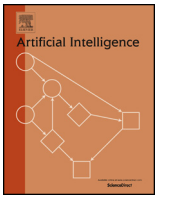
Artificial Intelligence 302 (2022) 103579

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint



Probabilistic modelling of general noisy multi-manifold data sets

M. Canducci [a,*], P. Tiňo [a], M. Mastropietro [b]

[a] School of Computer Science, University of Birmingham, Edgbaston, Birmingham B15 2TT, UK
[b] Ghent University, Krijgslaan 281, S9, B-9000 Gent, Belgium

ARTICLE INFO

Article history:
Received 6 March 2021
Received in revised form 19 June 2021
Accepted 9 August 2021
Available online 31 August 2021

Keywords:
Latent variable models
Dimensionality estimation
Multi-manifold Learning
Riemannian manifolds
Generative topographic mapping
Density estimation
Probabilistic modelling

ABSTRACT

The intrinsic nature of noisy and complex data sets is often concealed in low-dimensional structures embedded in a higher dimensional space. Number of methodologies have been developed to extract and represent such structures in the form of manifolds (i.e. geometric structures that locally resemble continuously deformable intervals of $\mathbb{R}^j$[1]). Usually a-priori knowledge of the manifold's intrinsic dimensionality is required. Additionally, their performance can often be hampered by the presence of a significant high-dimensional noise aligned along the low-dimensional core manifold. In real-world applications, the data can contain several low-dimensional structures of different dimensionalities. We propose a framework for dimensionality estimation and reconstruction of multiple noisy manifolds embedded in a noisy environment. To the best of our knowledge, this work represents the first attempt at detection and modelling of a set of coexisting general noisy manifolds by uniting two aspects of multi-manifold learning: the recovery and approximation of core noiseless manifolds and the construction of their probabilistic models. The easy-to-understand hyper-parameters can be manipulated to obtain an emerging picture of the multi-manifold structure of the data. We demonstrate the workings of the framework on two synthetic data sets, presenting challenging features for state-of-the-art techniques in Multi-Manifold learning. The first data set consists of multiple sampled noisy manifolds of different intrinsic dimensionalities, such as Möbius strip, toroid and spiral arm. The second one is a topologically complex set of three interlocked toroids. Given the absence of such unified methodologies in the literature, the comparison with existing techniques is organized along the two separate aspects of our approach mentioned above, namely manifold approximation and probabilistic modelling. The framework is then applied to a complex data set containing simulated gas volume particles from a particle simulation of a dwarf galaxy interacting with its host galaxy cluster. Detailed analysis of the recovered 1D and 2D manifolds can help us to understand the nature of Star Formation in such complex systems.

* Corresponding author.
E-mail addresses: M.Canducci@bham.ac.uk (M. Canducci), P.Tino@cs.bham.ac.uk (P. Tiňo), Michele.Mastropietro@ugent.be (M. Mastropietro).

[1] j is the manifold dimensionality. Mathematically, continuous deformation corresponds to homehomorphism: one-to-one continuous mapping with continuous inverse.

https://doi.org/10.1016/j.artint.2021.103579

1. Introduction

Dimensionality reduction and Density Estimation of raw data, are commonly used tools to extract information from complex and noisy data sets. Due to dependencies among measured attributes of real world data, the data is often distributed along low-dimensional structures in a higher dimensional measurement space. This realisation has driven the development of a variety of Manifold Learning algorithms. Principal Component Analysis (PCA) [1] is a well understood and widely used linear dimensionality reduction scheme. However, by design, PCA cannot appropriately capture non-linear low-dimensional structures. This lack of flexibility has been addressed by non-linear dimensionality reduction algorithms such as Isomap [2] and Locally Linear Embedding (LLE, [3]), where the manifold is approximated by a neighbourhood graph and a neighbouring preserving map respectively. Both methods take advantage of the definition of a manifold as a locally linear low dimensional structure of dimension j. Many other Manifold Learning algorithms aiming to provide suitable approximations to low-dimensional data manifold have been proposed. Examples include Laplacian eigenmaps [4], Hessian eigenmaps [5], Local Space Tangent Alignment (LSTA, [6]), C-Isomap [7] (an extension of Isomap to conformal embeddings), NMDS (non-metric formulation of Multidimensional Scaling (MDS) [8], [9], [10]), Riemannian Manifold Learning (RML [11]).

In [12] (and bibliographic references therein), a different angle, based on computational geometry, has been proposed in order to extract low-dimensional manifolds from data samples. Here, simplicial complexes (such as Delaunay triangulations, α-complexes and filtrations) are used for manifold reconstruction. With these techniques it is possible to infer geometrical and topological properties of data points that uniformly sample a single manifold embedded in a higher dimensional space.

While such techniques are potentially powerful and theoretically well grounded, their capability to naturally handle high-dimensional noise aligned along low-dimensional manifolds is limited. Besides the sensitivity to the noise issues, it is often required that the intrinsic data dimensionality is known a priori.

Generative Topographic Mapping (GTM) [13] was proposed as a probabilistic formulation of the Self-Organizing Map [14]. Its main advantage is that instead of treating noisy manifold as a core low-dimensional manifold to be discovered, plus some "noise" around it that somehow needs to be dealt with, it formulates a consistent manifold-aligned density model in the form of a constrained mixture of Gaussians.[2] The original GTM formulation is trained in the maximum likelihood framework using the E-M algorithm [15]. Because of the sensitivity to initialization, a suitable initialization is required e.g., using PCA, or assuming a latent space topology if known a priori [16]. Bayesian formulations of GTM have also been proposed [17].

Other global density estimators, whether non-parametric, e.g. Parzen windows [18] (and its extensions such as Manifold Parzen Windows [19], Fast-Parzen Windows [20]), or semi-parametric, e.g. Infinite Gaussian Mixture model [21], are not designed to extract a representation of the embedded low-dimensional structures. They also may be computationally expensive to train and/or evaluate.

To deal with complex data sets, where several manifolds of different dimensionalities can co-exist, generalizations of the previous methods have been developed: Multi-Manifold Discriminant Analysis (MMDA, [22]), Sparse-Manifold Clustering and Embedding (SMCE, [42]), Multi-Manifold Isomap (M-ISOMAP [23]), Multi-manifold Proximity Embedding (MPE, [24]), Multi-Manifold LLE (MM-LLE, [25]), S-Isomap++ [26], Hierarchical GTM [27]. However, based on the same assumptions as their predecessors, the methods still need to be informed about the dimensionalities of the different manifolds and struggle when dealing with topologically complex, noisy structures. The works proposed in [28] and [29], are particularly relevant in terms of dimensionality estimation and clustering. The first methodology relies on the construction of *Translation Poisson Mixture Models* (TPMM) for estimating the dimensionality of local neighbourhoods in the data. However, by design, it is prone to separate a unique manifold if the local density of points changes throughout the manifold. The second methodology, *Hidalgo*, is a Bayesian extension of *TWO-NN*[3] [30]. The dimensionality of individual points is obtained by maximization of the posterior distribution over the model's parameters. Despite the appealing formulation, the estimated complexity of $\mathcal{O}(N^2)$, N being the number of points in the data set, is not well suited for applications considered in this study, where N is generally large.

We propose a framework for automated dimensionality estimation and reconstruction of multiple noisy manifolds embedded in a noisy environment. We generalize the GTM model so that densities aligned along arbitrary manifolds (even non-orientable ones - such as Möbius strip) can be captured. This is achieved by replacing the simple Euclidean latent space (generally parametrized as a discretized interval of $\mathbb{R}^j$) with an abstract graph reflecting the topology of the data manifold that, when embedded in the data space, provides a manifold skeleton around which the noise models can be organized. This work is inspired by [31], but extends and generalizes it threefold: **(1)** it proposes a new robust dimensionality index estimation for data points, **(2)** through a dedicated manifold crawling mechanism it allows for completely abstract manifold representations in the GTM latent space (instead of a regular grid) and **(3)** it has Gaussian noise components naturally aligned along the manifold, unlike the spherical noise models in the original GTM and [31]. Manifold aligned noise models in GTM were also considered in [32], but under the assumption of simple latent space structure in the form of j-dimensional interval. The key idea was to impose larger and smaller variances in directions locally parallel and perpendicular, respectively, to the manifold. Since our latent space is a discrete structure (abstract graph representing a skeleton

[2] Location parameters (means) of the Gaussian components are constrained to lie on a smooth manifold - most commonly a smooth image in the data space of a two-dimensional interval (latent space).

[3] The methodology is based on distances to the two nearest neighbours of each point.

of a given manifold), we formulate local noise models through kernel based estimates of the local covariance matrix of the data, with trainable scale parameter to allow for optimized overlapping of the neighbouring Gaussian components.

Our work presents a radical reformulation of GTM to capture and model low-dimensional general noisy manifolds through a dedicated abstract graph-structured latent space reflecting the core manifold structure, specific to each noisy manifold. Bacciu et al. [33] present another radical generalization of GTM in the reverse direction - this time keeping the original simple latent space structure, but allowing for abstract structure in the data space - the space of trees.

The paper has the following organization: in section 2 we set up the scene, explain the broad outline of the methodology and introduce the synthetic data set on which different steps of the methodology will be demonstrated. Section 3 introduces the core model of our methodology, Abstract GTM (AGTM), representing density aligned along a single manifold. We explain how the abstract latent space graph representing topology of the data manifold is extracted through manifold crawling and how this graph is then embedded in the data space along with the suitable set of noise models. We also show how to calculate local curvature at the edges of the embedded graph, taking advantage of the smooth manifold description provided by AGTM. Section 4 defines our notion of Dimensionality index for individual points and extends the framework of section 3 to the multi-manifold case. We also offer a computationally efficient alternative to Multi-Manifold Crawling. The price to pay for gains in efficiency is weaker detection stability of low dimensional manifold entities buried in the data. Section 5 presents an experimental comparison on two synthetic data sets of our methodology with alternative multi-manifold learning and probabilistic modelling methods. Even though our methodology aims to provide density models of multiple low-dimensional manifolds buried in the data, we organise the comparative experiments separately for the multi-manifold capture and probabilistic modelling aspects of our work. This is because to the best of our knowledge, no other method exists that can simultaneously recover low dimensional representations of an unknown number of manifolds while building their probabilistic models.

Our main contributions can be summarized as follows:

- Formulation of a new dimensionality index assigned to individual points based on which point cloud can be partitioned into background points and sets representing cloud points organised along noisy low-dimensional manifold structures;
- Development of a recursive crawling algorithm for the extraction of multiple low-dimensional, noisy manifolds embedded in a higher dimensional space: Multi-Manifold crawling;
- Extension of GTM's applicability to a broad class of manifolds (e.g. non-orientable or closed manifolds) by reformulating the latent space as an abstract graph and introducing a manifold-aligned noise attached to the embedded nodes of the latent graph: Abstract GTM. The abstract latent space and its embedding allows us to *understand the important global structural features of the underlying manifolds* - an important aspect of our methodology bringing manifold learning under the umbrella of Artificial Intelligence.

The methodology[4] is applied in section 6 to an astrophysical data set resulting from a mixed *N-body/Smoothed Particle Hydrodynamics* numerical simulation of a dwarf galaxy falling into the gaseous halo of a galaxy cluster. The point cloud generated by the simulation presents non linear, noisy, low dimensional structures, providing for an ideal test bed for our methodology. We extract and model two of the most significant manifolds, suggesting a possible scenario for formation of new stars in such a disrupted dwarf galaxy. Section 8 summarizes the main achievements and concludes the paper.

2. Methodology overview

Consider a point cloud $\mathcal{Q} = \{\mathbf{t}_1, \mathbf{t}_2, ..., \mathbf{t}_L\}$, $\mathbf{t}_i \in \mathbb{R}^d$ containing points sampled from an unknown number of noisy lower-dimensional manifolds embedded in a noisy environment (e.g., points generated from a broad d-dimensional distribution). As an example, Fig. 2 shows a data set in $\mathbb{R}^3$ obtained from a collection of noisy two-dimensional (central panel) and one-dimensional (left panel, red points) manifolds. All underlying true manifolds are shown in Fig. 1 - the two one-dimensional ones (Figs. 1a and 1b: parabolic arm and spiral) and the four two-dimensional manifolds (Figs. 1c–1f: cap (hyperbolic surface), 2-toroid, S-shape surface, and Möbius strip). The noisy manifolds are embedded in a uniform noise (Fig. 2, rightmost panel). We also included a uniformly sampled 3-dimensional ball (Fig. 2, left panel, green points). Characteristics of the manifolds can be found in Table 1 and details of their parametric forms and sampling are presented in the Appendix A. Note that the total number of manifold points is only 33% of the whole data set, meaning that 67% of points would ideally be discarded in the initial filtering process.

In the following we give a brief outline of our methodology to robustly detect the manifolds and build their corresponding manifold-aligned density models. We first apply a physics-based diffusion method, structure-Aware Filtering Technique (SAF) [34], that collapses points in close vicinity of dense structures onto them, resulting in a diffused data set $\tilde{\mathcal{Q}} = \{\tilde{\mathbf{t}}_1, \tilde{\mathbf{t}}_2, ..., \tilde{\mathbf{t}}_L\}$, $\tilde{\mathbf{t}}_i \in \mathbb{R}^d$. The SAF method moves points towards high density regions, enabling points in the vicinity of a noisy manifold to migrate towards its "spine" or "mean surface". Fig. 3 shows the noisy manifolds described previously (Figs. 3a–3f) together with the corresponding recovered mean manifolds via SAF (Figs. 3g–3l). Despite a few imperfections

[4] A MATLAB implementation for the whole methodology and the data sets generation can be found at https://github.com/MarcoCanducci/general-noisy-multi-manifold-learning.git.

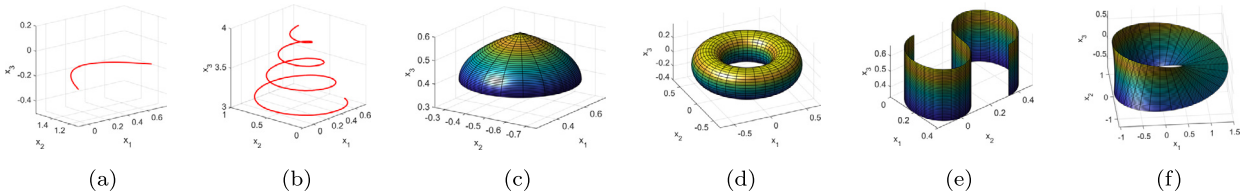


Fig. 1. Plots showing the ground-truth for each manifold used in the dataset. (For interpretation of the colours in the figure(s), the reader is referred to the web version of this article.)

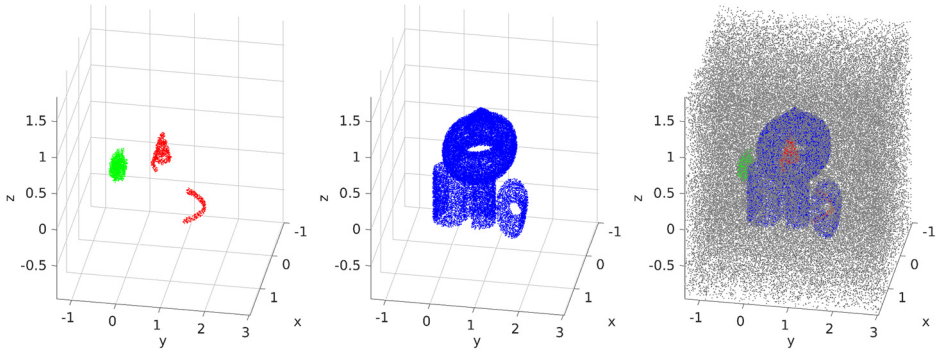


Fig. 2. Data sets containing manifolds of different dimensionalities as sampled from a uniform distribution. Merging them together and adding uniform background noise creates the final dataset for the testing phase.

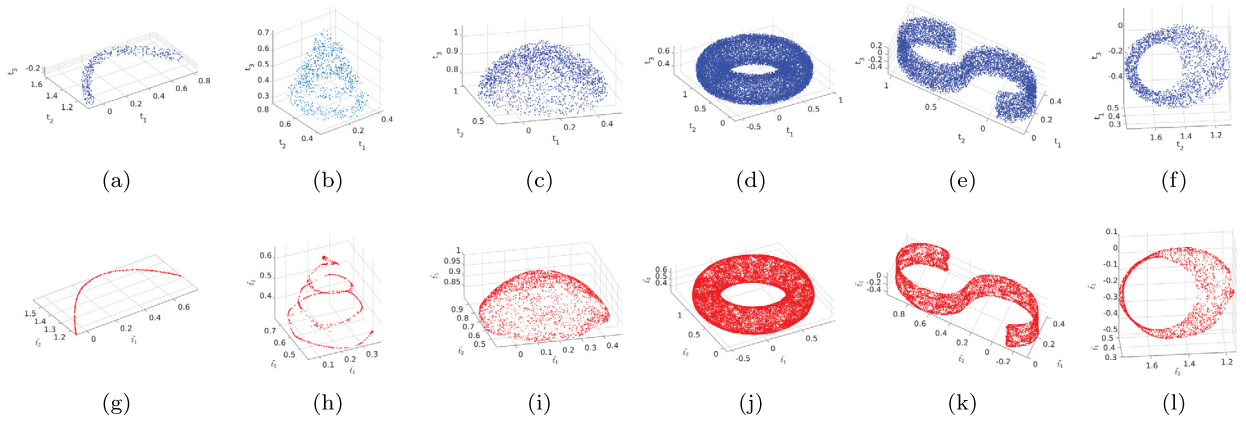


Fig. 3. Top row, noisy manifolds. Bottom row, Diffused manifolds using SAF technique.

Table 1
Manifolds' intrinsic dimensionality, underlying distribution and number of points.

Manifold ID	Dim	Distribution	# points
1	1	Spiral Arm	689
2	1	Parabolic arm	424
3	2	Hyperbolic surface	2297
4	2	toroidal surface	18959
5	2	"S" surface	8307
6	2	Möbius strip	2338
7	3	Uniformly sampled 3-d ball	1958
Summed over manifolds			34972
Background noise			70463
Dataset total			105435

when compared to their ground-truths (e.g. deformation of the spiral in the second ring from the bottom), the method generally reduces transverse noise to the manifolds. Assuming that the data structures to be modelled are more densely sampled than the noisy environment, we first filter the data sets $\mathcal{Q}$ and $\tilde{\mathcal{Q}}$ by removing point couples $(\mathbf{t}_i, \tilde{\mathbf{t}}_i)$ that have sparse neighbourhood in *both* $\mathcal{Q}$ *and* $\tilde{\mathcal{Q}}$. In particular, around each $\mathbf{t}_i \in \mathcal{Q}$ and $\tilde{\mathbf{t}}_i \in \tilde{\mathcal{Q}}$ we construct a hyperball $\mathcal{B}(\mathbf{t}_i; r)$ and $\mathcal{B}(\tilde{\mathbf{t}}_i; r)$, respectively, in $\mathbb{R}^d$ of radius $r > 0$. In case a point $\mathbf{t}_i \in \mathcal{Q}$ lies further apart from a manifold, both $\mathcal{B}(\mathbf{t}_i; r)$ and $\mathcal{B}(\tilde{\mathbf{t}}_i; r)$ will be sparsely populated. Hence, if both $\mathcal{B}(\mathbf{t}_i; r)$ and $\mathcal{B}(\tilde{\mathbf{t}}_i; r)$ contain less points from $\mathcal{Q}$ and $\tilde{\mathcal{Q}}$, respectively, than a pre-specified threshold $\tau > 0$, the points $\mathbf{t}_i$ and $\tilde{\mathbf{t}}_i$ are removed from their corresponding data sets.

