MLVis: Machine Learning Methods in Visualisation for Big Data (2024)
Daniel Archambault, Ian Nabney, and Jaakko Peltonen (Editors)

Visualizing Riemannian data with Rie-SNE

A. Bergsson[1] and S. Hauberg[2]

[1]Marel, Reykjavík, Iceland
[2]Technical University of Denmark, Kgs. Lyngby, Denmark

Abstract

Faithful visualizations of data residing on manifolds must take the underlying geometry into account when producing a flat planar view of the data. In this paper, we extend the stochastic neighbor embedding (SNE) *algorithm to data on general Riemannian manifolds. We replace standard Gaussian assumptions with Riemannian diffusion counterparts and propose an efficient approximation that only requires access to calculations of Riemannian distances and volumes. We demonstrate that the approach also allows for mapping data from one manifold to another, e.g. from a high-dimensional sphere to a low-dimensional one.*

1. Introduction

Visualizations are crucial to investigators trying to make sense of high-dimensional data. The most common visualization is a two-dimensional plot (e.g. on a piece of paper or a computer screen), so we rely on dimensionality reduction when handling high-dimensional data. Most dimensionality reduction techniques assume that data resides in an Euclidean domain, which presents a problem when data is not quite that simple. Data residing on Riemannian manifolds, such as the sphere, appear in many domains where either known constraints or other modeling assumptions impose a Riemannian structure. How should we visualize such data?

There are many concerns and questions when visualizing Riemannian data. The first is generic: all dimensionality reduction tools amplify parts of the signal while reducing the remainder. This inherent limitation should always be in mind when interpreting the visualization. Since some loss of information is inevitable, should we then loosen our grip on the data or its underlying Riemannian structure? Gauss's *Theorema Egregium* [GP05] informs us that if the final plot is to be presented on a *flat* screen or piece of paper, then a distortion of the Riemannian structure is inevitable.

In practice, even if we accept the limitations of a visualization, actual algorithms for Riemannian data are missing. In this paper, we develop an extension of *Stochastic Neighbor Embedding (SNE)* [HR03] to Riemannian data and thereby provide one such tool. We call this *Riemannian Stochastic Neighbor Embedding*, or *Rie-SNE* for short. Our approach can embed data observed on one Riemannian manifold on another manifold. This allows for mapping data from a Riemannian space to a two-dimensional Euclidean plane (for plotting), but also mapping to a two-dimensional sphere, or similar when the Euclidean topology is inappropriate. Rie-SNE does not claim to solve the above-mentioned limitations of visualization-through-dimensionality-reduction, but it does provide a working tool with practical merit.

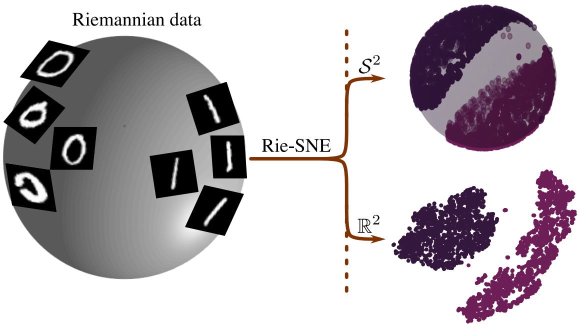

Figure 1: *Rie-SNE visualizes high-dimensional Riemannian data by mapping to a low-dimensional Riemannian (or Euclidean) manifold. The figure shows high-dimensional spherical data mapped to either a low-dimensional sphere or a low-dimensional plane. The method also supports other manifolds, both as input and output.*

2. Background and related work

Euclidean visualization. General data visualization is a vast topic, and a complete review is beyond our scope [Van16]. We here focus on the setting where data is represented as vectors (points) in an Euclidean space of high dimension. When data is two- or three-dimensional a scatter plot can directly reveal its structure, and we focus on the more difficult setting where data dimensions vastly exceed the easily plottable. Here one may explore the data through multiple projections, such as pairwise scatter plots, which are usually manageable for data of up to around 10 dimensions. Eventually, the approach tends to become unwieldy and the greater picture is lost. Alternatives include continuously interpolating between dimensions to provide an (interactive) animation [Asi85].

