emphasis of the KL objective makes t-SNE well-suited to some visualisation tasks and misleading on others. Jeon et al. [24] classify three tasks as appropriate uses of t-SNE: neighbourhood identification, outlier identification, and cluster identification. Each depends only on local structure, which t-SNE preserves well. Four further tasks are classified as inappropriate: pairwise distance investigation, class separability assessment, cluster distance investigation, and cluster density investigation. All four inappropriate tasks depend on global distance structure, which the standard objective is known to distort.

3.2 Non-Euclidean Geometry

The limitations of t-SNE in Euclidean embeddings motivate considering alternative geometries for the embedding space. Instead of a flat Euclidean space, the embedding space can be a non-Euclidean one, with non-zero curvature. These non-Euclidean spaces have some structural properties which are difficult to represent in Euclidean space, such as hierarchical growth or cyclic structure. This section introduces the geometric concepts required to understand these alternatives, followed by the projections used to visualise the resulting embeddings.

3.2.1 Riemannian Manifolds and Curvature

A Riemannian manifold is a smooth space equipped with a metric tensor that defines distances, angles, and curvature at each point. The sectional curvature K describes the intrinsic curvature of the manifold. For constant-curvature spaces considered here, K has the same value everywhere and in every direction. At $K = 0$ the space is locally Euclidean. Positive K curves it like a sphere: geodesics converge and the volume of a ball grows more slowly than in Euclidean space. Negative K curves it like a saddle, with diverging geodesics and faster-than-Euclidean volume growth. For a non-zero constant curvature K, it is useful to define the curvature radius $r = \frac{1}{\sqrt{|K|}}$. It provides the characteristic geometric scale against which distances in the manifold can be compared.

The three spaces of constant curvature, Euclidean space ($K = 0$), the sphere ($K > 0$), and hyperbolic space ($K < 0$), are the simplest Riemannian manifolds and underpin everything that follows. They are homogeneous, meaning the geometry is the same at every point, and isotropic, meaning the same in every direction; both properties make them analytically tractable.

Two spaces are in play whenever we describe a non-Euclidean geometry, and we keep them apart throughout. The *embedding space* is the space we mean: the manifold the points live on, whose dimension is the number of coordinates needed to move around within it. The *ambient space* is the surrounding flat space we need in order to write that manifold down. For Euclidean space the two coincide and a d-dimensional embedding is simply a set of points in $\mathbb{R}^d$. For the sphere and for hyperbolic space they do not: a d-dimensional embedding space is realised as a surface inside a $(d+1)$-dimensional ambient space, so ambient dimensionality is embedding dimensionality plus one, and a two-dimensional spherical or hyperbolic embedding carries three ambient coordinates per point. Any dimension quoted in this thesis is the embedding dimension unless it is explicitly called ambient.

3.2.2 Euclidean Space

Euclidean space $\mathbb{R}^d$ is the standard flat geometry. The distance between two points is the familiar ℓ_2 norm:

$$d(x, y) = \sqrt{\sum_{i=1}^{d} (x_i - y_i)^2}. \tag{3.5}$$

Euclidean space has constant curvature $K = 0$. Geodesics are straight lines, and the volume of a ball of radius r grows polynomially as r^d.

3.2.3 Hyperbolic Space

Hyperbolic space $\mathbb{H}^d$ is the unique simply connected Riemannian manifold of constant negative curvature. Several models exist; this thesis uses the hyperboloid model, also called the Lorentz model, following Nickel and Kiela [5].

In the hyperboloid model, d-dimensional hyperbolic space of curvature $K = -1/r^2$ is realised as the upper sheet of a hyperboloid in a $(d+1)$-dimensional ambient space equipped with the Lorentzian inner product:

$$\mathbb{H}^d_r = \left\{ x \in \mathbb{R}^{d+1} : \langle x, x \rangle_{\mathcal{L}} = -r^2,\ x_0 > 0 \right\}, \tag{3.6}$$

where x_0 is the time-like coordinate of x and $\langle \cdot, \cdot \rangle_{\mathcal{L}}$ is the Lorentzian inner product:

$$\langle x, y \rangle_{\mathcal{L}} = -x_0 y_0 + \sum_{i=1}^{d} x_i y_i. \tag{3.7}$$

The geodesic distance between two points is the arc length along the hyperboloid, recovered from their Lorentzian inner product:

$$d(x, y) = r \cdot \operatorname{arcosh}\left(\frac{-\langle x, y \rangle_{\mathcal{L}}}{r^2} \right). \tag{3.8}$$

The defining property of hyperbolic space is that the volume of a ball of radius r grows exponentially as $e^{(d-1)r}$, compared to the polynomial growth r^d in Euclidean space. The exponential volume growth matches the exponential branching of trees and makes hyperbolic space a natural home for hierarchical structures: the number of nodes at depth k in a tree grows as fast as the available volume. Hyperbolic embeddings of hierarchical data [3] achieve significantly lower distortion than Euclidean embeddings, even at two or three dimensions.

3.2.4 Spherical Space

The d-dimensional sphere $\mathbb{S}^d$ of curvature $K = 1/r^2$ is the set of points at distance r from the origin:

$$\mathbb{S}^d_r = \left\{ x \in \mathbb{R}^{d+1} : \|x\|^2 = r^2 \right\}. \tag{3.9}$$

The geodesic distance is the great-circle arc length, equal to the sphere radius times the angle subtended at the origin:

$$d(x, y) = r \cdot \arccos\left(\frac{\langle x, y \rangle}{r^2} \right). \tag{3.10}$$

Spherical space is compact: it has finite volume and a maximum possible distance of πr between any two points. The volume of a geodesic ball initially grows slower than in Euclidean space and eventually decreases as the ball wraps around the sphere.

Spherical geometry is well matched to data with cyclic or periodic structure. Points arranged in a loop can be embedded on a great circle without the artificial endpoints that a Euclidean embedding would impose, and seasonal data, angular measurements, and rotationally symmetric data sit naturally on a sphere.

3.2.5 Projections

To view a non-Euclidean embedding, we project it to two-dimensional Euclidean space, and the result of that projection is the visualisation. There are various projection techniques. We will discuss the Poincaré disk model for hyperbolic embeddings and the azimuthal-equidistant projection for spherical embeddings.

3.2.5.1 Hyperbolic projection

For hyperbolic embeddings, we use the Poincaré disk model to map an embedding on the hyperboloid to a bounded two-dimensional Euclidean domain. The hyperboloid has curvature $K < 0$ and curvature radius $r = \frac{1}{\sqrt{|K|}}$, and a point $x = (x_0, x_1, ..., x_n)$ satisfying Eq. (3.6) is projected according to

$$u = \frac{(x_1, ..., x_n)}{x_0 + r}, \tag{3.11}$$

The resulting points lie within the unit disk. This projection is conformal, meaning that it preserves angles, but it does not preserve distances. In particular, a point at hyperbolic distance ρ from the centre is mapped to a Euclidean radius

$$|u| = \tanh\left(\frac{\rho}{2r}\right). \tag{3.12}$$

Consequently, distances increasingly close to the boundary of the disk are compressed in the visualisation, while the boundary itself corresponds to points at infinite hyperbolic distance. The projection therefore provides a bounded visualisation of an unbounded hyperbolic space while preserving its local angular structure. This relation shows that the projected radial position is not only dependent on ρ, but also dependent on the curvature radius r. This dependence will be important when considering how curvature and embedding scale interact.

The projection is invertible: given a point u in the Poincaré disk, the corresponding point on the hyperboloid is recovered by the lift

$$\Phi(u) = \left(\frac{r\left(1 + \|u\|^2\right)}{1 - \|u\|^2}, \frac{2ru}{1 - \|u\|^2}\right), \tag{3.13}$$

which satisfies the hyperboloid constraint of Eq. (3.6) by construction.

3.2.5.2 Spherical projection

For spherical embeddings, we use the azimuthal-equidistant projection to represent the sphere in the Euclidean plane. For curvature $K > 0$, the sphere has radius $r = \frac{1}{\sqrt{K}}$. A point with polar angle θ and azimuth φ has geodesic distance $\rho = r\theta$ from the centre. The projection maps this point to

$$(u, v) = \frac{\rho}{\pi r}(\cos\varphi, \sin\varphi) = \frac{\theta}{\pi}(\cos\varphi, \sin\varphi), \tag{3.14}$$

normalising the projection so that the antipode lies on the unit circle. Unlike the Poincaré projection, this map preserves radial distances from the centre, since the projected radial coordinate is proportional to the geodesic distance ρ. However, distances in directions transverse to the radial direction are increasingly compressed with increasing distance from the centre. Thus, the azimuthal-equidistant projection preserves distances from the projection centre at the expense of angular and transverse distance distortion, with the antipode collapsed onto the boundary circle. Just as in the hyperbolic case, the projection represents distances relative to the curvature radius rather than absolute distances.

3.2.6 Scale Invariance and the Dimensionless Curvature

It is tempting to read the curvature magnitude $|K|$ of an embedding as a single measure of how strongly the geometry deviates from Euclidean space. However, the effect of a given curvature depends on the scale at which the embedding is considered. A space with a given curvature can appear nearly Euclidean over sufficiently small distances, while the same curvature can have a substantial effect over larger distances.

Consequently, $|K|$ alone is not sufficient to characterise the geometric significance of curvature. It must be considered relative to the embedding's scale.

The projection equations make the role of this scale particularly explicit. Both the hyperbolic and spherical projection equations rely on the ratio $\frac{\rho}{r}$, and thus depend on the curvature radius $r = \frac{1}{\sqrt{|K|}}$. Rewriting with this in mind, the ratio becomes $\rho\sqrt{|K|}$. For an embedding with characteristic length L, the corresponding dimensionless extent is therefore

$$L\sqrt{|K|} = \sqrt{|K|L^2} = \sqrt{\kappa}, \tag{3.15}$$

where

$$\kappa = |K|\, L^2. \tag{3.16}$$

κ expresses the extent of the embedding relative to its curvature radius, up to a square.

As the characteristic length we take the radius of gyration of the embedding,

$$L = R_g, \quad R_g^2 = \frac{1}{2N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\rho\big(y_i, y_j\big)^2, \tag{3.17}$$

where $\rho\big(y_i, y_j\big)$ is the geodesic distance between two embedded points. In Euclidean space R_g is exactly the root-mean-square distance of the points to their centroid. Because it is computed from pairwise distances only, R_g needs no reference point, so it means the same thing in all three geometries. On the sphere κ is also bounded, since no geodesic is longer than πr; points spread uniformly over the whole of $\mathbb{S}^2$ give $\kappa = \big(\pi^2/2 - 2\big)/2 \approx 1.47$.

Under the rescaling $L \mapsto cL$ and $|K| \mapsto \frac{|K|}{c^2}$, κ remains unchanged, but the curvature $|K|$ does not [11].

These projection equations show why curvature cannot be analysed independently of the scale of the embedding. In both geometries, the position of a point in the visualisation depends on its distance relative to the curvature radius. For the sphere this dependence appears explicitly through $\frac{\rho}{r}$, while for the hyperbolic space it appears through $\frac{\rho}{2r}$ in the relation $|u| = \tanh\Big(\frac{\rho}{2r}\Big)$. Since $r = \frac{1}{\sqrt{|K|}}$, the relevant dimensionless quantity is therefore $\rho\sqrt{|K|}$. For an embedding with characteristic length L, this gives $L\sqrt{|K|} = \sqrt{\kappa}$, where $\kappa = |K|L^2$. Consequently, changing $|K|$ alone does not necessarily correspond to a change in the geometry visible in the projection: its effect depends on the scale of the embedding. We therefore use κ, rather than $|K|$ alone, as the scale-invariant quantity for analysing and comparing the curvature of embeddings.

Curvature magnitude alone is insufficient for comparing the embeddings in non-Euclidean spaces. An embedding with a small curvature magnitude can exhibit strong curvature effects if its extent is large relative to the curvature radius, while a larger curvature magnitude can have little effect over a sufficiently small region. The dimensionless quantity κ therefore provides the appropriate scale-invariant measure for comparing the curvature of the embeddings considered in this thesis.

4 Methods

This chapter develops the methodology used to answer the main research question: "How does the curvature of the embedding space influence the quality of two-dimensional visualisations?" This thesis considers embeddings into three constant-curvature geometries: Euclidean space with zero curvature, spherical space with positive curvature, and hyperbolic space with negative curvature. The main research question is investigated by answering the four sub-questions posed in Chapter 1.

1. **RQ1: Does matching the curvature of the dataset and embedding space improve visualisation quality?** Synthetic datasets with known intrinsic curvature allow us to test whether embedding a dataset into a matching geometry produces better visualisations than embedding it into a geometry with a different curvature.
2. **RQ2: What effect does curvature have on established metrics in visualisation literature?** To quantify the quality of a visualisation, there are already established metrics in the visualisation field. These metrics all measure different properties of the visualisation. This research question aims to measure the effect of embedding into non-Euclidean spaces onto these metrics.
3. **RQ3: Can curvature be tuned as its own hyperparameter?** Instead of looking at metrics, we look at the qualitative properties of a visualisation. By comparing visualisations of the same dataset at low, medium and high curvature, we explore what changes as the curvature is increased. We also examine whether the curvature chosen for the embedding space determines how curved the embedding turns out, independently of the other hyperparameters.
4. **RQ4: Can auxiliary global loss terms improve the visualisations produced by non-Euclidean t-SNE?** t-SNE preserves local structure well, but is not suited for global structure. We ask which auxiliary loss terms from the literature, together with a centering term of our own, extend the Pareto front beyond the standard objective, and whether any such benefit depends on the embedding curvature.

To answer these research questions, we require visualisation corpora generated using different embedding geometries and curvature values. To produce embeddings in a non-Euclidean space, we describe the mathematical formulas necessary to extend t-SNE to such spaces. Each embedding is then projected to the plane to obtain the visualisation. Because the hyperparameters of t-SNE can substantially affect the resulting visualisation, comparing arbitrary hyperparameter configurations could conflate the effect of curvature with differences in quality. We therefore aim to compare high-quality visualisations for each combination of dataset and embedding geometry. Since visualisation quality is multidimensional, we formulate this selection of high-quality visualisations as a multi-objective optimisation problem using established visualisation metrics. The collection of non-dominated visualisations that the optimisation procedure produces for a given dataset, geometry and loss setting is the *visualisation corpus*. We build one corpus per combination of dataset, geometry and loss setting, and every comparison in this thesis is a comparison between corpora.

Therefore, to create the visualisation corpora, we follow three steps. First, we create the infrastructure necessary to generate embeddings in Euclidean, spherical and hyperbolic spaces, including the auxiliary loss terms to restore global structure. Next, we select established metrics from the visualisation literature. Finally, we employ multi-objective optimisation to produce high-quality visualisations. Figure 4.1 summarises these three steps.

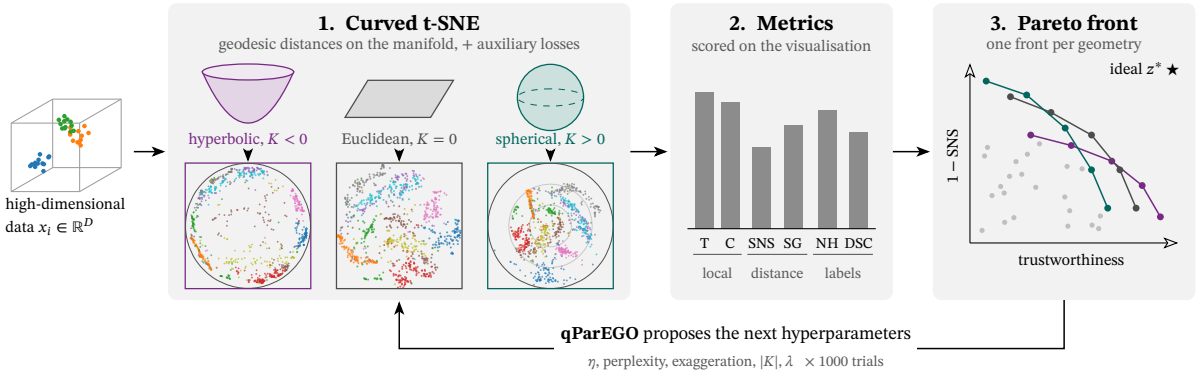


Figure 4.1: Overview of the method. (1) t-SNE is generalised to hyperbolic, Euclidean and spherical embedding spaces by computing the low-dimensional affinities from geodesic distances, optionally with auxiliary loss terms for global structure (Subsection 4.1.1). Each embedding is projected to a two-dimensional visualisation. (2) Six metrics score the visualisation on local structure (trustworthiness, continuity), distance preservation (scale-normalised stress (SNS), Shepard goodness) and label structure (neighbourhood hit, distance consistency) (Section 4.2). (3) qParEGO uses these scores to propose the next hyperparameters; after 1000 trials the non-dominated visualisations form the Pareto front, which is the visualisation corpus of one dataset, geometry and loss setting. Fronts of different geometries are compared with the R_2 indicator (Subsection 4.3.1, Subsection 4.3.4). The fronts in (3) are illustrative.

4.1 Infrastructure

To generate embeddings in non-Euclidean spaces, we extend t-SNE in three respects. First, the low-dimensional similarity measure is generalised from Euclidean distance to geodesic distance on an arbitrary constant-curvature manifold. Second, the resulting objective is optimised on the manifold using Riemannian gradient descent. Finally, auxiliary loss terms are implemented to encourage preservation of global structure.

4.1.1 Generalised t-SNE on Constant-Curvature Manifolds

Given an embedding space $\mathcal{M}$ with curvature K, we define the low-dimensional kernel as:

$$q_{ij} = \frac{\left(1 + d_{\mathcal{M}}(y_i, y_j)^2\right)^{-1}}{\sum_{k \neq l} \left(1 + d_{\mathcal{M}}(y_k, y_l)^2\right)^{-1}}, \tag{4.1}$$

where $y_i \in \mathcal{M}$ is the embedded point i and $d_{\mathcal{M}}$ is the geodesic distance on $\mathcal{M}$. The generalisation therefore affects only the low-dimensional affinity calculation; the high-dimensional affinities remain identical to standard t-SNE. The loss function is the same KL divergence as in standard t-SNE.

The curvature K determines the curvature radius of the embedding space. Setting $K = 0$ recovers Euclidean t-SNE. Setting $K < 0$ embeds into hyperbolic space using the hyperboloid model with radius $r = 1/\sqrt{|K|}$. Setting $K > 0$ embeds onto a sphere of radius $r = 1/\sqrt{K}$.

For each geometry, the geodesic distance between embedded points y_i, y_j takes a different form.

$$d(y_i, y_j) := \begin{cases} r \cdot \operatorname{arcosh}\left(\frac{-\langle y_i, y_j \rangle_{\mathcal{L}}}{r^2}\right) & \text{if } K < 0 \\ \|y_i - y_j\|_2 & \text{if } K = 0, \\ r \cdot \arccos\left(\frac{\langle y_i, y_j \rangle}{r^2}\right) & \text{if } K > 0 \end{cases} \tag{4.2}$$

where $\langle \cdot, \cdot \rangle_{\mathcal{L}}$ is the Lorentzian inner product, the points on the hyperbolic manifold satisfy the hyperboloid constraint $\langle y, y \rangle_{\mathcal{L}} = -r^2$, and the points on the spherical manifold satisfy the sphere constraint $\|y\|^2 = r^2$. These distance functions provide the only geometry-dependent component of the t-SNE affinity calculation. All further optimisation operations are defined with respect to the selected embedding space.

4.1.1.1 Centroid Alignment

For spherical embeddings, the projection is defined relative to a pole. However, the t-SNE optimiser has no awareness of this pole: it places the embedding wherever the KL divergence objective directs, so the centroid of that embedding may point in an arbitrary direction in ambient space. The azimuthal-equidistant projection is defined relative to a pole, and its distortion characteristics hold only when the data is centred at that pole. If the data centroid is far from the pole, different parts of the embedding experience different amounts of distortion, making the visualisation misleading.

To correct this, we rotate the entire embedding so that its centroid direction aligns with the projection pole before projecting. Since rotations are isometries of S^d, this operation preserves all pairwise geodesic distances and neighbourhood relationships exactly. The rotation is implemented as a Householder reflection that maps the normalised ambient-space centroid to the projection pole in a single step; the construction is given in Appendix C.

4.1.2 Riemannian Optimisation

The t-SNE objective is defined in terms of distances between points in the embedding space, so its optimisation must respect the geometry of that space. Applying standard gradient descent directly to the ambient coordinates does not in general preserve the manifold constraint. On the sphere, for instance, a step $y - \eta a$ along a tangent direction a, which satisfies $\langle y, a\rangle = 0$, has $\|y - \eta a\|^2 = r^2 + \eta^2 \|a\|^2$, so every non-zero step moves the point off the sphere. On the hyperboloid the same computation with the Lorentzian inner product gives $-r^2 + \eta^2\langle a, a\rangle_{\mathcal{L}}$, where $\langle a, a\rangle_{\mathcal{L}} > 0$ for every non-zero tangent vector. We therefore use Riemannian gradient descent (RGD), which performs each update along the manifold [13].

The optimisation, following Nickel and Kiela [5] and Gu et al. [7], proceeds by first obtaining a gradient direction in the tangent space of the current point. The tangent space provides a locally Euclidean space in which gradient descent can be performed, while the exponential map then moves the point along the corresponding geodesic on the manifold. When momentum is used, the previous momentum must additionally be transported from the tangent space at the previous point to the tangent space at the current point. The three operations, tangent-space projection, exponential map and parallel transport, depend on the embedding geometry; they are defined below for a point y on the manifold, after which Section 4.1.2.4 combines them into one iteration.

Points are initialised using PCA, following the standard t-SNE initialisation procedure [1]. For non-Euclidean embedding spaces, the ambient coordinates obtained from PCA are converted to valid manifold points by computing the constraint coordinate from the spatial coordinates.

4.1.2.1 Tangent Space Projection

The ambient gradient obtained by automatic differentiation is not necessarily a valid direction of motion in the embedding space. It must first be converted to a tangent vector at the current point.

$$\mathrm{proj}_y(a) = \begin{cases} g^{-1}a + \frac{\langle y, g^{-1}a\rangle_{\mathcal{L}}}{r^2} y & \text{if } K < 0 \\ a & \text{if } K = 0 \\ a - \frac{\langle a, y\rangle}{r^2} y & \text{if } K > 0 \end{cases} \tag{4.3}$$

Here, a is the ambient Euclidean gradient. For the hyperboloid, $g^{-1} = \mathrm{diag}(-1, +1, \ldots, +1)$ converts the ambient gradient according to the Lorentzian metric. The second term removes the component normal to the hyperboloid, yielding a vector satisfying the tangent-space constraint $\langle y, v\rangle_{\mathcal{L}} = 0$. For Euclidean space, the tangent space coincides with the ambient space and no projection is required. For the sphere, the subtraction removes the component parallel to y, leaving a vector orthogonal to y.

4.1.2.2 Exponential Map

Once a tangent-space update vector has been obtained, the exponential map moves the point along the geodesic defined by that vector [5,7]. Unlike ordinary vector addition in Euclidean space, this operation preserves the manifold constraint.

$$\mathrm{Exp}_y(v) = \begin{cases} \cosh\left(\frac{\|v\|_{\mathcal{L}}}{r}\right) y + \sinh\left(\frac{\|v\|_{\mathcal{L}}}{r}\right) \frac{rv}{\|v\|_{\mathcal{L}}} & \text{if } K < 0 \\ y + v & \text{if } K = 0 \\ \cos\left(\frac{\|v\|}{r}\right) y + \sin\left(\frac{\|v\|}{r}\right) \frac{rv}{\|v\|} & \text{if } K > 0 \end{cases} \tag{4.4}$$

The exponential map moves y along the geodesic in the tangent direction v. The Euclidean case reduces to ordinary vector addition, while the spherical and hyperbolic cases follow their respective geodesics. The norm is taken with respect to the metric of the embedding space.

4.1.2.3 Parallel Transport

Momentum-based optimisation requires transporting the momentum vector from the tangent space at the old point to the tangent space at the new point. The transport is done via parallel transport along the geodesic connecting the two points.

$$\Gamma_{y \to y'}(v) = \begin{cases} v + \frac{\langle y', v \rangle_{\mathcal{L}}}{r^2 - \langle y, y' \rangle_{\mathcal{L}}} (y + y') & \text{if } K < 0 \\ v & \text{if } K = 0 \\ v - \frac{\langle y', v \rangle}{r^2 + \langle y, y' \rangle} (y + y') & \text{if } K > 0 \end{cases} \tag{4.5}$$

Parallel transport moves a tangent vector from the tangent space at y to that at y' along the geodesic from y to y'. In Euclidean space, all tangent spaces coincide and transport is therefore the identity. For the hyperboloid, the equation is provided in earlier work [36]. For the sphere, an analogous formulation is derived in Appendix B. After transport, the vector is re-projected onto the tangent space at y' to limit numerical errors.

4.1.2.4 Iteration

For an embedded point y_i, let a_i denote the ambient Euclidean gradient of the loss with respect to y_i, obtained through automatic differentiation, and let v_i be its momentum vector, which lies in the tangent space at the previous position y_i^{old}. Starting from the initialisation, each optimisation iteration updates every point as follows (Figure 4.2):

1. Convert the ambient gradient into a tangent-space gradient: $\nabla_i = \mathrm{proj}_{y_i}(a_i)$.
2. Transport the previous momentum to the tangent space at y_i: $\tilde{v}_i = \Gamma_{y_i^{\text{old}} \to y_i}(v_i)$.
3. Update the momentum: $v_i \leftarrow \mu \cdot \tilde{v}_i - \eta \cdot \nabla_i$, where μ is the momentum coefficient and η is the learning rate.
4. Move along the manifold using the exponential map: $y_i^{\text{new}} = \mathrm{Exp}_{y_i}(v_i)$.

After each exponential-map update, the points are renormalised to the manifold to limit numerical drift.

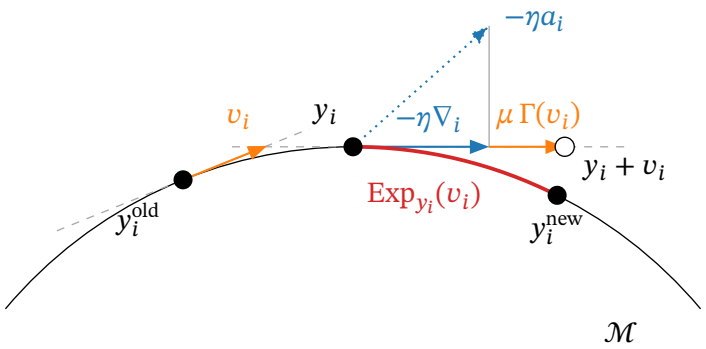


Figure 4.2: One iteration of Riemannian gradient descent, drawn on a cross-section of the sphere $\mathcal{M}$, so that each tangent space is a line (dashed). The ambient descent direction $-\eta a_i$ is projected onto the tangent line at y_i, and the previous momentum v_i is transported from y_i^{old} to y_i and damped by μ. Their sum v_i is a tangent vector: adding it to y_i leaves the manifold (open circle), whereas the exponential map follows the geodesic of the same length to y_i^{new}.