Following this, the first task in capturing the multi-manifold structure in $\tilde{\mathcal{Q}}$ is to estimate local dimensionality of the cloud point around each $\tilde{\mathbf{t}}_i$ in the form of a *dimensionality index* δ_i (section 4.1). Using the dimensionality indices, we partition the data into subsets $\mathcal{Q}_j, \tilde{\mathcal{Q}}_j$ according to the local dimensionalities $j = 1, 2, ..., d$. Since $\mathcal{Q}_j, \tilde{\mathcal{Q}}_j$ can contain several distinct sampled manifolds of dimensionality j, we use a dedicated "manifold crawling" procedure operating on $\tilde{\mathcal{Q}}$ to separate the individual manifolds (section 4.2). Moreover, the crawling also produces for each manifold a graph structure embedded in $\mathbb{R}^d$ representing a piece-wise linear "skeleton" approximation of the spine of the noisy manifold (section 3.2). For every manifold, the associated graph will then function as an abstract latent space of a generalized form of Generative Topographic Mapping (GTM) [13] that we call *Abstract GTM* (AGTM). Using points in $\mathcal{Q}$, the generalized GTM produces manifold aligned density models (section 3.1). Finally, if a single summary model is required, density models of all detected manifolds (AGTMs) can be grouped together in a hierarchical mixture model [27] representing in a concise manner the global density of the low-dimensional structures in the dataset $\mathcal{Q}$.

3. Density model of a single noisy manifold

In this section we deal with construction of manifold-aligned density model for a single manifold. For the sake of presentation clarity, slightly abusing mathematical notation, we use $\mathcal{D} = \{\mathbf{t}_1, \mathbf{t}_2, ..., \mathbf{t}_N\}$ to denote the subset of points from $\mathcal{Q}$ belonging to that manifold. The set $\tilde{\mathcal{D}} = \{\tilde{\mathbf{t}}_1, \tilde{\mathbf{t}}_2, ..., \tilde{\mathbf{t}}_N\}$ contains the corresponding diffused points from $\tilde{\mathcal{Q}}$. Section 3.2 presents a recursive algorithm (Manifold Crawling) capable of recovering this latent graph, both in its abstract form and as an embedded graph in the data space.

3.1. Abstract GTM

Let us consider a j-dimensional manifold $\mathcal{M}$ embedded in a higher dimensional space $\mathbb{R}^d$, $j < d$. In the following, we assume that the dimensionality j of the manifold $\mathcal{M}$ is known. We will later (section 4.1) provide a methodology for the estimation of the intrinsic dimensionality of points lying on low-dimensional manifolds. We consider $\tilde{\mathcal{D}} = \{\tilde{\mathbf{t}}_1, \tilde{\mathbf{t}}_2, \ldots, \tilde{\mathbf{t}}_N\}$ a data sample from $\mathcal{M}$, obtained from the noisy manifold sample $\mathcal{D}$ through SAF (see section 2). The original GTM [13] assumes that the manifold $\mathcal{M}$ is an image of the j-dimensional interval $[-1, +1]^j$ (latent space) under a smooth embedding $\mathbf{y} : [-1, +1]^j \to \mathbb{R}^d$. To add noise to $\mathcal{M}$, the latent space $[-1, +1]^j$ is covered by a regular grid $\{\mathbf{x}_i\}_{i=1}^K$ of K points, whose images $\mathbf{y}(\mathbf{x}_i)$ under $\mathbf{y}$ form a "skeleton" of $\mathcal{M}$ in the data space. A spherical Gaussian noise model is then positioned at each skeleton node $\mathbf{y}(\mathbf{x}_i)$, thus representing the noisy manifold as a mixture of K Gaussians centred at $\mathbf{y}(\mathbf{x}_i)$ (see Fig. 4a). The latent space grid structure can be represented by an abstract undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each grid point $\mathbf{x}_i$ corresponds to a vertex $v_i \in \mathcal{V}$ and edges $e_{ij} \in \mathcal{E}$ are connecting vertices corresponding to neighbouring grid points $\mathbf{x}_i$ and $\mathbf{x}_j$. The structure of $\mathcal{G}$ is directly reflected in the latent grid shown in Fig. 4b. We will generalize the GTM model in that the manifold skeleton will be obtained by direct embedding of the latent graph $\mathcal{G}$ into the data space through a parametrized mapping $\mathbf{f}$. Individual Gaussian noise models will be centred at the images $\overline{\mathbf{v}}_i = \mathbf{f}(v_i) \in \mathbb{R}^d$ of the nodes of $\mathcal{G}$. The manifold skeleton will then be formed by the embedded graph $\overline{\mathcal{G}} = (\overline{\mathcal{V}}, \overline{\mathcal{E}})$, where $\overline{\mathcal{V}} = \{\overline{\mathbf{v}}_i, ..., \overline{\mathbf{v}}_K\}$ and $\overline{e}_{ij} \in \overline{\mathcal{E}}$ whenever $e_{ij} \in \mathcal{E}$. This will enable us to naturally represent density models of noisy manifolds of much more intricate structure than that of a smoothly embedded low dimensional interval. For example, the sample from noisy Möbius strip shown in Fig. 4b (right panel) is captured by a Gaussian mixture with centres at fixed skeleton points in Fig. 4b (central panel), The skeleton points themselves are images under $\mathbf{f}$ of vertices $v \in \mathcal{V}$ (Fig. 4b, left panel). The abstract graph $\mathcal{G}$ is presented with the usual topological convention by identification of opposite sides of a rectangular grid with inversion of direction. The additional edges resulting from such an identification are shown in red.

Motivated by the original GTM model, we formulate the embedding $\mathbf{f} : \mathcal{V} \to \mathbb{R}^d$ as a nonlinear model, linear in parameters. In particular, we will use a set of M basis functions $\phi_m : \mathcal{V} \to \mathbb{R}$, $m = 1, 2, ..., M$, operating on the abstract latent space $\mathcal{G}$. The image of a vertex $v \in \mathcal{V}$ under $\Phi = (\phi_1, ..., \phi_M)^\top$ is obtained as

$$\overline{\mathbf{v}} = \mathbf{f}(v; \mathbf{W}) = \mathbf{W}\Phi(v), \tag{1}$$

where $\mathbf{W}$ is a $d \times M$ matrix of weights. To formulate the basis functions ϕ_m, we construct a regular ϵ-net of the graph $\mathcal{G}$ (see e.g., [12]) with M nodes. The nodes $\mathbf{c}_m \in \mathcal{V}$ of the ϵ-net form the centres of the corresponding basis functions:

$$\phi_m(v) = \exp\left\{-\frac{D^2(v, c_m)}{\gamma^2}\right\},$$

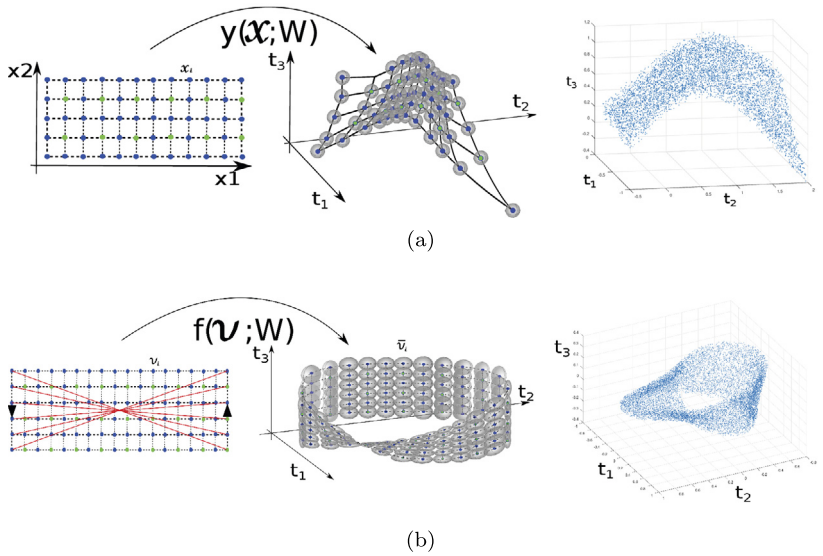


Fig. 4. Panel a shows the classic GTM setup; the leftmost panel is a representation of the latent space discretized in a grid of points x_i. The j-dimensional interval $[-1,+1]^j$ is mapped through the parametrized function $\mathbf{y}(\mathcal{X};W)$ onto the data space with a spherical Gaussian centred on each point assumed as noise. The noise model aims at describing the true distribution of points in the data space (rightmost panel). Panel b is a sketch of *AGTM*. The latent space is substituted with an abstract graph, shown here for the Möbius strip, together with its topological representation (the two arrows pointing in opposite directions on the shortest edges of the graph). Function $f(\mathcal{V};\mathbf{W})$ maps the graph onto the data space and manifold-aligned noise models are estimated on each graph's node $\overline{v}_i$. The true noisy distribution is displayed in the rightmost panel.

where $D(v,c_m)$ is the shortest distance in $\mathcal{G}$ between the vertices v and c_m. Analogously to the original GTM, to ensure smoothness of the embedding $\mathbf{f}$, the scale parameter was set[5] to 2ϵ.

The manifold aligned probabilistic model is a flat mixture model

$$p(\mathbf{t}|\mathbf{W},\boldsymbol{\zeta}) = \frac{1}{K}\sum_{i=1}^{K} p(\mathbf{t}|v_i,\zeta_i\Sigma_i,\mathbf{W}), \tag{2}$$

where the mixture components are locally manifold-aligned multivariate Gaussians centred at the embedded vertices $\overline{\mathbf{v}}_i \in \overline{\mathcal{V}}$:

$$p(\mathbf{t}|v_i,\zeta_i\Sigma_i,\mathbf{W}) = \frac{1}{[(2\pi\zeta_i)^d|\Sigma_i|]^{\frac{1}{2}}}\exp\left\{-\frac{\Delta\mathbf{t}^\top\Sigma_i^{-1}\Delta\mathbf{t}}{2\zeta_i}\right\} \tag{3}$$

with $\Delta\mathbf{t} = \mathbf{f}(v_i;\mathbf{W}) - \mathbf{t}$. Unlike in [32], we model the local manifold-aligned covariance matrix as a scaled version (scaled by $\zeta_i > 0$) of the local covariance matrix estimated as

$$\Sigma_i = \frac{1}{\sum_{a=1}^{N}\kappa(\overline{\mathbf{v}}_i,\mathbf{t}_a;\ell_i)}\sum_{n=1}^{N}\kappa(\overline{\mathbf{v}}_i,\mathbf{t}_n;\ell_i)(\mathbf{t}_n-\overline{\mathbf{v}}_i)(\mathbf{t}_n-\overline{\mathbf{v}}_i)^\top,$$

with Gaussian smoothing kernel

$$\kappa(\overline{\mathbf{v}}_i,\mathbf{t}_n;\ell_i) = \exp\left\{-\frac{\|\overline{\mathbf{v}}_i-\mathbf{t}_n\|^2}{3\ell_i^2}\right\}.$$

The kernel scale parameter ℓ_i is determined as the average of the distances between the embedded vertex $\overline{\mathbf{v}}_i$ and its neighbouring vertices in $\mathcal{G}$, embedded in $\mathbb{R}^d$,

$$\ell_i = \frac{1}{|\mathcal{N}(v_i)|}\sum_{a\in\mathcal{N}(v_i)}\|\overline{\mathbf{v}}_i-\overline{\mathbf{a}}\|,$$

where $\mathcal{N}(v_i)\subset G$ is the set of neighbours of v_i in the graph $\mathcal{G}$.

[5] Alternatively, the value of γ can be set according to a suitable criterion through cross-validation.

As a latent variable model, AGTM can be trained to maximize log-likelihood

$$\mathcal{L}(\mathbf{W}, \boldsymbol{\zeta}) = \sum_{n=1}^{N} \ln \left\{ \frac{1}{K} \sum_{i=1}^{K} p(\mathbf{t}_n | v_i, \zeta_i \Sigma_i, \mathbf{W}) \right\} \tag{4}$$

via the E-M algorithm. In the E-step, which is the same as in the original GTM model (in fact, in any mixture model), the responsibilities R_n^i are calculated as the posterior distribution over nodes $v_i \in \mathcal{V}$, given the data points $\mathbf{t}_n \in \mathcal{D}$ [13]. The M-step updates for parameters $\mathbf{W}$ and $\boldsymbol{\zeta}$ are obtained by differentiating the expected value of the complete data log-likelihood $\langle \mathcal{L}(\mathbf{W}, \boldsymbol{\zeta}) \rangle$ (equation (4)) w.r.t. the corresponding parameters and setting to zero:

$$\nabla_{\mathbf{W}} \langle \mathcal{L}(\mathbf{W}, \boldsymbol{\zeta}) \rangle = 0; \tag{5}$$

$$\nabla_{\zeta} \langle \mathcal{L}(\mathbf{W}, \boldsymbol{\zeta}) \rangle = 0, \tag{6}$$

where

$$\langle \mathcal{L}(\mathbf{W}, \boldsymbol{\zeta}) \rangle = \sum_{n=1}^{N} \sum_{i=1}^{K} R_n^i(\mathbf{W}^{old}, \zeta^{old}) \ln[p(\mathbf{t}_n | v_i, \zeta_i \Sigma_i, \mathbf{W})] \tag{7}$$

In particular,

$$\begin{aligned} \nabla_{\mathbf{W}} \langle \mathcal{L}(\mathbf{W}, \boldsymbol{\zeta}) \rangle &= -\sum_{n=1}^{N} \sum_{i=1}^{K} R_n^i \nabla_{\mathbf{W}} \frac{1}{2\zeta_i} [\mathbf{W}\Phi_i - \mathbf{t}_n]^\top \Sigma_i^{-1} [\mathbf{W}\Phi_i - \mathbf{t}_n] \\ &= -\sum_{n=1}^{N} \sum_{i=1}^{K} \frac{R_n^i}{\zeta_i} \Sigma_i^{-1} [\mathbf{W}\Phi_i - \mathbf{t}_n] \Phi_i^\top = 0. \end{aligned}$$

We rewrite the last equality in matrix notation:

$$\sum_{i=1}^{K} \mathcal{C}_i \mathbf{W} \Psi_i = \sum_{i=1}^{K} \frac{\Sigma_i^{-1}}{\zeta_i} \mathbf{T}\mathbf{R}_i \Phi_i^\top, \tag{8}$$

where $\mathbf{R}_i$ is the $(N \times 1)$ column vector containing the responsibilities for every point $\mathbf{t}_n$ w.r.t. centre $\overline{v}_i$, $\mathbf{T}$ the $D \times N$ matrix having all $\mathbf{t}_n$'s as column vectors, $\Phi_i = \Phi(v_i)$ $(M \times 1)$, $\Psi_i = \Phi_i \Phi_i^\top$ and $\mathcal{C}_i = \left(\sum_{n=1}^{N} R_n^i \right) \Sigma_i^{-1} / \zeta_i$.

We can solve equation (8) by taking advantage of the properties of the Kronecker product and vec operator (see [35], sections 5.1 and 10.2), obtaining:

$$\mathrm{vec}(\mathbf{W}^{new}) = \left(\sum_{i=1}^{K} \Psi_i^\top \otimes \mathcal{C}_i \right)^{-1} \mathrm{vec}\left(\sum_{i=1}^{K} \frac{\Sigma_i^{-1}}{\zeta_i} \mathbf{T}\mathbf{R}_i \Phi_i^\top \right). \tag{9}$$

We now turn our attention to equation (6). We have:

$$\begin{aligned} \frac{\partial}{\partial \zeta_i} \langle \mathcal{L}(\mathbf{W}, \boldsymbol{\zeta}) \rangle &= \sum_{n=1}^{N} \frac{\partial}{\partial \zeta_i} \left[-\frac{D}{2} \ln \zeta_i - \frac{(\mathbf{W}\Phi_i - \mathbf{t}_n)^\top \Sigma_i^{-1} (\mathbf{W}\Phi_i - \mathbf{t}_n)}{2\zeta_i} \right] \\ &= \sum_{n=1}^{N} R_n^i \left[\frac{D}{\zeta_i} - \frac{(\mathbf{t}_n - \overline{\mathbf{v}}_i)^\top \Sigma_i^{-1} (\mathbf{t}_n - \overline{\mathbf{v}}_i)}{\zeta_i^2} \right] = 0. \end{aligned}$$

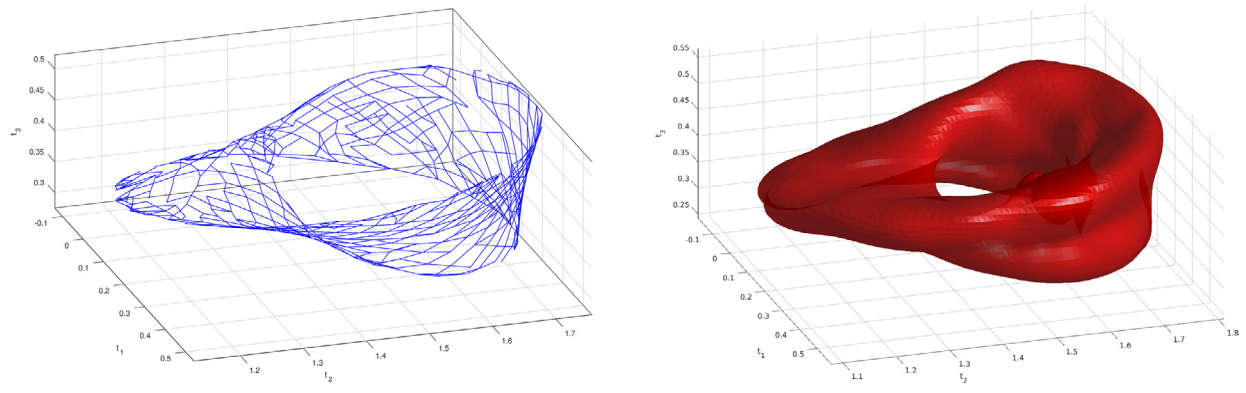
Solving this for every ζ_i, the scale parameters $\boldsymbol{\zeta}$ of the local covariance matrices are updated as

$$\zeta_i^{new} = \frac{1}{d \sum_{n=1}^{N} R_i^n} \sum_{n=1}^{N} R_n^i D_M^2(\mathbf{t}_n, \overline{\mathbf{v}}_i), \tag{10}$$

where $D_M^2(\mathbf{t}_n, \overline{\mathbf{v}}_i)$ is the squared Mahalanobis distance between $\mathbf{t}_n$ and $\overline{\mathbf{v}}_i = \mathbf{W}\Phi_i$:

$$D_M^2(\mathbf{t}_n, \overline{v}_i) = (\mathbf{t}_n - \overline{\mathbf{v}}_i)^\top \Sigma_i^{-1} (\mathbf{t}_n - \overline{\mathbf{v}}_i). \tag{11}$$

This has an intuitive interpretation: the scale parameter ζ_i of the i-th Gaussian mixture component is obtained as the (effective) squared mean Mahalanobis distance of points around component i to its centre $\overline{\mathbf{v}}_i$, per dimension. As an example, consider the noisy sample around Möbius strip shown in Fig. 4b (right panel). Fig. 5a presents the embedded abstract graph and Fig. 5b an iso-surface of the density model given by the AGTM model trained on the sample.



(a) (b)

Fig. 5. Left panel: Embedded graph of the Möbius strip used for initialization of AGTM. Right panel: Probabilistic model obtained after 2 iterations of AGTM when initialised over graphs $\mathcal{G}$ and $\overline{\mathcal{G}}$ with a manifold aligned noise.

