



GrassCaré: Visualizing the Grassmannian on the Poincaré Disk

Huanran Li^{1,3} · Daniel Pimentel-Alarcón^{2,3}

Received: 26 July 2023 / Accepted: 29 December 2023
© The Author(s) 2024

Abstract

This paper introduces a novel method for visualizing high-dimensional Grassmannians through 2D embeddings on the Poincaré disk. The proposed approach involves the construction of affinity matrices on each manifold, followed by the minimization of KL-divergence between the geodesics affinity. This process enables the identification of an optimal projection that effectively preserves the geometry of the original high-dimensional Grassmannian. Our main theoretical contribution lies in bounding the embedding loss with respect to two factors: the logarithm of the number of subspaces and a term dependent on the distribution of subspaces within the Grassmannian. Notably, this term is smaller when subspaces exhibit well-defined clusters, and larger in the absence of any discernible structure. We complement our theoretical analysis with comprehensive experiments on both synthetic and real datasets. The experimental results showcase the superiority of our embedding in accurately visualizing Grassmannians compared to existing representations.

Keywords Grassmannian · Manifold learning · Poincaré disk · t-SNE · High-dimensional data and dimensionality reduction

Introduction

Subspaces are a cornerstone of data analysis, with applications ranging from linear regression to principal component analysis (PCA) [1–3], low-rank matrix completion (LRMC) [4, 5], computer vision [6–9], recommender systems [10–12], classification [13–15], and more [16]. However, there exist few tools to visualize the Grassmann manifold $\mathbb{G}(m, r)$ of r -dimensional subspaces of $\mathbb{R}^m$. Perhaps the

most intuitive of such visualizations is the representation of $\mathbb{G}(3, 1)$ as the closed half-sphere where each point in the hemisphere represents the one-dimensional subspace (line) in $\mathbb{R}^3$ that crosses that point and the origin (see Fig. 1). While intuitive, this visualization bears two major limitations. First, this representation *wraps* around the edge, so geodesic distances can be deceiving. For instance, two points (subspaces) that may appear diametrically far may in fact be arbitrarily close (see Fig. 1). But more importantly, the main caveat of this semi-sphere representation is that it is unclear how to generalize it to $m > 3$ or $r > 1$, which makes it quite restrictive, specially for analysis of modern high-dimensional data.

In response to the limitations of classical 3D presentations, we introduce a novel approach, which we call **GrassCaré**, that addresses two crucial aspects of visualizing collections of points (subspaces) in the Grassmannian, namely

- **Efficient projection:** Our method determines the optimal angle for projecting the points onto the Grassmannian. This process can be likened to rotating the view until distances are accurately represented by geodesic distances, as depicted in Fig. 1.
- **High-dimensional support with preserved geometry:** We ensure full support for high-dimensional Grass-

This article is part of the topical collection “Recent Trends on Computer Vision, Imaging and Computer Graphics Theory and Applications” guest edited by Kadi Bouatouch, A. Augusto Sousa, Thomas Bashford-Rogers, Mounia Ziat and Helen Purchase.

✉ Huanran Li
hli488@wisc.edu

Daniel Pimentel-Alarcón
pimentelalar@wisc.edu

¹ Department of Electrical Engineering, University of Wisconsin-Madison, 415 Engineering Dr, Madison, WI 53706, USA

² Department of Biostatistics, University of Wisconsin-Madison, 610 Walnut Street, Madison, WI 53726, USA

³ Wisconsin Institute for Discovery, University of Wisconsin-Madison, 330 N Orchard Street, Madison, WI 53715, USA

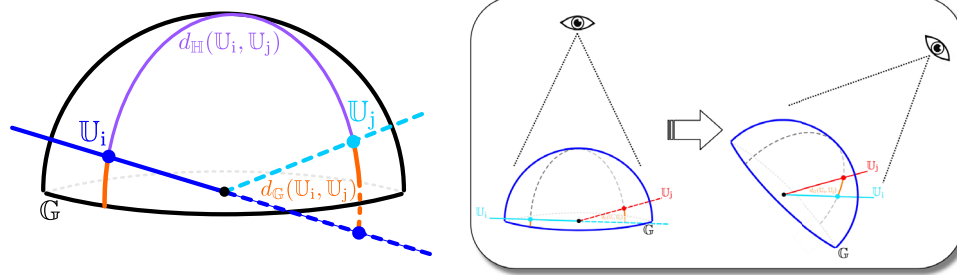


Fig. 1 *Left:* Classical 3D Representation of the Grassmannian $G(3, 1)$. Each point represents the subspace $\mathbb{U}_i$ that connects that point to the origin. This representation wraps around the edge. Two points (subspaces) that appear diametrically far w.r.t. the geodesic distance on the hemisphere ($d_H(\mathbb{U}_i, \mathbb{U}_j)$) are in fact close w.r.t. the geodesic distance on the Grassmannian ($d_G(\mathbb{U}_i, \mathbb{U}_j)$); see (1)]. An intuitive way to

mannian spaces while maintaining the geometry of the Riemannian manifold on a low-dimensional plane. This preservation is essential to prevent distortion during dimension reduction, which could otherwise lead to misleading perceptions of the scales of different subareas. This phenomenon is analogous to the challenge of representing a sphere perfectly on a 2D plane without deformation, as observed in certain world maps, affecting how humans perceive the scale of central and edge areas on the Mercator map (resulting, for example, in a common misperception of the size of the Greenland relative to the rest of the world).

To achieve these objectives, we aim to match two affinity matrices—one corresponding to the geodesics on the Grassmannian and the other to the distances on the 2D space. We minimize the discrepancy (measured by *Kullback–Leibler* (KL) divergence) between these two matrices. In doing so, we ensure that points close to each other on the Grassmannian are also close in the new 2D space. In this manner, we can obtain a 2D embedding that accurately captures the structure of points on the Grassmannian.

To replicate the landscape of the Grassmannian, we utilize the Poincaré disk as the space of the 2D embeddings. The Poincaré disk is a 2D hyperbolic geometric model, often represented as a unit circle where the geodesic distance between two points in the disk is depicted as the circular arc orthogonal to the unit circle [17]. This representation corresponds to the projection of the hyperbolic arc of their geodesic (see Fig. 2 for intuitive understanding). Essentially, the Poincaré disk provides a view from the bottom of the 3D hyperbolic bowl, sharing significant structural similarity with the hemisphere in Fig. 1.

Compared to the Euclidean space, which requires deformation to generate embeddings, the Poincaré disk allows

see this is to extend the lines to the opposite side of the hemisphere and compute their smallest angle. *Right:* Our method introduces an efficient projection technique that determines the optimal angle for projecting points onto the Grassmannian. Conceptually, this can be likened to skillfully rotating the view until the distances between points are accurately represented by geodesic distances

the embeddings to have a focused area (central area) that is resistant to deformation from hyperbolic effects and a background area (around the edges) where only the global structure is significant. This characteristic enables us to maintain an accurate global representation of the Grassmannian within a unit circle while simultaneously preserving local structures.

Our main theoretical result shows that the loss of our embedding (measured in terms of the KL-divergence *with respect to* (w.r.t.) the Grassmannian geodesics) is bounded

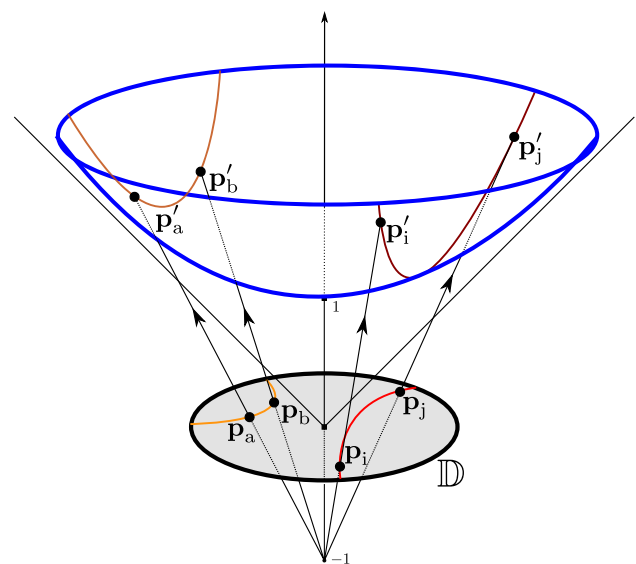


Fig. 2 Geodesics in the Poincaré disk $\mathbb{D}$. The geodesic distance $d_D(\mathbf{p}_i, \mathbf{p}_j)$ is given by the Euclidean length of the hyperbolic arc between $\mathbf{p}_i$ and $\mathbf{p}_j$ and is often depicted in the disk by the arc between $\mathbf{p}_i$ and $\mathbf{p}_j$ (and similarly for $d_D(\mathbf{p}_a, \mathbf{p}_b)$). Points closer to the disk's boundary will be projected higher on the hyperbolic space, resulting in larger distances [see (2)]. In words, distances near the edge are larger than they appear

by a log-factor of the number of subspaces, and a term that depends on the distribution of the subspaces in the Grassmannian. This term will be smaller if the subspaces form well-defined clusters, and it will be larger if the subspaces have no structure whatsoever. In words, this result shows that under reasonable assumptions, our embedding can be an accurate representation of the Grassmannian. Equipped with this result, we believe that the GrassCaré embedding can be a powerful tool for subspace tracking, classification, multi-dataset analysis, and any application where there is an interest in visualizing subspaces.

Paper Organization In the next section, we discuss several applications of our GrassCaré embedding. The subsequent section briefly summarizes related work followed by which we introduce the main formulation that determines our embedding, together with the gradient steps for the optimization. Then our main theorem, bounding the loss of our embedding, followed by its proof is presented. Finally, we demonstrate the applicability of our GrassCaré embedding on real and synthetic data, we compare it to naive alternatives, and we discuss its advantages and limitations in the final section.