4.1.3 Auxiliary Loss Terms for Global Structure

The asymmetry of the standard KL objective is what makes t-SNE so good at preserving local neighbourhoods and so prone to distorting global distances and cluster sizes. A line of recent work supplements the KL loss with extra terms that target global structure directly. We evaluate three such auxiliary terms alongside the standard objective: a centering term of our own design, a global-structure term from Zhou and Sharpee [9], and a depth-norm term from CO-SNE [10]. The total loss is

$$\mathcal{L} = \mathrm{KL}(P \parallel Q) + \lambda_{\mathrm{cen}}\mathcal{L}_{\mathrm{cen}} + \lambda_{\mathrm{glb}}\mathcal{L}_{\mathrm{glb}} + \lambda_{\mathrm{norm}}\mathcal{L}_{\mathrm{norm}}, \tag{4.6}$$

with non-negative weights $\lambda_{\mathrm{cen}}, \lambda_{\mathrm{glb}}, \lambda_{\mathrm{norm}}$ that are zero in the standard objective.

4.1.3.1 Centering Loss

The centering loss penalises geodesic spread of the embedding from the origin of the embedding space. For a hyperbolic embedding $\{y_i\}_{i=1}^N$ on the hyperboloid of radius r, write $\rho_i = r \cdot \operatorname{arcosh}\left(\frac{y_{i,0}}{r}\right)$ for the geodesic distance from the origin. We use the mean of these radii, so that the penalty grows linearly with how far the cloud has drifted outward and applies a uniform inward force regardless of where a point sits:

$$\mathcal{L}_{\mathrm{cen}} = \frac{1}{N}\sum_{i=1}^{N} \rho_i. \tag{4.7}$$

The loss is defined through the hyperbolic radius and acts only in the hyperbolic geometry; for $K \geq 0$ the implementation contributes zero loss and zero gradient. On the sphere all points already have the same ambient radius r, so there is no radial degree of freedom to regularise, and in the Euclidean geometry there is no outward drift to counteract. In hyperbolic space the loss counteracts the tendency of optimisation to push points outward toward the boundary of the Poincaré disk, where distances diverge and visualisations degenerate into a ring. Because the embedding is recentred on its mean after every optimisation step, ρ_i measures the spread of the embedding about its centre rather than its position, and the penalty acts as a soft constraint on the extent of the embedding.

4.1.3.2 Global-Structure Loss

The standard Student-t kernel decreases with distance, giving local pairs larger weights than distant pairs [9]. A complementary loss based on an inverted heavy-tailed kernel emphasises large distances:

$$\hat{p}_{ij} = \frac{1 + d_X(x_i, x_j)^2}{\sum_{m \neq n}\left(1 + d_X(x_m, x_n)^2\right)}, \quad \hat{q}_{ij} = \frac{1 + d_{\mathcal{M}}(y_i, y_j)^2}{\sum_{m \neq n}\left(1 + d_{\mathcal{M}}(y_m, y_n)^2\right)}, \tag{4.8}$$

where d_X is the distance in the original space and $d_{\mathcal{M}}$ is the geodesic distance in the embedding space. The global loss is the KL divergence between these distributions:

$$\mathcal{L}_{\text{glb}} = \mathrm{KL}\big(\hat{P} \parallel \hat{Q}\big), \tag{4.9}$$

where $\hat{P} = \{\hat{p}_{ij}\}$ and $\hat{Q} = \{\hat{q}_{ij}\}$ are the joint distributions assembled from the kernels above.

Because the unnormalised kernel grows with d^2, $\hat{p}_{ij}$ is large precisely when x_i and x_j are far apart; matching $\hat{q}_{ij}$ therefore encourages the embedding to preserve large-scale distance structure. Adding this loss to the standard t-SNE objective therefore captures both local neighbourhoods and global geometry, which should reduce the local bias of the original formulation.

4.1.3.3 Depth-Norm Loss

CO-SNE [10] augments hyperbolic t-SNE with a per-point regulariser that pulls each embedded point's norm toward a target value derived from the input. For datasets given as feature vectors, the target is the high-dimensional norm:

$$\mathcal{L}_{\text{norm}} = \frac{1}{N}\sum_{i=1}^{N}\Big(\|y_i\|^2 - \|x_i\|^2\Big)^2, \tag{4.10}$$

where x_i is the original-space vector of point i and y_i its embedding.

For graph- or tree-structured datasets such as WordNet, no input feature vectors exist; we use the hop distance h_i from each node i to a designated root node as a direct target. Using the hop distance as a direct target contrasts with the original CO-SNE workflow, which first learns a high-dimensional hyperbolic embedding of the graph, such as a Poincaré embedding [3], and then visualises that embedding by matching the radii of the 2D points to the radii of their high-dimensional counterparts. We instead skip the intermediate embedding and map hop distances directly to radii in the embedding space. This is a modification of the original CO-SNE approach. For a hyperbolic embedding of curvature radius $r = 1/\sqrt{|K|}$, the target Poincaré-disk radius is the radius at which an ideal hyperbolic tree embedding would place a node at geodesic distance h_i from the origin,

$$\tau_i = \tanh(h_i/(2r)), \tag{4.11}$$

and the loss matches each embedded point's Poincaré-disk radius $|u_i|$, the projection of y_i under Subsection 3.2.5, to this target, $\sum_i (|u_i| - \tau_i)^2$. For Euclidean embeddings the target is the hop distance itself, $\tau_i = h_i$.

This formulation preserves the depth of each node, which is the property most directly aligned with hierarchy and is precisely what the standard KL loss is least sensitive to. On Euclidean embeddings the loss reduces to matching plain Euclidean norms, which is comparatively weak. On spherical embeddings the term is undefined because all points share a common radius, so it is disabled.

4.2 Evaluation Metrics

No single number captures the quality of a visualisation. We adopt the framework of Espadoto et al. [14], who organise dimensionality-reduction quality along three axes: local structure preservation, distance preservation, and label-aware structure. We supplement it with cluster-separation metrics surveyed by Behrisch et al. [37].

For non-Euclidean embeddings, metrics that depend on distances or neighbourhoods can be computed on two distinct surfaces. The *manifold variant* uses geodesic distances in the embedding space, and therefore evaluates the embedding according to its intrinsic geometry. The *projected variant* uses Euclidean distances in the visualisation the analyst inspects, after applying an azimuthal-equidistant projection from the sphere or a Poincaré-disk projection from the hyperboloid. For Euclidean embeddings, the manifold and projected variants of each metric are identical.

Because this is ultimately a visualisation tool, and not an embedding tool, we have chosen to report the metrics after projecting. This is important because projecting strongly curved embeddings introduces distortion. As discussed in Subsection 3.2.6, the magnitude of projection distortion depends on the dimensionless extent of the embedding, $\sqrt{\kappa} = L\sqrt{|K|}$. The consequence is that embeddings with a larger κ are expected to

show a larger discrepancy between their manifold and projected metrics, while embeddings with a κ close to zero approach the Euclidean case.

Every metric below is therefore computed on the visualisation rather than on the embedding. We write x_i for point i in the original space, y_i for its position in the embedding, and u_i for its position in the visualisation, so that u_i is the image of y_i under the projection of Subsection 3.2.5. Unless otherwise stated, for each metric a higher value is better.

4.2.1 Local Structure Preservation

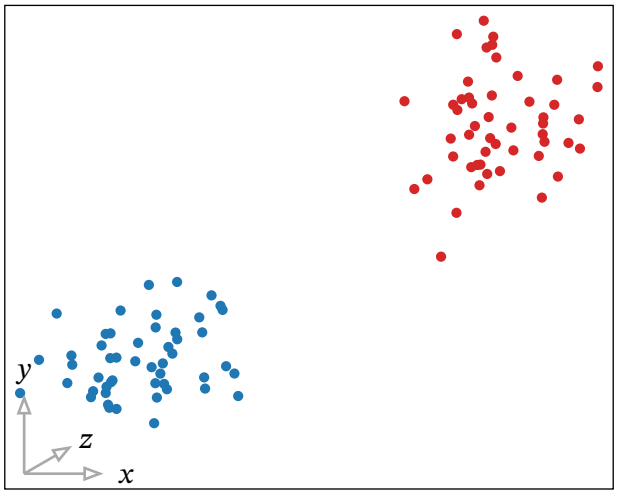
Local structure metrics assess how well nearest-neighbour relationships in the original space are maintained in the visualisation.

Trustworthiness [38] measures the proportion of points in the k-nearest neighbourhood of a point in the visualisation that are also neighbours in the original space. High trustworthiness means that if two points appear close in the visualisation, they are genuinely close in the original space. Formally:

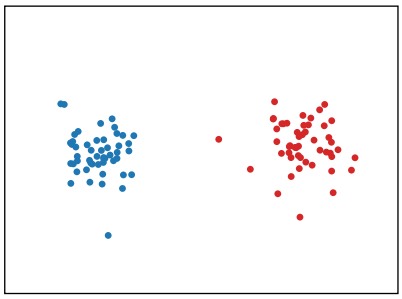
$$T(k) = 1 - \frac{2}{Nk(2N - 3k - 1)} \sum_{i=1}^{N} \sum_{j \in U_k(i)} (\pi(i, j) - k), \tag{4.12}$$

where $U_k(i)$ is the set of points that are among the k nearest neighbours in the visualisation but not in the original space, and $\pi(i, j)$ is the rank of point j among the neighbours of i in the original space.

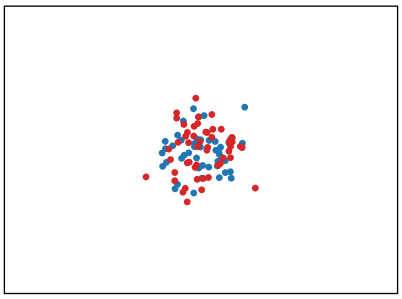
Continuity is the symmetric complement of trustworthiness: it penalises original neighbours that are torn apart in the visualisation, using the same formula with the roles of the two spaces swapped. Trustworthiness captures the absence of false neighbours: unrelated points appearing close in the visualisation. Continuity captures the absence of missing neighbours: related points pulled apart in the visualisation.



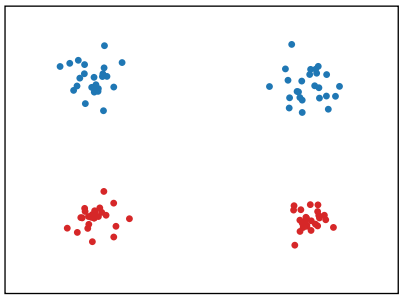
(a) High-dimensional source



(b) Faithful embedding



(c) Low trustworthiness



(d) Low continuity

Figure 4.3: Extremes of the local-structure metrics. The high-dimensional source (a) is two well-separated Gaussian clusters in $\mathbb{R}^3$, and (b) embeds them faithfully. Each of the other 2D panels fails a different metric while keeping the others comparatively intact. Collapsing both clusters onto each other (c) fabricates cross-cluster neighbours without tearing within-cluster ones, lowering trustworthiness. Splitting each cluster into two halves (d) tears original neighbourhoods apart without introducing false ones, lowering continuity.

4.2.2 Distance Preservation

Whereas the local metrics consider only neighbour identities, distance-preservation metrics compare the full pairwise distance structure between the original space and the visualisation.

A commonly used metric is normalised stress, a measure of deviation between the pairwise distances in the original space and those in the visualisation. Smelser et al. [39] show that the standard formulation is sensitive to uniform scaling: stretching or shrinking a visualisation changes its stress despite not altering any of its structural properties. We therefore use a scale-invariant variant: α is the least-squares scaling that best aligns visualisation distances to original-space distances, and scale-normalised stress (SNS) is the residual stress remaining after that scaling has been applied:

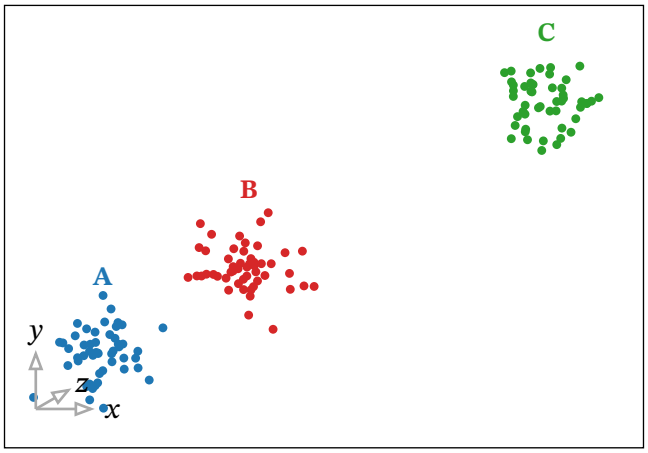
$$\alpha = \frac{\sum_{i<j} d_X(x_i, x_j) \cdot d_Y(u_i, u_j)}{\sum_{i<j} d_Y(u_i, u_j)^2}, \quad \mathrm{SNS} = \sqrt{\frac{\sum_{i<j} \left(d_X(x_i, x_j) - \alpha \cdot d_Y(u_i, u_j)\right)^2}{\sum_{i<j} d_X(x_i, x_j)^2}}, \tag{4.13}$$

Here $d_X(x_i, x_j)$ is the original-space distance and $d_Y(u_i, u_j)$ the distance in the visualisation, and α is the scaling factor that best aligns all visualisation distances with the corresponding original-space distances. SNS takes values in $[0, 1]$, where, unlike the other metrics, zero is best.

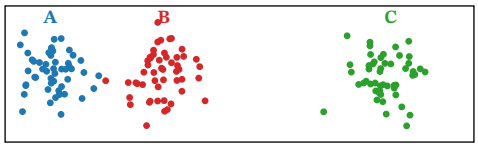
Shepard goodness is the Spearman rank correlation between the pairwise distances in the original space and those in the visualisation. That correlation measures the strength of a monotonic, not necessarily linear, relationship between two variables, so unlike SNS it is invariant to any monotonic transformation of the distances. Let π_{ij}^X and π_{ij}^Y be the fractional ranks of the distance between points i and j in the original space and in the visualisation respectively, among all $m = \frac{N(N-1)}{2}$ distinct pairs. Fractional ranks assign tied distances their average rank. Shepard goodness is then computed as the Pearson correlation between these two rank vectors:

$$S_G^{\text{raw}} = \frac{\sum_{i<j}\left(\pi_{ij}^X - \overline{\pi^X}\right)\left(\pi_{ij}^Y - \overline{\pi^Y}\right)}{\sqrt{\sum_{i<j}\left(\pi_{ij}^X - \overline{\pi^X}\right)^2 \sum_{i<j}\left(\pi_{ij}^Y - \overline{\pi^Y}\right)^2}}. \tag{4.14}$$

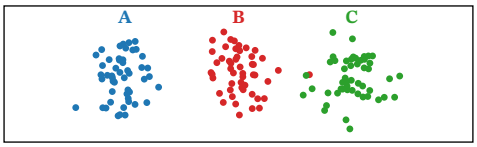
The raw coefficient lies in $[-1, 1]$. A negative coefficient means that the rank order is inverted rather than preserved, which is no more useful for a visualisation than the absence of any correlation, so we clip it at zero: $S_G = \max(S_G^{\text{raw}}, 0)$. Shepard goodness therefore takes values in $[0, 1]$, with 1 indicating perfect rank-order preservation and 0 indicating no preservation or an inverted rank order. Where SNS compares absolute distances, Shepard goodness ignores them entirely and only checks whether their rank order is preserved, providing a complementary scalar view of the Shepard diagram [14].



(a) High-dimensional source (d_X: $AB < BC < AC$)



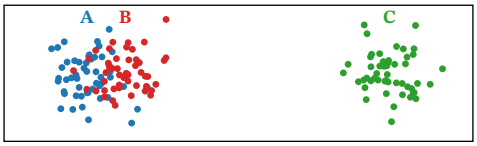
(b) Low SNS, high Shepard



(c) High SNS, moderate Shepard



(d) High SNS, low Shepard



(e) High SNS, high Shepard

Figure 4.4: Extremes of the distance-preservation metrics, scale-normalised stress (SNS) and Shepard goodness. The high-dimensional source (a) is three Gaussian clusters in $\mathbb{R}^3$ arranged as an asymmetric chain: $d(A,B) \approx 1$, $d(B,C) \approx 2$, $d(A,C) \approx 3$ (ratio $1:2:3$). The faithful embedding (b) preserves these proportions. The equidistant collapse (c) places all three clusters at equal spacing, nullifying the $AB:BC$ ratio and therefore giving high SNS, yet the rank order $AB < BC < AC$ is nearly intact so Shepard goodness stays moderate. Swapping A and C (d) makes AC the closest pair in 2D, fully inverting the rank order (Spearman $= -1$) and also breaking SNS. The non-linear spacing (e) compresses AB and expands BC, keeping the rank order perfectly ($\rho = 1$) while the proportions remain wrong, illustrating that high Shepard goodness does not necessarily preserve global structure as opposed to SNS.

4.2.3 Label-Aware Structure

When ground-truth class labels are available, additional metrics can verify that the visualisation respects the labelled structure. In this section, we treat each class as a single cluster. We use neighbourhood hit from the Espadoto survey [14] and four cluster-separation measures from the Behrisch survey [37].

Neighbourhood hit [14] is the average fraction of a point's k nearest neighbours in the visualisation that share its class label:

$$M_{\mathrm{NH}} = \frac{1}{N}\sum_{i=1}^{N} \frac{\left|\left\{j \in N_k(i) : \ell_j = \ell_i\right\}\right|}{k}, \tag{4.15}$$

where $N_k(i)$ are the k nearest neighbours of point i in the visualisation and ℓ_i is its class label. M_{NH} is high when classes form well-separated neighbourhoods in the visualisation and low when they are intermixed.

Neighbourhood hit is a local measure: it asks only about the immediate neighbours of each point, so it cannot distinguish classes that are cleanly separated from classes that merely fail to interleave. Distance consistency [40], as surveyed in [37], supplies the complementary global view by asking whether each point falls nearest to the centroid of its own class:

$$\mathrm{DSC} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\left[\|u_i - \mu_{\ell_i}\| < \min_{c\neq\ell_i}\|u_i - \mu_c\|\right], \tag{4.16}$$

where u_i is the position of point i in the visualisation, μ_c is the centroid of class c, and $\mathbb{1}[\cdot]$ is the indicator function. DSC is the fraction of points that are closer to their own class centroid than to any other, so it lies in $[0, 1]$ by construction and needs no neighbourhood size or clustering step. Because it compares each point against every class centroid rather than against its neighbours, it responds to whether classes occupy separated regions of the visualisation as a whole. Its weakness is the reliance on centroids: a class that wraps around another, or is otherwise strongly non-convex, can score poorly even when it is perfectly separable.

The cluster density measure (ClDM) of Albuquerque et al., as surveyed in [37], is a cheaper alternative computed directly from the point coordinates:

$$\mathrm{ClDM} = \frac{1}{C}\sum_{c=1}^{C}\sum_{c'=c+1}^{C}\frac{d_{cc'}^2}{s_c s_{c'}}, \tag{4.17}$$

where C is the number of classes, $d_{cc'}$ is the Euclidean distance between the centroids of classes c and c' in the visualisation, and s_c is the mean distance from points in class c to their centroid. Compact clusters with large inter-centroid distances yield high ClDM.

The Davies-Bouldin index [41] is a metric which measures the similarity between clusters by dividing the intra-cluster scatter by the inter-cluster distance. A lower index is better, since cluster tightness and separation is encouraged. The Davies-Bouldin ratio, as surveyed in [37], addresses a limitation of using the raw Davies-Bouldin index as a quality score: a high-dimensional dataset whose classes are poorly separated to begin with will produce a high Davies-Bouldin index in 2D regardless of how good the visualisation is. The fix is to take the ratio of the index computed on the original-space distances to the index computed on the visualisation:

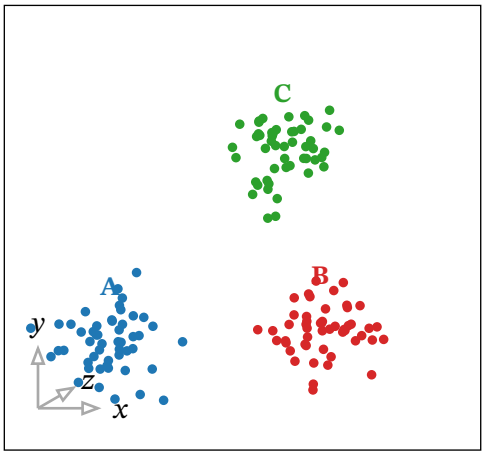
$$\text{DB-ratio} = \frac{\mathrm{DB}_X}{\mathrm{DB}_Y}, \tag{4.18}$$

so the metric rewards visualisations that preserve or improve the relative cluster separability of the original data.

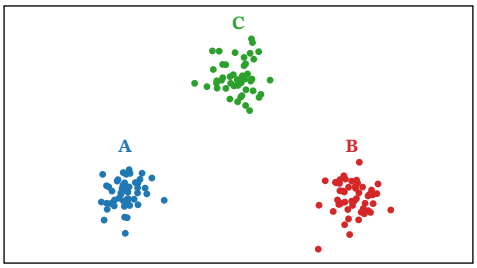
The Dunn index is the ratio of the minimum inter-cluster distance to the maximum intra-cluster diameter:

$$\mathrm{Dunn} = \frac{\min_{\ell_i\neq\ell_j} d_Y(u_i, u_j)}{\max_{\ell_i=\ell_j} d_Y(u_i, u_j)}. \tag{4.19}$$

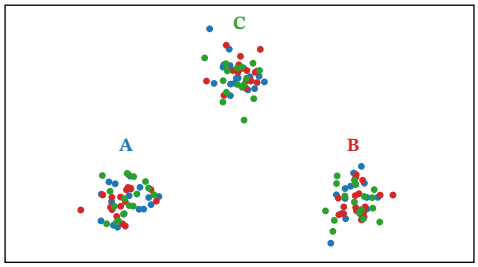
The numerator is the smallest visualisation distance between points of different classes, and the denominator is the largest visualisation distance between points of the same class. Higher values indicate clusters that are compact and well-separated.



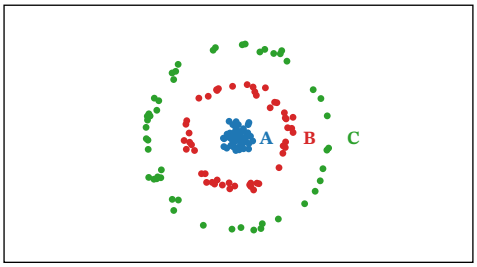
(a) High-dimensional source: three labelled clusters in $\mathbb{R}^3$



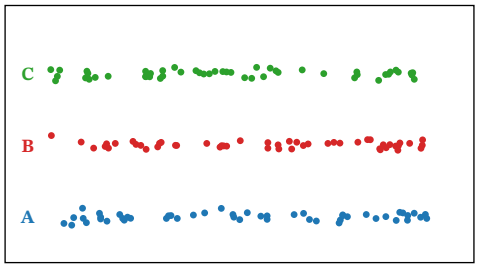
(b) Faithful



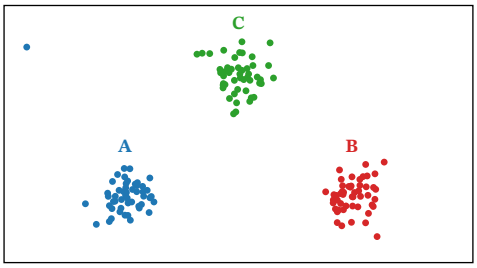
(c) Shuffled labels



(d) Nested classes



(e) Stretched classes



(f) Outlier in A

Figure 4.5: Extremes of the label-aware metrics. The high-dimensional source (a) is three labelled Gaussian clusters arranged as a triangle in $\mathbb{R}^3$. The faithful embedding (b) reproduces the triangle with compact, label-coherent clusters and scores high on every label-aware metric. Shuffling labels (c) keeps the spatial layout but each spatial cluster contains roughly a third of every label; no class corresponds to a spatial cluster, so every label-aware metric collapses. Nesting the classes as concentric rings (d) leaves them perfectly separable and keeps almost every point's neighbours inside its own class, so neighbourhood hit stays near one. All three classes nevertheless share a common centroid, so a point on an outer ring is no nearer its own class centroid than the centroid of the class it encloses, and distance consistency falls to roughly chance. The cluster density measure falls with it, being centroid-based too. Replacing each cluster with a long horizontal strip (e) again keeps every point's neighbours within its own class, producing a high neighbourhood hit. The wide spread within the cluster relative to the gap in between clusters produces low ClDM and Dunn scores. A single outlier in cluster A (f) barely moves the mean-based metrics, yet the Dunn index drops, as the intra-cluster diameter is now significantly larger, while the inter-cluster distance remains the same. This difference illustrates that Dunn alone is sensitive to outliers. The Davies-Bouldin ratio is not shown because it is a relative measure that depends on the high-dimensional layout as well as the embedding.

4.3 Generating visualisation corpora

The goal of this methodology is to compare visualisation corpora against each other. To properly compare visualisations, we need a representative visualisation corpus which actual analysts would produce. To that end, we optimise for numerous metrics, as described in Section 4.2, to get a high-quality corpus. In this section, we describe this process of optimising the hyperparameters on the metrics.

4.3.1 Hyperparameter Optimisation Procedure

Visualisation metrics can be competitive: a configuration that achieves high trustworthiness may sacrifice SNS, and vice versa. Therefore we want a large set of visualisations, where different metrics are prioritised simultaneously. The corpus is built by varying the hyperparameters of the t-SNE method, since each configuration gives a different visualisation with its own metric values. Rather than collapsing the metrics into a single scalar score, we treat the metric tuple as a multi-objective optimisation problem in which each metric is one objective, because depending on the task an analyst might prefer a visualisation which scores better on one metric than on another, so one scalar score can never capture that. The corpus is then the set of visualisations that no other visualisation strictly dominates on every metric, and its image in metric space is a Pareto front.

Six of the nine metrics of Section 4.2 serve as objectives, two per family: trustworthiness and continuity for local structure preservation, SNS and Shepard goodness for distance preservation, and neighbourhood hit and distance consistency for label-aware structure. The cluster density measure, the Davies-Bouldin ratio and the Dunn index are reported but not optimised, because their values are not confined to a bounded range and so cannot be placed on a common scale with the others. We write $J = 6$ for the number of objectives.

The multi-objective optimisation method we employ is qParEGO [31], which combines Bayesian optimisation with random scalarisation of the objectives. We call one assignment of all hyperparameters a *configuration*, one evaluation of a configuration a *trial*, and one complete qParEGO optimisation a *run*; a run therefore yields one corpus. Each run consists of two phases: an initialisation phase, which establishes the starting visualisations from which to optimise by employing Latin-hypercube initialisation on the search space, and a main phase, which employs the actual optimisation. The main phase consists of 1000 trials.

To stabilise the metric estimates, every configuration is re-evaluated on three independent random seeds and the mean of the three is reported as the metric value seen by the optimiser.

4.3.2 Training Schedule

Within each trial, and following standard t-SNE practice [1], we use a two-phase training schedule. During the early exaggeration phase, which comprises the first 250 of 800 gradient-descent iterations, the high-dimensional affinities p_{ij} are multiplied by an early-exaggeration factor and the momentum is set to 0.5. In the main phase, which covers the remaining iterations, the affinities return to their true values and the momentum increases to 0.8. The exaggeration factor is treated as a hyperparameter rather than fixed at the customary value of 4 used in the original t-SNE formulation, or 12 as in the revised hyperparameter formulation of [15]. This phase encourages the formation of separated local structures.