3.2. Manifold crawling

The previous section specified how to obtain probabilistic models of noisy manifolds using AGTM. In the following, we will describe a recursive algorithm that enables us to recover the abstract graph and the embedding function $\mathbf{f}$, given the data set $\tilde{\mathcal{D}}$. We will refer to this procedure as *Manifold Crawling*, or simply *Crawling*. Recall that the manifold dimensionality is $j < d$.

Initialization

Initially, a single "seed" $\tilde{\mathbf{t}}_0 \in \tilde{\mathcal{D}}$ is randomly chosen and Principal Component Analysis (PCA [36]) is applied to its neighbourhood $\mathcal{B}(\tilde{\mathbf{t}}_0, r) \cap \tilde{\mathcal{D}}$. We assume that the unit length eigenvectors obtained through PCA are ordered in descending order of their corresponding eigenvalues.

The first j eigenvectors $\mathbf{u}_1, \ldots, \mathbf{u}_j$ span the tangent space to manifold $\mathcal{M}$ at $\tilde{\mathbf{t}}_0$, $T_{\tilde{\mathbf{t}}_0}\mathcal{M} := \text{span}\{\mathbf{u}_1, \ldots, \mathbf{u}_j\}$. For every eigenvector $\mathbf{u}_l$, $l = 1, \ldots, j$, two new points are computed at a distance $\eta \cdot r$ from $\tilde{\mathbf{t}}_0$ along directions $\pm\mathbf{u}_l$:

$$\mathbf{z}_l^{\pm} = \tilde{\mathbf{t}}_0 \pm \eta \cdot r \cdot \mathbf{u}_l^{\top} \tag{12}$$

where $0 < \eta < 1$ is a step-size parameter. In order to keep the crawling adherent to the diffused sample $\tilde{\mathcal{D}}$, the closest neighbour to every $\mathbf{z}_l^{\pm}$ is found:

$$\tilde{\mathbf{t}}_l^{\pm} = \underset{\tilde{\mathbf{t}} \in \tilde{\mathcal{D}}}{\text{argmin}} \, ||\tilde{\mathbf{t}} - \mathbf{z}_l^{\pm}||. \tag{13}$$

We start the construction of the embedded graph $\overline{\mathcal{G}}$ by initializing the set of nodes and edges:

$$\overline{\mathcal{V}} = \{\overline{\mathbf{v}}_0, \overline{\mathbf{v}}_1, \ldots, \overline{\mathbf{v}}_{2j-1}, \overline{\mathbf{v}}_{2j}\}, \tag{14}$$

$$\overline{\mathcal{E}} = \{(\overline{\mathbf{v}}_0, \overline{\mathbf{v}}_1), (\overline{\mathbf{v}}_0, \overline{\mathbf{v}}_2), \ldots, (\overline{\mathbf{v}}_0, \overline{\mathbf{v}}_{2j-1}), (\overline{\mathbf{v}}_0, \overline{\mathbf{v}}_{2j})\}, \tag{15}$$

where

$$\overline{\mathbf{v}}_0 = \tilde{\mathbf{t}}_0, \ \overline{\mathbf{v}}_1 = \tilde{\mathbf{t}}_1^{+}, \ \overline{\mathbf{v}}_2 = \tilde{\mathbf{t}}_1^{-}, \ \ldots, \ \overline{\mathbf{v}}_{2j-1} = \tilde{\mathbf{t}}_j^{+}, \ \overline{\mathbf{v}}_{2j} = \tilde{\mathbf{t}}_j^{-}.$$

At the same time we initialize construction of the abstract graph $\mathcal{G}$ by imposing that every node $\overline{\mathbf{v}}_l$ in $\overline{\mathcal{V}}$ corresponds to a node $v_l \in \mathcal{V}$ in the abstract counterpart $\mathcal{G}$ of $\overline{\mathcal{G}}$. The node $\overline{\mathbf{v}}_l$ can be considered the image of v_l through an unknown embedding function $\mathbf{f} : \mathcal{V} \longrightarrow \mathbb{R}^d$, $\overline{\mathbf{v}}_l = \mathbf{f}(v_l)$. We define the pull-back edges in the abstract graph $\mathcal{G}$ as $\mathcal{E} = \{(v_0, v_1), (v_0, v_2), \ldots, (v_0, v_{2j})\}$. We now compute the degree of each node in graph $\overline{\mathcal{G}}$ (analogously in $\mathcal{G}$). The degree $deg(\overline{\mathbf{v}}_i)$ of node $\overline{\mathbf{v}}_i \in \overline{\mathcal{G}}$ is the number of edges that are incident to that node. We select vertices having only one edge in $\overline{\mathcal{V}} \supset \overline{\mathcal{V}}_1^E = \{\overline{\mathbf{v}} \mid deg(\mathbf{v}) = 1\}$.

After the initialization phase, each successive iteration is split into two phases: *expansion* and *contraction*. The two phases are alternated for each newly discovered node $\overline{\mathbf{v}}_i \in \overline{\mathcal{V}}_1^E$ (blue dots in Fig. 6) having only one edge[6] connecting it to its parent node $\overline{\mathbf{v}}_p$.

[6] Note that the newly discovered nodes necessarily will be of degree 1.

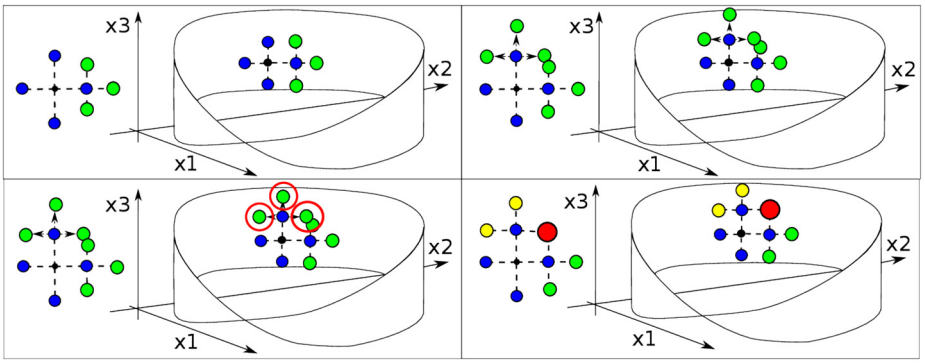


Fig. 6. Visual representation of the Expansion and Contraction phases in the *Crawling* algorithm. Starting from the initial 1-edge nodes (blue points), the first set of node candidates (green points) are found (top panel) along the directions represented by black dashed lines. For every candidate $\tilde{\mathbf{t}}_c$, we construct $\mathcal{B}(\tilde{\mathbf{t}}_c, \beta \cdot r) \cap \overline{\mathcal{V}}$ (red circles in bottom left panel). If the ball is non empty, its only element takes the place of $\tilde{\mathbf{t}}_c$ and is added to the graph (bottom right panel, red point).

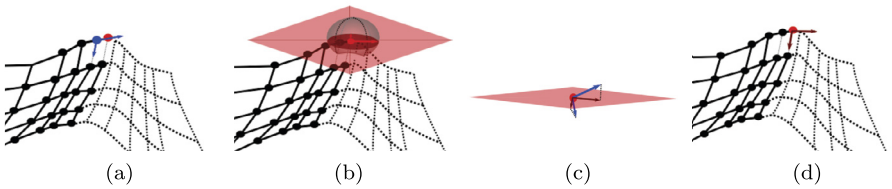


Fig. 7. Projection of tangent space (spanned by blue arrows) onto new node's estimate (red arrows).

Expansion phase

Given a node $\overline{\mathbf{v}}_i$ in $\overline{\mathcal{V}}_1^E$, the eigen-decomposition of its neighbourhood $\mathcal{B}(\overline{\mathbf{v}}_i, r) \cap \tilde{\mathcal{D}}$ is performed, yielding the first j leading eigenvectors $\mathbf{g}_1, \ldots, \mathbf{g}_j$. To preserve the crawling directions, the eigenvectors of the parent's node $\mathbf{u}_1^p, \mathbf{u}_2^p, \ldots, \mathbf{u}_j^p$ are projected onto the tangent space at $\overline{\mathbf{v}}_i$, $T_{\overline{\mathbf{v}}_i}\mathcal{M} := \text{span}\{\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_j\}$ and renormalized to unit length:

$$\mathbf{u}_i = \frac{1}{||\mathbf{P}\mathbf{u}_i^p||}\mathbf{P}\mathbf{u}_i^p; \qquad i = 1, \ldots, j. \tag{16}$$

Here $\mathbf{P} = \mathbf{V}\mathbf{V}^+$ is the orthogonal projection operator onto $T_{\overline{\mathbf{v}}_i}\mathcal{M}$, with $\mathbf{V}$ the matrix containing $\mathbf{g}_l$ as columns and $\mathbf{V}^+$ its pseudo-inverse. (see Fig. 7). As in the initialization step, new points are then computed using equation (12) and their closest neighbours $\tilde{\mathbf{t}}_l^{\pm}$ in data set $\tilde{\mathcal{D}}$ stored as candidate nodes to be added to $\overline{\mathcal{V}}$. The result of the expansion phase for a given iteration is shown in Fig. 6. The green points are the new candidates, the dashed black lines represent the principal crawling directions. Construction of the set of new candidate nodes $\mathcal{C}(\overline{\mathbf{v}}_i) = \{\tilde{\mathbf{t}}_l^{\pm}, l = 1, \ldots, j\}$ around $\overline{\mathbf{v}}_i$ terminates the Expansion phase.

Contraction phase

In the *contraction* phase we check if the candidate nodes together with $\overline{\mathcal{V}}$ can be collapsed into a smaller set of nodes.

For every new candidate $\tilde{\mathbf{t}}_c \in \mathcal{C}(\overline{\mathbf{v}}_i)$ from the Expansion phase, we check if it lands within a small "tolerance" neighbourhood $\mathcal{B}(\mathbf{v}_e, \beta \cdot r)$ of an already existing node $\overline{\mathbf{v}}_e$ of $\overline{\mathcal{G}}$. Here $0 < \beta < \eta < 1$ is the tolerance scale parameter. If this is the case, and $||\overline{\mathbf{v}}_e - \overline{\mathbf{v}}_i|| \leq r$, the candidate $\tilde{\mathbf{t}}_c$ is identified with $\overline{\mathbf{v}}_e$ and the nodes $\overline{\mathbf{v}}_e$ and $\overline{\mathbf{v}}_i$ are connected by an edge $(\overline{\mathbf{v}}_e, \overline{\mathbf{v}}_i)$ added to $\overline{\mathcal{E}}$ (red point in the lower right plot of Fig. 6). Otherwise, $\tilde{\mathbf{t}}_c$ is confirmed as a new node (under the constraint $||\tilde{\mathbf{t}}_c - \overline{\mathbf{v}}_i|| \leq r$) and added to $\overline{\mathcal{V}}$ with the edge $(\tilde{\mathbf{t}}_c, \overline{\mathbf{v}}_i)$ added to $\overline{\mathcal{E}}$ (yellow points in the lower right plot of Fig. 6). This marks the end of the *contraction* phase. The iteration is concluded when *expansion* and *contraction* have been performed for every $\overline{\mathbf{v}}_i$ in the single-edge node set $\overline{\mathcal{V}}_1^E$ from the previous iteration.

After the *expansion* and *contraction* phases, the single-edge node set $\overline{\mathcal{V}}_1^E$ is updated and the new iteration of *expansion-contraction* steps is started. Manifold crawling stops when the number of nodes in $\overline{\mathcal{V}}$ no longer increases in two consecutive iterations. The application of Manifold crawling to the point cloud obtained by diffusion from a sampled noisy Möbius strip, is presented in Fig. 8. Iteration 1 (Fig. 8a) of the algorithm, after the initialization, recovers the principal crawling directions. The crawling is then propagated along the tangent bundle of the manifold, building a series of growing abstract and embedded latent graphs representing the manifold's structure (Figs. 8b–8j).

Initialization of the latent graph embedding $\mathbf{f}(v; \mathbf{W})$

The node sets $\overline{\mathcal{V}}$ and $\mathcal{V}$ will be used to initialize the embedding $\mathbf{f}(v; \mathbf{W})$ (equation (1)) of the abstract latent space $\mathcal{G}$ into the data space $\mathbb{R}^d$ by setting the weight matrix $\mathbf{W}$ through linear regression so that $\mathbf{f}(v_i; \mathbf{W}) \approx \overline{\mathbf{v}}_i$ for all corresponding

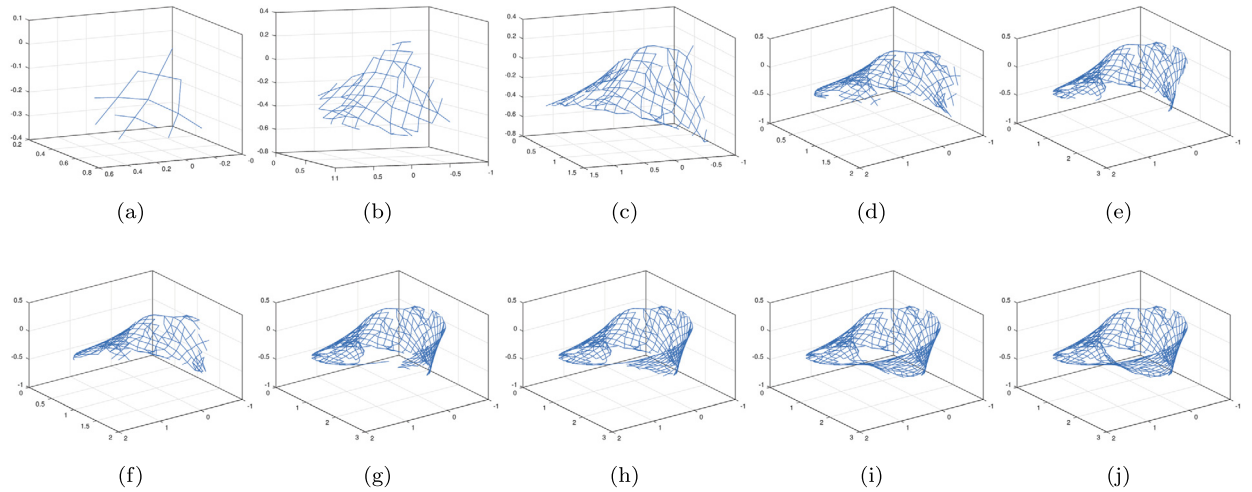


Fig. 8. Consecutive iterations of the Crawling algorithm shown for the Möbius strip. At the end of the first iteration after initialization (panel a) the main directions of crawling are already identified and the first version of the graph formed. The following iterations spread the graph over the whole manifold while crawling on local tangent spaces, until completion (b–j).

couples $(v_i \in \mathcal{V}, \overline{\mathbf{v}}_i \in \overline{\mathcal{V}})$. While $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is unweighted, the edges in $\overline{\mathcal{G}}$ are weighted by the distance between the two connected nodes.

If needed, one can regularize the parameters $\mathbf{W}$ by applying *ridge regression* instead of linear regression, by minimization of the cost function:

$$\mathcal{F} = \sum_{i=1}^{|\mathcal{V}|} (\overline{\mathbf{v}}_i - \mathbf{f}(v_i))^2 + \sum_{j=1}^{M} \chi_j \mathbf{w}_j^2, \tag{17}$$

solving the system of equations:

$$(\Phi^\top \Phi + \mathbf{X})\hat{\mathbf{W}} = \Phi^\top \overline{\mathcal{V}}. \tag{18}$$

Here, $\mathbf{X}$ is the diagonal matrix having as diagonal elements the regularization parameters $\{\chi_j\}_{j=1}^{M}$. The solution to the system in equation (18) is the *normal equation* [37]:

$$\hat{\mathbf{W}} = (\Phi^\top \Phi + \mathbf{X})^{-1} \Phi^\top \overline{\mathcal{V}}. \tag{19}$$

In order to recover a smooth mapping function with small optimal weights $\hat{\mathbf{W}}$, in our tests we chose $\chi_j = \chi = 1e-5$ for all $j = 1 \ldots, M$.

3.2.1. Local curvatures

As outlined in [27], having obtained a smooth embedding of the latent space, we can now estimate local curvatures along edges of the abstract graph. Consider a node $v_0 \in \mathcal{V}$ and its edge $e_{0,1}$ connecting it to node v_1. Since $\mathcal{G}$ is an unweighted graph, we assume, without loss of generality, that the weight on the edge $e_{0,1}$ is $w_{0,1} = 1$. We would like to estimate the effect of a small perturbation $0 < h << 1$ on the position of the node v_0 along the edge $e_{0,1}$, relative to the basis functions centres $c_m,\ m = 1, \ldots, M$.

We can consider $e_{0,1}$ as a segment of unit length and estimate the change in tangent vectors as we move along the embedded image of $e_{0,1}$. It is crucial that the tangent vectors have normalised length (in our case - unit vectors) to remove the effects of parameterisation speed. The tangent vectors are estimated using finite differences. To that end, we insert five new nodes $x_1, \ldots, x_5$ along the edge $e_{0,1}$ and connect them in a chain from node v_0 to node v_1, with edges $e_{i,i+1}$ and respective weights:

$$w(v_0, x_1) = \ldots w(x_1, x_2) \cdots = w(x_4, x_5) = h;$$
$$w(x_5, v_1) = 1 - 5h,$$

where, as mentioned above, $h > 0$ is a small perturbation parameter (in this study, $h = 0.01$). The chain of newly defined nodes, connecting v_0 and v_1, replaces the edge $e_{0,1}$ and acts as a discretization of the continuous segment in a small neighbourhood of the node v_0. We define a discretized "lifted line" $\omega(x_i) = \mathbf{f}(x_i; \mathbf{W})$ given by the mapping $\mathbf{f}$ through the new nodes $x_i, i = 1 \ldots, 5$. Given this discretization, we can approximate unit tangent vectors at $\omega(x_2)$ and $\omega(x_4)$ via central differences:

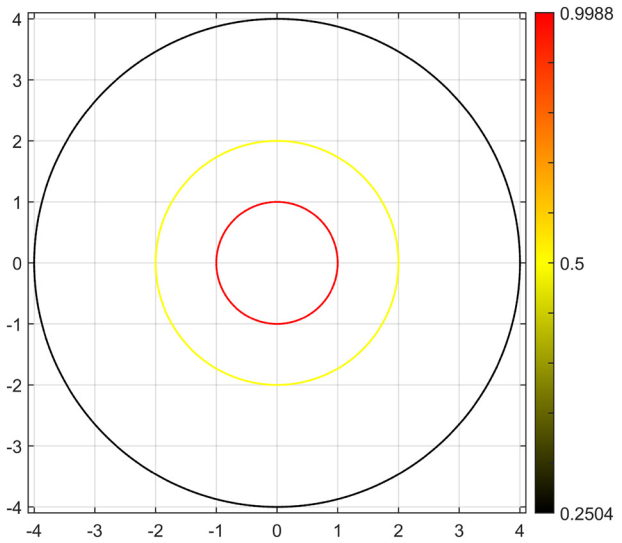


Fig. 9. Curvature computed on the recovered graphs of three concentric circles data sets. The radii of the circles are 1, 2 and 4, respectively. The curvature values for the corresponding circles are shown in the colormap. Note the relation to the inverse of the radius.

$$\vartheta(x_2) \approx \frac{\omega(x_3) - \omega(x_1)}{\|\omega(x_3) - \omega(x_1)\|} \tag{20}$$

$$\vartheta(x_4) \approx \frac{\omega(x_5) - \omega(x_3)}{\|\omega(x_5) - \omega(x_3)\|}. \tag{21}$$

The curvature at $\omega(x_3)$ can be evaluated through the difference of the unit tangent vectors at $\omega(x_2)$ and $\omega(x_4)$, relative to the geodesic distance from $\omega(x_2)$ to $\omega(x_4)$ (approximated through $\|\omega(x_3) - \omega(x_2)\| + \|\omega(x_4) - \omega(x_3)\|$):

$$\Upsilon(x_3) \approx \frac{\vartheta(x_4) - \vartheta(x_2)}{\|\omega(x_3) - \omega(x_2)\| + \|\omega(x_4) - \omega(x_3)\|},$$

which can be written as

$$\Upsilon(x_3) \approx \frac{\frac{\omega(x_3)-\omega(x_1)}{\|\omega(x_3)-\omega(x_1)\|} - \frac{\omega(x_5)-\omega(x_3)}{\|\omega(x_5)-\omega(x_3)\|}}{\|\omega(x_3) - \omega(x_2)\| + \|\omega(x_4) - \omega(x_3)\|}. \tag{22}$$

This perturbation based estimation can be regularised by observing that for sufficiently small h, the distances between the embedded neighbouring nodes, $\|\omega(x_2) - \omega(x_1)\|$, $\|\omega(x_3) - \omega(x_2)\|$, $\|\omega(x_4) - \omega(x_3)\|$ and $\|\omega(x_5) - \omega(x_4)\|$, are approximately equal and can be represented by $\tilde{h}$, and $\|\omega(x_3) - \omega(x_1)\| \approx |\omega(x_5) - \omega(x_3)\| \approx 2\tilde{h}$. We have,

$$\Upsilon(x_3) \approx \frac{\omega(x_5) - 2\omega(x_3) + \omega(x_1)}{4\tilde{h}^2}. \tag{23}$$

In practice, we calculate the mean length $\tilde{\ell}$ of edges of the abstract latent graph embedded in the data space and set $\tilde{h} = h \cdot \tilde{\ell}$. The norm $\|\Upsilon(x_3)\|$ quantifies the directional curvature of $\mathbf{f}$ in the vicinity of node v_0, along edge $e_{0,1}$.

