UCLouvain
Institute for Information
and Communication Technologies,
Electronics and Applied Mathematics

Reliable and Interactive Neighbour Embedding for Visualisation, Dimensionality Reduction and Machine Learning

Pierre Lambert

Thesis for the degree of
Docteur en sciences de l'ingénieur et technologie

Dissertation committee:

Prof. John Lee (UCLouvain, Belgium) – Advisor
Prof. Michel Verleysen (UCLouvain, Belgium)
Prof. Bruno Dumas (UNamur, Belgium)
Prof. Jaakko Peltonen (Tampere university, Finland)
Prof. Dmitry Kobak (Tübingen University, Germany)
Prof. Christophe Craeye (UCLouvain, Belgium) – Chair

February 2026.

Acknowledgments

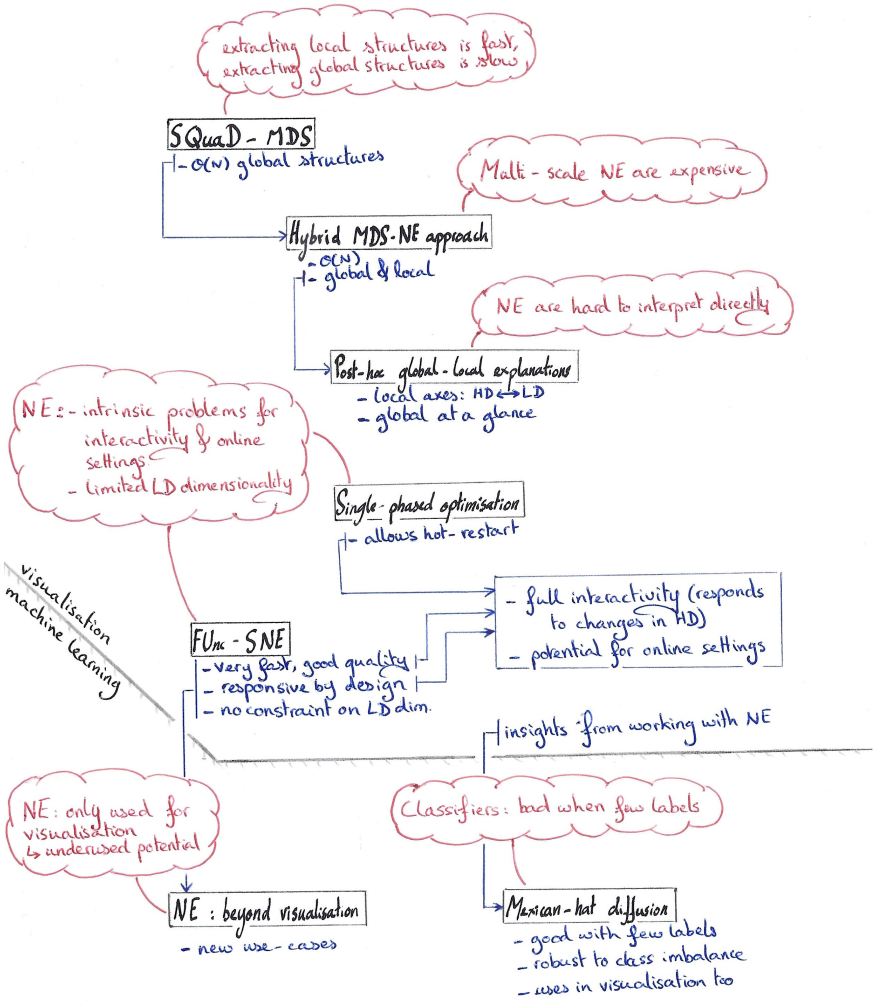
My lifelong gratitude is reserved for those who believed in me and opened the way toward what I hope will be a life in service of science rather than of individuals. I also acknowledge the hand of providence, for fortune has been unreasonably kind from the beginning of my life to this day.

Abstract

This thesis presents research conducted over the past four years on neighbour embeddings, algorithms performing non-linear dimensionality reduction intended for data visualisation. The initial plan was to focus on visualisation and on the scaling of such methods, but during the final stretch of the thesis, the research was steered in another direction, exploring the possibility of applying lessons learned from neighbour embeddings to broader machine learning tasks.
As a reductive summary whose brevity does not do justice to the contributions, this endeavour has introduced new algorithms allowing to capture relationships in data at different scales, always with scalability in mind. To circumvent some core issues with neighbour embeddings that hindered responsiveness to changes in data or hyperparameters, a rethinking of how to perform neighbour embeddings has been undertaken, and an algorithm was established. The newfound properties of NE manifested by this algorithm open potential for unexplored applicative horizons and set in motion reflection on how to transpose properties of neighbour embeddings to other machine learning algorithms. This reflection, after extensive trial and error, gave rise to an innovative algorithm for label diffusion which shows particularly promising results.

The published or under-review contributions of the thesis are added directly to this document, with some accompanying words. This thesis deliberately favours intuitive explanations over strict mathematical formalism, mirroring this author's own way of apprehending the matter. The first chapter motivates the use of neighbour embeddings in visualisation, outlines their remaining limitations, and shows how the contributions build upon one another, all while maintaining a high-level, conceptual perspective that avoids technical detail. The second chapter formalises the inner

workings of neighbour embeddings and relevant dimensionality reduction algorithms. The third chapter details every contribution. The last chapter puts to light some insights on neighbour embeddings gained throughout the thesis and suggests some interesting research directions.



Graphical representation of the main contributions (black), with their principal properties (blue text), requirements (blue arrows), and motivations (red).

Publications as 1^{st} author

- **Impact of data subsamplings in Fast Multi-Scale Neighbor Embedding**
  Pierre Lambert,John Lee, Michel Verleysen, Cyril de Bodt
  ESANN 2021 - poster session

- **Stochastic quartet approach for fast multidimensional scaling**
  Pierre Lambert, Cyril de Bodt, Michel Verleysen, John Lee
  ESANN 2021 - talk

- **SQuadMDS: A lean Stochastic Quartet MDS improving global structure preservation in neighbor embedding like t-SNE and UMAP**
  Pierre Lambert, Cyril de Bodt, Michel Verleysen, John Lee
  Neurocomputing - journal (2022)

- **Globally local and fast explanations of t-SNE-like nonlinear embeddings**
  Pierre Lambert, Rebecca Marion, Julien Albert, Emmanuel Jean, Sacha Corbugy, Cyril de Bodt
  CIKM 2022 workshop - talk

- **Nesterov momentum and gradient normalization to improve t-SNE convergence and neighborhood preservation, without early exaggeration**
  Pierre Lambert,John Lee, Edouard Couplet, Cyril de Bodt
  ESANN 2023 - poster session

- **Estimated neighbour sets and smoothed sampled global interactions are sufficient for a fast approximate t-SNE.**
  Pierre Lambert, Edouard Couplet, Cyril de Bodt ,John Lee
  ESANN 2024 - talk

- **Neighbour Embeddings: Beyond Visualisation.**
  Pierre Lambert, Edouard Couplet, Michel Verleysen, John Lee
  MLVis 2025 - talk

- **FUnc-SNE: A flexible, Fast, and Unconstrained algorithm for neighbour embeddings.**
  Pierre Lambert, Edouard Couplet, Michel Verleysen, John Aldo Lee
  Neurocomputing - journal (UNDER REVIEW)

Abbreviations

- **DR**: Dimensionality Reduction
- **NE**: Neighbour Embedding
- **HD**: High(er)-Dimensional
- **LD**: Low(er)-Dimensional
- **(K)-NN**: (K)-Nearest Neighbours
- **(t-)SNE**: (t-distributed) Stochastic Neighbour Embedding
- **PC(A)**: Principal Component (Analysis)
- **ML**: Machine Learning
- **QA**: Quality Assessment
- **RF**: Random Forest
- **SVM**: Support Vector Machine
- **CNN**: Convolutional Neural Network
- **IDIM**: Intrinsic Dimensionality

Contents

Acknowledgments **1**

Abstract **3**

Papers as 1st author **7**

Abbreviations **9**

Contents **11**

1 Neighbour Embeddings: Their Uses, Limitations, and Overview of the Contributions **13**
1.1 The uses of neighbour embeddings 14
1.2 Limitations and challenges 18
1.3 Overview of the contributions 35

2 Dimensionality Reduction and Neighbour Embeddings **39**
2.1 Dimensionality reduction 40
2.1.1 Principal component analysis 40
2.1.2 Multidimensional scaling 42
2.2 Neighbour embeddings 44
2.2.1 Technical aspects 44
2.2.2 Properties resulting from the formulation 47
2.2.3 Accelerating neighbour embeddings 51

3 Contributions **55**
3.1 Impact of data subsampling on fast multi-scale neighbour embeddings 56

3.2 SQuaD-MDS: a fast approximation of multi-dimensional scaling . . . 56
3.3 A hybrid MDS-NE method . . . 64
3.4 Post-hoc explainability for neighbour embeddings . . . 77
3.5 Enhancing optimisation with Nesterov momentum and gradient normalisation . . . 85
3.6 FUnc-SNE: a fast, accurate, and unconstrained NE method . 92
3.7 Neighbour embeddings: beyond visualisation . . . 123
3.8 Enhancing label diffusion with lessons from neighbour embeddings . . . 130
3.8.1 First observations, and the Label Spreading baseline 131
3.8.2 The first component: attractive forces . . . 133
3.8.3 The second component: repulsive forces . . . 136
3.8.4 Results . . . 141

4 Discussion and Perspectives 149
4.1 General insights . . . 150
4.2 Conclusion and Future Directions . . . 155

References 171

5 Annexes 173
5.1 Intuition on intrinsic dimensionality . . . 174
5.1.1 Intuition on intrinsic dimensionality and its estimation 174
5.1.2 Can ImageNet fit in 32 dimensions? . . . 181
5.2 Are neighbour embeddings robust to noise? . . . 188
5.3 Datasets . . . 194
5.4 Paper : Impact of data subsampling on fast multi-scale neighbour embeddings . . . 196

Chapter 1

Neighbour Embeddings: Their Uses, Limitations, and Overview of the Contributions

This thesis contributes to the development and maturation of neighbour embeddings (NE), a family of non-linear dimensionality reduction (DR) algorithms conceptualised for data visualisation. This section describes the motivations behind this work and brings an overview of the contributions as they developed throughout this thesis. Technicalities are left to later sections.

Subsection 1.1 explains intuitively why DR in general, and NE in particular, are crucial tools for data analysis across various domains. Subsection 1.2 presents the limitations and challenges that NE face in practice, without entering into the technical details. Subsection 1.3 provides a high-level overview of the contributions and their interrelations.

1.1 The uses of neighbour embeddings

Fashioned by evolution in a 3-dimensional environment, the human brain is particularly well adapted to perceive and understand spatial relationships in 2-dimensional and 3-dimensional spaces. In particular, we excel at detecting distinct objects, breaks in patterns, movements, at recognising familiar shapes, and at inferring depth. These characteristics, shared by most terrestrial vertebrates living on the surface, stem from the structures of our visual cortex across multiple scales. Interestingly, these structures share resemblance with deep convolutional networks using max pooling [1–4], with some level of feedback connections at higher levels [5].

These highly effective and efficient hard-wired perceptual abilities operate on visual data encoded as a 2-dimensional arrangement of activations on the retina. Binocular vision and movement allow us to effortlessly assimilate 3-dimensional information from these 2-dimensional projections; however, as soon as the observed structures lay in higher-dimensional (HD) spaces, all intuition is lost. This is problematic, as the data encountered in science and in the industry generally lives in HD spaces.

It is for this purpose that Stochastic Neighbour Embedding (SNE) [6] was introduced in 2002, the first of the progenitive family of NE methods. The technicalities of NE are detailed in Section 2. For the purpose of understanding why the NE family has quickly become a staple of data visualisation, it suffices to know that, as methods of DR, they aim to represent HD data in low-dimensional (LD) spaces (typically 2− or 3-dimensional), but, unlike other methods, they excel at preserving the fine-grained intricacies of the structures drawn by the data points, if these are of sufficiently low intrinsic dimensionality. Annex 5.1 gives a quick intuitive explanation of the notion of intrinsic dimensionality.

A sense of the type of structures in HD data that can be made manifest by the grace of NE is given in Figure 1.1. The dataset visualised there consists of a processed version of the dataset introduced in [7]. There, the concentration of thousands of distinct messenger RNA types was measured in the cytoplasm of 23822 individual cells from the mouse neocortex. Before being processed by the DR methods, the dataset is subject to feature selection and scaling as in [8], and then the first 50 principal components are

extracted. Principal components, the output of an method called Principal Component Analysis (PCA) [9], are linear combinations of the original features that capture most of the variance in the data; these are orthogonal in order to avoid linear redundancy. More details on PCA and other methods of DR are given in Section 2.

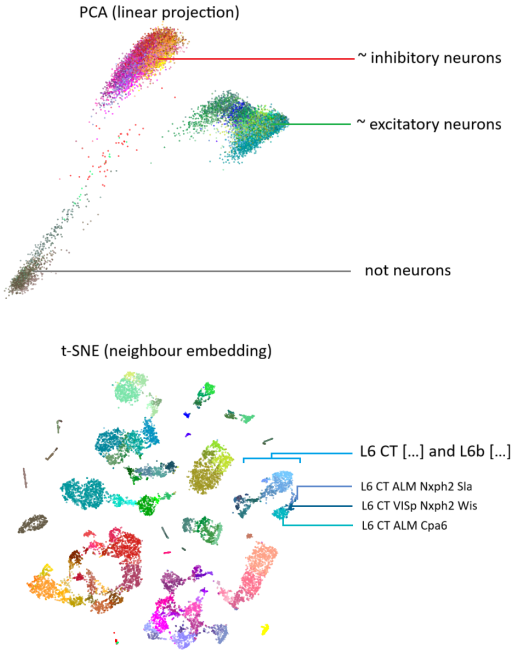


Figure 1.1: Embeddings of the mouse brain sc-dataset [7] using PCA (top) and t-SNE (bottom). The axes are voluntarily hidden, as they bring no useful information in these embeddings.

In this figure, the data, now reduced to a 23822 by 50 matrix, is transformed to two 23822 by 2 matrices using two methods, this allows visualisation in scatterplots. The first method, which produced the embedding on the top, is PCA: the same method that was used during the initial preprocessing of the data. Variance being a global measure of spread, PCA tends to preserve global structures but often fails to capture fine-grained patterns in HD data. In Figure 1.1, the PCA representation clearly shows the broad distinction between inhibitory neurons (GABAergic, warm colours)

and excitatory neurons (glutaminergic, cold colours), as well as the non-neuronal cells (grey-brown colours). However, each of these 3 broad categories appear as a blurry cloud of points: one couldn't guess that further structure exists within these clouds without prior biological knowledge or experience with DR.

On the other hand, the bottom view is a representation produced by a NE method, t-distributed Stochastic Neighbour Embedding (t-SNE) [10] using the algorithm developed in 3.6. One can notice that under the lenses of NE, the 3 broad categories identified with PCA are in fact hiding further internal structure, showing distinct clusters with sharp boundaries. Within the visual clusters, the colours indicating different cell categories tend to be clearly separated in distinct zones, hinting that further structure probably exists inside the clusters. This is further alluded by the shape of the clusters themselves featuring irregular boundaries. From these considerations, a biologist might emit as hypotheses that profound distinctions in the gene expression levels exist between the distinct clusters visible in the NE, and that some cell types present in the same clusters share a lot in common, but still differ in some aspects. The intra-cluster organisation of cells may indicate a continuous spectrum of gene expression levels in which hard boundaries were drawn by biologists, but one cannot exclude the possibility that these apparent continuities are in fact composed of distinct "sub-clusters" with finer discontinuities than those distinguishing the visible clusters from one another, which the NE failed to capture. The granularity at which clusters are extracted is studied in [11, 12], and further information on the subject is brought throughout the thesis.

This figure illustrates how NE can complement traditional data analysis methods to help biologists develop a more intuitive understanding of their data and thus help formulate and refine hypotheses, many works in biology put these methods at the centre of their data analysis toolbox. In [13], the authors developed a process applied to research on Leukaemia, where NE are used to visualise bone marrow samples from healthy and cancerous patients, showing clear distinctions between the two. Another notable example is CEBRA [14], where neuroscientists developed a NE method allowing joint embeddings of neural activity data together with behavioural variables, to better understand how biological brains internally represent the world as they experience it. In [15], the authors use a NE-based approach to integrate and analyse single-cell data from multiple modulari-

ties, to allow for more precise cell state identification. Other papers introduce good practices for the use of NE in biological data [8] and assess the quality and reliability of the available NE methods in the context of biology [16–18].

While biology is a domain where NE methods have found a strong foothold, these methods are used in myriad fields: as long as a domain generates data and humans seek to understand it visually, there is a place for NE. In chemistry, one can use NE to visualise an embedding representing a global reaction space [19], aiding chemists in synthesis planning. Proteins are large chains of amino-acids that fold in 3D space depending on their sequence and their environment, with a stochastic component; studying these conformations is crucial to understand their functions and can be aided by NE [20].
Another field where NE are applied is in the study of deep neural networks, which are notorious for being hard to interpret black boxes. In [21, 22], the transformation of the data as it passes through the layers of a neural network is made apparent using NE. The learning dynamics can also be better understood through the use of NE [23]. In [24], the authors use NE to analyse and build insights on the effects of fine-tuning large language models. In the paper presented in section 3.7, a NE is used to reveal that the text encoder and image encoder of a large vision-language model do not synchronise as one might expect.
Cheese analytics is also a domain benefiting from such methods: as the authors of [25] put it, "intelligent cheese selection is critical [...]", and NE helps them devise a strategy.
Some studies in geoscience use NE to extract significant patterns in HD data. One can use NE to visualise and analyse large quantities of hyperspectral images [26] or satellite images in the visual spectrum [27].
Cybersecurity has also benefited from NE. In [28], the authors use NE to visualise and identify hostile network traffic both visually and by using the NE as a feature extraction step prior to classification by random forests [29]. Utilising neighbour embeddings for data pre-processing is a possibility that remains largely untapped, for reasons that will become apparent throughout the document. This thesis claims that some modern NE methods such as the one introduced in 3.6 have properties that make them particularly well suited for this purpose, and that core principles of NE can be repurposed to yield competitive algorithms in various machine learning (ML) tasks.

One last application domain worth mentioning here is industrial process monitoring and anomaly detection. When a process generates HD data through multiple sensors, having a visual representation of the state of the system can be a powerful tool for human monitoring operators [30, 31]. Most papers on the matter use NE to process static data collected after a batch because NE methods are not designed for online use: adding new data can lead to high overhead, and drifting data distributions require to fully re-run the algorithms.

In short, independently from the domain of application, NE are currently used for:

- Developing intuition on HD data, helping the formulation and refinement of hypotheses.
- Monitoring processes visually.
- Showing information visually, to support an explanation.
- Assisting in ML tasks by being used as feature extractors.

It is the opinion of this author that they should never, however, be used alone to accept or reject a hypothesis, because the process of DR is lossy in general and non-linear DR such as NE rely on particularly impactful inductive biases that can lead to misinterpretations if not used carefully. Section 1.2 discusses these limitations in more details.

1.2 Limitations and challenges

Although NE methods are now well established in various domains, the paradigm is still maturing and important challenges remain, to improve their reliability and unlock their full potential. This subsection exposes these challenges in order to motivate the contributions of the thesis, which will be listed in subsection 1.3 and later detailed in Section 3.

To understand why some of the following challenges arise, let us briefly introduce how algorithms of NE build LD representations of HD data, still avoiding technical details for now. As the name suggests, *neighbour embeddings* seek to preserve neighbourhoods around each point given a metric in the HD space and a neighbourhood size: where many DR methods seek

to capture some measure of spread throughout the whole dataset, NE flip this perspective and focus on the most local structures.
To do so, for each point, a set of neighbours is found in HD; then, a LD embedding is built such that neighbour sets in LD are similar to those found in HD. Building the embedding is an iterative process where points interact with each other through attractive and repulsive forces. Intuitively, neighbours in HD are attracted to each other in the LD space, and "intruders" in LD are pushed away. The dynamics are non-linear and the result is initialisation-dependent. The methods are often described as "non-parametric", meaning that the parameters being optimised are the coordinates themselves and not a function directly mapping HD vectors to LD. The process bears strong resemblances with n-body problems in physics and some methods to accelerate NE were in fact inspired by techniques used by physicists [32, 33].

A first challenge that users of NE face originates not from the methods, but from our brains. It is now well accepted that predictive processing plays a central role in biological cognition [5, 34, 35]. The brain constantly compares incoming sensory information with its internal predictions, discrepancies between the two allows to draw conscious attention to relevant stimuli and is likely a key mechanism in learning. This is also a core principle of active inference [36].
In the *The Predictive Mind* [5], the author produces a visualisation similar to the two leftmost scatterplots of the top row in Figure 1.2. There, the scatterplots show unbiased samples of points drawn from a uniform distribution in 2-dimensions. When scrutinised, the brain tends to see patterns in these random distributions, a phenomenon called *pareidolia*. In *The Predictive Mind*, the author explains that two brains can receive the same input, but the conscious experience will differ depending on prior knowledge and expectations. When we fail to minimise prediction errors, which tends to be the case when observing random noise, the brain overfits the input to probable patterns in order to align with prior beliefs.

The tendency of the brain to yearn for familiarity in its sensory information can be detrimental when analysing data visually, as this can lead to misinterpretations. The experience of sensory information can also be modulated by conscious effort, raising the risk of confirmation bias. The NE representations corresponding to the two leftmost scatterplots in Figure 1.2 are shown below them. There, clear clusters appear separated by

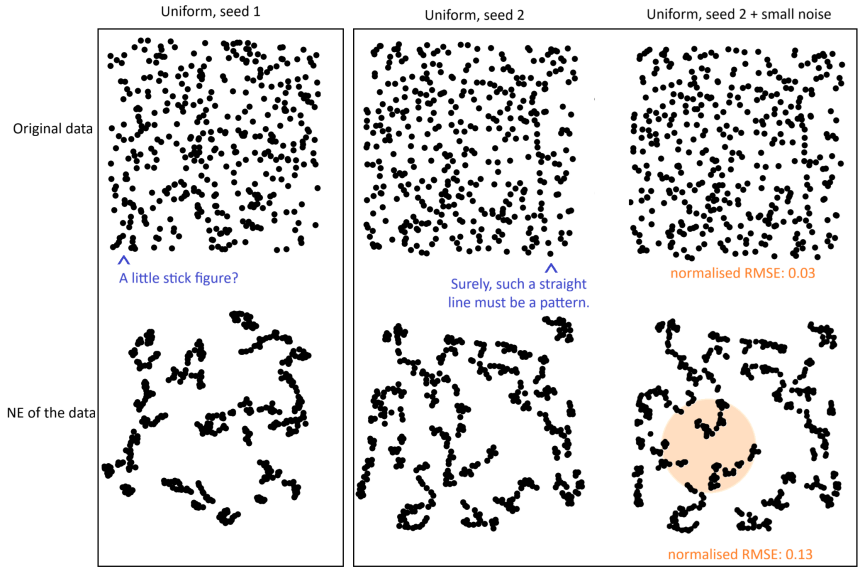


Figure 1.2: Example of how our brain biases perception, and how NE don't play in our favour in that regard. Top row: uniform distributions in 2-dimensions, the two rightmost ones share the same seed but differ by the addition of Gaussian noise. Bottom row: the NE representations of the scatterplots figuring above them, all runs share hyperparameters and were initialised by the initial data, which is already laying in 2-dimensions. The normalised RMSE scores refer to the errors between the two upper-right scatterplots and between the two bottom-right scatterplots, after min-max scaling. This measures how much the points have moved after application of noise: in this case, the noise has been greatly amplified by the application of NE. The uniform distributions can be reproduced in Python with the following code:

```
np.random.seed(2155800840) # or 2306102284
uniform2d = np.random.rand(400, 2)
```

large gaps. It is difficult to resist the urge to interpret these clusters as meaningful patterns, when in reality they are artefacts resulting from the inductive biases of NE methods. Section 4 explains that there seems to be a difficult to avoid trade-off in non-linear DR balancing their capacity to extract patterns from HD data with their tendency to introduce distortions (mainly: tearing, over-clustering, and changing the local densities).
In the rightmost column of Figure 1.2, slight Gaussian noise was added to the middle scatterplot of the top row, and a NE of this noisy data is shown under it. Visually, the noise is barely noticeable before application of NE, but some areas of the NE are strongly affected by the small changes in the input. This illustrates that, regardless of the brain's biases, NE methods

themselves can be sensitive to small perturbations in the input data and exaggerate these perturbations in the output. This result can come as a surprise to the practised reader, as it is generally accepted that NE methods are quite robust to noise [37,38]. An enquiry on this matter is presented in Annex 5.2.

Re-wiring human brains to be less prone to pareidolia and confirmation bias is often frowned upon due to ethical considerations; thankfully, general good practices can be followed to avoid falling into tasseography when analysing data using DR. As a rule of thumb, **it is recommended to visualise data through multiple points of view, using a variety of algorithms and hyperparameter configurations, and interacting with the data and algorithms in real time. It is also recommended to use quality assessment (QA) tools to support the visualisations.**

A second challenge arises naturally whenever DR is used on HD data for visualisation. When representing HD data in a 2- or 3-dimensional space, it is inevitable that some information is lost due to the lack of expressiveness of the LD space. For instance, if a 3-dimensional HD space features 4 equidistant points, it is impossible to represent these points in 2-dimensions while keeping them equidistant. The question when designing DR methods is: which information should be preserved, and which can be discarded?
In PCA, the objective is to preserve global variance, this comes at the cost of losing fine-grained structures; the loss of local details is generally shared by all non-sparse linear projections. In NE, information beyond the user-defined horizon of neighbours around each point is discarded. As a result, the local structures are well preserved, but the global organisation of the data can be heavily misrepresented, especially when the data is composed of well separated clusters in the HD space. Figure 1.3 illustrates this phenomenon on a dataset consisting of a curved 2-dimensional plane laying in a 3-dimensional space, forming a "S" shape. The leftmost representations are linear projections of the data and the rightmost ones are t-SNE embeddings. The colours reflect the values of a pointwise quality criterion: bright points indicate zones of low quality, and black points zones of good quality. The scalars under the embeddings indicate the mean value of the metric over all points, serving as a global summary. In the bottom row, the quality criterion is the Pearson correlation between pairwise distances in HD and LD, effectively measuring the preservation of global structures.

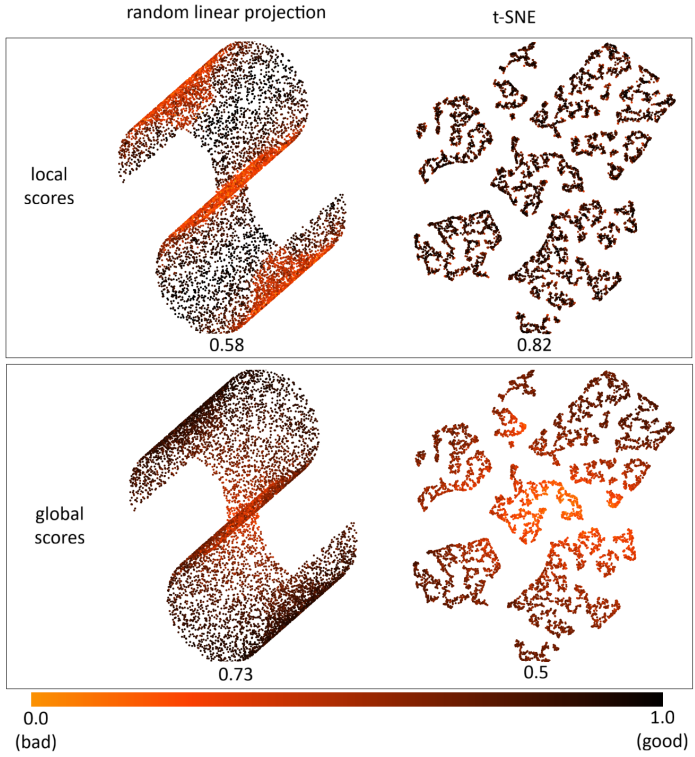


Figure 1.3: Comparison of a linear projection and a NE (*t*-SNE) of a curved 2-dimensional manifold laying in a 3-dimensional space. Top: the preservation of local structures is evaluated on each point using the $Q^i_{NX}(K)$ score for $K = 15$. Bottom: the preservation of global structures is evaluated using the Pearson correlation between pairwise distances in HD and LD.

The top row evaluates the preservation of the local structures: for a chosen neighbourhood of size K, calling the K-nearest neighbour (KNN) sets of the i^{th} point KNN_i^{HD} and KNN_i^{LD} in HD and LD respectively, the local criterion [39] at point i is:

$$Q^i_{NX}(K) = \frac{1}{K} \left| \text{KNN}_i^{\text{HD}} \cap \text{KNN}_i^{\text{LD}} \right|. \tag{1.1}$$

In Figure 1.3, the "S" shaped manifold is apparent in the linear projection, the global scores reflect this by being mostly high. However, the local structures are poorly preserved in the zones where the curve folds onto itself: some points that are close in LD are actually far apart in HD. This

type of error is called an *intrusion* or a *false positive*, these are quite common in DR methods that do not actively repel non-neighbours and can lead to erroneous interpretations. Such errors were also witnessed in the PCA embedding of the mouse brain dataset in Figure 1.1, where cells from within the three coarse categories were grouped together in blurry clusters.
In the *t*-SNE embeddings, the properties being preserved are flipped: the S-shape is entirely lost, but the neighbourhoods around each point are well preserved. Knowing what the actual HD structure looks like, we witness that the manifold has been ripped in multiple places. This is typical of NE and a consequence of their inductive biases responsible for the trade-off mentioned earlier: what enables NE to extract patterns from HD data is also what makes them prone to tearing the data in "areas of weakness". The inductive biases in question and what a NE considers as a "area of weakness" are explained and discussed in Sections 3.6 and 4.
Given how severely the loss of global structure affects the visualisation of such a simple dataset, the impact of this shortcoming in real-world HD data should not be underestimated. **This once again motivates the need for robust evaluation metrics and algorithms that can model global structures quickly, allowing the interactive setting necessary for effective exploration.**

As mentioned earlier, embedding HD to 2− or 3-dimensional spaces require decisions to be made on what to preserve and what to discard. Beside the selection of the algorithm itself, one panoply of knobs that users can turn to influence these decisions is the hyperparameters of the NE algorithm. Indeed, NE are particularly sensitive to hyperparameter changes, and each algorithm comes with its own sets of hyperparameters. One type of hyperparameter common to most NE methods is the scale at which the neighbourhoods are defined. Large neighbourhood scales tend to moderately help in the preservation of global structures at the cost of fine grained details and speed, and inversely for small neighbourhood scales. Figure 1.4 illustrates the effect of changing the neighbourhood scale and initialisations in NE of the mouse brain dataset. There, using larger neighbourhoods introduces a more distinct separation of points into large clusters of similar cell types, which makes biological sense [7]. However, the very local details aren't as refined as when using smaller neighbourhoods, as indicated by the $Q_{NX}(K)$ scores. In the bottom view, different initialisations lead to significant variations in the inter-cluster organisation.

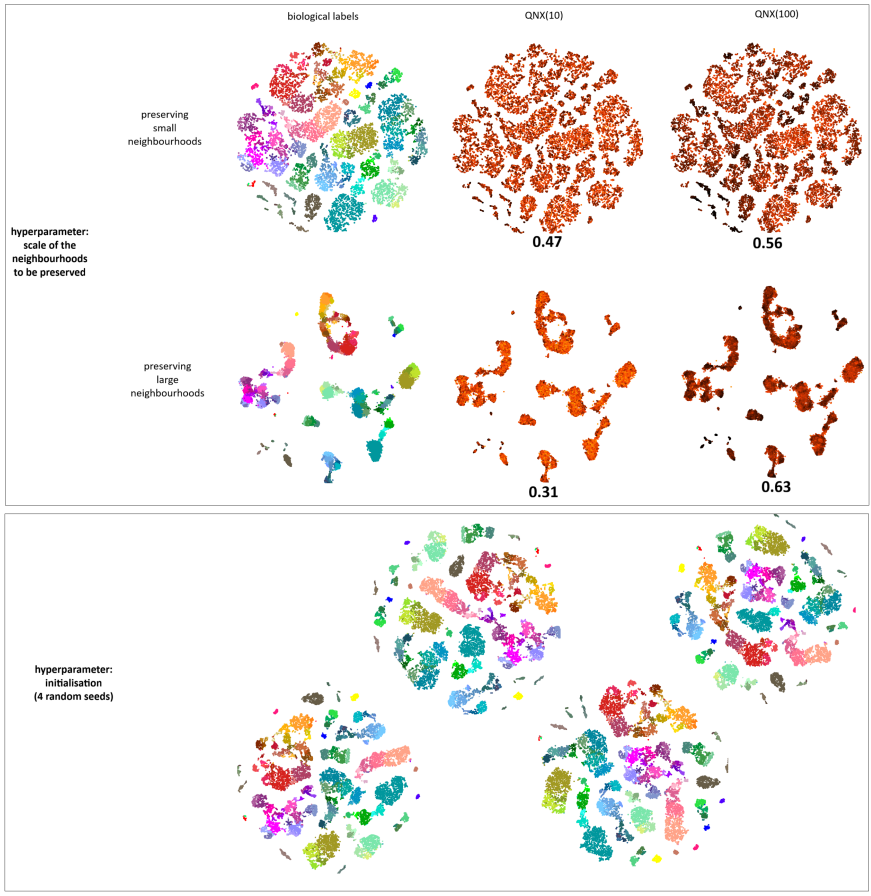


Figure 1.4: Effect of varying two hyperparameter values on t-SNE embeddings of the mouse brain dataset.
Top: effect of changing the neighbourhood scale (perplexity) on the embeddings. The local quality $Q^i_{NX}(K)$ are shown for $K = 10$ and $K = 100$, with the same colour code as in Fig. 1.3: bright colours indicate low quality. The average scores are indicated under each embedding.
Bottom: effect of changing the initialisation on the embeddings, keeping all other hyperparameters constant.

There are positives and negatives to having algorithms dependent on hyperparameters in data visualisation. Having an array of hyperparameters to tweak can be dangerous, as excessive fiddling can make any NE method produce any result, so to speak, rising the risk of confirmation bias. On the other hand, having a meaningful selection of hyperparameters to play with enables the user to visualise the data through multiple lenses, allowing the retrieval of complementary information and the cross-validation of the observed patterns. The benefits of this are greatly enhanced when the methods are integrated in interactive environments, allowing the user to explore data dynamically by changing hyperparameters on the fly or by modifying the data itself, for instance by filtering variables in HD. In order to allow such interactivity in NE, the algorithms must satisfy two properties.