Applications

Our GrassCaré embedding could be a valuable tool in the following applications:

Subspace Clustering involves clustering a collection of data points $\mathbf{x} \in \mathbb{R}^m$ that are located near a union of subspaces [18]. The primary objective is to estimate such union. This technique finds applications in various domains such as motion segmentation [19, 20], face clustering [21], data mining [22], time series analysis [23], among others.

Our experiments demonstrate that the GrassCaré embedding we propose provides valuable insights and summaries regarding the characteristics and relationships among the clusters. By employing this embedding, we can analyze the results of a subspace clustering algorithm in a more comprehensive manner, moving beyond simple accuracy metrics. Moreover, it serves as a useful tool for debugging and gaining a deeper understanding of algorithmic performance. In fact, the analysis and comprehension of subspace clustering algorithms served as the initial motivation behind this research.

Low-Rank Matrix Completion is a fundamental problem that tackles the challenge of recovering missing entries in a low-rank matrix $\mathbf{X}$ [24]. The goal is to unveil the underlying structure by identifying the low-dimensional row and column spaces of $\mathbf{X}$. This versatile technique finds

applications in various domains, making it indispensable for recommender systems in e-commerce [25], enhancing image processing algorithms in computer vision [26], accelerating drug discovery processes in pharmaceutical research [27], and even enabling effective analysis of electronic health records (EHR) [28].

Our novel contribution lies in the utilization of the GrassCaré embedding, which emerges as a powerful tool in the analysis of LRMC algorithms. Through a series of carefully designed experiments, we demonstrate how our approach sheds light on the algorithmic behavior and performance of LRMC methods. By traversing the Grassmannian, our embedding offers valuable insights into crucial aspects such as the proximity of the completed matrix to the ground truth, convergence behavior, step sizes during optimization, and discernible patterns that arise throughout the iterative process. This comprehensive analysis not only enriches our understanding of LRMC algorithms but also empowers researchers and practitioners to fine-tune and enhance the performance of these methods in real-world scenarios.

Subspace tracking is a dynamic process that involves the continuous estimation of a subspace $\mathbb{U}_t$ as it evolves over time, navigating through the intricate landscape of the Grassmannian [3, 29, 30]. This versatile modeling technique finds its application in various domains, including signal processing [31], low-rank matrix completion [32], and computer vision [29]. For instance, in computer vision, subspace tracking enables the estimation of the underlying subspace corresponding to the background in a moving video scene.

In this context, our proposed GrassCaré embedding emerges as a powerful tool for monitoring the trajectory of the subspace as it traverses the Grassmannian. By harnessing the capabilities of our embedding, we gain valuable insights into the behavior of the evolving subspaces. This encompasses the assessment of their movement speed, spatial distance covered, and identification of distinctive patterns such as zig-zag or cyclic motion. Through the lens of the GrassCaré embedding, we can unravel the dynamic nature of the subspace path, providing a deeper understanding of its evolution and behavior.

Multi-Dataset Analysis is a crucial area of study that explores the relationships and interactions between multiple datasets. Principal Component Analysis (PCA) stands as the cornerstone of dimensionality reduction techniques and finds widespread application in diverse fields. From electronic health records (EHR) analysis [28] to genomics research [33, 34], and even vehicle detection systems [35, 36], PCA has proven its versatility and utility. By identifying the low-dimensional subspace that best captures the essential

structure of high-dimensional datasets, PCA enables efficient data representation.

In contemporary scenarios, multiple datasets often exhibit distributional patterns or possess underlying relationships. For example, the EHRs of different geographical regions might share significant correlations. However, due to challenges such as privacy concerns, security considerations, data size, ownership, and logistical constraints, sharing such information becomes daunting, if not impossible. Nonetheless, the principal subspaces derived from each dataset can be shared more efficiently, mitigating these concerns and offering a potential avenue for gaining insightful knowledge. Leveraging our GrassCaré embedding, we present a visualization tool that facilitates the analysis of interrelationships between related datasets. By examining the embedding, we can unveil similarities, identify clusters, and detect meaningful patterns, unlocking a wealth of valuable insights.

Prior Work

To the best of our knowledge, general visualizations of Grassmannians have been studied using *Self-Organizing Mappings* (SOM) [37], which were introduced first for general dimensionality reduction [38–41]. The extension of SOM to Grassmannians iteratively updates points on a 2D index space to find the best arrangement, such that points that are neighbors in the Grassmannian are still close in the embedding. However, SOM presents several limitations. For instance, like most neural networks, they require large datasets, which may not always be available in practice. They also suffer from large parameter spaces, and are quite difficult to analyze, making it hard to derive theoretical guarantees about the accuracy of their embeddings. It is worth mentioning that there are numerous methods for general high-dimensional data visualization, including UMAP [42], LargeVis [43], Laplacian eigenmaps [44, 45], isomap [46], and more [47–50]. However, since these embeddings are not compact, they are not appropriate to represent the Grassmannian.

Another more suitable alternative are the Grassmannian Diffusion Maps (GDMaps) [51] introduced as an extension of Diffusion Maps [52]. GDMaps consist of two separate stages. The first stage projects the given data point (i.e. vector, matrix, tensor) onto the Grassmannian using a singular value decomposition. The second stage uses diffusion maps to identify the subspace structures on the projected Grassmannian. Although the embedding can be quickly generated, it is unfortunately less accurate than other methods in this paper. On the other hand, *Stochastic Neighbor Embeddings* (SNE) were first presented by Hinton and Roweis [53]. It formed the basis for t-SNE, which was introduced later by

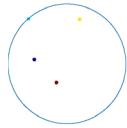
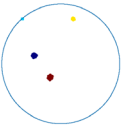
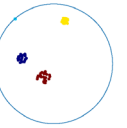
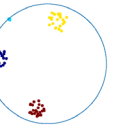
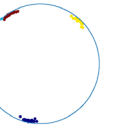
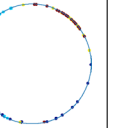
Maaten and Hinton [54]. Both algorithms minimize the KL-divergence between the distributions representing the probability of choosing the nearest neighbor on the high and low-dimensional spaces. These embeddings have become some of the most practical tools to visualize high-dimensional data on Euclidean space. However, Euclidean distances are poor estimators of geodesics of Grassmannians, so a direct application of these methods would result in an inaccurate representation of subspaces arrangements.

Motivated by these issues we decided to explore the use of the *Poincaré disk*, which has recently received increasing attention for high-dimensional embeddings [55, 56]. Intuitively, the Poincaré disk is a 2D hyperbolic geometric model, usually displayed as a unit circle where the geodesic distance between two points in the disk is represented as the circular arc orthogonal to the unit circle [17], which corresponds to the projection of the hyperbolic arc of their geodesic (see Fig. 2 to build some intuition). This unique feature brings several advantages for serving as the embedding space for Grassmannian. First, since these hyperbolic arcs get larger (tending to infinity) as points approach the disk boundary, the Poincaré disk is an effective model to accurately represent the global structure of complex hierarchical data while retaining its local structures. Specifically, the Poincaré disk can be viewed as a continuous embedding of tree nodes from the top of the tree structure, where the root node is at the origin, and the leaves are distributed near the boundary. So, it is naturally suited to represent hierarchical structures. This is suitable to represent structured clusters, where the points from the same cluster can be regarded a branch of the tree, because they share a similar distance to other clusters. Second, the hyperbolic disk has a Riemannian manifold structure that allows us to perform gradient-based optimization, which is crucial to derive convergence guarantees, and for parallel training of large-scale dataset models. Finally, our main result showing the accuracy of our embedding enables efficient clustering using the Poincaré low-dimensional representation. That is, instead of clustering subspaces on the high-dimensional dataset, the clustering method can be performed on the mutual distances acquired from the embedding, with the knowledge that the embedding would represent the high-dimensional subspace accurately enough.

Setup and Formulation

In this section, we present the mathematical formulation of our GrassCaré embedding. To this end let us first introduce some terminology. Recall that we use $\mathbb{G}(m, r)$ to denote the Grassmann manifold that contains all the r -dimensional subspaces of $\mathbb{R}^m$. For any two subspaces $\mathbb{U}_1, \mathbb{U}_2 \in \mathbb{G}(m, r)$, the *geodesic distance* between them is defined as:

Table 1 The embeddings of synthetically generated subspaces with varying β values exhibit distinct characteristics

β	0.001	0.01	0.1	1	10	100
Embedding						

As β increases, the embeddings on the Poincaré disk become more widely dispersed

$$d_{\mathbb{G}}(\mathbb{U}_i, \mathbb{U}_j) := \left(\sum_{\ell=1}^r \arccos^2 \sigma_{\ell}(\mathbf{U}_i^T \mathbf{U}_j) \right)^{\frac{1}{2}}, \quad (1)$$

where $\mathbf{U}_i, \mathbf{U}_j \in \mathbb{R}^{m \times r}$ are orthonormal bases of $\mathbb{U}_i, \mathbb{U}_j$, and $\sigma_{\ell}(\cdot)$ denotes the ℓ th largest singular value. As for the embedding space, recall that the Poincaré disk $\mathbb{D}$ is the Riemannian manifold defined as the open unit ball in $\mathbb{R}^2$ equipped with the following distance function between two points $\mathbf{p}_i, \mathbf{p}_j \in \mathbb{D}$:

$$d_{\mathbb{D}}(\mathbf{p}_i, \mathbf{p}_j) := \operatorname{arcosh} \left(1 + 2 \frac{\|\mathbf{p}_i - \mathbf{p}_j\|^2}{(1 - \|\mathbf{p}_i\|^2)(1 - \|\mathbf{p}_j\|^2)} \right). \quad (2)$$

Notice from (2) that the geodesic distance in the disk is amplified smoothly as $\mathbf{p}_i$ or $\mathbf{p}_j$ move away from the origin. Intuitively, this means that an arc of the same Euclidean length in the disk represents a larger geodesic distance (tending to infinity) as it approaches the edge of the disk. In other words, distances near the edge of the disk are larger than they appear (see Fig. 4 to build some intuition). Conversely, distances at the center of the disk are smaller than they appear. This allows to plot denser regions of the Grassmannian with higher granularity (thus retaining *local* structure) while at the same time keeping an accurate *global* representation of the Grassmannian inside an open circle.