To demonstrate the workings of our local curvature estimation we first performed a set of controlled experiments on concentric circles with known radii. We expect that the estimated curvatures will approximate the inverse radius.[7] After generating points on circles of radii $r_0 = 1, 2, 4$, we performed crawling and latent graph construction on each circle manifold. We then evaluated curvatures on all edges of the embedded graphs. The results are presented in Fig. 9. As expected, the curvatures closely follow the r_0^{-1} relation - the curvatures on circles of radii 1, 2 and 4 are approximately 1, 0.5 and 0.25, respectively.

As a further illustrative example, we consider the case of a 2-dimensional manifold $\mathcal{M}$, a "sphere with a bump", embedded in $\mathbb{R}^3$, shown in Fig. 10, left panel. Manifold $\mathcal{M}$ can be expressed in terms of a spherical and "peak" components, respectively $\mathcal{C}_S$ and $\mathcal{C}_P$. The component $\mathcal{C}_S$ is sampled by a uniform distribution $\mathcal{U}(\theta, \phi)$ over the angular components of the spherical coordinate system $(\theta, \phi) \in \mathcal{I}_\theta \times \mathcal{I}_\phi = [0, 2\pi] \times [-\pi/2, \pi/2]$ with fixed radius $r = 1$. The "peak" component is an additive term to coordinate r generated by a Gaussian distribution $\mathcal{N}(\mu_P, \mathbf{C}_P)$, where $\mu_P = (\theta_P, \phi_P) = (\pi/2, -\pi/4)$ and

[7] We are thankful to the anonymous reviewer for this suggestion.

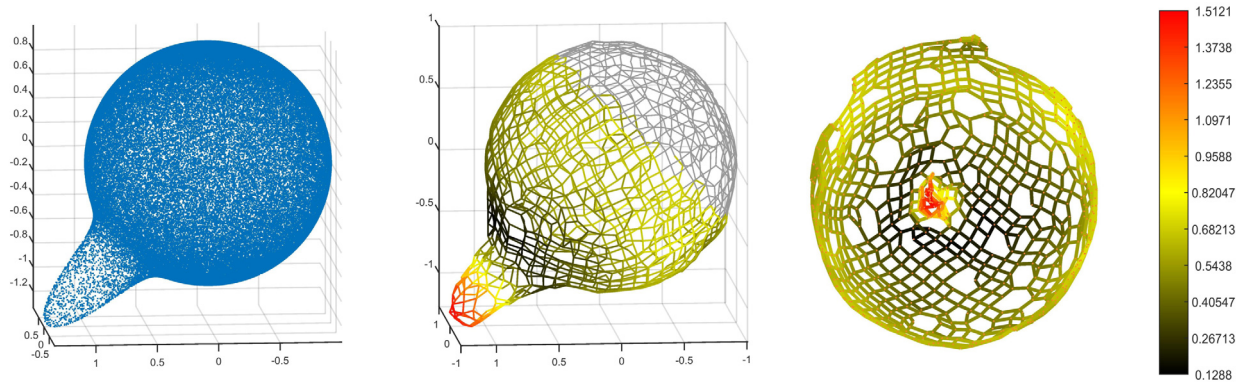


Fig. 10. Data set (left panel), extracted graph (central panel) and planar representation (right panel) for manifold $\mathcal{M}(x, y, z)$, and its on-edge curvature distribution.

covariance matrix $\mathbf{C}_P = \text{diag}(\pi/90, \pi/180)$. In other words, we define a varying radius over the angular component of the coordinates as:

$$r_{\mathcal{M}} \sim 1 + 1.5 \cdot \mathcal{N}(\mu_P, \mathbf{C}_P); \tag{24}$$

Sample around $\mathcal{M}$ is then obtained by converting the spherical coordinate system to the Cartesian coordinates x, y, z using:

$$x = r\sin\theta\cos\phi; \quad y = r\sin\theta\sin\phi; \quad z = r\cos\theta.$$

By applying manifold crawling and AGTM to this sample we recover a smooth manifold through the basis functions formulation, and compute, for every edge in the abstract latent graph a value for the curvature as defined in equation (23). Fig. 10 (central panel) presents the extracted graph where edges are colour-coded by their curvature (we only show curvature for the portion of the graph containing the peak and the adjacent spherical structure). The visualization in Fig. 10, right panel presents the same portion of the graph as in the central one, but unfolded on a plane. Here the central region captures the extreme curvature of the culminating part of the peak. The curvature decreases to a minimum where the Gaussian peak smoothly approaches the sphere's surface, but steadily converges to the constant curvature of the spherical surface in the outer edges of the planar representation. This behaviour can also be verified in the central panel and is to be expected by construction of the data set.

4. Multi manifold learning

Until now we have only considered a single noisy manifold embedded in $\mathbb{R}^d$. We now extend our method to multiple manifolds of possibly different dimensionalities, embedded in a noisy background. We will use as an illustrative example, the data set $\mathcal{Q}$ described in section 2 and shown in Fig. 2. The data is processed by removing the low-density background "noise" and producing a pair of corresponding data sets $\mathcal{Q}$ and $\tilde{\mathcal{Q}}$, as outlined in section 2. Recall that while $\tilde{\mathcal{Q}}$ contains samples of noisy manifolds diffused to their "core" or "spine", $\mathcal{Q}$ collects the original samples distributed along the manifolds. In order to build the initial skeleton model for individual manifolds by Manifold Crawling (section 3.2) and subsequent density modelling around it through Abstract GTM (section 3.1), we are missing only one piece of information: the intrinsic dimensionality of each manifold, given a characteristic scale r.

4.1. Local dimensionality estimation

Around each $\tilde{\mathbf{t}}_i \in \tilde{\mathcal{Q}}$ we perform local Principal Component Analysis (PCA) using points from $\mathcal{B}(\tilde{\mathbf{t}}_i; r) \cap \tilde{\mathcal{Q}}$, obtaining eigenspectrum $\lambda_{i,1} \geq \lambda_{i,2} \geq \ldots \geq \lambda_{i,d}$.

The dimensionality index of $\tilde{\mathbf{t}}_i \in \tilde{\mathcal{Q}}$ used in [31] (limited to 3-dimensional data and derived from [38]) was obtained as

$$\delta_i^O = \underset{j}{\text{argmax}}\, S_{i,j} \tag{25}$$

where $S_{i,1} = \lambda_{i,1} - \lambda_{i,2}$, $S_{i,2} = 2(\lambda_{i,2} - \lambda_{i,3})$ and $S_{i,3} = 3\lambda_{i,3}$.

Here, we suggest a general method for computing dimensionality index of points distributed in spaces of arbitrary finite dimension d, based on renormalized eigenvalues,

$$\tilde{\lambda}_{i,j} = \frac{\lambda_{i,j}}{\sum_{k=1}^{d} \lambda_{i,k}},$$

viewed as "likelihoods" of different dimensionalities j of the cloud of points around $\tilde{\mathbf{t}}_i$. Note that $\tilde{\Lambda}_i = (\tilde{\lambda}_{i,1}, \tilde{\lambda}_{i,2}, ..., \tilde{\lambda}_{i,d})$ lies in the $(d-1)$-dimensional simplex $\mathcal{S}_0$ with vertices $\mathbf{s}_1 = (1, 0, 0, \ldots, 0)$, $\mathbf{s}_2 = (1/2, 1/2, 0, \ldots, 0)$, $\mathbf{s}_3 = (1/3, 1/3, 1/3, \ldots, 0)$, $\ldots \mathbf{s}_d = (1/d, 1/d, \ldots, 1/d)$. The simplex $\mathcal{S}_0$ is a subset of the standard simplex $\mathcal{S}$ with vertices equal to the standard basis $\mathbf{e}_1 = (1, 0, 0, \ldots, 0)$; $\mathbf{e}_2 = (0, 1, 0, \ldots, 0)$; $\mathbf{e}_d = (0, 0, \ldots, 1)$. Considering $\mathcal{S}$ as the simplex of multinomial distributions, the appropriate Riemannian geodesic distance is the Fisher distance [39]. In particular, for any $\tilde{\Lambda}_k, \tilde{\Lambda}_l \in \mathcal{S}$,

$$d_J(\tilde{\Lambda}_k, \tilde{\Lambda}_l) = 2 \arccos\left(\sum_{i=1}^{d} \sqrt{(\tilde{\lambda}_{ki} \cdot \tilde{\lambda}_{li})} \right). \tag{26}$$

Note that the vertex $\mathbf{s}_j$ of $\mathcal{S}_0$ corresponds to the ideal normalized eigen-spectrum of a j-dimensional neighbourhood. Hence, we will quantify the degree to which the neighbourhood of $\tilde{\mathbf{t}}_i$ resembles a j-dimensional hyperplane in $\mathbb{R}^d$ by the closeness of $\tilde{\Lambda}_i$ to $\mathbf{s}_j$ in terms of the geodesic distance (26),

$$\delta_i^G = \operatorname*{argmin}_j d_J(\tilde{\Lambda}_i, \mathbf{s}_j). \tag{27}$$

We also propose a 'soft' and spatially smoothed version of dimensionality index. This is done by positioning an isotropic kernel

$$K(\alpha; \mathbf{e}_j) = \exp\left[-\frac{d_J(\alpha, \mathbf{e}_j)^2}{2\kappa^2} \right], \quad \alpha \in \mathcal{S},$$

on top of each vertex $\mathbf{e}_j$ of $\mathcal{S}$ with kernel scale κ set to the geodesic distance between the equidistant point $\mathbf{s}_d \in \mathcal{S}$ to every vertex of $\mathcal{S}$ (the 'centre' of $\mathcal{S}$), i.e. $\kappa = d_J(\mathbf{e}_j, \mathbf{s}_d)$, for any j. As explained above, due to the imposed eigenvalue order, the normalized eigen-spectra $\tilde{\Lambda}_i$ live in the sub-simplex $\mathcal{S}_0$ of the standard simplex $\mathcal{S}$. To transform $\mathcal{S}_0$ to $\mathcal{S}$, we calculate the barycentric coordinates [40] $\alpha_i = (\alpha_{i,1}, \alpha_{i,2}, ..., \alpha_{i,d})$ of $\tilde{\Lambda}_i$ by solving,

$$\tilde{\Lambda}_i = \sum_{j=1}^{d} \alpha_{i,j} \mathbf{s}_j, \quad \alpha_{i,j} \geq 0, \sum_{j=1}^{d} \alpha_{i,j} = 1.$$

The eigen-spectrum $\tilde{\Lambda}_i$ is thus mapped to the point in $\mathcal{S}$ with barycentric coordinates α_i. This leads to a soft kernel-based dimensionality distribution

$$P_i(j) = \frac{K(\alpha_i; \mathbf{e}_j)}{\sum_{k=1}^{d} K(\alpha_i; \mathbf{e}_k)}.$$

It may happen that the normalized index distribution P_i is abruptly changing across neighbouring points due to low sample density (e.g., at the manifold edges). We therefore spatially smooth P_i using a Gaussian filter[8]

$$c(i, l) = \exp[-\|\tilde{\mathbf{t}}_i - \tilde{\mathbf{t}}_l\|^2 / (2r^2)].$$

The smoothed normalized index distribution reads:

$$P_i^S(j) = \frac{1}{\sum_{\tilde{\mathbf{t}}_l \in \mathcal{B}(\tilde{\mathbf{t}}_i, r)} c(i, l)} \sum_{\tilde{\mathbf{t}}_l \in \mathcal{B}(\tilde{\mathbf{t}}_i, r)} c(i, l) \cdot P_l(j), \tag{28}$$

where the sum is taken over diffused points $\tilde{\mathbf{t}}_l$ in the spherical neighbourhood of $\tilde{\mathbf{t}}_i$ of radius r. The smoothed dimensionality index of $\tilde{\mathbf{t}}_i$ is then

$$\delta_i^S = \operatorname*{argmax}_j P_i^S(j). \tag{29}$$

Fig. 11 shows the results obtained using dimensionality indices δ^G (top row) and δ^S (bottom row). Each column presents the results of the two indices on the toy data set. The first column shows the 1- (red), 2- (blue) and 3-dimensional (green) eigenvalue-related distributions on the simplices $\mathcal{S}_0$ and $\mathcal{S}$ for δ_i^G and δ_i^S, respectively, for all points in the data set (after removing the low-density background noise). Central column shows points from $\mathcal{Q}$ with dimensionality index equal to 1 (red) and 3 (green). Points with dimensionality index 2 (blue) are shown in the last column. The central column clearly shows the beneficial effect of spatial smoothing of the dimensionality index. In both columns we also show a small ball of radius $r = 0.1$, the value of r used as the characteristic scale in local manifold linearisations employed in dimensionality index calculations and manifold crawling on this data set.

[8] r is the radius used for the local PCA estimation.

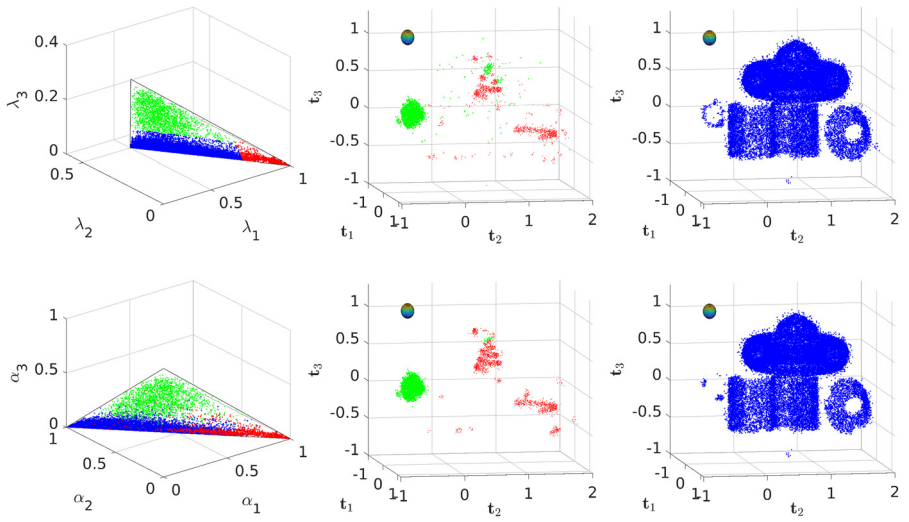


Fig. 11. Eigenvalue distribution on simplex and dataset labelled using the dimensionality index δ^G (upper row) and δ^S (lower row).

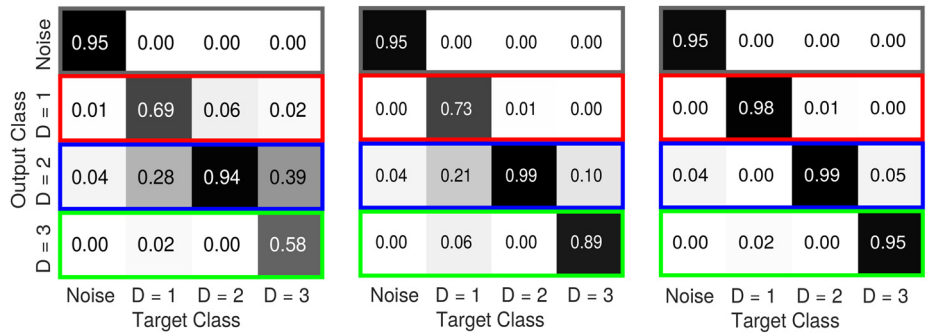


Fig. 12. Confusion matrices for dimensionality indices δ_i^O (left), δ_i^G (centre) and δ_i^S (right). The superior performance of δ_i^S is confirmed in Fig. 11.

Knowing a-priori the true dimensionality of each point, we provide a qualitative analysis of the different dimensionality index formulations in the form of confusion tables shown in Fig. 12. The confusion tables present percentages of correctly and incorrectly classified points depending on their true dimensionality and their corresponding dimensionality indices. We have also included the "Noise" class containing points generated as the lower density background noise and supposed to be filtered out in the initial pre-processing step outlined in section 2. Left, middle and right panels show confusion matrices of indices δ_i^O, δ_i^G and δ_i^S, respectively. Compared with δ_i^O, reformulation of dimensionality index as proximity of renormalized eigen-spectrum to the "ideal" eigen-spectrum for each dimensionality $1 \leq j \leq d$ (index δ_i^G) stabilizes the estimation. The index is further stabilized by spatial smoothing of the soft version of δ_i^G, provided by index δ_i^S.

4.2. Multi manifold crawling

Having obtained dimensionality index δ_i^S for each point $\tilde{\mathbf{t}}_i \in \tilde{\mathcal{Q}}$ (and hence also for the corresponding point $\mathbf{t}_i \in \mathcal{Q}$), we perform a partition of the two data sets according to their dimensionality indices, $\mathcal{Q}_j = \{\mathbf{t}_i \in \mathcal{Q} |\ \delta_i^S = j\}$ and $\tilde{\mathcal{Q}}_j = \{\tilde{\mathbf{t}}_i \in \tilde{\mathcal{Q}} |\ \delta_i^S = j\}$. For a given $1 \leq j < d$, the data sets $\mathcal{Q}_j$ and $\tilde{\mathcal{Q}}_j$ contain the noisy and "crisp" samples, respectively, along manifolds of intrinsic dimension j.