The approach we here explore is to find a non-linear mapping from the high-dimensional observation space into a two- or three-

DOI: 10.2312/mlvis.20241123

dimensional space, which is suitable for plotting. *Principal component analysis (PCA)* [Jol02] is arguably the most commonly used approach to dimensionality reduction. PCA is restricted to spanning a linear subspace of the observation space, which often implies that the low-dimensional view reveals little structure. Several non-linear variants exist (e.g. [Law05, TSL00, RS00, HR03]), but they tend to be brittle [FHH21]. One of the currently most popular tools is *t-SNE* [vdMH08], which we review in Sec. 2.1.

Riemannian data. Data is often equipped with additional knowledge such as constraints or given smooth structures that give the data a Riemannian structure. For example, if all observations have unit norm, they reside on the unit sphere, which has a well-studied Riemannian manifold. The a priori available knowledge giving rise to a Riemannian data interpretation differs between domains, and we here only name a few. The most prominent example is that of *directional statistics* [MJM00, KGH13, Hau18] where data resides on the unit sphere. Other examples include *shape data* [Ken84, FB12, SJML05, You12, KKG*12], *DTI images* [LDF04, PFA06, WSCV14], *image features* [TPM06, PTM06, FHB14], *motion models* [TVSC11, cV09], *human poses* [SCLBS07, HSP12, HSP10, HLP11], *robotics* [GKJH15, JBS*22], *social networks* [MLLM*19] and more.

Even if Riemannian data is becoming increasingly common, tools for visualization have not followed. The most common approach for visualizing Riemannian data is to locate a point on the manifold, e.g. the intrinsic mean [Pen06], and map the data to its tangent space. That gives an Euclidean view of the data, which can be visualized with one of the many methods for this domain. Applying PCA tangentially is the gold standard [FLPJ04]. Unless data concentrates around the point of tangency this approach gives a highly distorted view of the data, and in practice, most knowledge reflected in the geometry is lost by the linearization [SLHN10]. Several Riemannian PCA generalizations exist, e.g. [HHM10, PPY14, JDM12]. These generalize the classic linear model, and less work has been done on extending nonlinear methods. Two notable extensions are a 'wrapped' extension of the GP-LVM [MHF18], and Riemannian principal curves [Hau15].

Measuring on Riemannian manifolds. A Riemannian manifold is a space that is locally Euclidean. This implies that in a neighborhood around a point $\boldsymbol{\mu}$ on the manifold, we can get a Euclidean view of the manifold in the form of a tangent space, which is equipped with an inner product, $\langle \mathbf{x}_i, \mathbf{x}_j \rangle_{\boldsymbol{\mu}} = \mathbf{x}_i^\top G_{\boldsymbol{\mu}} \mathbf{x}_j$, where $G_{\boldsymbol{\mu}}$ is a symmetric positive definite matrix that reflects the inner product at $\boldsymbol{\mu}$. This is allowed to change smoothly between tangent spaces, to compensate for the approximation error induced by the linear view of the manifold. This distortion can be characterized by the change in volume between the manifold and its tangent, which follows $\sqrt{\det G_{\boldsymbol{\mu}}}$ [Pen06].

The inner product allows us to define local distances, which can be integrated to provide a notion of *curve length*. That is, given a curve $\mathbf{c}$ on a manifold, we may compute its length as $\int \sqrt{\langle \dot{\mathbf{c}}_t, \dot{\mathbf{c}}_t \rangle_{\mathbf{c}_t}} \mathrm{d}t$, where $\mathbf{c}_t$ and $\dot{\mathbf{c}}_t$ to denote the position and velocity of the curve, respectively. From the notion of a curve length, it is trivial to define the distance between two points as the length of the shortest curve, i.e dist $= \min_{\mathbf{c}} \text{Length}[\mathbf{c}]$. The shortest curve commonly goes under the name *geodesic*.