To find our embedding, we will mimic the symmetric SNE approach in [54]. That is, we will first compute a *probability* matrix $\mathbf{P}_{\mathbb{G}} \in [0, 1]^{N \times N}$ whose (i, j) th entry represents the probability that $\mathbb{U}_i$ is chosen as a nearest neighbor of $\mathbb{U}_j$, which is equal to zero if $i = j$, and for $i \neq j$ is given by:

$$\begin{aligned} [\mathbf{P}_{\mathbb{G}}]_{ij} := & \frac{1}{2N} \frac{\exp(-d_{\mathbb{G}}(\mathbb{U}_i, \mathbb{U}_j)^2 / 2\gamma_i^2)}{\sum_{k \neq i} \exp(-d_{\mathbb{G}}(\mathbb{U}_i, \mathbb{U}_k)^2 / 2\gamma_i^2)} \\ & + \frac{1}{2N} \frac{\exp(-d_{\mathbb{G}}(\mathbb{U}_j, \mathbb{U}_i)^2 / 2\gamma_j^2)}{\sum_{k \neq j} \exp(-d_{\mathbb{G}}(\mathbb{U}_j, \mathbb{U}_k)^2 / 2\gamma_j^2)}, \end{aligned} \quad (3)$$

where γ_i is adapted to the data density: smaller values for denser regions of the data space. In our experiments, we choose it to be the variance of distances from point i to other points. Next we create the *probability* matrix $\mathbf{P}_{\mathbb{D}} \in [0, 1]^{N \times N}$,

whose (i, j) th entry represents the probability that point $\mathbf{p}_i$ in our embedding $\mathbb{D}$ is chosen as a nearest neighbor of point $\mathbf{p}_j \in \mathbb{D}$, which is equal to zero if $i = j$, and for $i \neq j$ is given by:

$$[\mathbf{P}_{\mathbb{D}}]_{ij} := \frac{\exp(-d_{\mathbb{D}}(\mathbf{p}_i, \mathbf{p}_j)^2 / \beta)}{\sum_{k \neq i} \exp(-d_{\mathbb{D}}(\mathbf{p}_i, \mathbf{p}_k)^2 / \beta)}. \quad (4)$$

The tuning parameter β , which is set to be any positive real number, serves as a control factor for the scattering of the embedding [56]. A higher value of β results in a wider spread of the embeddings on the Poincaré disk. To examine the influence of various β values on the final output, we carried out a series of experiments in the following sections (the outcomes are presented in Table 1). Our findings indicate that setting β to an excessively large value, such as 100, results in poor embeddings. Generally, to achieve favorable results, it is recommended to keep β within the range of 1 to 2, as also suggested in [56].

Our goal to obtain the embedding is to maximize the similarity between the two distributions $\mathbf{P}_{\mathbb{G}}$ and $\mathbf{P}_{\mathbb{D}}$, which we do by minimizing their *Kullback–Leibler* (KL) divergence:

$$\operatorname{KL}(\mathbf{P}_{\mathbb{G}} \parallel \mathbf{P}_{\mathbb{D}}) = \sum_{i,j} [\mathbf{P}_{\mathbb{G}}]_{ij} \log \frac{[\mathbf{P}_{\mathbb{G}}]_{ij}}{[\mathbf{P}_{\mathbb{D}}]_{ij}}.$$

Since $\mathbf{P}_{\mathbb{G}}$ is a constant given $\{\mathbf{U}_i\}$, this is the same as minimizing the following *loss*

$$\mathcal{L} = - \sum_{i,j} [\mathbf{P}_{\mathbb{G}}]_{ij} \log [\mathbf{P}_{\mathbb{D}}]_{ij}.$$

To minimize this loss over the Poincaré disk $\mathbb{D}$ we will use Riemannian Stochastic Gradient Descent [57], which updates $\mathbf{p}_i^{t+1}$ according to:

$$\mathbf{p}_i^{t+1} \leftarrow R(\mathbf{p}_i^t - \eta \nabla_i \mathcal{L}), \quad (5)$$

where $\eta > 0$ is the step size (set as $\eta = 1$ in the implementation), $\nabla_i \mathcal{L}$ denotes the Riemannian gradient of $\mathcal{L}$ w.r.t. $\mathbf{p}_i$, and R denotes a *retraction*¹ from the tangent space of $\mathbf{p}_i$ onto $\mathbb{D}$. It is easy to see that

¹ A mapping R from the tangent bundle $T\mathcal{M}$ to the manifold $\mathcal{M}$ such that its restriction to the tangent space of $\mathcal{M}$ at $\mathbf{p}_i$ satisfies a local rigidity condition which preserves gradients at $\mathbf{p}_i$; see Chapters 3 and 4 of [58] for a more careful treatment of these definitions.

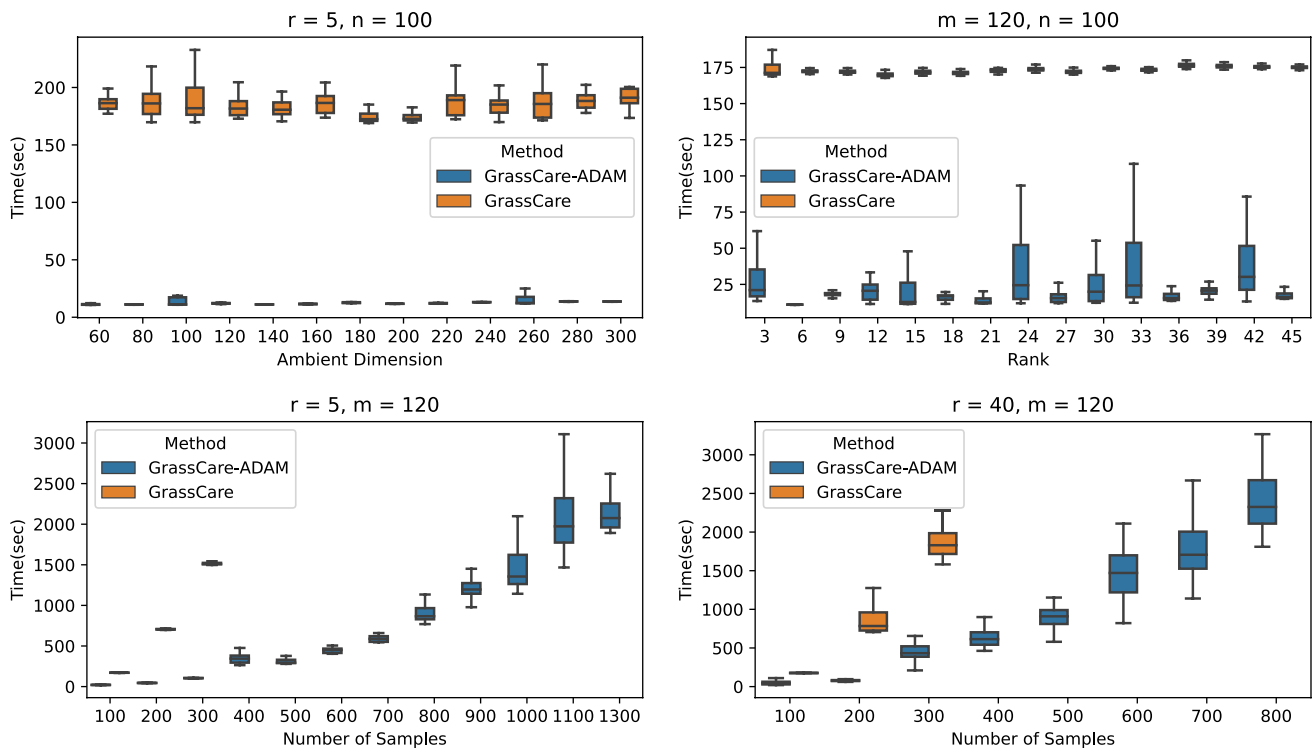


Fig. 3 Time cost of GrassCaré with different setups. In the graph, m represents ambient dimension, n represents the number of samples, and r represents the rank of each sample (subspace). Certain statistics

of the original method are omitted due to failure in generating results within 3600 s

$$\nabla_i \mathcal{L} = \frac{4}{\beta} \sum_j ([\mathbf{P}_G]_{ij} - [\mathbf{P}_D]_{ij}) \left(1 + d_D(\mathbf{p}_i, \mathbf{p}_j)^2\right)^{-1} d_D(\mathbf{p}_i, \mathbf{p}_j) \nabla_i d_D(\mathbf{p}_i, \mathbf{p}_j), \quad (6)$$

where the gradient of d_D w.r.t. $\mathbf{p}_i$ is given by:

$$\nabla_i d_D(\mathbf{p}_i, \mathbf{p}_j) = \frac{4}{b\sqrt{c^2 - 1}} \left(\frac{\|\mathbf{p}_j\|^2 - 2\langle \mathbf{p}_i, \mathbf{p}_j \rangle + 1}{a^2} \mathbf{p}_i - \frac{\mathbf{p}_j}{a} \right).$$

Here $a = 1 - \|\mathbf{p}_i\|^2$, $b = 1 - \|\mathbf{p}_j\|^2$, and $c = 1 + \frac{2}{ab} \|\mathbf{p}_i - \mathbf{p}_j\|^2$. Finally, the retraction step is given by

$$R(\mathbf{p}_i - \eta \nabla_i \mathcal{L}) = \text{proj} \left(\mathbf{p}_i - \eta \frac{(1 - \|\mathbf{p}_i\|^2)^2}{4} \nabla_i \mathcal{L} \right),$$

where

$$\text{proj}(\mathbf{p}_i) = \begin{cases} \mathbf{p}_i / (\|\mathbf{p}_i\| + \varepsilon) & \text{if } \|\mathbf{p}_i\| \geq 1 \\ \mathbf{p}_i & \text{otherwise,} \end{cases}$$

and ε is a small constant number; in our experiments we set this to 10^{-5} .