While the Manifold Crawling algorithm of section 3.2 can be used to crawl and assist model building of a single manifold, the method can be extended to crawl and isolate multiple manifolds of a given dimensionality j. We initialize the "to be processed" data set $\mathcal{R}$ as $\tilde{\mathcal{Q}}_j$. Then the Manifold Crawling algorithm is run on $\mathcal{R}$ with a scale parameter r. This results in graphs $\mathcal{G}^1$ and $\overline{\mathcal{G}}^1$, forming skeletons of individual manifolds of dimensionality j. Here we assume that the separation between manifolds in $\tilde{\mathcal{Q}}$ is greater than r.

For every node $\overline{\mathbf{v}}_l$ of $\overline{\mathcal{G}}^1$ we compute its neighbourhood in $\mathcal{R}$, $\overline{\mathcal{O}}_l = \mathcal{B}(\overline{\mathbf{v}}_l, r) \cap \mathcal{R}$. The neighbourhoods of all nodes of $\overline{\mathcal{G}}^1$, $\bigcup_l \overline{\mathcal{O}}_l$, are then subtracted from $\mathcal{R}$. At the same time, by computing $\mathcal{O}_l = \mathcal{B}(\overline{\mathbf{v}}_l, r) \cap \mathcal{Q}_j$ for every $\overline{\mathbf{v}}_l \in \overline{\mathcal{V}}^1$, we recover "noisy" data set $\mathcal{A}^1 = \bigcup_l \mathcal{O}_l$ aligned along the 1-st manifold of dimensionality j.

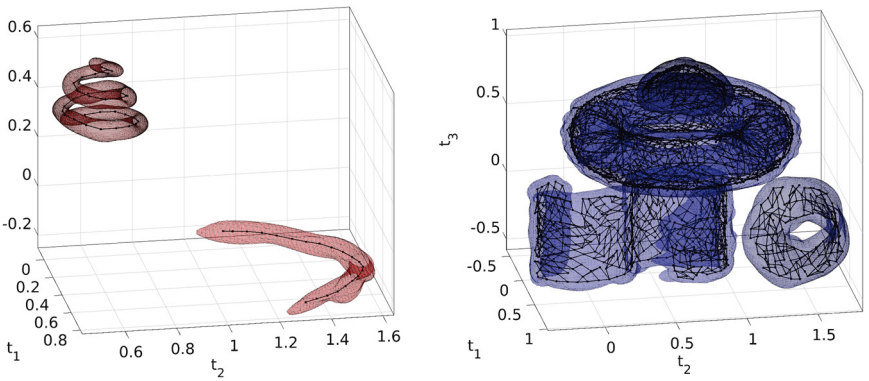


Fig. 13. Both panels show the probabilistic model of each manifold in data set $\mathcal{D}$, recovered through AGTM, highlighted as an isosurface of the respective Probability Density Function. The black skeletons in each panel are the embedded graphs, as obtained after MultiManifold Crawling and training of AGTM.

The manifolds sampled in $\tilde{\mathcal{Q}}_j$ could be separated by larger distances, so that the previous crawling run may not have reached them. Therefore, while $\mathcal{R} \neq \emptyset$, another seed is picked at random from $\mathcal{R}$ and a new crawling procedure is initiated. This produces graphs $\overline{\mathcal{G}}^2$ and $\mathcal{G}^2$, accompanied by the corresponding manifold-aligned data $\mathcal{A}^2$. After reduction of $\mathcal{R}$ as described above, while $\mathcal{R} \neq \emptyset$, the whole process is repeated.

Finally, assuming the processing of $\tilde{\mathcal{Q}}_j$ took H runs, the procedure results in a collection of H extracted j-dimensional manifolds represented by the graphs $\left\{\overline{\mathcal{G}}^\ell\right\}_{\ell=1}^H$, $\left\{\mathcal{G}^\ell\right\}_{\ell=1}^H$ and the associated manifold specific data sets $\left\{\mathcal{A}^\ell\right\}_{\ell=1}^H$. The individual graphs $\overline{\mathcal{G}}^\ell$, $\mathcal{G}^\ell$ and data sets $\mathcal{A}^\ell$ can now be used by the AGTM algorithm (section 3.1) to construct the manifold-aligned density models for the manifolds of intrinsic dimensionality j.

To guarantee smoothness of the AGTM density estimates we conservatively set the ϵ-net parameter for basis function positioning in the abstract graphs (see section 3.1) to $\epsilon = 1$. Due to relatively modest curvature in the low-dimensional structures, the crawling step size parameter η (see section 3.2) was set to $\eta = 0.75$. The tolerance scale parameter in the contraction phase of the crawling procedure was set to $\beta = 0.4$. As mentioned earlier, we used $r = 0.1$ for the characteristic scale parameter. Fig. 13 shows the embedded graphs $\overline{\mathcal{G}}^\ell$ (black lines) and overlaid AGTMs (red and blue pdf isosurfaces) of manifolds in the example data set described in section 2, plotted by intrinsic dimensionality of the corresponding manifolds ($j = 1$, left; $j = 2$, right). The results were insensitive to small variations in the parameters r, η and β. Larger values of η can cause "overshooting" of highly curved local manifold structures, while smaller values of η are harmless, modulo increased computational complexity. Likewise, larger values of β can result in neglecting important details of the manifold structure, while smaller values are harmless, but can lead to more mixture components in the final probabilistic model. The most important parameter is the characteristic scale r governing local manifold linearisations used in determination of dimensionality indices and manifold crawling. This can be set based on prior knowledge, or after inspection of the data set. Alternatively, the multimanifold learning method can be used as a semi-automatic exploratory tool by the domain experts, where the focus and characteristic scale r of the structures to be mined can be varied continuously, starting from capturing coarse structures in the data (larger r), and proceeding with more detailed analysis, as potentially more and more low-dimensional manifolds start to appear with decreasing r.

4.3. An efficient alternative to multi-manifold crawling

One can approximate the results of Multi-manifold crawling by constructing an ϵ-neighbourhood graph on specific samples of the data set $\tilde{\mathbf{Q}}_j$, with ϵ set to r, the radius used for local PCA in the crawling algorithm.

We first cover $\tilde{\mathbf{Q}}_j$ via L hyper-balls $\mathcal{B}(\tilde{\mathbf{t}}_k, c)$, $k = 1, 2, ..., L$, of radius $c = r/2$, centred at $\tilde{\mathbf{t}}_k \in \tilde{\mathbf{Q}}_j$ located using the recursive algorithm of [20]. Hence, $\tilde{\mathbf{Q}}_j = \cup_{k=1}^L \mathcal{S}_k$, where $\mathcal{S}_k \subseteq \mathcal{B}(\tilde{\mathbf{t}}_k, c) \cap \tilde{\mathbf{Q}}_j$. For every set $\mathcal{S}_k$, we define its centre $\mathbf{s}_k$ as its sample mean, obtaining the set of centres $\mathcal{S} = \{\mathbf{s}_1, \ldots, \mathbf{s}_L\}$. By design, we have for the minimum Euclidean distance between any pair of centres $\mathbf{s}_m, \mathbf{s}_n \in \mathcal{S}$, $m \neq n$:

$$\frac{r}{2} < \min_{m \neq n} d_E(\mathbf{s}_m, \mathbf{s}_n) < r.$$

We now compute the distance between all centres obtaining a square, symmetric matrix $D_{m,n} = d_E(\mathbf{s}_m, \mathbf{s}_n), \forall m \neq n, m, n = 1, \ldots, L$, from which we can recover an initial estimate of the adjacency matrix $\mathbf{A}$ of graph $\overline{\mathcal{G}}$. The adjacency matrix has non-zero elements only for elements of $\mathbf{D}$ smaller than r:

$$A_{m,n} = \begin{cases} 1 & \text{if } D_{m,n} \leq r; \\ 0 & \text{Otherwise.} \end{cases} \tag{30}$$

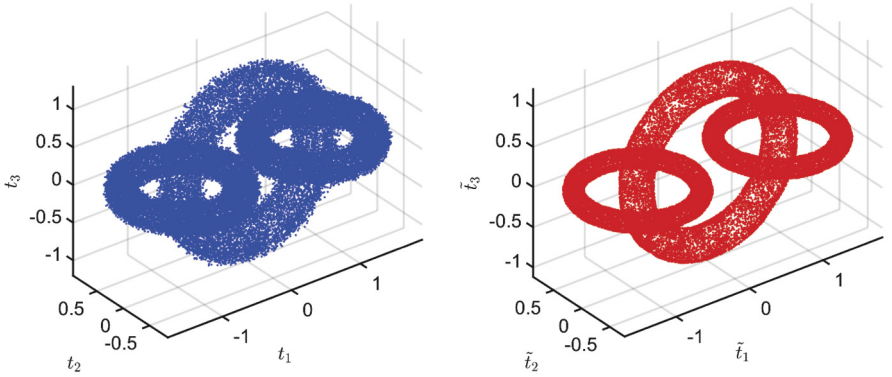


Fig. 14. Data set $\mathcal{D}^2$ with the three interlocked noisy toroids (right panel) and its diffused and filtered counterpart $\tilde{\mathcal{D}}^2$ (left panel).

This (initial) adjacency matrix $\mathbf{A}$ tends to be overly connected, because it does not take into consideration the alignment of the tangent spaces of neighbouring partition centres. In order to include this geometrical information in $\mathbf{A}$, for every set $\mathcal{S}_k$ we compute its eigen-decomposition and approximate the local tangent space to the unknown manifold $\mathcal{M}_\ell$ at centre $\mathbf{s}_k$ by the first j eigenvectors $\mathbf{u}_1, \ldots, \mathbf{u}_j$ collected in matrix $\mathbf{U}_k = [\mathbf{u}_1, \ldots, \mathbf{u}_j]$. For each pair of connected points (whose corresponding entry in the initial adjacency matrix is 1), we compute the angle between the corresponding tangent spaces [41] as $|\Theta_{m,n}| = \arccos(|\det \mathbf{M}_{m,n}|)$, where $\mathbf{M}_{m,n} = \mathbf{U}_m^\top \mathbf{U}_n \in \mathbb{R}^{j \times j}$. We retain the existing edge only if the angle between the connected points is $|\Theta_{m,n}| \leq \pi/3$. Alternatively, one could define an additional hyper-parameter $0 < T_\theta < 1$ so that smaller or larger angles are tolerated in the construction of the graph. The condition on the final adjacency matrix would then be:

$$A_{m,n} = \begin{cases} 1 & \text{if } D_{m,n} \leq r \ \ \& \ \ |\Theta_{m,n}| < T_\theta \cdot \pi/2; \\ 0 & \text{Otherwise.} \end{cases} \tag{31}$$

In our case, we simply imposed $T_\theta = \frac{2}{3}$.

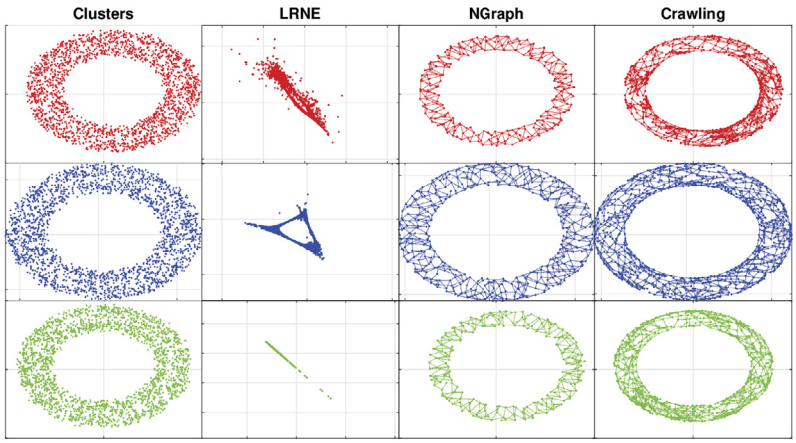
Having constructed the adjacency matrix $\mathbf{A}$ for a set $\tilde{\mathcal{Q}}_j$, we split the graph $\overline{\mathcal{G}}$ corresponding to $\mathbf{A}$ into its maximal connected components $\mathcal{K}_c = (\overline{\mathcal{V}}_c, \overline{\mathcal{E}}_c)$, $c = 1, \ldots, N_G$, Here, $\overline{\mathcal{V}}_c$ and $\overline{\mathcal{E}}_c$ are the sets of centres (nodes) and edges, respectively, of the component $\mathcal{K}_c$. For each component we define the corresponding abstract graph as $\mathcal{K}_c = (\mathcal{V}_c, \mathcal{E}_c)$ as in the previous section. We can now use the N_G extracted pairs of embedded and abstract graphs as initialization to AGTM, as an alternative to the Multi-manifold crawling initialization presented in the last section.

5. Experimental comparison to existing methods

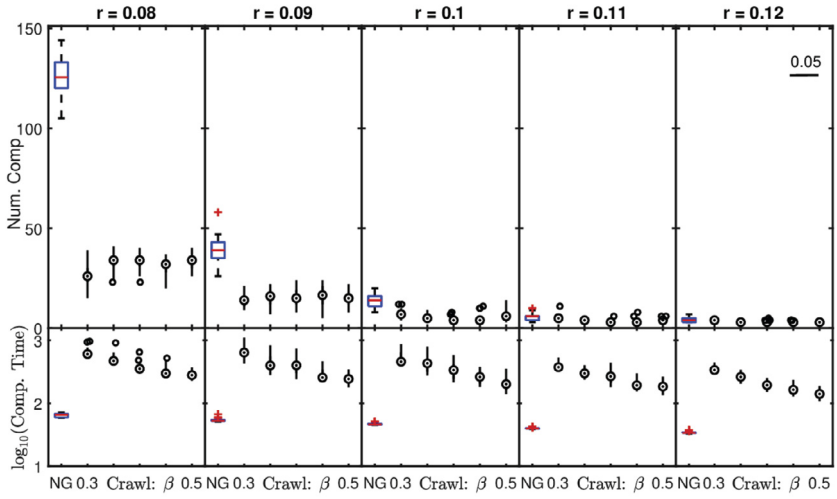
While multi-manifold AGTM covers both multi-manifold learning and probabilistic modelling simultaneously, most methodologies in the literature focus on one or the other aspect of capturing spatial structures in point clouds. We therefore split the evaluation of AGTM in comparative experiments along two lines: (1) recovery of a collection of smooth manifolds of possibly different dimensionalities and (2) full probabilistic modelling of the data distribution. We will use two main data sets. The first one is data set $\mathcal{D}^1 = \mathcal{Q}$ presented in section 2 composed of six sampled manifolds of different dimensionalities embedded in $\mathbb{R}^3$. We recall the presence of two non-linear 1-dimensional (curved spiral and hyperbolic arm) and four 2-dimensional manifolds (toroid, Möbius strip, parabolic cap, S-shape). The second data set $\mathcal{D}^2$ consists of three interlocked noisy toroids (Fig. 14). Both data sets are embedded in uniformly distributed noise over the domain. All methods for this comparison are applied to the data sets after the initial filtering methodology (section 2), obtaining both a diffused $\tilde{\mathcal{D}}^i$ and noisy $\mathcal{D}^i$ version of each data set.

5.1. Multi-manifold learning

In this section we will compare manifold crawling with the alternative efficient graph construction presented in section 4.3, as well as with two main algorithms found in the literature, for which the codes are publicly available: *Sparse Manifold Clustering and Embedding* (SMCE) [42] and *Low Rank Neighbourhood Embedding* (LNRE) [43]. We test the four methodologies on data sets $\tilde{\mathcal{D}}^1$ and $\tilde{\mathcal{D}}^2$ in order to verify their ability to deal with multiple manifolds of different dimensionalities and topologies. Both SMCE and LRNE are designed to cluster a mixture of low-dimensional manifolds embedded in a higher dimensional space, while recovering a low-dimensional embedding for each manifold. In both methods the number of manifolds in the data set has to be provided by the user. In contrast, Multi-manifold crawling and its alternative graph construction recover the number of manifolds in the data set automatically, once the characteristic size of the manifolds



(a)



(b)

Fig. 15. Comparison of embeddings obtained for data set $\mathcal{D}^2$ via LRNE (second column) Multi-Manifold Crawling (rightmost column) and its ϵ-mesh alternative (third column). The clusters shown in the leftmost column are equally recovered by the three methods (panel a). Panel b shows the number of components (top row) and computational time (bottom row) for Multi-Manifold Crawling (black boxplots) and its ϵ-mesh alternative (blue boxplots, identified by label "NG") evaluated over 30 samplings of synthetic data set $\mathcal{D}^2$.

is provided. Fig. 15a, presents a qualitative comparison[9] of the embeddings recovered by LRNE (second column), the efficient alternative to Manifold Crawling (third column) and Manifold Crawling (fourth column) on data sets $\tilde{\mathcal{D}}^2$. While the clustering performance is similar for all methods (first row), the embeddings recovered by LRNE do not explicitly display the properties of the three toroids. The graphs recovered both by Manifold Crawling and its alternative graph construction provide much more insight into the topological structures underlying the sampled noisy manifolds. We have also tested the stability of Multi-manifold crawling and the ϵ-graph alternative with respect to (hyper-)parameter settings. To that end, we evaluated the number of recovered manifolds over a range of hyper-parameter values (Fig. 15b, upper row). The blue box-plots show the statistics over the number of manifolds recovered by the ϵ-graph construction, for 30 different samplings of data set $\mathcal{D}^2$ at a given radius. The black compact box-plots[10] show the same measure for 30 iterations of Multi-Manifold crawling over the same data set at the same radius, for different values of the parameter β. Not surprisingly, Multi-manifold crawling displays a higher stability than its alternative, over a range of values of the parameter r, while little change is

[9] Parameters of each method were tuned to optimize performance on $\tilde{\mathcal{D}}^2$.

[10] For visualization purposes, the black compact boxplots are shown as hollow dots for the means and black vertical lines for the variances, plus smaller black dots for outliers were present.

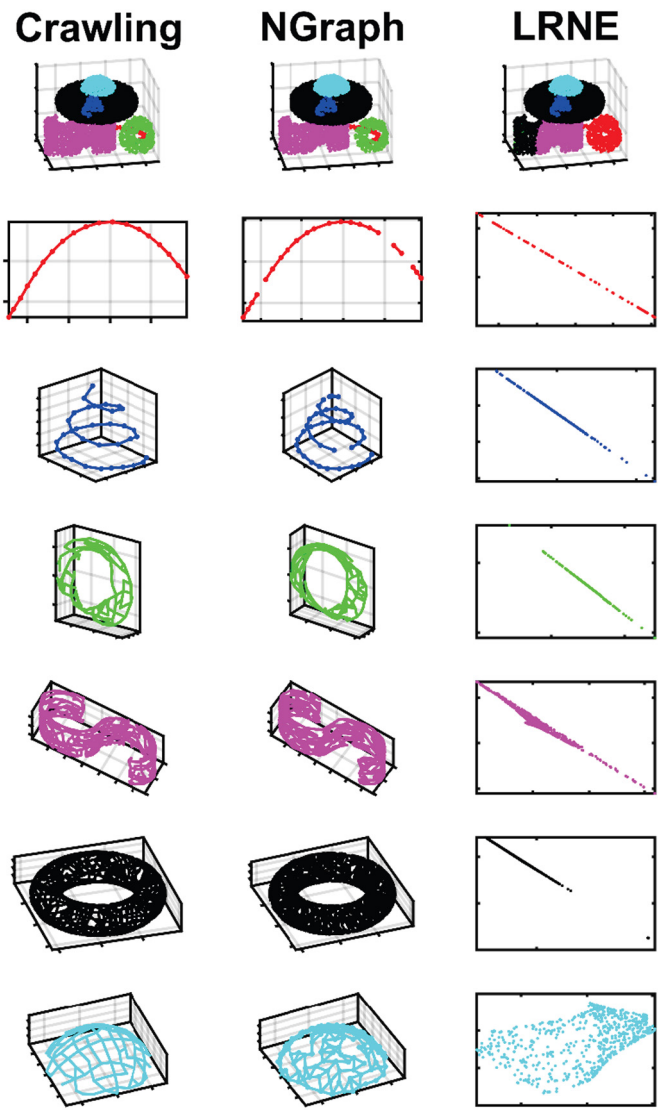


Fig. 16. Same as Fig. 15a, but for data set $\mathcal{D}^1$.

due to the parameter β for a given radius. While both methods converge to the correct number of manifolds (three in this case) starting at $r = 0.1$, the efficient ϵ-graph method vastly overestimates the number of manifolds when the radius is too small. The computational times are shown in Fig. 15b, lower row. This discrepancy in the quality of the results stems from the imposition, in the alternative methodology to crawling, of a "hard" angular threshold that controls variability of the angles between neighbouring tangent spaces. Even a slight deviation from this threshold may prevent two nodes belonging to the same manifold from being connected. The efficient alternative critically relies on angle relations between local PCA subspaces. The subspace decompositions become increasingly unreliable with decreasing radius, while no such reliance on subspace angles is needed in the crawling procedure.