2.1. Stochastic neighbor embedding

Stochastic neighbor embedding (SNE) [HR03] preserves similarity between neighboring points when mapped to a low-dimensional representation. Assume that we have access to functions s_{high} and s_{low}, which measure the similarity between observation pairs in the high-dimensional observation space and the low-dimensional representation space, respectively. Now define the conditional probability, $p_{j|i}$ that $\mathbf{x}_i$ would pick $\mathbf{x}_j$ as its neighbor [vdMH08]

$$p_{j|i} = \frac{s_{\text{high}}(\mathbf{x}_j|\mathbf{x}_i)}{\sum_{k\neq i} s_{\text{high}}(\mathbf{x}_k|\mathbf{x}_i)}, \tag{1}$$

with $pi|i = 0$. We can renormalize this to form a distribution over all n observations

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

To learn a low-dimensional representation, the key idea is to repeat the above over the low-dimensional space to form

$$q_{j|i} = \frac{s_{\text{low}}(\mathbf{y}_j|\mathbf{y}_i)}{\sum_{k\neq i} s_{\text{low}}(\mathbf{y}_k|\mathbf{y}_i)}, \qquad q_{ij} = \frac{q_{j|i} + q_{i|j}}{2n}. \tag{3}$$

We can now compare the similarity of our data and the representation with the Kullback-Leibler divergence between p_{ij} and q_{ij},

$$\text{KL}(P||Q) = \sum_{i=1}^{n}\sum_{j=1}^{n} p_{ij} \log \frac{p_{ij}}{q_{ij}}. \tag{4}$$

This can then be minimized using gradient descent with respect to the low-dimensional representation $\{\mathbf{y}_i\}_{i=1}^n$. In its classic form, SNE picks the measures of similarity as Gaussian functions

$$\begin{aligned} s_{\text{high}}(\mathbf{x}_j|\mathbf{x}_i) &= \left(2\pi\sigma_i^2\right)^{-D/2} \exp\left(-\frac{\|\mathbf{x}_j - \mathbf{x}_i\|^2}{2\sigma_i^2}\right), \\ s_{\text{low}}(\mathbf{y}_j|\mathbf{y}_i) &= (2\pi)^{-d/2} \exp\left(-\frac{\|\mathbf{y}_j - \mathbf{y}_i\|^2}{2}\right). \end{aligned} \tag{5}$$

With this choice, the normalization constants of Eq. 5 cancel out when computing $p_{j|i}$ and $q_{j|i}$. Note that this approach gives a per-observation variance σ_i^2, such that different points effectively can have different sizes of neighborhoods. To determine σ_i^2, the user specifies a *perplexity* parameter, which can be thought of as a measure of the effective number of neighbors [vdMH08],

$$\text{perplexity} = 2^{H(P_i)}, \tag{6}$$

where $H(P_i)$ is the Shannon entropy [Sha48] of P_i. For a specific user-provided perplexity, we can perform a binary search over σ_i^2 such that Eq. 6 holds. In practice, the user experiments with different choices of perplexities to see which reveals a pattern.

The t-distributed stochastic neighbor embedding (t-SNE) is the most popular variant of SNE [vdMH08]. This is motivated by the 'crowding problem', where the low-dimensional representations significantly overlap without revealing much underlying structure. The idea is to use a similarity in representation space with heavy tails. Specifically, s_{low} is a t-distribution with one degree of freedom centered around one representation,

$$s_{\text{low}}(\mathbf{y}_j|\mathbf{y}_i) = \pi^{-1}\left(1 + \|\mathbf{y}_j - \mathbf{y}_i\|^2\right)^{-1}. \tag{7}$$