First, the iterations during optimisation must be fast enough to allow for fluid visuals without compromising the quality of the results. The methods of NE which yield the results of highest quality in published studies (t-SNE variants like Barnes-Hut t-SNE [32] or FIt-SNE [33]) are quite fast and can reach computational complexities as low as $\mathcal{O}(\mathrm{N})$ for FIt-SNE, but in effect, they remain slower than the fastest methods like UMAP [40], which sacrifice quality over speed [41–43]. This is elaborated on in Section 2.2 and in 3.6. Therefore, **there remains a need for NE methods that have high effective speed as UMAP does, but without sacrificing too much in the quality of the results**. Advances in measuring methods and coupled with the increasing number of use-cases for NE mean that large datasets of more than hundreds of thousands of points are becoming common, further emphasising the need for speed. To further drive the point home, let us elaborate on why having fast iterations is crucial in NE, the benefits are double:
There is the comfort aspect, which is not to be underestimated: a smooth experience allows the user to focus on the task at hand without getting distracted or frustrated by technical limitations. In turn, this allows for a more effective utilisation of the user's working memory, as information tends to be forgotten or warped as time passes. In Figure 1.5, the movement of points throughout the iterations can be quite complex and are far from straight lines. This hints that the learning rate must be limited to keep the dynamics stable: NE rely on large numbers of small steps rather than on a few large steps, so these steps ought to be quick.
The second reason why having fast iterations is important is that, when an embedding reacts smoothly to user inputs, the movement of points act

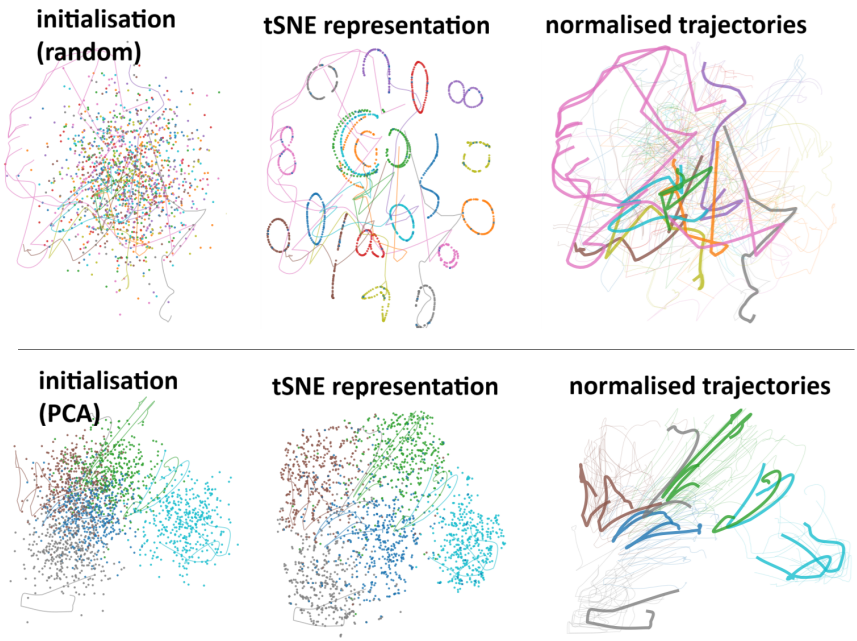


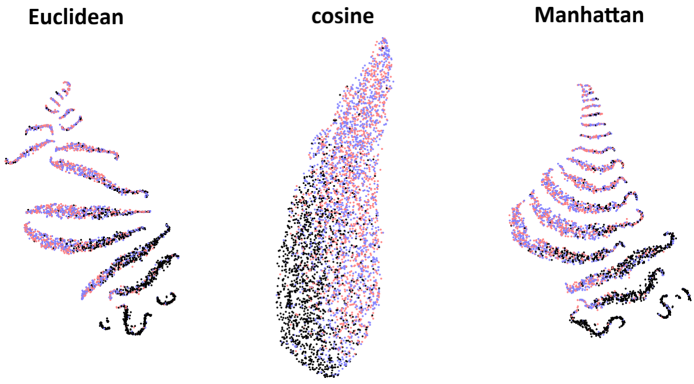
Figure 1.5: Trajectories taken by some of the points during the optimisation of a NE, from their initial positions to their final coordinates, on the COIL-20 and blobs datasets. The trajectories are normalised to account for changes of scales between the initialisation and the final embedding. In this experiment, the blobs dataset consists of 2000 points laying in a 10-dimensional space. Both NE were stopped after 1000 iterations.

as an additional effective dimension to the visualisation, a concept that has been explored in some works like [44]. Indeed, our brains are particularly well adapted to perceive motion, and movement allows us to see patterns that would be difficult to spot in static images. It is not rare that a user-driven change in hyperparameters or in the data leads to targeted movements of points, these immediately stand out in contrast to the static background drawn by the other points.

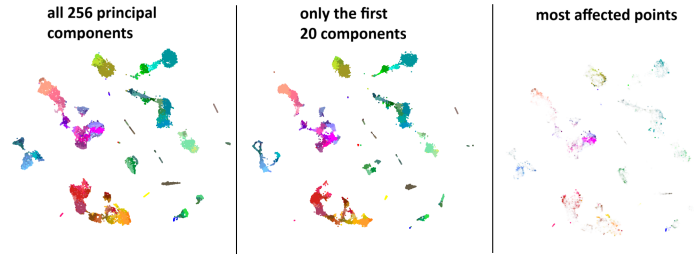
All NE methods lack a second property necessary for effective interactivity: seamless reactivity to changes in hyperparameters or in the data itself. As will be described in detail further on, up until now, all NE methods have in common that they separate the initial computations taking place in the HD space from the optimisation loop taking place in the LD space. This seems reasonable, as the optimisation loop requires information from the

HD space if the points in LD are to be guided toward meaningful representations of the HD structures. This approach, however, leads to significant overhead when modifying the data in HD (filtering variables, adding new points, ...), or when changing hyperparameters that impact the HD side of the algorithm like the neighbourhood size or the distance metric used to determine these neighbourhoods. This precludes smooth transitions after taking an action.
Static images aren't the ideal medium to illustrate the benefits of having smooth movements and responsive algorithms, nevertheless, Figure 1.6 attempts to motivate the need for responsive algorithms. In the top view, 3 NE of the abalone dataset are represented. In each one, a different distance metric in HD was used to determine neighbourhoods. Marine gastropod enthusiasts would without doubt revel in witnessing the clusters merge and organise across the three configurations. When visualised with a method developed in this thesis introduced in Section 3.6, passing from one configuration to another takes only a second, allowing the user to move back and forth between the configurations and develop a sense of which points and groups of points move in concert. In the bottom view, NE of the mouse brain dataset are shown. These were performed with the same new method, which allows for instantaneous visual feedback when changing the data in HD or hyperparameters. There, the leftmost embedding uses all 256 principal components extracted from the data, the middle one uses only the first 20 components, the rightmost embedding shows which points reacted most to the perturbation. Comparing the two leftmost embeddings here is quite tedious necessitates conscious effort, whereas in the visualisation software accompanying the method, going from one configuration to the other is a matter of 1 or 2 seconds, and information on which parts of the clusters shift and how they deform is assimilated effortlessly.
Additionally to the two-phased approach of the algorithms, the optimisation process itself is often divided into multiple phases with specialised dynamics. For instance, t-SNE relies on an initial "early exaggeration" phase [10], where the cost function is modified to allow points to move more freely and often quite explosively. If a change in hyperparameters or in the data results in movements in the embedding, re-starting this exaggeration phase would induce sudden and large movements in all the points, breaking the current structures and disorienting the user rather than helping them explore the data. **This motivates the need for NE methods that distance themselves from the typical approach separating the**

initial computations from the optimisation loop, and optimisation strategies that do not rely on multiple phases.



(a) Effect of changing the distance metric used to determine the neighbourhoods in HD, on the NE of the Abalone dataset. From left to right, the distance metrics are: Euclidean, cosine (for the i^{th} and j^{th} of the data matrix X, the cosine distance is defined as $d_{ij}^{(\cos)} = 1 - \frac{\mathbf{x}_i^\top \mathbf{x}_j}{\|\mathbf{x}_i\|_2 \|\mathbf{x}_j\|_2}$), and Manhattan (L1 norm of $X_i - X_j$).



(b) Effect of filtering out the last 236 components of the 256 principal components extracted from the mouse brain dataset, on the NE of the data. The leftmost embedding uses all 256 components, the middle one uses only 20 components, and the rightmost one shows the points most affected by this change by making the less affected points transparent.

Figure 1.6: Illustration of effects of (a) changing the distance metric used to determine neighbourhoods in HD, and (b) changing the data in HD, on NE embeddings. These particular changes require significant re-computations in typical NE algorithms, inducing delays in the response.

The final challenge mentioned here is not a shortcoming of NE per se; quite the opposite, it arises from the frustration of observing that NE may

have fundamental lessons that could benefit ML beyond visualisation, yet this potential seems to remain largely untapped or only used superficially rather than being integrated as core principles in ML algorithms.
Let us illustrate this claim with the case depicted in Figure 1.7. There, in the "ground truth labels" view on the top, two 2-dimensional representations of the MNIST handwritten digits dataset are shown, one produced by PCA (left) and one by a NE algorithm (right), with points coloured by their actual digit class. As tends the be the case with PCA, the classes are mixed together without clear boundaries. *It is important to realise that, since PCA is a linear projection, this cloudy organisation is an incomplete but undistorted view on how the data truly lies in the HD space: while a NE algorithm represents points by looking at how these integrate themselves in their surrounding local structures, PCA considers how the points place themselves relative to dataset-wide directions*. On the right of the PCA representation, the NE depicts 10 well separated clusters; if one were to ignore the colours, separating the clusters visually would also closely align with separating the digit classes. This ease of visual classification in the NE also applies to automatic processing, as witnessed in the bottom view of Fig. 1.7 and in Table 1.1.

In the bottom view of Fig. 1.7, two clustering algorithms (K-means and DBSCAN [45], both staples in ML) were applied on different representations of the MNIST dataset: the high-dimensional pixel values (HD space of 784 dimensions), a 2-dimensional PCA representation, a 10-dimensional PCA representation, and a 2-dimensional NE. Despite MNIST being considered as an easy dataset in ML, the cluster assignments and the high mean Gini impurity scores indicate that clustering algorithms struggle to extract meaningful clusters in the HD space or in PCA representations (assuming the labels correspond to true distinctions in the data, which seems to be the case when looking at the NE). This is a well known shortcoming of clustering algorithms that can be explained by the concentration of norms which tends to happen in HD spaces [46–48]. It is the phenomenon where, in the absence of strong structure, all points tend to become "more and more equidistant" as the dimensionality grows. The problem is significantly mitigated when using NE as the representation in which to cluster the data. The advantage of first applying NE before clustering is now well known in the community and has been adopted in many works such as [49,50].

Clustering algorithms often fail to disentangle meaningful structures

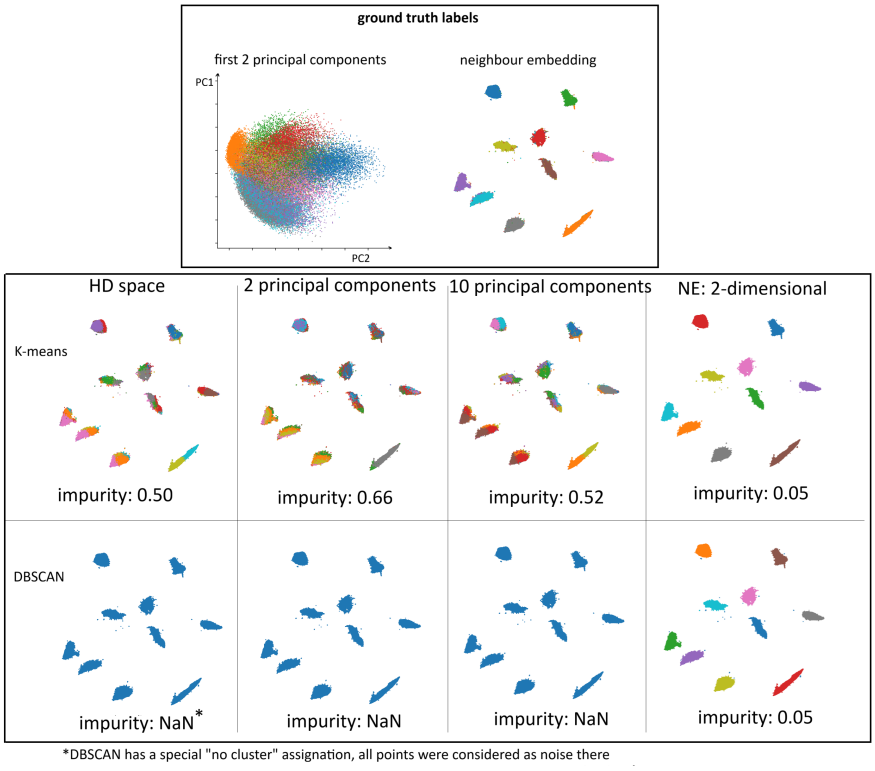


Figure 1.7: Top view: 2-dimensional PCA representation and NE of the train set of the MNIST dataset, coloured by digit value.
Bottom view: K-means and DBSCAN [45] clustering results on different representations of the MNIST dataset: raw high-dimensional pixel values (HD space), PCA in 2-dimensions, PCA in 10-dimensions, and NE in 2-dimensions. The colours indicate the cluster assignments and are independent from the per-digit colours of the top view. The scalars under each embedding show the mean Gini impurity of the clusters, measuring how mixed the digit classes are within each cluster (lower is better). The DBSCAN hyperparameters were kept to default values, the K-means algorithm was run with $K = 10$ clusters and uses Euclidean distances.

in data without appropriate representations, however, one might wonder if classification algorithms would fare better, perhaps helped by the additional information contained in the labels. Table 1.1 shows the accuracies of three classifiers: random forests (RF), support vector machines (SVM), and K-nearest neighbours (KNN), in a transductive one-shot learning scenario. The classifiers are trained on three representations of the MNIST dataset: the raw HD pixel values, a 2-dimensional PCA representation, and a 2-dimensional NE. There, only one label per class is revealed and the labels of the remaining points are inferred from it using the unsupervised information of the full dataset and the supervised information from the 10 points of known labels. The random forest and SVM classifiers were trained in a semi-supervised fashion using scikit-learn's "self training mode" [51]: in this configuration, training is performed in multiple rounds; in each round, the points with the most confident predictions are added to the support set originally composed of the 10 labelled points and the models are trained again using the extended set.
The table shows that in scenarios with limited supervised signals, classifiers also struggle to extract meaningful information from HD data or PCA representations, but perform much better when using NE as the representation. It should be noted that the accuracy of the KNN classifier, over 95%, is quite remarkable considering that only 10 labelled points were used for training and that the raw pixel values were used to perform the NE. Such high scores are generally achieved with stronger supervised signals and using specialised architectures like convolutional neural networks (CNNs) [52]. This indicates that, while CNNs allow the extraction of meaningful features from individual images, the NE approach, agnostic to the structure of the variables in HD, can leverage the structure drawn by the data points in HD directly to disentangle the classes effectively.

Theoretical foundations have been laid explaining the similarities between NE and spectral clustering when attraction forces applied to points are exaggerated [53], and there is some empirical understanding of the configurations in which NE naturally tend to separate clusters [54]. In fact, when teaching principles of NE to students in data science, many refer to NE as clustering algorithms, and perhaps they are right. However, **the principles of NE enabling such robust disentanglement of overlapping structures in HD data are yet to be isolated and integrated to new ML algorithms designed to handle HD data in low supervision contexts.**

Data Type	RF	SVM	KNN
High Dimensional (HD)	0.20 ± 0.04	0.10 ± 0.00	0.43 ± 0.03
PCA (2D)	0.26 ± 0.05	0.15 ± 0.02	0.32 ± 0.03
t-SNE (2D)	0.87 ± 0.04	0.58 ± 0.18	0.95 ± 0.04

Table 1.1: Classification accuracies ($\pm$ std) of three classifiers (Random Forest - RF, Support Vector Machine - SVM, K-Nearest Neighbours - KNN) in a transductive one-shot learning scenario on three representations of the MNIST dataset: the raw high-dimensional pixel values (HD), a 2-dimensional PCA representation, and a 2-dimensional t-SNE representation. The mean and standard deviation are computed over 7 random selections of one labelled point per class, the labels of the remaining points being inferred.

Regardless of the successful distillation of NE principles into new ML algorithms, using NE directly as a feature extraction step for ML is conceptually simple and the practice appears to withstand the trials of empirical scrutiny. **The problem is, to effectively use NE as preprocessing, new data points must be mapped to their LD coordinates with low overhead**.

One way to achieve this is to train parametric models to directly learn the mapping from HD to LD. In the non parametric case, the mapping can be described as a recursive function involving the whole dataset where the LD coordinates at iteration t are a non-linear function of the LD coordinates at iteration $t-1$ and of the HD coordinates. Due to the highly non-linear nature of this mapping, deep neural networks are generally used when attempting to approximate the process as a single-pass function [55, 56]. Performing a direct mapping by using deep learning proves particularly challenging, and the parametric nature of these approaches tends to take a toll on the quality of the local structures in the resulting embeddings [56, 57]. In [56], the authors claim comparable performance between parametric and non-parametric UMAP, but the UMAP algorithm itself systematically shows poor preservation of local structure when assessed objectively [41–43], so the bar is set quite low.

One reason why parametric NE is so difficult to achieve may be that the usual high dimensional mappings typically learned in deep learning applications are not required to be as intricate. For instance, when classifying data points through a final softmax-activated layer, small perturbations do

not drastically change the predicted class (softmax of $[0.0, 0.0, 2.5]$ is sensibly the same as softmax of $[0.1, -0.1, 2.4]$, and the slope of the function is gentle), whereas when performing NE, a small perturbation in LD, which could originate from the previous gradient descent step, can significantly alter the local structures around a point, leading to large gradients which will further break the local structures in the next forward pass. The original t-SNE paper [10] shows the gradient landscape as a heat-map, and there, indeed, the topography can quickly increase in steepness with small changes in LD.
Another difficulty may arise from over-parametrisation: in "non parametric" NE, the positions of the points in LD are the parameters being learned, these are are somewhat self-regulated as neighbours in HD tend to stick together, and, importantly, they are much less numerous than the parameters in a typical deep neural network, which can result in overfitting.
One last difficulty may stem from batch effects: in typical deep learning applications, the focus is on minimising a loss function which only considers the intricate details if necessary. In NE, the cost function is entirely built around the fine-grained details of the neighbourhoods, so batches for stochastic gradient descent would not capture these structures if sampled uniformly at random. If the batches are sampled to capture specific neighbourhoods, border effects can arise, and the specificity of the gradients applied to a given neighbourhood can have detrimental effects on other neighbourhoods in future batches, a common problem in deep learning at the root of catastrophic forgetting [58, 59].

Another approach which would allow the use of NE as preprocessing for ML would be to design new methods allowing the embedding of points with low overhead, in the non parametric case. Some works propose embedding new points through fast interpolations using their neighbours in HD [8], but these approaches risk failing to capture the true positions of the points if they were to be embedded through a full optimisation process. The complex non-linear dynamics at hand can yield point coordinates that are strikingly different from what a simple interpolation would predict, as witnessed in Figure 1.8 where the HD neighbours of selected points are shown in the NE of a subset of the MNIST dataset. In addition, allowing new points to integrate by participating fully to what would be a perpetual optimisation process would allow the use of the unsupervised information contained in the new points to refine the embedding further, and possibly adapt to drifts in the data distribution.

A solution which would correctly embed new points without significant overhead would also align with the need for responsive algorithms by allowing changes in HD to reflect in LD. Therefore, **designing NE methods capable of embedding new points with low overhead remains an open challenge**.

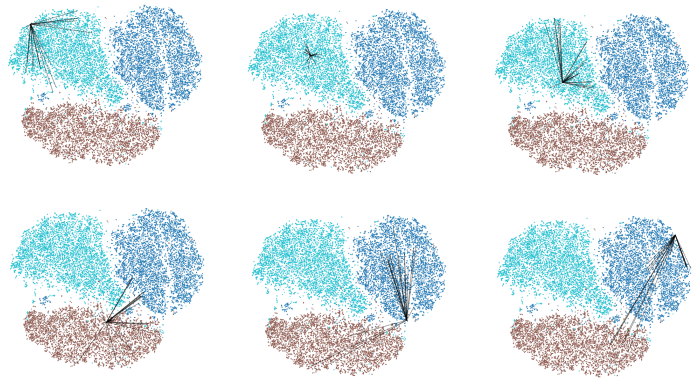


Figure 1.8: Embeddings of digits 3, 5, and 8 from the MNIST dataset using a NE algorithm. In each view, a point was selected and its 30 neighbours computed in HD are shown as black lines connecting the selected point to its neighbours.

Another shortcoming of current NE methods regarding their applicability to pre-process data for further algorithmic processing is their limited scalability to LD dimensionalities greater than 3. Indeed, the methods that perform best in terms of quality [32, 33] when evaluated objectively [41–43] do not scale well to larger dimensionalities in LD, in fact, their time complexity grows exponentially with the dimensionality in LD. Some fast methods like UMAP [40] scale linearly to the dimensionality in LD, but at the cost of quality. Therefore, if distancing NE methods from visual applications, **mapping points to intermediate dimensionalities greater than 3 without sacrificing in quality remains a challenge which would allow more expressive embeddings for ML applications**.

To summarise, the field of NE could benefit from a multitude of improvements to better serve both visualisation and ML applications. These improvements include:

- Fast QA tools for integration in interactive environments.

- Fast methods that capture global structures, either as stand-alone methods or as improved NE algorithms.
- NE methods that seamlessly adapt to changes in hyperparameters or in the data itself (filtering variables, adding new points, reacting to drifts, ...).
- Optimisation strategies that do not rely on multiple phases, allowing warm restarts or continual learning.
- NE algorithms capable of embedding points to intermediate dimensionalities greater than 3, without sacrificing quality.
- Learning exactly what allows NE to so effectively disentangle overlapping structures in HD data, to integrate these principles in new ML algorithms.

1.3 Overview of the contributions

This section outlines the contributions of the thesis to the domain of NE and more generally to machine learning. Each contribution mentioned here is detailed in Section 3. This thesis was initially centred on data visualisation through NE. However, as the work progressed, new insights and opportunities arose, shifting the focus in the latter part of the thesis towards broader machine learning applications.

The first contribution of this thesis is modest in scope, it consists of assessing if an accelerated method [60] for multi-scale t-SNE [61] is sensitive to randomness, as it relies on random subsamples of the data to achieve its speed-up. The study was accepted as a poster in a conference [62]. The accelerated method has a $\mathcal{O}(N \log^2 N)$ time complexity, which makes it difficult to use in interactive environments.

This thesis started a couple of years after the development of the first fast NE methods exhibiting $\mathcal{O}(N)$ time and space complexity [33,40], opening their use to very large datasets. Since the lack of global structure preservation in NE can be dangerous when visualising data, the newfound speed of this family of algorithms required fast complementary methods that can preserve global structures effectively.
It is in this context that the second contribution of this thesis, and first

of significance, was to devise a fast and straightforward Multidimensional Scaling (MDS) algorithm [63] with low complexity, allowing very fast visualisation of global structures in large datasets. The algorithm, working on samples of 4 points (quartets), is coined **SQuaD-MDS**, short for *Stochastic Quartet Descent MDS*. The method was published and presented as a talk in a conference [64], and the paper is included in Section 3.2.

Since both SQuaD-MDS and the fastest NE methods shared a $\mathcal{O}(N)$ computational complexity and both relied on gradient descent to minimise their cost function, it seemed opportune to combine the two approaches into a hybrid method capable of preserving both global and local structures while remaining fast. This led to the third contribution of this thesis, a **fast hybrid MDS-NE algorithm**. The method was published in a journal [41], the paper is included in Section 3.3.

Neighbour embeddings embed points non-parametrically through non-linear dynamics, yielding LD representations where the relationship between the LD structures and the variables in HD is difficult to apprehend. This effect is exacerbated by the fact that classical NE methods do not capture global structures, resulting in embeddings where the proximity and orientation of close-by but distinct structures do not necessarily correspond to a straightforward relationship in HD. Having fast NE methods that preserve both global and local structures can result in more interpretable embeddings by inducing some continuity throughout the representation.
The fourth contribution of this thesis is to devise a **post-hoc explainability method for NE embeddings**, where a multitude of little axes superimposed on the embedding indicate directions in which HD variables vary the most. This method was sometimes called "global-local" during development, because, while the axes allow local explanations in the embedding, when considering multiple axes throughout a region in LD, a global picture of how variables behave in HD can be understood at a glance. The method was presented as a talk in a conference [65], the paper is included in Section 3.4.

At this point, a NE method of complexity $\mathcal{O}(N)$ capable of preserving both global and local structures has been developed, with tools helping interpret the embeddings. As section 1.2 outlined, the typical t-SNE variant has 2-phases in its optimisation loop, relying on an initial *early exagger-*

ation phase. This is detrimental to interactive visualisation or continual learning scenarios, as the exaggeration strongly modifies the cost function landscape, leading to widespread sudden movements of points when enabled, breaking the structures in LD. The fifth contribution of this thesis is to show that **NE can be optimised without relying on early exaggeration, by using strong momentum and the Nesterov trick [66], bringing the first component to enabling seamless adaptations of NE to user inputs.** This work was accepted as a poster in a conference [67], the paper is included in Section 3.5.

In practice, despite sharing the same $\mathcal{O}(\mathrm{N})$ time complexity, methods estimating the repulsive forces exerted on each point through unbiased negative sampling like UMAP [40], are faster than methods estimating these forces accurately like FIt-SNE [33], which also happens to be the method used for the NE part of the hybrid MDS-NE method. Moreover, FIt-SNE scales exponentially with the dimensionality in LD, restricting its use to 2 or 3 dimensions; this is fine for visualisation purposes, but it limits it's horizon of applications. As a trade-off to these limitation, methods that precisely model the repulsive field throughout the embedding systematically outperform the other methods in terms of local structure preservation when evaluated objectively [41–43].
Ideally, one would want a method as fast as UMAP, but with the quality of FIt-SNE or BH-t-SNE, preferably without restrictions on the dimensionality of the LD space. Better yet, as explained in Section 1.2, the ideal method should adapt seamlessly to changes in hyperparameters or in the data itself, a property that is often overlooked despite being central for interactive visualisation or online applications.
The sixth contribution of this thesis is to **rethink how to perform NE, introducing an algorithm exhibiting high effective speed, good structure preservation, unconstrained dimensionality in LD, and seamless reactivity to changes in hyperparameters or in the data itself**, this work is detailed in Section 3.6. The method was first introduced in its prototypical form as a talk in a conference [68], and a more refined version with extensive experiments is now under review for the journal *Neurocomputing* [42]. The first paper is not included in this document, to spare space, but the second one is included in Section 3.6. The method is named *FUnc-SNE*, short for *Fast Unconstrained Stochastic Neighbour Embedding*.

The *FUnc-SNE* method addresses many of the challenges outlined in

Section 1.2, allowing unprecedented interactivity between the user, the data, and the algorithm. In addition, the capacity of the method to embed points to dimensionalities greater than 3 while keeping good structure preservation brings new potential to the use of NE in broader ML contexts, beyond visualisation. It is with this consideration in mind that the paper of Section 3.7 was presented as a talk in a conference [69]. The paper does not invent the wheel, it merely claims that NE have now sufficiently matured to be used directly in various ML tasks, potentially to great effect.

The final contribution mentioned in this thesis is **an algorithm which transposes some of the NE principles to label diffusion, greatly improving accuracy in low-supervision scenarios**. Its strong performance and potential implications justify a publication whose preparation should not be rushed. Therefore, this contribution is only outlined in Section 3.8 without an accompanying paper. This method contributes directly to machine learning by enabling transductive learning under limited supervision, such as in so-called zero-shot and one-shot learning. Interactive visualisation can also directly benefit from this method, where classes can be defined dynamically and visualised as diffusing colours in the embedding space. Indirectly, this approach may inspire more label-efficient algorithmic designs in general.

Chapter 2

Dimensionality Reduction and Neighbour Embeddings

To focus on the challenges faced by NE methods from a practical standpoint, the previous chapter avoided detailing the technical aspects of DR methods. This chapter formalizes the DR methods encountered throughout this thesis, and introduces additional properties that naturally arise from their mathematical formulation. Some redundancy with the introductory chapter is difficult to avoid, this chapter will attempt to keep the overlap minimal. Since the papers included in Section 3 already each pose the relevant context for their respective topics, this chapter doesn't aim to bring an exhaustive review of DR and NE methods.
Section 2.1 briefly describes the main non-NE methods appearing in this thesis. Section 2.2 then focuses on NE methods.

2.1 Dimensionality reduction

This Section briefly reviews the DR methods that are mentioned in this thesis, excluding NE methods which are detailed in Section 2.2.

Dimensionality reduction methods aim to represent HD data in a LD space. In the context of data visualisation, the LD space is restricted to 2 or 3 dimensions. Often-times, datasets observed as vectors of dimensionality M actually contain simpler structures of lower dimensionality $M' < M$, this is the notion of *intrinsic dimensionality*. This principle is explored in Annex 5.1. When the data expresses an intrinsic dimensionality M' larger than the LD space dimensionality, the process of DR inevitably leads to some loss of information. The multitude of DR methods originates from the necessity to chose what information to preserve when reducing the dimensionality of the data.
In addition to the property being preserved, DR methods separate into linear and non-linear methods. Non-linear methods can be further split into so-called parametric methods and non-parametric methods: in the parametric case, a function is learned to directly map HD vectors to LD vectors, while in the non-parametric case, the LD vectors are learned without an explicit encoding function.

2.1.1 Principal component analysis

In biological neurons, the learning dynamics are much more complex than in the deep learning models used in ML. In the latter, the transformations of data from input to output are fully differentiable, and the objective function is well defined; this allows end-to-end training through backpropagation. In biological networks of neurons, signals are transmitted through punctual events, the network is highly recurrent, the cost function is hard to define, time is central to the dynamics which introduces the notion of causality and the problem of temporal credit assignment, and long range signals (in time and space) are sparse, making coordination difficult; moreover, all the mechanisms at hand are implemented through interconnected chemical and biophysical processes [70,71]. In addition, neurons are constrained by energy consumption, space, and the speed of signal transmission.
These constraints lead to a large part of learning in biological networks be-

ing local: synaptic weights are updated based on information available in the space surrounding the synapse, through associative learning rules. If ignoring global signals and simplifying the principles of associative learning to its core, one can describe the local learning in neurons through the famous sentence "neurons that fire together, wire together", first pronounced by Donald Hebb [72]. Formalising this principle, a linear neuron having it's incoming synaptic weights organised in a vector $\mathbf{w}$, receiving an input $\mathbf{x}$, and responding to the stimulus with an activation $a = \mathbf{w}^\top \mathbf{x}_i$, would update its weights proportionally to: $\dot{\mathbf{w}} = a\,\mathbf{x}_i$; with regularisation to prevent runaway growth.
This lesson in biology seems far from the subject of DR methods at first glance, however, an interesting phenomenon arises when using Hebbian learning on a stable distribution of inputs: if the learning rate is small enough, the weights $\mathbf{w}$ will align with the direction of greatest variance in the input distribution, a relation formulated by Oja [73]. This means that a single linear neuron can find the "point of view", or direction, that allows it to best discriminate between samples from the input distribution, through the application of a simple local learning rule. Better yet, if multiple neurons are experiencing the same distribution of inputs while being interconnected through lateral inhibition [74], which is a ubiquitous pattern in nervous systems, the dynamics will lead to neurons that specialise to different, mostly orthogonal, complementary directions of greatest variance [75]! This means that, if the input signal is $M > 2$ -dimensional and there are only 2 linear neurons to observe it, if the neurons are connected together by inhibitory synapses, then they will represent the signal in a way that best extracts the variance on uncorrelated axes (their respective activations), effectively performing DR on the data.

This principle of finding orthogonal directions of greatest variance is called *Principal Component Analysis* (PCA) [9], a DR method predating programmable computers. In practice, the principal components, or axes of greatest variance, correspond to the top eigenvectors of the covariance matrix of the centred sample. To obtain the LD embedding, the centred HD data is simply projected onto these vectors.

In ML, PCA is valued as a preprocessing tool for its speed, parametric nature, and guarantees: it is common to plot the cumulative variance explained in a projection as a function of the number of principal components kept for the projection. This way, a practitioner can chose the target

dimensionality of the projection based on a pre-defined variance threshold rather than on a number of components, allowing for a more data-driven approach.
In the context of visualisation, PCA is either used as a preprocessing step before use by other methods [8], or used directly to visualise data, generally on the first 2 or 3 principal components. Because of the square term involved in the variance, PCA tends to capture large variations in the data, extracting global structures before local structures; this also makes it particularly sensitive to noise, outliers, and heterogeneous sampling. Furthermore, being a linear method, PCA is limited in its ability to capture non-linear relations in the data, often requiring many more dimensions to represent the data faithfully than non-linear methods. This is particularly problematic in visualisation, where the target dimensionality is very small.

2.1.2 Multidimensional scaling

A generally better choice for extracting global structures in visualisation is *Multidimensional Scaling* (MDS) [63, 76]. Multidimensional Scaling can take multiple variations; in this thesis, it will mostly be considered in its metric form, and the abbreviation *MDS* will refer to this variant. In metric MDS, the process of DR is tackled through the preservation of pairwise distances between samples. Considering a set of N samples in HD, let δ_{ij} be the distance between the i^{th} and j^{th} samples in HD according to a user determined metric, and d_{ij} be the Euclidean distance in the LD space; MDS aims to find coordinates in LD that minimises what is often called the *stress*:

$$\text{Stress} = \sum_{i \neq j} \left(\delta_{ij} - d_{ij}\right)^2 \quad . \tag{2.1}$$

Some MDS variants like *Sammon mapping* [77] introduce a normalisation factor to the residuals, giving more importance to the distances that are small in HD. Other methods introduce a distinct weight to each squared residual, filtering or scaling some contributions. For instance, in *Isomap* [78], only the distances between K-nearest neighbours are considered, resulting on a geodesic approach to distance based DR. Other variants relax the problem by working with dissimilarities rather than distance metrics in the HD space.

In practice, MDS is more computationally expensive than PCA as it re-

quires iterative optimisation and the cost function involves N^2 terms. Like PCA, MDS tends to capture global structures before local structures if the LD dimensionality is constraining. Unlike PCA, MDS is not constrained to linear projections, and can capture more information per dimension in the LD space.
However, as explained in Section 1.2, when the dimensionality of the data rises, the distribution of pairwise distances tends to concentrate [46–48], rendering "all points increasingly equidistant". This phenomenon can be observed in Figure 2.1, where, in the top part, the distribution of pairwise distances in HD is shown for the S-curve dataset, the mouse brain dataset, and the MNIST dataset. There, the 3-dimensional S-curve dataset has more spread in its distribution of distances than the distribution in the MNIST dataset, which resembles a tight Gaussian bell. This is problematic when attempting to represent MNIST in very LD spaces through distance preservation, as the number of points that can be equidistant in a space decreases with the dimensionality of the space. This intuition verifies when looking at the MDS embeddings of Figure 2.1, where clear structure is only visible in the S-curve dataset and in the mouse brain dataset, while the MNIST MDS embedding is mostly a homogenous cloud of points. It is interesting to note that the mouse brain dataset, despite having a high ambient dimensionality (here, the HD space consists of the first 256 principal components of the data), appears to exhibit structure allowing some level of detail to be captured by the MDS embedding, hinting at a low intrinsic dimensionality (refer to Annex 5.1 for an empirical study on intrinsic dimensionality).
When observing the t-SNE embedding of MNIST in the same figure, one can see that NE can be particularly resilient to the concentration of norms, the next section explores how NE methods achieve this.