4.3.3 Hyperparameter Search Space

The full hyperparameter search space of the proposed method is summarised in Table 4.1. Three parameters are always optimised: the learning rate, the perplexity ratio, and the early-exaggeration factor. Three further parameters are optimised conditionally on the selected loss setting: the centering weight, the global-loss weight, and the depth-norm weight. The curvature magnitude $|K|$ is optimised on a logarithmic scale for non-Euclidean geometries and is fixed at zero for the Euclidean geometry. The early- and main-phase momentum values are held fixed at empirically chosen values to keep the search-space dimensionality manageable.

Table 4.1: Hyperparameter search space. The perplexity is parameterised as a fraction of the dataset size, perp $= f_{\text{perp}} \cdot N$, so that datasets of $N = 1000$ and $N = 5000$ points explore the same neighbourhood-fraction range. The most general case, where all loss functions are applied, has seven optimised dimensions; the remaining parameters are fixed at their specified values.

Hyperparameter	**Range**	**Scale**	**Status**
Learning rate η	$[0.5, 80]$	log	optimised
Perplexity ratio f_{perp}	$[4 \cdot 10^{-4}, 8 \cdot 10^{-2}]$	log	optimised
Early-exaggeration factor	$[1, 44]$	linear	optimised
Centering weight λ_{cen}	$[0, 2]$	linear	conditional
Global-loss weight λ_{glb}	$[0, 1]$	linear	conditional
Depth-norm weight λ_{norm}	$[0, 0.02]$	linear	conditional
Curvature magnitude $\lvert K \rvert$	$[10^{-6}, 5]$	log	non-Euclidean only
Main-phase momentum	0.8	—	fixed
Early-phase momentum	0.5	—	fixed
Iterations	800	—	fixed
Early-exaggeration iterations	250	—	fixed

4.3.4 Front Quality Indicator

Comparing loss settings or embedding geometries requires reducing each approximated Pareto front to a scalar, since two fronts in J objectives are rarely comparable by dominance alone. We summarise a front with the R2 indicator [42,43], which measures how well a set of solutions serves a family of scalarised preferences.

All objectives are first oriented so that higher is better and rescaled to $[0, 1]$. The rank-based metrics, the neighbourhood hit and distance consistency are already in that range and are used directly, SNS enters as $1 - \text{SNS}$ because zero is best, and a diverged trial whose metric is undefined is assigned 0 so that failures score badly rather than being dropped. The ideal point in this space is $z^* = (1, \ldots, 1)$. Let $A \subset [0, 1]^J$ be the oriented Pareto front of a run, and let W be a finite set of weight vectors drawn from the discrete simplex of [31], that is vectors w with $\sum_j w_j = 1$ and $w_j = l/g$ for integer l, at $g = 5$. The indicator is the mean, over that weight set, of the best weighted Chebyshev utility any point of the front attains:

$$R_2(A; W) = \frac{1}{|W|} \sum_{w \in W} \min_{a \in A} \max_{j=1}^{J} w_j \left(z_j^* - a_j \right). \tag{4.20}$$

Smaller values are better, and $R_2 = 0$ only for a front containing the ideal point. We therefore report the gain of a loss setting over the `all_off` baseline as the reduction in the indicator,

$$\Delta R_2(W) = R_2\left(A_{\text{all\_off}}; W\right) - R_2\left(A_{\text{setting}}; W\right), \tag{4.21}$$

so that positive values mean improvement, matching the reading direction of the metrics themselves.

The indicator is weakly Pareto compliant [43], so a front that dominates another can never receive a worse value, which is what makes $\Delta R_2 > 0$ mean what it appears to mean.

As a secondary summary we report the binary additive ε-indicator [44]. The ε-indicator is defined as

$$I_\varepsilon(A, B) = \max_{b \in B} \min_{a \in A} \max_{j=1}^{J} \left(b_j - a_j \right), \tag{4.22}$$

the smallest amount by which every objective of A must be shifted before A dominates B, so that $I_\varepsilon(A, B) \leq 0$ exactly when A covers B. The measure is asymmetric and neither direction alone settles the comparison when the two fronts cross, so both are computed and reported. Because the indicator says nothing about how many points a front carries, the size of each front is reported next to it. The ε-indicator is exact and parameter-

free, and, unlike R_2, it is a worst-case measure rather than an average, it remains sensitive to regressions in a single region of objective space.

4.3.5 Preference Regions and Recommended Configurations

An analyst arrives at a visualisation with a priority: wanting neighbourhoods they can trust, or wanting the large-scale distances in the visualisation to mean something. The question they ask is which configuration serves that priority, and whether the embedding geometry changes the answer.

The weight set W of Eq. (4.20) is where a priority is expressed: concentrating the weight distribution on a region of the simplex focuses the indicator on the corresponding region of the front [45]. We therefore evaluate Eq. (4.20) under several weight sets, each answering a different question of the same front:

- W_{all}, the full discrete simplex, containing all vectors. This asks whether a loss setting improves the front across all trade-offs equally weighted, and is the only weight set under which ΔR_2 is a claim about overall front quality.
- W_m for each metric m, containing the vectors that place at least half their mass on that metric. This asks whether a loss setting improves the front for an analyst who cares mainly about m.
- W_{struct}, W_{dist} and W_{class} for the three metric families, containing the vectors that place at least half their mass on the two local-structure, the two distance-preservation, respectively the two label-aware metrics together.

We report $\Delta R_2(W)$ per weight set, to find an improvement in the Pareto front.

5 Results

This chapter examines how embedding-space curvature affects the quality of two-dimensional visualisations. Following the methodology developed in Chapter 4, we compare visualisation corpora generated for different datasets, embedding geometries and loss settings. Every metric in this chapter is computed on the visualisation, and every number is a property of a corpus, that is, of a Pareto front over the six projected objectives of Subsection 4.3.1; no trial the search discarded enters any figure or table. The four sections after the datasets follow the four research questions of this thesis:

1. **RQ1: Does matching the curvature of the dataset and embedding space improve visualisation quality?**
2. **RQ2: What effect does curvature have on established metrics in visualisation literature?**
3. **RQ3: Can curvature be tuned as its own hyperparameter?**
4. **RQ4: Can auxiliary global loss terms improve the visualisations produced by non-Euclidean t-SNE?**

We first describe the datasets, then examine geometry matching, individual metrics, qualitative effects of curvature, and auxiliary losses.

5.1 Datasets

We use two categories of datasets: synthetic datasets with known intrinsic geometry, and real-world datasets with identifiable geometric structure.

5.1.1 Synthetic Datasets

Synthetic datasets allow controlled evaluation because their intrinsic geometry is known by construction. Four synthetic generators are used, one per geometric class identified in Chapter 4: Each generator produces points with $d = 10$ coordinates, so the grid fills a ten-dimensional cube and the sphere, the hyperbolic shells and the tree lie on nine-dimensional manifolds. Two-dimensional variants of the same generators, with $d = 2$ for the grid and the sphere and a two-dimensional Poincaré disk for the shells and the tree, are used only for the visual comparisons of Figure 5.4 and Figure 5.12. We write $x_i \in \mathbb{R}^d$ for the ith generated point, ℓ_i for its integer label, and D_{ij} for an entry of the provided ground-truth distance matrix; $\mathbb{1}[\cdot]$ denotes the indicator function.

- **Grid** (Euclidean): points sampled uniformly from the hypercube $[-1, 1]^d$, labelled by quadrant of the first two coordinates, serving as the flat dataset.

$$x_{ik} \sim \mathrm{Unif}(-1, 1), \quad \ell_i = 2\mathbb{1}[x_{i1} \geq 0] + \mathbb{1}[x_{i2} \geq 0], \quad D_{ij} = \|x_i - x_j\|_2, \tag{5.1}$$

- **Sphere** (spherical): points sampled uniformly on the unit sphere $\mathbb{S}^{d-1}$ by normalising standard Gaussian vectors [46], labelled by hemisphere, with pairwise distances given by the great-circle metric:

$$z_i \sim \mathcal{N}(0, I_d), \quad x_i = \frac{z_i}{\|z_i\|_2}, \quad \ell_i = \mathbb{1}[x_{i1} \geq 0], \quad D_{ij} = \arccos\big(\langle x_i, x_j \rangle\big), \tag{5.2}$$

- **Hyperbolic shells** (hyperbolic): points concentrated on three concentric circles at fixed hyperbolic radii, labelled by shell, testing distance preservation in the hyperbolic geometry. Points are laid out in the Poincaré disk model with curvature radius $r = 1$, stored on the hyperboloid through the lift Φ of Eq. (3.13), and equipped with the geodesic distances of Eq. (3.8). The shells divide the points into three equal groups and place group $s \in \{0, 1, 2\}$ on the circle of hyperbolic radius ρ_s, perturbed by radial noise $\eta_i \sim \mathcal{N}(0, 1)$ of standard deviation $\sigma = 0.05$, with direction w_i uniform on $\mathbb{S}^{d-2}$:

$$p_i = \left(\tanh\left(\frac{\rho_{s_i}}{2}\right) + \sigma\eta_i\right) w_i, \quad (\rho_0, \rho_1, \rho_2) = (0.5, 1.5, 2.5), \tag{5.3}$$

and the shell index s_i serves as the label.

- **Tree** (hierarchical): a binary tree laid out in the hyperbolic plane with constant-length edges, labelled by depth, testing the hierarchical case which is conjectured to favour hyperbolic geometry. Every node at depth δ sits on the circle of hyperbolic radius $c\delta$ around the root, where $c = 0.8$ is the constant hyperbolic length of an edge, and the bth node of its level, $b \in \{0, \ldots, 2^\delta - 1\}$, is placed at

$$p_{\delta,b} = \tanh\left(c\frac{\delta}{2}\right)\left(\cos\theta_{\delta,b}, \sin\theta_{\delta,b}\right)^T, \quad \theta_{\delta,b} = 2\pi\frac{b}{2^\delta} + 0.3\delta, \tag{5.4}$$

  Levels are filled breadth-first until N nodes have been placed, a node's label is its depth capped at four, and the disk coordinates are stored through the lift of Eq. (3.13). In the high-dimensional variant, the two-dimensional layout is embedded into a higher-dimensional Poincaré ball whose remaining coordinates receive small Gaussian noise.

5.1.2 Real-World Datasets

Real-world datasets test whether the gains observed on the synthetic datasets transfer to practical scenarios. Four real datasets are used, chosen to span the geometric classes that the synthetic datasets isolate:

- **MNIST** [47]: 70,000 images of handwritten digits (784 dimensions). The digit classes form relatively compact clusters without strong hierarchical or cyclic structure, making this the Euclidean-like dataset.
- **Fashion-MNIST** [48]: 70,000 grayscale images of clothing articles (784 dimensions), structurally analogous to MNIST but with greater within-class variability and stronger inter-class confusions.
- **PBMC** [49]: single-cell RNA-sequencing of 2638 peripheral blood mononuclear cells, reduced to their first 50 principal components. Immune cells differentiate along a tree of lineages, which makes PBMC a candidate hierarchical dataset.
- **WordNet mammals** [50]: the mammals subtree of the WordNet hierarchy, with 1170 nodes, presented as graph distances. This is a paradigm hierarchical dataset and the canonical benchmark for hyperbolic embeddings [3].

Every corpus in this chapter is computed with a budget of $N = 5000$ points per dataset. Datasets with fewer points are used in full, so the PBMC and WordNet corpora hold 2638 and 1170 points. For the feature-vector datasets, pairwise distances are computed by t-SNE itself as Euclidean distances between the input vectors, and for the synthetic datasets they are the ground-truth distances of the generators. WordNet has no feature vectors, so its input is the matrix of shortest-path distances between the nodes of the tree. Table 5.1 summarises the datasets.

Table 5.1: The datasets of this chapter. N is the number of points in every corpus, and the dimension is the number of input coordinates. The geometry of the synthetic datasets is known by construction, while for the real datasets it is the expected class, which cannot be confirmed from their distances (Chapter 8). The two-dimensional variants of the synthetic datasets, with $N = 1000$, are used only for the visual comparisons.

Dataset	Geometry	N	Dimension	Input
Grid	Euclidean	5000	10	coordinates, Euclidean distances
Sphere	spherical	5000	10	coordinates, great-circle distances
Hyperbolic shells	hyperbolic	5000	10	coordinates, geodesic distances
Tree	hyperbolic	5000	10	coordinates, geodesic distances
MNIST	Euclidean-like	5000	784	pixel intensities
Fashion-MNIST	Euclidean-like	5000	784	pixel intensities
PBMC	hierarchical	2638	50	principal components
WordNet mammals	hierarchical	1170	n/a	shortest-path distances

5.2 Does Matching Dataset and Embedding Curvature Improve Quality?

This section answers RQ1 by comparing the three embedding geometries on synthetic datasets with known generating geometry. The two curved corpora are compared against the Euclidean corpus and against each other, to show whether a matching sign gains over both. Comparisons use the `all_off` corpora and the optimisation budget as defined in Subsection 4.3.1.

5.2.1 Overall Quality of the Synthetic Corpora

Figure 5.1 reports the available synthetic results for the ten-coordinate datasets. We use the R_2 value for scoring the corpora. R_2 is a cost, so smaller values indicate a higher corpus quality under the chosen preferences. Here the gain is measured against the matching corpus, so R_2 gain equals $R_2(\text{mismatched}) - R_2(\text{matched})$; a positive value means the matched corpus gains.

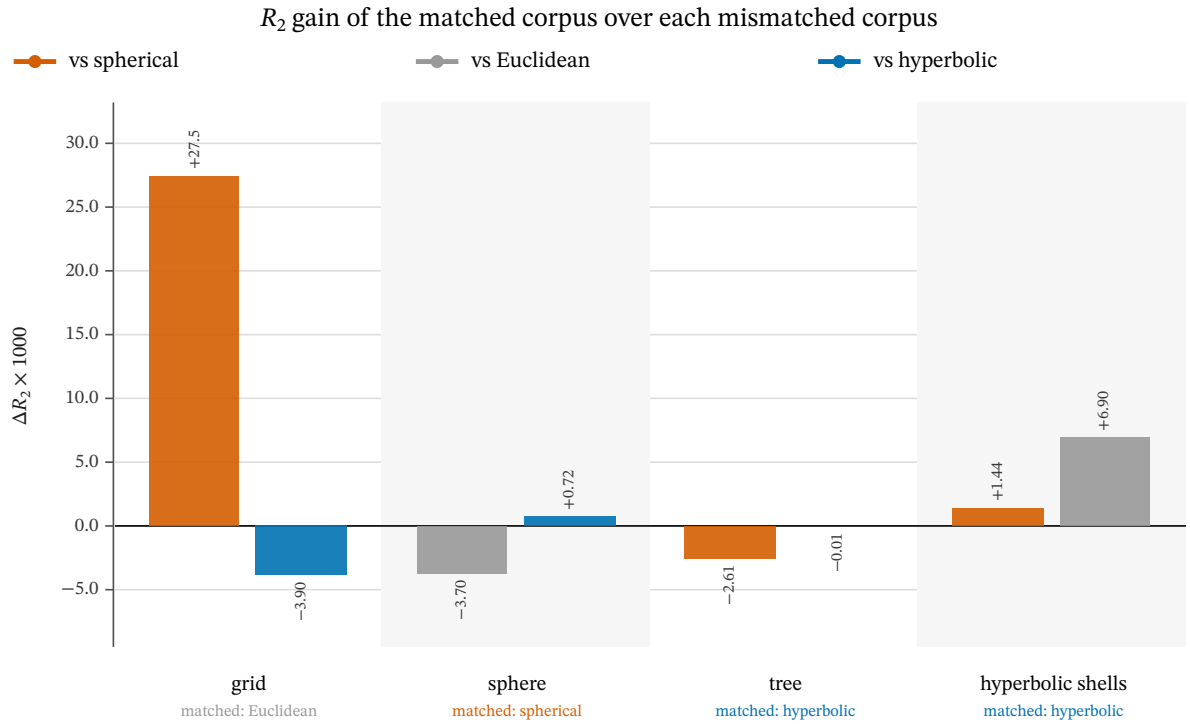


Figure 5.1: Matched embedding geometry on the synthetic datasets of known intrinsic geometry. Each dataset is embedded under all three geometries, and each corpus is the Pareto front over the six projected objectives of Subsection 4.3.1, scored under W_{all}. R_2 gain equals $R_2(\text{mismatched}) - R_2(\text{matched})$, which is the improvement on the metric, as defined in Eq. (4.20). A positive value means the visualisation of the matched geometry improved over the mismatched visualisation. The R_2 gain is multiplied by 1000 for legibility, and an R_2 gain of $|\Delta R_2| < 0.001$, below 1 on this scale, is read as a tie.

The hyperbolic shells are the one dataset where the matched corpus gains over both others, and it gains mostly over the Euclidean corpus and only a little over the spherical one, so the gain is not one of the matching sign. It is not a gain of curvature either: about three quarters of the front points of both curved corpora of the hyperbolic shells lie below $\kappa \approx 10^{-3}$, where the embedding is effectively Euclidean (Table 5.2), so the gain is specific to this dataset. On the tree the matched hyperbolic corpus ties with the Euclidean corpus and loses to the spherical corpus. On the sphere dataset the matched spherical corpus loses to the Euclidean corpus and ties with the hyperbolic one. On the grid, a flat dataset, the matched Euclidean corpus gains strongly over the spherical corpus and loses to the hyperbolic one. Matching the generating geometry therefore does not stand out in the overall indicator. $R_2(W_{\text{all}})$ describes the corpus as a whole and does not take into account the strengths and weaknesses for particular metrics; Subsection 5.3.2 takes those apart.

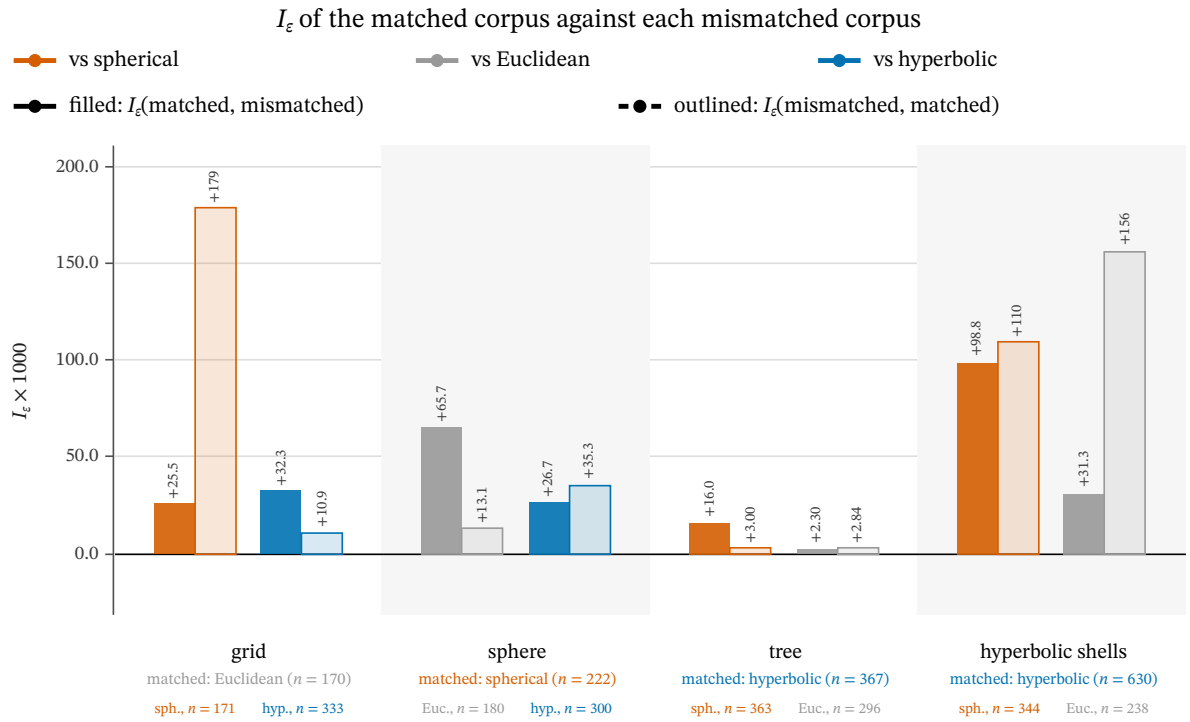


Figure 5.2: The ε-indicator, as defined in Eq. (4.22), both ways, for the corpora of Figure 5.1. The filled bar shows how much each metric of the matched arm must improve to dominate the Pareto front of the mismatched arm; the outlined bar shows how much the metrics of the mismatched arm must improve to dominate the matched arm. The ε-indicator is multiplied by 1000 for legibility.

The ε-indicator (Figure 5.2), which measures how far one corpus falls short of the other at its worst point instead of on average, agrees in direction. On the tree the ε values are small in both directions against both other corpora, so the corpora nearly cover each other. Elsewhere the largest ε values are one-sided and belong to the corpora that lose under R_2: the Euclidean corpus of the hyperbolic shells, the spherical corpus of the grid and the spherical corpus of the sphere dataset. The two curved corpora of the hyperbolic shells cross, with ε values of similar size in both directions.

5.2.2 Where the Front Points Lie

The dimensionless curvature κ of Subsection 3.2.6 measures how much of the curvature an embedding expresses. Figure 5.3 shows where the search places the front points of the curved corpora along the range of κ, pooled over all datasets, and Table 5.2 breaks it down per corpus.

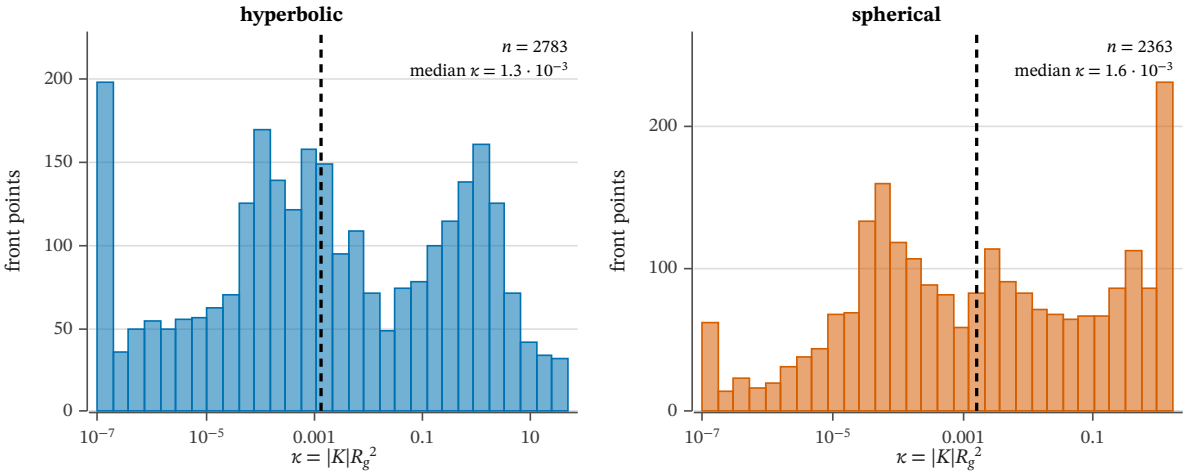


Figure 5.3: The distribution of κ, as defined in Eq. (3.16), over the front points of the `all_off` corpora of all datasets, synthetic and real. Hyperbolic on the left, spherical on the right; the horizontal axis is logarithmic. The dashed line marks the median κ of each geometry, and the bin at $\kappa \approx 10^{-7}$ holds the collapsed embeddings.

In both geometries about half of the front points lie below $\kappa \approx 10^{-3}$, where the embedding covers only a few hundredths of the curvature radius and is effectively Euclidean. Collapsed embeddings, in which the points have contracted onto nearly a single location, form a separate peak at $\kappa \approx 10^{-7}$, larger in the hyperbolic than in the spherical geometry. The rest of the front spreads towards high κ, with a quarter of the points or more above $\kappa \approx 0.1$: in the hyperbolic geometry they form a second peak between $\kappa \approx 0.1$ and 1, and in the spherical geometry they pile up at the bound, where the embedding covers the whole sphere. The share of front points at small κ differs between datasets. The curved corpora of the hyperbolic shells lie at the smallest κ of all datasets: in both geometries about three quarters of their front points lie below $\kappa \approx 10^{-3}$, and nearly all below $\kappa \approx 0.1$. The spherical corpus of the sphere dataset comes next, with two thirds of its front points below $\kappa \approx 10^{-3}$.

Table 5.2: The distribution of κ over the front points of each curved `all_off` corpus: the median κ, and the share of front points below $\kappa \approx 10^{-3}$, where the embedding is effectively Euclidean, and below $\kappa \approx 0.1$. The shares below 10^{-3} include the collapsed embeddings at $\kappa \approx 10^{-7}$.

Dataset	**Hyperbolic**			**Spherical**		
	median κ	$\kappa < 10^{-3}$ (%)	$\kappa < 0.1$ (%)	median κ	$\kappa < 10^{-3}$ (%)	$\kappa < 0.1$ (%)
Grid	0.52	23	33	$4.9 \cdot 10^{-3}$	34	77
Sphere	0.80	35	44	$1.4 \cdot 10^{-4}$	66	94
Tree	$2.9 \cdot 10^{-3}$	43	68	$3.0 \cdot 10^{-3}$	44	73
Hyperbolic shells	$1.3 \cdot 10^{-4}$	75	99	$5.2 \cdot 10^{-5}$	71	95
MNIST	$3.2 \cdot 10^{-3}$	37	87	$4.3 \cdot 10^{-2}$	29	56
Fashion-MNIST	$8.8 \cdot 10^{-3}$	43	59	$5.4 \cdot 10^{-4}$	54	88
PBMC	$1.1 \cdot 10^{-3}$	50	83	$6.7 \cdot 10^{-2}$	34	53
WordNet mammals	$1.1 \cdot 10^{-3}$	50	78	$5.0 \cdot 10^{-3}$	35	69

5.2.3 Visual Evidence for Geometry Matching

Figure 5.4 places the available synthetic visualisations side by side. The figure provides examples of the structures retained or distorted by each geometry; the corpus-level comparison is given by Figure 5.1. To illustrate the effect of the geometry better, we have chosen to show the visual evidence of the two-dimensional variants,

instead of the ten-coordinate datasets. The hyperparameters of a sample of the relevant visualisation corpora were taken to create the visualisations. The visualisations are made with 1000 points each, under the `all_off` setting.