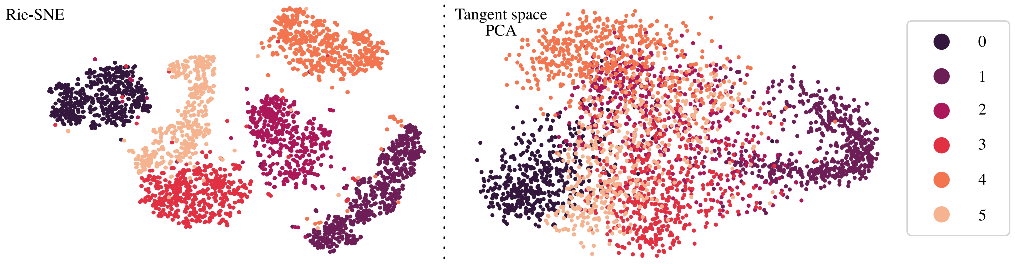

Figure 2: *Two-dimensional Euclidean embeddings of spherical MNIST. Rie-SNE (left) discovers structure, which is lost on tangent space PCA (right).*

3. Method

Brownian motion on a Riemannian manifold. The key building block for generalizing SNE to Riemannian manifolds is a suitable generalization of the Gaussian distribution. Here we consider a density derived from a Brownian motion on a Riemannian manifold for high-dimensional probability computations.

The end-point $\mathbf{x}$ of Brownian motion in Euclidean space follows a Gaussian distribution. Infinitesimal increments of the Brownian motion can be projected onto the tangent space of corresponding points on a Riemannian manifold without error. Then, a Brownian motion starting at point $\boldsymbol{\lambda}$ running for some time t can be projected onto a D-dimensional Riemannian manifold. The resulting random variable can be interpreted as the probability that a Brownian motion starting at $\boldsymbol{\lambda}$ will end in point $\mathbf{x}$ on the manifold. This approximately has density [HS02, KEAH20]:

$$\mathcal{BM}(\mathbf{x}|\boldsymbol{\lambda},t) \approx (2\pi t)^{-\frac{D}{2}} H_0 \exp\left(-\frac{\text{dist}^2(\mathbf{x},\boldsymbol{\lambda})}{2t}\right) \quad (8)$$

where $t > 0$ is the duration of the Brownian motion (variance in Euclidean space), $\boldsymbol{\lambda}$ is the starting point of the Brownian motion, and H_0 is the ratio of Riemannian volume measures evaluated at points $\mathbf{x}$ and $\boldsymbol{\lambda}$ respectively. Superficially, Eq. 8 looks like the density of the normal distribution, with the Euclidean distance being replaced by its Riemannian counterpart. The normalization factor H_0, however, differs from the Euclidean distribution.

3.1. Rie-SNE

Rie-SNE works similarly as SNE and t-SNE, i.e. it will also produce two probability distributions P and Q from the data and aim to make them as similar as possible and in the process capture some underlying structure in the produced low dimensional embedding. However, computing the high-dimensional probability distribution P comes with an added cost. To preserve the Riemannian nature of the data, a different density is used when computing high-dimensional probabilities belonging to P, namely the approximate density induced by the heat kernel of a Brownian motion on a Riemannian manifold given in Eq. 8, i.e. we pick

$$s_{\text{high}}(\mathbf{x}_j|\mathbf{x}_i) = \mathcal{BM}\mathbf{x}_j|\mathbf{x}_i,t_i. \quad (9)$$

The added computational cost is that the evaluation of $\mathcal{BM}(\cdot|\cdot)$ is more demanding than the conventional Gaussian similarity (5). Specifically, the normalization H_0 and the geodesic distance may be demanding, depending on the manifold on which the data resides.

With the Brownian motion model, we get

$$p_{j|i} \approx \frac{H_0[i,j]\cdot\exp\left(-\frac{\text{dist}^2[i,j]}{2t_i}\right)}{\sum_{k\neq i} H_0[i,k]\cdot\exp\left(-\frac{\text{dist}^2[i,k]}{2t_i}\right)}, \quad (10)$$

where we use the notations $\text{dist}^2[i,j] = \text{dist}^2(\mathbf{x}_i,\mathbf{x}_j)$ and $H_0[i,j] = \sqrt{\det G_{\mathbf{x}_i}/\det G_{\mathbf{x}_j}}$ to emphasize that these quantities can be precomputed. As with SNE, we can optimize t_i to match a prespecified perplexity using a binary search.