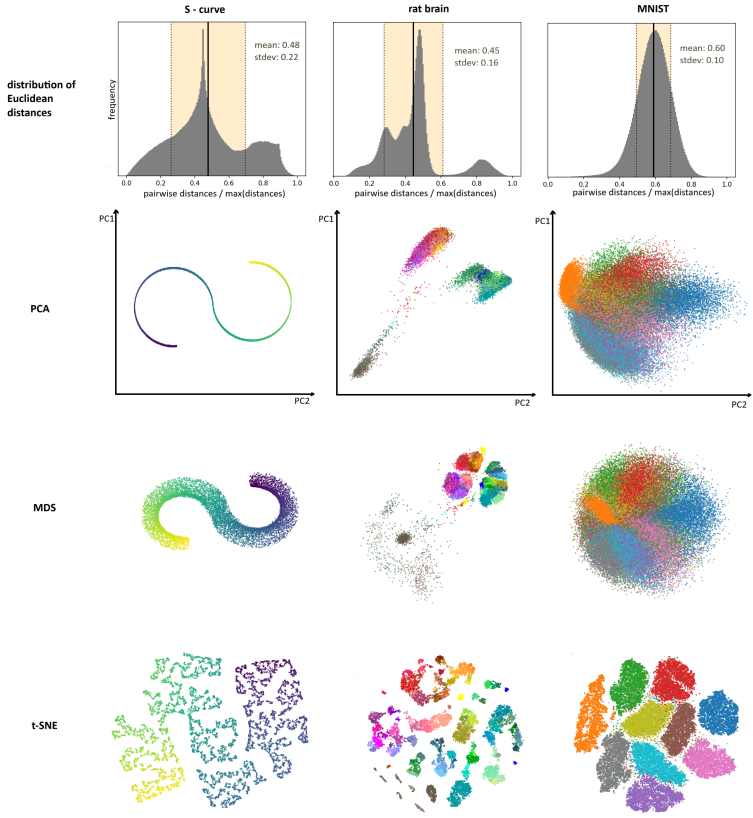


Figure 2.1: Top row: distribution of pairwise distances in HD for the S-curve dataset (left), the mouse brain dataset (middle), and the MNIST dataset (right). The distances are normalised by dividing with the maximal distance in each dataset; the mean and standard deviation are also indicated. Remaining rows: LD embeddings produced by PCA (2nd row), MDS (3rd row), and t-SNE (4th row) on the same datasets.

2.2 Neighbour embeddings

2.2.1 Technical aspects

The shortcomings of linear methods like PCA and distance preserving approaches led to the exploration of other ways to robustly represent HD data in LD. One foundational paradigm that might have paved the way to the formulation of NE are *self-organising maps*, first introduced by Kohonen in 1982 [79]. Intuitively, self-organising maps can be understood as fitting

an elastic grid of predetermined topology to the HD data, potentially modelling underlying lower dimensional structures. Each node in the grid, or neuron, is associated with a vector in the HD space. When a HD observation is presented to the self-organising map, the closest neuron according to a metric moves towards the observation and, as it moves, its neighbours in the grid are dragged alongside it.
Self organising maps thus build around a notion of neighbourhoods in a LD space, and points that are neighbours in the HD space will tend to also share a neuron or be inside the neighbourhood of a particular neuron in the grid space after fitting the model. While the winner-takes-most approach of self-organising maps allows the extraction of meaningful structures in HD data despite the concentration of norms, their uses in data visualisation depend on design choices such as the grid topology and granularity, and they do not map each point in HD to a distinct location in a continuous LD space.
In contrast, *neighbour embeddings* do not rely on choosing a topology and define neighbourhoods around each point in HD rather than on prototypes in a LD grid. This section formalises concepts central to NE.

Neighbour embeddings were first formalised by Hinton and Roweis with *Stochastic Neighbour Embedding* (SNE) [6], the authors might have conceptualised the method through the following consideration. While distances tend to lose contrast in HD and their distributions are not compatible with LD spaces, limited neighbourhood relations may allow for better discriminative power and, by reducing the number of constraints to satisfy, may be easier to represent in LD. This is the approach taken by NE methods.

A problem immediately arises when working with neighbourhoods instead of a continuous scale: how to optimise the cost function? Neighbourhoods are discrete sets, if fixing all the points but one in LD and moving this point along a direction, the error in neighbourhoods preservation between the LD and HD space would vary in a stepwise manner, precluding the use of gradients. A straightforward solution might be to use evolutionary algorithms, perhaps leveraging the continuous nature of the LD space through differential evolution [80]. However, the large number of parameters to optimise and the cost of computing the fitness function on each stumbling individual makes this approach impractical. Before going further, the reader should be made aware that many NE methods exist, in order to describe the essence of what a NE is, the formal relations pre-

sented in this section are those of the *t*-SNE algorithm [10], a particularly influential method of NE.

The SNE paper introduced a trick allowing gradient descent while working with neighbourhoods: instead of using sets to model neighbourhoods, SNE models the neighbourhoods through a surrogate continuous measure of similarity. This allows SNE to smooth out the steps in the error landscape, an approach quite common in ML when encountering non-differentiability issues, sometimes used for instance in spiking neurons [81].
Formally, let $\mathbf{X} = [\mathbf{x}_i]_{1 \leq i \leq N}$ denote the HD coordinates of the data and δ_{ij} be the distance between points $\mathbf{x}_i$ and $\mathbf{x}_j$ according to a distance metric. The soft, probabilistic neighbourhoods in HD are modelled by the symmetrised pairwise similarities p_{ij}, defined as

$$p_{j|i} = \frac{\exp(-\delta_{ij}^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\delta_{ik}^2 / 2\sigma_i^2)}, \quad p_{i|i} = 0, \quad \text{and } p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N}, \tag{2.2}$$

where the individual, point-specific radii $\sigma_\cdot$ are adjusted during initialisation to comply with a user-set perplexity. The entropy of a distribution is the logarithm of its perplexity, therefore, the *t*-SNE algorithm equalises the entropy of a Gaussian ball modelling neighbours around each point in HD. Intuitively, this allows an adaptation to the local density: points in sparse regions will have a large $\sigma_\cdot$, because they need to reach further to encompass a sufficient number of neighbours, while points in dense regions will have a small $\sigma_\cdot$. Figure 2.2 illustrates this principle.

Once the similarities are defined, the *t*-SNE algorithm enters its iterative optimisation phase. There, the points are initialised to distinct values in LD, then the coordinates are updated through gradient descent minimising the Kullback-Leibler divergence between the similarity distributions in HD and LD. Naming $\mathbf{Y} = [\mathbf{y}_i]_{1 \leq i \leq N}$ the LD representation of the data, the similarities in LD are defined following a distribution with heavier tails than the Gaussian used in HD:

$$q_{ij} = \frac{(1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1}}{\sum_{k \neq l} (1 + \|\mathbf{y}_k - \mathbf{y}_l\|^2)^{-1}}, \quad q_{ii} = 0 \ . \tag{2.3}$$

With this definition of LD neighbourhoods, the gradient for the LD coor-

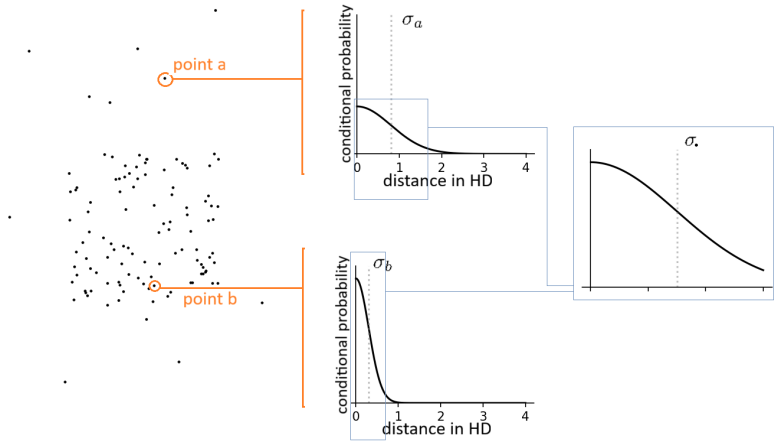


Figure 2.2: Illustration on the invariance to distances in HD of the Gaussian kernel used in (t-)SNE. On the left, points in a 2-dimensional space, with heterogeneous density. Middle plots: conditional probabilities ($p_{\cdot|\cdot}$ in the above equations) computed for a point a in a low density region, and a point b in a dense region, as a function of distance in HD. Right plot: conditional probabilities but with an x-axis normalised by the radius ($\sigma_\cdot$) of both points and the y-axis min-max normalised: the curves overlap perfectly. Note that in practice, the conditional probabilities are computed on distinct pairs of points rather than on $\mathbb{R}^+$; since the radius adapts to the local density, the maximal value taken by the discrete conditional probability distribution would typically not vary as much from point to point than in this figure.

dinates of the Kullback-Leibler divergence between the two distributions takes the form:

$$\frac{\partial L}{\partial \mathbf{y}_i} = 4 \sum_{j \neq i} (p_{ij} - q_{ij})(1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1}(\mathbf{y}_i - \mathbf{y}_j) \ . \tag{2.4}$$

The gradients naturally decompose to a repulsion and attraction terms: the i^{th} point will move towards the j^{th} point if $p_{ij} > q_{ij}$, and away from it if $p_{ij} < q_{ij}$. Large ensembles of points in LD can exert strong repulsion to other points, hampering movement; to alleviate this issue, during the early stages of optimisation, the attraction forces are often exaggerated [10], a trick called *early exaggeration*.

2.2.2 Properties resulting from the formulation

At first glance, the mismatch of kernels between HD and LD spaces may seem counter-intuitive: it is akin to forcing a square peg in a round hole

of equal surface area. The initial SNE algorithm used Gaussian kernels in both HD and LD, however, this led to many intrusions in the LD space, a symptom called "the crowding problem" [6, 10]. Using larger tails in LD induces longer ranged repulsions, leading to better distinction between structures; the vast majority of NE methods follow this principle. This choice increases the quality of the embeddings, though the precise mechanisms at play when varying the tails in LD remains difficult to intuitively grasp due to the highly non-linear dynamics involved in the optimisation. This inductive bias of NE is further studied in Section 3.6 and discussed in Section 4.

The high level objective of NE methods is to correctly capture neighbourhoods in the LD structures. However, when examining where the HD neighbours of a point i are located in a NE, it is common to observe that a large portion of these neighbours are distant from i in LD. This was apparent in Figure 1.8 and hinted in the difference of performance between a KNN classifier trained in the HD space and one trained in a NE, in Table 1.1. Therefore, while every point attracts its HD neighbours in NE, the repulsive field emanating from every point in LD in combination to the dynamics of the optimisation can lead to non-obvious arrangements in LD.
Intuitively, a HD neighbour j of point i can in effect be repelled from the approximate location of i in LD if points surrounding i (be it other neighbours of i in HD, or neighbours of neighbours of i) are not neighbours of j in HD. As j gets repelled, the norm of its attraction to i tampers with distance instead of increasing linearly, as it is scaled by $(\mathbf{y}_i - \mathbf{y}_j)(1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1}$. This opens the possibility for j to be driven by other currents, either pulled by attraction from other, more available HD neighbours, or carried adrift by the ambient repulsive field. Therefore, while not explicit in the method definition, *the dynamics of NE involve notions of higher order relationships between points*, such as mutuality of neighbourhoods and communities.
Empirical evidence suggests that these arrangements can bring particularly well generalisable embeddings, as witnessed in the works of Section 3.7, in [55, 56], or in the broad adoption of NE to visualisation tasks across domains.

Similarities in HD do not necessarily require to be built around SNE's principle of equalising entropy around each point. Figure 2.3 compares t-SNE embeddings of the COIL-20 dataset using two different schemes to determine $\sigma_\cdot$. In the top view, the similarities in HD are built by equalis-

ing the entropy of neighbourhoods, as in standard t-SNE; in the bottom view, the similarities are built by fixing σ_i to half the distance to the 4^{th} nearest neighbour of each point. In the modified version, tweaking the attraction-repulsion ratio allows the embeddings to qualitatively resemble those of the standard t-SNE version. This experiment suggests that, if some adaptation to local density is performed, the precise way it is achieved is not as critical as the balance of attraction and repulsion forces in the dynamics, a study on the attraction-repulsion balance is given in [54]. Naturally, another particularly important hyperparameter is the shape of the LD similarities, as these profoundly shape the dynamics during optimisation. Works like [11, 12] explore the influence of LD kernel shapes, experiences in Section 3.6 bring more insight on this matter, and discussions in Section 4 further reflect on this.



Figure 2.3: Qualitative comparison of t-SNE embeddings of the COIL-20 dataset when using different methods to define the radii σ_i used to adapt the HD similarities to local density. Top view: standard t-SNE, with radii defined by equalising the entropy of neighbourhoods. Bottom view: modified t-SNE, with radii fixed to half the distance to the 4^{th} nearest neighbour of each point; multiple attraction/repulsion multipliers are tested in both views.

Using density adaptive Gaussians to model neighbours in HD allows the HD similarities to contrast between nearby points, even if the distances concentrate. This property comes at the cost of losing information on density when representing the data in LD. This effect is striking in Figure 2.4. There, the S-curve dataset is sampled heterogeneously, the top half (black) being sampled 10 times more often than the bottom half (lavender). The linear projection does not appear affected by the heterogeneous sampling, whereas the NE distorts the data significantly: the lavender part is shrinked, and there appears to be a border effect detaching it from the rest of the structure.

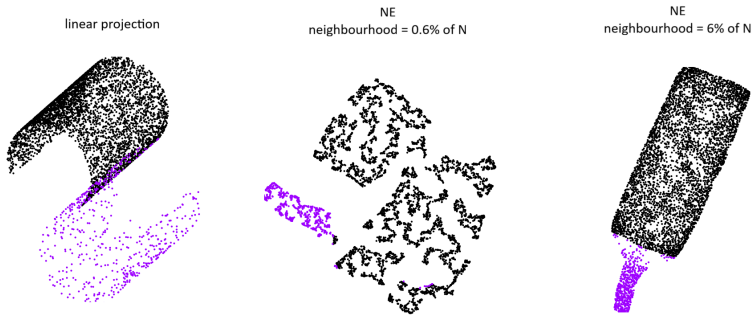


Figure 2.4: Linear projection, NE with small neighbourhood size, and NE with large neighbourhood size of the S-curve dataset with a heterogeneous sampling density: the top half (black) is sampled 10 times more often than the bottom half (lavender).

The observations of Figure 2.4 can be worrying: it can be difficult to assess the quality of the sampling process, therefore survival bias can easily creep in unnoticed. This raises the question: how many separated clusters in NE were mistakenly interpreted as distinct structures when they were in fact the result of heterogeneous sampling? The same question can be raised on the comparison of cluster sizes in NE. Figure 2.5 is not particularly reassuring, as it shows that common datasets can indeed express heterogeneous densities throughout the structures. In the mouse brain dataset, for instance, one of the larger clusters is particularly dense in HD: it is possible

that this cluster is erroneously detached from another, or appears inflated. Herein is made manifest the virtue of circumspection in the reading of NE: these methods must always be used in conjunction with other tools.



Figure 2.5: Neighbour embeddings of the mouse brain, abalone, and satellite datasets, showing the heterogeneous spread of radii σ_i used to define the HD similarities: these reflect the local density in the HD data.

The lack of preservation of density in NE can be difficult to evade as it directly involves the notion of short ranged distances in the HD space, a quantity that NE methods attempt to work around. A method introduced in [16] nevertheless captures information on density in the LD embeddings by regularising NE cost functions. When visually comparing the embeddings obtained by the density the preserving method with others obtained by classical NE, some fine grained details appear to be lost when preserving density. Blurry regions in embeddings produced by this method resemble what one would expect from a distance preserving method, hinting that there might be a trade-off there, reverting some of the benefits of NE.

2.2.3 Accelerating neighbour embeddings

The cost function of NE involves N^2 terms, making the optimisation costly when N grows large. In practice, most methods of NE use fast approximations, the fastest ones scaling as $\mathcal{O}(N)$. There a many ways to achieve faster NE, these paragraphs draw in broad strokes the main approaches.

The first element of a NE needing acceleration is the computation of the HD similarities, the computational bottleneck there being the search for nearest neighbours in HD. For this, some methods use tree based approaches: Barnes-Hut t-SNE uses VP-trees [32], FIt-SNE uses multiple randomised KD-trees [33], UMAP [40] uses nearest-neighbour descent [82], and other methods like LargeVis [83] combine random projections with nearest-neighbour descent.

Some methods preserve neighbourhoods across multiple or even all scales [61]; empirical evaluations show that these attain good quality embeddings [41], but the multi-scale nature of the neighbourhoods complicates such accelerations, as neighbourhoods of large scale require dense, N-by-N similarity matrices. The work [60] proposes an acceleration to multi-scale NE based around the principle that, if one can afford some level of approximation, then fixing a constant perplexity but using it on subsamples of the data captures the same large scale structure as using a large perplexity on the full dataset.

The iterative optimisation of the NE cost function is trickier to accelerate because the LD similarities do not quickly decay to 0 as the Gaussians used in HD do. This calls for an approximation of interactions across all ranges rather than between neighbours only. To tackle this problem, accelerated NE method can be split into two categories: methods approximating the repulsive forces exerted on each points through sampling, and methods approximating these by modelling the occupancy of regions in LD.
In the former, like UMAP, each point will sample randomly and without bias a handful of other points; these are then used to compute a repulsive force. These methods tend to be very fast, do not constrain the dimensionality of the LD space, but, since the random points being selected are unlikely to be close from the point being considered, the repulsions computed this way only capture effects of the global landscape in LD, impacting the quality of the fine-grained structures [41–43]. In addition to struggling to capture local repulsions, relying on unbiased negative sampling shifts the balance of the attraction/repulsion forces exerted between points [54, 84], further impacting the resulting embedding. This shift in the attraction/repulsion balance can however be controlled when crafting the cost function around the idea of positive and negative sampling from the start, this is sometimes called the noise contrastive approach [84, 85].
In the latter, methods like Barnes-Hut t-SNE and FIt-SNE directly model the occupancy of regions in LD, through tree structures in Barnes-Hut t-SNE and through a grid structure for FIt-SNE. The methods tend to be slower in practice, despite FIt-SNE sharing the same $\mathcal{O}(N)$ complexity as UMAP, and constrain the LD space to very low dimensionalities, which is fine for visualisation purposes. However, they model the repulsive field more faithfully, leading to better quality embeddings, in particular when evaluating intricate structures [41–43].

In essence, to close this introduction to NE, the myriad methods can differ in their way to compute the HD similarities, in how they define their error function, or in the way they compute repulsions in LD, but they all tend to share the same core principles. Similarities in HD adapt to local density and quickly decay to 0, the dynamics are defined through attraction determined in HD and repulsion forces determined in LD, and these repulsive forces range farther out than the attractive forces, introducing an important inductive bias.

Chapter 3

Contributions

This section details the contributions of the thesis. Published or submitted works are reproduced here by including the corresponding papers, each preceded by a brief introductory paragraph to situate the work within its broader context. As research progress is unpredictable, the last contribution presented in Section 3.8 was completed only shortly before and even during the writing of this manuscript. Consequently, this works remain in preparation for submission, and a summary of the current progress and preliminary results is provided in place of a full paper.

3.1 Impact of data subsampling on fast multi-scale neighbour embeddings

A couple of years before the start of this thesis, a fast multi-scale NE method was proposed in [60]. This method accelerates an exact multi-scale t-SNE method [61], preserving both local and global structures. In the exact method, the multi-scale nature of the algorithm stems from modelling the structures in HD using multiple perplexities. Using large perplexities to capture global structures renders the HD similarity matrix dense, which severely impacts performance. The accelerated version of the algorithm circumvents this issue by using a constant low perplexity but applied on increasingly sparse subsamplings of the data. The accelerated algorithm has a time complexity of $\mathcal{O}(N \log^2 N)$, restricting its use to non-interactive contexts if the datasets are moderately large.
This work studies the impact of re-sampling multiple times the data in an attempt increase the robustness of the method to randomness, at the cost of a linear increase in computation time. Since this work is a study of an existing method and is of low impact, the full paper is delegated to Annex 5.4 in order to prioritise more significant contributions in the main body of the thesis.

3.2 SQuaD-MDS: a fast approximation of multi-dimensional scaling

Since modern NE algorithms have computational complexities as low as $\mathcal{O}(N)$ [33, 40], it is imperative to also have fast supplementary methods allowing the visualisation of global structures in large datasets. In their absence, users might be inclined to rely solely on NE visualisations, opening the door to misinterpretations.
While PCA is fast, it is severely constrained by its linear nature, making it more suitable for pre-processing by projecting data to intermediate dimensions than for visualisation, where it is important to capture as much information as possible in only 2 or 3 dimensions. In practice, MDS tends to favour the preservation of global structures and with more details than PCA, making it a better choice for visualising global structures.

Since MDS aims to preserve pairwise distances between points, and that these distances do not conveniently decay to zero like HD similarities in NE methods do, standard MDS algorithms like the stress reducing SMACOF procedure [63] tend to have at least a time complexity of $\mathcal{O}(N^2)$. As mentioned in the following paper, some accelerations have been proposed. Some of these are quite sophisticated, involving significant approximations or being demanding in hyperparameters; other simpler approaches take the sampling route to minimise a surrogate of the full MDS cost function [86,87]. In the latter methods, pairs of points are sampled and their distance in LD driven towards the value determined in HD. These approaches report good results in linear time, highlighting the redundant nature of the full distance matrix when the number of observations is larger than the number of variables in the HD space.

In this work, a very fast MDS algorithm is introduced. In addition to being fast, this approximation of MDS is simple and does not involve additional significant hyperparameters beyond those of standard MDS. The method takes a similar approach to those sampling pairs of points to minimise an approximation of the MDS cost function, but it is built around the sampling of groups of four points. This allows for the preservation of *relative distances* rather than absolute distances, by dividing each distance by the sum of the six distances involved when working with four points. This enables the embedding to adapt to different scales in LD, a property allowing hybridisation with other gradient based methods that might favour some scales, such as NE [41] or deep learning models [57]. In addition to removing the constraint of scale, the paper argues that working with relative distances can also induce more efficient movement of points, as illustrated in Figure 1 of the paper.
Since publication, the work [18], as well as [57] in which I am a co-author, have used SQuaD-MDS as a fast MDS alternative and confirmed its effectiveness.

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

Stochastic quartet approach for fast multidimensional scaling

Pierre Lambert[1], Cyril de Bodt[1], Michel Verleysen[1], John A. Lee[1,2]

1- UCLouvain.be - ICTEAM/ELEN
Place du Levant 3 L5.03.02, 1348 Louvain-la-Neuve - Belgium
2- UCLouvain.be - IREC/MIRO
Avenue Hippocrate 55 B1.54.07, 1200 Brussels - Belgium

Abstract. Multidimensional scaling is a statistical process that aims to embed high-dimensional data into a lower-dimensional, more manageable space. Common MDS algorithms tend to have some limitations when facing large data sets due to their high time and spatial complexities. This paper attempts to tackle the problem by using a stochastic approach to MDS which uses gradient descent to optimise a loss function defined on randomly designated quartets of points. This method mitigates the quadratic memory usage by computing distances on the fly, and has iterations in $\mathcal{O}(N)$ time complexity, with N samples. Experiments show that the proposed method provides competitive results in reasonable time. Public codes are available at `https://github.com/PierreLambert3/SQuaD-MDS.git`.

1 Multidimensional scaling and its limitations

Dimensionality reduction (DR) is the process of mapping high-dimensional (HD) observations into a lower-dimensional (LD) space such that the LD embedding is a faithful representation of the HD data. The main DR uses are in machine learning, to curb the curse of dimensionality, and in visualisation. Mapped data can reveal structures that would lay hidden from the human perception if left in HD. Typically, some information is lost by the DR and, therefore, each DR method has a take on what kind of information should be preserved and what can be lost. Used frequently in visualisation, t-SNE [1] aims at retaining the neighbourhood of each point according to a distance metric and a perplexity, which reflects the size of the neighbourhood to preserve. While t-SNE excels at retaining local structures, sufficiently remote points tend to be considered equally distant by the algorithm and, therefore, the larger-scale structures can be distorted. Such distortions can lead to erroneous conclusions by the human user, who might overestimate the dissimilarity between two clusters that are distant in the LD embedding. For this reason, using multiple DR paradigms in conjunction is a good practice in visualisation: another embedding that preserves distances instead of neighbourhoods would have prevented this erroneous conclusion.

This paper considers metric multidimensional scaling (MDS): a DR technique that produces a LD embedding such that the pairwise distances in LD reflect those in HD. MDS minimises a cost function which, in its simplest form, is the sum of the squared differences between distances in HD and the Euclidean distances in LD. A common strategy to optimize this cost function is based on

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

the SMACOF algorithm [2], involving iterations of $\mathcal{O}(N^2)$ time complexity with N samples. This quickly becomes problematic when N grows. SMACOF also requires the full matrix of pairwise HD distances, consuming $\mathcal{O}(N^2)$ memory.

Some alternatives with lower time or memory complexities exist, such as the iterative spring-based model [3], where each point is subject to forces computed according to two constant-sized sets of points. This enables iterations with $\mathcal{O}(N)$ time complexity. However, the sets are subject to iterative refinements, making the total number of required iterations increase with the size of the data set. A different divide-and-conquer approach [4] applies MDS on sub-matrices of the HD distance matrix and then stitches together the results. Another take on the problem is proposed in landmark MDS [5], which first embeds a small subset of the data in LD and then arranges the rest of the points with respect to these landmarks. Alternatively, the effects of high computational complexities can be reduced by focusing on certain parts of the LD embedding thanks to the input of a user [6].

While these accelerated MDS methods provide interesting results, they often involve multi-step approaches that complicate the algorithm and tend to make it monolithic in the sense that the optimisation process becomes an intractable block once the optimisation started. This paper proposes an algorithm that has $\mathcal{O}(N)$ time complexity and does not require a full distance matrix, making it more memory-efficient. In addition to showing that the algorithm can yield good distance correlation between the HD and LD spaces, this paper shows that by using a simple gradient descent optimisation, the algorithm is open for hybrid approaches, where gradients of different types are mixed.

Section 2 presents the algorithm, while Section 3 is dedicated to an empirical assessment of its quality and speed performance. Section 4 sketches some conclusions and perspectives.

2 Proposed method

The proposed algorithm performs MDS by using gradient descent on a cost function defined on quartets of points. Before defining the gradients, this section starts by explaining how to articulate a distance scaling problem around the use of quartets.

2.1 Quartet-based optimisation

Let us consider a quartet formed by four points that has a HD distance matrix δ and a LD Euclidean distance matrix d. If these points have a LD embedding that perfectly preserves their HD distances, then the relative distance between the i^{th} and the j^{th} points in LD, d^{r}_{ij}, also perfectly matches their relative distance in HD, δ^{r}_{ij}. These relative distances are defined as

$$d^{\text{r}}_{ij} = d_{ij} \Big/ \Big(\sum_{a=1}^{3} \sum_{b=a+1}^{4} d_{ab} \Big) \quad \text{and} \quad \delta^{\text{r}}_{ij} = \delta_{ij} \Big/ \Big(\sum_{a=1}^{3} \sum_{b=a+1}^{4} \delta_{ab} \Big) \ .$$

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

A cost function for the preservation of relative distances within this quartet of points can be defined:

$$C_{\text{quartet}} = \sum_{a=1}^{3} \sum_{b=a+1}^{4} (\delta_{ab}^{\text{r}} - d_{ab}^{\text{r}})^2 \quad . \tag{1}$$

To apply this quartet approach to the whole data set, the algorithm iteratively and randomly splits the observations in groups of 4, and the gradient for each point is computed according to (1). This use of quartets should not be confused with triplet methods such as stochastic triplet embedding [7], as they work with data of different natures. Triplet embeddings use partial similarity rankings of the form "A is more similar to B than to C" often resulting from human judgement, whereas the proposed fast MDS method uses HD observations.

Using quartets of points gives constant-time gradients and has a natural analogy to triangulation (1 point versus 3 others) when applied to a target dimension of 2. Quartets are also the reason why the full HD distance matrix isn't needed : only 6 HD distances are used at any instant. It is noteworthy that using relative distances precludes any preservation of scale: the LD distances tend to correlate to those in HD, but the scale in LD will stay close to the scale at which the embedding is initialised.

2.2 The gradients

Considering a quartet of points with a LD Euclidean distance matrix d and a HD distance matrix δ, I is the 4x4 identity matrix and x_q is the value taken for some LD dimension by the q^{th} point in the quartet. For the sake of concision, only the gradients for one term in the sum of (1) is shown here, as the other terms in the sum share similar forms:

$$\frac{\partial(\delta_{ij}^{\text{r}} - d_{ij}^{\text{r}})^2}{\partial x_q} = \frac{2}{S}(d_{ij}^{\text{r}} - \delta_{ij}^{\text{r}})\left(\frac{I_{qi}(x_q - x_j) + I_{qj}(x_q - x_i)}{d_{ij}} - d_{ij}^{\text{r}} \sum_{b \in \{1,..,4\} \setminus q} \frac{x_q - x_b}{d_{qb}}\right)$$

with $S = \sum_{i<j} d_{ij}$. The motivation for using relative distances is that each of the six LD distances in the quartet contributes to the update for each LD point, as opposed to direct distance preservation where only three distances would contribute. This results in the behaviour illustrated in Fig. 1, where each point moves to minimise the error on the quartet as a whole.

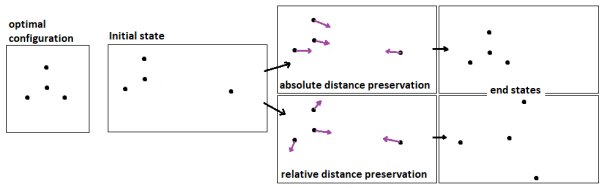


Fig. 1: Difference in behaviour when using relative distances and raw distances.

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

3 Experiments and discussion

To assess the performance of the proposed algorithm in terms of speed and distance preservation, it is compared to the SMACOF-based MDS [2] on various data sets. Section 3.1 details our implementation, while Section 3.2 describes the results.

3.1 Implementation

Our implementation uses a gradient descent optimizer in its simplest form: the gradients are scaled by a learning rate and then directly subtracted to the positions of the LD points. During the first part of the optimisation, the HD distances are squared before being transformed to relative distances. This helps the optimisation by exaggerating the differences between the distances within a quartet. Because of the stochastic nature of the optimisation, the updates on the position of the points tend to be erratic; this isn't a problem during the first part of the optimisation, but it renders the fine-tuning of the final positions difficult. To counter this behaviour, it is important that the learning rate decays to small values. We determined experimentally that, for embeddings initialised with a standard deviation around 10, a learning rate starting in the hundreds that decays to 10^{-3} is adequate. Having it decay to greater values tends to reduce the quality of the results. All the following embeddings using the proposed fast MDS are given the same hyper-parameters. They result from a single run of 1000 iterations, with a target dimension of 2 and a PCA initialisation scaled to have a standard deviation of 10. Non-PCA initialisations can produce similar results, but they typically require a larger number of iterations before convergence. The code can be found at: `https://github.com/PierreLambert3/SQuaD-MDS.git`.

3.2 Empirical assessment

Seven data sets are featured in this paper, with size N and dimensionality M. Abalone: $(N,\ M) = (4177,\ 8)$; coil20: $(N,\ M) = (1440,\ 1024)$; a subset of MNIST digits: $(N,\ M) = (3400,\ 784)$; a subset of MNIST fashion: $(N,\ M) = (4000,\ 784)$; satellite: $(N,\ M) = (4434,\ 36)$; a subset of Gisette: $(N,\ M) = (3700,\ 4900)$; RNAseq: $(N,\ M) = (10^4,\ 50)$. Most sets are available from the UCI machine learning repository. RNAseq contains single-cell data used in [8]; the same 50 principal components from the selected features are used, as in [8].

Table 1 shows the Pearson correlation between the pairwise distances in HD and LD for various models on the data sets. The 1$^{\text{st}}$ row shows results obtained with PCA. The 2$^{\text{nd}}$ row shows the best result of 10 different initialisations for the SMACOF MDS algorithm as implemented by scikit-learn. The 3$^{\text{rd}}$ row is also the result of a MDS using SMACOF but this time initialised with PCA. The 4$^{\text{th}}$ row is our fast MDS algorithm in its default version, and the last row is an experiment where the HD distances for the quartets are first transformed non-linearly before being used by the cost function. This transformation is of the form $\delta'_{ij} = 1 - \exp(-(\delta_{ij} - \delta_{\min})/(2\sigma))$ with σ being the standard deviation of the HD

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

distances in the quartet and $\delta_{\min}$ the smallest HD distance in the quartet. This transformed distance is then divided by the sum of the transformed distances to become relative. The purpose of this transformation is to increase the contrast between the HD distances within the quartet in the hope of reducing the effect of the concentration of norms [9] when the input dimension is high. In terms of correlation of distances, Table 1 shows that the proposed method can provide competitive results, and that its results tend to be particularly good when the data dimension is high.

	abalone	satellite	RNAseq	digits	fashion	coil20	gisette
PCA	0.82	0.98	0.93	0.50	0.69	0.76	0.45
10x MDS	0.82	0.93	0.93	0.63	0.82	0.80	0.84
PCA-MDS	0.99	0.99	0.90	0.66	0.89	0.84	0.67
fastMDS	0.98	0.97	0.95	0.73	0.94	0.86	0.85
fastMDS-rbf	0.98	0.96	0.96	0.75	0.92	0.87	0.89

Table 1: Correlation of the distances in LD and HD; the data sets are of increasing dimensionality from left to right.

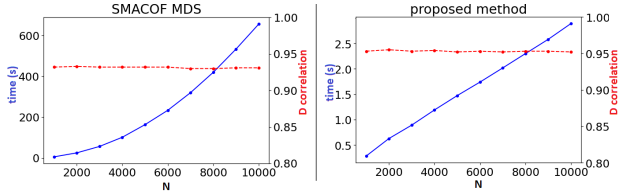


Fig. 2: Evolution of computation time and distance correlation for samples of RNAseq of size varying between 10^3 and 10^4 with steps of 10^3.

Figure 2 shows that the fast MDS algorithm runs in linear time with N, as opposed to the quadratic scaling of SMACOF MDS; the distance correlation is stable with different sizes N, which can be surprising considering that the proportion of HD distances that are used during the optimisation decreases when N increases. Actually, the number of HD distances used by the proposed algorithm is *number_of_iterations*6_distances_per_quartet*number_of_quartets*, in this case $10^3 * 6 * \frac{N}{4}$, but the total number of HD distances is $\frac{N*(N-1)}{2}$. This supports the intuition that when N is high, using only a subset of the distances can be sufficient for a good embedding, which is a principle of landmark-based approaches.

Figure 3 shows some 2-D embeddings of a subset of RNAseq; 'D correlation' is the Pearson correlation between HD and LD distances and 'R_{NX} AUC' is the AUC of the $R_{\text{NX}}(K)$ curves [10], which reaches 1 when the multi-scale neighbourhoods of the data are well preserved. Since the proposed method relies on a simple gradient descent, it is possible to combine it with another gradient-based algorithm. In the 3$^{\text{rd}}$ scatter plot, we added the quartet based distance gradients to those of t-SNE to obtain a hybrid embedding. The figure shows that

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

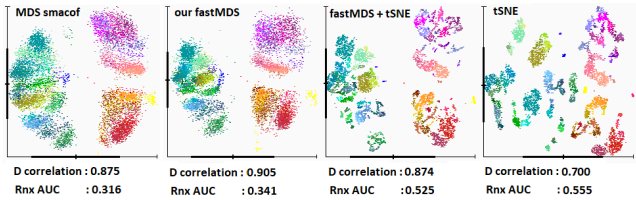


Fig. 3: 2-D embeddings of a subset of RNAseq of size 9300.

this hybrid approach can preserve neighbourhoods quite well while keeping the distances meaningful.

4 Conclusion and perspectives

Experiments show that the proposed fast MDS method produces competitive results while keeping the algorithm simple and the computational complexities low. We may have only scratched the surface of the possibilities that a stochastic quartet-based approach can offer; we can imagine an extension that would enable the use of weighted distances, the use of different quartet-based metrics, or an adaptation for ordinal MDS. Another perspective is to further study the use of quartet-based distance gradients in a hybrid approach to DR, by mixing the distance preservation and the neighbourhood preservation paradigms.

References

[1] L. van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[2] J. de Leeuw and P. Mair. Multidimensional scaling using majorization: Smacof in r. *Journal of Statistical Software, Articles*, 31(3):1–30, 2009.

[3] M. Chalmers. A linear iteration time layout algorithm for visualising high-dimensional data. *Proceedings of the 7th conference on Visualization*, pages 127 – 131, 12 1996.

[4] Tzeng, Henry Lu, and Wen-Hsiung Li. Multidimensional scaling for large genomic data sets. *BMC bioinformatics*, 9:179, 02 2008.