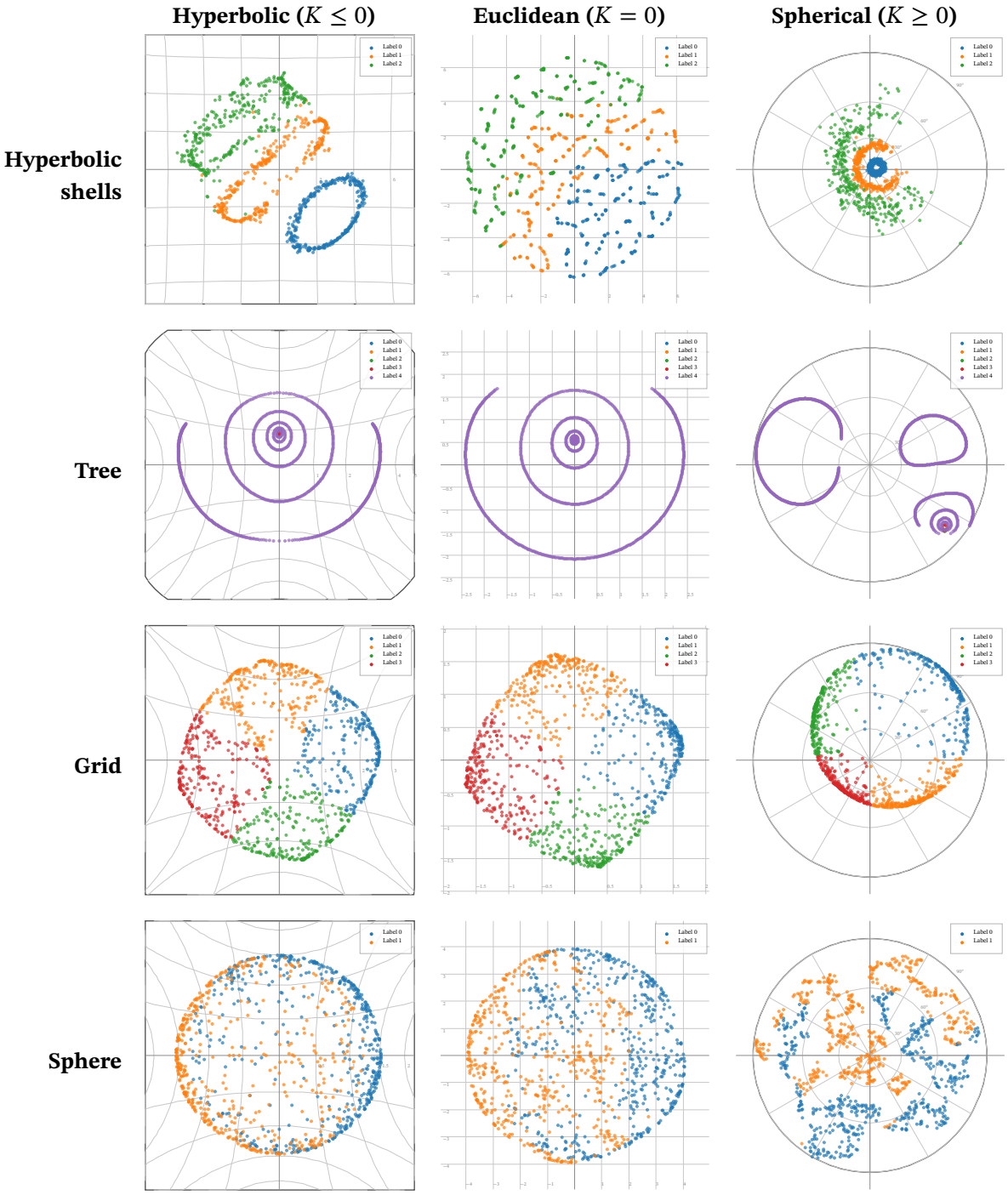


Figure 5.4: Curvature-geometry grid: t-SNE embeddings of four synthetic datasets (rows) into spaces of different constant curvature (columns). Matching geometries are hyperbolic for the shells and tree, Euclidean for the grid, and spherical for the sphere.

We see that in the hyperbolic shells dataset, the shells are clearly separated in the hyperbolic visualisation, although they appear tilted. We do recognise the ring structure, it has remained circular. In the tree dataset, we see that the levels of the tree remain correctly clustered, and that the hierarchy is preserved in the hyperbolic and the Euclidean geometry. We do see that the hierarchy is lost in the spherical geometry, where the levels have disconnected from the main structure. On the grid the visualisation is largely faithful in all geometries,

although we do see the points pushing to a ring in the spherical geometry. For the sphere dataset, we see overlapping points in the hyperbolic and the Euclidean geometry, while the spherical visualisation shows a clear distinction in the clusters.

Together, these results answer RQ1 differently for the two views of the synthetic datasets. On the ten-coordinate datasets, matching the curvature of the dataset and embedding space does not improve visualisation quality: the matched corpus gains over both others only on the hyperbolic shells, where most of both curved corpora is effectively Euclidean, so that gain is dataset-specific. On the two-dimensional variants the sampled visualisations suggest that the matched geometry is the most faithful, although on the tree the Euclidean visualisation keeps the hierarchy as well, and the grid looks similar in all geometries.

5.3 What Effect Does Curvature Have on Established Metrics?

RQ2 concerns which properties of a visualisation change with embedding geometry. We examine local structure, distance preservation, and label-aware structure separately, since an improvement in overall front quality can conceal a loss in one of these properties.

5.3.1 Metric-Level Comparisons Across Geometries

The `all_off` corpora of all datasets have been pooled per geometry, and we have plotted the κ of every front point against the quality metrics defined in Section 4.2. The scale-normalised stress (SNS) is reported as $1 -$ SNS, such that a higher value is better for all metrics. This plot reads the metrics directly from the front points, so no indicator and hence no preference region is involved; each curve is the median in a bin of κ, drawn only where the bin holds at least twenty front points. The result is shown in Figure 5.5.

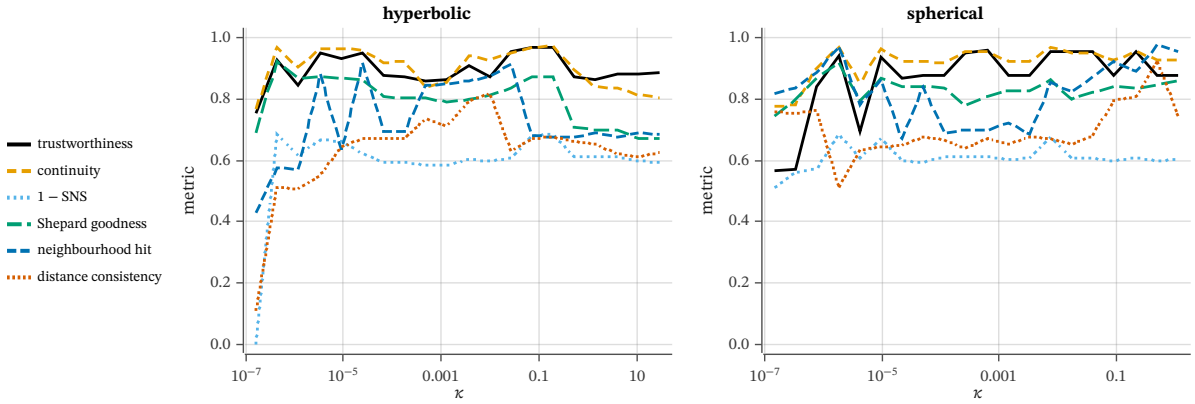


Figure 5.5: The six optimised metrics as a function of κ, as defined in Eq. (3.16). The metrics that map from 0 to 1 were chosen, as they can be directly compared to each other. Here, scale-normalised stress (SNS) is reported as $1 -$ SNS, so for all metrics, higher is better. We show the median value of the metric over the front points of the `all_off` corpora of all datasets, synthetic and real, in bins of κ holding at least twenty points. The gaps in the lines mean that there were fewer than 20 points in that bin. κ is plotted logarithmically in both geometries.

Both geometries cover a wide range of κ, from about 10^{-7} up to the upper edge of each geometry. On the sphere that edge lies near $\kappa \approx 1.5$, the value of an embedding that covers the whole sphere (Subsection 3.2.6); the hyperbolic geometry has no such bound, and its front points reach larger κ values.

In the hyperbolic geometry the lowest bin, at $\kappa \approx 10^{-7}$, holds the collapsed embeddings of Figure 5.3: every metric is low there, SNS drops to 0, and distance consistency almost to 0. Above it the local-structure metrics stay high and flat up to $\kappa \approx 0.3$, while the neighbourhood hit and distance consistency vary between bins without a trend. From $\kappa \approx 0.3$ onwards Shepard goodness, the neighbourhood hit and continuity are

lower than below it, so strongly curved hyperbolic visualisations lose distance order and label separation at once.

In the spherical geometry no bin collapses, but the lowest bins have a low trustworthiness. Across the middle of the range every metric is roughly flat. Towards the upper edge, where the embedding covers most of the sphere, the neighbourhood hit and distance consistency rise to their highest values, while the other metrics stay flat. The spherical geometry therefore does not lose at high κ the way the hyperbolic geometry does.

In Figure 5.6, we plot the curvature magnitude $|K|$ of the embedding space against the metrics. κ is a result of t-SNE expanding and contracting, eventually leading to a certain geometry; $|K|$ is the input hyperparameter of the curvature of the embedding space. In the hyperbolic geometry the local-structure metrics and Shepard goodness are highest for $|K|$ between about 10^{-4} and 10^{-2} and decline above $|K| \approx 0.1$, while the label-aware metrics vary without a trend. In the spherical geometry trustworthiness, continuity, Shepard goodness and SNS are flat over the whole searched range, and the neighbourhood hit rises with $|K|$ up to $|K| \approx 0.01$ and stays high above it.

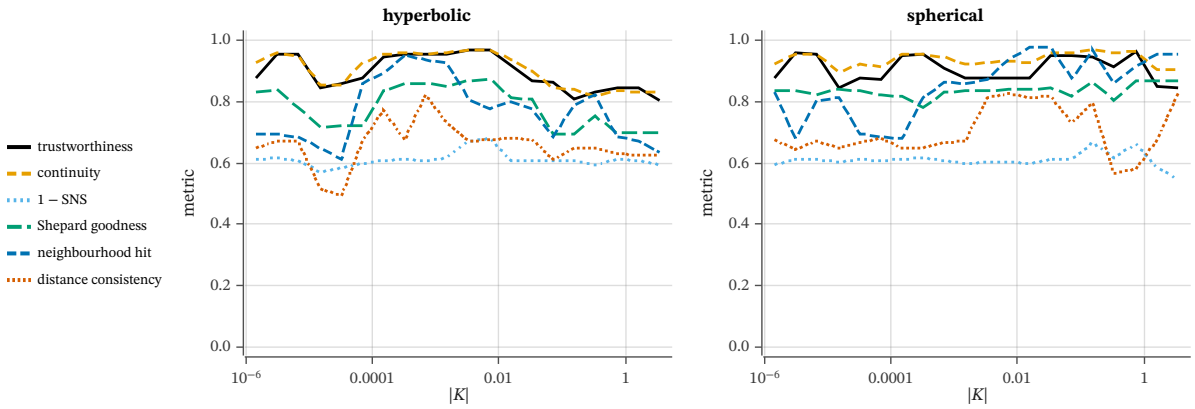


Figure 5.6: The six optimised metrics as a function of $|K|$, the curvature magnitude of the embedding space. The metrics that map from 0 to 1 were chosen, as they can be directly compared to each other. Here, scale-normalised stress (SNS) is reported as $1 - \text{SNS}$, so for all metrics, higher is better. We show the median value of the metric over the front points of the `all_off` corpora of all datasets, synthetic and real, in bins of $|K|$ holding at least twenty points.

The Dunn index, the Davies-Bouldin ratio, and the cluster density measure (ClDM) are unbounded above, so they cannot share an axis with the six metrics above and are reported but not optimised (Subsection 4.3.1). Figure 5.7 shows ClDM as a representative example, against κ in both geometries.

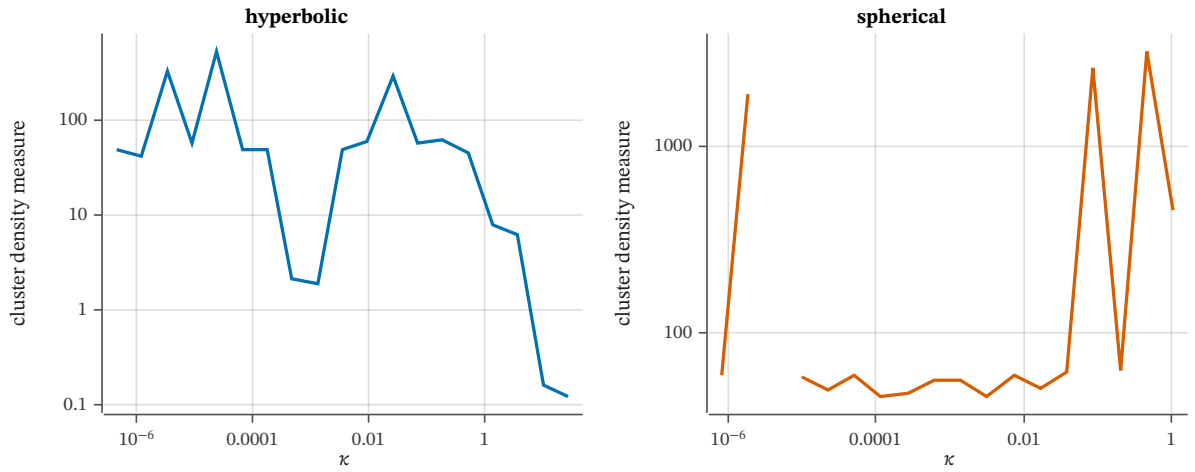


Figure 5.7: The cluster density measure (ClDM), one of the three metrics of Section 4.2 that are unbounded above and so are excluded from Figure 5.5 and Figure 5.6, shown as a representative example. ClDM is plotted against κ, as defined in Eq. (3.16), in both geometries. As in Figure 5.5, each curve is the median over the front points of the `all_off` corpora of all datasets, synthetic and real, in bins holding at least twenty points, and a gap means the bin held fewer. The axis is logarithmic, since the medians span more than a decade. Appendix A reports all three unbounded metrics against both κ and $|K|$, in both geometries.

In the hyperbolic geometry ClDM varies over two decades between neighbouring bins without a trend, with a sharp dip near $\kappa \approx 10^{-3}$, and falls by more than two decades above $\kappa \approx 1$, the same range in which Figure 5.5 shows the bounded metrics lose. On the sphere ClDM is flat over most of the range and departs from that level only in isolated bins at both ends, where it spikes by more than a decade. The Davies-Bouldin ratio and the Dunn index follow the bounded label-aware metrics more closely than ClDM does: in the hyperbolic geometry both drop to 0 in the collapsed bin, peak near $\kappa \approx 0.1$ and fall above it, and in the spherical geometry both rise towards the upper edge. The unbounded metrics therefore confirm the two patterns of Figure 5.5, the hyperbolic loss at high κ and the spherical rise of label separation near full coverage of the sphere. Appendix A reports the full set of curves, against both κ and $|K|$, in both geometries.

5.3.2 Which Metrics Carry the Curvature Gain?

Figure 5.1 compared the corpora of a dataset under W_{all}, but an overall indicator cannot say which metrics a gain or a loss comes from, nor whether a tie overall hides a trade-off between metrics. The preference regions of Subsection 4.3.5 answer both questions. For each metric m, W_m holds the vectors that place at least half their mass on m, and W_{struct}, W_{dist} and W_{class} hold those that place at least half their mass on the two local-structure, the two distance-preservation, respectively the two label-aware metrics together. Scoring a corpus under W_m is scoring it as an analyst who cares mainly about m would. Figure 5.8 reports the R_2 gain of each curved corpus over the Euclidean corpus of the same dataset under every region, for the synthetic and the real datasets alike.

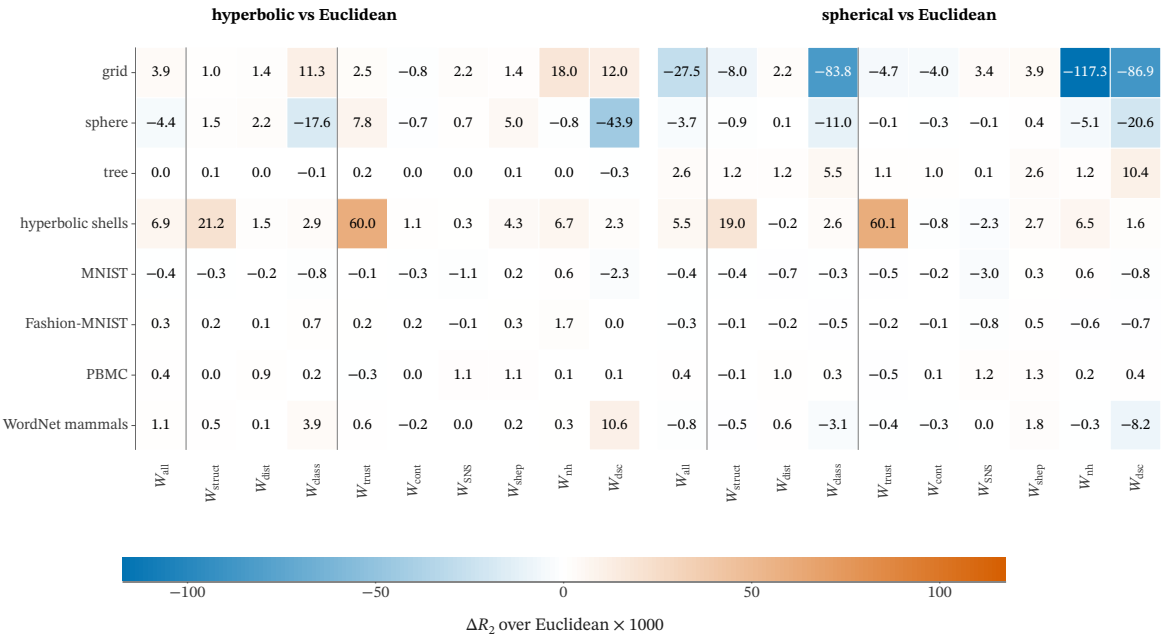


Figure 5.8: R_2 gain of each curved corpus over the Euclidean corpus of the same dataset under every preference region of Subsection 4.3.5, without additional loss terms. Each cell is $1000 \cdot (R_2(\text{euclidean}) - R_2(\text{curved}))$ for the region of its column, so a positive value means the curved visualisations served that preference better. The gain is multiplied by 1000 for legibility, and a cell with $|\Delta R_2| < 0.001$, below 1 on this scale, is read as a tie. W_{all} is the full weight simplex; W_{struct}, W_{dist} and W_{class} hold the vectors that place at least half their mass on the two local-structure, the two distance-preservation, respectively the two label-aware metrics; for each metric m, W_m holds the vectors that place at least half their mass on m. Both panels share one colour scale.

Under the distance-preservation regions the curved corpora differ little from the Euclidean corpus. Every cell under W_{dist} is a tie or a small gain of the curved corpus: the hyperbolic corpus gains on the grid, the sphere dataset and the hyperbolic shells, and the spherical corpus on the grid, the tree and PBMC. These gains do not follow matching, since the hyperbolic corpus of the tree and the spherical corpus of the sphere dataset tie. On the real datasets most cells are ties or small differences; the larger ones are on WordNet under the label-aware regions, where the hyperbolic corpus gains and the spherical corpus loses. The large differences in the figure sit under the label-aware regions, on the grid, the sphere dataset and the tree, and under the local-structure regions, on the hyperbolic shells.

The label-aware differences are the largest in the figure. The spherical corpus loses strongly under W_{class} on the grid and less on the sphere dataset, and the hyperbolic corpus gains under W_{class} on the grid and loses under W_{dsc} on the sphere dataset.

On the hyperbolic shells both curved corpora gain over the Euclidean corpus under W_{trust} and W_{struct}, by almost the same amount. As in Figure 5.1, the gain is specific to this dataset rather than an effect of curvature, since most front points of both curved corpora are effectively Euclidean (Table 5.2), and it lies in trustworthiness rather than in the distance-preservation metrics. The matched pairs elsewhere do not gain: the tree in the hyperbolic geometry ties under every region, the sphere dataset in the spherical geometry loses under the label-aware regions, and WordNet in the hyperbolic geometry gains mainly under the label-aware regions, through W_{dsc}. Matching the generating geometry does not stand out under any metric.

Read together with Figure 5.5, this places the effect of curvature on the established metrics: it is small under the distance-preservation metrics, dataset-specific rather than tied to matching, and largest under the label-aware metrics on the grid, the sphere dataset and the tree.

5.3.3 Metric Dependence

We use Spearman rank correlations to examine whether metrics order the same visualisations similarly. This helps interpret whether apparent improvements across several metrics reflect related properties, and whether the trade-offs a front is made of are the same in every geometry. We compute ρ between each pair of metrics over the front points of each `all_off` corpus, one dataset at a time, and Figure 5.9 shows the median over datasets per geometry, with the individual datasets behind it.

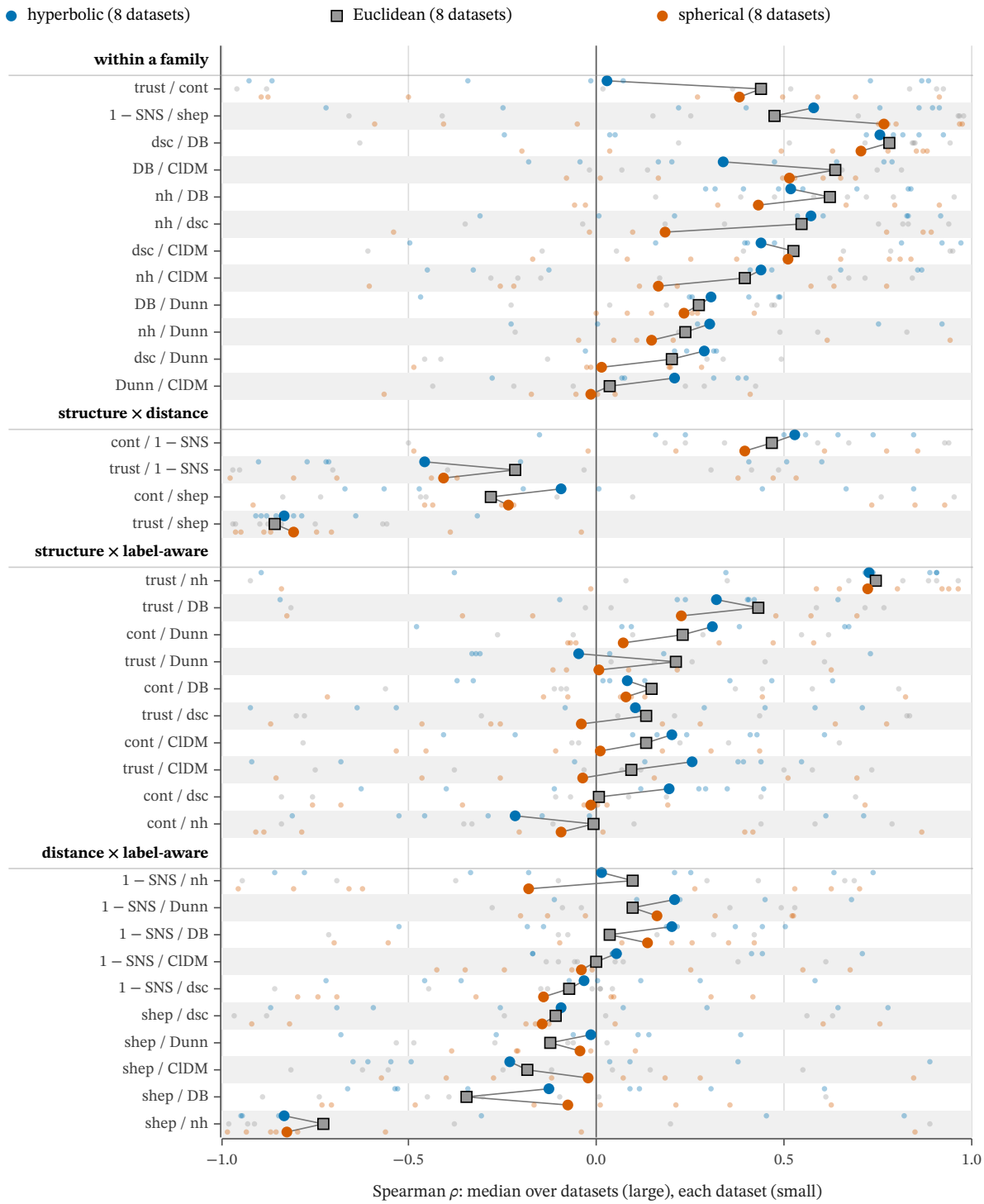


Figure 5.9: Spearman rank correlation between the projected metrics, one row per metric pair, one marker per embedding geometry. ρ is computed over the front points of each `all_off` corpus separately; the large marker is the median over datasets and the small dots behind it are the datasets themselves. The geometries sit in order of curvature, hyperbolic above the Euclidean reference and spherical below, joined through their medians. Rows are grouped into pairs within one metric family and pairs across two families, and sorted by the Euclidean ρ within each group; scale-normalised stress (SNS) enters as $1 - \text{SNS}$, so a positive ρ always means better-with-better. Metric abbreviations: trust (trustworthiness), cont (continuity), shep (Shepard goodness), nh (neighbourhood hit), dsc (distance consistency), ClDM (cluster density measure), DB (Davies-Bouldin ratio) and Dunn (Dunn index).

Curvature does not reorder the metrics: the medians of the three geometries agree on almost every row. The two distance-preservation metrics agree in all three geometries, and most in the spherical geometry, so W_{dist} treats a single property. Two within-family pairs do depend on the geometry. Trustworthiness and continuity agree moderately in the Euclidean and spherical geometries but are independent in the hyperbolic geometry. The neighbourhood hit and distance consistency agree in the Euclidean and hyperbolic geometries but only weakly in the spherical geometry.

Across families, the pattern is one metric against the rest. Shepard goodness opposes trustworthiness and the neighbourhood hit, the largest and most geometry-consistent correlations in the figure, while trustworthiness and the neighbourhood hit agree about as strongly. Along a front, higher Shepard goodness therefore tends to come with lower trustworthiness and a lower neighbourhood hit, so a front trades distance order against the preservation of neighbourhoods and classes. SNS takes a smaller part in this trade-off: it opposes trustworthiness moderately, agrees with continuity, and is independent of the label-aware metrics. Distance consistency is independent of every local-structure and distance-preservation metric, and within its own family it agrees most strongly with the Davies-Bouldin ratio. The two label-aware objectives therefore measure different things: the neighbourhood hit follows trustworthiness, and distance consistency follows the cluster-separation metrics.

The per-dataset markers qualify all of this. For almost every pair the datasets span both signs, sometimes from strong opposition to near-perfect agreement on one row, so the medians summarise dataset-specific relationships rather than properties of the metrics. This holds for trustworthiness and continuity as well: in every geometry, including the Euclidean and spherical ones whose medians agree, some datasets oppose the two strongly and others make them agree strongly, so their independent hyperbolic median is not a property of the hyperbolic geometry. Even the strongest median relations, those around the neighbourhood hit, have datasets of the opposite sign. A gain that several metrics register at once on one dataset is therefore not evidence that they measure related properties in general.

5.3.4 Metrics in the Embedding and in the Visualisation

All metrics so far are computed in the visualisation, after projection. The Poincaré and azimuthal-equidistant projections distort distances (Subsection 3.2.5), so a metric can read differently in the visualisation than in the embedding it projects. We call the difference the projection gap, and compute it for every front point by evaluating each metric a second time on the geodesic distances in the embedding. Figure 5.10 shows the projection gap against κ.