Choice of representation. As mentioned in Sec. 1, Gauss's *Theorema Egregium* [GP05] inform us that we cannot isometrically embed data from a curved space into a space of different curvature without introducing distortion. Specifically, if we embed data from a nonlinear manifold onto a *flat* two-dimensional representation (for plotting) then the curvature mismatch between spaces induces a distortion. This is a fundamental limitation that any visualization of Riemannian data will face, but we may nonetheless try to limit its impact. One approach is to embed the data onto a manifold of similar curvature as that of the manifold on which the data resides. For example, if the data resides on a high-dimensional sphere, it is perhaps more prudent to embed onto a two-dimensional sphere for plotting, rather than a Euclidean space.

With this in mind, we choose different distributions over the low-dimensional representation, depending on user preference.

- **Euclidean.** If the user prefers a Euclidean low-dimensional representation, we use a student-t as in regular t-SNE.
- **Spherical.** If the data manifold has positive curvature it may be beneficial to embed it on a sphere, in which case we opt to use a von Mises-Fisher distribution [MJM00]. Spherical embeddings have been explored elsewhere, e.g. [LCY19].
- **Other.** When the user wants to embed on another low-dimensional manifold, we suggest using the Riemannian Brownian over the low-dimensional representation.

Once we have defined both s_{high} and s_{low}, we can estimate the representations using gradient descent just as regular SNE. Having performed T iterations (with a sufficiently large value for T) of the gradient descent, the two probability distributions P and Q will have a minimal KL-divergence resulting in near-optimal positions of the points in the low-dimensional embedding.

Implementation details. Our implementation of Rie-SNE relies on two approximation techniques that are traditionally also used when performing regular t-SNE. First, we compute the high-dimensional probabilities in P using a sparse nearest neighbor-based approximation technique [vdM14]. Second, we use the

Proceedings published by Eurographics - The European Association for Computer Graphics.

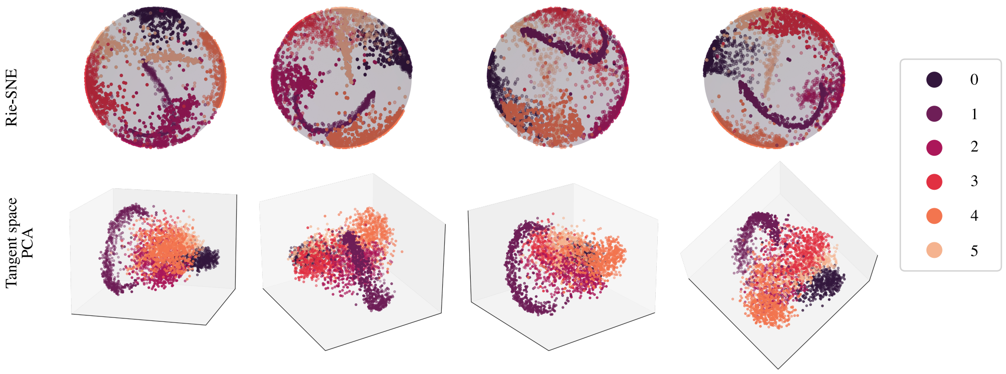

Figure 3: *Four different views of embeddings of spherical MNIST. The top row shows spherical embeddings obtained using Rie-SNE, while the bottom row shows three-dimensional embeddings using the gold standard tangent space PCA. Note that the clustering structure is significantly more evident in Rie-SNE.*

Barnes-Hut approximation [BH86], whenever we use a student's t-distribution over the low-dimensional representation. This reduces the complexity to $\mathcal{O}(n \log n)$.

4. Results

The performance of Rie-SNE is shown by comparing it to the gold standard of visualizing non-Euclidean data. This amounts to first computing the intrinsic mean, mapping all data to the tangent space at this point, and performing PCA over the tangential data.