[5] V. Silva and J. Tenenbaum. Sparse multidimensional scaling using landmark points. *Technology*, 01 2004.

[6] M. Williams and T. Munzner. Steerable, progressive multidimensional scaling. In *IEEE Symposium on Information Visualization*, pages 57–64, 2004.

[7] L.J.P. van der Maaten and K.Q. Weinberger. Stochastic triplet embedding. *Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing*, 2012.

[8] D. Kobak and P. Berens. The art of using t-sne for single-cell transcriptomics. *Nat Commun 10, 5416*, 2019.

[9] D. Francois, V. Wertz, and M. Verleysen. The concentration of fractional distances. *IEEE Trans. Knowl. Data Eng.*, 19(7):873–886, 2007.

[10] J. A. Lee, D. H. Peluffo-Ordóñez, and M. Verleysen. Multi-scale similarities in stochastic neighbour embedding: Reducing dimensionality while preserving both local and global structure. *Neurocomputing*, 169:246–261, 2015.

3.3 A hybrid MDS-NE method

This work introduces a fast NE that preserves both local and global structures by combining the SQuaD-MDS method introduced in Section 3.2 with a fast NE method, both sharing the $\mathcal{O}(N)$ time and memory complexity [33]. Stochastic gradients from SQuaD-MDS are combined with the deterministic estimations of t-SNE gradients of [33] in an hybrid optimisation. In practice, the cost of computing the MDS gradients is negligible compared to that of computing the NE gradients, but empirical results show that the impact on the quality of the global organisation of the embeddings is significant, without sacrificing local structure preservation.

If compared to multi-scale NE methods [60, 61] studied in Section 3.1, in its philosophy, the hybrid approach takes a shortcut by only considering one local scale through the NE gradients, and a global scale through the MDS gradients; whereas the multi-scale NE methods consider a whole range of exponentially increasing scales. Interestingly, in practice, unpublished comparisons show that the multi-scale neighbourhood preservation curves of the hybrid method does not differ much from those of the fast multi-scale NE method of [60], hinting that "medium-range" structures often tend to correctly organise themselves when both local and global structures are well preserved. This observation suggests that, if one was to depart from the somewhat inelegant distance-neighbourhood hybrid approach of this work, a purely neighbourhood based method of the type of the multi-scale NE methods could be devised that only considers two scales: a very local scale and a very global scale, allowing these to also attain $\mathcal{O}(N)$ time and memory complexities if using the subsampling trick of [60] combined with modern accelerations like the ones in [33] or introduced in Section 3.6.

In retrospective, one consideration is that, while the hybrid method was tested quite extensively on many datasets without revealing issues, there is little guarantee that the MDS gradients would stay effective in cases where the distances in HD concentrate heavily. This would happen in datasets expressing high global intrinsic dimensionality. After all, the very reason why NE methods were invented is that distance-based methods like MDS tend to fail in high dimensions. If such cases would arise, the original SQuaD-MDS paper [64] shows that non-linear transformations of the dis-

tances can be performed better discriminate between points in a quartet of points, using this approach in the hybrid method should be straightforward.

Neurocomputing 503 (2022) 17–27

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

SQuadMDS: A lean Stochastic Quartet MDS improving global structure preservation in neighbor embedding like *t*-SNE and UMAP

Pierre Lambert [a,*], Cyril de Bodt [a,b], Michel Verleysen [a], John A. Lee [a,c]

[a] UCLouvain – ICTEAM/ELEN. Place du Levant 3, bte L5.03.02, 1348 Louvain-la-Neuve, Belgium
[b] MIT Media Lab – Human Dynamics Group, 20 Ames Street, Wiesner Building (E15), Cambridge, MA 02139, USA
[c] UCLouvain – IREC/MIRO, Avenue Hippocrate 55, B1.54.07, 1200 Brussels, Belgium

ARTICLE INFO

Article history:
Received 8 February 2022
Revised 29 April 2022
Accepted 27 June 2022
Available online 30 June 2022
Communicated by Zidong Wang

Keywords:
Dimensionality reduction
t-SNE
UMAP
Neighbor embedding
MDS
Global structure
Stochastic algorithms
Data visualization
Distance preservation

ABSTRACT

Multidimensional scaling is a process that aims to embed high dimensional data into a lower-dimensional space; this process is often used for the purpose of data visualisation. Common multidimensional scaling algorithms tend to have high computational complexities, making them inapplicable on large data sets. This work introduces a stochastic, force directed approach to multidimensional scaling with a time and space complexity of $\mathcal{O}(N)$, with N data points. The method can be combined with force directed layouts of the family of neighbour embedding such as *t*-SNE, to produce embeddings that preserve both the global and the local structures of the data.

Experiments assess the quality of the embeddings produced by the standalone version and its hybrid extension both quantitatively and qualitatively, showing competitive results outperforming state-of-the-art approaches. Codes are available at https://github.com/PierreLambert3/SQuaD-MDS-and-FItSNE-hybrid.

© 2022 Elsevier B.V. All rights reserved.

Multidimensional scaling (MDS) has been for long a popular method of embedding high-dimensional data nonlinearly in lower-dimensional spaces, in order to better visualize the proximities in data, for instance. This paper revisits the paradigm of MDS and distance preservation in general, by proposing a fast algorithm to compute gradients for a stochastic, force-directed optimisation of metric multidimensional scaling (mMDS) [1]. An empirical evaluation of the proposed method shows that it yields results that are competitive with those obtained by the commonly used SMACOF algorithm[2], while its quadratic time and memory complexities with respect to the number of observations are reduced down to linear complexities. This work also shows that the stochastic mMDS gradients can be combined easily with gradients from neighbour embedding algorithms in order to produce solutions that both preserve the global and the local structures of the data. Public, free implementations are accessible at https://github.com/PierreLambert3/SQuaD-MDS-and-FItSNE-hybrid.

Section 1 introduces the current state of dimensionality reduction (DR) for data visualisation, and motivates this work in the broader context. Section 2 explains the mMDS problem, showing that the problem has an inherently high algorithmic complexity. The section also provides a brief overview of the existing accelerated methods for mMDS and shows how the proposed method differs from these solutions. Section 3 describes the proposed method and its hybrid extension, which combines neighbour embedding to mMDS. Section 4 reports an empirical evaluation of both the standalone method and its hybrid extension.

1. Dimensionality reduction for visualisation

Dimensionality reduction is the process of embedding HD data into a lower-dimensional (LD) space such that some properties are preserved [3]. Having a visual representation of HD data can help understand the dynamics of certain industrial processes [4], assist in monitoring [5], in research [6,7], or in the study of the relationships between certain products and consumer habits for marketing

Abbreviations: DR, Dimensionality reduction; mMDS, metric multidimensional scaling; LD/HD space, lower/higher-dimensional space; SGD, stochastic gradient descent.

* Corresponding author.
E-mail address: pierre.h.lambert@uclouvain.be (P. Lambert).

https://doi.org/10.1016/j.neucom.2022.06.108
0925-2312/© 2022 Elsevier B.V. All rights reserved.

purposes [8]. When used for visualisation, the LD space is either 2- or 3-dimensional; for the sake of clarity, this paper will assume a LD space composed of 2 variables, although all developments can be carried out with larger numbers of LD features.

Loss of information is often inevitable when reducing the dimensionality of the data. For this reason, many DR methods exist: each one focuses on the preservation of a certain property of the data.

Introduced in 1901, PCA [9] was the first DR method. PCA is a linear projection of the data such that most of the variance is captured along the new axes, which are also constrained to be orthogonal. As variance reflects the global spread of data, projections produced by PCA tend to poorly reflect the small scale structures of data. Across the 20^{th} century, the quality of the preservation of medium to large scale structures in the data improved with the introduction of nonlinear DR methods such as stress-based MDS [1]. Nonlinear DR has also diversified into other paradigms, like kernel methods [10], the use of alternative distance metrics [11], or with parametric neural models such as auto-encoders [12].

The preservation of the local, fine-grained structure of data remained problematic until the introduction of neighbour embedding in 2002 by G. Hinton in the paper *Stochastic Neighbor Embedding* [13]. Neighbour embedding focuses on the preservation of the small-scale structures in the data, by preserving fast-decaying similarities defined in both the HD and LD spaces. The similarities are a nonlinear function of the Euclidean distances such that close neighbours have similarities that are greater than zero, and points that are not neighbours have near-zero values.

In practice, data sets from the real world can get very large and visualising such big data sets requires methods that scale well when the number of points grows. The fast decay of the pairwise similarities involved in neighbour embeddings makes them particularly well suited for large data sets, since small similarities might be deemed negligible and thus be decimated to get sparsity. This lead to a variety of accelerated neighbour embedding methods using fast neighbour retrieval algorithms, such as Barnes-Hut *t*-SNE [14], FI*t*-SNE [15] and UMAP [16].

While neighbour embedding excels at revealing the fine details of the data, it tends to overlook the global structures because all points laying beyond a certain distance threshold are considered equally dissimilar. This severely hinders the capability of such similarity-based DR methods to capture the global properties of the data. It has been empirically shown that the global organisation of the embeddings produced by *t*-SNE and UMAP are reminiscent of the embedding initialisation, rather than a meaningful result of the optimisation process [17]: two successive runs with different random initialisation can yield two global arrangements of data in the embedding that are very likely to differ.

When the global properties of the data are not retained in the embedding, the interpretation of the LD visualisation relies more heavily on the expert knowledge of the user. This brings a risk of confirmation bias. For this reason, when using DR to explore data, it can be desirable to use embeddings produced by algorithms that preserve large scale structures in conjunction to embeddings displaying information on the finer structures [18].

Fig. 1 illustrates why relying on embeddings that preserve small-scale structures alone is problematic. The data set used in the figure comes from [19]; each observation corresponds to a cell in a mouse brain, and the data was preprocessed as in [18]. The cells were coloured by the authors such that brown colours correspond to cells which aren't neurons, warm colours to inhibitory neurons, and cold colours to excitatory neurons.

The leftmost two embeddings were produced with PCA [9] and mMDS [1]; these tend to preserve the global structures in the data. The rightmost two embeddings were produced with *t*-SNE and UMAP, focusing on small-scale structures. The global structures present in the PCA and mMDS embeddings are lost in the rightmost two embeddings: if the user has no prior knowledge about the data, its hierarchical organisation would have remained hidden. This figure also shows that mMDS embeddings are more detailed and thus preferable to PCA embeddings for visualisation purposes.

Global methods cannot benefit directly from recycling accelerations used in neighbour embedding. Global dissimilarities have no obvious sparse representation like local similarities do.

This work proposes a fast stochastic approximation to mMDS, which is a nonlinear DR method that tends to preserve the medium- to large-scale structures of data. The proposed method uses gradient descent with iterations having $\mathcal{O}(N)$ time complexity, which is equal to the time complexity of the fastest accelerated neighbour embedding methods. The fast mMDS gradients are combined with FI*t*-SNE gradients to produce a fast hybrid method. Empirical evaluations show that the hybrid approach embeds data in a way that captures the global structure of data better than FI*t*-SNE with identical hyperparameters, without losing in quality on the small scale structures.

2. Metric multidimensional scaling and its challenges

Metric multidimensional scaling is a statistical process that takes as input a distance matrix or HD data points and aims to embed these points into a LD space such that the Euclidean distances in the embedding are as close as possible to the distances in the HD space [1]. Notice that the distances in HD are not necessarily Euclidean. As illustrated in Fig. 1, mMDS tends to produce embeddings that preserve the global structures of the data more than the local structures.

The cost function for mMDS is often called "stress". In this work, the stress function is the mean squared error between the pairwise distances of the N observations in HD and their LD counterparts. If δ denotes the distance matrix in HD, d denotes the Euclidean dis-

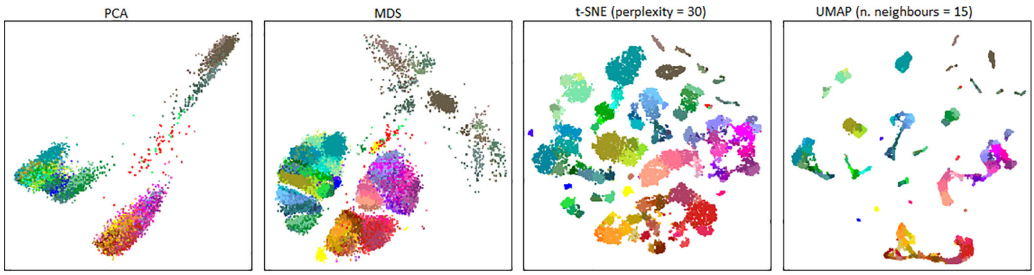


Fig. 1. Embeddings of the mouse cortex data set.

tance matrix in LD, and $\Delta_{ij} = \delta_{ij} - d_{ij}$, then the objective of mMDS is to minimise

$$\mathscr{L}^{mMDS} = \frac{1}{\binom{N}{2}} \sum_{i<j} \Delta_{ij}^2.$$

One of the most common algorithms to perform mMDS is the SMACOF algorithm (Scaling by MAjorizing a COmplicated Function) [2]: it involves an iterative optimisation with iterations of $\mathcal{O}(N^2)$ time complexity. This relatively high computational requirement stems from the fact that the SMACOF algorithm minimises the exact pairwise formulation of the stress function, which has $\binom{N}{2}$ terms in the summation. The volume of data that is being accumulated nowadays far outpaces Moore's law, making such a high computational complexity problematic.

The high computational complexity of mMDS often forces the visualisation community to perform mMDS on a random subsample of the dataset or on the prototypes resulting from a clustering or vector quantization of the data in HD space [18]. This can be dangerous as it brings a risk of over- or under-representing some parts of the data, which would lead to embeddings that poorly reflect the true data distribution.

A large number of data sets have a number of observations N that is much larger than the number of dimensions M, which implies that the original data matrix has fewer entries than the HD distance matrix ($N \times M$ entries compared to $\frac{N(N-1)}{2}$ effective entries). This can lead to the intuition that the higher the ratio $\frac{N}{M}$, the more the HD distance matrix contains redundant information. As a consequence, the full distance matrix is not necessary to find a good mMDS solution. Faster mMDS algorithms have been designed building directly or indirectly on this intuition; the next paragraph presents a brief overview of some of these algorithms.

Morrison et al. [20] propose an iterative multi-staged force-directed layout with a temporal complexity of $\mathcal{O}(N\sqrt{N})$; this algorithm places the bulk of the data points around an initial embedding of a small sample of the data set. Silva et al. follow a comparable approach in landmark-MDS [21]. Yang et al. propose a divide-and-conquer approach to mMDS [22] by dividing the HD distance matrix along the diagonal into smaller matrices. The small distance matrices are then independently used to embed the corresponding points; afterwards, the multiple embeddings are stitched together for a total time complexity of $\mathcal{O}(N \log N)$. Another interesting approach is by Williams and Munzner [23], where the high computational complexity of mMDS is mitigated by asking the input of a human supervisor to focus the computations on specific parts of the embedding.

While these algorithms significantly reduce the computational resources necessary to compute a usable mMDS embedding, they come with their own set of limitations. The multiple steps required by these algorithms often entail some additional hyperparameters, which need to be adjusted by the user, depending on the data set and the purpose of the visualisation, a potentially long process. Some multi-staged algorithms such as the divide-and-conquer approach also become intractable black-boxes to the user, who needs to wait for the completion of the algorithms to get a visual feedback. This can be detrimental to the process of data visualisation which often involves interactivity between the user and the data. Finally, even algorithmic complexities of $\mathcal{O}(N \log N)$ or $\mathcal{O}(N\sqrt{N})$ can become demanding when the data sets grow very large.

This work attempts to address these shortcomings by proposing an algorithm that perform mMDS in $\mathcal{O}(N)$ time and space complexity, using a simple stochastic gradient descent (SGD) optimisation. Assuming a target dimensionality of 2, the base method involves no other hyperparameters than the maximum number of iterations for the SGD optimisation. The simplicity of the proposed method also enables it to be mixed easily with neighbour-embedding methods, as a hybridization between the distance-scaling paradigm and the similarity-preservation paradigm. The hybrid approach is intended to mitigate the difficulties of the similarity-based methods such as t-SNE and UMAP to preserve large-scale structures, by allowing the distance-scaling gradients to guide the optimisation towards a representative global organisation.

3. Proposed method

The proposed method performs mMDS by using SGD to optimise a stress function. This section details the standalone mMDS method as well as the hybrid extension, which uses gradients from both the distance-scaling and the similarity-preservation paradigms.

SubSection 3.1 introduces the standalone version of the method, describing the overall SGD strategy and how the gradients are computed at each iteration. SubSection 3.2 explains the motivations for using relative distances instead of absolute distances. SubSection 3.3 proposes a scheme to blend the mMDS gradients with those of t-SNE, a similarity-based DR algorithm which is also optimised through gradient descent.

3.1. Computing the gradients

To keep the computational complexity of each SGD iteration low, this works takes a divide-and-conquer approach to compute the gradients. At each iteration, the data set is randomly partitioned into groups of 4 points called quartets [24]. The previously stated mMDS stress function is redefined to put independent quartets of points to the forefront, in order to compute the gradients for each point by looking only at the three other points in the quartet at each SGD iteration. The proposed method has been coined "SQuaD-MDS", standing for "Stochastic Quartet Descent MDS".

The motivation behind the use of quartets of points comes from triangulation: in a 2-dimensional plane, a point can be accurately placed given its distances to three other points and their locations. This work assumes a target dimensionality of 2 but adapting it to a target dimensionality of 3 would only require using groups of 5 points instead of 4.

Let us start by rewriting the previously defined stress function of mMDS in a way that highlights quartets of points:

$$\mathscr{L}^{mMDS} = \frac{1}{\binom{N}{2}} \sum_{i<j} \Delta_{ij}^2 = \frac{1}{\binom{N}{4}} \sum_{i<j<k<l} \frac{1}{\binom{4}{2}} \mathscr{L}^{2}_{ijkl}, \tag{1}$$

where $\mathscr{L}^{Q}_{ijkl}$ is the stress term for a single quartet composed of the points indexed by i, j, k, and l; that is,

$$\mathscr{L}^{Q}_{ijkl} = \Delta_{ij}^2 + \Delta_{ik}^2 + \Delta_{il}^2 + \Delta_{jk}^2 + \Delta_{jl}^2 + \Delta_{kl}^2 .$$

Each possible quartet of data points appears exactly once in the right hand side of (1). Such a formulation is particularly well suited to the deployment of stochastic gradient descent, by considering the quartets separately during the optimization process.

Instead of deriving the gradients directly from $\mathscr{L}^{Q}_{ijkl}$, SQuaD-MDS uses an alternate version of the quartet stress that uses relative distances, $\mathscr{L}^{Q_{rel}}_{ijkl}$. The relative distances are defined as being the absolute distance divided by the sum of the six distances that appear in the quartet. The relative distance between the i^{th} and j^{th} points belonging to the quartet composed of the points indexed by i, j, k, and l are noted δ^{rel}_{ijkl} and d^{rel}_{ijkl} in HD and LD, respectively

$$\delta^{rel}_{\mathbf{ijkl}} = \frac{\delta_{ij}}{\delta_{ij}+\delta_{ik}+\delta_{il}+\delta_{jk}+\delta_{jl}+\delta_{kl}} \;;$$
$$d^{rel}_{\mathbf{ijkl}} = \frac{d_{ij}}{d_{ij}+d_{ik}+d_{il}+d_{jk}+d_{jl}+d_{kl}}.$$

The quartet stress term using relative distances simply replaces the absolute distances in $\mathscr{L}^{Q}_{ijkl}$ with relative distances:

$$\mathscr{L}^{Q_{rel}}_{ijkl} = \Delta^{rel\,2}_{\mathbf{ijkl}} + \Delta^{rel\,2}_{\mathbf{ikjl}} + \Delta^{rel\,2}_{\mathbf{ijkl}} + \Delta^{rel\,2}_{\mathbf{jkil}} + \Delta^{rel\,2}_{\mathbf{jikl}} + \Delta^{rel\,2}_{\mathbf{kilj}},$$

where

$$\Delta^{rel}_{\mathbf{ijkl}} = \delta^{rel}_{\mathbf{ijkl}} - d^{rel}_{\mathbf{ijkl}}.$$

The stress term $\mathscr{L}^{Q_{rel}}_{ijkl}$ is derivable with respect to each LD coordinate in the quartet. By naming $\mathbf{I}$ the N-by-N identity matrix, $\mathbf{x}_q$ the LD coordinates of the q^{th} point with $q \in \{i,j,k,l\}$, and $S = d_{ij}+d_{ik}+d_{il}+d_{jk}+d_{jl}+d_{kl}$, the gradient for one element of the sum in $\mathscr{L}^{Q_{rel}}_{ijkl}$ is

$$\frac{\partial(\Delta^{rel\,2}_{ijkl})}{\partial \mathbf{x}_q} = \frac{2(d^{rel}_{ijkl}-\delta^{rel}_{ijkl})}{S}\left(\overbrace{\frac{\mathbf{I}_{qi}(\mathbf{x}_q-\mathbf{x}_j)+\mathbf{I}_{qj}(\mathbf{x}_q-\mathbf{x}_i)}{d_{ij}}}^{\text{A}} \overbrace{- d^{rel}_{ijkl}\sum_{\substack{b\in\{i,j,k,l\}\\ b\neq q}}\frac{\mathbf{x}_q-\mathbf{x}_b}{d_{qb}}}^{\text{B}}\right)$$

(A) is zero if $q \notin \{i,j\}$, this part of the gradient directly moves the point towards or further from the i^{th} or j^{th} point, similarly to the forces that would be exerted using gradients derived from a stress function using absolute distances. (B) comes from the relative distances. This part of the equation is a more global term that influences the quartet stress by changing the denominator in the definition of the relative distances. For instance, if $q \notin \{i,j\}$ and $d^{rel}_{\mathbf{ijkl}}$ is too small compared to $\delta^{rel}_{\mathbf{ijkl}}$, then the q^{th} point will be attracted to the i^{th} and j^{th} points in order to increase $d^{rel}_{\mathbf{ijkl}}$ by decreasing its denominator.

The gradients for each point are computed in constant time relative to the number of observations N and each point receives an update at each iteration; the time complexity of each SGD iteration is therefore $\mathcal{O}(N)$. The full HD and LD distance matrices are not necessary as the distances can be computed on the fly when the quartets are designated; the spatial complexity of the proposed method is therefore also low with respect to N.

So far, this subsection has explained how the proposed method randomly partitions the data into quartets of points at each SGD iteration. Once the data is organised into quartets, the gradients are derived from the quartet-specific cost function $\mathscr{L}^{Q_{rel}}_{ijkl}$. These gradients, however, are not applied directly to the LD coordinates; the remaining of this subsection introduces the specificities of the SGD strategy taken by SQuaD-MDS.

The partitioning into quartets at each iteration is a sampling process where $\lfloor \frac{N}{4} \rfloor$ non-overlapping quartets are sampled from all the $\binom{N}{4}$ possible combinations of quartets. Because of the stochastic nature of this process, the learning rate is decayed following $(a \cdot t + b)^{-1}$ with a and b as constants and t the iteration number, in accordance to the Robbins-Monro principles for stochastic optimisation [25].

Additionally, the optimisation updates each coordinate by keeping track of their momentum. The intuition motivating this approach is that the momenta capture a larger part of the sum of quartet costs in (1), bringing updates that better follow the global energy potentials than if using standalone gradients. This makes the optimisation process more robust with respect to randomness.

In practice, this work uses an improved version of the momentum introduced by Yurii Nesterov in [26]. The Nesterov method first computes temporary coordinates by updating the current coordinates with their momentum; the gradients are computed using these temporary coordinates. The gradients are then used to update the momenta; finally, the real coordinates are changed by following the updated momenta. This process is described below, with θ_t and v_t a parameter at iteration t and its momentum, $\nabla_\theta J(\theta)$ the gradient of the cost function with respect to the parameter θ, and η and γ constants balancing the weight of the momentum relatively to the gradient:

$$v_t = \gamma \cdot v_{t-1} - \eta \nabla_\theta J(\theta_{t-1} + \gamma \cdot v_{t-1}),$$
$$\theta_t = \theta_{t-1} + v_t.$$

The Nesterov method for gradients brings faster convergence in a non-stochastic context. Despite the fact that the proposed optimisation is a stochastic process, the usage of Nesterov momenta improves the quality of the results while only requiring a negligible supplementary cost in terms of computations.

Some gradient descent methods add an additional sophistication to the optimisation by using the Hessian matrix or an approximation of the matrix to determine norm of the gradients [27]. This can yield efficient updates in a non-stochastic context, but at the cost of additional computations. This work relies on a large number of fast iterations cemented together by a strong momentum; computing an approximation of the Hessian matrix at each iteration would significantly slow down the algorithm and risk propagating the noise from the stochastic environment to the Hessian matrix. For this reason, the SGD process proposed in this work does not adopt principles used in second order methods.

Algorithm 1 shows the optimisation loop proposed for SQuaD-MDS, using Nesterov's momentum. The procedure *fill_MDS_grads()* computes the gradients for $\mathscr{L}^{Q_{rel}}_{ijkl}$, for each point in a quartet defined by the point indices $\{i,j,k,l\}$. Multiplying the momenta by 0.99 at each iteration allows them to decay and thus to keep pace with the decreasing learning rate without overwhelming it. The value 0.99 was determined empirically, but values in the interval $[0.9, 1]$ bring similar results. The PCA initialisation can be replaced with a random initialisation, but informed initialisation using (very) fast methods like PCA are standard in DR as they tend to yield results of better quality [17].

Algorithm 1 Optimisation loop for SQuaD-MDS

embedding ← *PCA*(X) ▷initialise the embedding with PCA
momenta ← *zeros*(N, 2) ▷contains the momentum of each parameter
learning_rate ← 1
for *iteration* **in** $[0, \dots, max_iterations]$ **do**
gradients ← *zeros*(N, 2)
learning_rate ← *decay*(*learning_rate*)
temp_params = *embedding* + *momenta* ▷used for Nesterov's momentum
quartets ← *random_partition*() ▷partitions the data into quartets
for *quartet* **in** *quartets* **do**
fill_quartet_gradients(X, *temp_params*, *quartet*, *gradients*)
end for
momenta ← *momenta* $*$ 0.99 − *learning_rate* $*$ *gradients*
embedding ← *embedding* + *momenta*
end for

3.2. Motivation for the use of relative distances

The main reason behind the usage of relative distances is to remove the implicit scale constraint that appears when preserving the absolute distances. Constraining the scale of the embedding is deemed unnecessary as the LD space is composed of abstract variables that are not directly observable: the information in the embedding is carried by the relative positions of the points, not by their absolute distances. Not having a scale constraint is also useful to evade conflicts concerning the scale when mixing the SQuaD-MDS gradients with those of other DR methods as described in the following subsection.

3.3. Hybrid extension

Since this work optimises a distance preserving cost function using SGD, the gradients can be blended with those of other DR methods that also rely on gradient descent. The objective of such a hybridisation is to produce embeddings that minimise multiple cost functions for better overall quality. This work suggests mixing SQuaD-MDS's gradients with those of t-SNE [14]. The motivation behind the choice of t-SNE for hybridisation is that t-SNE and mMDS have complementary objectives: mMDS focuses on global structures and t-SNE on local structures.

The exact version of t-SNE has a high time complexity of $\mathcal{O}(N^2)$; blending it with fast mMDS gradients would defeat the purpose of this work entirely. Since the cost function of t-SNE focuses on the small-scale structures of data, t-SNE requires some knowledge on the nearest neighbours in the low dimensional space. This cannot be done directly by using the quartet sampling proposed for mMDS, as the probability of having close-by points in a quartet is too low if the quartet sampling is kept unbiased.

A few accelerated approximations of t-SNE exist. We use FIt-SNE [15], which uses gradient descent with iterations of $\mathcal{O}(N)$ time complexity, the same complexity as SQuaD-MDS. The details of t-SNE and FIt-SNE are outside of the scope of this paper and can be found in [14,15].

In order to combine both methods, this work keeps the same protocol as for the standalone version of SQuaD-MDS: the learning rates are decayed and Nesterov's momentum is used.

Since the FIt-SNE and SQuaD-MDS gradients are computed using different cost functions, there is no guarantee that their norms are of comparable magnitude. This is problematic when aiming to mix the gradients together, as one gradient type could overwhelm the effect of the other type by having naturally bigger norms. In order to reduce the difference in magnitude between the gradients of both type, the standard deviation of the norm of the SQuaD-MDS gradients and FIt-SNE gradients are computed independently over the LD points. The gradients of each type are then divided by the corresponding standard deviations as a form of normalisation. Once the gradients normalised, learning rates individual to each method are used to combine the gradients. Changing the balance between the learning rates of FIt-SNE and SQuaD-MDS enables the user to prioritize one paradigm over the other, focusing on either local or global preservation. This work suggests initial learning rate values of 1 for FIt-SNE and 0.5 for SQuaD-MDS as default values, based on empirical experiments.

t-SNE and FIt-SNE require a crucial scalar hyperparameter called perplexity. A simplistic but sufficient way to view the perplexity here is to consider it as the rough number of neighbours that will be considered when embedding the points: a small perplexity will produce an embedding that preserves well the close neighbours of each point and a large perplexity will consider larger sets of neighbours. A high perplexity is not always preferable as it comes at a price in terms of preservation of small-scale structures.

The purpose of FIt-SNE's gradients in the hybridisation with SQuaD-MDS is to preserve the small-scale structures. For this reason, this work uses multi-scale similarities [28,29] limited to small perplexities. As a default, this work suggests perplexities equal to 4 and 50. These values were determined empirically; for very large data sets, the perplexity of 50 may be slightly increased for better results.

4. Empirical quality assessment

Qualitative and quantitative assessments of SQuaD-MDS and its hybrid extension are presented in this section.

SubSection 4.1 introduces the data sets used for the evaluation and brings a short introduction to $R_{NX}(K)$ curves [28], which are used to assess the quality of the neighbour preservation across all scales. In order to evaluate the performance of the standalone SQuaD-MDS algorithm, SubSection 4.2 compares its results to mMDS embeddings as yielded by the SMACOF algorithm. SubSection 4.3 assesses the quality of embeddings obtained with the hybrid approach by comparing them to embeddings produced by FIt-SNE and UMAP.

4.1. Data sets and quality curves

Eight data sets are featured in this paper, with size N and dimensionality M. MNIST digits: (N, M) = (60000, 784); MNIST fashion: (N, M) = (60000, 784); gesture phase segmentation (gestures) [30]: (N, M) = (9901, 784); airfoil self-noise (airfoil): (N, M) = (1502, 5); Boston housing (housing): (N, M) = (404, 13); Statlog landsat satellite: (N, M) = (4434, 36); single-cell RNA-seq data from the adult mouse cortex (mouse cortex) [19]: (N, M) = (23822, 50); single-cell RNA-seq data of a zebrafish embryo coloured by time after fertilisation (zebrafish) [31]: (N, M) = (63530, 50).

Both single-cell data sets are composed of their 50 principal components after going through the preprocessing pipeline described in [18]. Most of the other data sets are available on the UCI machine learning data repository [32].

The HD neighborhood reproduction in the LD embedding is deemed as driving the DR quality in numerous studies [33–36]. Some criteria have been developed to quantify this reproduction [37,38]. One can first denote as ν_i^K and n_i^K the K nearest neighbors of the i^{th} data point, respectively in the HD and LD spaces. These K-

ary neighborhoods have an average normalized intersection developing as

$$Q_{\mathrm{NX}}(K) = \frac{1}{N}\sum_{i=1}^{N}\frac{|\nu_i^K \cap n_i^K|}{K} \in [0,1].$$

As defining the LD coordinates randomly leads to $\mathbb{E}[Q_{\mathrm{NX}}(K)] = K/(N-1)$,

$$R_{\mathrm{NX}}(K) = \frac{(N-1)Q_{\mathrm{NX}}(K) - K}{N-1-K}$$

is an indicator that allows to compare neighborhoods with varying sizes [39]. Since the literature generally believes local neighborhoods as more important [14,40,41] and as larger scales outnumber smaller ones by far, $R_{\mathrm{NX}}(K)$ can be plotted using a log-scale for K. The area under the obtained curve (AUC),

$$\mathrm{AUC} = \left(\sum_{K=1}^{N-2} K^{-1}\right)^{-1} \cdot \sum_{K=1}^{N-2}\frac{R_{\mathrm{NX}}(K)}{K} \in [-1,1],$$

quantifies DR quality by considering all scales, focusing on smaller ones preferentially [28,42]. Greater (resp. smaller) values indicate better (resp. lower) quality.

4.2. Assessment of the standalone SQuaD-MDS

An empirical evaluation of the time taken by 5000 iterations of SQuaD-MDS across multiple data set sizes N is shown in Fig. 2. The time taken by the SMACOF algorithm using the default set of hyperparameters proposed by the *scikit-learn* library in Python [43] is shown on the leftmost plot.

This figure shows that the proposed method took less than 6 s to produce an embedding when $N = 10^4$, whereas the SMACOF algorithm required more than 600 s in the same testing environment. The linear time complexity of SQuaD-MDS with regards to N is apparent on the rightmost plot.

Fig. 3 shows the evolution of the mean quartet loss $\mathscr{L}_{ijkl}^{Q_{\mathrm{nrl}}}$ over the course of its optimisation, for 3 samples sizes (600, 6000, and 60000) of the zebrafish data set [31]. This plot shows that the number of points in the data set does not appear to have an impact on the total number of iterations required to minimise the cost function of SQuaD-MDS.

The quality of embeddings produced by the SMACOF algorithm and SQuaD-MDS can be compared in Figs. 4 and 5. The data sets larger than 5000 points were sampled down to 5000 for both tested methods in Fig. 4, in order to cope with the high computa-

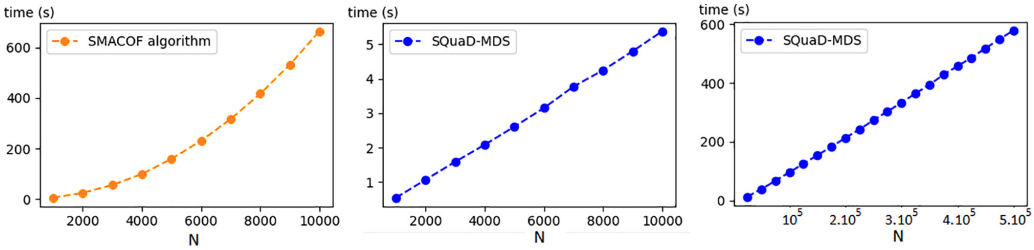


Fig. 2. Time taken by the SMACOF algorithm and SQuaD-MDS to yield an embedding, as a function of the data set size N.

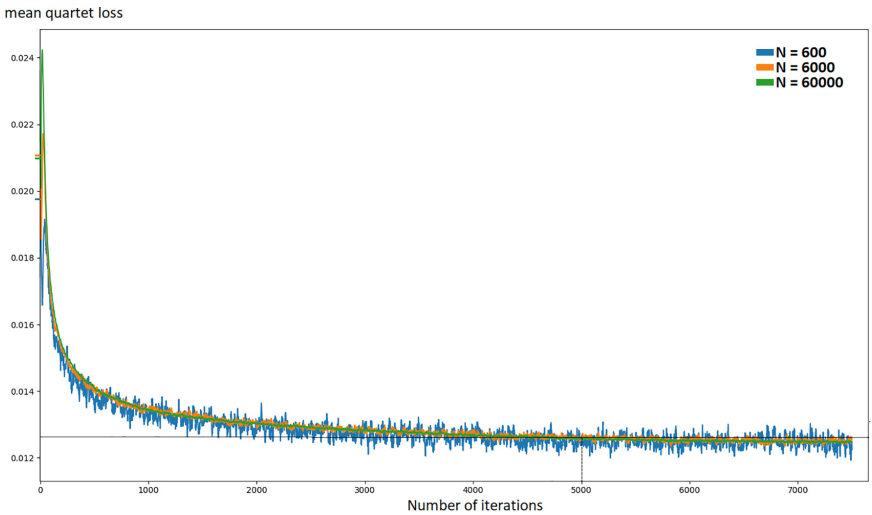


Fig. 3. Evolution of the mean quartet loss across 7500 iterations for three sample sizes N of the zebrafish data set.