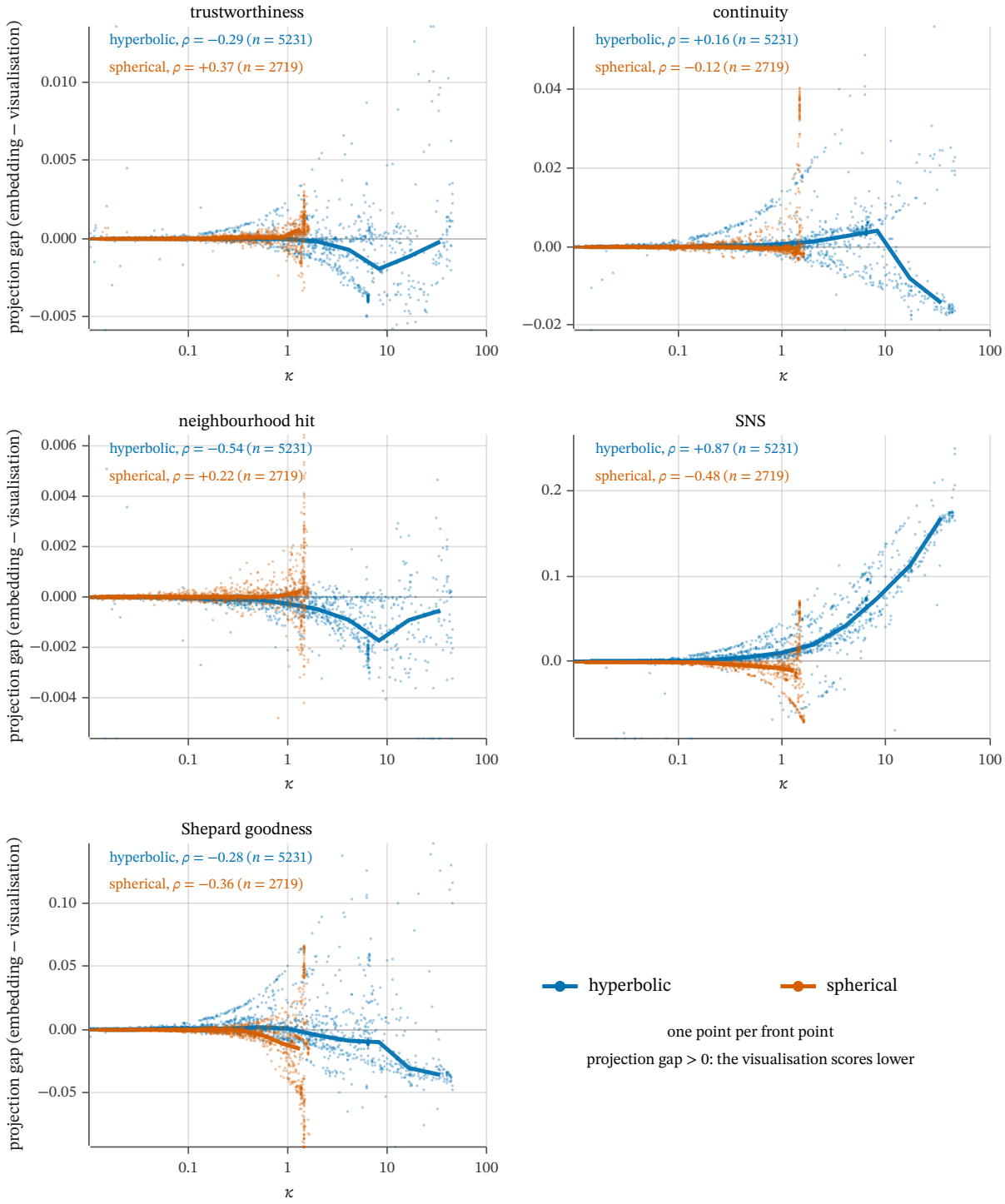


Figure 5.10: The projection gap of each metric against κ, for every front point with $\kappa \geq 0.01$. The gap is the metric computed from the geodesic distances in the embedding minus the same metric computed in the visualisation, signed so that a positive gap means the visualisation reads worse than the embedding. The line is the median gap in bins of κ, and ρ is the Spearman correlation between the gap and κ per geometry. The y range of each panel leaves out the most extreme 0.5 % of its points.

Below $\kappa \approx 1$ the projection gap is close to zero for every metric in both curved geometries, so in that range the metrics of the visualisation are the metrics of the embedding. Above $\kappa \approx 1$ the hyperbolic projection gap grows, but it stays small for the rank-based metrics: trustworthiness, continuity and the neighbourhood hit change by at most a few hundredths. SNS is the one metric with a large projection gap, and it grows steadily

with κ in the hyperbolic geometry. The projection gap of Shepard goodness is negative in the hyperbolic geometry at high κ, so there the visualisation orders distances better than the embedding does. The spherical projection gap stays close to zero up to the bound of $\kappa \approx 1.5$, where it spreads to both signs. The hyperbolic metrics of Figure 5.5 decline from $\kappa \approx 0.3$ onwards, where the projection gap is still close to zero, so that decline arises in the embedding rather than in the projection.

Together, these results answer RQ2 in three parts. The effect of the geometry is small under the distance-preservation metrics, largest under the label-aware metrics, and dataset-specific rather than tied to matching (Figure 5.8). Within a curved geometry the metrics are flat in κ over most of its range, except that strongly curved hyperbolic visualisations lose distance order and label separation, a loss that arises in the embedding rather than in the projection (Figure 5.5, Figure 5.10). Curvature does not reorder the metrics, and the correlations between them depend more on the dataset than on the geometry (Figure 5.9).

5.4 Can Curvature Be Tuned as Its Own Hyperparameter?

RQ3 asks whether the curvature $|K|$ chosen for the embedding space determines the dimensionless curvature κ of the embedding, and how curvature changes the qualitative properties of a visualisation. We examine whether adjusting curvature produces useful and reproducible changes in features such as branch separation, neighbourhood continuity, and the distribution of points across the plot. Curvature magnitude must be interpreted alongside embedding extent: Subsection 3.2.6 defines the scale-invariant quantity $\kappa = |K|\, L^2$.

5.4.1 Controlling the Curvature

Rescaling an embedding changes its curvature magnitude $|K|$ without changing κ, so tuning the curvature as its own hyperparameter requires that the chosen $|K|$ determines κ independently of the other hyperparameters. Figure 5.11 shows how κ follows $|K|$ over the front points of all corpora.

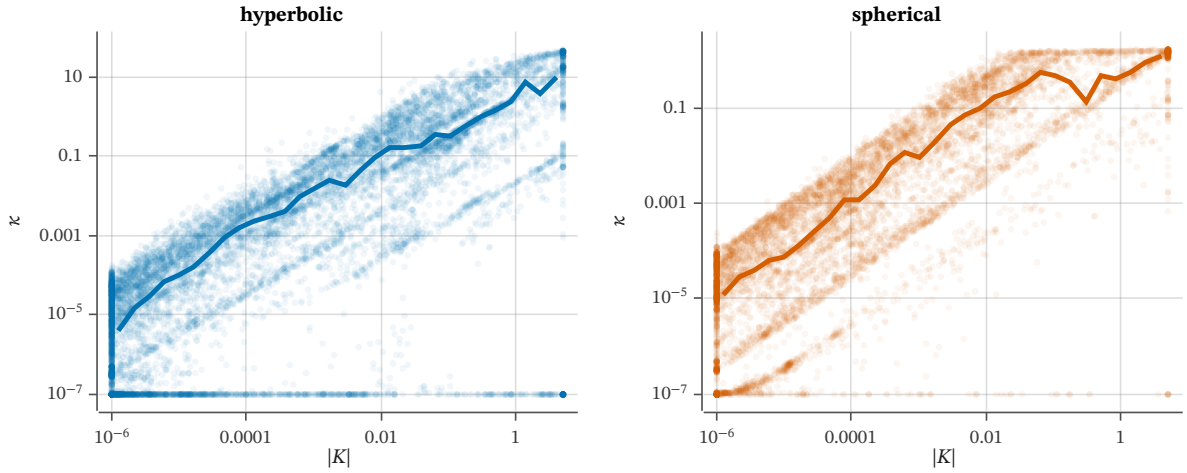
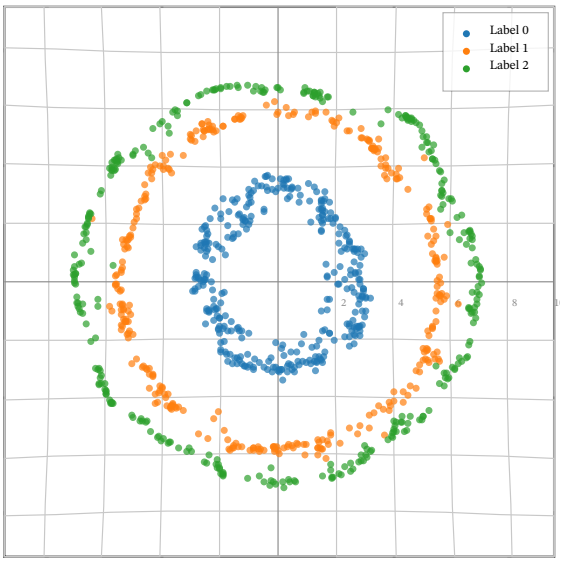


Figure 5.11: The κ of Eq. (3.16) as a result of the curvature magnitude $|K|$ of the embedding space. Hyperbolic on the left, spherical on the right; both axes are logarithmic. Every front point of every corpus is drawn faintly, and the line is the median κ in bins of $|K|$.

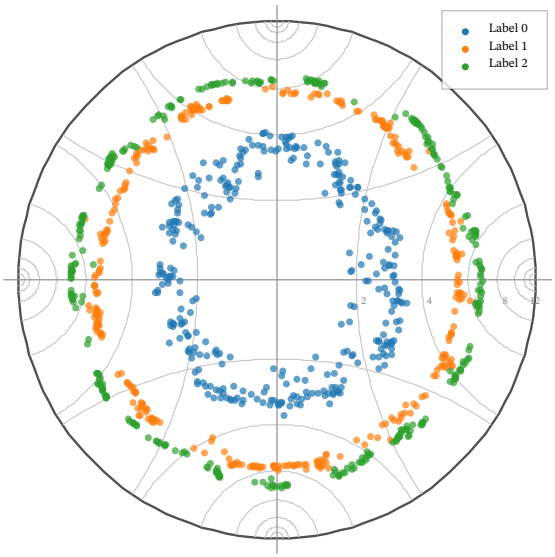
In both geometries κ increases with the curvature magnitude over the six searched decades, roughly in proportion, so $|K|$ does set the curvature the visualisation ends up with. The spread around that trend is one to two decades of κ at a given $|K|$, so the characteristic length of a visualisation still varies a lot with the other hyperparameters. Both geometries also carry a band of collapsed embeddings at $\kappa \approx 10^{-7}$, at every $|K|$. The two geometries part at the high end: in the spherical geometry κ saturates near 1.5 from $|K| \approx 1$ onwards, where the embedding covers the whole sphere (Subsection 3.2.6), while in the hyperbolic geometry it keeps growing to several tens.

5.4.2 Qualitative Changes Across Curvature Magnitudes

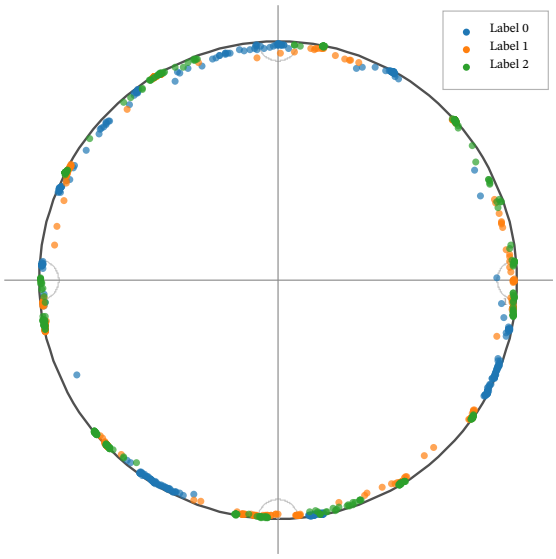
We analyse the hyperbolic shell dataset visualisations in Figure 5.12. For each visualisation, we set the perplexity to 30, the number of iterations to 800, the learning rate to 5, and the early exaggeration factor to 4. Following that, we generated three visualisations; one with $K = -0.001$, one with $K = -0.1$, and one with $K = -10$. To illustrate the qualitative features of a dataset, we have chosen to portray this with the hyperbolic shells dataset in two dimensions, as that has two properties we can clearly identify in the visualisation: The blue shell is inside of the yellow shell, which is inside of the green shell. Also, the shells are one connected cluster. These are the two properties we are looking for.



(a) $K = -0.001$.



(b) $K = -0.1$.



(c) $K = -10$.

(d) The metrics associated with each visualisation.

Hyperparameter / metric	**Fig (a)**	**Fig (b)**	**Fig (c)**
Curvature	−0.001	−0.1	−10
κ	0.030	2.901	34.611
Trustworthiness	**0.9943**	0.9854	0.9444
Continuity	**0.9981**	0.9960	0.9741
Scale-normalised stress (SNS)	**0.2494**	0.3139	0.4385
Shepard goodness	**0.9190**	0.8737	0.7439
Neighbourhood hit	**0.8204**	0.7366	0.6170
DB ratio	**1.1560**	0.9551	0.1064
Dunn index	**0.0092**	0.0056	0.0001
Cluster density measure	0.0043	0.0053	**0.0058**

Figure 5.12: A comparison of visualisations in the hyperbolic shell dataset, in a hyperbolic embedding. Three visualisations are generated with fixed hyperparameters; one with a low curvature magnitude, one with a medium curvature magnitude, and one with a high curvature magnitude.

As can be seen in the visualisations, the hyperbolic visualisation has the tendency to go towards the edges, where there is space. In the embedding there is a lot of room near the boundary, so the KL-divergence loss, which is evaluated in the embedding, favours spreading outwards. Qualitatively, we do see that for the visualisations at low and medium curvature magnitude, the hierarchy of the shells is preserved, and the shells still show as one cluster. In the visualisation at high curvature magnitude, the points are gathered at the edge of the disk and we cannot discern the shells, so the hierarchy is lost. Additionally, the shells are not a single cluster anymore, rather they are split up into multiple clusters. So on both criteria, the global structure is lost in high curvature magnitudes. Together, these results answer RQ3 in part. $|K|$ sets the κ a visualisation ends up with, but the other hyperparameters shift it by one to two decades (Figure 5.11), so curvature is not tuned independently of them. In the hyperbolic geometry low and medium curvature magnitudes keep the global structure of the shells and a high one loses it, so curvature has a useful range bounded above; the qualitative comparison covers the hyperbolic geometry only. This loss of global structure motivates RQ4.

5.5 Can Auxiliary Global Loss Terms Improve the Visualisations?

RQ4 asks whether auxiliary losses recover global structure while retaining useful local structure. We compare the standard KL-only objective (`all_off`) with the single-loss settings `centering_only`, `global_only` and `norm_only`, and with `all_free`, which optimises all applicable auxiliary weights jointly (Subsection 4.1.3). Each comparison is made within the same dataset and embedding geometry.

5.5.1 Comparison Settings

Table 5.3: Loss-weight experiment settings. In every setting the learning rate, perplexity, and early-exaggeration factor are optimised, and the curvature magnitude is optimised when the geometry is non-Euclidean. Reading the table row by row gives the ablation design: `all_off` is the standard t-SNE baseline, each `*_only` row activates a single auxiliary term, and `all_free` enables all three jointly; comparing any single-loss row against `all_off` measures one term's marginal contribution, and comparing `all_free` against the non-dominated union of single-loss corpora assesses whether joint optimisation adds useful trade-offs. Inapplicable terms must be excluded according to the geometry (Subsection 4.1.3).

Setting	λ_{cen}	λ_{glb}	λ_{norm}
`all_off`	0	0	0
`centering_only`	optimised	0	0
`global_only`	0	optimised	0
`norm_only`	0	0	optimised
`all_free`	optimised	optimised	optimised

The single-loss settings test what each auxiliary loss adds to the `all_off` corpus. The `all_free` setting tests the quality achievable when applicable auxiliary weights are optimised together. Other hyperparameters are re-optimised in each setting, so these comparisons measure the gain from allowing an auxiliary loss during the search. Not every weight acts in every geometry (Section 4.1.3.1, Section 4.1.3.3): the centering weight is inert for $K \geq 0$ and the depth-norm weight is undefined on the sphere. The Euclidean rows therefore carry `global_only`, `norm_only` and `all_free`, the spherical rows `global_only` alone, and only the hyperbolic rows all four settings.

5.5.2 Overall Gains from Each Auxiliary Loss

Figure 5.13 reports the R_2 gain of each loss setting over the `all_off` corpus of the same dataset and geometry under W_{all}, the one region under which ΔR_2 is a claim about overall front quality. The gains are multiplied by 1000 for legibility, and gains of $|\Delta R_2| < 0.001$, below 1 on that scale, are drawn on the rule and read as ties.

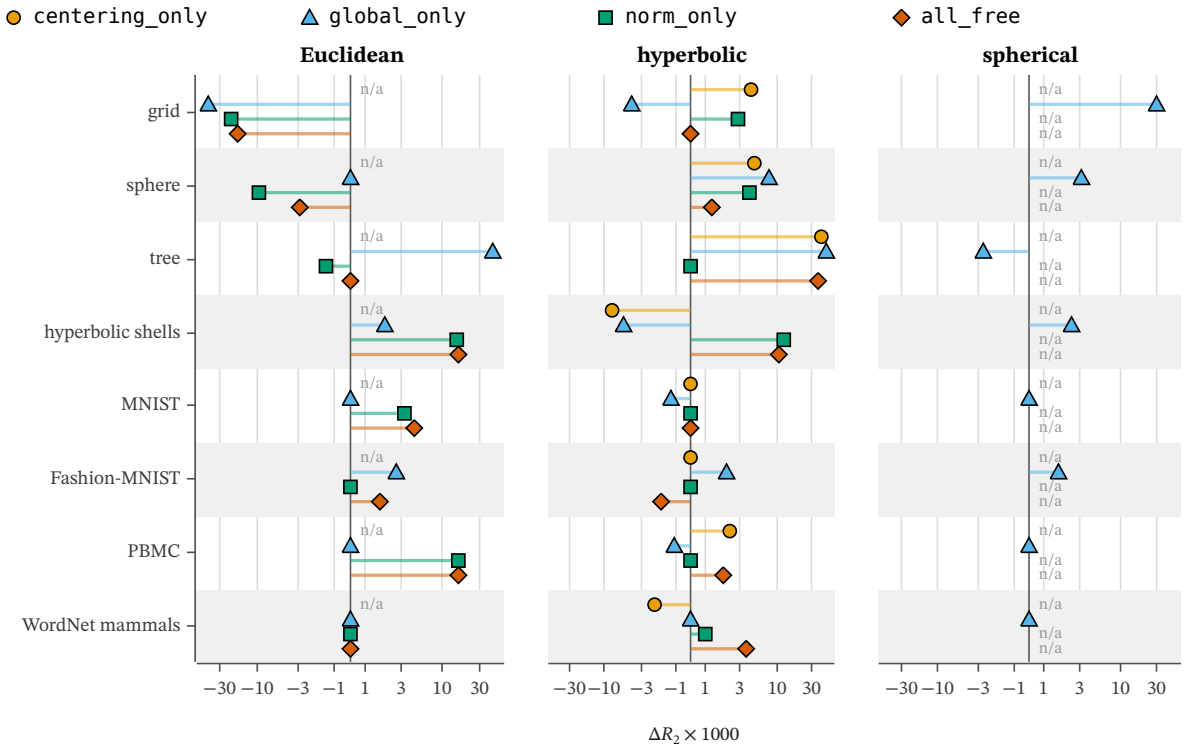


Figure 5.13: The R_2 gain for the different loss settings. A positive value means that the setting beats `all_off` (R_2 is a cost, so $\Delta R_2 = R_2(\text{all\_off}) - R_2(\text{setting})$). The gain is multiplied by 1000 for legibility. The horizontal scale is logarithmic on each side of zero, and a gain of $|\Delta R_2| < 0.001$, below 1 on this scale, is drawn on the rule and read as a tie. The symbol "n/a" marks a setting whose weight does not act in that geometry (Subsection 4.1.3): the centering weight for $K \geq 0$, the depth-norm weight on the sphere, and hence `all_free` on the sphere, where it reduces to `global_only`.

The depth-norm loss behaves differently in the two geometries in which it acts. In the hyperbolic geometry `norm_only` gains most on the hyperbolic shells and by smaller amounts on the grid, the sphere dataset and WordNet, and it loses on no dataset. In the Euclidean geometry `norm_only` gains more, on the hyperbolic shells, PBMC and MNIST, but it also loses, strongly on the grid and the sphere dataset and slightly on the tree. `all_free` follows `norm_only` in the Euclidean geometry, with gains on the same datasets. In the hyperbolic geometry `all_free` gains on most datasets, most strongly on the tree, and it keeps the gain on the hyperbolic shells even though `centering_only` and `global_only` lose there, so allowing these extra weights in the search does not remove the gain of the depth-norm loss.

The centering loss acts only in the hyperbolic geometry, and its effect there is dataset-specific: it gains on the grid, the sphere dataset and PBMC, and strongly on the tree, but loses on the hyperbolic shells and WordNet. `global_only`, which was designed to preserve global structure, mostly ties with `all_off` on the real datasets or differs from it by a few thousandths. Its larger differences are on the grid, the sphere dataset and the tree, and their sign changes with the geometry. For example, `global_only` gains on the tree in the Euclidean and hyperbolic geometries but loses on it on the sphere, and on the grid it is the other way around. The largest gains in the figure, on the tree, therefore come from `global_only`, `centering_only` and `all_free`, and not from the depth-norm loss, which ties there in the hyperbolic geometry.

R_2 averages over the weight set, so a corpus that gains much in one part of objective space and loses slightly elsewhere can register a small gain. Figure 5.14 shows the two directions of the ε-indicator of Eq. (4.22) for the same pairs. The hollow marker is $I_\varepsilon(\text{setting}, \text{all\_off})$, how far the setting falls short of `all_off`, and the filled marker is $I_\varepsilon(\text{all\_off}, \text{setting})$, how far `all_off` falls short of the setting, so a setting that covers `all_off` shows a hollow marker near zero and a filled marker far from it.

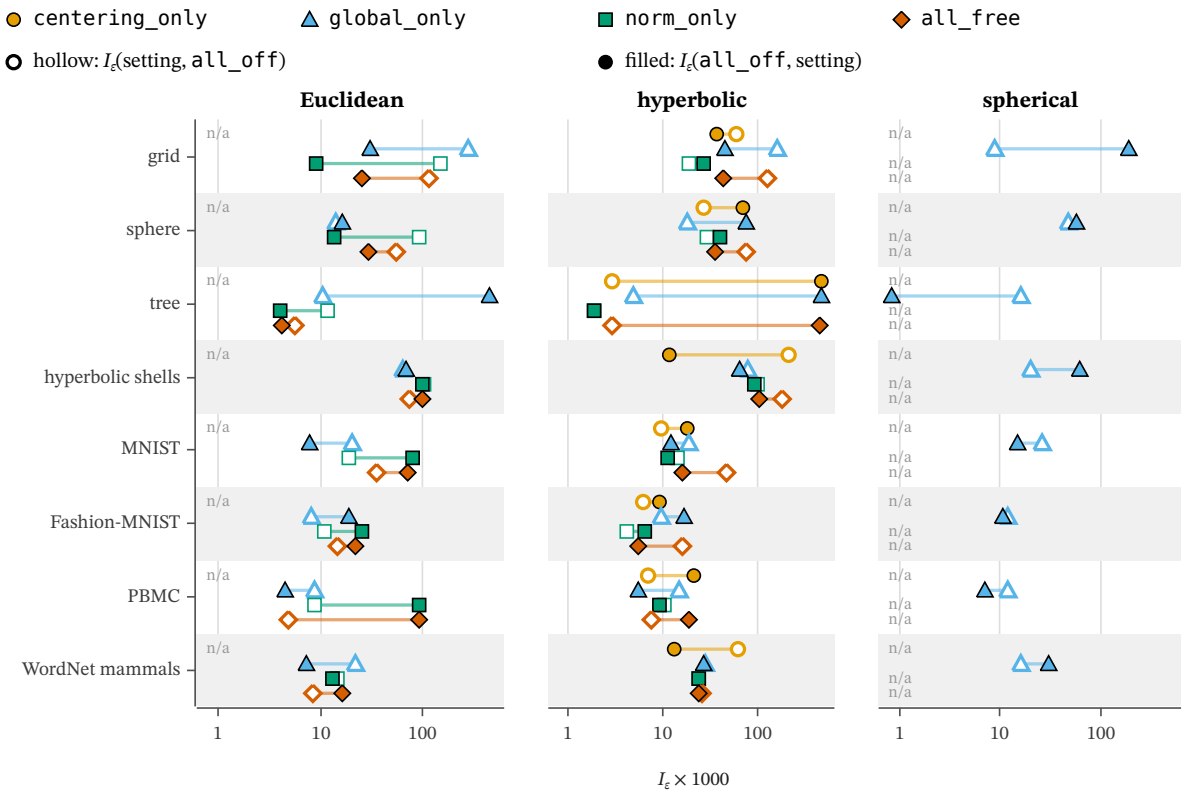


Figure 5.14: The ε-indicator values for the different loss settings. The value represents the minimum value the front needs to move in all metrics, to cover the other front. The hollow version is the ε-indicator value from the loss setting compared to the `all_off` baseline, and the filled version is the ε-indicator value from the `all_off` baseline compared to the loss setting. The horizontal scale is logarithmic. The symbol "n/a" marks a setting whose weight does not act in that geometry, as in Figure 5.13.

For PBMC in the Euclidean geometry, the ε-indicator agrees with R_2. The filled markers of `norm_only` and `all_free` are about ten times larger than the hollow ones, so these corpora nearly cover `all_off`. MNIST shows the same direction, with a smaller difference between the markers. On the hyperbolic shells, both markers of `norm_only` and `all_free` are far from zero in both geometries, which means that the corpora cross. For `all_free` in the hyperbolic geometry the hollow marker is even larger than the filled one. Although `all_free` gains on average, `all_off` still contains visualisations that `all_free` does not reach. `centering_only` on the hyperbolic shells shows the opposite case, a corpus that `all_off` nearly covers. On the tree the markers agree with the sign of ΔR_2, and the filled markers of the gaining settings in the hyperbolic geometry are the largest in the figure. On the grid both markers are large in the Euclidean and hyperbolic geometries, and for `centering_only` in the hyperbolic geometry the ε-indicator even disagrees with the sign of ΔR_2. The ties on the real datasets are corpora that cross, with both markers well above zero.

5.5.3 Global Improvements and Local Costs

We hypothesise that what we gain in distance preservation, we lose in local structure, since t-SNE optimises for local structure, and an auxiliary loss would diminish the effectiveness of focusing on it. We investigate this by plotting the R_2 value of the W_{struct} and W_{dist} preference regions against each other in Figure 5.15.