The same comparison is shown in Figs. 16 and 17 for data set $\mathcal{D}^1$. For both Crawling and its approximation, the clusters are obtained by applying the procedure outlined in section 4.2 and 4.3 respectively. While in the case of data set $\mathcal{D}^2$, LRNE was able to successfully recover the manifolds for a set of carefully tuned parameter configurations, the results proved unsatisfactory when applied to data set $\mathcal{D}^1$. In particular, no parameter setting was able to distinguish between the Möbius strip and the parabolic arm, as shown in Fig. 16, top right panel, red points. All the resulting embeddings (Fig. 16, rightmost column), with the exception for the hyperbolic surface (cyan points, bottom panel) and the spiral (third top panel, blue points), fail in capturing the low-dimensional structure of the corresponding manifold.

The first two columns of Fig. 16 illustrate the performance of Multi-manifold crawling and the ϵ-graph alternative on $\mathcal{D}^1$. Although the representation of the 2-dimensional manifolds is coherent over the two methodologies, the ϵ-graph tends to break prematurely the 1-dimensional structures into smaller segments. This behaviour is confirmed over various values for the radius, as shown in Fig. 17, top row. Here, in contrast to the data set $\mathcal{D}^2$, convergence to the correct number of

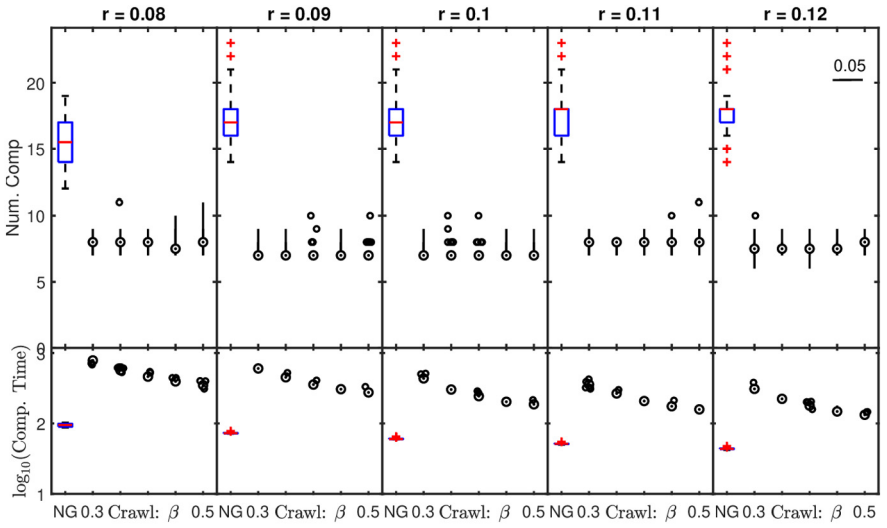


Fig. 17. Same as Fig. 15b, but for data set $\mathcal{D}^1$.

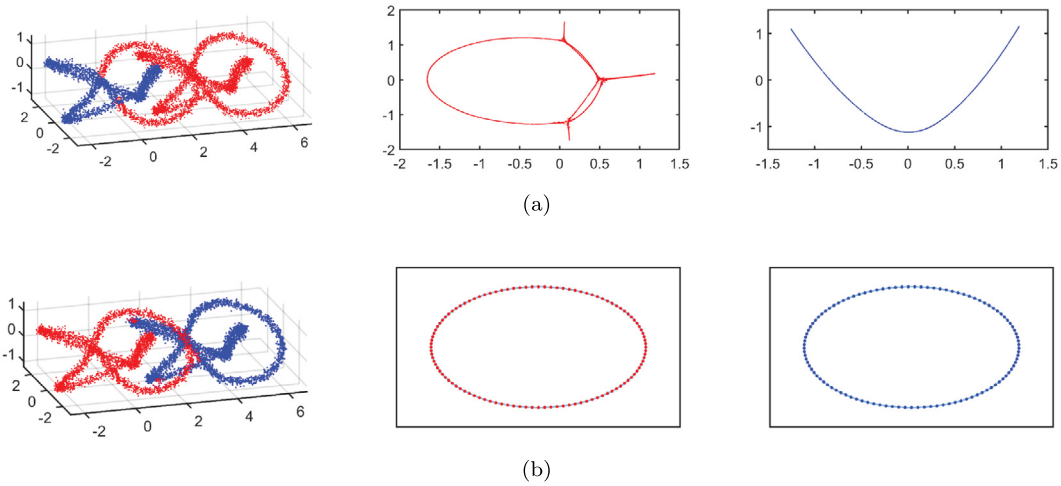


Fig. 18. Top row shows the results obtained with SMCE on the data set described in [42], while bottom row shows the results obtained with Multi-Manifold Crawling. From left to right: two obtained clusters, embedding (bottom: graph) of the first cluster and embedding (bottom: graph) of the second cluster.

Table 2
Parameters used for all Manifold Learning methods in this analysis for both data sets $\mathcal{D}^1$ and $\mathcal{D}^2$.

Methods	Parameters	
ϵ-graph	$r \in [0.08, 0.09, \ldots, 0.17]$	
Crawling	$r \in \{0.08, 0.09, \ldots, 0.17\}$	$\beta \in \{0.3, 0.35, \ldots, 0.5\}$
LRNE	$k \in \{25, 30, \ldots, 50\}$	$l_{LRNE} \in \{1e-3, 5e-3, 1e-2, \ldots, 10\}$
SMCE	$k \in \{30, 35, 40, 45, 50\}$	$l_{SMCE} \in \{1e-3, 1e-2, \ldots, 1e2\}$

manifolds is achieved only by Multi-Manifold Crawling, while the efficient ϵ-graph alternative, consistently overestimates this measure. This clearly demonstrates the enhanced accuracy and stability of Multi-manifold crawling, of course, at the price of a higher computational time (Fig. 17, bottom row).

The application of SMCE with the set of parameters in Table 2 has proven unsuccessful on both data sets $\mathcal{D}^1$ and $\mathcal{D}^2$. In order to perform at least a qualitative comparison with our methodology, we applied manifold crawling to the data set described in [42]. The data set consists of two noise-less, 3-foiled interlocked knots. The application of both methods on the presented data set proved equally satisfying for various parameter settings. However, when introducing a small amount on noise along the manifold, the performance of SMCE greatly deteriorates, both in terms of clustering accuracy and quality of the embedding (Fig. 18a, top row). The results do not improve when the data set is diffused and filtered beforehand, for

any parameter setting. In contrast, Manifold crawling recovers the underlying ground truth by building two 1-dimensional, closed graphs and clustering the noisy data set accordingly (Fig. 18b).

5.2. Probabilistic modelling

In the last subsection we evaluated the construction of embedded latent space of the AGTM model as a manifold detection and approximation method, specifically designed to handle general low-dimensional manifolds embedded in finite dimensional Euclidean spaces. In this section we will treat AGTM (section 3) as a generative probabilistic model of data aligned along such (noisy) manifolds. As explained above, we needed to split the model evaluation into those two distinct strands as up to our best knowledge we could not find a competitor that would naturally span general manifold learning and probabilistic modelling.

We will compare AGTM (initialized with Manifold crawling or the efficient ϵ-graph alternative) with two probabilistic modelling techniques, namely *Fast Parzen Window* (FPW, [20]) and another unsupervised algorithm for learning *finite mixture models from multivariate data* [44]. For both data sets $\mathcal{D}^1$ and $\mathcal{D}^2$ we apply a 5-fold cross-validation scheme. Each method (except initialization via Multi-manifold crawling) is applied thirty times on training folds for a given parameter setting and the average log-likelihood per point of the generated model is evaluated on the corresponding hold-out fold. The Multi-manifold crawling initialization is repeated five times on each training set, each run with a different β parameter. The results for the two data sets, multiple manifolds of different dimensionalities ($\mathcal{D}^1$) and three interlocked toroids ($\mathcal{D}^2$), are presented in Figs. 19b and 19a, respectively. Top row in each plot shows the average log-likelihood for different parameters as a boxplot over the initialization repetitions. Bottom row shows the computational time required for training of each model. The performance of FPW is generally higher and faster than all other methods, although this is a natural consequence of it being an efficient unconstrained probabilistic modelling algorithm. The manifold structure was often lost in the set of mixture components. However, for both data sets, AGTM initialized with Multi-Manifold crawling is comparable to FPW and outperforms both its alternative initialization and the algorithm proposed in [44] in terms of average log-likelihood. It is also worth noting that the computational time of AGTM is comparable to, if not lower than, the finite mixture method of [44]. In conclusion, AGTM provides faithful probabilistic model of the data, while providing a clear interpretation of the low-dimensional general manifold structures along which the data is organized.

6. Experiments on a jellyfish galaxy

We will demonstrate our methodology on the analysis of formation of a curious astronomical object, “jellyfish galaxy”, through a detailed astrophysical simulation of its formation. The term “jellyfish galaxy” refers to an observed galaxy showing signs of gas stripping [45], whose signatures are a dense “head” of mainly gas and stars and an elongated gaseous, star-forming tail. These galaxies are usually observed when falling into large clusters of other galaxies, where the hot ionized gas filling the cluster is able to strip away “tentacles” of relatively cold gas from the galactic body.

The technique described above can be a valuable tool to investigate the behaviour of physical quantities in the head and in the gaseous tail. We will focus our study on identifying star formation regions in the head and in the tail of the galaxy for two main reasons:

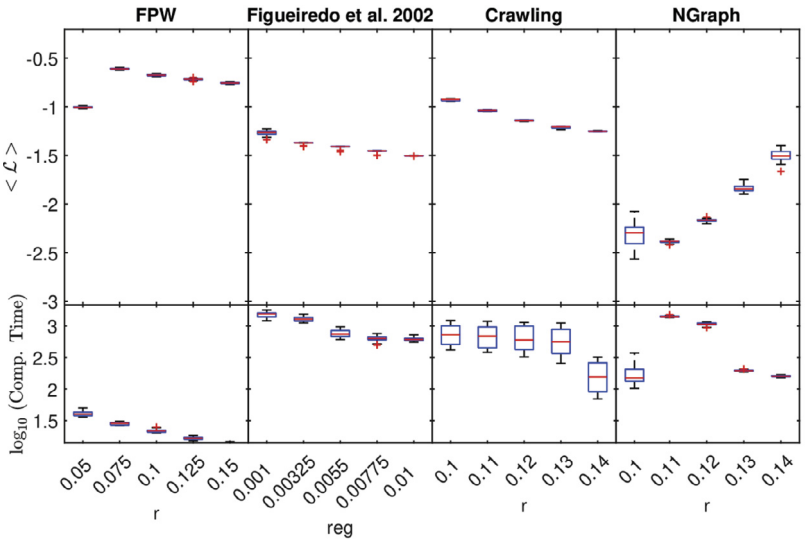
i) presence of new stars born in the head is an important indicator of how much the galactic gas is affected by the stripping pressure;
ii) since stars are created from dense and cold regions of gas, the presence (or lack thereof) of stars formed in the gaseous tail carries information about how much the galaxy gas is mixed with the hot gas in the surrounding environment.

To quantify the star formation we will measure the intensity of the [C II] emission line. Recent observations with the Herschel Space Observatory showed a tight correlation between the intensity of [C II] and other well known tracers of Star Formation Rate (SFR) [46,47].

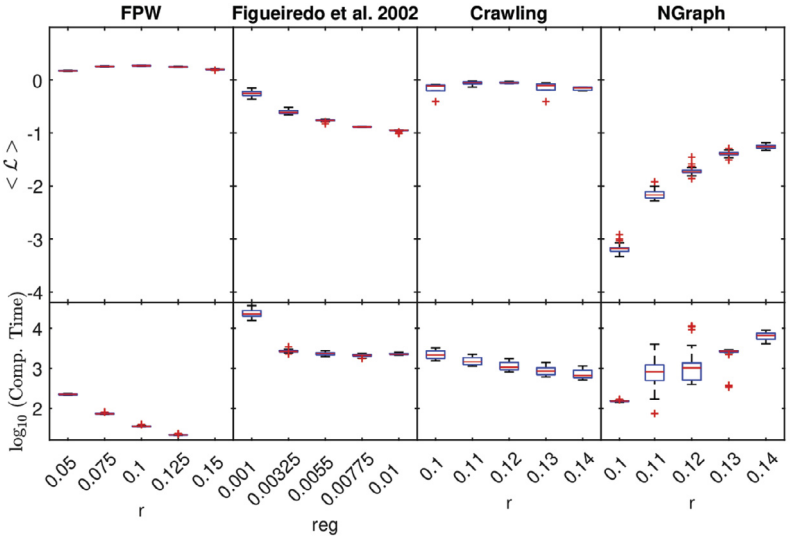
The study can provide useful insights on the formation scenario of galaxies infalling in a cluster [48]. As an example, dwarf galaxy NGC 1427A in the Fornax cluster provides an interesting case of still unclear formation scenario and a generally accepted common interpretation is still lacking [49–51].

Starting from an existing suite of simulations of dwarf galaxies evolving in a Fornax-like cluster environment [51], we chose a single simulated snapshot representing an irregular, gas rich galaxy, exposing an elongated star forming gaseous tail during intense ram pressure stripping. The simulation is performed using a modified version of the mixed N-body/Smoothed Particle Hydrodynamics (SPH) code GADGET-2 [52,53].

In order to simulate the evolution of the galaxy with high resolution and to reduce the computational cost of simulating the whole galaxy cluster, we take advantage of the *moving box* technique [54]. In this formulation, we follow closely the evolution of the galaxy by immersing a cubic volume (“box”) enveloping the galaxy, in the gravitational potential of the galaxy cluster. The gravitational potential is modelled via a spherically symmetric, static, Navarro-Frenk-White profile (NFW, [55]). The box position and orientation is updated at every time-step so that its x-axis is always tangential to the orbit of the galaxy at every moment. Also at each time-step, new gas particles are injected into the simulation box from its open “front”



(a)



(b)

Fig. 19. Panels a and b present the average log-likelihood per point of the probabilistic models generated by the four methodologies with different parameter settings (top row) and the computational time required to train the models (bottom row) for data sets $\mathcal{D}^2$ and $\mathcal{D}^1$ respectively.

face and existing ones removed when close to the opposite side of the box.[11] This injection-ejection technique is used to simulate the motion of the galaxy through the cluster's gas and it is crucial in studying processes such as gas stripping. The density and temperature of the injected particles change according to the position of the box within the cluster, following the radial profiles described in [56] and assuming hydrostatic equilibrium of the gas.

This setup allows us to study the evolution of the galaxy in high detail, while being consistent with the characteristics of the ambient space (physical properties of the galaxy cluster) at a comparatively lower computational cost.

The computation model in SPH simulations is based on a particle formulation of hydrodynamics where each particle samples physical properties (mass, temperature, density etc.) of a volume of radius r_N - radius of the sphere containing N neighbouring particles (smoothing length). A continuous distribution of the physical variables over the full domain is then obtained by spatial smoothing with Gaussian kernels centred on each particle [57]. Associated with each gas particle (representing the corresponding volume) are values of physical quantities such as density, temperature, pressure etc.

[11] The particles "falling out of the box".

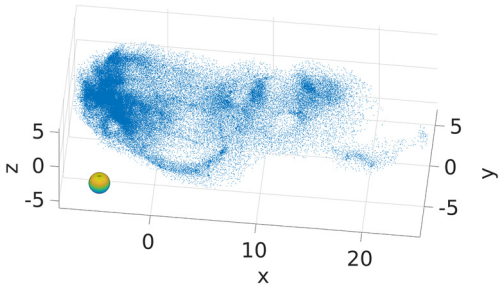


Fig. 20. Gas particles of the simulated dwarf galaxy falling into the halo of the Fornax Galaxy Cluster.

An estimate of [C II] emission in this kind of simulations is obtained by using evolved quantities of the gas (metallicity, density and temperature) as inputs of chemical evolution models of the radiating gas, taking into account its ionization equilibrium and ion level occupation model [58,53]. Here we focus on the state of the galaxy after 2.5 Gyrs of evolution since its injection in the galaxy cluster. The data set consists of 135530 gas particles (observations), each having 11 observed variables[12] (dimensions): r_N (smoothing length); x-, y-, z-coordinates, representing the position in the moving box; v_x, v_y, v_z, the velocities along the three axis of the box; m, ρ, T and [C II]: the mass, density, temperature and [C II] emission for the volume of gas sampled by the particle.

Since the particles, by construction of the SPH methodology, fill the whole volume of the moving box, we first apply a physically motivated filtering by removing all particles whose density is $\rho \leq 10^{-3}$ atom/cm^3. This is a lower limit in density for the particles to be able to emit in the radio band, and so to be observed as part of the galaxy.

After this initial filtering, the remaining data set consists of 83772 particles. Their distribution over the box's volume is shown in Fig. 20.

Several low dimensional structures are clearly visible, for example the long 1-dimensional manifold departing from the head and elongating along the x-axis of the simulation box. Visually inspecting the data set, we chose a radius $r = 1$ kpc as the characteristic scale parameter for the manifolds (shown as the small sphere in Fig. 20). The chosen radius is in agreement with the spatial resolution of recent observations for galaxies as distant as NGC 1427A. This parameter is fixed for preprocessing through diffusion and filtering (section 2), local dimensionality estimation (section 4.1) and manifold crawling (sections 3.2 and 4.2). The other parameters are set as in the synthetic data experiments of section 4.2: $\epsilon = 1$, $\eta = 0.75$ and $\beta = 0.4$.

As mentioned earlier, the main body of the data set can be visually divided into head and tail parts. However, from the topological standpoint there is no justification for clear segregation into 1-D manifolds in the tail and 2-D manifolds in the head. Dimensionality index estimation clearly identified 1-D structures in the tail, such as the elongated stream of particles starting from the head. However, points in the head were also predominantly identified as[13] 1-D, due to its complicated, intertwined filamentary structure. On the other hand, distribution of 2-D points was more localized in the main body of the tail.