In our implementation we use random initialization for the points in the embedding. We point out that

initialization is crucial for t-SNE. This is because t-SNE is generally used to embed points in the Euclidean space, which is open. In contrast, the Grassmannian is spherical and compact, and hence, we observed that varying initialization resulted in similar/equivalent embeddings of the Grassmannian, *observed* from different angles. This is further demonstrated in our experiments section (Fig. 6), where the average loss of GrassCaré over 100 trials varies very little in comparison to all other embeddings, showing that besides this point-of-view difference, our results do not depend heavily on the initialization.

Implementation with Adam

Our first prototype of GrassCaré [59] employed regular gradient descent due to its simplicity and smooth convergence. However, we soon discovered that algorithm 1 in [59] became prohibitively slow when dealing with larger datasets. To address this issue and enhance the algorithm's speed, we introduced the ADAM optimizer [60]. By incorporating ADAM into the process, we were able to achieve a significantly faster and more efficient algorithm (orders of magnitude faster, capable of handling datasets orders of magnitude larger; see Fig. 3). The entire embedding

process is detailed in Algorithm 1. It is important to note that whenever $(\cdot)^2$ or $\sqrt{\cdot}$ is applied to a vector, it represents an element-wise operation.

Algorithm 1 GrassCaré using ADAM

Input: A collection of subspaces $\{\mathbb{U}_1, \mathbb{U}_2, \dots, \mathbb{U}_N\} \in \mathbb{G}(m, r)$.

Output: A collection of points $\{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_N\}$ in the Poincaré disk $\mathbb{D}$.

Require: $\beta \in [1, 2]$: Tuning parameter in (4)

Require: $\varepsilon < 10^{-5}$: Boundary tolerance

Require: $\eta \in \mathbb{R}$: Stepsize

Require: $\alpha_1, \alpha_2 \in [0, 1)$: Exponential decay rates for moment estimates

- 1: Construct $\mathbf{P}_{\mathbb{G}}$ according to (3)
- 2: Randomly initialize $\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_N \in \mathbb{D}$
- 3: $\zeta^0 \leftarrow 0$
- 4: $\xi^0 \leftarrow 0$
- 5: **repeat**
- 6: Construct $\mathbf{P}_{\mathbb{D}}$ according to (4)
- 7: Compute KL-loss gradient $\nabla \mathcal{L}$ as in (6)
- 8: Update moment estimates:

$$\zeta^t \leftarrow \alpha_1 \cdot \zeta^{t-1} + (1 - \alpha_1) \cdot \nabla \mathcal{L}, \quad \xi^t \leftarrow \alpha_2 \cdot \xi^{t-1} + (1 - \alpha_2) \cdot (\nabla \mathcal{L})^2$$

- 9: Compute bias-corrected moment estimates:

$$\hat{\zeta}^t \leftarrow \zeta^t / (1 - (\alpha_1)^t), \quad \hat{\xi}^t \leftarrow \xi^t / (1 - (\alpha_2)^t),$$

- 10: Update point $\mathbf{p}_i^{t+1}$ with:

$$\mathbf{p}_i^{t+1} = \text{proj} \left(\mathbf{p}_i^t - \eta \cdot \frac{(1 - \|\mathbf{p}_i^t\|^2)^2 \hat{\zeta}_i}{4 \left(\sqrt{\hat{\xi}_i} + \varepsilon \right)} \right)$$

- 11: **until** Converge
-

We conducted tests using synthetic data to compare the performance of the original and improved methods. The results are summarized in Fig. 3. Our findings indicate that the number of samples remains the primary determinant of computing time. Notably, when using 100 samples, the improved method consistently outperforms the original approach in terms of speed. As the number of samples increases, the original method fails to converge after 300 points within 3600 s, while the improved method effectively solves the problem in a short amount of time.

Main Theoretical Results and Proofs

First observe that convergence of our embedding follows directly by now-standard results in Riemannian optimization (see, e.g. Proposition in [61]). In fact, local

convergence of our embedding follows directly because our Riemannian steps are *gradient-related* [61]. Our main theoretical result goes one step further, bounding the loss

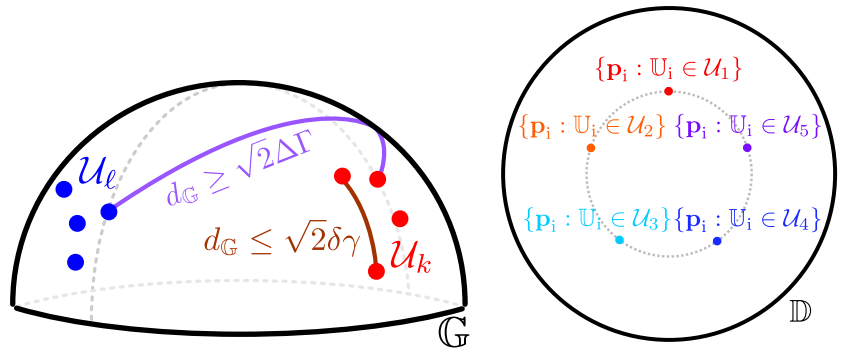
of our embedding by a log-factor of the number of subspaces, and a term that depends on the arrangement of the subspaces in the Grassmannian. This term will be smaller if the subspaces form well-defined clusters, and larger if the subspaces have no structure whatsoever. Intuitively, this result shows that under reasonable assumptions, our embedding can provide an accurate representation of Grassmannians.

Theorem 1 Suppose $N > 3$. Define $\gamma := \min_i \gamma_i$ and $\Gamma := \max_i \gamma_i$. Let $\{\mathcal{U}_1, \dots, \mathcal{U}_K\}$ be a partition of $\{\mathbb{U}_1, \dots, \mathbb{U}_N\}$ such that $|\mathcal{U}_k| \geq n_K > 1 \forall k$. Let

$$\delta := \frac{1}{\sqrt{2\gamma}} \max_k \max_{\mathbb{U}_i, \mathbb{U}_j \in \mathcal{U}_k} d_{\mathbb{G}}(\mathbb{U}_i, \mathbb{U}_j),$$

$$\Delta := \frac{1}{\sqrt{2\Gamma}} \min_{\mathbb{U}_i \in \mathcal{U}_k, \mathbb{U}_j \in \mathcal{U}_{\ell}: k \neq \ell} d_{\mathbb{G}}(\mathbb{U}_i, \mathbb{U}_j).$$

Fig. 4 *Left:* Theorem 1 requires that the intra-cluster distances are smaller than $\sqrt{2}\delta\gamma$, and the outer-cluster distances are larger than $\sqrt{2}\Delta\Gamma$. *Right:* Example of the artificial embedding (with $K = 5$) in the proof of Theorem 1, which maps all subspaces in cluster $\mathcal{U}_k$ to the same point in the circle of radius $1/2$



Then the optimal loss of GrassCaré is bounded by:

$$\mathcal{L}^* < \log D + \frac{5e^{\delta^2 - \Delta^2}}{\beta(n_K - 1)},$$

where

$$D := N(n_K - 1) + N(N - n_K) \cdot \exp\left(-\operatorname{arcosh}^2\left(1 + \frac{2\sin(\pi/K)}{0.75^2}\right)/\beta\right). \quad (7)$$

In words, Theorem 1 requires that the subspaces can be arranged into clusters of size $n_k > 1$ such that the intra-cluster distances are smaller than $\sqrt{2}\delta\gamma$, and the outer-cluster distances are larger than $\sqrt{2}\Delta\Gamma$ (see Fig. 4). Notice that this can always be done as long as $N > 3$. However, depending on the arrangement, δ could be too large or Δ too small, resulting in a loose bound. Ideally we want a small δ and a large Δ , so that the subspaces form well-defined clusters and $e^{\delta^2 - \Delta^2}$ is small, resulting in a tighter bound.

Proof Theorem 1 follows by a similar strategy as in [62], which essentially bounds the optimal loss by that of an artificial embedding. In our case we will use an embedding that maps $\{\mathcal{U}_1, \dots, \mathcal{U}_K\}$ to K points uniformly distributed in the circle of radius $1/2$, i.e., $\mathbf{p}_i = \mathbf{p}_j$ for every $\mathbb{U}_i, \mathbb{U}_j \in \mathcal{U}_k$ (see Fig. 4). This way, for any subspaces $\mathbb{U}_i, \mathbb{U}_j$ in different clusters $\mathcal{U}_k, \mathcal{U}_\ell$, the geodesic distance of their embeddings on the Poincaré disk is upper and lower bounded by

$$\begin{aligned} 2.2 &> \left(1 + \frac{2}{0.75^2}\right) \geq d_{\mathbb{D}}(\mathbf{p}_i, \mathbf{p}_j) \\ &\geq \operatorname{arcosh}\left(1 + \frac{2\sin(\pi/K)}{0.75^2}\right) =: \Phi. \end{aligned}$$

It follows that the (i, j) th entry of $\mathbf{P}_{\mathbb{D}}$ is bounded by

$$\begin{aligned} [\mathbf{P}_{\mathbb{D}}]_{ij} &:= \frac{\exp(-d_{\mathbb{D}}(\mathbf{p}_i, \mathbf{p}_j)^2/\beta)}{\sum_{k \neq l} \exp(-d_{\mathbb{D}}(\mathbf{p}_k, \mathbf{p}_l)^2/\beta)} \\ &\geq \frac{\exp(-d_{\mathbb{D}}(\mathbf{p}_i, \mathbf{p}_j)^2/\beta)}{N(n_K - 1) + N(N - n_K) \exp(-\Phi^2/\beta)}, \end{aligned} \quad (8)$$

where the denominator is precisely D as defined in (7). Now, if $\mathbb{U}_i$ and $\mathbb{U}_j$ are in the same cluster $\mathcal{U}_k$, the bound in (8) simplifies to $1/D$. Otherwise, it simplifies to $\exp(-2.2^2/\beta)/D$. Plugging these bounds in the loss, we see that:

$$\begin{aligned} \mathcal{L}^* &< \sum_{i,j \text{ in same cluster}} [\mathbf{P}_{\mathbb{D}}]_{ij} \log D \\ &\quad + \sum_{i,j \text{ in different clusters}} [\mathbf{P}_{\mathbb{D}}]_{ij} (2.2^2/\beta + \log D) \\ &\leq \log D + \sum_{i,j \text{ in different clusters}} [\mathbf{P}_{\mathbb{D}}]_{ij} (2.2^2/\beta). \end{aligned} \quad (9)$$