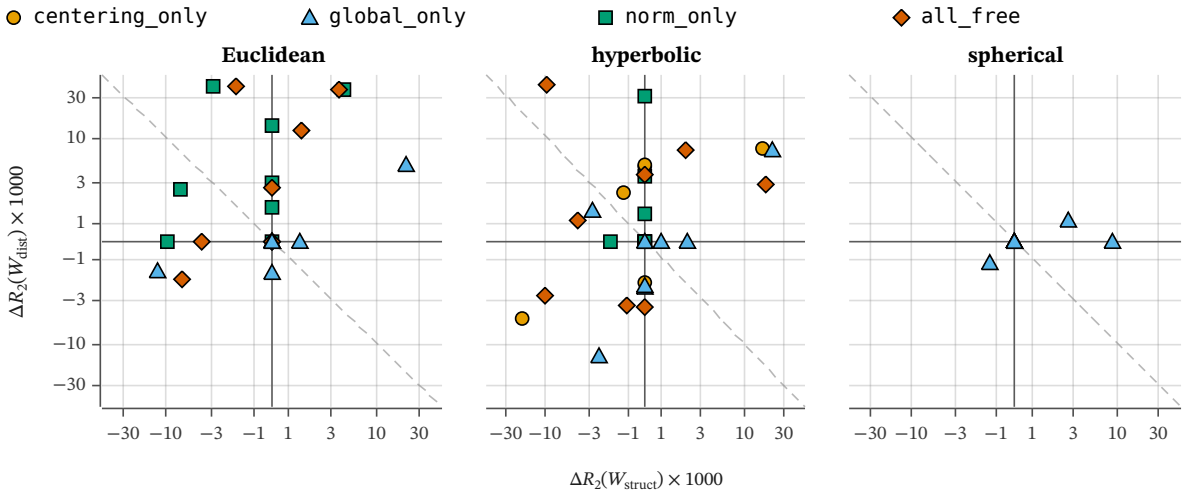


Figure 5.15: The relation between the distance-preservation and the local-structure metric family. The R_2 gain over the `all_off` setting is plotted, a positive value means that that setting has a lower R_2 value, and is therefore better. Both gains are multiplied by 1000 for legibility. The axes are plotted logarithmically, and a gain of $|\Delta R_2| < 0.001$, below 1 on this scale, is drawn on the rule and read as a tie. The dashed line is $\Delta R_2(W_{\text{dist}}) = -\Delta R_2(W_{\text{struct}})$, which is the hypothesis.

The results support our hypothesis only in part. Apart from the ties at the origin, no point lies near the dashed line, so no loss setting trades one family against the other at a one-for-one rate. The loss settings differ in the shape of their points. The depth-norm loss moves W_{dist} far more than W_{struct}: its points spread upwards, towards gains in distance preservation, while W_{struct} stays near zero in the hyperbolic geometry and ties or drops in the Euclidean geometry. Where it trades, it gives up far less local structure than it gains in distance preservation. The points of `centering_only` and `global_only` lie mostly in the quadrants where both families gain or both lose, so these losses move the local and the distance-preservation metrics together rather than against each other. `all_free` spreads the widest and shows both shapes. In the spherical geometry, where only `global_only` acts, the points stay close to the origin.

Together, these results answer RQ4: no auxiliary loss gains over the `all_off` corpus on every dataset and in every geometry, and whether a loss gains depends on the geometry. The depth-norm loss comes closest in the hyperbolic geometry, where it loses on no dataset and gains mainly in distance preservation at little cost in local structure. The centering and global losses give the largest gains, on the tree, but their sign changes with the dataset and the geometry, and several gains under R_2 are corpora that cross `all_off` (Figure 5.14).

6 Discussion

This chapter interprets the findings of Chapter 5 and places them in a wider context. First, we interpret the results per research question: whether curvature matching improves visualisation quality, which metrics curvature moves, whether curvature can be tuned as a hyperparameter, and what the auxiliary losses contribute, after which we answer the main research question. We then discuss the limitations of this research and the directions for future work that follow from them. Finally, we reflect on how this research relates to the broader computer science field, on its implications and applicability for different stakeholders, and on the use of AI in this thesis.

Throughout this chapter we use the terms introduced in Chapter 3 and Chapter 4. A corpus is the Pareto front of visualisations that the multi-objective search of Subsection 4.3.1 returns for one dataset, geometry and loss setting. Corpora are compared with the R_2 indicator and the ε-indicator of Subsection 4.3.4. A gain is a reduction in R_2, and a gain of $|\Delta R_2| < 0.001$ is read as a tie. The six optimised metrics of Section 4.2 form three families: the local-structure metrics (trustworthiness and continuity), the distance-preservation metrics (Shepard goodness and scale-normalised stress (SNS)), and the label-aware metrics (the neighbourhood hit and distance consistency). The preference regions W_m, W_{struct}, W_{dist} and W_{class} of Subsection 4.3.5 score a corpus as an analyst who cares mainly about metric m or about one family would. Finally, K is the curvature of the embedding space and $r = 1/\sqrt{|K|}$ its curvature radius. The dimensionless curvature $\kappa = |K|\, L^2$ of Eq. (3.16), with L the radius of gyration of the embedding (Eq. (3.17)), measures the extent of the embedding relative to r (Subsection 3.2.6). A small κ means that the embedding covers only a small part of the curved space, so its visualisation is nearly Euclidean whatever the value of K. Unlike $|K|$, κ does not change when an embedding is rescaled.

6.1 Interpretation of the Results

Chapter 5 compared corpora of visualisations across geometries, datasets and loss settings, with every metric computed on the visualisation. For RQ1, the matched geometry gains over both others only on the hyperbolic shells, where the spherical corpus reaches most of the same gain and most of both curved corpora is effectively Euclidean, so the gain is specific to that dataset. On the other three datasets the matched corpus loses to at least one other corpus: on the grid it gains strongly over the spherical corpus but loses to the hyperbolic one, on the sphere dataset it loses to the Euclidean corpus, and on the tree to the spherical corpus. These corpora are all computed on the high-dimensional synthetic datasets, and on their two-dimensional versions the visualisations suggest that the matched geometry is the most faithful. For RQ2, the curved corpora tie with or gain slightly on the Euclidean corpus under the distance-preservation metrics, although within the hyperbolic geometry those metrics decline at high κ. Under the local-structure metrics the corpora differ by small amounts, except for a large gain of both curved corpora on the hyperbolic shells. The label-aware differences are the largest, and they are dataset-specific. Curvature leaves the correlations between metrics largely unchanged. Those correlations depend more on the dataset than on the geometry. The hyperbolic metrics decline from $\kappa \approx 0.3$ onwards, whereas the spherical geometry, whose κ is bounded, shows no such decline. For RQ3, κ follows $|K|$ in both curved geometries, with a spread set by the other hyperparameters, and the hyperbolic qualitative comparison keeps the shell hierarchy at low and medium curvature and loses it at high curvature, where the points gather at the edge of the disk. For RQ4, the depth-norm loss never loses in the hyperbolic geometry and gains mainly under the distance-preservation metrics, while in the Euclidean geometry it gains more on some datasets but loses on others. The centering and global losses give the largest gains, on the tree, but their sign changes with the dataset and the geometry.

6.1.1 Curvature Matching

This section answers RQ1: does matching the curvature of the dataset and embedding space improve visualisation quality? The central hypothesis of this thesis, that a matching embedding curvature improves visualisation quality, is not supported by the corpora of Section 5.2, which are all computed on the high-dimensional synthetic datasets. The matched corpus gains over both others only on the hyperbolic shells, and there the spherical corpus reaches most of the same gain over the Euclidean corpus, so it is not a gain of the matching sign. Nor is it a gain of curvature: about three quarters of the front points of both curved corpora of the shells lie at κ where the embedding is effectively Euclidean (Table 5.2), so the gain is specific to this dataset. On the other synthetic datasets the differences between geometries do not favour the matched geometry, and the largest of them, on the grid, is a loss of the spherical corpus rather than a gain of the matched one. Two explanations account for why matching does not stand out, and they are not exclusive.

The first is a selection effect of the search towards small κ. The search keeps the trials whose visualisations are non-dominated, and in both curved geometries about half of the front points lie below $\kappa \approx 10^{-3}$ (Figure 5.3). At such κ the embedding extends over only a few hundredths of the curvature radius, and geodesic and Euclidean distances differ by a relative amount of the order of κ, so these visualisations act effectively as Euclidean visualisations. That half of each curved corpus can at best reproduce the trade-offs of the Euclidean corpus, so it carries no effect of matching. The other half does express the curvature, up to $\kappa \approx 1$ and beyond in the hyperbolic geometry and up to the bound in the spherical geometry, and where it moves the corpus, the direction does not follow matching: a curved corpus gains over the Euclidean one on the grid, the tree and the hyperbolic shells, and loses on the grid and the sphere dataset (Figure 5.8). This does not make the result an artefact of the search: the search keeps the best visualisations, so the trials at larger κ, which could express the curvature of the data, enter a corpus only where they reach better trade-offs, and those trade-offs do not favour the matched geometry. The projection is not the reason: below $\kappa \approx 1$ the metrics read the same in the embedding as in the visualisation (Figure 5.10), so a larger κ does not gain in the embedding either.

The second is that in the high-dimensional synthetic datasets the dimensional information outweighs the curvature information. t-SNE sees a dataset only through its neighbourhood affinities, and over the small distances that a neighbourhood spans every constant-curvature manifold is nearly flat, so curvature enters the affinities only through how neighbourhoods fit together at larger scales. In a high-dimensional dataset those larger scales are dominated by the extra dimensions, which a two-dimensional embedding has to discard whatever its curvature. Two results support this. The visualisations of the two-dimensional datasets (Figure 5.4) suggest that the matched geometry is the most faithful, whereas in the corpora of the high-dimensional datasets the matched geometry does not stand out (Figure 5.1). Independently of t-SNE, the constant-curvature fit of Chapter 8 recovers the geometry of the two-dimensional grid, sphere and tree by orders of magnitude, but on every high-dimensional synthetic dataset it favours the spherical model, whatever the generating geometry (Table 8.1). A fit that prefers the sphere for flat, hierarchical and spherical data alike does not read the curvature from the distances, and if the curvature of a dataset cannot be read from its distances, an embedding cannot match it.

Together, the two explanations account for the absence of a matching gain in Section 5.2. The first explains why about half of each curved corpus carries no effect of matching: it is nearly flat. The second explains why the differences that the strongly curved part does produce do not follow matching, even on datasets whose two-dimensional versions favour the matched geometry: in the high-dimensional versions, the distances are dominated by the extra dimensions rather than by the curvature that the embedding could match. The one matched gain, on the hyperbolic shells, is not an effect of curvature, since most of both curved corpora there is effectively Euclidean. Neither explanation depends on whether the sign of the curvature matches the data, which is why we conclude that, for the high-dimensional synthetic datasets and the search of this thesis, matching the curvature does not improve visualisation quality, and the answer to RQ1 is negative for those datasets. On the two-dimensional datasets, the matched geometry keeps the structure that

a mismatched geometry distorts (Figure 5.4): the hyperbolic visualisation separates the shells and, like the Euclidean one, keeps the levels of the tree together, the spherical visualisation separates the clusters of the sphere dataset, and on the grid the three geometries look alike. This suggests that matching the curvature does improve a visualisation as long as the curvature information in the distances is not outweighed by the dimensional information, so whether a curved geometry pays off depends on which of the two dominates the distances of the data, not only on which class of curvature the data belongs to.

6.1.2 Which Metrics Curvature Moves

This section answers RQ2: what effect does curvature have on established metrics in visualisation literature? The answer is read from the preference regions of Subsection 5.3.2, which show under which metrics a curved corpus gains or loses, and from the correlations of Subsection 5.3.3, which show whether curvature changes how the metrics relate.

At the corpus level the curved corpora tie with or gain slightly on the Euclidean corpus under the distance-preservation metrics, without following matching (Figure 5.8), and the two metrics agree in every geometry (Figure 5.9). That the differences stay small follows from the selection effect of the previous section. Both projections are monotone in distance from the centre and nearly isometric at the low κ where most front points lie (Figure 5.10), so the distances in the visualisation of a curved embedding are those of a nearly flat one, and Shepard goodness and SNS measure nearly the same distances in all three geometries. The local-structure metrics also differ by small amounts, except under trustworthiness on the hyperbolic shells, which is the gain discussed under RQ1. The label-aware metrics differ between geometries on the grid, the sphere dataset and the tree, but the sign of the difference changes with the dataset and does not follow matching. These differences therefore reflect how the labels of a particular dataset fall in a particular geometry, not a property of curvature.

Curvature does not change how the metrics relate in a way that holds across datasets. The medians differ between geometries for two pairs, most visibly trustworthiness and continuity, which are independent in the hyperbolic median and agree moderately in the other two. Per dataset, however, that pair ranges from strong opposition to strong agreement in every geometry (Figure 5.9), so the relations between the metrics depend more on the dataset than on the geometry. The projection does not produce the hyperbolic median either, since trustworthiness reads nearly the same in the embedding as in the visualisation (Figure 5.10). The opposition between Shepard goodness and the neighbourhood hit appears in the medians of all three geometries alike, so it is not caused by curvature. It is consistent with the documented behaviour of t-SNE, whose exaggeration and perplexity separate classes at the expense of the distances between them [24,35]. Within a corpus, the configurations that separate classes most are the ones that order distances worst.

The two curved geometries differ in how they can lose. The sphere is compact, so κ cannot exceed about 1.5, the value of an embedding that covers the whole sphere, and at that edge the label-aware metrics are at their highest rather than their lowest (Figure 5.5). Hyperbolic space is unbounded, so κ keeps growing with $|K|$, and from $\kappa \approx 0.3$ onwards the metrics of distance order and label separation decline together, because the KL-divergence loss of t-SNE spreads the points outwards in the embedding, and this loss is already present in the embedding before projection (Figure 5.10).

For RQ2, this means that an analyst who cares mainly about distance preservation gains at most slightly from a curved geometry, and without necessarily matching the geometry of the data. An analyst who cares mainly about local structure gains nothing that can be attributed to curvature, since the one local-structure gain, on the hyperbolic shells, is specific to that dataset. An analyst who cares mainly about the label-aware metrics has to compare geometries per dataset, since the effect of curvature on those metrics does not follow from the geometry of the data. Because the relations between the metrics depend on the dataset, no metric stands in for another in general, and the metrics of a family should be read separately rather than as one.

6.1.3 Curvature as a Hyperparameter

This section answers RQ3: can curvature be tuned as its own hyperparameter? That asks two things: whether the curvature $|K|$ chosen for the embedding space determines the dimensionless curvature κ of the embedding independently of the other hyperparameters, and whether changing $|K|$ produces a predictable change in what the visualisation shows.

Choosing $|K|$ determines the direction in which κ moves, but not its value. In both curved geometries κ grows with $|K|$ (Figure 5.11), but at a given $|K|$ it spreads over one to two decades, depending on the other hyperparameters. The extent L is an outcome of the optimisation rather than an input, so the same $|K|$ can give different values of κ, and an analyst should report κ, the scale-invariant quantity of Subsection 3.2.6, rather than K.

The visualisation does change predictably with the curvature in the hyperbolic qualitative comparison of Section 5.4. At low and medium curvature the visualisations keep the shell hierarchy and show the shells as one cluster, and at high curvature they lose both, as the points gather at the edge of the disk and the shells split into several clusters (Figure 5.12). The qualitative comparison shows that global structure is lost at high curvature, but not at which curvature between the medium and the high visualisation the loss begins.

The mechanism lies in the KL-divergence loss of t-SNE rather than in the projection. The loss is evaluated in the embedding, where the room near the boundary grows exponentially, so spreading the clusters outward lowers the loss and splits the global structure. The projection then gathers the spread points at the edge of the disk. That the projection displays this loss rather than causing it agrees with the metrics, which decline in the embedding itself (Subsection 5.3.4).

The answer to RQ3 is therefore that curvature can be tuned, but not entirely on its own. A larger $|K|$ gives a larger κ, while the other hyperparameters shift its value, so the curvature has to be reported as κ. In the hyperbolic geometry it can be tuned within a moderate range, up to at least the medium-curvature visualisation, in which the global structure is kept, and beyond it the global structure is lost. The qualitative comparison covers the hyperbolic geometry only, so that part of the answer does not extend to the spherical geometry.

6.1.4 Auxiliary Losses

This section answers RQ4: can auxiliary global loss terms improve the visualisations produced by non-Euclidean t-SNE? The depth-norm loss behaves differently in the two geometries in which it acts (Figure 5.13). In the hyperbolic geometry, for which it was designed, it is reliable: it gains on four datasets, most on the hyperbolic shells, and loses on none. In the Euclidean geometry it varies more: it gains more on the hyperbolic shells, PBMC and MNIST, but loses about as much on the grid and the sphere dataset. In the Euclidean geometry, whether the loss gains plausibly depends on whether its target describes structure the data has. For datasets given as feature vectors the target is the norm of the input vector (Section 4.1.3.3). On the sphere dataset every input has the same norm, and on the grid the norm is unrelated to the labels, so there the loss pulls the embedding towards a structure the data does not have. On the hyperbolic shells the input norm is the radius of the shell, so there the target describes structure the data does have, and the loss gains. On MNIST and PBMC the input norm varies between points and the loss gains as well, while on WordNet, the one graph dataset, whose target is the hop distance, it ties. In the hyperbolic geometry the target radii scale with the curvature radius r, so the loss prescribes an extent of the embedding relative to r, that is, a value of κ, while the metrics decline once κ exceeds about 0.3. The loss weight and the curvature then cannot be tuned independently, and a plausible consequence is that the search keeps both moderate, which limits the gains and the losses alike.

`global_only` changes sign with the dataset and the geometry. It gains strongly on the tree in the Euclidean and hyperbolic geometries and on the grid in the spherical geometry, but loses on the grid in the Euclidean geometry and on the hyperbolic shells in the hyperbolic geometry. The loss on the hyperbolic shells is plausibly because the inverted kernel is dominated by the few largest distances, which in the hyperbolic geometry

grow fastest as the embedding spreads towards the boundary. `centering_only` acts only in the hyperbolic geometry, and there its effect is dataset-specific as well. It gains on the grid, the sphere dataset and PBMC, and strongly on the tree, where it counteracts the outward drift, but it loses on the hyperbolic shells and WordNet. Which datasets it helps does not follow from their structure: the tree and WordNet are both hierarchical, yet it gains on one and loses on the other.

The hypothesised trade-off between global and local structure appears only in part. The depth-norm loss mostly gains in distance preservation while the local structure ties, and where it does trade, it gives up far less local structure than it gains (Figure 5.15). The centering and global losses instead move both families together. That joint movement is what degenerate visualisations would produce. A degenerate visualisation, such as one that has drifted to the boundary or collapsed onto a ring, scores badly on every metric at once, so a loss that prevents such visualisations gains under both families together, and a loss that causes them loses under both. And each setting re-optimises the other hyperparameters, so a trade-off that exists at fixed hyperparameters is absorbed inside the corpus. Finally, most ties are crossing corpora (Figure 5.14), meaning that a setting redistributes the front rather than moving it.

The answer to RQ4 is therefore that auxiliary losses can improve the visualisations, but no loss does so on every dataset and in every geometry. The depth-norm loss is the most reliable in the hyperbolic geometry, for which it was designed: it never loses there, and it gains mainly under the distance-preservation metrics at little or no cost in local structure. In the Euclidean geometry it gains more, but only where its target describes structure the data has. The centering and global losses give the largest single gains, on the tree, but their sign changes with the dataset and the geometry.

6.1.5 Answer to the Main Question

The main research question asks how the curvature of the embedding space influences the quality of two-dimensional visualisations. On the high-dimensional datasets the curvature of the embedding space improves the visualisation little, and not through matching: it changes the distance preservation a corpus reaches only slightly, and the matched geometry gains over both others only on the hyperbolic shells, where the spherical geometry reaches most of the same gain and most of both curved corpora is effectively Euclidean, so that gain is specific to the dataset rather than an effect of curvature. Curvature can, however, degrade a visualisation considerably. The largest differences between corpora are losses under the label-aware metrics, such as those of the spherical corpus on the grid, and their sign changes with the dataset (Figure 5.8). At high κ the KL-divergence loss spreads the points outwards in the hyperbolic embedding, the projection gathers them at the edge of the disk, and global structure is lost. The projection explains neither the small gains nor the losses: up to $\kappa \approx 1$ the metrics read the same in the embedding as in the visualisation (Figure 5.10). On data whose curvature can be recovered from its distances, the two-dimensional visualisations suggest that matching the curvature does help (Figure 5.4). The answer differs from the representation-learning literature because the criteria differ, not the surface: that literature measures downstream tasks, and this thesis measures how well a two-dimensional embedding preserves neighbourhoods, distances and labels. For the analyst, the guideline is to start from Euclidean t-SNE, since curvature improves the metrics little, can degrade them considerably, and the curvature class of the data does not predict a gain. A curved geometry is worth trying mainly when the curvature of the data can be recovered from its distances, which the constant-curvature fit of Chapter 8 achieves for two-dimensional data but not yet for high-dimensional data. A curved visualisation should report its κ, keep κ moderate in the hyperbolic geometry, where the metrics decline from $\kappa \approx 0.3$ and global structure is visibly lost at high κ, and be shown next to a Euclidean visualisation.

6.2 Limitations

This work has limitations. First, a limitation of this research is the runtime of the t-SNE algorithm. Due to the nature of this research, there is no acceleration structure for the particular Riemannian stochastic gradient descent implementation used. Although such a structure was developed [51], this does not include

momentum, or a structure for spherical data. For proper comparison, multiple large experiments, with multiple datasets must be performed. This was parallelisable due to use of the DelftBlue cluster, which has reduced simulation time a lot. In addition to the long simulation times, there is also a memory cap: The space complexity is quadratic in the number of points. In practice, running a simulation with 10000 points already gave a memory overflow on the personal machine of the researchers.

Second, RQ1 could only be posed on synthetic datasets, because the intrinsic curvature of a dataset is known only when it is generated from a manifold of known curvature. For a real dataset we cannot say which geometry is the matched one. To address this, Chapter 8 explores whether the curvature of a dataset can be estimated from its pairwise distances alone, by fitting the constant-curvature model of Wilson et al. [52] to every dataset of this thesis. The fit recovers the geometry of the two-dimensional grid, sphere and tree by orders of magnitude, but on the high-dimensional synthetic datasets and the real datasets it favours the spherical model whatever the generating geometry, and most hyperbolic fits end on the boundary of the search range (Table 8.1, Figure 8.1). The real datasets could therefore not be assigned a geometry, and the conclusions about matching rest on the four synthetic datasets.

Third, the metrics of Section 4.2 are proxies for visualisation quality, not measurements of it. They quantify properties that the literature associates with good visualisations, but whether a visualisation is useful depends on what an analyst can read from it, which can only be established with qualitative evaluation. The metric survey of Behrisch et al. [37] calls for exactly this comparison of computed measures with human perception, and t-viSNE [29] is an example of a t-SNE tool whose effectiveness was evaluated with a user study. This thesis contains no such study, so every statement about quality is a statement about the metrics.

Fourth, the metrics were designed and validated for Euclidean visualisations, and it is not known whether their values mean the same after projection from a curved space. The Poincaré projection compresses the outer region of the disk, and the azimuthal-equidistant projection compresses distances transverse to the radial direction (Subsection 3.2.5), so equal distances in the visualisation do not correspond to equal distances in the embedding. A metric value reached by a hyperbolic visualisation may therefore not describe the same visual property as the same value reached by a Euclidean one. The projection gap of Subsection 5.3.4 limits this concern: below $\kappa \approx 1$ every metric reads the same in the embedding as in the visualisation, and above it the rank-based metrics still differ by only a few hundredths, so only SNS at high hyperbolic κ is clearly changed by the projection. The gap does not show whether an analyst reads the same property from equal metric values in different geometries, and the comparisons across geometries in Chapter 5 assume that they do.

Fifth, the embedding geometries have constant curvature, so a single sign and magnitude of curvature applies to the whole embedding. Real data can mix flat, hierarchical and cyclic structure in different regions, and no constant-curvature geometry can match all of them at once. The synthetic datasets were each generated with a single geometry, so this limitation does not affect the answer to RQ1, but it limits how far that answer transfers to real data.

Finally, all results concern t-SNE. Other dimensionality-reduction methods, such as UMAP or MDS, optimise different objectives and may respond differently to the curvature of the embedding space. We chose t-SNE because non-Euclidean variants of it already exist [9,10,12,51], which allowed this work to build on and compare with previous work, but it means that the conclusions cannot be generalised to dimensionality reduction as a whole.

6.3 Future Work

Variable curvature is the most consequential extension. The constant-curvature manifolds used in this work impose a single sign and magnitude across the whole embedding, which penalises datasets whose intrinsic geometry varies between regions, such as those mixing flat, hierarchical, and cyclic structure. Allowing curvature to vary smoothly across the manifold, as in Riemannian metric learning [53], would let a single embedding accommodate hyperbolic and spherical regions at once.

The curvature framework should also transfer beyond t-SNE. UMAP, the Sammon family, and MDS all rely on pairwise-distance objectives that can be re-expressed on a constant-curvature manifold without re-deriving the loss. A parallel study across these methods would clarify whether the curvature effects reported here are specific to t-SNE or a property of distance-based embeddings in general.

The metrics introduced in Section 4.2 still need validation. They describe certain properties that we think produce good visualisations, but this might not be the case in non-Euclidean constant curvatures. A user study comparing the same dataset rendered into Euclidean, hyperbolic, and spherical embeddings would reinforce the metric-level conclusions of this thesis to the readability of the underlying visualisations.

Finally, the experiments here are bounded in scale. They cap at $N = 5000$ because the Barnes-Hut and FFT accelerations that make Euclidean t-SNE practical at larger N are not directly available for the geometries used here. A polar-quadtree acceleration structure for hyperbolic t-SNE has been developed [51], but it sits on top of the pre-correction gradient identified by [12]. Transferring it to the corrected formulation, and developing a spherical analogue, would make substantially larger experiments tractable. Alongside using an acceleration structure, it would also be a good idea to implement this framework for the GPU, which would make use of the parallel operations performed on each point.

Input dimensionality is another approach worth pursuing: a two-stage pipeline that first reduces to an intermediate dimension and then applies curvature-aware t-SNE on the result, like in Co-SNE [10], may preserve more global structure than the single-stage mapping studied here.

6.4 Reflection on Research in the Broader Computer Science Field

As stated in the introduction, embedding into Euclidean space is rarely questioned, even though it carries an implicit assumption about the geometry of the data. The most direct way to describe what this thesis does for the wider field is that it turns that unquestioned default into a hyperparameter with measurable consequences. Curvature can now be treated like perplexity or the learning rate: a setting that is chosen, swept, and reported, rather than a fixed property of the tool.

The work sits at the intersection of two research fields, which are rarely combined. The first is the move in representation learning towards hyperbolic, spherical, and mixed-curvature spaces [3,4,7,8]. This research field focuses on high quality in the embedding space, measured on downstream tasks: link prediction, classification loss, reconstruction distortion, but this research field does not investigate the visualisation qualities of non-Euclidean embeddings. The second field is the visualisation community's long-running effort to understand what t-SNE and UMAP do and do not show [19,24,35]. This thesis asks the first field's question with the second field's criteria. The answer is not simply that the benefits of non-Euclidean embeddings carry over: a geometry that suits the data on downstream tasks does not give a visualisation that is more readable, even though the metrics read almost the same in the embedding as in the visualisation (Subsection 5.3.4), and which metrics gain depends on the dataset rather than on the geometry (Subsection 5.3.2). That discrepancy is itself a finding of broader relevance, because it suggests that claims about the benefits of non-Euclidean geometry should state against which criteria the benefit was measured.