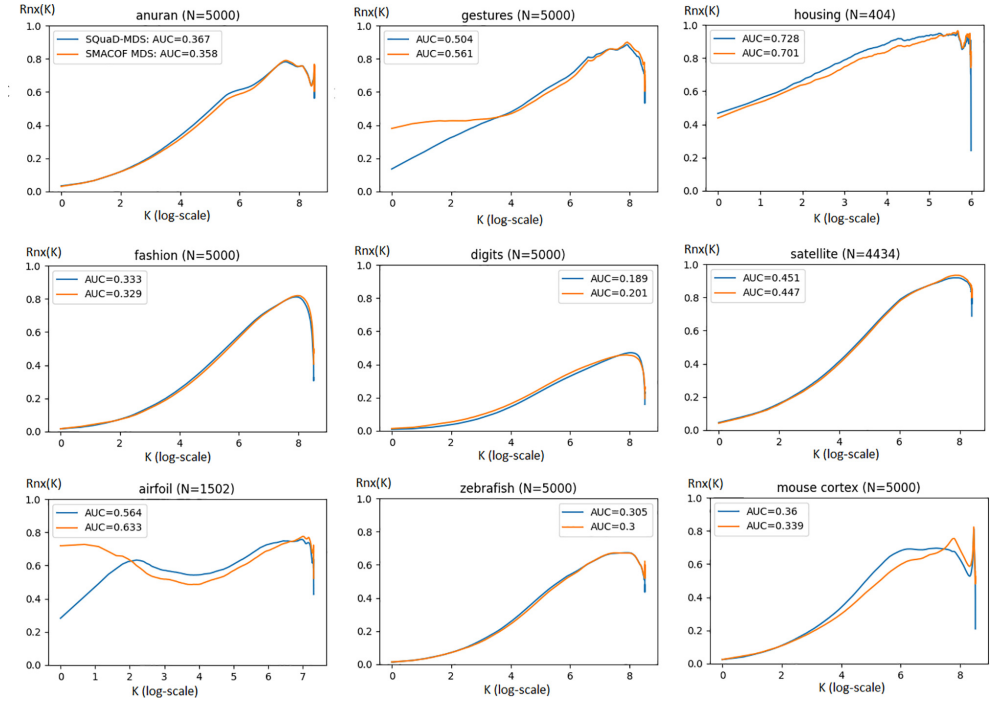


Fig. 4. $R_{NX}(K)$ curves of embeddings produced by the SMACOF algorithm and SQuaD-MDS.

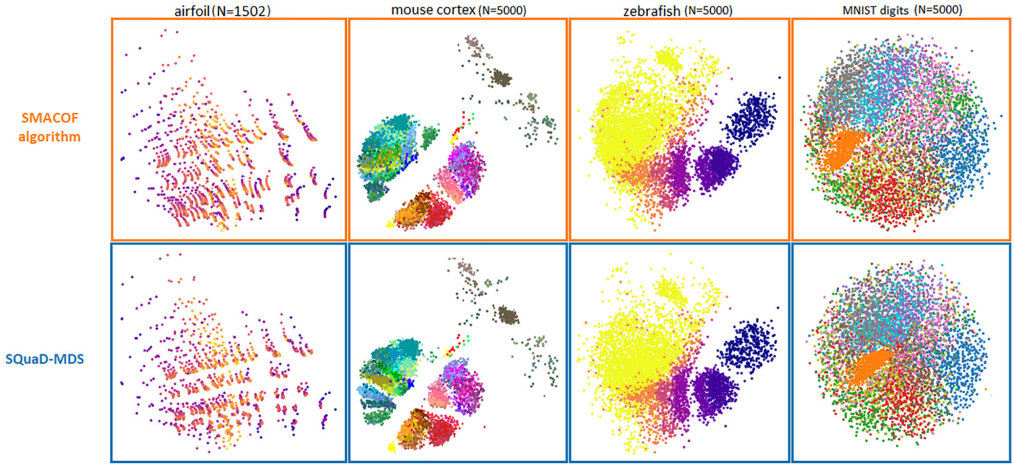


Fig. 5. Embeddings produced by the SMACOF algorithm and SQuaD-MDS.

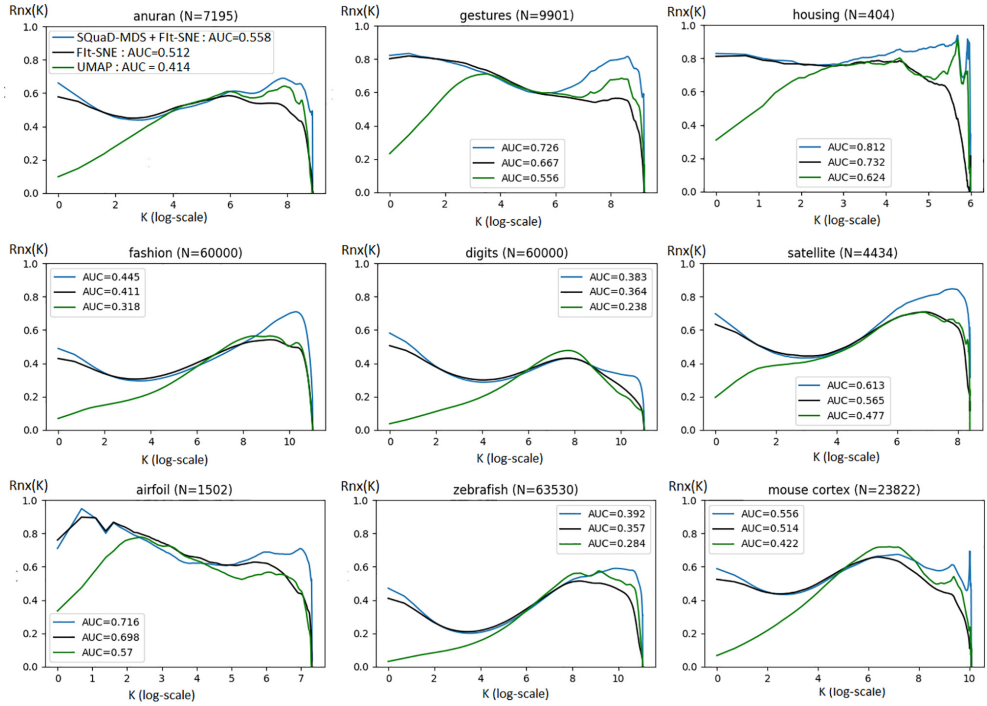


Fig. 6. $R_{NX}(K)$ curves of embeddings produced by the proposed hybrid method, FI*t*-SNE, and UMAP.

tional complexity of the SMACOF algorithm and keep a fair comparison.

The number of iterations for the SQuaD-MDS embeddings and the hyperparameters of the SMACOF algorithm are the same as for the time evaluation in Fig. 2.

The $R_{NX}(K)$ curves of Fig. 4 show that both methods tend to favor the preservation of global structures (high $R_{NX}(K)$ values on large K). The AUC are often similar for both methods, although in the gestures and airfoil data sets the SMACOF algorithm performs better at preserving small-scale structures than SQuaD-MDS.

This can be explained by the fact that SQuaD-MDS is stochastic and largely subsamples the population of all possible quartets, neglecting many of the small-area quartets, which leaves some noise in the shortest distances and hence also in the final coordinates of the data points. This phenomenon can be qualitatively observed in Fig. 5 in the airfoil embeddings: the fine 1-dimensional manifolds observed in the SMACOF embedding are not as delicate in the SQuaD-MDS embedding. The shortcomings of SQuaD-MDS in terms of small-scale structures with some data sets motivate the hybrid approach with *t*-SNE. The weakness of SQuaD-MDS concerning the preservation of fine structures on certain data sets is limited in practice, as the purpose of mMDS is generally to explore the global organisation of the data more than the local structures.

4.3. Assessment of the hybrid extension

The hybrid method combines the distance scaling paradigm to the neighbour embedding paradigm by using gradients from SQuaD-MDS and FI*t*-SNE. In order to assess the quality of the embeddings produced by the hybrid method, they are compared to those produced by FI*t*-SNE [15] and UMAP [16]; both are popular DR methods focusing on the small-scale structures in the data.

For the sake of comparison, all the methods use the same PCA initialization of the embedding.

The hybrid method was applied using the proposed default hyperparameters, notably perplexity values of 4 and 50 to compute the multi-scale similarities. The number of iterations of the optimisation was set to 750, which is the default value in the implementation of FI*t*-SNE made by the authors.

For the sake of comparison, the FI*t*-SNE embeddings were also produced by combining perplexities of 4 and 50; all other hyperparameters were kept to their default values.

The UMAP method builds on different principles than *t*-SNE and its accelerated versions; for this reason, the choice of hyperparameters to conduct a comparison is not obvious. Similarly to the perplexity, UMAP takes a hyperparameter called the number of neighbours; it is responsible of the balance between the preservation of local and global structures in the data. Despite having the same purpose, the number of neighbours used by UMAP does not translate exactly to the perplexity used by *t*-SNE in terms of effective balance between small-scale and large-scale structures. To produce UMAP embeddings of quality while achieving a similar trade-off between the preservation of small-scale and large-scale structures as in the other methods, the following experiments apply UMAP ten times per data set with a number of neighbours taking values in $5, 10, 15, \ldots, 55$. From the ten embeddings, the one with the best AUC of the $R_{NX}(K)$ curve is kept and considered in the results.

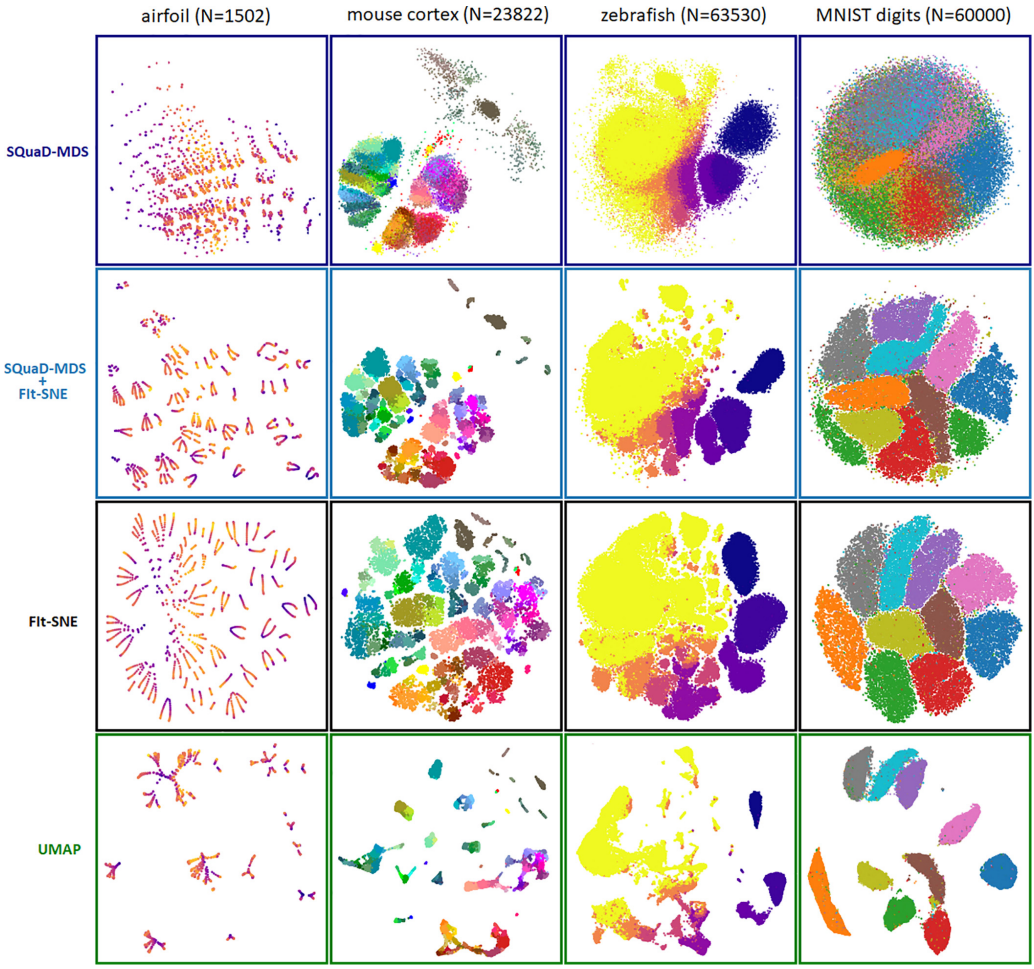


Fig. 7. Embeddings produced by SQuaD-MDS, the proposed hybrid method, FI*t*-SNE, and UMAP.

The $R_{NX}(K)$ curves presented in Fig. 6 show a comparison of the quality of the embeddings produced by the three methods across multiple scales. On the tested data sets, the hybrid method has similar $R_{NX}(K)$ values to those of FI*t*-SNE on the small and medium scales, but its scores on large K values are higher. This shows that the combination of distance preserving gradients to those of FI*t*-SNE can enhance the preservation of the large-scale structures of the data, without impeding on the preservation of the small-scale structures.

A qualitative comparison of the methods can be observed in Fig. 7. Visually, the global organisation of the points in the embeddings produced by the hybrid method is close to the organisation observed in the mMDS embeddings. However, the points tend to form the sharp and easy-to-differentiate clusters that are typical of neighbour embeddings [14].

In the zebra fish data set, the colours correspond to an amount of time since fertilisation of the embryo. On the bottom of the embeddings produced by SQuaD-MDS and the hybrid method, the transition from purple to yellow representing the passing time is more linear from right to left than in the FI*t*-SNE embedding, revealing another point of view on the organisation of the data.

Similarly, the hierarchical organisation of the mouse cortex data set is clearly visible in the embeddings produced by SQuaD-MDS and the hybrid method. The gray non-neuron cells are kept far from the coloured neurons, and the neurons are themselves organised in two large clusters of inhibitory (warm colours) and excitatory (cold colours) neurons. Some finer-grained structures are also apparent within each cluster, especially in the hybrid embedding thanks to the influence of the FI*t*-SNE gradients.

5. Conclusion

The fast distance-preserving gradients proposed in this work enable the application of mMDS to large data sets. Empirically,

the large-scale and medium-scale structures of the data are preserved similarly between SQuaD-MDS and the popular SMACOF algorithm, but the small-scale structures can suffer from the stochastic nature of SQuaD-MDS on some data sets. It is however noteworthy that the quadratic time complexity of the SMACOF algorithm with respect to the number of data points prohibits its application on large data sets, while the linear time complexity of SQuaD-MDS is considerably more affordable.

Experiments show that merging the SQuaD-MDS gradients with those of a neighbour embedding method can yield embeddings that both preserve the large-scale structures of the data and the local structures, thereby addressing an often reported issue of almost exclusively local methods like *t*-SNE and its variants, including UMAP.

Multiple perspectives arise from the quartet-based computation of distance gradients. For instance, the HD distances computed on each considered quartet could be first transformed nonlinearly to emphasize either large or small distances, or even to mitigate the effects of the concentration of norms, which tends to make the application of MDS challenging on very high dimensional data sets [44]. Yet another possibility is to consider a different strategy when sampling the quartets; for instance, increasing the probability to regroup points that are close by in the embedding when defining the quartets could help to better preserve the local structures of data.

CRediT authorship contribution statement

Pierre Lambert: Conceptualization, Writing – original draft. **Cyril de Bodt:** Writing – review & editing, Writing – original draft. **Michel Verleysen:** Supervision, Writing – review & editing, Writing – original draft. **John A. Lee:** Supervision, Writing – review & editing, Writing – original draft.

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.

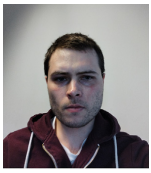
Acknowledgements

This work was supported by Service Public de WallonieRecherche under grant n°2010235 -ARIAC by DIGITALWALLONIA4.AI. JAL is a Senior Research Associate with the Fonds de la Recherche Scientifique-FNRS.

References

[1] J. de Leeuw, Modern multidimensional scaling: Theory and applications (second edition), J. Stat. Software 14. doi:10.18637/jss.v014.b04..
[2] J. de Leeuw, P. Mair, Multidimensional scaling using majorization: Smacof in r, J. Stat. Software 31 (3) (2009) 1–30, https://doi.org/10.18637/jss.v031.i03, URL: https://www.jstatsoft.org/v031/i03.
[3] J.A. Lee, M. Verleysen, Nonlinear Dimensionality Reduction, first ed., Springer Publishing Company, Incorporated, 2007.
[4] M.N. Joswiak, Y. Peng, I. Castillo, L.H. Chiang, Dimensionality reduction for visualizing industrial chemical process data, Control Eng. Practice 93 (2019) 104189.
[5] B. Wang, Z. Li, Z. Dai, N. Lawrence, X. Yan, Data-driven mode identification and unsupervised fault detection for nonlinear multimode processes, IEEE Trans. Industr. Inf. 16 (6) (2020) 3651–3661, https://doi.org/10.1109/TII.2019.2942650.
[6] P. Gautam, K. Hamashima, Y. Chen, Y. Zeng, B. Makovoz, B. Parikh, H. Lee, K. Lau, X. Su, R. Wong, W. Chan, H. Li, T. Blenkinsop, Y. Loh, Multi-species single-cell transcriptomic analysis of ocular compartment regulons, Nat. Commun. 12(1)..
[7] A.P.A. Janssen, S.H. Grimm, R.H.M. Wijdeven, E.B. Lenselink, J. Neefjes, C.A.A. van Boeckel, G.J.P. van Westen, M. van der Stelt, Drug discovery maps, a machine learning model that visualizes and predicts kinome-inhibitor interaction landscapes, J. Chem. Inf. Model. 59(3) (2019) 1221–1229, pMID: 30372617. doi:10.1021/acs.jcim.8b00640..
[8] H.T. Lawless, N. Sheng, S.S. Knoops, Multidimensional scaling of sorting data applied to cheese perception, Food Qual. Prefer. 6 (2) (1995) 91–98.
[9] K.P.F.R.S. Liii, on lines and planes of closest fit to systems of points in space, London Edinburgh Dublin Philos. Mag. J. Sci. 2(11) (1901) 559–572. doi:10.1080/14786440109462720..
[10] B. Schölkopf, A. Smola, K.-R. Müller, Kernel principal component analysis, in: W. Gerstner, A. Germond, M. Hasler, J.-D. Nicoud (Eds.), Artificial Neural Networks — ICANN'97, Springer Berlin Heidelberg, Berlin, Heidelberg, 1997, pp. 583–588.
[11] J.B. Tenenbaum, V. de Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (5500) (2000) 2319–2323. doi:10.1126/science.290.5500.2319..
[12] D. DeMers, G. Cottrell, Non-linear dimensionality reduction, NIPS (1992).
[13] G.E. Hinton, S. Roweis, Stochastic neighbor embedding, in: S. Becker, S. Thrun, K. Obermayer (Eds.), Advances in Neural Information Processing Systems, vol. 15, MIT Press, 2002, URL: https://proceedings.neurips.cc/paper/2002/file/6150ccc6069bea6b5716254057a194ef-Paper.pdf.
[14] L. van der Maaten, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (2008) 2579–2605.
[15] G.C. Linderman, M. Rachh, J.G. Hoskins, S. Steinerberger, Y. Kluger, Fast interpolation-based t-sne for improved visualization of single-cell rna-seq data, Nat. Methods 16 (3) (2019) 243–245, https://doi.org/10.1038/s41592-018-0308-4.
[16] L. McInnes, J. Healy, J. Melville, Umap: Uniform manifold approximation and projection for dimension reduction (2020). arXiv:1802.03426..
[17] D. Kobak, G.C. Linderman, Initialization is critical for preserving global data structure in both t-sne and umap, Nat. Biotechnol..
[18] D. Kobak, P. Berens, The art of using t-sne for single-cell transcriptomics, Nat. Commun. 10, 5416..
[19] B. Tasic, Z. Yao, L.T. Graybuck, K.A. Smith, T.N. Nguyen, D. Bertagnolli, J. Goldy, E. Garren, M.N. Economo, S. Viswanathan, et al., Shared and distinct transcriptomic cell types across neocortical areas, Nature 563 (7729) (2018) 72–78.
[20] A. Morrison, G. Ross, M. Chalmers, Fast multidimensional scaling through sampling, springs and interpolation, Inf. Visualiz. 2 (2003) 68–77, https://doi.org/10.1057/palgrave.ivs.9500040.
[21] V. Silva, J. Tenenbaum, Sparse multidimensional scaling using landmark points, Technology..
[22] T. Yang, J. Liu, L. Mcmillan, W. Wang, A fast approximation to multidimensional scaling..
[23] M. Williams, T. Munzner, Steerable, progressive multidimensional scaling, in: IEEE Symposium on Information Visualization, 2004, pp. 57–64. doi:10.1109/INFVIS.2004.60..
[24] P. Lambert, C. de Bodt, M. Verleysen, J.A. Lee, Stochastic quartet approach for fast multidimensional scaling, ESANN (2021) 417–422.
[25] H. Robbins, S. Monro, A Stochastic Approximation Method, Ann. Math. Stat. 22 (3) (1951) 400–407, https://doi.org/10.1214/aoms/1177729586.
[26] Y. Nesterov, A method for solving the convex programming problem with convergence rate $o(1/k^2)$, Proc. USSR Acad. Sci. 269 (1983) 543–547.
[27] D.C. Liu, J. Nocedal, On the limited memory bfgs method for large scale optimization, Math. Program. 45 (1989) 503–528.
[28] J.A. Lee, D.H. Peluffo-Ordóñez, M. Verleysen, Multi-scale similarities in stochastic neighbour embedding: Reducing dimensionality while preserving both local and global structure, Neurocomputing 169 (2015) 246–261, https://doi.org/10.1016/j.neucom.2014.12.095.
[29] C. de Bodt, D. Mulders, M. Verleysen, J.A. Lee, Perplexity-free t-SNE and twice Student tt-SNE, ESANN (2018) 123–128.
[30] R.C.B. Madeo, C.A.M. Lima, S.M. Peres, Gesture unit segmentation using support vector machines: Segmenting gestures from rest positions, in: Proceedings of the 28th Annual ACM Symposium on Applied Computing, SAC '13, Association for Computing Machinery, New York, NY, USA, 2013, p. 46–52. doi:10.1145/2480362.2480373..
[31] D.E. Wagner, C. Weinreb, Z.M. Collins, J.A. Briggs, S.G. Megason, A.M. Klein, Single-cell mapping of gene expression landscapes and lineage in the zebrafish embryo, Science 360 (6392) (2018) 981–987. doi:10.1126/science.aar4362..
[32] D. Dua, C. Graff, UCI machine learning repository (2017). URL: http://archive.ics.uci.edu/ml..
[33] J. Venna, J. Peltonen, K. Nybo, H. Aidos, S. Kaski, Information retrieval perspective to nonlinear dimensionality reduction for data visualization, J. Mach. Learn. Res. 11 (13) (2010) 451–490.
[34] L. Chen, A. Buja, Local multidimensional scaling for nonlinear dimension reduction, graph drawing, and proximity analysis, J. Am. Stat. Assoc. 104 (485) (2009) 209–219.
[35] C. de Bodt, D. Mulders, M. Verleysen, J.A. Lee, Nonlinear Dimensionality Reduction with Missing Data using Parametric Multiple Imputations, IEEE Trans. Neural Networks Learn. Syst. 30 (4) (2019) 1166–1179, https://doi.org/10.1109/TNNLS.2018.2861891.
[36] B. Mokbel, W. Lueks, A. Gisbrecht, B. Hammer, Visualizing the quality of dimensionality reduction, Neurocomputing 112 (2013) 109–123.
[37] J.A. Lee, M. Verleysen, Quality assessment of dimensionality reduction: Rank-based criteria, Neurocomputing 72 (7) (2009) 1431–1443.
[38] J.A. Lee, M. Verleysen, Scale-independent quality criteria for dimensionality reduction, Pattern Recogn. Lett. 31 (14) (2010) 2248–2257.
[39] J.A. Lee, E. Renard, G. Bernard, P. Dupont, M. Verleysen, Type 1 and 2 mixtures of kullback–leibler divergences as cost functions in dimensionality reduction based on similarity preservation, Neurocomputing 112 (2013) 92–108.

[40] C. de Bodt, D. Mulders, D. López-Sánchez, M. Verleysen, J.A. Lee, Class-aware t-SNE: cat-SNE, ESANN (2019) 409–414.
[41] S. Roweis, L. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (5500) (2000) 2323–2326, https://doi.org/10.1126/science.290.5500.2323.
[42] C. de Bodt, D. Mulders, M. Verleysen, J.A. Lee, Fast multiscale neighbor embedding, IEEE Trans. Neural Networks Learn. Syst. (2020) 1–15, https://doi.org/10.1109/TNNLS.2020.3042807.
[43] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830.
[44] D. Francois, V. Wertz, M. Verleysen, The concentration of fractional distances, IEEE Trans. Knowl. Data Eng. 19 (7) (2007) 873–886.



Pierre Lambert received the M.Sc. degree in computer science from the Université catholique de Louvain (UCLouvain) in 2021, he is now a PhD student in UCLouvain, working on nonlinear dimensionality reduction. His interests include (nonlinear) dimensionality reduction, data visualization, optimization, unsupervised learning, and machine-learning methods inspired by biology.



Cyril de Bodt was born in Brussels, Belgium, in 1993. He received the M.Sc. degree in Applied Sciences (Mathematical Engineering) in 2016 and the Ph.D. degree in Engineering Science and Technology in 2020, both from UCLouvain, Belgium. From Nov. 2020 to Nov. 2021, he was a Postdoctoral Research Fellow in the Human Dynamics group of the Massachusetts Institute of Technology (MIT) Media Lab, Cambridge, USA. He is now a Postdoctoral Researcher in the ICTEAM institute of UCLouvain under the ARIAC project of the TRAIL inter-university institute, and a Research Affiliate with the Human Dynamics group of the MIT Media Lab. His research interests include (nonlinear) dimensionality reduction, machine learning, data visualization, neighbor embedding algorithms, optimization, clustering and unsupervised learning.



Michel Verleysen is a Professor of Machine Learning at the UCLouvain, Belgium. He has been an invited professor at EPFL (Switzerland), Université d'Evry Val d'Essonne, Université ParisI-Panthéon-Sorbonne and Université Paris Est (France). He is an Honorary Research Director of the Belgian F.N.R.S. (National Fund for Scientific Research), and the Dean of the Louvain School of Engineering. He is editor-in-chief of the Neural Processing Letters journal (Springer), chairman of the annual ESANN conference (European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning), past associate editor of the IEEE Trans. on Neural Networks journal, and member of the editorial board and program committee of several journals and conferences on neural networks and learning. He is author or co-author of more than 250 scientific papers in international journals and books or communications to conferences with reviewing committee. He is co-author of the "Nonlinear Dimensionality Reduction" book (Springer). His research interests include machine learning, feature selection, nonlinear dimensionality reduction, visualization, high-dimensional data analysis, self-organization, time-series forecasting and biomedical signal processing.



John Lee is a Professor of machine learning, data visualization, and medical imaging at UCLouvain, Belgium. He is Senior Research Associate with the Belgian F.R.S.-FNRS (Fonds National de la Recherche Scientifique) and he is currently the head of the UCLouvain Center of Molecular Imaging, Radiotherapy, and Oncology. Together with Michel Verleysen, he is an author of the book "Nonlinear Dimensionality Reduction" published by Springer in 2007. He is also author or coauthor of more than 170 scientific articles in internation journals or conferences. His research interests include machine learning, dimensionality reduction, clustering, data visualization, as well as medical physics and image processing applied to cancer care.

3.4 Post-hoc explainability for neighbour embeddings

This paper takes advantage the structure preservation properties across multiple scales from the fast hybrid MDS-NE method introduced in Section 3.3 to provide a new post-hoc explainability method for NE, which links local directions in the embeddings with variables in HD. These explanations are shown through local axes in the embeddings, providing local explanations, a particularity here is that when considering multiple adjacent axes, a global picture of how the LD structures relate to HD variables also becomes apparent.
As a second author, contributions in the conceptualisation and in the prototyping of an ad-hoc approach to explaining NE was also made in the publication [88]. There, the LD coordinates are defined as linear projections of HD variables, the gradients propagating to the projections rather than being applied to coordinates directly. A strong regularisation is required to force similar points to have similar projections, else the model can become over-parametrised. The paper for the ad-hoc approach is not included here, as most of the contributions were made by the first author.

As a quick commentary on the subject of explainability in NE, it is difficult to say whether ad-hoc explanations or post-hoc explanations are preferable. Post-hoc approaches have the quality of not constraining the NE optimisation by design, generally allowing embeddings to better capture the HD structures. However, the embeddings of post-hoc approaches are not steered towards configurations that allow for easy explanations. On the other hand, in the case where multiple minima are accessible in the landscape of the cost function, ad-hoc approaches can guide the optimisation towards more explainable configurations. However, the constraints imposed by the explainability model can have a cost on the preservation of HD structures in LD.
Choosing between one approach or the other ultimately depends on priorities: does the user want good interpretations or good structure preservation? A question that is sometimes divisive in ML, as some would say we employ complicated models precisely to do things that are hard to understand for a human, while others would argue that interpretability is a requirement for automatic processing of data or decision making. In the case of data visualisation, the importance of understanding the representa-

tions is more obvious, as the user is expected to draw conclusions from the visualisations. However, one could argue that NE are, in essence, already a form of explanation taking the same approach as works on explainable ML based on examples or counter-examples like [89]. From this standpoint, adding constrains that affect the quality of the embeddings is counterproductive. Within this view, a proper approach to visualising data with NE would involve interactive exploration of the data, where the "explanations" provided by the relative position of observations in the embedding would be used as a basis for the user to select the appropriate tool to extract further explanations tailored for the task.

Globally local and fast explanations of *t*-SNE-like nonlinear embeddings

Pierre Lambert[1,*], Rebecca Marion[2], Julien Albert[2], Emmanuel Jean[3], Sacha Corbugy[2] and Cyril de Bodt[1,4]

[1]UCLouvain - ICTEAM & TRAIL, Louvain-la-Neuve, Belgium
[2]UNamur - NaDI/PReCISE & TRAIL, Namur, Belgium
[3]Multitel & TRAIL, Mons, Belgium
[4]MIT Media Lab, Cambridge [MA], USA

Abstract

Nonlinear dimensionality reduction (NLDR) algorithms such as *t*-SNE are often employed to visually analyze high-dimensional (HD) data sets in the form of low-dimensional (LD) embeddings. Unfortunately, the nonlinearity of the NLDR process prohibits the interpretation of the resulting embeddings in terms of the HD features. State-of-the-art studies propose post-hoc explanation approaches to locally explain the embeddings. However, such tools are typically slow and do not automatically cover the entire LD embedding, instead providing local explanations around one selected data point at a time. This prevents users from quickly gaining insights about the general explainability landscape of the embedding. This paper presents a **globally local** and **fast** explanation framework for NLDR embeddings. This framework is **fast** because it only requires the computation of sparse linear regression models on subsets of the data, without ever reapplying the NLDR algorithm itself. In addition, the framework is **globally local** in the sense that the entire LD embedding is automatically covered by multiple local explanations. The different interpretable structures in the embedding are directly characterized, making it possible to quantify the importance of the HD features in various regions of the LD embedding. An example use-case is examined, emphasizing the value of the presented framework. Public codes and a software are available at https://github.com/PierreLambert3/glocally_explained.

Keywords

dimensionality reduction, data visualization, interactivity, interpretability, explainability, *t*-SNE, data exploration

1. Introduction

Dimensionality reduction (DR) computes low-dimensional (LD) representations of high-dimensional (HD) data, e.g., to visually explore them or to curb the curse of dimensionality [1]. The relevance of a DR method for a given visualization task typically depends on its preservation of the HD neighborhoods in the resulting LD embedding [2]. Two major frameworks have been proposed for projecting from HD to LD coordinates [1]: one is based on preserving distances [3], while the other is based on reproducing neighborhoods [4, 5]. For instance, distance-preserving methods like principal component analysis (PCA) [6] and classical metric multidimensional scaling (MDS) [3] project HD samples linearly; nonlinear variants of these methods (e.g., [7, 8]) aim to preserve weighted Euclidean or approximately geodesic distances. Numerous other schemes have also been developed that determine the LD embedding design based on HD affinity matrices [9, 10]. Regrettably, the local neighborhood preservation of all of these techniques is limited in visualization contexts by the norm concentration phenomenon [11, 12], most probably due to their distance-preserving nature [1, 13]. In contrast, the native shift invariance of neighbor embedding (NE) algorithms [14] such as Stochastic Neighbor Embedding (SNE) [5] mitigates this phenomenon, leading to astonishing DR quality. These achievements have naturally encouraged the development of numerous SNE-based methods, such as the popular *t*-SNE [15], UMAP [16], multi-scale perplexity-free approaches [17, 18, 19], etc.

While these nonlinear DR (NLDR) algorithms deliver impressively faithful LD embeddings with respect to the HD data, their intrinsic nonlinearity greatly affects the interpretability of the LD representations. Indeed, the obtained LD dimensions are hardly or most often not interpretable in terms of the HD features [20]. Since NLDR methods are not interpretable by design, previous studies have developed techniques to analyze and interpret the LD embeddings, which is known as post-hoc explanation or interpretability [21]. One can for instance cite [22], which proposes to explain visual LD clusters thanks to

Advances in Interpretable Machine Learning and Artificial Intelligence, 2022 October 21, Atlanta, Georgia, USA
*Corresponding author.
pierre.h.lambert@uclouvain.be (P. Lambert);
cyril.debodt@uclouvain.be (C. de Bodt)
https://github.com/PierreLambert3 (P. Lambert);
https://github.com/cdebodt (C. de Bodt)
0000-0003-2347-1756 (C. de Bodt)
© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
CEUR Workshop Proceedings (CEUR-WS.org)

CEUR-WS.org/Vol-3318/short23.pdf

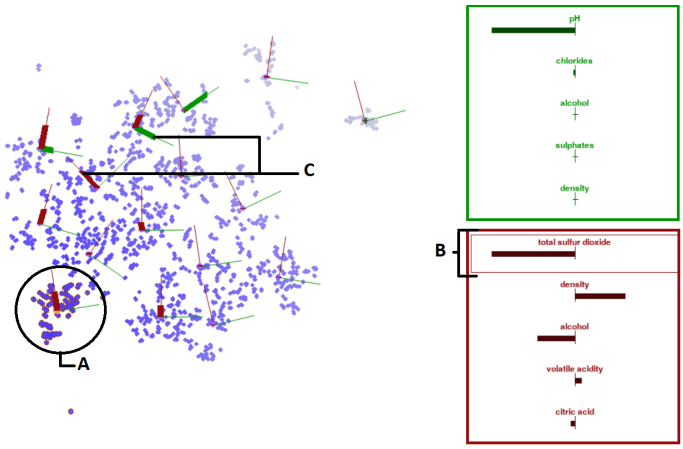


Figure 1: Interface for the proposed globally local and fast explanation framework.

decision trees. On the other hand, [21] locally explain *t*-SNE embeddings by adapting LIME; the authors argue that explaining the entire embedding at once would be difficult, as *t*-SNE usually does not preserve large HD distances well [20]. However, the local nature of *t*-SNE motivates the computation of local explanations in the LD embedding; LIME can then be revisited and performed locally around a user-selected data point. Nevertheless, such an approach has two main limitations: (1) it is slow, and (2) it does not cover the entire LD embedding automatically, as local explanations are only provided around data points that have been selected, one at a time. This approach is slow because, in order to explain a given data point's position in the embedding, *t*-SNE must be reapplied to many artificially simulated points around that data point; the non-parametric nature of *t*-SNE, combined with its significant computational cost, greatly increases computation time, which decreases the potential for interactivity. The second limitation of the method is that the user only receives a local explanation around the selected point in the embedding; she must thus explore the various regions of the embedding manually. This is not realistic in practice, especially when working with large databases, and even more so since the approach is not fast.