Next notice that if $\mathbb{U}_i, \mathbb{U}_j$ are not in the same cluster,

$$\begin{aligned} [\mathbf{P}_{\mathbb{D}}]_{ij} &:= \frac{1}{2N} \frac{\exp(-d_{\mathbb{G}}(\mathbb{U}_i, \mathbb{U}_j)^2/2\gamma_i^2)}{\sum_{k \neq i} \exp(-d_{\mathbb{G}}(\mathbb{U}_i, \mathbb{U}_k)^2/2\gamma_i^2)} \\ &\quad + \frac{1}{2N} \frac{\exp(-d_{\mathbb{G}}(\mathbb{U}_j, \mathbb{U}_i)^2/2\gamma_j^2)}{\sum_{k \neq j} \exp(-d_{\mathbb{G}}(\mathbb{U}_j, \mathbb{U}_k)^2/2\gamma_j^2)} \\ &\leq \frac{1}{2N} \frac{e^{-\Delta^2}}{\sum_{k \neq i} \exp(-d_{\mathbb{G}}(\mathbb{U}_i, \mathbb{U}_k)^2/2\gamma_i^2)} \\ &\quad + \frac{1}{2N} \frac{e^{-\Delta^2}}{\sum_{k \neq j} \exp(-d_{\mathbb{G}}(\mathbb{U}_j, \mathbb{U}_k)^2/2\gamma_j^2)} \\ &\leq \frac{1}{N(n_K - 1)e^{-\delta^2}}. \end{aligned}$$

Plugging this into (9) we obtain the desired result. $\square$

Experiments

Recall that the main motivation of this paper is to develop a novel method to visualize Grassmannians of high *ambient* dimension. Our bound above describes the theoretical accuracy of our embedding. We now present a series of experiments on real and synthetic datasets to analyze its practical performance. In particular, we will test on normal simulated data, and one canonical dataset [63]. These datasets have moderately high ambient dimension (i.e., many features), but low intrinsic dimension (i.e., lie in a low-dimensional subspace). In other words, these datasets would fit in high-dimensional Grassmannians of low-dimensional subspaces. We believe that these well-studied datasets are a perfect fit for our setting, and convenient for an initial exploration and comparison against existing baselines.

Comparison Baseline. To evaluate the effectiveness of our method we used t-SNE [54], GDMaps [51], and a naive visualization based on the most common dimensionality reduction technique: Principal Components Analysis (PCA). To this end we first *vectorize* (stack the columns of) each orthonormal basis $\mathbf{U}_i$ into a vector $\mathbf{u}_i \in \mathbb{R}^{mr}$. Next we concatenate all vectors $\mathbf{u}_i$ into a matrix of size $mr \times N$, on which we apply PCA. In this *naive PCA* (nPCA) visualization, the subspace $\mathbb{U}_i$ is represented in the (x, y) plane by $\mathbf{v}_i \in \mathbb{R}^2$, the coefficients of $\mathbf{u}_i$ w.r.t. the leading principal plane $\mathbb{V}$.

Visualizing Synthetic Subspace Clusters. In our synthetic experiments we study our embedding when the subspaces are uniformly distributed among K clusters. To this end we first generated K centers in the Grassmannian $\mathbb{G}(m, r)$, each defined by a $m \times r$ matrix $\mathbf{C}_k$ with i.i.e. standard normal entries whose columns are later orthonormalized. Then for each k we independently generate n_k subspaces, each spanned by a basis $\mathbf{U}_i$ whose entries are equal to those of $\mathbf{C}_k$ plus i.i.d. normal random variables with variance σ^2 . This will produce K clusters in $\mathbb{G}(m, r)$, each with n_k subspaces. The smaller σ , the closer the subspaces in the same cluster will be to one another, and vice versa.

In our first experiment we study a controlled setting where we can actually visualize the low-dimensional Grassmannian $\mathbb{G}(3, 1)$, and compare it with our embedding on the Poincaré disk. We hope that this experiment provides a visual intuition of how points are embedded in higher dimensional cases. To this end we generated $n_k = 50$ subspaces per cluster ($m = 3, r = 1$), and we set $\sigma = 0.1$, which produced visually well-defined cluster clouds. Figure 5 shows the results for $K = 2, 3, 4, 5$ clusters. At first glance it might appear like our GrassCaré embedding is not too different from the other approaches, especially as t-SNE and nPCA seems to be doing a decent job displaying the clusters. However, a more careful look

reveals that t-SNE clearly agglomerates several pairs of clusters, while the GrassCaré can separate them nicely.

In particular, notice that in Fig. 5, both nPCA and even the classic 3D representation fail to show the true local structure of the Grassmannian that the GrassCaré plot reveals. To see this pay special attention to the cyan and yellow clusters. Based on the first two rows (classic 3D representation and nPCA) these clusters would *appear* to be nearly diametrically apart (in the 3D representation, the cyan cluster is in the back side of the hemisphere). However, computing their geodesics one can verify that the subspaces that they represent are in fact quite close in the Grassmannian. An intuitive way to see this is to extend the lines to the opposite side of the hemisphere and compute their smallest angle, or to remember that in the 3D representation, the hemisphere *wraps* around the edge (see Fig. 4). In contrast, our GrassCaré plot accurately displays the true global structure of the Grassmannian, mapping these two clusters close to one another. Also notice that the embeddings are plotted with equal scale on horizontal and vertical axis. The GrassCaré makes a better use of the visual space, spreading all data more broadly while at the same time keeping the clusters well-defined. In contrast, GDMaps has much less range on the horizontal axis, which makes it look like a straight line and not be able to display the full information.

The previous experiment shows the qualitative superiority of the GrassCaré embedding over alternative embeddings in the low-dimensional case $m = 3$ and $r = 1$ (where no vectorization is needed for nPCA). In our next experiment we will show in a more quantitative way that the advantages of GrassCaré are even more evident in higher dimensional cases. First notice that the classic 3D representation only applies to the case $m = 3, r = 1$, and there is no clear way how to extend it to higher dimensions. On the other hand, recall that for $r > 1$, nPCA requires vectorizing the bases $\mathbf{U}_i$, which will naturally interfere even more with the structure of the Grassmannian. To see this consider:

$$\mathbf{U} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{U}' = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 0 \end{bmatrix}.$$

While both span the same subspace in $\mathbb{G}(3, 2)$, the Euclidean distance of their vectorizations is large, which would result in distant points in the nPCA embedding. t-SNE and GDMaps present similar inaccuracy behavior. To verify this we generated subspaces in the exact same way as described before (with $K = 3, n_k = 17$, and different values of m and r), except this time we measured the quality of the visualization in terms of the *representation error*, which we define as

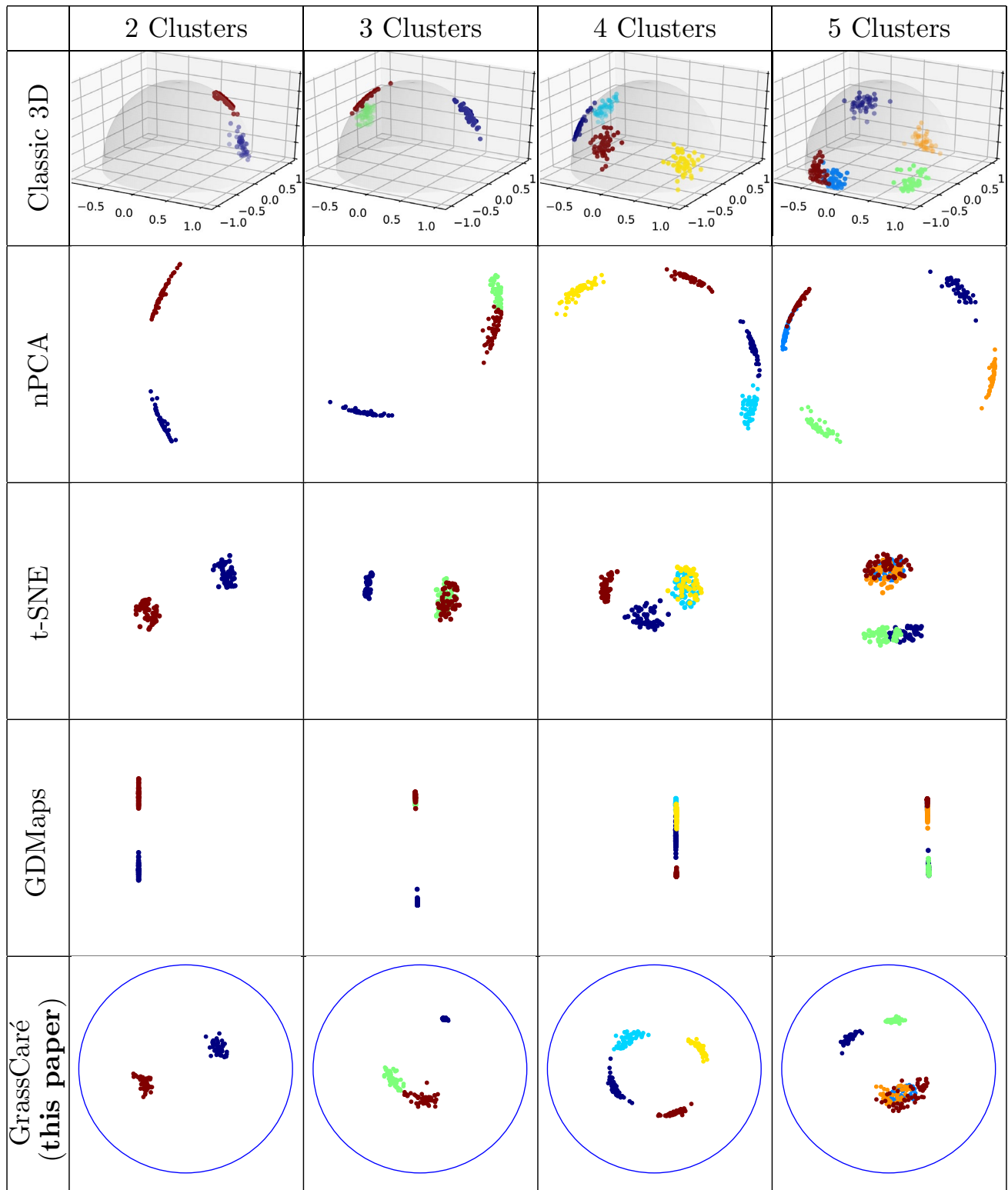


Fig. 5 Alternative visualizations of clusters in $\mathbb{G}(3, 1)$. GrassCaré produces a more accurate representation of the Grassmannian, e.g., the case of $K = 4$ clusters, where nPCA and even the 3D representation