6.1. A multi manifold analysis of a dwarf jellyfish galaxy

We evaluate the performance of AGTM with both its initializations and compare it with the probabilistic modelling methods introduced above using 5-fold cross-validation scheme on the 1- and 2-dimensional gas particles. The average out-of-sample log-likelihood per point and computational times are reported in Fig. 22 for all methods and their parameter settings. From the probabilistic modelling standpoint, all methods perform similarly, with FPW providing for the best score. Both initializations of AGTM are comparable to the results obtained by the method described in [44], with lower computational effort. Initialization by crawling also shows a high stability of the results when compared to both its alternative initialization and the competing methodology. The ϵ-graph initialization, despite its larger sensitivity to the randomness of the sampling process, reaches higher values of the average log-likelihood than the initialization obtained via crawling. This behaviour is explained by the tendency of the former to break manifolds into sub-structures. Having more structures to cover the same spatial region, leads to a better performance of the probabilistic modelling algorithm. However, the information about low-dimensional structures retained by the sub-structures, gradually deteriorates as this phenomenon occurs more frequently. The limiting case of this information loss is represented by FPW, where the distribution of the data set is globally captured with a high efficiency, but no additional information on the low-dimensional local structures can be inferred. Having obtained the multi-manifold probabilistic profile of the gas particles in the tail of the jellyfish galaxy, it is possible to perform various kinds of detailed analysis of how physical properties vary along the manifolds. Here we concentrate on the curvature (Fig. 23), as defined in section 3.2.1, and the star formation potential by analysing the behaviour of emission line [C II] over the 1-D and 2-D structures in the jellyfish tail shown as black dots in Fig. 21 (left) and (right),

[12] The number of observed variables is larger than the one presented here. We decided to focus here on the ones relevant for our analysis.

[13] We use loose terminology referring to points with dimensionality index j as j-D points.

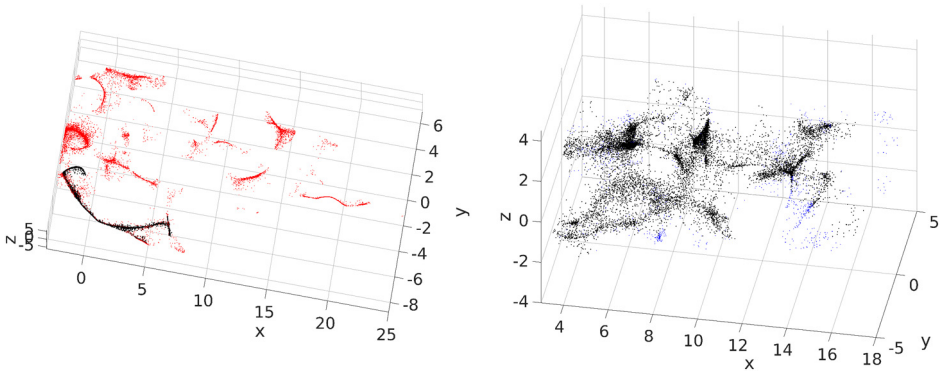


Fig. 21. 1-D (left) and 2-D (right) distributions of the diffused particles in the tail of the jellyfish structure. Highlighted in black are the points belonging to two distinct 1-D and 2-D structures discussed in section 6.1.

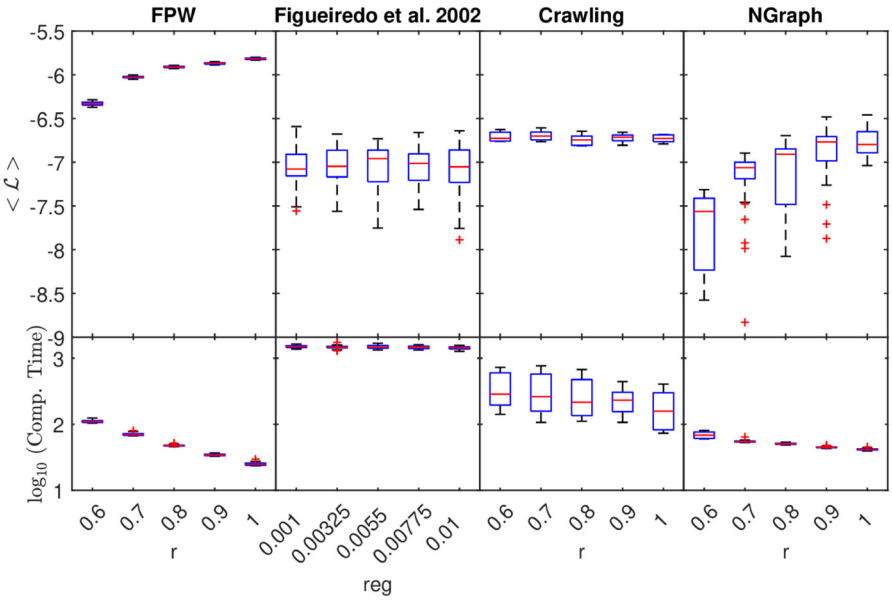


Fig. 22. Average log-likelihood (top row) and log computational time of the probabilistic modelling techniques described in section 5.2 applied to the jellyfish data set.

respectively - the gaseous stream of gas particles departing from the head and reaching half way through the tail and the predominant 2-D structure in the tail. Fig. 23, left panel shows two regions of high curvature, for the 2-dimensional manifold presented in Fig. 21, right panel. The far right region of intense curvature, is located towards the end of the tail, where the gas motion is more chaotic due to the motion of the galaxy through the halo of the galaxy cluster. However, the spherical region on the left side of the manifold, presenting a coherent curvature throughout its elongation (top circle of Fig. 23, left panel) suggests, as a possible cause of formation, an isotropic expansion, typical of Supernova remnants.

Taking advantage of the probabilistic nature of the AGTM model, we show in Fig. 24a the embedded vertices $\overline{\mathbf{v}}_j$ of the graph $\overline{\mathcal{G}}$ for the stream model, with intensity modulated by the weighted mean of [C II] values $\mathcal{I}_i^{[\text{C II}]}$ of particles $\mathbf{t}_i$ in the manifold, where the weights are the posterior probabilities of the node v_j, given particles $\mathbf{t}_i$:

$$\overline{\mathcal{I}}_j^{[\text{C II}]} = \frac{\sum_{i=1}^{N_{\mathcal{M}}} p(v_j|\mathbf{t}_i, \zeta_j\Sigma_j, \mathbf{W}_j)\, \mathcal{I}_i^{[\text{C II}]}}{\sum_{k=1}^{N_{\mathcal{M}}} p(v_j|\mathbf{t}_k, \zeta_j\Sigma_j, \mathbf{W}_j)} \tag{32}$$

Analogous figure for the 2-D structure is presented in Fig. 24b.

In the 1-D case, the manifold is located at the outskirts of the jellyfish (Fig. 20, left panel), meaning that it is more exposed during the evolution to the surrounding gas of the galaxy cluster. This implies that the manifold is subject to a higher ram pressure than the tail, leading to a higher density and lower temperature of the gas - necessary conditions for the formation of new stars. These conditions are reflected in an increase of the [C II] emission line over the middle section of the manifold, $3 < x < 6$, thus informing us of an enhanced Star Formation Rate, compared to the rest of the manifold. The

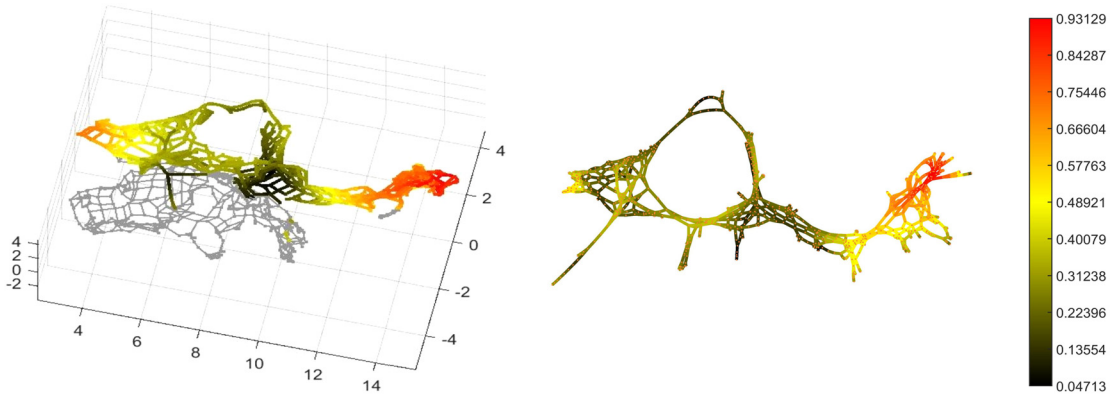


Fig. 23. Embedded graph (left panel) and planar representation (right panel) of a 2-dimensional manifold extracted from the jellyfish data set, and its on-edge curvature distribution.

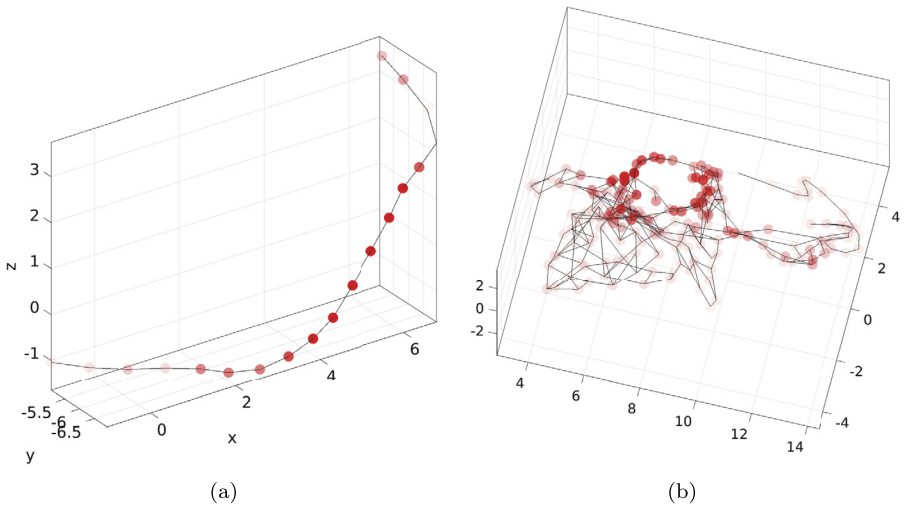


Fig. 24. 1D (top) and 2D (bottom) manifolds extracted from the data set in Fig. 20. The graphs' nodes are coloured based on the value of [C II] of the surrounding particles lying not further than 1 kpc from the local tangent space of the manifold, weighted by the nodes' responsibilities.

2-D structure shows an overall constant [C II] intensity whereas the region at $5 < x < 9$ presents sharply higher values. The shape of this region is particularly interesting. It is, in fact, a hole with an almost spherical section. This structure detected with AGTM and Manifold Crawling, with potent [C II] emission at its boundary, is the remnant of a supernova explosion. This process is modelled in the simulation via an injection of 10^{51} erg of energy for a short amount of time and a transfer of metallic elements (in the case of our simulation, the model tracks iron and magnesium, [53]) to the neighbouring gas particles. The metal-enriched gas particles (like in the surrounding of a supernova explosion) are then able to cool down more efficiently and show strong [C II] emission line.

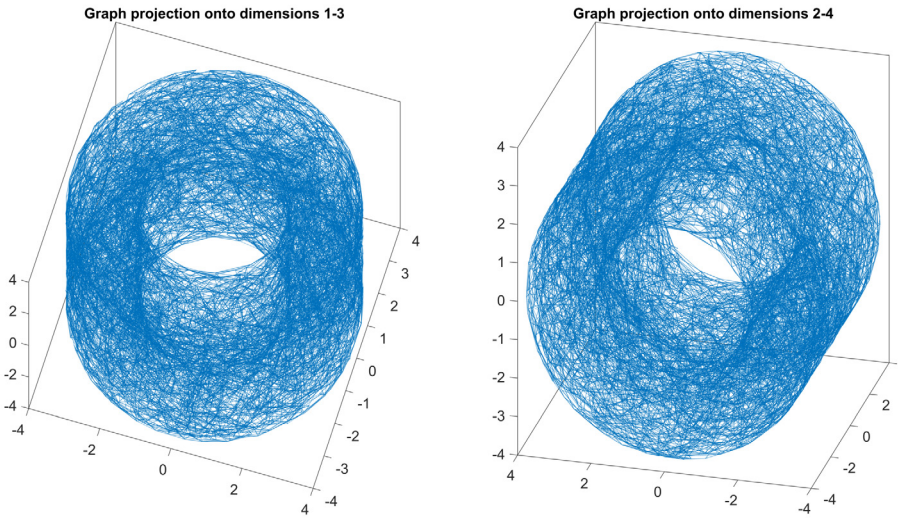
Our methodology provides a strong tool for extracting such an information from the morphology of gas particles and can be used to effectively calibrate feedback models[14] in simulations.

Such a detailed analysis of low dimensional structures (remnants of galaxy interactions) is not currently possible with tools routinely used to calibrate and analyse astrophysical simulations of galaxy evolution. As mentioned in section 4.2, the technique presented in this paper can be used as a semi-automatic exploratory tool by the domain experts, where the focus and characteristic scale of the structures to be mined can be varied continuously with analysis of their physical properties of interest (after necessary computations) performed and studied on the fly.

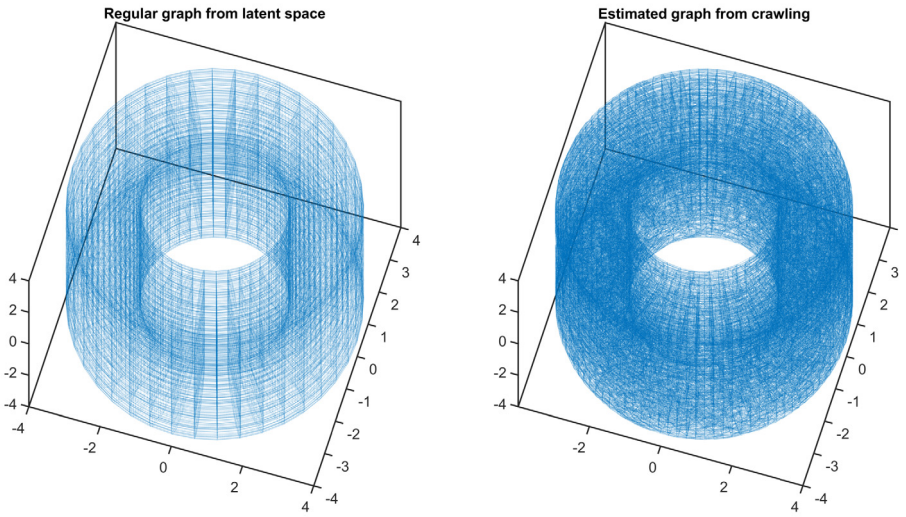
7. Detecting manifolds of higher dimensions

We performed an additional experiment on a 3-dimensional torus embedded in four dimensions. The torus is parametrized in the angle space $(\theta, \phi, \psi) \in [0, 2\pi] \times [0, 2\pi] \times [0, 2\pi]$ by

[14] A feedback mechanisms, is any process that allows to exchange energy, matter and/or momentum among galaxy components.



(a) Recovered graph (by crawling) for the 3−torus under projections on different coordinates triplets.



(b) Embedded regular latent graph (left) and graph recovered by crawling on a regular sampling of the 3−torus.

Fig. 25. Results of crawling on sampled 3-torus (top row) and comparison between ground truth graph (bottom left) and crawled regularly sampled 3-torus (bottom right).

$$r(\theta,\phi,\psi) = \begin{pmatrix} [3+\cos(\phi)]\cos(\theta) \\ [3+\cos(\phi)]\sin(\theta) \\ [3+\sin(\phi)]\cos(\psi) \\ [3+\sin(\phi)]\sin(\psi) \end{pmatrix}. \tag{33}$$

The 3-torus is covered by sampling the angle parameters with a large number of uniformly distributed values. We then applied the same procedure for producing a sample from the noisy manifold as used in the previous synthetic data experiments. After pre-processing the noisy data set by SAF and filtering, we proceed with estimating the dimensionality index on every remaining point, finding that we were able to retain most of the structure as made of 3-dimensional points. We then apply the Crawling algorithm, constructing an abstract-embedded pair of graphs. The embedded latent space is shown in Fig. 25a, projected onto two different triplets of dimensions. The global structure is clearly recovered through crawling. However, given that the abstract latent space was induced from a diffused sample of points not regularly covering the 3-

torus, the internal structure of the manifold is not easily identifiable. To demonstrate more clearly the ability of the crawling algorithm to recover the underlying 3-dimensional manifold structure, we let the crawling operate on a data set created as a regular grid of points lying on the 3-torus. This way we can directly compare the ground-truth embedded latent space graph (obtained by connecting neighbouring grid points in the 3-torus topology, shown in a sub-sampled version in Fig. 25b, left panel) to one recovered by crawling (Fig. 25b, right panel). The two embedded graphs (shown here in the projection corresponding to that of Fig. 25a, left panel) clearly sample the same manifold structure. As shown in this example, the methodology could in principle be applied to higher dimensional manifolds embedded in even higher dimensional space. This is also true if the ambient dimension increases, provided the manifold is well sampled and local neighbourhoods are populated enough to perform PCA. However, the computation time of the crawling algorithm is bound to increase, since at each expansion phase, $2j$ new children node estimates are created per parent node.

8. Conclusion

We presented a novel, semi-automated framework for denoising, dimensionality estimation, multi-manifold extraction and manifold aligned density estimation from complex data sets containing samples from noisy manifolds of diverse dimensionalities embedded in a noisy environment. The framework uses only a few easy-to-understand hyper-parameters, such as characteristic scale, that can be manipulated to obtain an emerging picture of the multi-manifold structure of the data. We have illustrated the workings of the methodology on two synthetic data sets containing a number of manifolds, including a toroid and a Möbius strip, embedded in an extremely noisy environment. For each manifold we were able to estimate its intrinsic dimensionality, and using Abstract GTM, to extract its embedded and abstract graphs through Manifold Crawling, while constructing a probabilistic model, describing its density. We provided an efficient alternative to the Crawling algorithm that is less computationally intensive. While competitive with other manifold learning algorithms, the alternative to Crawling proved less accurate than its slower, but more stable, graph construction method. We showed how it is possible to compute the local curvature on the recovered manifolds taking advantage of the smooth mapping function formulation of AGTM. We also performed a detailed comparison with both Multi-Manifold learning and Probabilistic modelling algorithms, widely used in the literature. The Multi-Manifold learning alternatives proved sensitive to noise, or impractical when dealing with multiple manifolds of different dimensionalities. The probabilistic modelling techniques, although capable of representing globally the data sets, are not designed to explicitly capture general noisy low-dimensional structures along which the data can be organized. AGTM proved equally capable of modelling the global distribution of the data sets, while still carrying meaningful information about their underlying low-dimensional manifolds.

The methodology was then applied to a complex data set containing simulated gas volume particles from a numerical simulation of a dwarf galaxy interacting with its host galaxy cluster. We have shown how the extracted 1-D and 2-D manifolds can help us to understand the nature of star formation inside such disrupted dwarf galaxies. Besides many other possible applications, our methodology offers new exciting ways of analysing details emerging from astrophysical simulations that are not possible with the current tools commonly used in astronomy.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This project has received financial support from the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement No. 721463 to the SUNDIAL ITN Network. PT and MC were supported by the Alan Turing Institute Fellowship 96102.

Appendix A. Manifolds equations and data sets generation

All synthetic data sets documented in the main body of this work are generated by first defining parametric functions of the "Ground Truth" manifolds, then adding noise to the manifolds, rearranging them via affine operators (scaling, rotation and shift) and adding a background noise. In the following, the parametric equations for each structure are shown and the noise addition process described.