This paper aims to address these limitations by developing a **fast** and **globally local** explanation framework for NLDR embeddings. Based on the BIOT explanation approach [23], this framework learns sparse linear regression models for subsets of the data set and does not require a reapplication of the NLDR algorithm, making it **fast**. The **globally local** nature of our approach refers to the fact that multiple local explanations are automatically computed over the entire LD embedding (i.e., globally). Such an automatic processing enables the user to directly glimpse the overall explainability landscape of the embedding, as well as a structured overview of the impact of the HD features in the various parts of the LD embedding. The regions for which local explanations are learned in the LD embedding can be determined in different ways [24]: using a clustering algorithm such as K-means, as in this work, thanks to a manual selection performed by the user, or by recursively splitting the embedding into subcells along the LD dimensions based on a model error criterion.

Our fast and globally local explanation framework can be viewed as taking the best of both *linear* and *nonlinear* projection worlds: the LD embedding can indeed be generated by a *nonlinear* DR algorithm, achieving much better DR quality in terms of data visualization thanks to increased flexibility and adaptability [12, 15, 2]. On the other hand, the computed local explanations are *linear* and sparse, which promotes interpretability. Moreover, the globally local explanations make it possible to readily depict the importance of the HD features in the different regions of the LD embedding. As an experiment, an example use-case on a public data set is presented, highlighting the usefulness of the proposed approach. Free code and software are publicly available online (https://github.com/PierreLambert3/glocally_explained), enabling the easy use of the proposed framework.

This paper is organized as follows: Section 2 first re-

views some related works. Section 3 then presents our proposed approach, while Section 4 discusses an example use-case. Section 5 draws final conclusions.

2. Related works

Interpreting NLDR techniques is a challenging task. To tackle this challenge, various approaches have been proposed. Some papers (e.g., [25, 26, 27]) have proposed methods for explaining the LD embedding dimensions with respect to the HD features. Since local NLDR algorithms such as *t*-SNE do not effectively preserve large distances, explaining the resulting embedding dimensions with these methods may be misleading. Other methods attempt to interpret NLDR results by explaining visual clusters [22, 28, 29]. For example, in [22], the authors propose an interactive pipeline for explaining clusters in the LD embedding using decision trees; this pipeline enables the user to manually select LD clusters, which are then explained in terms of the HD features with a decision tree, an interpretable model. The resulting model can be used to explain why certain data points are clustered together and to identify the HD features that distinguish the different clusters. In contrast, our proposed approach aims to understand intra-cluster positions, i.e., the HD features that make two points from the same cluster lie at different corners of this cluster. Moreover, our framework makes it possible to not only explain LD clusters, but more generally interpret the overall positions of the points in the embedding.

Other existing methods aim to locally and linearly explain the position of a specific instance in the LD space. In particular, [21] adapts LIME [30] to locally explain *t*-SNE embeddings. The original version of LIME involves three steps. First, it samples instances around a point of interest. Then, it queries the model for these instances. Finally, it fits an interpretable model with the result of the queries. In [21], the authors use a SMOTE oversampling technique [31] to create new artificial neighbors for the point of interest. To query *t*-SNE, the entire DR process is re-applied for each sampled instance, since the *t*-SNE mapping function is unknown. Finally, BIR [32] —which is the predecessor of BIOT [23], a method employed in our work —is used to produce local explanations; BIR finds the rotation of the queried sampled data that results in the best explanation model (in terms of model sparsity and error). While the approach presented in [21] provides nice intuitions about the LD embedding structure, it has several limitations. First, it can only compute one local explanation at a time, for one selected point. Second, the obtained explanation is highly dependent on the artificial sampling. Finally, running the entire NLDR process for all sampled instances is (very) time consuming, and thus prohibits interactivity. The approach presented in this paper addresses the limitations of [21] by (1) directly providing local explanations everywhere in the LD embedding (i.e., globally local explanations), (2) avoiding the need to sample new artificial data points, and (3) relying only on the calculation of linear regression models, which ensures fast processing and hence facilitates interactivity.

3. Proposed approach

This section introduces our proposed approach for globally local and fast explanations of NLDR embeddings. Section 3.1 first summarizes our notations. Section 3.2 then details our methodology, and Section 3.3 finally presents an optional fine-tuning strategy.

3.1. Notations

Matrices are denoted with bold-faced capital letters (e.g., $\vec{X}$), vectors with bold-faced lower-case letters (e.g., $\vec{x}$) and scalars with lower-case letters (e.g., x). A single element from a matrix is denoted with a lower-case letter with two subscripts (e.g., x_{ij}), the first indicating the row and the second indicating the column. Instances are indexed by the letter $i \in \{1, ..., n\}$, features by the letter $j \in \{1, ..., d\}$, embedding dimensions by the letter $k \in \{1, ..., m\}$ and regions or subcells of the embedding by the letter $\ell \in \{1, ..., L\}$.

3.2. General methodology

In [23], the Best Interpretable Orthogonal Transformation (BIOT) method was proposed to explain the dimensions of multidimensional scaling (MDS) embeddings. In the case of *t*-SNE, such an explanation strategy is not directly applicable because *t*-SNE only preserves local structure from the high-dimensional data. However, as proposed in [21], *t*-SNE embeddings may be explained locally. Instead of learning a BIOT explanation model for the entire embedding (i.e., a single global explanation), we propose learning different BIOT models for different regions (or subcells) of the embedding (i.e. local explanation). For a given region, the BIOT model identifies the features that best explain the positioning of points within that region of the embedding, independently of all other regions. This approach can be applied to any non-linear 2-D embedding, including embeddings generated by *t*-SNE and its extensions (e.g., [33, 19]) or by other NLDR algorithms (e.g., [16, 17, 9, 34]).

Let $\vec{X}(n \times d)$ be the matrix of d features used to generate the embedding $\vec{Y}(n \times 2)$. Furthermore, let $\vec{W}(d \times 2)$ and $\vec{w}_0$ (2×1) contain the weights and intercepts for the linear models relating the features in $\vec{X}$ to each dimension of the embedding $\vec{Y}$, where there is one model per dimension.

Finally, $\vec{R}$ $(2x2)$ is an orthogonal transformation matrix that is applied to $\vec{Y}$ to promote model sparsity and prediction quality, and $\lambda > 0$ is a hyperparameter to control model sparsity. For 2-D embeddings, the BIOT objective function for global explanation is

$$J_0(\vec{W}, \vec{w}_0, \vec{R}) = \frac{1}{2n} \sum_{i=1}^{n} \sum_{k=1}^{2} (\vec{y}_i^{\top} \vec{r}_k - w_{0k} - \vec{x}_i^{\top} \vec{w}_k)^2 + \lambda \sum_{k=1}^{2} \|\vec{w}_k\|_1, \tag{1}$$

which is minimized w.r.t $\vec{W}$, $\vec{w}_0$ and $\vec{R}$ under the constraint that $\vec{R}$ is an orthogonal matrix ($\vec{R}\vec{R}^{\top} = \vec{R}^{\top}\vec{R} = \vec{I}_2$).

Clearly, this objective function can be extended to the case where different model parameters $\vec{W}^{(\ell)}$, $\vec{w}_0^{(\ell)}$ and $\vec{R}^{(\ell)}$ are optimized for different regions ℓ of the embedding, where the set of instances in region ℓ is denoted $\mathcal{S}_\ell$. In practice, the best segmentation of the embedding into regions is unknown. In this paper, we propose segmenting the embedding automatically by performing K-means on the embedding data. The choice of the hyperparameter K depends on the topology apparent in the embedding and of the granularity of details desired by the user. Other strategies are possible, for instance by recursively dividing the LD dimensions along their medians.

3.3. Fine-tuning

In Section 3.2, the proposed strategy for automatic segmentation (K-means) depends on the coordinates of the instances in the embedding. However, the shape and size of the zone that can be explained may not directly depend on the spatial coordinates of the embedding. This means that the regions identified using K-means may not be the most optimal with respect to the quality of the resulting explanations. In some cases, it is hence useful to fine-tune the final regions by directly considering explanation quality. To do so, we propose a method called Clustered BIOT, which reassigns instances i to explanation regions $\mathcal{S}_\ell$ based on a modification of BIOT. Further details on Clustered BIOT can be found in Appendix A.

4. Experiments and discussion

This section presents an example use-case for the proposed method using an interactive user interface. This user interface is available on the public repository indicated in the abstract. All of the featured embeddings are representations of the *winequality-red* dataset, available in the UCI machine learning repository [35]. This data set contains 11 physico-chemical variables describing various red wines. The embeddings are produced by a recent NE algorithm that mixes t-SNE gradients with those of a fast stochastic approximation of MDS, which preserves HD data structures across multiple scales [34].

The interface displayed in Fig. 1 shows an embedding with multiple local linear explanations: each explanation is composed of a green and a burgundy axis. Explanation Ⓐ has been selected by the user; the color transparency of the points increases linearly with the absolute difference between their position in the embedding and the position predicted by the selected linear model (i.e., the greater the error, the more transparent). This enables the user to visualize the portion of the embedding for which the selected linear model is faithful. The right panel depicts the relative importance of the HD features for each axis of the selected explanation (i.e., Ⓐ in this case), as quantified by the local linear model weights; the horizontal bar under each feature name represents the feature's signed linear projection weight (LPW) on the considered axis, highlighting the importance of the feature in the local explanation. For visual clarity, only the 5 features with the greatest LPW magnitudes are depicted for each local explanation axis. The feature *total sulfur dioxide* has been selected by the user (mark Ⓑ). When selecting a feature in the right panel, thick indicators appear on both axes of all local explanations, with lengths proportional to the LPW magnitudes of the corresponding feature on all axes; mark Ⓒ shows two such indicators. This makes it possible to grasp the influence of an HD feature in the various regions of the entire embedding.

Each view in Fig. 2 shows the importance of a particular feature in the embedding, with the respective feature indicated at the bottom of the panel. The left view highlights that free sulfur dioxide is particularly important when explaining the top portion of the embedding along a vertical direction, whereas the horizontal direction can be partly explained by the concentration of citric acid. We observe that the structures apparent in the bottom-left part of the embedding are not very dependent on the three analyzed features.

5. Conclusion

This work proposes a globally local and fast explanation framework that provides multiple local linear explanations for 2-D data embeddings, enabling the user to assess, at a glance, the importance of different HD features, both locally and across the whole LD embedding. An example use-case demonstrates that the method can effectively reveal zones in the embedding where points are organized according to specific HD features. Finally, some accompanying software is provided (https://github.com/PierreLambert3/glocally_explained), targeting both DR researchers and experts seeking to analyse their data with nonlinear dimensionality reduction visualization tools.

Further works will include testing our framework with actual end-users in the context of a real use case; their

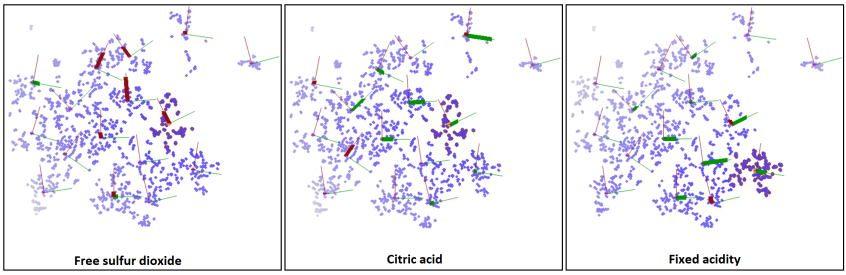


Figure 2: Importance of 3 features in the local explanations of an embedding.

feedback will enable the improvement of the various design choices of our interface. In addition, a qualitative comparison with other explainability methods such as LIME will enable a more comprehensive evaluation of the proposed method.

Acknowledgments

This work was supported by Service Public de Wallonie Recherche under grant n° 2010235-ARIAC by DIGITALWALLONIA4.AI. SC is supported by a FRIA grant (F.R.S.-FNRS).

References

[1] J. A. Lee, M. Verleysen, Nonlinear dimensionality reduction, Springer Science & Business Media, 2007.

[2] J. Venna, J. Peltonen, K. Nybo, H. Aidos, S. Kaski, Information retrieval perspective to nonlinear dimensionality reduction for data visualization., Journal of Machine Learning Research 11 (2010).

[3] I. Borg, P. J. F. Groenen, Modern Multidimensional Scaling: Theory and applications, Springer Science & Business Media, 2005.

[4] T. Kohonen, The self-organizing map, Proceedings of the IEEE 78 (1990) 1464–1480. DOI: 10.1109/5.58325.

[5] G. Hinton, S. Roweis, Stochastic neighbor embedding, in: NIPS, volume 15, 2002, pp. 833–840.

[6] I. T. Jolliffe, Principal component analysis and factor analysis, in: Principal component analysis, Springer, 1986, pp. 115–128.

[7] J. W. Sammon, A nonlinear mapping for data structure analysis 100 (1969) 401–409.

[8] J. B. Tenenbaum, V. De Silva, J. C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (2000) 2319–2323. DOI: 10.1126/science.290.5500.2319.

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

[10] J. Suykens, Data visualization and dimensionality reduction using kernel maps with a reference point, IEEE Trans. Neural Netw. 19 (2008) 1501–1517.

[11] D. Francois, V. Wertz, M. Verleysen, The concentration of fractional distances 19 (2007) 873–886.

[12] J. A. Lee, M. Verleysen, Quality assessment of dimensionality reduction: Rank-based criteria, Neurocomputing 72 (2009) 1431–1443.

[13] B. Schölkopf, A. Smola, K.-R. Müller, Nonlinear component analysis as a kernel eigenvalue problem, Neural computation 10 (1998) 1299–1319.

[14] J. A. Lee, M. Verleysen, Shift-invariant similarities circumvent distance concentration in stochastic neighbor embedding and variants, Procedia Computer Science 4 (2011) 538–547.

[15] L. Van der Maaten, G. Hinton, Visualizing data using t-sne., Journal of machine learning research 9 (2008).

[16] L. McInnes, J. Healy, J. Melville, Umap: Uniform manifold approximation and projection for dimension reduction, arXiv preprint arXiv:1802.03426 (2018).

[17] J. A. Lee, D. H. Peluffo-Ordóñez, M. Verleysen, Multi-scale similarities in stochastic neighbour embedding: Reducing dimensionality while preserving both local and global structure, Neurocomputing 169 (2015) 246–261.

[18] C. de Bodt, D. Mulders, M. Verleysen, J. A. Lee, Perplexity-free t-SNE and twice Student tt-SNE, in: ESANN, 2018, pp. 123–128.

[19] C. de Bodt, D. Mulders, M. Verleysen, J. A. Lee, Fast multiscale neighbor embedding, IEEE Transactions on Neural Networks and Learning Systems (2020).

[20] M. Wattenberg, F. Viégas, I. Johnson, How to use t-sne effectively, Distill 1 (2016) e2.
[21] A. Bibal, V. M. Vu, G. Nanfack, B. Frénay, Explaining t-sne embeddings locally by adapting lime., in: ESANN, 2020, pp. 393–398.
[22] A. Bibal, A. Clarinval, B. Dumas, B. Frénay, Ixvc: An interactive pipeline for explaining visual clusters in dimensionality reduction visualizations with decision trees, Array 11 (2021) 100080.
[23] A. Bibal, R. Marion, R. von Sachs, B. Frénay, Biot: Explaining multidimensional nonlinear mds embeddings using the best interpretable orthogonal transformation, Neurocomputing 453 (2021) 109–118.
[24] L. Pagliosa, P. Pagliosa, L. G. Nonato, Understanding attribute variability in multidimensional projections, in: 2016 29th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), IEEE, 2016, pp. 297–304.
[25] D. B. Coimbra, R. M. Martins, T. T. Neves, A. C. Telea, F. V. Paulovich, Explaining three-dimensional dimensionality reduction plots, Information Visualization 15 (2016) 154–172.
[26] M. Cavallo, Ç. Demiralp, A visual interaction framework for dimensionality reduction based data exploration, in: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, 2018, pp. 1–13.
[27] X. Yuan, D. Ren, Z. Wang, C. Guo, Dimension projection matrix/tree: Interactive subspace visual exploration and analysis of high dimensional data, IEEE Transactions on Visualization and Computer Graphics 19 (2013) 2625–2633.
[28] T. Fujiwara, O.-H. Kwon, K.-L. Ma, Supporting analysis of dimensionality reduction results with contrastive learning, IEEE transactions on visualization and computer graphics 26 (2019) 45–55.
[29] W. E. Marcilio-Jr, D. M. Eler, Explaining dimensionality reduction results using shapley values, Expert Systems with Applications 178 (2021) 115020.
[30] M. T. Ribeiro, S. Singh, C. Guestrin, " why should i trust you?" explaining the predictions of any classifier, in: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, 2016, pp. 1135–1144.
[31] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, Smote: synthetic minority oversampling technique, Journal of artificial intelligence research 16 (2002) 321–357.
[32] R. Marion, A. Bibal, B. Frénay, Bir: A method for selecting the best interpretable multidimensional scaling rotation using external variables, Neurocomputing 342 (2019) 83–96.
[33] B. Kang, D. García García, J. Lijffijt, R. Santos-Rodríguez, T. De Bie, Conditional t-sne: more informative t-sne embeddings, Machine Learning 110 (2021) 2905–2940.
[34] P. Lambert, C. de Bodt, M. Verleysen, J. A. Lee, Squadmds: A lean stochastic quartet mds improving global structure preservation in neighbor embedding like t-sne and umap, Neurocomputing 503 (2022) 17–27.
[35] D. Dua, C. Graff, UCI machine learning repository, 2017. URL: http://archive.ics.uci.edu/ml.

A. Clustered BIOT

As mentioned in Section 3.3, the main method proposed in this paper can be fine-tuned with a method we call Clustered BIOT. Let $z_{i\ell} = 1$ if instance i is in region ℓ and 0 otherwise. The matrix $\vec{Z}$ containing all elements $z_{i\ell}$ respects the general conventions of hard clustering (each instance belongs to exactly one cluster and each cluster contains at least one instance). Then, the objective function for Clustered BIOT is

$$J_1(\vec{Z}, \{\vec{W}^{(\ell)}, \vec{w}_0^{(\ell)}, \vec{R}^{(\ell)}\}_{\ell=1}^{L}) = \frac{1}{2n} \sum_{i=1}^{n} \sum_{\ell=1}^{L} z_{i\ell} \sum_{k=1}^{2} (\vec{y}_i^{\top} \vec{r}_k^{(\ell)} - w_{0k}^{(\ell)} - \vec{x}_i^{\top} \vec{w}_k^{(\ell)})^2 + \lambda \sum_{k=1}^{2} \|\vec{w}_k^{(\ell)}\|_1 \tag{2}$$

which is minimized w.r.t $\vec{Z}$ and $\{\vec{W}^{(\ell)}, \vec{w}_0^{(\ell)}, \vec{R}^{(\ell)}\}_{\ell=1}^{L}$ under the constraints that (i) $\vec{R}^{(\ell)}$ is an orthogonal matrix $\forall \ell$ and (ii) $\vec{Z}$ respects the clustering conventions above.

For fixed $\vec{Z}$, the solution for $\{\vec{W}^{(\ell)}, \vec{w}_0^{(\ell)}, \vec{R}^{(\ell)}\}_{\ell=1}^{L}$ can be found by training BIOT on each subset of instances $\mathcal{S}_\ell$, where $\mathcal{S}_\ell := \{i \mid z_{i\ell} = 1\}$. For fixed $\vec{W}^{(\ell)}$, $\vec{w}_0^{(\ell)}$ and $\vec{R}^{(\ell)}$ and a given instance i, the solution for $\vec{z}_i$ is the vector that minimizes

$$\sum_{\ell=1}^{L} z_{i\ell} \sum_{k=1}^{2} (\vec{y}_i^{\top} \vec{r}_k^{(\ell)} - w_{0k}^{(\ell)} - \vec{x}_i^{\top} \vec{w}_k^{(\ell)})^2. \tag{3}$$

Since only one element of $\vec{z}_i$ can be equal to one (instance i can belong to only one cluster), the optimal cluster for instance i is whichever model ℓ minimizes the prediction error:

$$\arg\min_{\ell} (\vec{y}_i^{\top} \vec{r}_k^{(\ell)} - w_{0k}^{(\ell)} - \vec{x}_i^{\top} \vec{w}_k^{(\ell)})^2. \tag{4}$$

Thus, Clustered BIOT can be optimized by alternating between clustering instances according to prediction error and fitting BIOT models to the clusters. An instance i is assigned to cluster ℓ if BIOT model ℓ has the lowest prediction error for that instance compared to the other models.

3.5 Enhancing optimisation with Nesterov momentum and gradient normalisation

The fast methods developed until this point are a step in the direction of allowing interactive NE visualisations of large datasets. However, as seen in Section 1.2, many NE methods rely on a 2-phase optimisation scheme where an initial *early exaggeration* period is used to help groups of points move more freely [10]. During this phase, the attractive forces are multiplied by a factor larger than 1, decreasing the chance of points getting restrained by large "barriers" of repulsion from other groups of points. If one was to re-enable early exaggeration during interactive NE visualisations, for instance following a change in a hyperparameter value, the embedding would violently re-organise. This is detrimental to visualisation, as the sudden movements throughout the embedding would prevent the user from identifying the effects of the hyperparameter change through precise movements of points.

This paper shows that one can obtain good quality embeddings without resorting to early exaggeration, by using Nesterov momentum [66] along with particularly strong momentum term as well as normalised gradients. The intuition behind this approach is that strong momentum allows points to carry enough "inertia" to overcome repulsion barriers. Note that some methods take a hierarchical approach to kick-start the optimisation instead of early exaggeration [90,91], but there the aforementioned problem of having multiple phases persists.

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

Nesterov momentum and gradient normalization to improve t-SNE convergence and neighborhood preservation, without early exaggeration

Pierre Lambert[1], John A. Lee[1,2], Edouard Couplet[1], Cyril de Bodt[1] *

1- Université catholique de Louvain - ICTEAM/ELEN
Place du Levant 3 L5.03.02, 1348 Louvain-la-Neuve - Belgium

2- Université catholique de Louvain - IREC/MIRO
Avenue Hippocrate 55 B1.54.07, 1200 Brussels - Belgium

Abstract. Student t-distributed stochastic neighbor embedding (t-SNE) finds low-dimensional data representations allowing visual exploration of data sets. t-SNE minimises a cost function with a custom two-phase gradient descent. The first phase is called early exaggeration and involves a hyper-parameter whose value can be tricky and time-consuming to set. This paper proposes another way to optimise the cost function without early exaggeration. Empirical evaluation shows that the proposed method of optimization converges faster and yields competitive results in terms of neighborhood preservation.

1 Effective visualization of data with t-SNE

High-dimensional (HD) data is abundant in our digital world, with medical images, genomics, or financial transactions, to cite only a few examples. Most of the time, the data records lie in HD spaces and dependencies or relationships cannot be visualised directly by humans, turning intuitive understanding of HD data sets into a challenging task. Dimensionality reduction (DR) tackles the problem by providing their users with low-dimensional (LD) representations of the data, enabling visual exploration.

Representing HD data into a less expressive LD space requires concessions as information is usually lost in the process. Many families of DR algorithms exist, each one focusing on the preservation of information of different nature; some methods aim to preserve distances between points [1], others called neighbour embeddings (NE) [2] attempt to preserve the local neighbourhoods around the data points. Neighbour embedding is increasingly popular due to its capability to mitigate the concentration of norms and distances [3] and the availability of fast, approximate and yet accurate implementations [4, 5], making NE applicable to very large data sets [6, 7].

Algorithms that carry out NE, such as Student t-distributed stochastic neighborhood embedding (t-SNE) [8], usually minimise a cost function in a non-parametric way, by adjusting iteratively the coordinates of the data points in the

*PL is a FRIA grantee of the Fonds de la Recherche Scientifique - FNRS. JAL is a Research Director with the F.R.S.-FNRS. EC is supported by a FSR grant (UCLouvain). CdB is supported by Service Public de Wallonie Recherche under grant n°2010235-ARIAC by DIGITALWALLONIA4.AI.

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

LD space with gradient descent. This optimization process can be interpreted as a force-directed layout or mechanical system converging to equilibrium where attractive and repulsive forces are exerted on pairs of LD points.

In practice, t-SNE and its variants come with custom implementations of gradient descent, whose course is typically split in two successive phases, namely, *early exaggeration*, a relatively short preliminary phase where the attractive forces are deliberately amplified, followed by a longer second phase, where the gradients come back to their genuine value and the LD coordinates are fine-tuned with gradient and momentum. Intuitively, amplification of attractive forces in early exaggeration contributes to momentarily shrinking the clusters and increasing the inter-cluster gaps, to compensate for the otherwise local nature of NE. Even if default settings are provided, the user can specify the length and factor of force amplification in the early exaggeration phase [8, 9, 5]. This peculiar hyper-parameter, coined *early exaggeration* after the phase it is active in, was introduced in an effort to help the clusters move freely within the embedding during the early iterations, by concentrating the sources of repulsive forces into zones of tightly-packed points [8].

From the users' broader perspective, data exploratory analysis is an interactive and possibly iterative process where the users select different subsets of points, DR methods, and hyper-parameters in order to observe the data through multiple virtual lenses. However, deciding on a value for the early exaggeration amplification factor can be quite technical and hinders the overall visualisation process by diverting attention from the data to the optimiser. This paper hence proposes a new optimiser for t-SNE, which eliminates the need for early exaggeration. It is compared to the standard t-SNE optimiser on 10 data sets. The results show that the proposed optimiser achieves competitive neighbourhood preservation and faster convergence. The experiments also support the claim that the best values for early exaggeration can vary widely across different data sets, when using the standard optimiser.

Section 2 provides an overview of the t-SNE algorithm. Section 3 introduces the proposed optimisation method. Section 4 describes the experiments, reports their results, and discuss them.

2 t-SNE with early exaggeration

This section briefly overviews t-SNE and the role of *early exaggeration* in its optimisation.

Methods of NE such as t-SNE aim to preserve the neighbourhoods around each data point; t-SNE [8] defines pairwise similarities in the HD and LD space to model the neighbourhoods as smooth functions of the locations of the data points $\mathbf{X} = [\mathbf{x}_i]_{1 \leq i \leq N}$. Considering a data set of N points in the HD space, let δ_{ij} and d_{ij} be the pairwise distance between the i^{th} and j^{th} observations in the HD and LD spaces, respectively, for $i \in \mathcal{I} = \{1, ..., N\}$ and $j \in \mathcal{I} \setminus \{i\}$. The

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

pairwise similarities σ_{ij} and s_{ij} in both space are defined as

$$\sigma'_{ij} = \frac{\exp(-\pi_i \delta_{ij}^2/2)}{\sum_{k \in \mathcal{I} \setminus \{i\}} \exp(-\pi_i \delta_{ik}^2/2)}, \sigma_{ij} = \frac{\sigma'_{ij} + \sigma'_{ji}}{2N}, \quad s_{ij} = \frac{(1 + d_{ij}^2)^{-1}}{\sum_{k \in \mathcal{I}, l \in \mathcal{I} \setminus \{k\}} (1 + d_{kl}^2)^{-1}} \ . \tag{1}$$

The similarity between a point and itself is set to 0 in both spaces. The precision π_i modulates the similarities to capture information relevant to a user-defined scale. Embeddings are produced by minimising the mismatch between the HD and LD pairwise similarities. As the similarities are normalized, they can be interpreted as probabilities of the points to be the neighbor of one another. For this reason, t-SNE defines the mismatch as the Kullback-Leibler (KL) divergence $C_{t-\mathrm{SNE}}(\mathbf{X}) = \sum_{i \in \mathcal{I}, j \in \mathcal{I} \setminus \{i\}} \sigma_{ij} \log(\sigma_{ij}/s_{ij})$. Minimisation is carried out with gradient descent.

Early exaggeration is an empirical tweak that consists in multiplying all HD similarities with a constant greater than one during the first iterations of the optimisation, typically one fifth or quarter of them. This changes temporarily the cost function and its gradient, with an increased tendency of similar HD points to group together in the LD embedding, because the similarities s_{ij} sum up to 1 while the σ_{ij} sum up to a larger value.

3 Proposed t-SNE optimizer, without early exaggeration

This section introduces the proposed optimisation method, which removes the need for early exaggeration during the first phases of optimisation.

Early exaggeration aims to help the clusters in formation move inside the embedding by accentuating the forces at hand and making the clusters smaller. Without early exaggeration, data points in movement are more likely to encounter large groups of dissimilar points and be subject to barriers or large fluctuations in the gradients. This paper proposes an alternative optimisation method without early exaggeration, where a strong momentum compensates for the occasional impeding forces, giving points enough inertia to pass the barrier of local gradient fluctuations. This work also normalises the gradients before updating the momenta.

The proposed method uses Nesterov's momentum [10], which makes (non-stochastic) gradient descent converge faster. If J denotes a cost function to be minimised, θ^t a parameter at iteration t, v^t its momentum, and $\nabla_\theta J(\theta)$ the gradient of J with respect to θ, Nesterov's momentum update the parameter θ^t following the pair of rules:

$$\begin{aligned} v^t &= \gamma \cdot v^{t-1} - \eta \nabla_\theta J(\theta^{t-1} + \gamma \cdot v^{t-1}) \ , \\ \theta^t &= \theta^{t-1} + v^t \ . \end{aligned}$$

The proposed optimiser uses a very strong momentum with $\gamma = 0.995$; the learning rate η is set to 1.

In order to keep a steady learning rate across data sets of varied sizes, the gradient $\nabla_{\mathbf{x}_i} C_{t-\mathrm{SNE}}$ for each parameter $\mathbf{x}_i$ is multiplied by $\frac{\sqrt{2N}}{100 \|\nabla_{\mathbf{X}} C_{t-\mathrm{SNE}}\|_2}$ before being used in the momentum update, as a normalization. Operator $\|\cdot\|_2$

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

denotes the L_2 norm, while $\sqrt{2N}$ is the norm of a unit vector of the same dimensionality as $\nabla_{\mathbf{X}} C_{t-\mathrm{SNE}}$, and $1/100$ is empirical. This scaling term yields better experimental results and complies with the documented recommendation of increasing the learning rate of t-SNE with the data set size [7, 6].

4 Experiments, Results, and Discussion

The classical, legacy optimizer of t-SNE and the proposed optimiser without early exaggeration are here compared. The Barnes-Hut [4] acceleration of t-SNE is used in both cases. The methods were tested on 10 data sets of size N and dimensionality M [11]: airfoil self-noise ($N = 1502$, $M = 5$), abalone ($N = 4176$, $M = 8$), COIL-20 ($N = 1440$, $M = 1024$), Statlog landsat satellite ($N = 4434$, $M = 36$), Gaussian blobs ($N = 1000$, $M = 25$), forest cover type ($N = 3000$, $M = 54$), Californian housing data set ($N = 2500$, $M = 8$), gesture phase segmentation [12] ($N = 4000$, $M = 18$), plant ($N = 1000$, $M = 4$), Anuran ($N = 4000$, $M = 22$), and single-cell RNA-seq data from the adult mouse cortex ($N = 23822$, $M = 50$). Some of the data sets are random subsamples of larger data sets; the features in the RNA-seq data set are the first 50 principal components of selected scaled gene expression levels, as in [6]. The embeddings produced with and without early exaggeration are assessed quantitatively using the area under the curve (AUC) of the relative neighbourhood preservation R_{NX} indicator [13]; the closer the AUC gets to 1, the better the neighbourhood preservation.

The standard t-SNE optimisation relies on the default hyper-parameter values as specified in [9]; the perplexity is set to 30, the embeddings are initialised with PCA, and the optimisation runs for 1000 iterations. For each data set, regular t-SNE is run 12 times in total, as a grid search for the best combination of 2 learning rates and 6 early exaggeration factors. Early exaggeration factor e_e is picked in $\{1, 4, 8, 12, 16, 20\}$, while the learning rate is either 200 or $\max(\frac{N}{e_e}, 50)\}$. The former value 200 is a common default value for the learning rate, whereas $\max(\frac{N}{e_e}, 50)$ stems from works showing that greater learning rates are preferable on large data sets [6, 7, 9].

Concerning the proposed method, it relies on the same perplexity and number of iterations as the standard optimiser. It is also initialised with PCA whitening and thus the initial embedding is scaled to have unit standard deviation along both axes. The learning rate is set to 1 and there is no early exaggeration; no grid search is necessary here and therefore one run per data set suffices.

Table 1 displays the area under the curve (AUC) of the R_{NX} curve for both optimisers on each dataset, with the best scores highlighted in bold face. For the standard method, the AUC of the best run during the hyper-parameter grid searches is shown along with the corresponding early exaggeration factor. The results indicate that the proposed method performs competitively with the legacy optimiser on the considered data sets. Furthermore, this experiment emphasizes the time-consuming challenge of tuning the early exaggeration factor, as the best values vary widely across different data sets.

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

Data set	Proposed opt.	Legacy opt.	Early exag.
blobs	0.324	0.324	4
forest	**0.569**	0.555	20
plant	**0.694**	0.677	20
Anuran	**0.561**	0.535	1
COIL20	**0.671**	0.663	1
Abalone	0.594	**0.596**	4
airfoil	0.671	**0.688**	1
satellite	**0.577**	0.560	1
housing	**0.587**	0.573	1
RNAseq	**0.484**	0.466	4

Table 1: AUC of the R_{NX} curves for both optimizers across 10 data sets; the best value of early exaggeration for the standard method is reported.

Figure 1 shows the evolution of cost function $C_{t-\mathrm{SNE}}(\mathbf{X})$across the iterations for both optimizers. The blue curves correspond to the legacy optimizer, with early exaggerations of 4 and 1 in the leftmost and rightmost plots, respectively. The orange curves correspond to the proposed method. At the 1000^{th} iteration, the KL-divergence between the HD and LD similarities is seen to be lower with the proposed method than with the legacy optimiser for both data sets. Moreover, the proposed method seems to stabilize much faster close to its final arrangement.

The end of the early exaggeration is clearly visible at 250 iterations on the blue curve of the leftmost plot. A slight change of regime can be observed on the other plot as well, despite early exaggeration is set to 1 and hence turned off. This break in the curve is due to a change in the momentum parameter carried out in the standard implementation at the same time as the early exaggeration phase ends.

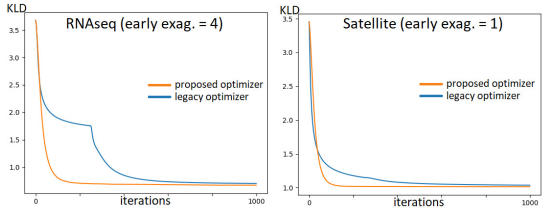


Fig. 1: Evolution of the t-SNE cost function across iterations.

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

5 Conclusions and outlook

This work proposes a new optimizer for the cost function of t-SNE, which frees the users from tediously searching an optimal early exaggeration factor. The legacy and proposed optimizers are compared, the latter producing competitive results across 10 data sets. The proposed optimisation method remains to be tested on very large data sets, which is the aim of further works.

References

[1] Jan de Leeuw. Modern multidimensional scaling: Theory and applications (second edition). *Journal of Statistical Software*, 14, 10 2005.

[2] Geoffrey E Hinton and Sam Roweis. Stochastic neighbor embedding. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems*, volume 15. MIT Press, 2002.