display Clusters 1 and 2 (cyan and yellow) nearly diametrically apart. In reality they are quite close, as depicted by GrassCaré. See discussion for details

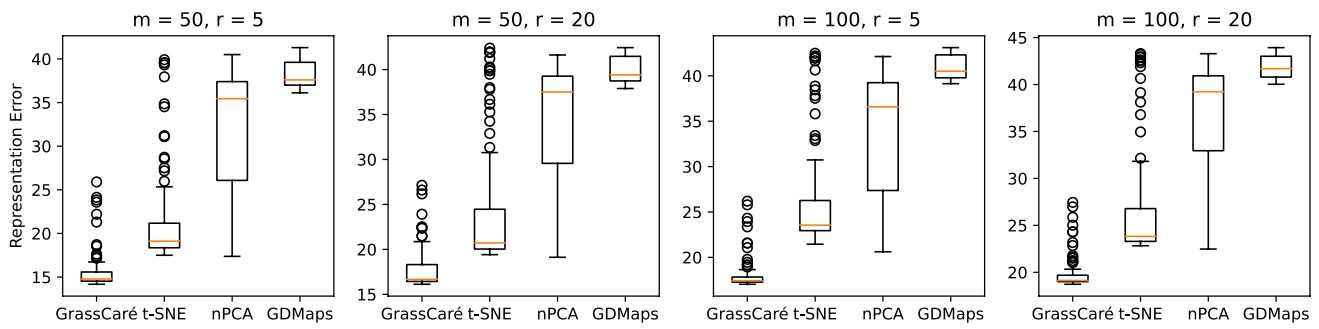


Fig. 6 Representation error of GrassCaré (this paper) and other methods for high-dimensional Grassmannians $\mathbb{G}(m, r)$

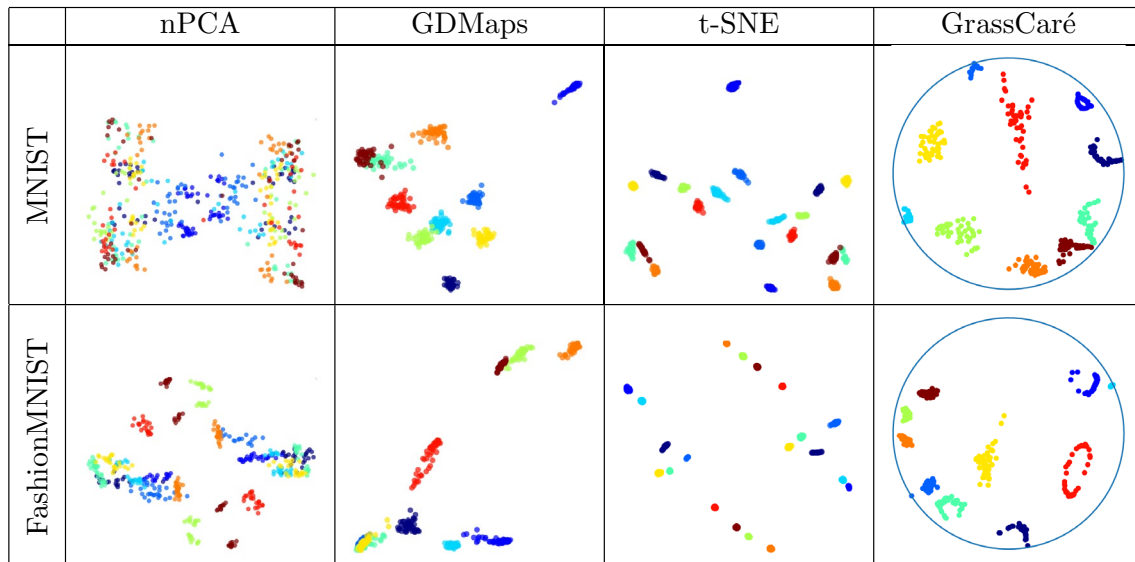


Fig. 7 Subspace visualization using nPCA, GDMaps, t-SNE, and GrassCaré (this paper)

the Frobenius difference between the (normalized) distance matrices produced by the subspaces in the Grassmannian and the points in each embedding. In the case of GrassCaré, distances in the embedding are measured according to the Poincaré geodesics, so the representation error of the GrassCaré embedding will be measured as

$$\epsilon^2(\mathbb{D}) = \sum_{i,j} \left(\frac{d_{\mathbb{G}}(\mathbb{U}_i, \mathbb{U}_j)}{Z_{\mathbb{G}}} - \frac{d_{\mathbb{D}}(\mathbf{p}_i, \mathbf{p}_j)}{Z_{\mathbb{D}}} \right)^2, \quad (10)$$

where $Z_{\mathbb{G}}^2 = \sum_{i,j} d_{\mathbb{G}}^2(\mathbb{U}_i, \mathbb{U}_j)$ and $Z_{\mathbb{D}}^2 = \sum_{i,j} d_{\mathbb{D}}^2(\mathbf{p}_i, \mathbf{p}_j)$ are normalization terms. Similarly, since the distance that all other embeddings aim to minimize is Euclidean, the representation error of the other embeddings will be measured as

$$\epsilon^2(\mathbb{V}) = \sum_{i,j} \left(\frac{d_{\mathbb{G}}(\mathbb{U}_i, \mathbb{U}_j)}{Z_{\mathbb{G}}} - \frac{\|\mathbf{v}_i - \mathbf{v}_j\|}{Z_{\mathbb{V}}} \right)^2, \quad (11)$$

where $Z_{\mathbb{V}}^2 = \sum_{i,j} \|\mathbf{v}_i - \mathbf{v}_j\|^2$ is a normalization term. The results of 100 trials are summarized in Fig. 6, which confirms the superiority of our GrassCaré embedding, and the loss of structure of the naive approach. The computation time is also summarized in Fig. 3.

Visualizing Subspace Clusters from Real Data. In this experiment, we aim to visualize the relationship between subspaces formed by groups of images in the MNIST and FashionMNIST datasets. To achieve this, we first grouped every 70 pictures randomly within each category. Next, we applied Singular Value Decomposition to extract the basis for these samples. From the decomposition, we selected the top 2–3 ranks to serve as the general basis for each subclass. Subsequently, we utilized nPCA, GDMaps, t-SNE, and GrassCaré (this paper) on the selected basis to further analyze the data.

In Fig. 7, we present the visualization results obtained through this comprehensive process. An in-depth analysis of these results reveals that GrassCaré exhibits a remarkable

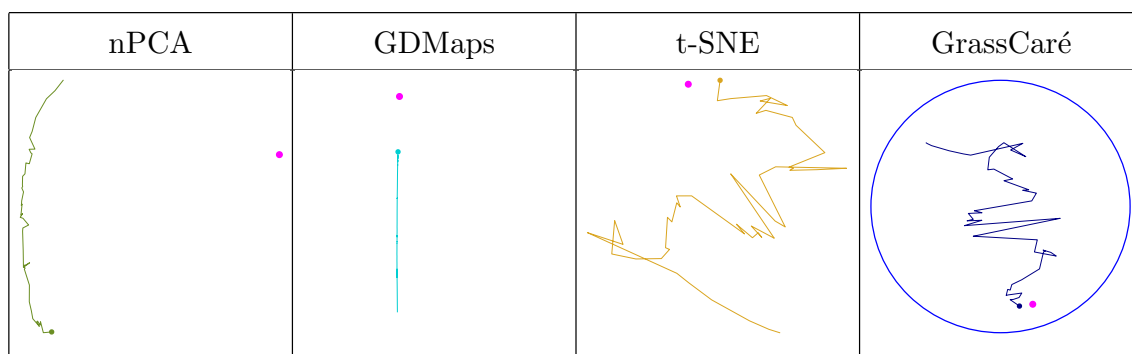


Fig. 8 Visualization of the path generated by GROUSE using nPCA, GDMaps, t-SNE, and GrassCaré (this paper)

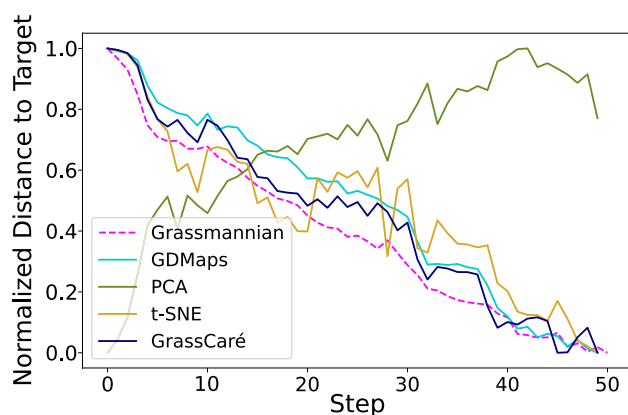


Fig. 9 Distance to target (in the Grassmannian and its embeddings) of the sequence generated by GROUSE

ability to visualize low-rank subspaces effectively. Notably, this method excels in distinctly presenting each subclass as a distinguishable cluster, capturing the intrinsic structure of the data. In contrast, other commonly used visualization techniques, such as GDMaps, show limitations in certain scenarios. Specifically, GDMaps may struggle to fully separate certain classes, resulting in overlapping or less well-defined clusters. Similarly, t-SNE might encounter challenges in correctly grouping certain subclasses together, potentially leading to less cohesive and less interpretable visualizations.