Spherical MNIST. We start with the classic MNIST dataset consisting of 24×24 dimensional gray-scale images. We consider digits 0–5 to reduce clutter. To induce a non-Euclidean data geometry, we project the data onto the unit sphere of $\mathbb{R}^{24\times24}$ and denote the resulting data *spherical MNIST*. We visualize the resulting data using both Rie-SNE and tangent space PCA. First, we embed the data onto the plane, $\mathbb{R}^2$, and show the resulting plots in Fig. 2. Here it can be seen that Rie-SNE captures well the underlying relationship between the data points (the same digits are grouped), while tangent space PCA produces a cluttered view that does not reveal the underlying structure. Since the data has a spherical geometry, it may be beneficial to embed onto a low-dimensional sphere to preserve topology and curvature better. Figure 3 shows the Rie-SNE embedding onto $\mathcal{S}^2$, where the clustering is again evident. As a baseline, the figure also shows a tangent space PCA embedding on $\mathbb{R}^3$. Although some structure can be captured with tangent space PCA here, Rie-SNE still gives better separation of the digit classes.

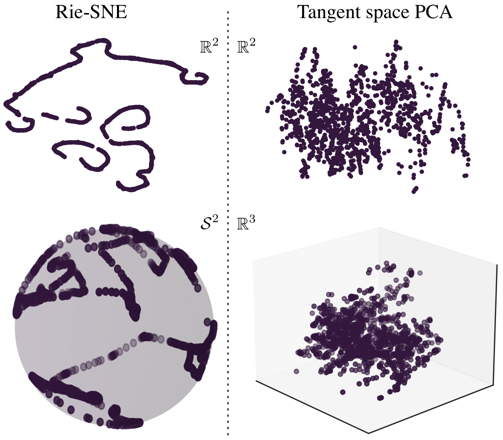

Figure 4: *Embeddings of symmetric positive definite matrices using Rie-SNE (left) and tangent space PCA (right). The top row shows two-dimensional Euclidean embeddings, while the bottom row shows spherical and $\mathbb{R}^3$ embeddings, respectively. In both cases, Rie-SNE recovers a one-dimensional signal matching the underlying time series, while tangent space PCA does not.*

Crypto-tensors. Following Mallasto et al. [MHF18] we consider the price of 10 popular crypto-currencies over the period *2.12.2014 — 15.5.2018*. As is common in economy [WG10], the relationship between prices is captured by a 10×10 covariance matrix constructed from the past 20 days. This gives rise to a time series of covariance matrices, each of which resides on the cone of symmetric positive definite matrices. We provide visualizations of the data in Fig. 4. Rie-SNE is used to produce visualizations on both the plane $\mathbb{R}^2$ and the sphere $\mathcal{S}^2$, showing a one-dimensional structure capturing the time evolution behind the data. In contrast, tangent space PCA produces $\mathbb{R}^2$ and $\mathbb{R}^3$ visualizations showing little to no structure in the embeddings.

5. Conclusions

We presented a new type of visualization technique, *Rie-SNE*, that is aimed at data residing on Riemannian manifolds, such as spheres. It is an SNE-based technique that can additionally produce different kinds of low-dimensional embeddings depending on user preference and the curvature of the original data manifold. We achieve favorable results using Rie-SNE compared to the gold standard tangential PCA. We hope that the present work also paves the way for other visualization tools for Riemannian data in order to support investigators relying on geometric models.

Acknowledgements. SH was supported by research grants (15334, 42062) from VILLUM FONDEN. This project has also received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement n° 757360). This work was funded in part by the Novo Nordisk Foundation through the Center for Basic Machine Learning Research in Life Science (NNF20OC0062606).

References

[Asi85] ASIMOV D.: The grand tour: a tool for viewing multidimensional data. *SIAM journal on scientific and statistical computing 6*, 1 (1985). 1

[BH86] BARNES J. E., HUT P.: A hierarchical O(n-log-n) force calculation algorithm. *Nature 324* (1986), 446. 4

[cV09] ÇETINGUL H., VIDAL R.: Intrinsic mean shift for clustering on stiefel and grassmann manifolds. In *IEEE CVPR* (2009). 2

[FB12] FREIFELD O., BLACK M.: Lie bodies: A manifold representation of 3D human shape. In *ECCV* (2012). 2