[3] John A. Lee and Michel Verleysen. Shift-invariant similarities circumvent distance concentration in stochastic neighbor embedding and variants. *Procedia Computer Science*, 4:538–547, 2011. Proceedings of the International Conference on Computational Science, ICCS 2011.

[4] Laurens van der Maaten. Accelerating t-sne using tree-based algorithms. *Journal of Machine Learning Research*, 15(93):3221–3245, 2014.

[5] George C. Linderman, Manas Rachh, Jeremy G. Hoskins, Stefan Steinerberger, and Yuval Kluger. Fast interpolation-based t-sne for improved visualization of single-cell rna-seq data. *Nature methods*, 16:243 – 245, 2019.

[6] D. Kobak and P. Berens. The art of using t-sne for single-cell transcriptomics. *Nat Commun 10, 5416*, 2019.

[7] Anna Belkina, Christopher Ciccolella, Rina Anno, Richard Halpert, Josef Spidlen, and Jennifer Snyder-Cappione. Automated optimized parameters for t-distributed stochastic neighbor embedding improve visualization and analysis of large datasets. *Nature Communications*, 10, 11 2019.

[8] L. van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[9] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[10] Yurii Nesterov. A method for solving the convex programming problem with convergence rate $o(1/k^2)$. *Proceedings of the USSR Academy of Sciences*, 269:543–547, 1983.

[11] Moshe Lichman. UCI Machine Learning repository, 2013.

[12] Renata C. B. Madeo, Clodoaldo A. M. Lima, and Sarajane M. Peres. Gesture unit segmentation using support vector machines: Segmenting gestures from rest positions. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, SAC '13, page 46–52, New York, NY, USA, 2013. Association for Computing Machinery.

[13] J. A. Lee, D. H. Peluffo-Ordóñez, and M. Verleysen. Multi-scale similarities in stochastic neighbour embedding: Reducing dimensionality while preserving both local and global structure. *Neurocomputing*, 169:246–261, 2015.

3.6 FUnc-SNE: a fast, accurate, and unconstrained NE method

Neighbour embedding methods that accurately model the repulsive forces exerted on points in LD, like FIt-SNE [33] and BH-t-SNE [32], manage to preserve intricate structures remarkably well [41–43]. However, in effect, these methods tend to be slower than approaches relying on coarser, sampling based approximations of the repulsive part of the NE gradients, such as UMAP [40], LargeVis [83], or NCVis [85].
This work proposes a new sampling-based NE method which is both fast and accurate in the preservation of local structures, bridging the gap between FIt-SNE and UMAP in terms of speed and quality. In short, this method efficiently tracks neighbourhoods in LD, allowing the sampling of close-by points to compute the repulsive interactions, resulting in more precise repulsions at local scales. In addition to poorly capturing the local repulsions, naive sampling-based NE tend to shift the attraction/repulsion balance of the forces applied on each point. This imbalance can be tamed by noise contrastive approaches [84, 85], where the cost function itself is built around the idea of sampling positive and negative pairs from the start. Instead of taking this route, the proposed method is built with interactivity in mind, where the user is free to tune the different hyperparameters, including one controlling the attraction/repulsion balance, allowing the extraction of different patterns with quick response times to user inputs. The method is called FUnc-SNE, for *Fast Unconstrained Stochastic Neighbour Embedding*.

As explained in Section 1.2, in order to have fully interactive methods, accelerating the iterations of the optimisation is not the only requirement: the methods also need to be able to quickly adapt to changes in data (filtering, drifts in the data, and changing the number of points) or in hyperparameters relating to the HD side of the algorithm (distance metric, neighbourhood size). The existing NE methods do not address this issue, as they all rely on a sequential two-step approach in which the HD similarities are first fully computed and then used in the optimisation phase. The first phase can be quite costly and entails a significant overhead when changes occur in data or hyperparameters, as the HD neighbourhoods and similarities need to be recomputed fully.
In addition to proposing fast yet accurate gradient computations, this work

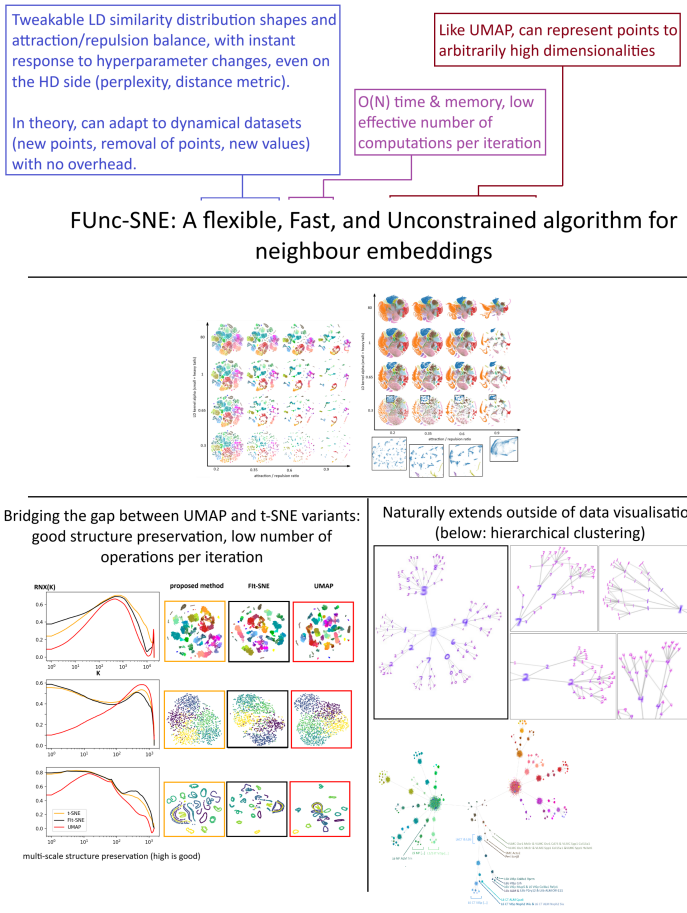
steps out of the conventional 2-step scheme to NE by refining neighbours in HD gradually during the optimisation. This approach puts to light a convenient positive feedback loop that allows the HD neighbour refinement to benefit from the improving LD configuration, while the embedding itself benefits from the refined HD neighbours. This feedback loop is in line with principles used by some modern KNN methods, which mix nearest neighbour descent with tree based partitioning methods or random projections [83]. This foundational rework in how to perform NE opens the door to fast adaptations to changes in data or hyperparameters; experience shows that modifying the distance metric or the neighbourhood size immediately reflects in the embeddings. In theory, adding new points, removing points, or modifying the HD data can also be handled without overhead, although these scenarios came as an afterthought and the current implementation does not allow the online modification of the dataset.
This work also removes the constraint that methods like FIt-SNE and BH-t-SNE impose on the LD dimensionality, allowing embeddings of arbitrary dimensionalities.

As retrospective cogitations, the paper and Section 3.7 argue that FUnc-SNE is a particularly good candidate for other uses than visualization, for instance as a data pre-processing step. In such use-cases, interactivity becomes no longer relevant and the need for an adequate configuration arises in order to get a good embedding from the first run, whatever the dataset size and distribution in HD. One could therefore argue in favour of designing another method which would tackle both shortcomings of sampling-based approaches to accelerating NE: modelling the neighbours in LD like FUnc-SNE would allow for precise local repulsions, and taking the noise contrastive route when building the cost function [84, 85] would mitigate the drift in the attraction/repulsion balance resulting from the sampling.
Another post-hoc realisation is that efficiently keeping track of both neighbourhoods in HD and in LD throughout the optimisation can also come useful in accelerating methods such as *NeRV* [90] or *Fair Neighbor Embedding* [92], whose cost functions involve divergences between neighbourhoods going both ways to control the precision/recall balance in the embedding.

Graphical Abstract

FUnc-SNE: A flexible, Fast, and Unconstrained algorithm for neighbour embeddings.

Pierre Lambert, Edouard Couplet, Michel Verleysen, John Aldo Lee



Highlights

FUnc-SNE: A flexible, Fast, and Unconstrained algorithm for neighbour embeddings.

Pierre Lambert, Edouard Couplet, Michel Verleysen, John Aldo Lee

- A novel take on **accelerating variable tailed (*t*-)SNE, embedding data points into arbitrary dimensionalities while keeping good local structure preservation**. This bridges a gab between UMAP (speed and arbitrary dimensionalities), and *t*-SNE variants (better structure preservation).
- In contrast to other methods, the algorithm in itself, but not the public implementation, **can naturally adapt to dynamical datasets** (receiving new points, deleting points, drifting points) with no computational overhead.
- A novel algorithm to iteratively refine neighbour sets, less prone to local minima than nearest-neighbour descent [1]. More importantly, the iterative KNN reveals **insights on a positive feedback loop between embedding iterations and KNN iterations**, which is exploited to the benefit of both sides.
- Insights on how NE can be repurposed to perform broader machine learning tasks.
- New insights on the effect of varying tails when modelling the LD neighbourhood distributions, extending the theoretical and experimental results introduced in papers studying the question.
- A GPU implementation with an accompanying GUI, allowing interactive exploration of datasets of hundreds of thousands of points directly.

FUnc-SNE: A flexible, Fast, and Unconstrained algorithm for neighbour embeddings.

Pierre Lambert[a], Edouard Couplet[a], Michel Verleysen[a], John Aldo Lee[a,b]

[a] *UCLouvain - ICTEAM/ELEN, Place du Levant 3, bte L5.03.02, Louvain-la-Neuve, 1348, , Belgium*
[b] *UCLouvain - IREC/MIRO, Avenue Hippocrate 55, B1.54.07, Brussels, 1200, , Belgium*

Abstract

Neighbour embeddings (NE) allow the representation of high dimensional datasets into lower dimensional spaces and are often used in data visualisation. In practice, accelerated approximations are employed to handle very large datasets. Accelerating NE is challenging, and two main directions have been explored: very coarse approximations based on negative sampling (as in UMAP) achieve high effective speed but may lack quality in the extracted structures; less coarse approximations, as used in FIt-SNE or BH-t-SNE, offer better structure preservation at the cost of speed, while also restricting the target dimensionality to 2 or 3, limiting NE to visualisation. In some variants, the precision of these costlier accelerations also enables finer-grained control on the extracted structures through dedicated hyperparameters.

This paper proposes to bridge the gab between both approaches by introducing a novel way to accelerate NE, requiring a small number of computations per iteration while maintaining good fine-grained structure preservation and flexibility through hyperparameter tuning, without limiting the dimensionality of the embedding space. The method was designed for interactive exploration of data; as such, it abandons the traditional two-phased approach of other NE methods, allowing instantaneous visual feedback when changing hyperparameters, even when these control processes happening on the high-dimensional side of the computations. Experiments using a publicly available, GPU accelerated GUI integration of the method show promising results in terms of speed, flexibility in the structures getting extracted, and show potential uses in broader machine learning contexts with minimal algorithmic modifications. Central to this algorithm is a novel approach to iterative approximate nearest neighbour search, which shows promising results compared to nearest neighbour descent.

Keywords:
Machine learning, data visualization, neighbor embedding, clustering, data pre-processing, t-SNE, UMAP

1. Introduction

Neighbour embeddings (NE) is a family of algorithms aiming to reduce the dimensionality of datasets non-linearly. For that purpose, NE define neighbourhoods around each point in the original high-dimensional (HD) space, and find lower-dimensional (LD) representations of the points such that the neighbourhoods around each data point in the reduced space are representative of those in the HD space, according to some given distance function. By focusing on the preservation of local structures in the data, NE can unfold manifolds and produce meaningful

embeddings of potentially very high-dimensional datasets. Therefore, NE is now a central tool in biology [2, 3, 4] and any domain generating HD data can benefit from NE for either visualisation or, less commonly, in algorithmic processing [5].

When used for visualisation, dimensionality reduction (DR) represents data in 2 or 3 dimensional spaces. The main purpose is then visual data exploration, which can help develop intuition on otherwise impalpable HD structures and, further, form or investigate hypotheses about data. The visual representations can also be a support to explain a result that is rooted in data.

To be of interest, datasets typically have some structure, usually making their intrinsic dimensionality smaller than the ambient dimensionality of the observed HD space, due to correlations or other functional dependencies. Nevertheless, most datasets express an intrinsic dimensionality that is still significantly larger than 2 or 3. As a consequence, DR for visualisation purposes can at best represent part of the structure in such data. The challenge of DR for visualisation is that the nature of the structures that will be preserved and represented somehow depends not only on data, but also on the specific method of DR, its assumptions, design choices, inductive biases, and hyperparameters.

Figure 1 illustrates these influences on the 2-dimensional embedding of a 2-dimensional 'S' shaped plane laying in a 3-dimensional space. In the top panel, the first row evaluates the pointwise correlation between the distances from each point to all other points in LD and HD, red indicating low quality. The second row in the panel evaluates the preservation of the first $\lceil K = 0.05N \rceil$ neighbours around each point, with N the number of points. In short, the colours of the first row can be interpreted as a measure of the preservation of large-scale structures, while the second row evaluates the soundness of apparent local structures. Principal component analysis (PCA) [6] yields a projection that preserves the global 'S' shape of the data, although the width of the 'S' is not well preserved visually, causing intrusions in the LD neighbourhoods. With t-SNE [7], belonging to the family of NE, the local structures are generally well preserved, at the price of distorting larger-scale structures. A change in the perplexity hyperparameter, which determines the neighbourhood size that t-SNE considers, can produce noticeable effects on the embedding, tearing the manifold apart in different places. In the top view, the two rightmost columns differ only in the number of points sampled in total from the S-shaped distribution; despite that, visually, the embeddings have very little in common. Unbalanced sampling can also lead to misleading embeddings, as shown in the bottom panel. There, the bottom half of the S-shaped distribution is sampled 10 times less frequently than the top half; as a result, the NE shrinks the less sampled half and even detaches both halves of the "S" with some hyperparameter configurations. These drastic differences are mostly due to the tendency of t-SNE to tear manifolds in "zones of weakness", where the local density in HD is low relative to surrounding areas. These occur naturally when then data distribution is too scarcely sampled to homogeneously capture the underlying structures in HD. As a consequence, when using NE for visualisation, a grid search over hyperparameters and methods does not systematically map to the extraction of the same predictable properties of data, and one can only use broad heuristics if not in the capacity to take into account the specificities of the data distribution and sampling process.

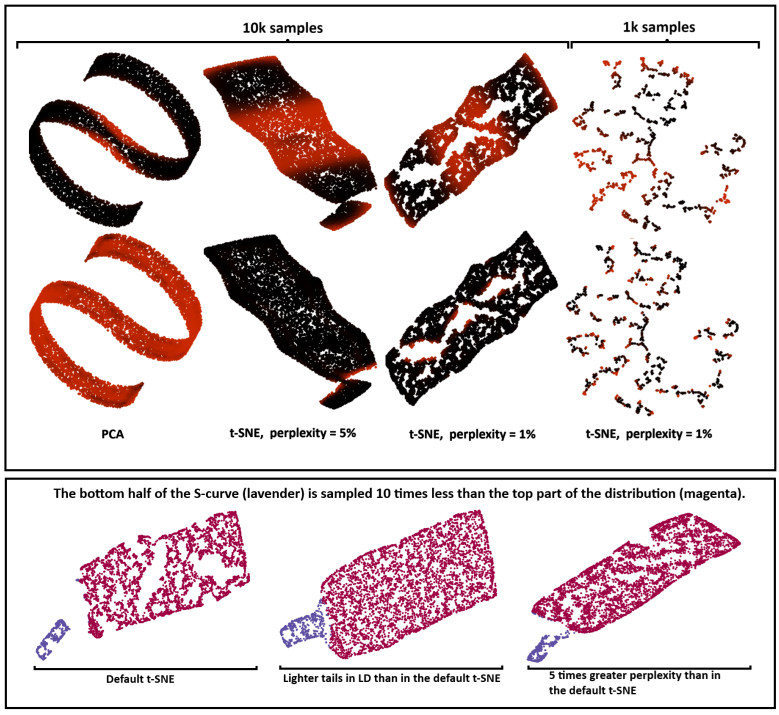


Figure 1: The impact of the DR method design choices, set of hyperparameters, and sampling of the data on the bidimensional representation of a 3-dimensional dataset. All *t*-SNE are performed with a PCA initialisation. *Top view:* the colours in the 1^{st} row represent how well the distances are preserved (global structures), the 2^{nd} row shows how well the neighbourhoods are preserved for each point (local structures); red means low quality. *Bottom view:* the colours label the top and bottom halves of the S-curve; the bottom half is less sampled than the top half.

The high intrinsic dimensionalities of data combined with the observation that embeddings exhibit high variability depending on the data-method-hyperparameters triad makes visual data exploration a delicate art. A proficient user must not interpret what is shown as the unique true structure of the data but rather as one partly correct, yet potentially misleading representation. To explore the dataset in a principled manner, across multiple LD representations, one must carefully balance confirmation bias (to filter out irrelevant information) with the capability to question one's assumptions about the data.

From these considerations, we define three high-level properties that the ideal method of NE should exhibit.

- The method should yield *correct* embeddings, in the sense that the embedding is a reason-

able solution given the objective of the method.

- The method of NE should also be *flexible*: the user must be able to adjust hyperparameters in order to explore data from different viewpoints.
- To allow for an intuitive and interactive exploration of data from those multiple viewpoints, NE must also be *fast*.

This paper proposes a novel approach to optimising a modified t-SNE objective function. The proposed method of NE satisfies the three desired properties by increasing t-SNE's speed and flexibility. The method produces embeddings of quality competitive with t-SNE and UMAP[8], while keeping $O(N)$ time and memory complexities and a reasonable effective number of computations per iteration. Building on the intrinsic speed and parallelisability of the proposed method, its algorithmic implementation is coded to run on GPUs and it is available in a public repository (`https://github.com/PierreLambert3/fast_htSNE`), reaching speeds compatible with interactivity for datasets of hundreds of thousands of points. The implementation allows for instantaneous visual feedback on hyperparameter changes, even if the hyperparameter controls properties on the HD side of the method, such as the perplexity or distance metric. Any such change in regular algorithms of NE would typically trigger a new, generally long, complete pre-computation of neighbour affinities before actually resuming the embedding optimisation. In addition to speed and correctness, the proposed method offers greater flexibility than the standard t-SNE or UMAP algorithms, as it allows for LD similarity distributions with tails of varying weight, following the works in [9, 10]. Experiments show that this particular property can extract structure and patterns in data that would otherwise remain hidden. The method allows for embedding data points into arbitrarily large dimensionalities, allowing its use outside of the strict visualisation context. This property was until then lacking in accelerated variants of t-SNE, a subgenre of NE that outperform other methods like UMAP in local structure preservation, as observed in this paper's empirical assessments and in [11, 12]. On a side note, the proposed method of NE required a novel iterative technique to refine neighbour sets. This procedure can be used in other applications and brings insights on the possible use of directed embeddings to perform fast iterative approximate neighbourhood searches in large and potentially changing datasets.

The rest of this paper is organised as follows. Section 2 presents a detailed summary of the current state of NE for visualisation, from their advent to key technical insights under the hood of t-SNE, which is the specific method of NE this paper builds upon. The same section also overviews the current state of the most used methods of NE, mentioning the principal directions of research and comparing them. Section 3 describes the proposed method of NE, starting from t-SNE and its close variant h-t-SNE [10]. Section 4 evaluates the proposed method across multiple criteria and datasets, comparing the method to others of the same breed. The novel iterative process to find K-nearest neighbours is also evaluated and compared to nearest-neighbour descent [1], showing promising performances.

2. The state of neighbour embeddings

The first half of the section reviews the history of NE, motivates its adoption, and describes t-SNE, one of the most popular variant of NE to date from which this work iterates.

The second half of this section introduces two major directions in which research and methods of NE are aimed at: speed and flexibility. As its most prominent challenger in popularity,

UMAP is compared to *t*-SNE with regards to the three introduced desirable properties (correctness of the LD embeddings, speed, and flexibility).

2.1. From linear projections to neighbour embeddings

Until 2002, the use of DR for visualisation was mainly limited to PCA and multidimensional scaling (MDS) [13], these methods are based on measures of variance for PCA or pairwise distances for MDS. PCA is a linear method which severely restricts is feature extracting power. When data dimensionality grows, the pairwise distances display a ratio of their mean over variance that increases too, a phenomenon referred to as the concentration of norms and distance [14]. Intuitively, norm concentration implies that points in HD datasets tend to be nearly equidistant. As a consequence, embedding HD data in LD spaces by explicitly preserving distances is as vain as squaring a circle: LD spaces are not expressive enough to accommodate the kind of distributions of distances that live in HD spaces. In the context of visualisation in 2D or 3D, norm concentration thus restricts the use of MDS to datasets with reasonably low intrinsic dimensionalities and low noise in the ambient dimensionalities.

In 2002, Hinton et al. introduced a new paradigm for dimensionality reduction with their work on Stochastic Neighbour Embedding (SNE) [15]. Their method aims to produce visual embeddings that are both resilient to norm concentration and able to capture fine-grained structures in the data by defining its loss function around the principle of neighbourhood preservation. In SNE, the authors define a differentiable function to model neighbourhoods around each point in HD and LD as a normalised Gaussian function of their distance. The distance is defined to be Euclidean in the LD space, but it can be measured otherwise in the HD space. The embedding is produced by first initialising the points randomly or in an informed manner, then, the embedding is iteratively updated by minimising the Kullback-Leibler divergence between the fixed HD Gaussian neighbourhoods and their changing LD counterparts. In fact, SNE revealed structures in HD data that had remained unseen until then. For instance, the method visually showed for the first time a clear aggregation of numbers of the same class in the MNIST dataset, without supervision [15].

The results of SNE were groundbreaking at the time of publication, although it later became obvious that modelling neighbourhoods with Gaussian functions in the LD space led to 'crowding' problems, with overly cluttered visualizations [7]. To address this issue, *t*-SNE models the LD neighbourhoods with a normalised hyperbolic function that relates to Student *t* distributions with a single degree of freedom. This counter-intuitive mismatch between the profiles of the HD and LD neighbourhood functions induces a distance stretch and therefore broadens the gap between clusters of points in the embedding, improving cluster visualisation but also introducing an inductive bias that fragments manifolds like dry soil. Such a tearing apart can happen on different scales, depending on the data sampling and hyperparameters, as showcased in Figure 1.

Figure 2 shows embeddings of the single-cell RNA sequencing of cells from a mouse neocortex [2], a data set that is further studied in [16]. Grey-brown colours show non-neuron cells, hot colours indicate inhibitory neurons, colder colours represent excitatory neurons. The PCA projection reveals an aspect of the global structure, with non neural cells located far from neurons, and the neurons themselves clearly distinguished between excitatory and inhibitory types. The MDS embedding adds further details to the structures shown by PCA. In contrast, *t*-SNE and UMAP discard the largest-scale structures, as a price to pay for revealing a finer-grained hierarchy of clusters that could not be highlighted by the use of raw distances.

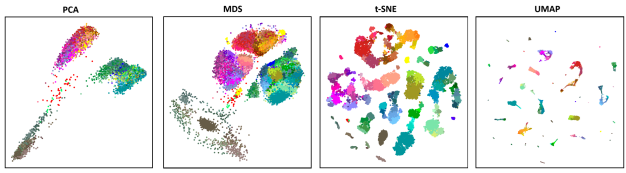


Figure 2: Visual comparison of embeddings of a single-cell transcriptomics dataset studying cells in mice brains. The global organisation in the t-SNE embedding is vestigial from an informed PCA initialisation: random initialisations shuffle the inter-cluster organisation. Neighbour embeddings are frequently initialised non-randomly [16].

2.2. *Student t-distributed stochastic neighbour embedding: t-SNE*

Formally, let $\mathbf{X} = [\mathbf{x}_i]_{1 \leq i \leq N}$ denote the HD coordinates of the data and $\mathbf{Y} = [\mathbf{y}_i]_{1 \leq i \leq N}$ be their corresponding LD representations. Building on the fixed distances δ_{ij} measured between $\mathbf{x}_i$ and $\mathbf{x}_j$ in HD, t-SNE models the soft, probabilistic neighbourhoods in HD by the symmetrised pairwise normalised similarities p_{ij}, defined as

$$p_{j|i} = \frac{\exp(-\delta_{ij}^2/2\sigma_i^2)}{\sum_{k \neq i} \exp(-\delta_{ik}^2/2\sigma_i^2)}, \quad p_{i|i} = 0, \quad \text{and } p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N}, \tag{1}$$

where the individual, point-specific radii σ_i are adjusted during initialisation to comply with a user-set perplexity, this allows the HD Gaussian model of neighbours around each point to scale to the local density of the HD sample.

In the LD space, the corresponding soft probabilistic neighbourhoods are modelled through normalised similarities q_{ij}, defined as

$$q_{ij} = \frac{(1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1}}{\sum_{k \neq l}(1 + \|\mathbf{y}_k - \mathbf{y}_l\|^2)^{-1}}, \quad q_{ii} = 0 \ . \tag{2}$$

To determine the optimal embedding $\mathbf{Y}$, t-SNE uses gradient descent to minimise the Kullback-Leibler divergence $\mathrm{KL}(P\|Q) = \sum_{i \neq j} p_{ij} \log(p_{ij}/q_{ij})$ between the two neighbourhood distribution. The gradient for the LD coordinates gets written down as

$$\frac{\partial L}{\partial \mathbf{y}_i} = 4 \sum_{j \neq i} (p_{ij} - q_{ij})(1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1}(\mathbf{y}_i - \mathbf{y}_j) \ . \tag{3}$$

At each iteration, the contribution of the j^{th} point to the representation $\mathbf{y}_i$ of the i^{th} point can be understood as a force aligned on $\mathbf{y}_j - \mathbf{y}_i$, which is either attractive, if the points are more similar in HD than they currently are in LD, or repulsive otherwise.

Most other methods of NE have their own peculiarities to model the neighbourhoods in both spaces or to define the cost function. However, they all share the common principle of constructing a representation of the dataset where embedded points are subject to pairwise attractive forces and repulsive forces, driven by the difference between sparse local affinities computed in HD, and affinities computed in LD.

2.3. The evolution of neighbour embeddings

The time complexity of standard t-SNE is quadratic in the size of data ($O(N^2)$), restricting its use to datasets of a few thousands of points at most. This is problematic in some fields like the study of single-cell data in biology, where datasets can grow to hundreds of thousands or millions of points. The following paragraphs bring a quick overview of the efforts that were made to reduce the time complexity of NE.

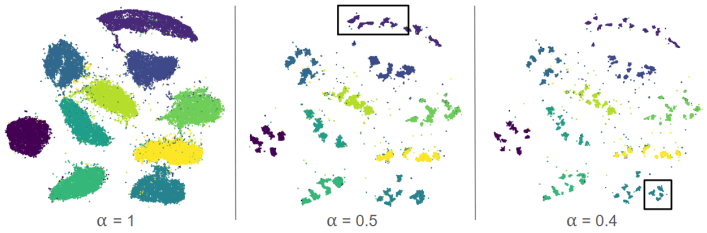
Algorithmically, NE naturally decomposes into two subsequent steps: (i) all preliminary computations involving the given, fixed data, like δ_{ij}, σ_i, and p_{ij}, and (ii) the actual optimization of $\mathbf{Y}$ and related variable quantities like q_{ij}. Computing the HD similarities p_{ij} can be particularly slow, as it involves $O(N^2)$ distances in the HD space. However, using them exclusively within Gaussian functions to characterise local neighbourhoods induces sparsity that allows accelerated methods to drastically reduce the number of considered pairwise interactions in HD, by limiting these to neighbours only. The neighbour sets in HD can be computed in various ways. Barnes-Hut t-SNE [17] uses VP trees, whereas FIt-SNE [18] uses multiple randomised K-D trees to quickly coalesce accurate neighbour sets. Uniform manifold approximation and projection (UMAP) [8] uses nearest-neighbour descent [1]: an iterative refinement of randomly initialised neighbourhoods, where new neighbour candidates are generated from neighbours of neighbours, until convergence. LargeVis [19] uses a combination of nearest-neighbour descent and random projections to determine the neighbour sets in HD. While these algorithms greatly accelerate the first phase of NE, computing the HD neighbours still takes a noticeable fraction of the total runtime of NE, especially with datasets that are both large and very highly dimensional.

Accelerating the second algorithmic phase of NE, involving gradient descent, is not as straightforward as for the preliminary computation of HD similarities. In the LD embedding space, the similarities are generally defined using heavier tailed distributions than in HD. As a consequence, acceleration cannot rely on sparsity and short-range interactions restricted to only a few neighbours like in the HD space. Instead, fast methods of NE usually accelerate the gradient descent iterations with (locally adaptive) undersampling, like Barnes-Hut trees [17] or negative sampling [8, 19].

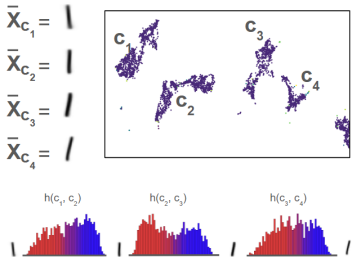
Negative sampling, used in UMAP and LargeVis, reduces the computational cost of estimating the ambient field of repulsive forces by ignoring the majority of pairwise repulsive interactions. At each iteration, for each point, the global landscape of the field in LD that the point 'perceives' is coarsely approximated by probing just a handful of randomly selected other points. The small resulting force is rescaled to mimic $N - 1$ interactions. Methods of NE with negative sampling have iterations of $O(N)$ time complexity, and the simplicity of the approach grants them generally higher effective speeds than other methods. However, by considering that the volume of a n-ball of radius r increases with r^n, the probability of sampling a distant point is much higher than that of sampling a similar point. As a result, the local and mid-range components of the repulsive field are poorly under-estimated with negative sampling, allowing points to easily intrude, that is, to trespass unauthorized neighbourhoods. As a consequence, many seemingly close neighbours in the LD space are not actual neighbours in the HD space. This downside is clearly apparent when evaluating the quality of embeddings with quantitative criteria such as in [11], in [12], and as will be witnessed again later in Figure 6.

Instead of approximating the repulsive forces with unbiased random samplings, other accelerated methods, like Barnes-Hut t-SNE and FIt-SNE, model the occupancy of the LD space more faithfully by aggregating repulsive forces in a locally adaptive way. Intuitively, the idea is that if a tight cluster of points lies away enough in a certain direction, all nearly aligned repulsion vectors can be summarized into a single large vector, hardly losing any information in the process. Compared to negative sampling, modelling the LD space occupancy to approximate the pairwise interactions allows for more precise gradients, in particular those regarding local repulsive interactions, as supported by Figure 6 and in [11]. Barnes-Hut t-SNE and FIt-SNE achieve time complexities of $O(N \log N)$ and $O(N)$, respectively, which are both significant improvements over the original $O(N^2)$, but their effective speed is generally noticeably lower than that of methods using negative sampling, due to the burden of better modelling the LD space occupancy at each iteration. Another hard limitation of such methods is that they scale exponentially with the target dimensionality, restricting their use to visualisation only.

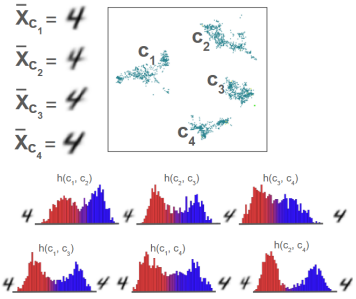
Beyond accelerating NE, researchers have also attempted to enhance the flexibility of methods, such that they can yield embeddings that reflect different points of view on the data, controlled by hyperparameters. In particular, researchers have investigated the effect of modifying the profile of the LD similarity distribution with respect to distance, shifting the dynamics of optimisation to reveal structures with different granularities [10, 20, 21]. The authors show that LD similarities with heavier tails tend to produce finer-grained, more fragmented clusters, as illustrated in Figure 3. In this figure, the handwritten MNIST dataset is represented in 3 embeddings, with varying tail heaviness in the LD similarities. Lower α values correspond to heavier tails, $\alpha = 1$ is equivalent to (an accelerated) t-SNE. Looking at the 3 whole embeddings of Figure 3a, the large digit-specific clusters apparent using a t distribution (leftmost embedding) get fragmented when increasing the 'heavytailedness' of the LD similarity distribution. The views 3b and 3c highlight how the dynamics using heavier tails result in finer clusters. The notation $\bar{X}_c$ designates the mean image (in HD) of the cluster c (determined in LD). In this figure, the histogram $h(c_x, c_y)$ represents the distribution of points from the set $c_x \cup c_y$ along the HD axis $(\bar{X}_{c_x} - \bar{X}_{c_y})$, after centring the set on either $\bar{X}_{c_x}$ or $\bar{X}_{c_y}$. The red bar colour represents c_x, and blue for c_y.



(a) Representations of the MNIST dataset by the proposed method, with increasingly heavy tails on the LD similarities, from left to right.



(b) Subset of the "1" digits, as displayed using moderately heavier tails than t-SNE.



(c) Subset of the "4" digits, as displayed using very heavy tails.

Figure 3: Effect of heavier tails in the LD similarities (small α) on the visual separation of groups of points, on the MNIST handwritten digits dataset. Embeddings produced by the proposed method.

Looking at a subset of the points with label '1' in Figure 3b, one sees that this part of the embedding appears to be organised by the tilt angle of the '1', along the horizontal direction in the embedding. Assuming a homogeneous sampling and uniformly distributed tilt angles, one would expect these points to lie on a continuous one-dimensional manifold. The separation into multiple sub-clusters when going from $\alpha = 1$ (t-SNE) to $\alpha = 0.5$ apparent in Figure 3b can therefore be surprising, and one can wonder whether this subdivision is truly driven by data or whether it is an unrelated artifact of the optimisation dynamics. The histograms reveal that the clusters get separated in LD around a dip in the probability distribution along certain directions in HD, which shows that the sub-cluster separation bears a meaning. At this point, one can only hypothesise to explain the dip in the histograms along these directions: does it stem from some unfortunate sampling? or is it a shared bias in the handwriting of humans? or perhaps the digitalisation by the camera misrepresents some angles?
Similar observations can be made by looking at Figure 3c, where a subset of the digit '4' gets fragmented into 4 clusters when going from $\alpha = 0.5$ to $\alpha = 0.4$. Looking at the 6 possible pairs of clusters, one can reasonably conclude that, when using heavier tails, this method of NE has found directions of weakness, along which a dip in the distribution of points initiated the fragmentation of a cluster from HD to LD.

Exploring the landscape of embeddings produced by different LD similarity distribution shapes can greatly enhance the visualisation process, in particular when integrated to an interactive visualisation software responding visually and in real time to changes in hyperparameters or data. Such an integration allows the user to intuitively grasp the links between different conformations through the movement of the points, effectively adding another accessible dimension to the usual static embeddings. Interactivity requires methods to be very fast to (re)start and to run. Unfortunately, having tweakable LD similarity distribution shapes is not particularly adapted to the fastest methods relying only on negative sampling to estimate the effect of repulsive field on each point. This comes from the fact that the tailedness of the LD similarities has a significant role in the delicate attraction/repulsion balance across all scales [22], including the local ones, resulting in the fine grained structures seen in (Figure 3).

To conclude with this quick overview of NE, methods like FI*t*-SNE can precisely model the repulsive forces applied to each point, at each iteration. This accuracy makes such methods modifiable to broaden the spectrum of extracted structures. However, their accurate way to evaluate repulsive interactions in the LD space decreases their speed, reduces their interactivity, and limits the target dimensionality to 2 or 3. Other methods like UMAP run faster indeed, at the cost of embedding quality and flexibility. The next section introduces a method that bridges this gap by improving the estimation of the local repulsive forces compared to using negative sampling only, allowing greater flexibility, while still running extremely fast (smooth interactivity-enabling speeds at hundreds of thousands of points on a GPU-bearing laptop, given reasonable HD dimensionalities in the order tens or low hundreds); without constraints on the target dimensionality to 2 or 3, a property explored later in the results.