A second contribution to the field is methodological and largely independent of curvature. Dimensionality reduction has a trust problem: the plots are persuasive, the hyperparameters are influential, and the community has repeatedly warned that cluster shapes and inter-cluster distances in a t-SNE plot are easy to over-read [24,35]. Responses to that problem include quality-metric surveys [14], automated parameter optimisation [54], and interactive assessment tools [29]. This thesis has a more holistic view on visualisation quality, by treating it as a Pareto front over several metrics rather than a single score. For the field, this changes what a comparison between methods or geometries claims. Instead of one visualisation against another, it compares two corpora, each covering the trade-offs its method can reach, and a gain has to state under which preference region it holds. Figure 5.8 shows why this matters: the gain of the curved geometries on the hyperbolic shells lies in trustworthiness, and a comparison on a single distance-preservation score would have

reported at most a small gain. The approach does not depend on curvature and applies to any dimensionality-reduction method with hyperparameters, so it can make comparisons between methods harder to decide by the choice of a favourable metric or a favourable hyperparameter setting.

The limitations reported in Section 6.2 also point to a field-level observation. Euclidean t-SNE is practical at millions of points because of two decades of engineering: Barnes-Hut trees, FFT-based interpolation, and GPU implementations. None of that transfers automatically to curved spaces, and the one existing hyperbolic acceleration structure [51] predates the gradient correction of [12]. The cap at $N = 5000$ in this thesis is a consequence of the gap in acceleration structures. The adoption of non-Euclidean methods, in visualisation and elsewhere, is therefore bottlenecked as much by software ecosystems, Riemannian optimisers such as geoopt [13], and acceleration structures as by open theoretical questions. The open-source implementation released with this thesis is a small step in that direction, and the acceleration work proposed in Section 6.3 is the larger one.

Finally, the third motivating example of Chapter 1, the document vector produced by a language model, is where the tool built here may matter most in the coming years. There is growing evidence that the representations of large language models carry hierarchical, hyperbolic-like structure [55,56]. Whether that structure is visible in a two-dimensional plot, and whether a hyperbolic plot shows it more clearly than a Euclidean one, is exactly the kind of question this thesis provides a method for, although it is not one we have tested.

6.5 Reflection on Implications and Applicability

Chapter 1 described an analyst with hierarchical, cyclic, or otherwise non-flat data, who wants to know whether a non-Euclidean embedding reveals more structure than Euclidean t-SNE, and which curvature to pick. It is worth asking what that analyst can do now that they could not do before, and what they still cannot.

Before this thesis, the analyst had to choose between an established Euclidean tool and a handful of hyperbolic t-SNE variants developed for specific domains [9,10], with no spherical counterpart and no comparison on shared metrics. They now have a single implementation in which curvature is a parameter and a breakdown in Subsection 5.3.2 of the metrics under which a curved geometry gains or loses. Matching the sign of the curvature to the data does not by itself imply an improvement, so the guideline is to start from Euclidean t-SNE and to try a curved geometry mainly when the curvature of the data can be recovered from its distances, rather than to pick the curvature from the class of the data. What the analyst still cannot do is read an absolute curvature off the data: the scale invariance discussed in Subsection 3.2.6 means only the dimensionless κ is meaningful, and estimating the intrinsic curvature of a real dataset remains exploratory, as shown in Chapter 8. Nor can they yet apply any of this at the scale of the first example in Chapter 1: a single-cell dataset of tens of thousands of cells lies beyond the $N = 5000$ at which these conclusions were demonstrated.

The analyst is not the only stakeholder. The biologist, clinician, or product team who reads the resulting figure has, in most cases, been trained on Euclidean scatter plots, and brings intuitions that do not hold on a Poincaré disk or a projected sphere: equal distances in the visualisation are not equal distances in the embedding, and points near the hyperbolic boundary are much further apart than they appear. A curved embedding can remove the distortion of forcing a tree into a plane and introduce the distortion of projecting a disk onto a plane. The practical implication is that a curved visualisation should not be published bare: it needs an explicit projection convention, an indication of the curvature used, and ideally a Euclidean companion view, so that the reader knows which intuitions to suspend. Whether readers actually find curved plots more legible is an empirical question this thesis does not answer. The metrics in Section 4.2 are proxies for readability, and answering it requires the user study proposed in Section 6.3.

For those who maintain dimensionality-reduction libraries, adding curvature to the interface is simple, since it is one more argument next to perplexity. The computational cost is the harder part. Without an acceleration structure the method is quadratic in memory and slow in wall-clock time, and the experiments here

required a compute cluster (Section 6.2). An analyst with a laptop cannot currently reproduce the searches reported in Chapter 5, which limits the applicability to small datasets or to groups with cluster access. This carries an access cost as well as an energy cost: a method that needs many CPU-hours to tune a picture is a method available to few.

For researchers in geometric machine learning, the results serve as a visual sanity check on the claim that hyperbolic space "fits" hierarchies. The finding that curved corpora gain or lose under different metrics on different datasets (Subsection 5.3.2), and that the auxiliary losses either gain in distance preservation at little local cost or move both metric families together (Section 5.5), suggests that a single distortion or accuracy number understates how much geometry changes a representation.

The main risk of this work is the one t-SNE already carries, amplified. The method is known for producing clusters that look more separated than the data justifies [24,35], and curvature is one more degree of freedom with which a plot can be tuned until it tells the desired story. Another risk is that the metrics themselves were designed and validated in Euclidean space and are assumed to transfer. An analyst trusting the guidelines in this thesis is trusting that assumption.

Returning to the opening of Chapter 1: the RNA assay, the clinical record, and the document embedding. For each, the question "is there structure here that a flat plot hides?" can now be asked with a concrete tool and answered against stated criteria, at least at modest scale. Whether the answer is yes for any particular dataset remains, as it should, an empirical matter for the analyst who holds the data.

6.6 Usage of AI

In this thesis, Anthropic's Claude was used to aid in the implementation of algorithms, generating code for the final plots, and extensively commenting the codebase, based on prompts by the author. Text passages in the written thesis have been written by the author, and have been enhanced with the use of AI. All core findings, analysis and arguments reflect original academic work. No AI-generated text was directly incorporated into the final thesis without revision.

7 Conclusion

This thesis asked how the curvature of the embedding space influences the quality of two-dimensional visualisations. To examine this question, Chapter 4 generalised t-SNE to the three constant-curvature geometries by replacing the Euclidean distance in the low-dimensional kernel with the geodesic distance of the manifold and optimising the objective with Riemannian stochastic gradient descent. Because no single metric captures visualisation quality and t-SNE is sensitive to its hyperparameters, every comparison is made between visualisation corpora, the Pareto fronts over six established metrics that the multi-objective search of Subsection 4.3.1 produces per dataset, geometry and loss setting. All metrics are computed on the visualisation, since the projection of the embedding (Subsection 3.2.5) is what the analyst sees. The four research questions of Chapter 1 are answered as follows.

RQ1. On the high-dimensional synthetic datasets, matching the curvature of the embedding to the intrinsic curvature of the data does not improve visualisation quality (Section 5.2). On the grid, the sphere dataset and the tree matched and mismatched corpora differ without favouring the matched geometry, and the largest difference, on the grid, is a loss of the spherical corpus rather than a gain of the matched one. The matched corpus gains over both others only on the hyperbolic shells, and the spherical corpus reaches most of that gain as well (Figure 5.1, Figure 5.8). The gain is not one of the matching sign, nor one of curvature: about three quarters of the front points of both curved corpora of the shells lie at κ where the embedding is effectively Euclidean (Table 5.2), so the gain is specific to this dataset. On the two-dimensional versions of the synthetic datasets, whose curvature the fit of Chapter 8 recovers for the grid, the sphere and the tree, the visualisations suggest that the matched geometry is the most faithful (Figure 5.4). Matching the curvature thus appears to pay off where the curvature information in the distances of the data outweighs the dimensional information, and the finding that matching does not improve visualisation quality is specific to data in which the dimensional information dominates.

RQ2. The choice of geometry changes the distance preservation a corpus reaches only slightly: the curved corpora tie with or gain slightly on the Euclidean corpus, without following matching, and the label-aware differences between geometries are dataset-specific rather than tied to matching. The largest gain is a trustworthiness gain of both curved geometries on the hyperbolic shells, which is specific to that dataset rather than an effect of curvature (Subsection 5.3.2). Within a geometry, the metrics are flat in κ over most of its range, except that hyperbolic visualisations lose distance order and label separation above $\kappa \approx 0.3$ (Figure 5.5). Curvature does not reorder the metrics, and the correlations between them depend more on the dataset than on the geometry (Subsection 5.3.3).

RQ3. Curvature can be tuned, but not entirely on its own. In both curved geometries κ follows the curvature magnitude $|K|$ of the embedding space, up to about 1.5 in the spherical geometry, where the embedding covers the whole sphere, but the other hyperparameters shift it by one to two decades (Figure 5.11), so an analyst should report κ rather than K (Subsection 3.2.6). In the hyperbolic geometry curvature can be tuned within a moderate range: the qualitative comparison of Figure 5.12 keeps the shell hierarchy at low and medium curvature. At high curvature the KL-divergence loss of t-SNE spreads the points outwards in the embedding, the projection gathers them at the edge of the disk, and the global structure is lost. The qualitative comparison covers the hyperbolic geometry only.

RQ4. No auxiliary loss improves the visualisations on every dataset and in every geometry (Section 5.5, Figure 5.13). The depth-norm loss is the most reliable in the hyperbolic geometry, for which it was designed: it never loses there, and it gains mainly under the distance-preservation metrics. In the Euclidean geometry it gains more on the hyperbolic shells, PBMC and MNIST, but loses on the grid and the sphere dataset, plausibly because its target describes no structure of those datasets. The centering and global losses give the largest single gains, on the tree, but their sign changes with the dataset and the geometry. The hypothesised trade-off between global and local structure appears only in part: the depth-norm loss gains in distance preservation

at little or no cost in local structure, and the centering and global losses move both families together (Figure 5.15).

Taken together, on the high-dimensional datasets the curvature of the embedding space improves a visualisation little and not through matching, but it can degrade one considerably, in label separation and, at high κ, in global structure. The projection explains neither the small gains nor the losses: up to $\kappa \approx 1$ the metrics read the same in the embedding as in the visualisation (Figure 5.10), so the answer differs from the representation-learning literature because the criteria differ, not because the projection takes back what the geometry gains. On data whose curvature can be recovered from its distances, the visualisations suggest that matching does help. The practical guideline is to start from Euclidean t-SNE and to try a curved geometry mainly when the curvature of the data can be recovered from its distances. A curved visualisation should report its κ, keep κ moderate in the hyperbolic geometry, and be shown next to a Euclidean visualisation.

One premise of this thesis remains untested by the experiments above, as discussed in Section 6.2. RQ1 could be posed on synthetic datasets because their intrinsic curvature is known by construction; for a real dataset it is not. The exploratory fit of Chapter 8 recovers the curvature of the two-dimensional synthetic datasets by a wide margin, but on the high-dimensional and real datasets it favours the spherical model whatever their geometry (Table 8.1), so selecting a geometry for real data remains an open problem, which also limits how far the guideline above can be applied today.

8 Exploratory research

A central premise of this thesis is that datasets carry an intrinsic geometry of their own, which may or may not match the geometry of the embedding space. Informally, intrinsic geometry is the large-scale structure visible through pairwise distances and neighbourhood relationships. The manifold hypothesis [2] posits that high-dimensional data often lies on or near a low-dimensional manifold; if this manifold has non-zero curvature, then embedding into a space of matching curvature should produce lower distortion than embedding into Euclidean space [57].

Curvature leaves a fingerprint in pairwise distances. We read that fingerprint with the constant-curvature embedding of Wilson et al. [52], which fits a curved manifold directly to a pairwise distance matrix. The construction reads the geodesic distances as inner products. On a sphere of radius r, two points at geodesic distance d have a central angle d/r at the centre, so their ambient inner product is $r^2\cos(d/r)$; the hyperboloid of radius r carries the Lorentzian counterpart $-r^2\cosh(d/r)$. Applying either elementwise to the distance matrix gives

$$Z_{\mathrm{sph}(r)_{ij}} = r^2\cos\big(d_{ij}/r\big), \quad Z_{\mathrm{hyp}(r)_{ij}} = -r^2\cosh\big(d_{ij}/r\big), \tag{8.1}$$

where Z_{sph} is the parameterized matrix describing the distances on the hypersphere, and Z_{hyp} is the hyperbolic counterpart. The rank of Z reflects the dimensionality that the data occupies. If the points lie on a d -dimensional constant-curvature manifold of radius r, whose ambient space is $(d+1)$-dimensional, then $Z(r)$ has rank $d+1$ and every eigenvalue beyond that vanishes. The radius is what we do not know, so it becomes the fitting parameter: sweeping r and minimising the surplus eigenvalues fixes the radius, and with it the sectional curvature $K = \pm 1/r^2$.

Wilson's criterion fits the highest-dimensional constant-curvature manifold that n points can fill, namely $n-2$ (with an ambient space of $n-1$ dimensions). In this configuration, $Z(r)$ has rank $n-1$ and a single structurally zero eigenvalue, and the fit selects the radius minimising that eigenvalue's magnitude, $\arg\min_r|\lambda_1\big[Z_{\mathrm{sph}}(r)\big]|$ and $\arg\min_r|\lambda_2\big[Z_{\mathrm{hyp}}(r)\big]|$, where λ_1 is the smallest eigenvalue and λ_2 the second-smallest (the most-negative slot being occupied in the Lorentzian case).

To properly compare embeddings, we embed not into $n-2$ dimensions but into a fixed two-dimensional embedding space, the same embedding space our visualisations target. The ideal $Z(r)$ then has rank three with a fixed signature, three positive eigenvalues on the sphere (ambient $\mathbb{R}^3$) and one negative plus two positive on the hyperboloid (Lorentzian $\mathbb{R}^{2,1}$). Now $n-3$ eigenvalues should be minimised rather than one, so instead of minimising a single eigenvalue, we minimise the sum of the squares of all remaining eigenvalues. Normalising by the dimensionality n and the squared maximum pairwise distance $d^2_{\max} = \max d^2_{ij}$ gives a scale-free residual:

$$\rho(r) = \frac{\sum_{i\in\mathcal{R}}|\lambda_i|}{n * d^2_{\max}} \tag{8.2}$$

where λ_i is an eigenvalue, sorted in ascending order, and $\mathcal{R}$ is the set of indices of the eigenvalues to be minimised. In the spherical case, $\mathcal{R} = \{1, 2, \ldots, n-3\}$, which are the smallest $n-3$ eigenvalues. In the hyperbolic case, $\mathcal{R} = \{2, 3, \ldots, n-2\}$, which has the smallest eigenvalue reserved for the negative time axis.

If $\rho(r) = 0$, the data lies on a two-dimensional constant-curvature manifold of radius r, fixing the best-fit radii $r^*_{\mathrm{sph}} = \arg\min_r \rho_{\mathrm{sph}(r)}$ and $r^*_{\mathrm{hyp}} = \arg\min_r \rho_{\mathrm{hyp}(r)}$. The signed sectional curvature is then $K_{\mathrm{data}} = +1/r^{*2}_{\mathrm{sph}}$ for a spherical fit and $K_{\mathrm{data}} = -1/r^{*2}_{\mathrm{hyp}}$ for a hyperbolic fit.

8.1 Defining the search bounds

To establish a baseline of the intrinsic curvature, the Wilson estimator is run for each of the three varieties: spherical, Euclidean, and hyperbolic. We use a minimisation algorithm on the residual $\rho(r)$ (Eq. (8.2)),

iterating within the lower- and upper bounds of r to find the best fit. We write r_{low} and r_{high} for the lower and upper bound of this search window; for each curvature, they are picked differently.

As r tends to infinity, the curvature K of the space tends to 0, so the fitted space approaches flat space, both in the hyperbolic and spherical case. Any Euclidean dataset can be fit by an arbitrarily large sphere or hyperboloid with arbitrarily small residual. Mathematically, we see the same thing happening. By applying the Taylor expansion for cosine, we see that as r increases, the Z-matrix dominates in the J-matrix:

$$Z_{\text{sph}}(r) = r^2 \cos\left(\frac{D}{r}\right) \approx r^2 J - \frac{1}{2}D' + O\left(\frac{1}{r^2}\right) \quad (r \gg 1) \tag{8.3}$$

where J is the matrix of all ones, D is the distance matrix, D' is the elementwise squared distance matrix and $O\left(\frac{1}{r^2}\right)$ are the rest of the Taylor expansion terms in big O notation. A similar thing happens in the hyperbolic formulation:

$$Z_{\text{hyp}}(r) = r^2 \cosh\left(\frac{D}{r}\right) \approx -r^2 J - \frac{1}{2}D' + O\left(\frac{1}{r^2}\right) \quad (r \gg 1) \tag{8.4}$$

We can rewrite $-\frac{1}{2}D'$:

$$-\frac{1}{2}D' = A - A' \tag{8.5}$$

where $A = -\frac{1}{2}HD'H$, A being the double-centered version of D'. A is called the kernel matrix. The centering matrix is defined as $H = I - \frac{1}{n}J$. A' is related to A with the relationship $A'_{ij} = \frac{A_{ii}+A_{jj}}{2}$. We can therefore rewrite equations Eq. (8.3) and Eq. (8.4) as follows:

$$Z_{\text{sph}} \approx r^2 J + A - A', \quad Z_{\text{hyp}} \approx -r^2 J + A - A' \tag{8.6}$$

If we look at the eigenvalues of this expression, there is one eigenvalue that lives in the $\mathbf{1}$ direction, and all other eigenvalues live in the orthogonal space of $\mathbf{1}$, which is $\mathbf{1}^\perp$. $r^2 J = r^2 \mathbf{1}\mathbf{1}^T$ lives along the $\mathbf{1}$-direction, and A' actually does as well: $A' = \frac{1}{2}\left(a\mathbf{1}^T + \mathbf{1}a^T\right)$, where $a = \text{diag}(A)$. The eigenvalue in this direction increases with r, resulting in a value of $nr^2 - \text{Tr}(A)$ in the spherical case, the smallest eigenvalue $-nr^2 - \text{Tr}(A)$ in the hyperbolic case. The remaining eigenvalues must lie on the orthogonal space $\mathbf{1}^\perp$ and can be extracted from the remaining A term. Therefore we minimise the smallest $n-2$ eigenvalues of A, leaving the largest 2 eigenvalues of A to become part of the signature. Since in the limit, the spherical and hyperbolic residuals approach the same number as just embedding into an Euclidean two-dimensional space, merely judging on the magnitude of the residual does not distinguish between the three curvatures.

To meaningfully claim that data lives on a curved space, we have to set the upper bound r_{high} for $r*$. For spherical data, we can use some intuition: On a perfectly embedded sphere, the maximum distance between two points along the geodesic is equal or less than $r * \pi$. If the maximum pairwise distance spans less than x % of the theoretical maximum distance of the sphere, then we can say that the points do not live on a spherical space. For our implementation, we have set this threshold to 80%, so that $r_{\text{high}} = d_{\max}/(0.8\pi)$. A mental image for this is a bowl-like shape: the points do not wrap around, but rather approach flatness as the best embedding.

The hyperbolic case does not have such an analogy, and therefore the upper limit is hand-tuned. The upper limit is set by a minimal dimensionless curvature $\kappa_{\min}$ (Eq. (3.16), Eq. (3.17)), measured on the configuration the fit implies at radius r, which places it at $r_{\text{high}} = R_g/\sqrt{\kappa_{\min}}$. Because that configuration depends on r itself, the bound is found as a fixed point of this relation. We set $\kappa_{\min} = 0.01$. This value was chosen because empirically, the residuals of almost all datasets had already flattened out at $\kappa = 0.1$, so the extra factor of ten was taken as a margin.

On the lower side of the asymptote, the two geometries behave differently. In the spherical case, there is a hard lower limit of $r_{\text{low}} = \frac{d_{\max}}{\pi}$. This is because on a sphere, the longest geodesic is exactly halfway around the sphere. Therefore the maximum circumference of the sphere $2d_{\max} = 2\pi r$. In the hyperbolic case, there is no such analogy. Therefore we can search starting from $r_{\text{low}} = 0 + \varepsilon$, corresponding to a curvature of $-\infty$.

To find the best fit, we compare the best found residual in the spherical, hyperbolic and euclidean case. The lowest residual means the data prefers that geometry.

8.2 Does the Estimated Curvature Track the Data's Geometry?

We test whether the sign and magnitude of the dimensionless curvature κ favoured by the Wilson fit tracks the data's intrinsic geometry. As a baseline, the Wilson estimator is run on all datasets. Additionally, the 2D version of the grid, tree, and sphere synthetic datasets are added. The results can be found in table Table 8.1.

Table 8.1: Wilson fits of the thesis datasets. The hyperbolic and spherical model are fitted to different datasets ($N = 1000$) and report the normalised signature residual $\rho(r^*)$ of Eq. (8.2). For each curvature model, r^* is the fitted radius, $K = \mp 1/r^{*2}$ the signed sectional curvature it implies, and κ the dimensionless curvature (Eq. (3.16)) of the configuration the fit implies. A dagger marks a fit pinned on the upper boundary of its search window, signifying that the residual is monotonically decreasing at the bound. The spherical fits of the sphere datasets lie on the lower boundary $r = d_{\max}/\pi$ of their window, which is not marked. The hyperbolic search is capped at $r \leq R_g/\sqrt{\kappa_{\min}}$ with $\kappa_{\min} = 0.01$. The spherical search is confined to the angular-coverage window $[d_{\max}/\pi, d_{\max}/0.8\pi]$, and pinning at its upper edge means the data prefers a flatter model than a well-covered sphere.

	Residual			**Hyperbolic fit**			**Spherical fit**		
Dataset	$\rho_{\text{sph}}(r^*)$	$\rho_{\text{euc}}(r^*)$	$\rho_{\text{hyp}}(r^*)$	r^*_{hyp}	K_{hyp}	κ_{hyp}	r^*_{sph}	K_{sph}	κ_{sph}
grid 2D	$7.7 \cdot 10^{-3}$	$\mathbf{7.7 \cdot 10^{-16}}$	$1.4 \cdot 10^{-4}$	$8.20^{\dagger}$	$-0.0149^{\dagger}$	$0.0100^{\dagger}$	$1.08^{\dagger}$	$0.862^{\dagger}$	$0.571^{\dagger}$
grid 10D	**0.12**	0.13	0.13	$8.74^{\dagger}$	$-0.0131^{\dagger}$	$0.0100^{\dagger}$	$1.79^{\dagger}$	$0.310^{\dagger}$	$0.661^{\dagger}$
sphere 2D	$\mathbf{3.1 \cdot 10^{-7}}$	0.20	0.21	$11.3^{\dagger}$	$-7.82 \cdot 10^{-3\dagger}$	$0.0100^{\dagger}$	1.00	1.00	1.46
sphere 10D	**0.085**	0.28	0.28	$5.99^{\dagger}$	$-0.0279^{\dagger}$	$0.0100^{\dagger}$	1.00	1.00	1.47
tree 2D	0.23	0.24	$\mathbf{1.1 \cdot 10^{-3}}$	1.00	-1.00	65.4	4.58	0.0476	1.44
tree 10D	**0.063**	0.067	0.068	$38.8^{\dagger}$	$-6.65 \cdot 10^{-4\dagger}$	$0.0100^{\dagger}$	7.31	0.0187	0.599
hyp. shells 10D	**0.063**	0.070	0.071	$7.54^{\dagger}$	$-0.0176^{\dagger}$	$0.0100^{\dagger}$	2.53	0.156	0.276
MNIST	**0.17**	0.19	0.19	$29.8^{\dagger}$	$-1.13 \cdot 10^{-3\dagger}$	$0.0100^{\dagger}$	$5.96^{\dagger}$	$0.0282^{\dagger}$	$1.35^{\dagger}$
Fashion-MNIST	**0.083**	0.085	0.085	$56.6^{\dagger}$	$-3.12 \cdot 10^{-4\dagger}$	$0.0100^{\dagger}$	$8.25^{\dagger}$	$0.0147^{\dagger}$	$0.886^{\dagger}$
WordNet mammals	**0.16**	0.17	0.17	$33.3^{\dagger}$	$-9.01 \cdot 10^{-4\dagger}$	$0.0100^{\dagger}$	3.82	0.0685	1.29
PBMC	**0.022**	0.024	0.025	$71.5^{\dagger}$	$-1.96 \cdot 10^{-4\dagger}$	$0.0100^{\dagger}$	26.2	$1.46 \cdot 10^{-3}$	0.135

Two patterns jump out: The two-dimensional datasets fit their own geometry best by a large margin, with residuals orders of magnitude lower than those of the other curvature families. For the two-dimensional tree, whose distances are exact hyperbolic distances, the hyperbolic fit recovers the true radius $r^* = 1$ and curvature $K = -1$. Meanwhile, on every high-dimensional and real dataset the spherical model has the lowest residual, whatever the generating geometry. On the ten-dimensional sphere the margin is large, but on all other datasets the three residuals lie within about a fifth of each other, so the fit prefers the sphere for flat and hierarchical data as well, and its preference does not identify the geometry. Compared to the two-dimensional synthetic datasets, the residuals are high, implying that no two-dimensional constant-curvature model fits these datasets well. This is further shown by almost all hyperbolic fits finding a minimum at the bound of the search space, meaning the residual decreases towards the upper edge. By sweeping over the input space, this behaviour is shown more clearly. This can be seen in Figure 8.1.

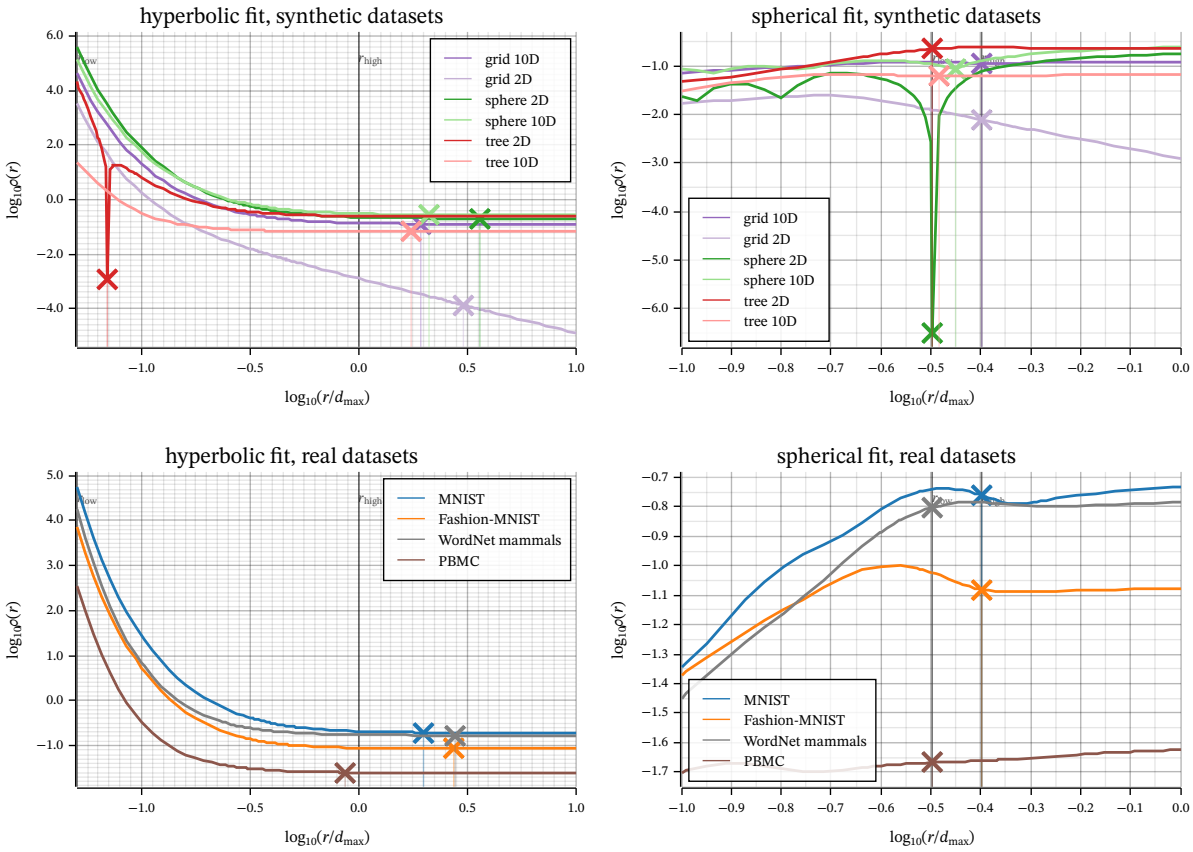


Figure 8.1: The Wilson fit residual values over r. In the left column, the hyperbolic fit is shown over synthetic datasets, in the first row, and real datasets, in the second row. In the right column, the same is plotted for the spherical fit. The vertical lines mark the search bounds r_{low} and r_{high} where they fall inside the plotted range. The two-dimensional sphere in the spherical fit and the two-dimensional tree in the hyperbolic fit show a sharp minimum at their true radius, where data and model match in dimension and curvature. The Euclidean two-dimensional dataset keeps decreasing in both the hyperbolic and the spherical case as r tends to infinity: This is because in the limit, the hyperbolic and spherical fit approach the Euclidean fit. The Euclidean fit for a two-dimensional Euclidean dataset into a two-dimensional embedding is expected to be near 0, so the residual will keep declining towards that residual as r increases. The real datasets do not show a strong preference for a certain curvature, as the residual does not dip sharply.