It is important to acknowledge that applying these methods directly to the raw dataset might yield different outcomes and potentially showcase superior results for certain techniques. However, comparing the results from raw data with those from the subclasses derived using GrassCaré would be akin to comparing oranges with apples, rendering such a comparison meaningless.

Subspace Estimation from Incomplete Data. In our next experiment we apply our GrassCaré embedding to visualize the path of subspaces produced by the subspace estimation algorithm known as GROUSE (Grassmannian

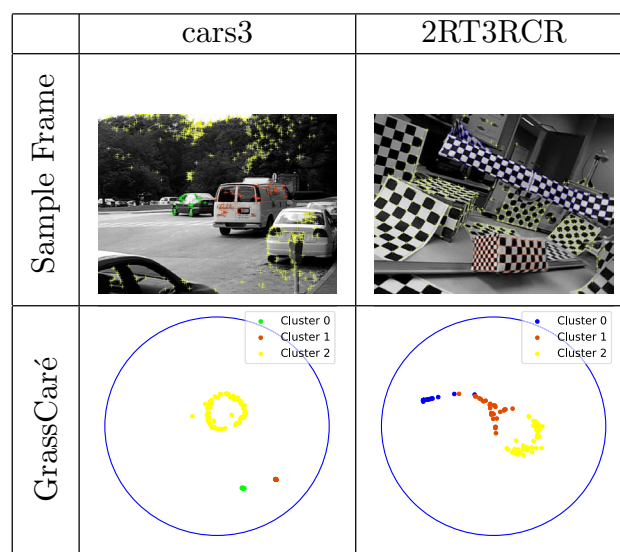


Fig. 10 GrassCaré embedding for two motion sequences of the Hopkins155 dataset

Rank-One Update Subspace Estimation) [32]. The applicability of this algorithm ranges from online video analysis (to track the subspace of the background in real time) to subspace clustering [18] and low-rank matrix completion (LRMC) [32]. In the latter, the algorithm receives a subset of the entries of a data matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$ whose columns lie in an unknown subspace $\mathbb{U}^* \in \mathbb{G}(m, r)$, and the goal is to estimate $\mathbb{U}^*$. To this end GROUSE starts with a subspace estimate $\mathbb{U}_0 \in \mathbb{G}(m, r)$, and iteratively *tilts* it in the direction of a column of $\mathbf{X}$, producing a sequence of subspaces $\mathbb{U}_1, \dots, \mathbb{U}_N$.

To emulate this setup we first generate *true* and *initial* subspaces with bases $\mathbf{U}^*, \mathbf{U}_0 \in \mathbb{R}^{m \times r}$ with i.i.d. standard normal entries. Next we generate a coefficient matrix $\mathbf{\Theta} \in \mathbb{R}^{r \times n}$ with i.i.d. standard normal entries, so that $\mathbf{X} = \mathbf{U}^* \mathbf{\Theta}$ is rank- r . Then we run GROUSE using a fraction Ω of the entries of $\mathbf{X}$, selected uniformly at random. We store each of GROUSE's steps, and visualize their path $\mathbb{U}_0, \mathbb{U}_1, \dots, \mathbb{U}_N$

(together with the target $\mathbb{U}^*$) using our GrassCaré embedding, nPCA, GDMaps, and t-SNE. Figure 8 shows sample plots when $m = 200$, $r = 5$, $\Omega = 0.7$ (corresponding to 30% missing data; both cases share the same initialization), and $n = N = 50$ (corresponding to the case where GROUSE only iterates once over each column). Here once again the GrassCaré plot shows a richer depiction of the subspaces and a better usage of the available visual space. From the GrassCaré plot we can clearly visualize each separate path, and see that, as expected, the full-data estimate gets much closer to the target than the missing-data estimate (and much faster). In contrast, the paths in the nPCA plot are hardly distinguishable and misleading, showing the opposite of the truth: an incomplete-data estimate much closer to the target than the full-data estimate. To verify once again (beyond our visual interpretation) that the GrassCaré embedding is much more representative of the true distribution of subspaces in the Grassmannian, we measured the distance of each iterate to the target, in the Grassmannian and in each embedding. The normalized results are in Fig. 9. They show that the trajectories in the GrassCaré embedding mimic closely those in the Grassmannian. In contrast, the nPCA embedding can be quite misleading, showing in fact an opposite representation, with distances in the embedding growing over iterations, while in reality they are decreasing in the Grassmannian.

Motion Segmentation. In our final experiment we use our GrassCaré plot to visualize the subspaces describing moving objects in videos from the Hopkins 155 dataset [63]. This dataset contains the locations over time of landmarks of several moving objects (e.g., cars, buses, or checkerboards) in 155 video sequences. Recall that the stacked landmarks of each rigid object over time approximately lie in a four-dimensional subspace [64, 65]. So for our experiment we split all landmarks of the same object in groups of 5 (if at any point there were fewer than 5 landmarks left, they were discarded), and for each group we performed a singular value decomposition to identify its four-dimensional principal subspace $\mathbb{U}_i$. Figure 10 shows the embedding of the subspaces of all groups, color-coded by object. Notice that GrassCaré displays the subspaces of the same object nearby. This is consistent with theory, as they represent slightly noisy versions of the subspace describing the object's trajectory. Notice the higher variance in the yellow cluster, which is consistent with its landmarks, corresponding to several trees, cars, and pavement, as opposed to just one rigid object. But not only that. From the GrassCaré embedding we can also analyze the trajectories themselves, and their relationships. For instance, in the traffic plot we can see that the green and red clusters (corresponding to the moving car and van) are close to one another, indicating that their trajectories resemble each other. In contrast, these clusters are farther

from the yellow one, which matches our observation that the trajectories of the moving car and van are quite different from the nearly static background.

Conclusions and Limitations

This paper introduces GrassCaré, a novel method for embedding Grassmannian points onto a 2-D disk with exceptional precision and a well-distributed visual representation. GrassCaré promises to be a potent tool for visualizing subspaces extracted from high-dimensional real-world data, enabling researchers to explore both local and global structures, such as paths and clusters, with greater ease and clarity.

Through the incorporation of the Adam optimizer in our enhanced prototype, we have achieved remarkable advancements over the original algorithm, resulting in significant speed and efficiency improvements. GrassCaré now operates at orders of magnitude faster speeds, empowering it to handle datasets of unprecedented scale. Furthermore, our main theoretical finding demonstrates that GrassCaré boasts a lower bound on representation loss under mild assumptions, further reinforcing the reliability and robustness of the method. Finally, we provide compelling evidence of GrassCaré's superiority by applying it to real-world datasets. The results not only showcase its ability to optimize space utilization within the unit circle but also effectively mitigate the visual distortion arising from disparate axis scales, unlike the shortcomings observed in GDMaps.

In conclusion, GrassCaré emerges as a promising and indispensable tool for data visualization, empowering researchers to glean meaningful insights from complex high-dimensional data while preserving fidelity and clarity in the representation. As we continue to explore and refine this method, we anticipate its widespread adoption in various domains, making a lasting impact on the field of data analysis and visualization.

Acknowledgements This work was partially supported by the National Science Foundation under Grant No. CAREER CCF-2239479.

Data availability The dataset of MNIST is available at <https://doi.org/10.24432/C53K8Q>. The dataset of FashionMNIST is available at <https://github.com/zalandoresearch/fashion-mnist>. The dataset of Hopkins155 is available at <http://www.vision.jhu.edu/data/hopkins155>