[FHB14] FREIFELD O., HAUBERG S., BLACK M. J.: Model transport: Towards scalable transfer learning on manifolds. In *CVPR* (2014). 2

[FHH21] FELDAGER C. W., HAUBERG S., HANSEN L. K.: Spontaneous symmetry breaking in data visualization. In *ICANN* (2021). 2

[FLPJ04] FLETCHER P. T., LU C., PIZER S. M., JOSHI S.: Principal Geodesic Analysis for the study of Nonlinear Statistics of Shape. *IEEE Transactions on Medical Imaging (TMI) 23*, 8 (2004), 995–1005. 2

[GKJH15] GILITSCHENSKI I., KURZ G., JULIER S. J., HANEBECK U. D.: Unscented orientation estimation based on the bingham distribution. *IEEE Transactions on Automatic Control* (2015). 2

[GP05] GAUSS K. F., PESIC P.: *General investigations of curved surfaces*. Courier Corporation, 2005. 1, 3

[Hau15] HAUBERG S.: Principal curves on riemannian manifolds. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2015). 2

[Hau18] HAUBERG S.: Directional statistics with the spherical normal distribution. In *Proceedings of FUSION 2018* (2018). 2

[HHM10] HUCKEMANN S., HOTZ T., MUNK A.: Intrinsic shape analysis: geodesic PCA for Riemannian manifolds modulo isometric Lie group actions. *Statistica Sinica 20*, 1 (2010), 1–58. 2

[HLP11] HAUBERG S., LAUZE F., PEDERSEN K. S.: Unscented Kalman Filtering on Riemannian Manifolds. *JMIV* (2011). 2

[HR03] HINTON G. E., ROWEIS S.: Stochastic neighbor embedding. In *NeurIPS* (2003), vol. 15, MIT Press. 1, 2

[HS02] HSU E., SOCIETY A. M.: *Stochastic Analysis on Manifolds*. American Mathematical Society, 2002. 3

[HSP10] HAUBERG S., SOMMER S., PEDERSEN K. S.: Gaussian-like Spatial Priors for Articulated Tracking. In *ECCV* (2010), Springer. 2

[HSP12] HAUBERG S., SOMMER S., PEDERSEN K. S.: Natural metrics and least-committed priors for articulated tracking. *Image and Vision Computing 30*, 6-7 (2012), 453–461. 2

[JBS*22] JAQUIER N., BOROVITSKIY V., SMOLENSKY A., TERENIN A., ASFOUR T., ROZO L.: Geometry-aware bayesian optimization in robotics using riemannian matérn kernels. In *CoRL* (2022). 2

[JDM12] JUNG S., DRYDEN I. L., MARRON J.: Analysis of principal nested spheres. *Biometrika 99*, 3 (2012), 551–568. 2

[Jol02] JOLLIFFE I. T.: *Principal component analysis for special types of data*. Springer, 2002. 2

[KEAH20] KALATZIS D., EKLUND D., ARVANITIDIS G., HAUBERG S.: Variational autoencoders with riemannian brownian motion priors, 2020. arXiv:2002.05227. 3

[Ken84] KENDALL D.: Shape manifolds, Procrustean metrics, and complex projective spaces. *Bulletin of the London Mathematical Society 16*, 2 (1984), 81–121. 2

[KGH13] KURZ G., GILITSCHENSKI I., HANEBECK U. D.: Recursive nonlinear filtering for angular data based on circular distributions. In *2013 American Control Conference* (2013), IEEE. 2

[KKG*12] KURTEK S., KLASSEN E., GORE J. C., DING Z., SRIVASTAVA A.: Elastic geodesic paths in shape space of parameterized surfaces. *IEEE TPAMI* (2012). 2

[Law05] LAWRENCE N.: Probabilistic non-linear principal component analysis with gaussian process latent variable models. *The Journal of Machine Learning Research 6* (2005), 1783–1816. 2

[LCY19] LU Y., CORANDER J., YANG Z.: Doubly stochastic neighbor embedding on spheres. *Pattern Recognition Letters* (2019). 3