References

[1] L. van der Maaten and G. Hinton, Visualizing High-Dimensional Data Using t-SNE, Journal of Machine Learning Research **9**, 2579 (2008).

[2] L. McInnes, J. Healy, and J. Melville, UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction, Arxiv E-Prints (2018).

[3] M. Nickel and D. Kiela, *Poincaré Embeddings for Learning Hierarchical Representations*, in *Advances in Neural Information Processing Systems*, edited by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Vol. 30 (Curran Associates, Inc., 2017).

[4] N. He, J. Liu, B. Zhang, N. Bui, A. Maatouk, M. Yang, I. King, M. Weber, and R. Ying, Position: Beyond Euclidean – Foundation Models Should Embrace Non-Euclidean Geometries, (2025).

[5] M. Nickel and D. Kiela, *Learning Continuous Hierarchies in the Lorentz Model of Hyperbolic Geometry*, in *Proceedings of the 35th International Conference on Machine Learning*, edited by J. Dy and A. Krause, Vol. 80 (PMLR, 2018), pp. 3779–3788.

[6] I. Chami, Z. Ying, C. Ré, and J. Leskovec, *Hyperbolic Graph Convolutional Neural Networks*, in *Advances in Neural Information Processing Systems*, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, Vol. 32 (Curran Associates, Inc., 2019).

[7] A. Gu, F. Sala, B. Gunel, and C. Ré, *Learning Mixed-Curvature Representations in Product Spaces*, in *International Conference on Learning Representations* (2019).

[8] M. Yang, Y. Zhang, J. Chen, M. Weber, and R. Ying, *Towards Non-Euclidean Foundation Models: Advancing AI Beyond Euclidean Frameworks*, in *Companion Proceedings of the ACM on Web Conference 2025* (2025), pp. 1811–1814.

[9] Y. Zhou and T. O. Sharpee, Hyperbolic Geometry of Gene Expression, Iscience **24**, 102225 (2021).

[10] Y. Guo, H. Guo, and S. X. Yu, *CO-SNE: Dimensionality Reduction and Visualization for Hyperbolic Data*, in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, New Orleans, LA, USA, 2022), pp. 11–20.

[11] A. Praturu and T. O. Sharpee, Adaptive Data Embedding for Curved Spaces, Iscience **27**, 111266 (2024).

[12] H. Xia, Revisiting Hyperbolic t-SNE, 2025.

[13] M. Kochurov, R. Karimov, and S. Kozlukov, *Geoopt: Riemannian Optimization in PyTorch*, in *Proceedings of the 37th International Conference on Machine Learning* (arXiv, Vienna, Austria, 2020).

[14] M. Espadoto, R. M. Martins, A. Kerren, N. S. T. Hirata, and A. C. Telea, Toward a Quantitative Survey of Dimension Reduction Techniques, IEEE Transactions on Visualization and Computer Graphics **27**, 2153 (2021).

[15] R. Gove, L. Cadalzo, N. Leiby, J. M. Singer, and A. Zaitzeff, New Guidance for Using T-SNE: Alternative Defaults, Hyperparameter Selection Automation, and Comparative Evaluation, Visual Informatics **6**, 87 (2022).

[16] K. Pearson, LIII. On Lines and Planes of Closest Fit to Systems of Points in Space, The London, Edinburgh, And Dublin Philosophical Magazine and Journal of Science **2**, 559 (1901).

[17] W. S. Torgerson, Multidimensional Scaling: I. Theory and Method, Psychometrika **17**, 401 (1952).

[18] J. B. Tenenbaum, V. D. Silva, and J. C. Langford, A Global Geometric Framework for Nonlinear Dimensionality Reduction, Science **290**, 2319 (2000).

[19] D. Kobak and P. Berens, The Art of Using T-SNE for Single-Cell Transcriptomics, Nature Communications **10**, 5416 (2019).

[20] S. T. Roweis and L. K. Saul, Nonlinear Dimensionality Reduction by Locally Linear Embedding, Science **290**, 2323 (2000).

[21] A. Goli, M. Alizadeh, and H. S. Yazdi, Adaptive Locally Linear Embedding, Arxiv Preprint Arxiv:2504.06829 (2025).

[22] P. Joia, F. V. Paulovich, D. Coimbra, J. A. Cuminato, and L. G. Nonato, Local Affine Multidimensional Projection, IEEE Transactions on Visualization and Computer Graphics **17**, 2563 (2011).

[23] M. Balasubramanian and E. L. Schwartz, The Isomap Algorithm and Topological Stability, Science **295**, 7 (2002).

[24] H. Jeon, J. Park, S. Shin, and J. Seo, Stop Misusing T-SNE and UMAP for Visual Analytics, Arxiv Preprint Arxiv:2506.08725 (2025).

[25] E. Becht, L. McInnes, J. Healy, C.-A. Dutertre, I. W. H. Kwok, L. G. Ng, F. Ginhoux, and E. W. Newell, Dimensionality Reduction for Visualizing Single-Cell Data Using UMAP, Nature Biotechnology **37**, 38 (2019).

[26] D. Kobak and G. C. Linderman, Initialization Is Critical for Preserving Global Data Structure in Both T-SNE and UMAP, Nature Biotechnology **39**, 156 (2021).

[27] D. C. Ilie-Ablachim, B. Dumitrescu, and C. Rusu, Kernel T-Distributed Stochastic Neighbor Embedding, Arxiv Preprint Arxiv:2307.07081 (2023).

[28] J. Miller, D. Bhatia, and S. Kobourov, State of the Art of Graph Visualization in Non-Euclidean Spaces, Computer Graphics Forum **43**, e15113 (2024).

[29] A. Chatzimparmpas, R. M. Martins, and A. Kerren, T-viSNE: Interactive Assessment and Interpretation of t-SNE Projections, IEEE Transactions on Visualization and Computer Graphics **26**, 2696 (2020).

[30] P. I. Frazier, A Tutorial on Bayesian Optimization, Arxiv Preprint Arxiv:1807.02811 (2018).

[31] J. Knowles, ParEGO: A Hybrid Algorithm with on-Line Landscape Approximation for Expensive Multiobjective Optimization Problems, IEEE Transactions on Evolutionary Computation **10**, 50 (2006).

[32] S. Daulton, M. Balandat, and E. Bakshy, *Differentiable Expected Hypervolume Improvement for Parallel Multi-Objective Bayesian Optimization*, in *Advances in Neural Information Processing Systems*, edited by H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Vol. 33 (Curran Associates, Inc., 2020), pp. 9851–9864.

[33] S. Daulton, M. Balandat, and E. Bakshy, *Parallel Bayesian Optimization of Multiple Noisy Objectives with Expected Hypervolume Improvement*, in *Advances in Neural Information Processing Systems*, edited by M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Vol. 34 (Curran Associates, Inc., 2021), pp. 2187–2200.

[34] S. Daulton, D. Eriksson, M. Balandat, and E. Bakshy, *Multi-Objective Bayesian Optimization over High-Dimensional Search Spaces*, in *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, edited by J. Cussens and K. Zhang, Vol. 180 (PMLR, 2022), pp. 507–517.

[35] M. Wattenberg, F. Viégas, and I. Johnson, How to Use T-SNE Effectively, Distill **1**, e2 (2016).

[36] N. He, M. Yang, and R. Ying, *Lorentzian Residual Neural Networks*, in *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1* (Association for Computing Machinery, New York, NY, USA, 2025), pp. 436–447.

[37] M. Behrisch et al., Quality Metrics for Information Visualization, Computer Graphics Forum **37**, 625 (2018).

[38] J. Venna and S. Kaski, *Neighborhood Preservation in Nonlinear Projection Methods: An Experimental Study*, in *Artificial Neural Networks — ICANN 2001*, edited by G. Dorffner, H. Bischof, and K. Hornik (Springer, Berlin, Heidelberg, 2001), pp. 485–491.

[39] K. Smelser, K. Gunaratne, J. Miller, and S. Kobourov, How Scale Breaks "Normalized Stress" and KL Divergence: Rethinking Quality Metrics, Arxiv Preprint Arxiv:2510.08660 (2025).

[40] M. Sips, B. Neubert, J. P. Lewis, and P. Hanrahan, Selecting Good Views of High-Dimensional Data Using Class Consistency, Computer Graphics Forum **28**, 831 (2009).

[41] D. L. Davies and D. W. Bouldin, A Cluster Separation Measure, IEEE Transactions on Pattern Analysis and Machine Intelligence 224 (1979).

[42] M. P. Hansen and A. Jaszkiewicz, Evaluating the Quality of Approximations to the Non-Dominated Set, n.d.

[43] D. Brockhoff, T. Wagner, and H. Trautmann, *On the Properties of the R2 Indicator*, in *Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation* (Association for Computing Machinery, New York, NY, USA, 2012), pp. 465–472.

[44] E. Zitzler, L. Thiele, M. Laumanns, C. Fonseca, and V. da Fonseca, Performance Assessment of Multiobjective Optimizers: An Analysis and Review, IEEE Transactions on Evolutionary Computation **7**, 117 (2003).

[45] T. Wagner, H. Trautmann, and D. Brockhoff, Preference Articulation by Means of the R2 Indicator, Evolutionary Multi-Criterion Optimization **7811**, 81 (2013).
[46] M. E. Muller, A Note on a Method for Generating Points Uniformly on \emph{n} -Dimensional Spheres, Communications of the ACM **2**, 19 (1959).
[47] Y. LeCun, C. Cortes, and C. Burges, MNIST Database of Handwritten Digits, (1998).
[48] H. Xiao, K. Rasul, and R. Vollgraf, Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms, Arxiv Preprint Arxiv:1708.07747 (2017).
[49] 10x Genomics, 3k PBMCs from a Healthy Donor, Single Cell Gene Expression Dataset by Cell Ranger 1.1.0, (2016).
[50] G. A. Miller, WordNet: A Lexical Database for English, Communications of the ACM **38**, 39 (1995).
[51] M. Skrodzki, H. Van Geffen, N. F. Chaves-de-Plaza, T. Höllt, E. Eisemann, and K. Hildebrandt, Accelerating Hyperbolic T-SNE, IEEE Transactions on Visualization and Computer Graphics **30**, 4403 (2024).
[52] R. C. Wilson, E. R. Hancock, E. Pekalska, and R. P. W. Duin, Spherical and Hyperbolic Embeddings of Data, IEEE Transactions on Pattern Analysis and Machine Intelligence **36**, 2255 (2014).
[53] F. D. Giovanni, G. Luise, and M. Bronstein, Heterogeneous Manifolds for Curvature-Aware Graph Embedding, Arxiv Preprint Arxiv:2202.01185 (2022).
[54] A. C. Belkina, C. O. Ciccolella, R. Anno, R. Halpert, J. Spidlen, and J. E. Snyder-Cappione, Automated Optimized Parameters for T-Distributed Stochastic Neighbor Embedding Improve Visualization and Analysis of Large Datasets, Nature Communications **10**, 5415 (2019).
[55] J. Hsu, J. Gu, G.-H. Wu, W. Chiu, and S. Yeung, Capturing Implicit Hierarchical Structure in 3D Biomedical Images with Self-Supervised Hyperbolic Representations, (2021).
[56] S. Patil, Z. Zhang, Y. Huang, T. Ma, and M. Xu, Hyperbolic Large Language Models, (2025).
[57] I. Abraham, Y. Bartal, and O. Neiman, Local Embeddings of Metric Spaces, Algorithmica **72**, 539 (2015).

Unbounded Quality Metrics Across Curvature

A

Section 5.3 shows the cluster density measure (ClDM) as a representative example of the three metrics of Section 4.2 that are unbounded above: the Dunn index, the Davies-Bouldin ratio, and ClDM itself. For completeness, this appendix reports all three against both κ and $|K|$, in both geometries.

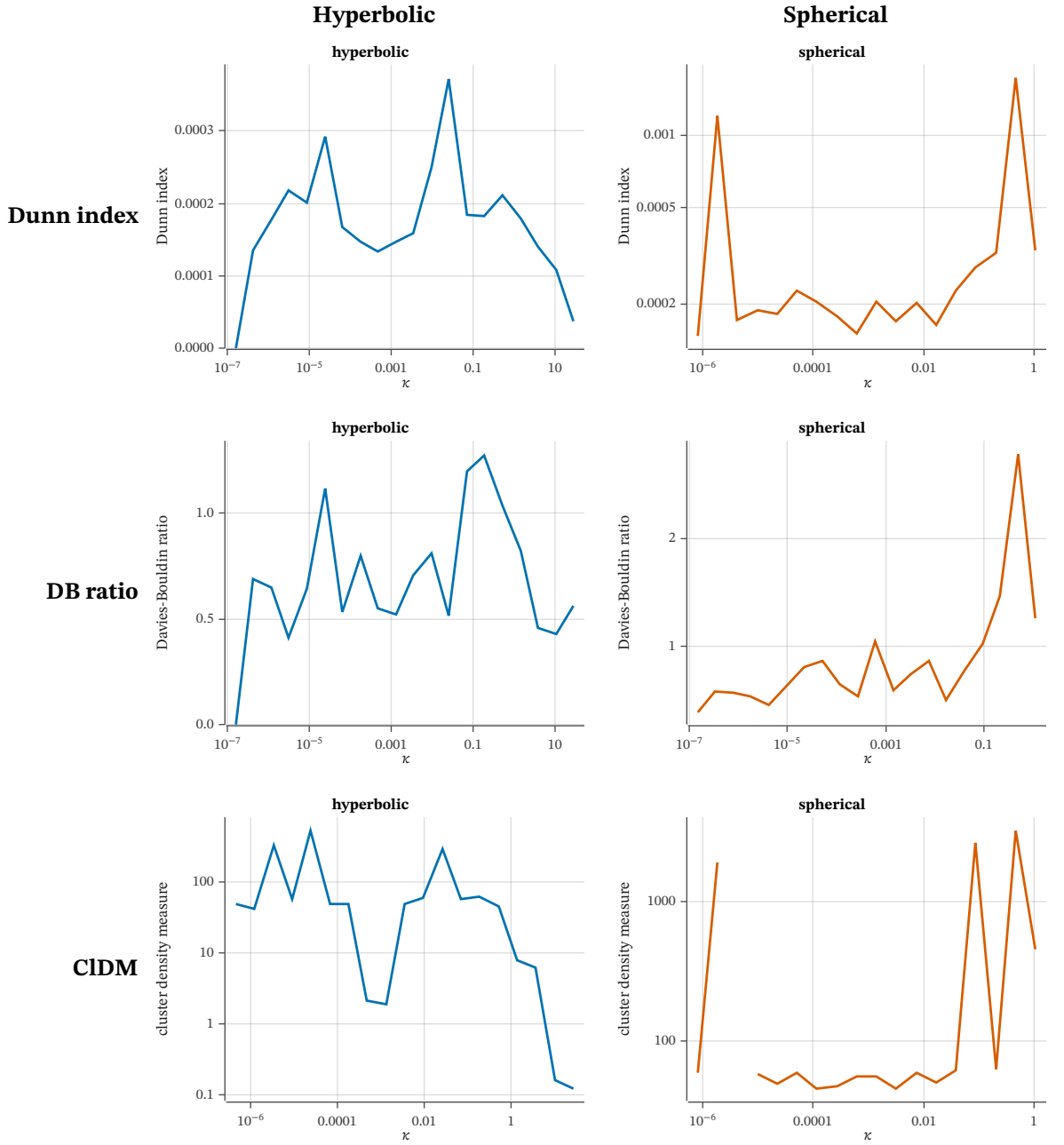


Figure A.1: The three unbounded metrics of Section 4.2 against κ, in both geometries. As in Figure 5.5, each curve is the median over the front points of the `all_off` corpora of all datasets, synthetic and real, in bins holding at least twenty points, and a gap means the bin held fewer.

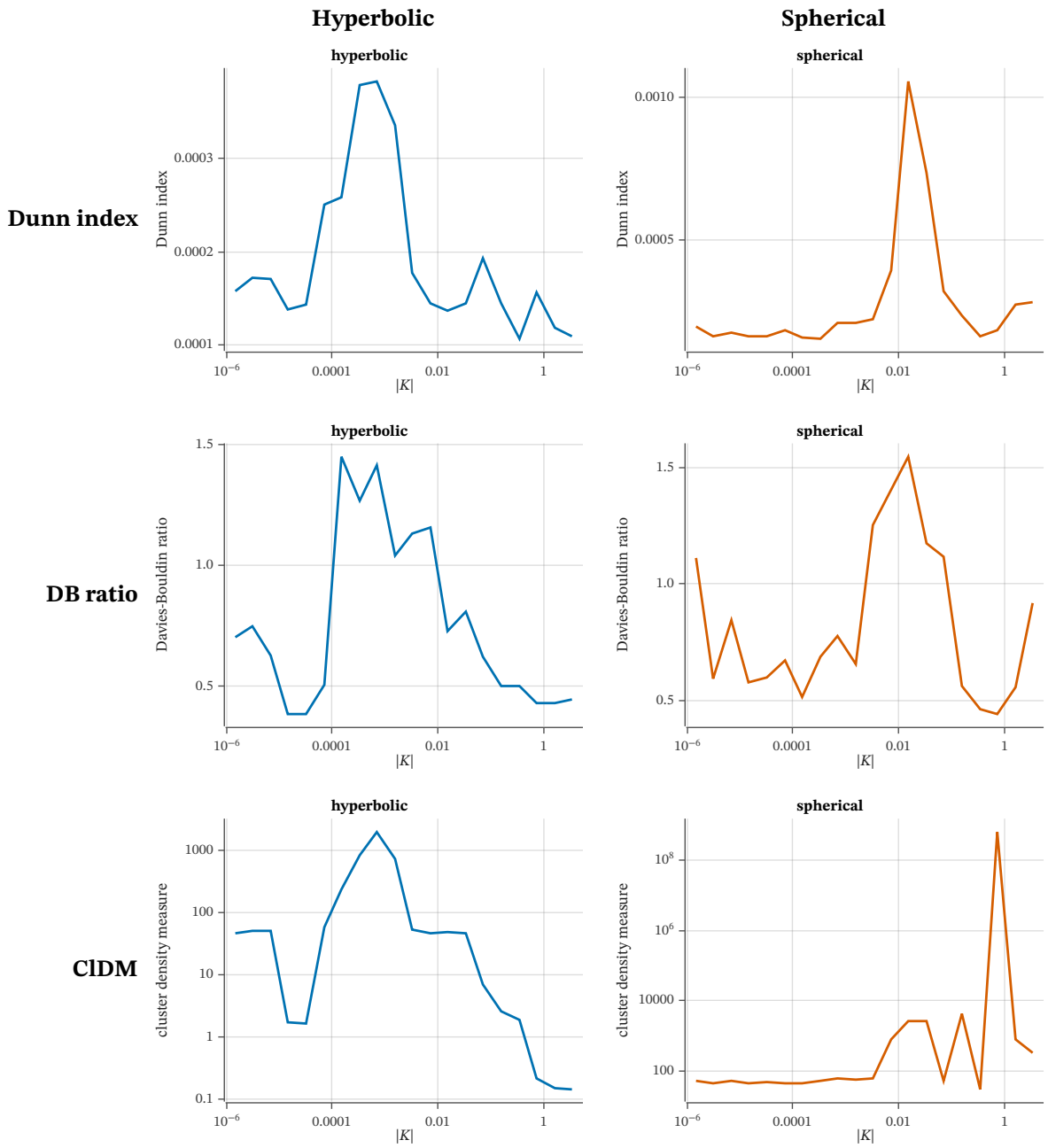


Figure A.2: The same three metrics against $|K|$, the curvature magnitude of the embedding space, in both geometries.

Derivation of Parallel Transport on the Sphere

B

We derive the closed-form expression for parallel transport along geodesics on the sphere $\mathbb{S}^n(r)$ of radius r, embedded in $\mathbb{R}^{n+1}$.

Let y, y' be two points on the sphere with $\|y\|^2 = \|y'\|^2 = r^2$, separated by geodesic angle θ where $\cos\theta = \frac{\langle y,y'\rangle}{r^2}$. Let v be a tangent vector at y, satisfying $\langle y, v\rangle = 0$.

The geodesic from y toward y' is a great circle arc lying in the plane spanned by y and y'. We decompose v into a component in this plane ($v_\parallel$) and a component orthogonal to it ($v_\perp$). The orthogonal component is unchanged by parallel transport, since the geodesic lies entirely in the plane and the orthogonal directions are not affected.

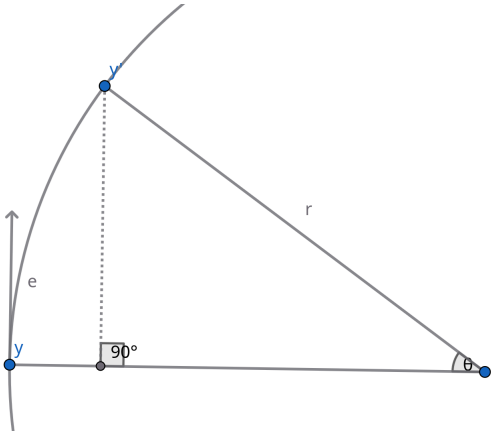


Figure B.1: The two points on the sphere lie in a plane on a great circle. The unit vector e is parallel to the direction of the projection of y' onto y, rescaled by the length of the vector to make it a unit vector.

The unit tangent to the geodesic at y pointing toward y' is:

$$e = \frac{y' - \cos\theta \cdot y}{r \sin\theta} \tag{B.1}$$

The geodesic is then parameterised as $\gamma(t) = \cos\left(\frac{t}{r}\right)y + \sin\left(\frac{t}{r}\right)re$, where t is the length along the geodesic. For circles, this length is equal to the arclength, which is $r\varphi$, where φ is the angle between the two points. The tangent of $\gamma(t)$ equals:

$$\dot\gamma(t) = -\frac{\sin\left(\frac{t}{r}\right)}{r}y + \cos\left(\frac{t}{r}\right)e \tag{B.2}$$

At $y' = \gamma(r\theta)$, the tangent direction is $\dot\gamma(r\theta) = -\frac{\sin\theta}{r}y + \cos\theta \cdot e$.

Parallel transport maps e at y to $\dot\gamma(r\theta)$ at y', while leaving the orthogonal component $v_\perp$ unchanged. Note that $v = v_\perp + v_\parallel$, and at y, $v_\parallel = \langle v, e\rangle e$. The full transport is therefore:

$$\tau_{y\to y'}(v) = v_\perp + \langle v, e\rangle\dot\gamma(r\theta) = v + \langle v, e\rangle(\dot\gamma(r\theta) - e) \tag{B.3}$$

To find $\langle v, e\rangle$, we can substitute the formula for e:

$$\langle v, e\rangle = \frac{\langle v, y'\rangle}{r\sin\theta} - \frac{\langle v, y\rangle}{r\sin\theta}\cos\theta \tag{B.4}$$

Since $\langle v, y\rangle = 0$, we have:

$$\langle v, e\rangle = \frac{\langle v, y'\rangle}{r\sin\theta} \tag{B.5}$$

Substituting and simplifying:

$$\Gamma_{y\to y'}(v) = v + \frac{\langle v, y'\rangle(\cos\theta - 1)}{r^2 \sin^2\theta}(y + y') \tag{B.6}$$

Using the identity $\frac{\cos\theta-1}{\sin^2\theta} = \frac{\cos\theta-1}{1-\cos^2\theta} = \frac{-(1-\cos\theta)}{(1-\cos\theta)(1+\cos\theta)} = \frac{-1}{1+\cos\theta}$ and substituting back $\cos\theta = \frac{\langle y,y'\rangle}{r^2}$, we obtain:

$$\Gamma_{y\to y'}(v) = v - \frac{\langle y', v\rangle}{r^2 + \langle y, y'\rangle}(y + y') \tag{B.7}$$

Centroid Alignment via Householder Reflection

C

This appendix gives the construction of the rotation used to align the embedding centroid with the projection pole for spherical embeddings, motivated in Chapter 3.

Let $\{y_i\}_{i=1}^N \in \mathbb{S}^d$ be the embedded points. The Euclidean centroid in ambient space is:

$$\hat{y} = \frac{1}{N}\sum_{i=1}^{N} y_i \tag{C.1}$$

This is not itself on $\mathbb{S}^d$, but its normalisation $c = \frac{\hat{y}}{\|\hat{y}\|}$ gives the mean direction of the data on the sphere. We seek an orthogonal transformation $H \in O(d+1)$ such that $Hc = -e_0$. The most efficient construction is a Householder reflection. Given the bisector vector:

$$v = c + e_0 \tag{C.2}$$

the Householder matrix is:

$$H = I - \frac{2vv^\top}{v^\top v} \tag{C.3}$$

which reflects any vector through the hyperplane orthogonal to v. By construction, this maps $c \mapsto -e_0$ in a single reflection. Applying it to each point:

$$y_i' = Hy_i = y_i - \frac{2\left(v^\top y_i\right)}{v^\top v} \cdot v \tag{C.4}$$

This is computed in $O(N \cdot d)$ without ever materialising the $(d+1) \times (d+1)$ matrix H. Two degenerate cases are handled explicitly:

- $\|\hat{y}\| < \varepsilon$: the data is approximately uniformly distributed on $\mathbb{S}^d$ with no preferred direction, which makes alignment unnecessary.
- $v^\top v < \varepsilon$: the centroid already coincides with $-e_0$, therefore no rotation is needed.