Code Availability The code for this project is accessible through a public Github repository, which can be accessed at the following link: <https://github.com/HuanranLi/GrassCaré-Plot>.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Knudsen T. Consistency analysis of subspace identification methods based on a linear regression approach. *Automatica*. 2001;37(1):81–9.
- Jansson M, Wahlberg B. A linear regression approach to state-space subspace system identification. *Signal Process*. 1996;52(2):103–29.
- Vaswani N, Bouwmans T, Javed S, Narayanamurthy P. Robust subspace learning: robust PCA, robust subspace tracking, and robust subspace recovery. *IEEE Signal Process Mag*. 2018;35(4):32–55.
- Dai W, Milenkovic O, Kerman E. Subspace evolution and transfer (set) for low-rank matrix completion. *IEEE Trans Signal Process*. 2011;59(7):3120–32.
- Vidal R, Favaro P. Low rank subspace clustering (LRSC). *Pattern Recogn Lett*. 2014;43:47–61.
- Cao J, Zhang K, Luo M, Yin C, Lai X. Extreme learning machine and adaptive sparse representation for image classification. *Neural Netw*. 2016;81:91–102.
- Chen G, Lerman G. Spectral curvature clustering (SCC). *Int J Comput Vis*. 2009;81(3):317–30.
- Hong W, Wright J, Huang K, Ma Y. Multiscale hybrid linear models for lossy image representation. *IEEE Trans Image Process*. 2006;15(12):3655–71.
- Lu L, Vidal R. Combined central and subspace clustering for computer vision applications; 2006. p. 593–600.
- Koohi H, Kiani K. A new method to find neighbor users that improves the performance of collaborative filtering. *Expert Syst Appl*. 2017;83:30–9.
- Ullah F, Sarwar G, Lee S. N-screen aware multicriteria hybrid recommender system using weight based subspace clustering. *Sci World J* 2014;2014:1–10.
- Zhang W, Wang Q, Yoshida T, Li J. RP-LGMC: rating prediction based on local and global information with matrix clustering. *Comput Oper Res*. 2021;129:105228.
- Sun W, Zhang L, Du B, Li W, Lai YM. Band selection using improved sparse subspace clustering for hyperspectral imagery classification. *IEEE J Sel Top Appl Earth Observ Remote Sens*. 2015;8(6):2784–97.
- Ahmed MS, Khan L. SISC: a text classification approach using semi supervised subspace clustering. In: 2009 IEEE international conference on data mining workshops. IEEE; 2009. p. 1–6.
- Xia C-Q, Han K, Qi Y, Zhang Y, Yu D-J. A self-training subspace clustering algorithm under low-rank representation for cancer classification on gene expression data. *IEEE/ACM Trans Comput Biol Bioinform*. 2017;15(4):1315–24.
- Mevel L, Hermans L, Auweraer H. Application of a subspace-based fault detection method to industrial structures. *Mech Syst Signal Process*. 1999;13(6):823–38.
- Goodman-Strauss C. Compass and straightedge in the poincaré disk. *Am Math Monthly*. 2001;108(1):38–49.
- Parsons L, Haque E, Liu H. Subspace clustering for high dimensional data: a review. *ACM SIGKDD Explor Newsl*. 2004;6(1):90–105.
- Yang AY, Wright J, Ma Y, Sastry SS. Unsupervised segmentation of natural images via lossy data compression. *Comput Vis Image Understand*. 2008;110(2):212–25.
- Vidal R, Tron R, Hartley R. Multiframe motion segmentation with missing data using powerfactorization and GPCA. *Int J Comput Vis*. 2008;79(1):85–105.
- Elhamifar E, Vidal R. Sparse subspace clustering: algorithm, theory, and applications. *IEEE Trans Pattern Anal Mach Intell*. 2013;35(11):2765–81.
- Agrawal R, Gehrke J, Gunopulos D, Raghavan P. Automatic subspace clustering of high dimensional data for data mining applications. In: Proceedings of the 1998 ACM SIGMOD international conference on management of data; 1998. p. 94–105.
- Bahadori MT, Kale D, Fan Y, Liu Y. Functional subspace clustering with application to time series. In: International conference on machine learning. PMLR; 2015. p. 228–237.
- Recht B. A simpler approach to matrix completion. *J Mach Learn Res*. 2011;12(12):3413–30.
- Kang Z, Peng C, Cheng Q. Top-n recommender system via matrix completion. In: Proceedings of the AAAI conference on artificial intelligence, vol. 30; 2016.
- Ji H, Liu C, Shen Z, Xu Y. Robust video denoising using low rank matrix completion. In: 2010 IEEE computer society conference on computer vision and pattern recognition. IEEE; 2010. p. 1791–8.
- Zhang H, Ericksen SS, Lee C-P, Ananiev GE, Wlodarchak N, Yu P, Mitchell JC, Gitter A, Wright SJ, Hoffmann FM, et al. Predicting kinase inhibitors using bioactivity matrix derived informer sets. *PLoS Comput Biol*. 2019;15(8):1006813.
- Lee D, Kim BH, Kim KJ. Detecting method on illegal use using PCA under her environment. In: 2010 International Conference on Information Science and Applications. IEEE; 2010. p. 1–6.
- He J, Balzano L, Lui J. Online robust subspace tracking from partial information. *arXiv preprint arXiv:1109.3827*; 2011.
- Xu J, Ithapu VK, Mukherjee L, Rehg JM, Singh V. GOSUS: Grassmannian online subspace updates with structured-sparsity. In: Proceedings of the IEEE international conference on computer vision; 2013. p. 3376–3383.
- Stewart GW. An updating algorithm for subspace tracking. Technical report; 1998.
- Balzano L, Nowak R, Recht B. Online identification and tracking of subspaces from highly incomplete information. In: 2010 48th Annual Allerton conference on communication, control, and computing (Allerton). IEEE; 2010. p. 704–711.
- Novembre J, Johnson T, Bryck K, Kutalik Z, Boyko AR, Auton A, Indap A, King KS, Bergmann S, Nelson MR, et al. Genes mirror geography within Europe. *Nature*. 2008;456(7218):98–101.
- Song Y, Westerhuis JA, Aben N, Michaut M, Wessels LF, Smilde AK. Principal component analysis of binary genomics data. *Brief Bioinform*. 2019;20(1):317–29.
- Wu J, Zhang X. A PCA classifier and its application in vehicle detection. In: IJCNN'01. International Joint Conference on Neural Networks. Proceedings (Cat. No. 01CH37222), vol. 1. IEEE; 2001. p. 600–604.
- Wu J, Zhang X, Zhou J. Vehicle detection in static road images with PCA-and-wavelet-based classifier. In: ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No. 01TH8585). IEEE; 2001. p. 740–744.

37. Kirby M, Peterson C. Visualizing data sets on the Grassmannian using self-organizing mappings. In: 2017 12th International Workshop on Self-organizing Maps and Learning Vector Quantization, Clustering and Data Visualization (WSOM). IEEE; 2017. p. 1–6.
38. Kohonen T. Self-organized formation of topologically correct feature maps. *Biol Cybern*. 1982;43(1):59–69.
39. Kohonen T. The self-organizing map. *Proc IEEE*. 1990;78(9):1464–80.
40. Kohonen T. The self-organizing map. *Neurocomputing*. 1998;21(1–3):1–6.
41. Kohonen T. Essentials of the self-organizing map. *Neural Netw*. 2013;37:52–65.
42. McInnes L, Healy J, Melville J. UMAP: uniform manifold approximation and projection for dimension reduction. *arXiv preprint [arXiv:1802.03426](https://arxiv.org/abs/1802.03426)*; 2018.
43. Tang J, Liu J, Zhang M, Mei Q. Visualizing large-scale and high-dimensional data. In: Proceedings of the 25th international conference on world wide web; 2016. p. 287–297.
44. Belkin M, Niyogi P. Laplacian eigenmaps and spectral techniques for embedding and clustering. *Adv Neural Inf Process Syst*. 2001;14:1–7.
45. Belkin M, Niyogi P. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput*. 2003;15(6):1373–96.
46. Tenenbaum JB, Silva V, Langford JC. A global geometric framework for nonlinear dimensionality reduction. *Science*. 2000;290(5500):2319–23.
47. Liu S, Maljovec D, Wang B, Bremer P-T, Pascucci V. Visualizing high-dimensional data: advances in the past decade. *IEEE Trans Vis Comput Graph*. 2016;23(3):1249–68.
48. Engel D, Hüttenberger L, Hamann B. A survey of dimension reduction methods for high-dimensional data analysis and visualization. In: Visualization of large and unstructured data sets: applications in geospatial planning, modeling and engineering-proceedings of IRTG 1131 Workshop 2011. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik; 2012.
49. Ashokkumar P, Don S. High dimensional data visualization: a survey. *J Adv Res Dyn Control Syst*. 2017;9(12):851–66.
50. Kiefer A, Rahman M, et al.: An analytical survey on recent trends in high dimensional data visualization. *arXiv preprint [arXiv:2107.01887](https://arxiv.org/abs/2107.01887)*; 2021.
51. Santos KR, Giovanis DG, Shields MD. Grassmannian diffusion maps based dimension reduction and classification for high-dimensional data. *arXiv preprint [arXiv:2009.07547](https://arxiv.org/abs/2009.07547)*; 2020.
52. Coifman RR, Lafon S, Lee AB, Maggioni M, Nadler B, Warner F, Zucker SW. Geometric diffusions as a tool for harmonic analysis and structure definition of data: diffusion maps. *Proc Natl Acad Sci*. 2005;102(21):7426–31.
53. Hinton G, Roweis ST. Stochastic neighbor embedding. In: NIPS, vol. 15. Citeseer; 2002. p. 833–840.
54. Maaten L, Hinton G. Visualizing data using t-SNE. *J Mach Learn Res*. 2008;9(11):1–2.
55. Nickel M, Kiela D. Poincaré embeddings for learning hierarchical representations. *Adv Neural Inf Process Syst*. 2017;30:6338–47.
56. Klimovskaia A, Lopez-Paz D, Bottou L, Nickel M. Poincaré maps for analyzing complex hierarchies in single-cell data. *Nat Commun*. 2020;11(1):1–9.
57. Bonnabel S. Stochastic gradient descent on Riemannian manifolds. *IEEE Trans Autom Control*. 2013;58(9):2217–29.
58. Absil P-A, Mahony R, Sepulchre R. Optimization algorithms on matrix manifolds. Princeton: Princeton University Press; 2009.
59. Li H, Pimentel-Alarcón D. Visualizing Grassmannians via poincaré embeddings; 2023.
60. Kingma DP, Ba J. Adam: a method for stochastic optimization. *arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980)*; 2014.
61. Adams LM, Nazareth JL, et al. Linear and nonlinear conjugate gradient-related methods, vol. 85. Philadelphia: SIAM; 1996.
62. Shaham U, Steinerberger S. Stochastic neighbor embedding separates well-separated clusters. *arXiv preprint [arXiv:1702.02670](https://arxiv.org/abs/1702.02670)*; 2017.
63. Tron R, Vidal RA, benchmark for the comparison of 3-d motion segmentation algorithms. In: 2007 IEEE conference on computer vision and pattern recognition. IEEE; 2007 p. 1–8.
64. Tomasi C, Kanade T. Shape and motion from image streams under orthography: a factorization method. *Int J Comput Vis*. 1992;9(2):137–54.
65. Kanatani K-I. Motion segmentation by subspace separation and model selection. In: Proceedings eighth IEEE international conference on computer vision. ICCV 2001, vol. 2. IEEE; 2001. p. 586–91.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.