The functional forms are presented in the following order: Parabolic arm $\mathcal{M}_1$ (1D), Spiral $\mathcal{M}_2$ (1D), S-shape $\mathcal{M}_3$ (2D), Hyperbolic surface $\mathcal{M}_4$ (2D), Toroid $\mathcal{M}_5$ (2D), Möbius strip $\mathcal{M}_6$ (2D), 3-ball $\mathcal{M}_7$ (3D) and 3-Toroid $\mathcal{M}_8$ (3D, embedded in 4D).

$$\mathcal{M}_1 : \begin{cases} x \in [0;1]; \\ y = -2x^2 + 2x + 1; \\ z = -0.2 \end{cases} \qquad \mathcal{M}_2(\theta_1, r) : \begin{cases} x = \frac{1}{2}[r\cos(\theta_1) + 1] - 0.1; \\ y = \frac{1}{2}[r\sin(\theta_1) + 1] + 0.4 \\ z \in [0.1;1]; \end{cases} \tag{A.1}$$

where $\theta_1 \in [2\pi; 10\pi]$ and $r \in [0.1; 1]$. In order to obtain the spiral data set, variables θ_1 and r have to be reordered in ascending order, while variable z in descending order.

$$\mathcal{M}_3 : \begin{cases} x = \frac{[\sin(\theta_2)+1]}{5}; \\ y \in [\frac{1}{3}; 1]; \\ z = \frac{\{\mathrm{sgn}(\theta_2)[\cos(\theta_2)+1]\}}{3}. \end{cases} \tag{A.2}$$

Here, $\theta_2 \in [-5; 3\pi - 5]$.

$$\mathcal{M}_4 : \begin{cases} x = R\sin(\theta_3) + 0.5; \\ y = R\cos(\theta_3) - 0.5; \\ z \in [0.4; 0.6] \end{cases} \quad \text{where} \quad \begin{cases} r \in [0; 0.5]; \\ R = -r^2 + 0.25; \\ \theta_3 \in [0; 2\pi], \end{cases} \tag{A.3}$$

variables r, θ_3 and z must be sorted in ascending order.

$$\mathcal{M}_5 : \begin{cases} x = [R + r\cos(v)]\cos(u); \\ y = [R + r\cos(v)]\sin(u); \\ z = r\sin(v). \end{cases} \tag{A.4}$$

Here we set $R = 0.5, r = 0.15$, $(u, v) \in [0; 2\pi] \times [0; 2\pi]$.

The Móbius strip is obtained by the equations:

$$\mathcal{M}_6 : \begin{cases} x = \left[1 + \frac{t}{2}\cos(\frac{u}{2})\right]\cos(u); \\ y = \left[1 + \frac{t}{2}\cos(\frac{u}{2})\right]\sin(u); \\ z = \frac{t}{2}\sin(\frac{u}{2}) \end{cases} \tag{A.5}$$

where $t \in [-1; 1]$.

Manifold $\mathcal{M}_7$ is the 3-ball of radius $r = 1$. The 3-torus, embedded in 4 dimensions (discussed in section 7) is parameterised by:

$$\mathcal{M}_8 : \begin{cases} x = [3 + \cos(\Phi)]\cos(\theta); \\ y = [3 + \cos(\Phi)]\sin(\theta); \\ z = [3 + \sin(\Phi)]\cos(\Psi); \\ \ell = [3 + \sin(\Phi)]\sin(\Psi); \end{cases} \tag{A.6}$$

where $(\theta, \Phi, \Psi) \in [0; 2\pi] \times [0; 2\pi] \times [0; 2\pi]$.

To generate data set $\mathcal{D}^1$, a two rotation operators are needed, namely $\mathcal{T}_1$ and $\mathcal{T}_2$:

$$\mathcal{T}_1 : \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} \qquad \mathcal{T}_2 : \begin{pmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & 0. \end{pmatrix} \tag{A.7}$$

Furthermore, a set of shifting and scaling operators is designed for each manifold, in order to obtain the presented configuration:

$$\hat{\mathcal{M}}_1 = \mathcal{M}_1 + [-0.1; 0.1; 0]; \tag{A.8}$$

$$\hat{\mathcal{M}}_2 = 0.4 \times \mathcal{M}_2 + [0.1; 0.4; -0.9]; \tag{A.9}$$

$$\hat{\mathcal{M}}_3 = \mathcal{T}_1(\mathcal{M}_3) + [0; 0; 0.5]; \tag{A.10}$$

$$\hat{\mathcal{M}}_4 = 1.2 \times \mathcal{M}_4 + [-0.4; 1.3; 0.3]; \tag{A.11}$$

$$\hat{\mathcal{M}}_5 = 1.2 \times \mathcal{M}_5 + [0.2; 0.68; 0.5]; \tag{A.12}$$

$$\hat{\mathcal{M}}_6 = 0.25 \times \mathcal{T}_2(\mathcal{M}_6) + [0.4; 1.5; -0.2]; \tag{A.13}$$

$$\hat{\mathcal{M}}_7 = 0.2 \times \mathcal{M}_7 + [0.2; 0.5; 0]; \tag{A.14}$$

The construction of data set $\mathcal{D}^2$ is simpler and relies on specific rotations and shifts of manifold $\mathcal{M}_5$. In particular:

$$\hat{\mathcal{M}}_9 = \mathcal{M}_5 - [0.95; 0; 0]; \tag{A.15}$$

$$\hat{\mathcal{M}}_{10} = 1.4 \times \mathcal{T}_1(\mathcal{M}_5) + [0; 0; 0.12]; \tag{A.16}$$

$$\hat{\mathcal{M}}_{11} = \mathcal{M}_5 + [1.9; 0; 0]. \tag{A.17}$$

For each manifold $\hat{\mathcal{M}}_i, i = 1, \ldots, 9$, we first sample a large number of $N_{GT} \sim 1e4$ points from its ground truth $\mathcal{M}_i$, densely covering the whole structure. Around each sample point on the manifold we position a hyper-ball of fixed radius $r_{GT} = 0.04$. Let us denote the union of the hyper-balls by $\mathcal{B}_{GT}$. To generate a sample from a noisy manifold, we first envelope the ground truth manifold with a cuboid in the ambient space. The cuboid is then covered with a large number $N_{Cub} \sim 1e6$ of points sampled from the uniform distribution and collected in $\mathcal{S}_{Cub}$. The sample from the noisy manifold is then $\mathcal{S}_{Cub} \cap \mathcal{B}_{GT}$.

After extensive testing of hyper-parameters for the SAF technique, we found that the best results were achieved with $r = 0.05, \mu = 1e-3, N_{iter} = 5$.

References

[1] K. Pearson, LIII. On lines and planes of closest fit to systems of points in space, Lond. Edinb. Dublin Philos. Mag. J. Sci. 2 (11) (1901) 559–572, https://doi.org/10.1080/14786440109462720, pCA beginnings.

[2] J.B. Tenenbaum, V.d. Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (5500) (2000) 2319–2323, https://doi.org/10.1126/science.290.5500.2319.

[3] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (2000) 2323–2326.

[4] M. Belkin, P. Niyogi, Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering, Advances in Neural Information Processing Systems, vol. 14, MIT Press, 2001, pp. 585–591.

[5] D.L. Donoho, C. Grimes, Hessian eigenmaps: locally linear embedding techniques for high-dimensional data, Proc. Natl. Acad. Sci. 100 (10) (2003) 5591–5596, https://doi.org/10.1073/pnas.1031596100.

[6] Z. Zhang, H. Zha, Principal manifolds and nonlinear dimension reduction via local tangent space alignment, SIAM J. Sci. Comput. 26 (2002) 313–338.

[7] V.d. Silva, J.B. Tenenbaum, Global versus local methods in nonlinear dimensionality reduction, in: Proceedings of the 15th International Conference on Neural Information Processing Systems, NIPS'02, MIT Press, Cambridge, MA, USA, 2002, pp. 721–728, http://dl.acm.org/citation.cfm?id=2968618.2968708.

[8] W.S. Torgerson, Warren S. Torgerson, Theory and methods of scaling. New York: John Wiley and Sons, Inc., 1958. Pp. 460, Behav. Sci. 4 (3) (1958) 245–247, https://doi.org/10.1002/bs.3830040308.

[9] M.A.A. Cox, T.F. Cox, Multidimensional Scaling, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 315–347.

[10] J.B. Kruskal, Nonmetric multidimensional scaling: a numerical method, Psychometrika 29 (2) (1964) 115–129, https://doi.org/10.1007/BF02289694.

[11] T. Lin, H. Zha, Riemannian manifold learning, IEEE Trans. Pattern Anal. Mach. Intell. 30 (5) (2008) 796–809, https://doi.org/10.1109/TPAMI.2007.70735.

[12] J.-D. Boissonnat, F. Chazal, M. Yvinec, Geometric and Topological Inference, Cambridge Texts in Applied Mathematics, Cambridge University Press, 2018, https://hal.inria.fr/hal-01615863.

[13] C.M. Bishop, M. Svensén, C.K.I. Williams, GTM: the generative topographic mapping, Neural Comput. 10 (1998) 215–234.

[14] T. Kohonen, Self-organized formation of topologically correct feature maps, Biol. Cybern. 43 (1) (1982) 59–69.

[15] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the em algorithm, J. R. Stat. Soc. B 39 (1) (1977) 1–38, http://www.jstor.org/stable/2984875.

[16] A. Taghribi, M. Canducci, M. Mastropietro, S. De Rijcke, K. Bunte, P. Tiňo, ASAP – a sub-sampling approach for preserving topological structures modeled with geodesic topographic mapping, Neurocomputing (2021), https://doi.org/10.1016/j.neucom.2021.05.108.

[17] I. Olier, A. Vellido, Variational Bayesian generative topographic mapping, J. Math. Model. Algorithms 7 (2008) 371–387.

[18] E. Parzen, On estimation of a probability density function and mode, Ann. Math. Stat. 33 (3) (1962) 1065–1076.

[19] P. Vincent, Y. Bengio, Manifold Parzen windows, in: S. Becker, S. Thrun, K. Obermayer (Eds.), Advances in Neural Information Processing Systems, vol. 15, MIT Press, 2003, pp. 849–856, http://papers.nips.cc/paper/2203-manifold-parzen-windows.pdf.

[20] X. Wang, P. Tino, M.A. Fardal, S. Raychaudhury, A. Babul, Fast Parzen window density estimator, in: 2009 International Joint Conference on Neural Networks, 2009, pp. 3267–3274.

[21] C.E. Rasmussen, The infinite Gaussian mixture model, in: Proceedings of the 12th International Conference on Neural Information Processing Systems, NIPS'99, MIT Press, Cambridge, MA, USA, 1999, pp. 554–560.

[22] W. Yang, C. Sun, L. Zhang, A multi-manifold discriminant analysis method for image feature extraction, Pattern Recognit. 44 (8) (2011) 1649–1657, https://doi.org/10.1016/j.patcog.2011.01.019.

[23] M. Fan, H. Qiao, B. Zhang, X. Zhang, Isometric multi-manifold learning for feature extraction, in: 2012 IEEE 12th International Conference on Data Mining, 2012, pp. 241–250.

[24] M. Fan, X. Zhang, H. Qiao, B. Zhang, Efficient isometric multi-manifold learning based on the self-organizing method, Inf. Sci. 345 (C) (2016) 325–339, https://doi.org/10.1016/j.ins.2016.01.069.

[25] R. Hettiarachchi, J. Peters, Multi-manifold LLE learning in pattern recognition, Pattern Recognit. 48 (9) (2015) 2947–2960, https://doi.org/10.1016/j.patcog.2015.04.003.

[26] S. Mahapatra, V. Chandola, S-Isomap++: multi manifold learning from streaming data, arXiv:1710.06462, Oct. 2017.

[27] P. Tino, I. Nabney, Hierarchical GTM: constructing localized nonlinear projection manifolds in a principled way, IEEE Trans. Pattern Anal. Mach. Intell. 24 (5) (2002) 639–656, https://doi.org/10.1109/34.1000238.

[28] G. Haro, G. Randal, G. Sapiro, G. Haro, G. Randall, G. Sapiro, Translated Poisson mixture model for stratification learning, Int. J. Comput. Vis. (2000).

[29] M. Allegra, E. Facco, F. Denti, A. Laio, A. Mira, Data segmentation based on the local intrinsic dimension, Sci. Rep. 10 (2020) 16449, https://doi.org/10.1038/s41598-020-72222-0, arXiv:1902.10459.

[30] E. Facco, M. d'Errico, A. Rodriguez, A. Laio, Estimating the intrinsic dimension of datasets by a minimal neighborhood information, Sci. Rep. 7 (2017).

[31] X. Wang, P. Tiňo, M.A. Fardal, Multiple manifolds learning framework based on hierarchical mixture density model, in: W. Daelemans, B. Goethals, K. Morik (Eds.), Machine Learning and Knowledge Discovery in Databases, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 566–581.

[32] C.M. Bishop, M. Svensén, C.K.I. Williams, Developments of the generative topographic mapping, Neurocomputing 21 (1998) 203–224.

[33] D. Bacciu, A. Micheli, A. Sperduti, Compositional generative mapping for tree-structured data—part II: topographic projection model, IEEE Trans. Neural Netw. Learn. Syst. 24 (2) (2013) 231–247.

[34] S. Wu, P. Bertholet, H. Huang, D. Cohen-Or, M. Gong, M. Zwicker, Structure-aware data consolidation, IEEE Trans. Pattern Anal. Mach. Intell. 40 (10) (2018) 2529–2537, https://doi.org/10.1109/TPAMI.2017.2754254.

[35] K.B. Petersen, M.S. Pedersen, The Matrix Cookbook, Technical University of Denmark, 2012, version 20121115, http://localhost/pubdb/p.php?3274.

[36] H. Hotelling, Analysis of a complex of statistical variables with principal components, J. Educ. Psychol. 24 (1933) 417–441.

[37] J.O. Rawlings, S.G. Pantula, D.A. Dickey, Applied Regression Analysis: A Research Tool, Springer Science & Business Media, 2001.

[38] P. Mordohai, G. Medioni, Unsupervised dimensionality estimation and manifold learning in high-dimensional spaces by tensor voting, in: Proceedings of the 19th International Joint Conference on Artificial Intelligence, IJCAI'05, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005, pp. 798–803, http://dl.acm.org/citation.cfm?id=1642293.1642421.

[39] G. Lebanon, Riemannian geometry and statistical machine learning, Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA, USA, 2005, aAI3159986.
[40] A.F. Möbius, Der barycentrische Calcul: ein neues Hülfsmittel zur analytischen Behandlung der Geometrie, Barth, 1827.
[41] H. Gunawan, O. Neswan, W.S. Budhi, A formula for angles between subspaces of inner product spaces, Beitr. Algebra Geom. 46 (01 2005).
[42] E. Elhamifar, R. Vidal, Sparse manifold clustering and embedding, in: J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, K.Q. Weinberger (Eds.), Advances in Neural Information Processing Systems, vol. 24, Curran Associates, Inc., 2011, pp. 55–63, https://proceedings.neurips.cc/paper/2011/file/fc490ca45c00b1249bbe3554a4fdf6fb-Paper.pdf.
[43] A.M. Saranathan, M. Parente, On clustering and embedding mixture manifolds using a low rank neighborhood approach, IEEE Trans. Geosci. Remote Sens. 57 (6) (2019) 3890–3903, https://doi.org/10.1109/TGRS.2018.2888983.
[44] M.A.T. Figueiredo, A.K. Jain, Unsupervised learning of finite mixture models, IEEE Trans. Pattern Anal. Mach. Intell. 24 (3) (2002) 381–396, https://doi.org/10.1109/34.990138.
[45] B.M. Poggianti, A. Moretti, M. Gullieuszik, J. Fritz, Y. Jaffé, D. Bettoni, G. Fasano, C. Bellhouse, G. Hau, B. Vulcani, et al., GASP. I. Gas stripping phenomena in galaxies with MUSE, Astrophys. J. 844 (1) (2017) 48, https://doi.org/10.3847/1538-4357/aa78ed.
[46] I. De Looze, M. Baes, G.J. Bendo, L. Cortese, J. Fritz, The reliability of [C II] as an indicator of the star formation rate, Mon. Not. R. Astron. Soc. 416 (4) (2011) 2712–2724, https://doi.org/10.1111/j.1365-2966.2011.19223.x, arXiv:1106.1643.
[47] R. Herrera-Camus, A.D. Bolatto, M.G. Wolfire, J.D. Smith, K.V. Croxall, R.C. Kennicutt, D. Calzetti, G. Helou, F. Walter, A.K. Leroy, et al., [C ii] 158 μm emission as a star formation tracer, Astrophys. J. 800 (1) (2015) 1, https://doi.org/10.1088/0004-637x/800/1/1.
[48] H. Ebeling, L.N. Stephenson, A.C. Edge, Jellyfish: evidence of extreme ram-pressure stripping in massive galaxy clusters, Astrophys. J. 781 (2) (2014) L40, https://doi.org/10.1088/2041-8205/781/2/l40.
[49] M.D. Mora, J. Chanamé, T.H. Puzia, A starburst in the core of a galaxy cluster: the dwarf irregular NGC 1427A in Fornax, Astron. J. 150 (3) (2015) 93, https://doi.org/10.1088/0004-6256/150/3/93, arXiv:1411.7314.
[50] K. Lee-Waddell, P. Serra, B. Koribalski, A. Venhola, E. Iodice, B. Catinella, L. Cortese, R. Peletier, A. Popping, O. Keenan, M. Capaccioli, Tidal origin of NGC 1427A in the Fornax cluster, Mon. Not. R. Astron. Soc. 474 (1) (2017) 1108–1115, https://doi.org/10.1093/mnras/stx2808.
[51] M. Mastropietro, S. De Rijcke, R.F. Peletier, A tale of two tails: insights from simulations into the formation of the peculiar dwarf galaxy NGC 1427A, Mon. Not. R. Astron. Soc. 504 (3) (2021) 3387–3398, https://doi.org/10.1093/mnras/stab1091, arXiv:2104.07671.
[52] V. Springel, The cosmological simulation code GADGET-2, Mon. Not. R. Astron. Soc. 364 (4) (2005) 1105–1134, https://doi.org/10.1111/j.1365-2966.2005.09655.x.
[53] S. De Rijcke, J. Schroyen, B. Vandenbroucke, N. Jachowicz, J. Decroos, A. Cloet-Osselaer, M. Koleva, New composition-dependent cooling and heating curves for galaxy evolution simulations, Mon. Not. R. Astron. Soc. 433 (4) (2013) 3005–3016, https://doi.org/10.1093/mnras/stt942, arXiv:1306.4860.
[54] M. Nichols, Y. Revaz, P. Jablonka, The post-infall evolution of a satellite galaxy, Astron. Astrophys. 582 (2015) A23, https://doi.org/10.1051/0004-6361/201526113, arXiv:1503.05190.
[55] J.F. Navarro, C.S. Frenk, S.D.M. White, The structure of cold dark matter halos, Astrophys. J. 462 (1996) 563, https://doi.org/10.1086/177173, arXiv:astro-ph/9508025.
[56] M. Paolillo, G. Fabbiano, G. Peres, D.-W. Kim, Deep ROSAT HRI observations of the NGC 1399/NGC 1404 region: morphology and structure of the X-ray halo, Astrophys. J. 565 (2) (2002) 883–907, https://doi.org/10.1086/337919.
[57] R. Gingold, J. Monaghan, Smoothed particle hydrodynamics - theory and application to non-spherical stars, Mon. Not. R. Astron. Soc. 181 (1977) 375–389, https://doi.org/10.1093/mnras/181.3.375.
[58] U. Maio, K. Dolag, B. Ciardi, L. Tornatore, Metal and molecule cooling in simulations of structure formation, Mon. Not. R. Astron. Soc. 379 (3) (2007) 963–973, https://doi.org/10.1111/j.1365-2966.2007.12016.x.