[LDF04] LENGLET C., DERICHE R., FAUGERAS O.: Inferring white matter geometry from diffusion tensor MRI: Application to connectivity mapping. In *ECCV* (2004). 2

[MHF18] MALLASTO A., HAUBERG S., FERAGEN A.: Probabilistic riemannian submanifold learning with wrapped gaussian process latent variable models. In *AISTATS* (2018). 2, 4

[MJM00] MARDIA K. V., JUPP P. E., MARDIA K.: *Directional statistics*, vol. 2. Wiley Online Library, 2000. 2, 3

[MLLM*19] MATHIEU E., LE LAN C., MADDISON C. J., TOMIOKA R., TEH Y. W.: Continuous hierarchical representations with poincaré variational auto-encoders. *NeurIPS 32* (2019). 2

[Pen06] PENNEC X.: Intrinsic Statistics on Riemannian Manifolds: Basic Tools for Geometric Measurements. *JMIV* (2006). 2

[PFA06] PENNEC X., FILLARD P., AYACHE N.: A Riemannian framework for tensor computing. *IJCV* (2006). 2

[PPY14] PANARETOS V. M., PHAM T., YAO Z.: Principal flows. *Journal of the American Statistical Association (JASA)* (2014). 2

[PTM06] PORIKLI F., TUZEL O., MEER P.: Covariance tracking using model update based on Lie algebra. In *CVPR* (2006). 2

[RS00] ROWEIS S. T., SAUL L. K.: Nonlinear dimensionality reduction by locally linear embedding. *science 290*, 5500 (2000), 2323–2326. 2

[SCLBS07] SAID S., COURTY N., LE BIHAN N., SANGWINE S. J.: Exact principal geodesic analysis for data on SO(3). In *The 15th European Signal Processing Conference* (2007). 2

[Sha48] SHANNON C. E.: A mathematical theory of communication. *Bell Syst. Tech. J. 27*, 3 (1948), 379–423. 2

[SJML05] SRIVASTAVA A., JOSHI S. H., MIO W., LIU X.: Statistical shape analysis: Clustering, learning, and testing. *TPAMI* (2005). 2

[SLHN10] SOMMER S., LAUZE F., HAUBERG S., NIELSEN M.: Manifold Valued Statistics, Exact Principal Geodesic Analysis and the Effect of Linear Approximations. In *ECCV* (2010). 2

[TPM06] TUZEL O., PORIKLI F., MEER P.: Region covariance: A fast descriptor for detection and classification. In *ECCV* (2006), Springer. 2

[TSL00] TENENBAUM J. B., SILVA V. D., LANGFORD J. C.: A global geometric framework for nonlinear dimensionality reduction. *science 290*, 5500 (2000), 2319–2323. 2

[TVSC11] TURAGA P. K., VEERARAGHAVAN A., SRIVASTAVA A., CHELLAPPA R.: Statistical computations on Grassmann and Stiefel manifolds for image and video-based recognition. *TPAMI* (2011). 2

[Van16] VANDERPLAS J.: *Python data science handbook: Essential tools for working with data*. " O'Reilly Media, Inc.", 2016. 1

[vdM14] VAN DER MAATEN L.: Accelerating t-sne using tree-based algorithms. *J. Mach. Learn. Res. 15*, 1 (Jan. 2014), 3221–3245. 3

[vdMH08] VAN DER MAATEN L., HINTON G.: Visualizing data using t-SNE. *Journal of Machine Learning Research 9* (2008). 2

[WG10] WILSON A. G., GHAHRAMANI Z.: Generalised wishart processes. *arXiv preprint arXiv:1101.0240* (2010). 4

[WSCV14] WANG Y., SALEHIAN H., CHENG G., VEMURI B. C.: Tracking on the product manifold of shape and orientation for tractography from diffusion MRI. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2014), pp. 3051–3056. 2

[You12] YOUNES L.: Spaces and manifolds of shapes in computer vision: An overview. *Image and Vision Computing 30*, 6 (2012), 389–397. 2

Proceedings published by Eurographics - The European Association for Computer Graphics.