3. A Fast, UNconstrained SNE: FUnc-SNE

Our proposal builds upon a recent generalised method of NE [10]. The authors extend the definition of NE, with a more flexible shape for the LD similarities, which gets an additional hyperparameter $\alpha \in \mathbb{R}_{>0}$ that determines how heavy the tails are in the distribution. The LD pairwise similarities q_{ij} are defined as

$$q_{ij} = \frac{w_{ij}}{\sum_{k \neq l} w_{kl}} \text{ with } q_{ii} = 0 \text{ and } w_{ij} = \frac{1}{\left(1 + \frac{\|\mathbf{y}_i - \mathbf{y}_j\|^2}{\alpha}\right)^{\alpha}} \ . \tag{4}$$

The gradient then becomes

$$\frac{\partial L}{\partial \mathbf{y}_i} = 4 \sum_j (p_{ij} - q_{ij}) w_{ij}^{1/\alpha} (\mathbf{y}_i - \mathbf{y}_j) \ . \tag{5}$$

When α takes values between 0 and 1, the tails of the LD similarity distribution are heavier than that of a Student t-distribution; $\alpha = 1$ is equivalent to t-SNE; when increasing alpha towards infinity, the tails decrease asymptotically.

In order to accelerate the algorithm without constraining the LD space to 2 or 3 dimensions while keeping a good approximation of the local repulsive forces, this paper proposes to cut the gradient definition for a point i into three terms, explicitly distinguishing neighbours in HD, neighbours in LD but not in HD, and the rest of the points. Let the twin sets of HD and LD neighbour indices for the i^{th} point be denoted $\mathfrak{N}_i^{\text{HD}}$ and $\mathfrak{N}_i^{\text{LD}}$, respectively, and $\frac{\partial L}{\partial \mathbf{y}_{ij}}$ be the gradient

on $\mathbf{y}_i$ resulting from interacting with the point number j. The cost function gradient can be re-written as

$$\frac{\partial L}{\partial \mathbf{y}_i} = \sum_{j \in \mathfrak{N}_i^{\text{HD}}} \frac{\partial L}{\partial \mathbf{y}_{i|j}} + \sum_{\substack{j \in \mathfrak{N}_i^{\text{LD}} \\ \& \; j \notin \mathfrak{N}_i^{\text{HD}}}} \frac{\partial L}{\partial \mathbf{y}_{i|j}} + \sum_{\substack{j \notin \mathfrak{N}_i^{\text{LD}} \\ \& \; j \in \mathfrak{N}_i^{\text{HD}}}} \frac{\partial L}{\partial \mathbf{y}_{i|j}} \quad . \tag{6}$$

Notice that the union of the these sets of indices encompasses all pairwise interactions for i, and their intersection is empty. In sufficiently large datasets, the majority of the interactions correspond to the third term of the sum, where point number i interacts with points that are neither neighbours in HD, nor in LD. This part is what is approximated by negative sampling in methods like UMAP and LargeVis. The proposed method also performs negative sampling to estimate this portion of the gradient.

The first term of the sum corresponds to indices j that are neighbours of i in HD. This sum is the sole portion of the cost function responsible for the attractive pairwise interactions, therefore, all NE methods consider this sum somehow in their loss function. When HD neighbours get too close in the embedding, negative forces between them can be generated either as a direct consequence of the loss function, or in some methods as an additional aesthetic layer of interactions. However, commonly, a large part of the neighbours in LD are not neighbours in the HD set, which leads to the second term of the sum, a novelty of the proposed method.

The second term of the sum concerns pairwise interactions with points that are neighbours of i in LD, but not in HD. This allows the proposed method to more precisely model the local LD structures around each point than methods using only (unstructured) negative sampling like UMAP. While the proposed method does not precisely capture the whole range of scales at which pairwise interactions occur, as FIt-SNE or BH t-SNE would, experiments show that this simple addition can bring satisfactory results with a reduced cost, assuming an efficient way to approximate neighbours in LD. This simplification also allows the proposed method to embed points into arbitrarily high dimensional spaces as UMAP can, because modelling the whole occupancy of the LD space is no longer required. Table 1 places the proposed approach between the two traditional directions for accelerating NE.

	Close range	Medium range	Far away
Negative sampling only	poor	poor	correct
Modelling the whole space	correct	correct	correct
FUnc-SNE (proposed method)	correct	poor	correct

Table 1: Comparison of the quality of approximation of the different components of the repulsive field around each point, for negative sampling based methods (like UMAP), methods that model the whole LD space occupancy (like FIt-SNE), and the proposed approach. The distinction between the different ranges is subjective and generally not explicitly formalised in the NE methods; the aim of this table is to bring a high-level overview.

Compared to negative sampling based approaches, the proposed formulation adds repulsions between neighbours in LD, to better represent local structures. Compared to approaches that model the LD space occupancy directly, this simple approach shifts the problem of quickly and accurately modelling the LD space, into that of quickly and accurately modelling the neighbourhoods in LD.

Let $\hat{\mathfrak{N}}_i^{\mathrm{HD}}$ and $\hat{\mathfrak{N}}_i^{\mathrm{LD}}$ denote the approximate sets of neighbours around point i in HD and LD, respectively. As a reminder, all other methods of NE proceed with two phases: (i) determining the exact or approximate HD KNN sets for all i, and (ii) using gradient descent to optimise the loss function. The proposed approach bypasses the first phase by interweaving neighbourhood discovery and gradient descent (GD), admitting that the first GD steps will yield very noisy estimates.

Algorithm 1: refining a KNN set for a point i

```
bool refine_KNN(i, X, KNN_here, KNN_other, K):
    candidate ← i
    if rand(0.0, 1.0) < 0.05 then
        candidate ← randint(0, N)                          // exploration
    else
        lookIn ← KNN_here                                  // exploitation
        if rand(0.0, 1.0) < 0.5 then
            lookIn ← KNN_other                 // exploitation (other space)
        for d ← 0 to randint(2, 4) do
            candidate ← lookIn[candidate][randint(0, K)]
    // Insertion is a success if the candidate is not already a
       neighbour, is not i itself, and is closer to i than i's
       farthest neighbour.
    success ← perhaps_insert_candidate(i, X, KNN_here, K, candidate)
    return success
```

Taking inspiration from nearest-neighbour descent [1], the method iteratively looks at neighbours of neighbours to generate candidates for the sets $\hat{\mathfrak{N}}^{\mathrm{HD}}$ and $\hat{\mathfrak{N}}^{\mathrm{LD}}$. A peculiarity of this algorithm is that both estimated neighbours sets communicate during candidate generation: a candidate neighbour destined for $\hat{\mathfrak{N}}_i^{\mathrm{HD}}$ can be generated from neighbours of neighbours in LD or in HD, as shown in Algorithm 1. Considering that the GD steps occur at the same time, the method benefits from a convenient positive feedback loop: the better the embedding (through the gradients), the better the KNN estimation for both spaces (better candidates); closing the loop: the better the KNN sets, the better the gradients, and the better the embedding.
This approach shares nearest-neighbour descent's linear complexity with respect to the number of points per iteration. However, where nearest-neighbour descent performs a complete search of neighbours of neighbours at each iteration, the proposed approach only samples a few candidates, spreading the search across more iterations.

The effect of the feedback loop can be witnessed in Figure 4, where the quality of $\hat{\mathfrak{N}}_i^{\mathrm{HD}}$ is evaluated across iterations, using a fixed embedding (no feedback) and an embedding progressively updated through gradient descent. The quality measure is the area under the curve $R_{\mathrm{NX}}(K)$ [23] (AUC of $R_{\mathrm{NX}}(K)$) for the first 256 neighbours ($1 \leq K \leq 256$); the estimate $\hat{\mathfrak{N}}_i^{HD}$ is compared to the exact $\mathfrak{N}_i^{\mathrm{HD}}$ computed for the purpose of this validation experiment. In short, $R_{\mathrm{NX}}(K)$ measures the intersection between two sets of neighbours across all scales K, subtracting the baseline associated with the expected rate of fortuitous neighbour matches and rescaling accordingly between 0 (not better than purely random) and 1 (perfect retrieval). The AUC of the $R_{\mathrm{NX}}(K)$ gives a multi-scale overview of the quality of the neighbour set. These plots show that using the NE gradients in the LD embedding increases the speed at which the neighbour sets in HD are re-

fined, despite not changing the KNN-finding portion of the method. Interestingly, increasing the dimensionality of the LD embedding from 2 to 8 seems to increase the positive effect of the feedback loop. One explanation is that the more voluminous space allows points to organise more freely, allowing for more faithful structures in LD, which in turn helps generating better candidates for neighbour refinements in HD. A more detailed evaluation of the KNN discovery scheme is presented in the next section.

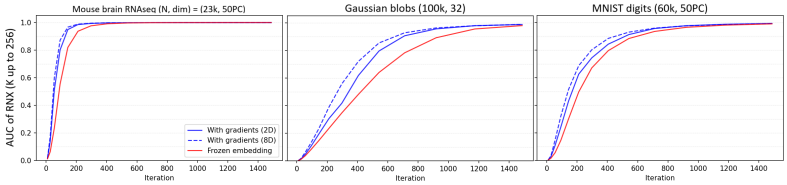


Figure 4: Evolution of the quality of the estimated KNN sets in HD throughout time, with a fixed embedding (red) or embeddings subject to optimisation (blue).

During optimisation, flags are used to keep track of points that discovered new neighbours in HD. Periodically, the flagged points have their associated adaptive bandwidth in HD (σ_i) updated using a warm restart from their previous value, for efficiency. The associated HD similarities are then updated accordingly. For datasets of reasonable size, for instance with less than 300×10^3 points, if running the algorithm on a modern laptop with a GPU, the quality of each KNN set in $\hat{\mathfrak{N}}^{\text{HD}}$ quickly approaches the maximal value. To prevent superfluous computations and favour the GD part of the algorithm once the sets in $\hat{\mathfrak{N}}^{\text{HD}}$ are close to those in $\mathfrak{N}^{HD}$, the refinement of $\hat{\mathfrak{N}}^{\text{HD}}$ can be skipped. The probability of performing the refinement of $\hat{\mathfrak{N}}^{\text{HD}}$ is arbitrarily set to $0.02 + 0.98\text{E}[\frac{\text{N}_{\text{new}}}{\text{N}}]$, where $\text{E}[\frac{\text{N}_{\text{new}}}{\text{N}}]$ is the smoothed fraction of points that have successfully received new neighbours in the recent past. Since the embedding is subject to a GD step at each iteration, $\hat{\mathfrak{N}}^{\text{LD}}$ is systematically refined at each iteration. Refining the LD KNN sets is less costly than in HD, due to the lower dimensionality and because changes in $\hat{\mathfrak{N}}^{\text{LD}}$ do not trigger the same cascade of adjustments as in HD. Naturally, iteratively updating the LD neighbours requires the learning rate to be reasonably small, to allow the sets to keep up with the changes in the LD coordinates. In practice, however, large learning rates do not seem to have a noticeable negative effect on the method, perhaps because allowing the use of HD neighbours to generate candidates provides a stable source of good candidates for LD neighbours that are unaffected by changes in the embedding. Algorithm 2 brings a summary of the procedure.

Algorithm 2: The FUnc-SNE loop

procedure `FUnc-SNE`$(N, K_{max}, PP_{init}, metric_{init}, X_{LD}, X_{HD})$**:**

- $\text{KNN}_{HD} \leftarrow$ randomIndices(N, K) `// KNN: distinct indices`
- $\text{KNN}_{LD} \leftarrow$ randomIndices(N, K) `// indexed as KNN__[i][k]`
- $\sigma \leftarrow$ onesFloat(N) `// allocate each radius in HD`
- $\text{P} \leftarrow$ onesFloat(N, K) `// Symmetrised HD similarities`
- $\text{F} \leftarrow$ onesBool(N) `// flags indicating an update in KNN`$_{HD}[i]$
- $\text{E}_{\text{HD}} \leftarrow 1$
- $\text{t} \leftarrow 0$
- estimated_denominator $\leftarrow 1$ `// for the LD similarities`
- **while** *not done* **do**
  - $\text{t} \leftarrow \text{t} + 1$
  - `// 1. radii and similarities in HD, if KNN`$_{HD}$ `updated`
  - **if** $(t \ \% \ 10) = 0$ **then**
    - `// Only points with an active flag are involved`
    - $\text{E}_{\text{HD}} \leftarrow 0.95 \times \text{E}_{\text{HD}} + 0.05 \times \text{sum(F)/N}$
    - **adjust_radii**$(\text{F}, \sigma, \text{X}_{\text{HD}}, \text{KNN}_{HD})$
    - **adjust_P**$(\text{F}, \sigma, \text{X}_{\text{HD}}, \text{KNN}_{HD}, \text{P})$
    - *memset*$(\text{F}, 0)$ `// clear all flags`
  - `// 2. refining neighbourhoods (not shown here: 32 candidates are generated per point per iteration instead of 1)`
  - **if** $rand(0.0, 1.0) < 0.02 + 0.98 \times E_{HD}$ **then**
    - **for** $i \leftarrow 0$ **to** $N - 1$ **do**
      - $\text{F}[i] \leftarrow$ **refine_KNN**$(i, \text{X}_{\text{HD}}, \text{KNN}_{HD}, \text{KNN}_{LD}, K)$ `// HD`
  - **for** $i \leftarrow 0$ **to** $N - 1$ **do**
    - **refine_KNN**$(i, \text{X}_{\text{LD}}, \text{KNN}_{LD}, \text{KNN}_{HD}, K)$ `// LD`
  - `// 3. one GD step: for a point i, interactions are computed on KNN`$_{HD}[i]$`, KNN`$_{LD}[i]$`, and a handful of random samples. The estimated_denominator variable is also updated globally from the sampled LD similarities.`
  - **GD_step**$(\text{X}_{\text{LD}}, \text{P}, \text{KNN}_{HD}, \text{KNN}_{LD}, \text{estimated_denominator})$

In addition to benefitting from the aforementioned positive feedback loop, refining the KNN sets concurrently with the GD steps makes the algorithm natively adaptable to online learning, where points can be added or removed on the fly, without disturbing the flow of iterations. However, the current implementation is not designed around this use case. In the same vein, another positive aspect of abandoning the traditional two-phase approach is that the algorithm is particularly well suited to integration in interactive visualisation software. Optimisation begins immediately after the initial memory allocations, giving the user immediate visual feedback. In practice, meaningful structures start appearing soon after launch and well before the estimated HD neighbours sets converge. Similarly, the proposed scheme allows the embedding to adapt seamlessly to hyperparameter changes in the HD side of the NE, such as the distance metric or perplexity.

In practice, having heavy tails in the LD kernels ($\alpha \ll 1.0$) tends to produce numerous visual clusters with very high intra-cluster point densities, hampering visualisation. For this reason, the proposed method keeps the attractive forces separated from the repulsive ones by distributing p_{ij} and q_{ij} in the gradient formula ($\frac{\partial L}{\partial \mathbf{y}_i} = 4 \sum_j \left[p_{ij} w_{ij}^{1/\alpha} (\mathbf{y}_i - \mathbf{y}_j) \; - \; q_{ij} w_{ij}^{1/\alpha} (\mathbf{y}_i - \mathbf{y}_j) \right]$); the two vectors are aggregated at the time of applying the gradients using a scaling factor as hyperparameter. This allows the user to counteract the visual collapse of clusters when using heavy kernel tails in LD by increasing the repulsive forces on the fly. The proposed method can also optionally increase the attractive forces at the start of the optimisation to facilitate the movements of points, as commonly done with early exaggeration [24] in other methods. The hyperparameter can also come handy when dealing with very large datasets, where enhancing the attractive forces is often recommended, for less crowded visualisations [16].

Since the HD KNN discovery is more effective when the embedding reflects the HD data, the algorithm can benefit from a bit of help in the very first iterations to jump start the HD KNN discovery. To do so, if the algorithm is used in an interactive GUI, playing with the hyperparameters in the early iterations tends to change the embedding sufficiently to allow a good initial exploration in the HD KNN finding routine. Outside of the interactive setting, the same effect can be obtained by performing a couple of random linear projections of the data towards the LD space every 10 or 20 iterations during the first 100 or 200 iterations instead of using the NE gradients. While not strictly necessary, this can help reduce the number of iterations required to get good estimated HD neighbour sets. This is similar in philosophy to how LargeVis [19] combines nearest neighbour descent with random projections to determine neighbours in HD.

A final trick worth mentioning is the recommended addition of an 'implosion' button in visual software interfaces for the method. The gradient formula shows that the bigger the scale of the embedding in LD, the smaller the relative gradient norms. This can be problematic if visualising the data in real time through a continuous optimisation, because some combinations of attraction / repulsion ratios and LD pairwise similarity distribution shapes can lead to a seemingly ever-expanding embedding. The software provided with the proposed method features an 'implosion' button where the whole embedding coordinates are scaled down by a large factor, allowing the gradients to be significant again.

The method scales linearly with the number N of points, and the dimensionality M of the data. The linear increase in time with M is particularly noticeable in the implementation accompanying this paper, because the distance computations are not parallelised, contrary to the rest of the algorithm. The time to compute the M-sized sum involved in most distances might be lessened by the use of parallel reduction in future implementations. When using the method for interactive data visualisation, it is recommended to first reduce the HD dimensionality of the data linearly to a manageable number of dimensions, depending on the machine and dataset size, in order to keep a high number of iterations per seconds for a smooth experience. In NE, it is quite common to see PCA being used for that, and keeping 50 to 100 components before running the NE itself [16]. When using NE for visualisation, the LD dimensionality is so low that reducing the HD dimensionality to 50 generally does not have a noticeable impact on the visual results. Exceptions to the rule exist, when datasets already have very little redundancy extractable by linear methods, which is often the case when exploring latent spaces of deep neural networks. It should be pointed out that the abrupt break in the quality of the repulsive field when passing from an accurate approximation for close relationships (through the approximate LD neighbour-

hoods) to a coarse approximation in further ranges (through negative sampling) can sometimes make the embeddings reach equilibria that are visually different from those of other methods like BH-t-SNE and FIt-SNE for identical sets of hyperparameters. In practice, changing the repulsion/attraction ratio can lead to configurations that strongly resemble those obtained by other variants of t-SNE like UMAP, in agreement with published results [22].

The next section evaluates the embedding method and the KNN finding subroutine empirically, both qualitatively and quantitatively.

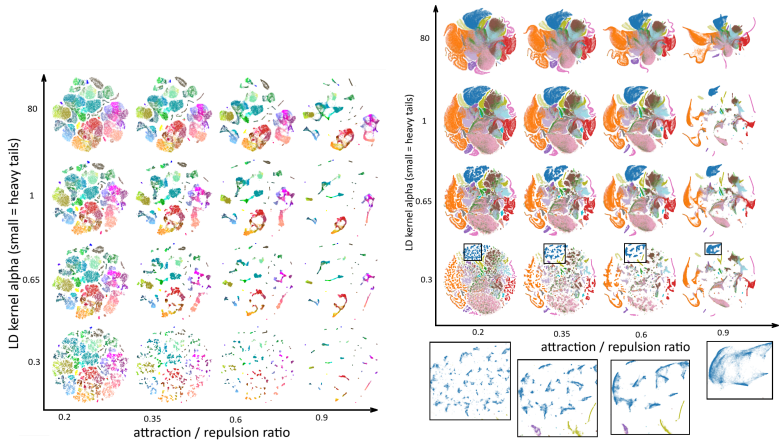
4. Empirical evaluation

Subsection 1 evaluates the method in its capability to produce embeddings for the purpose of visualisation both qualitatively and qualitatively. Its running time is also assessed. Subsection 2 evaluates the capacity of the algorithm to model data at dimensionalities higher than 2 or 3.

4.1. The case of data visualisation

In the following paragraphs, the method is evaluated in the context of data visualisation by considering 2-dimensional embeddings. Two hyperparameters are central to the method: the attraction/repulsion ratio of the forces exerted on the data points, and the thickness of the tails in the LD kernels. Figure 5 shows the effect of these two hyperparameters visually on embeddings of single-cell datasets.

The left panel corresponds to 'mouse brain' [2], where hundreds of messenger RNA sequences are measured in 23 thousand cells found in the brains of rats. The colours were tailored by biologists to reflect the nature of the cells at hand, hot colours represent inhibitory neurons, cold colours correspond to excitatory neurons, and grey-ish colours are other types of cells present in the brain. In the right panel figures the 'Tabula Muris Senis droplet' dataset [25]. This version of the dataset is comprised of a about 245 thousand cells found in various tissues in mouse bodies, the points are coloured by tissue type.
In both datasets, increasing the thickness of the tails in the LD kernels (decreasing α) fragments the data more and more, as observed previously on MNIST in Figure 3. When using the method for visualisation, heavy tails can produce LD clusters that are very dense, hampering visualisation. The attraction/repulsion ratio can be adjusted to prevent this collapse of points, and keep readable embeddings at low α values.
In unsupervised scenarios, the notion of cluster is ill-defined and subject to interpretation. This has been hinted in Figure 3, where looking at a dataset through different configurations of the proposed method can produce clusters of different granularity. If the proposed method is used in a continual optimisation scenario, where the user can modify the hyperparameter values on the fly and observe the resulting motion of points in real-time, the user can explore a broad band of the spectre of possible clusters by scanning through values of α and adjusting the attraction/repulsion occasionally. Capturing a hierarchy of different clustering solutions through multiple granularities is explored more thoroughly in the next subsection.



(a) RNA-seq.: mouse brain

(b) RNA-seq.: Tabula Muris Senis droplet

Figure 5: Visual evaluation of the effect of the attraction/repulsion ratio and LD kernel tail thickness in embeddings produced using the proposed method.

The embeddings produced in Figure 5 appear sound when comparing them to other representations and explorations of the datasets [2, 16, 25], but a more quantitative evaluation of the method is necessary to compare it to other modern NE algorithms. Figure 6 compares the proposed method to 2 staples of NE for visualisation: FI*t*-SNE [18] and UMAP [8]. These were chosen for their identical time complexity ($O(N)$) and their wide adoption in practical NE. The 3 algorithms are compared across multiple scales on 3 datasets using the $R_{\text{NX}}(K)$ [23] quality metric. Values close to 1 show good neighbourhood preservation at the given scale K, values close to 0 indicates preservation equivalent to a random placement of the points. The mouse brain dataset is figured in the top row, the middle row shows generated Gaussian blobs, and the 3$^{\text{rd}}$ row is COIL-20. The COIL-20 dataset consists of images of 20 objects rotating around an axis, drawing a ring like manifold in HD for each object. Default hyperparameters were used for FI*t*-SNE and UMAP, for the proposed method, the values for α and the attraction/repulsion ratio were chosen manually to produce embeddings that were visually similar to those obtained with FI*t*-SNE.

The quality curves show that the proposed method can preserve neighbourhoods across multiple scales similarly to FI*t*-SNE and UMAP. The systematic poor preservation of local structures in UMAP, consistent with the observations in [11, 12] may be due to its reliance on unbiased negative sampling and on HD neighbours to determine the pairwise repulsive forces exerted on each point, allowing intrusive points in LD to remain undetected.

The proposed algorithm introduces a twist to nearest-neighbour descent [1] by performing an iterative refinement of neighbours alongside the gradient descent steps producing the embeddings. When refining their neighbours, the two representations of the data communicate through their estimated KNN sets when producing candidates. Figure 7 compares the proposed KNN finding algorithm to nearest-neighbour descent on 4 datasets. The Y axis is the $R_{\text{NX}}(K)$ criterion,

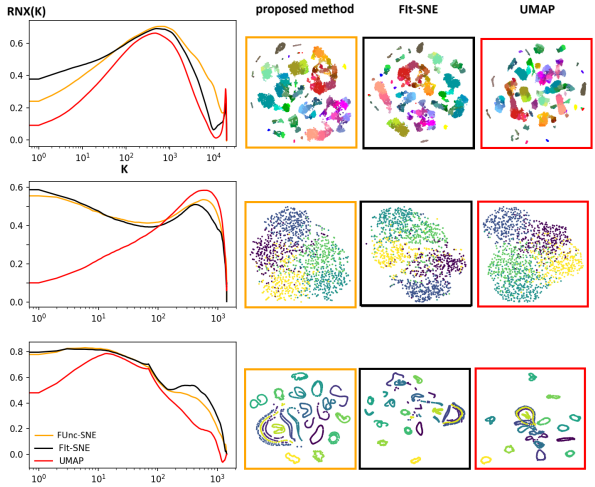


Figure 6: $R_{\mathrm{NX}}(K)$ curves of the proposed method, UMAP, and FIt-SNE on 3 datasets. From top to bottom: RNA-seq: mouse brain, Gaussian blobs (default scikit-learn implementation, 16-dimensional), COIL-20.

which indicates the overlap between the estimated KNN sets and the ground truth exact sets: the closer to 1, the closer the estimated sets are to the exact sets at scale K. The translucent bands indicate the standard deviation of $R_{\mathrm{NX}}(K)$ across the points: large bands indicate a high heterogeneity in the estimated KNN set quality across points. The sets that are evaluated here are the sets in the HD space.
Nearest-neighbour descent is a greedy refinement of neighbours which converges to a final solution, the proposed method requires an explicit termination criterion. For this experiment, the KNN refinement stops at the 3000^{th} iteration for the blue lines, and 9000^{th} iteration for the orange ones; the green lines are the $R_{\mathrm{NX}}(K)$ curves for nearest-neighbour descent.
The two 'Blobs' datasets are engineered to show two different scenarios: in the 'Overlapping' case, the data is generated from 5 Gaussian blobs with high overlap: the nearest neighbour descent manages to find good neighbours in its greedy approach. In the 'Disjointed' case, there are 1000 centres of 30 points each, with much smaller standard deviations. In the disjointed case, the isolation of the clusters in the HD space is a difficulty for nearest neighbour descent, whereas the proposed method managed to leap out of the local minimum and find better neighbour sets. Interestingly, the proposed method necessitated more time to reach close-to-perfect neighbour sets in the supposedly easier Overlapping case then in the Disjointed case. The figure shows that the proposed nearest neighbour search can produce estimated sets that are competitive with those found by nearest-neighbour descent, given enough iterations.

As stated earlier, NE need to be correct, flexible, and fast. The previous figures show that the proposed method can produce competitive results and allows for flexibility in what structures to

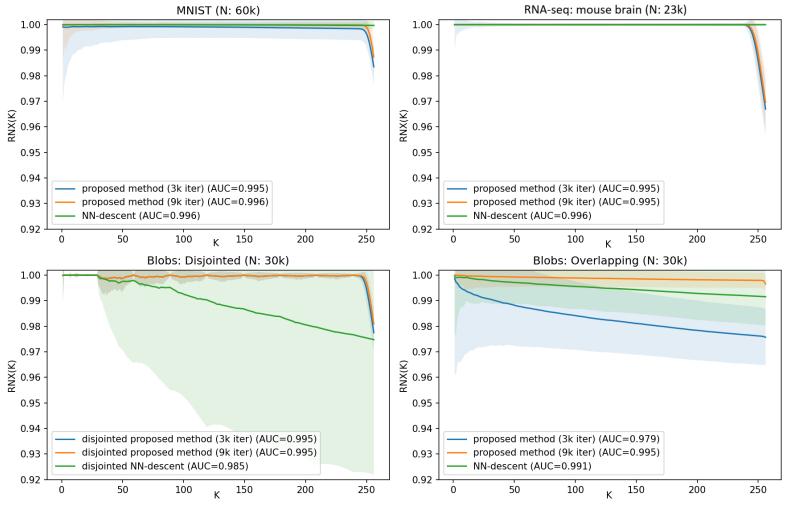


Figure 7: Comparison of the proposed neighbour finding algorithm with nearest neighbour descent. The x axis corresponds to the neighbourhood scale K, the y axis measures the quality of the estimated neighbourhood (higher is better).

extract, by tweaking hyperparameters. A measure of the speed of the algorithm across different dataset sizes using a modern laptop is given in Figure 8, the datasets in HD are generated by a uniform distribution in 8 dimensions.

The black line shows the time taken by FUnc-SNE with a perplexity set to 80; the purple line evaluates the time taken by UMAP as implemented by NVIDIA RAPIDS, with *n-neighbours*, the hyperparameter analogous to perplexity, set to 80 as well. Both ran for 1000 iterations. The time taken by the nearest neighbour descent part of UMAP is shown by the dashed pink line.
The linear nature of the proposed method with respect to the number of points is manifest, and the intrinsic speed of the UMAP algorithm aided by skilled GPU development is notable. However, the time taken by the HD neighbour search in UMAP (and by extension in the other NE methods since nearest-neighbour descent is used for its speed) is an important proportion of the total time taken: this motivates the property of FUnc-SNE to respond seamlessly to changes in HD by merging iterations and HD neighbour search, benefitting from the feedback loop whose effect was witnessed in Figure 4.
While the time complexity of the FUnc-SNE algorithm is linear with the HD and the LD dimensionalities, the current GPU implementation does not parallelise the dimensionalities, therefore, the impact of these on the runtime is noticeable in practice. For the purpose of visualisation, it is recommended to reduce first the HD dataset dimensionality to manageable values by a linear projection, a standard practice in NE [16]. A more skilful GPU implementation would likely reduce runtimes considerably, as GPU programming is particularly sensitive to coding prowess.

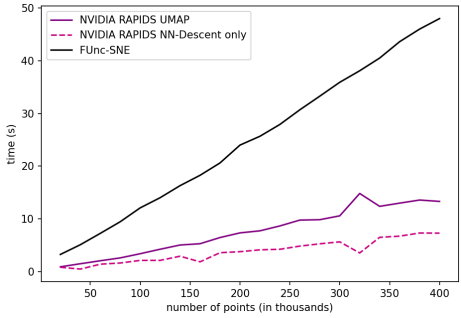


Figure 8: Effective time taken by the proposed method (in-house GPU implementation), UMAP (from *NVIDIA RAPIDS*), and nearest-neighbour descent (from *NVIDIA RAPIDS*) on a *NVIDIA GeForce RTX 4070* laptop GPU. The embedding methods ran for 1000 iterations.

4.2. Exploring intermediate dimensionalities

The proposed method is capable of producing embeddings of larger dimensionalities than 2 or 3, and can in theory adapt to new data points with little overhead. These two properties open the method to uses outside of the strict context of data visualisation. This section evaluates the method's capacity to extract relevant structures when representing data in more than 3 dimensions through two use cases. In the first case, the algorithm is extended to allow a hierarchical representation of datasets, with computations take place in low but larger than 3 dimensions. The representations are evaluated qualitatively on two datasets. In the second scenario, the proposed method is used as a step to pre-process data in the difficult top-1 ImageNet classification task in a one-shot learning setting. Accuracy scores and visual inspection indicate that algorithm can produce embeddings that are well generalisable for a downstream supervised task.

In NE, the mismatch in the HD and LD kernel shapes can facilitate visualisation by exaggerating the separation between clusters in the LD space. The effect is visible in Figure 5 and 3, where increasing the weight of kernel tails in LD increases the number of clusters that appear visually in the LD space. When using the method in an interactive environment, the user can get a sense of the hierarchical organisation of the data by observing how the clusters fragment and fusion depending on the changes in the hyperparameter α. While useful as-is, the restrictive nature of very low dimensionalities can interfere with the optimisation and lead to results of heterogeneous quality, in particular if areas of the dataset exhibit high intrinsic dimensionalities. Here, an algorithm to capture the hierarchy through a graph is proposed, where the process of fragmenting clusters by lowering α is done in less restrictive dimensionalities.

Let us consider an embedding subject to a continual optimisation, where the LD kernel tails are slowly getting heavier in time. Snapshots of the changing embedding are taken at given intervals, the saved embeddings can be written $\mathcal{X}^\ell$, the level ℓ indexes the advancement in the growing kernel tail weight. A clustering algorithm is performed on each embedding $\mathcal{X}^\ell$ to extract the clusters $C^{(\ell)}$. A graph is then built to capture how the clusters evolved through time: each

cluster is considered as a node, and the edges e_{ij} between $C_i^{(g)}$ and $C_j^{(h)}$ are defined:

$$e_{ij} = \beta \cdot \frac{|\, C_i^{(g)} \cap C_j^{(h)} \,|}{\min(|C_i^{(g)}|,\ |C_j^{(h)}|)}, \quad \beta = \begin{cases} 1 & \text{if } |h-g| = 1, \\ 0 & \text{else.} \end{cases}$$

Clustering is carried out here with DBSCAN [26], chosen for its speed and ability to adapt to different number of clusters. In practice, the reasonably low embedding spaces and the tendency of NE to broaden gaps seem to facilitate the work of DBSCAN in this experiment. Once the graph built, it can be used algorithmically or visualised using a force-directed layout, as in Figures 9 and 10. In both figures, a central node representing the whole dataset is added for aesthetic reasons, and the size of the nodes is proportional to $\sqrt{|C_i^\ell|}$, with bounds. It should be mentioned that this algorithm has a lot in common in philosophy with a method introduced in [27].

This introductory method mixes algorithms of different nature (NE, clustering, graph drawing), giving a straightforward initial approach. However, given the similarities that can be drawn between graph drawing and NE [28], perhaps one could devise another hierarchical approach which computes soft similarities between points at different levels of kernel tail heaviness, and represent these to LD directly without explicit clustering or graph drawing. Removing explicit clustering would likely produce more fine-grained details in the hierarchy.

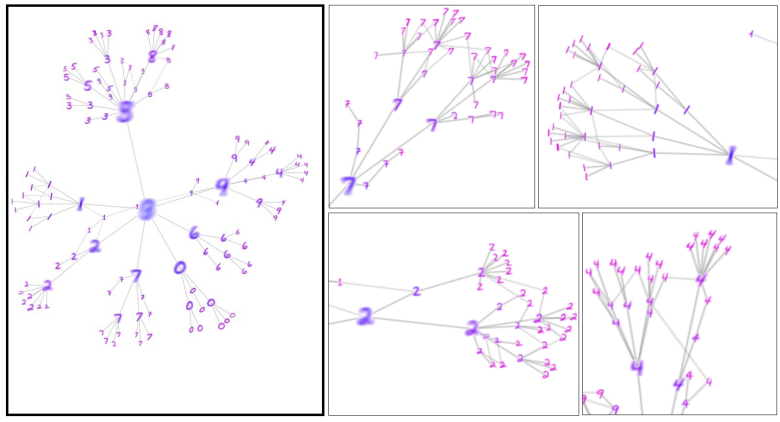


Figure 9: Hierarchical representation of MNIST, with LD dimensionality of 4.

In Figure 9, the nodes are represented by their mean image in HD. The view on the left represent the whole dataset using three levels, the views on the right show close ups if performing the same procedure with an additional level of LD kernel heaviness, revealing finer-grained details. A connoisseur in the art of writing down numbers by hand might identify a meaningful organisation in these representations. For instance, in the view on the left, the 5, 3, and 8 digits

speciate a level of depth later than the rest, and indeed they differ only by small segments. In the same view, the rare images of 1 that exhibit a bottom horizontal line are represented here as a mixture between 1 and 2, and indeed 2 is the only other digit that can have a straight horizontal line at the bottom. The finer grained organisation shown on the right also reveals subjectively meaningful structure upon scrutiny.

The hierarchical structure of the mouse brain dataset [29] is also extracted, the results are shown in Figure 10. In this figure, instead of showing the depiction of a prototype for each node, the points belonging to each cluster are rendered as Gaussian noise using the colours that were hand-picked by the authors of [29]. A hierarchy of clusters is apparent and bears a strong resemblance with the ground-truth dendrogram accompanying the dataset in original paper, which indicates the relationship between the cells as currently established in biology.

What constitutes a hierarchy in a dataset is traditionally a property captured by “zooming in” or by subsampling a dataset, either automatically or through human control [30, 31, 32, 33]. This experiment shows that precise preservation of local structure in intermediate, permissive, dimensionalities with varying kernel tails can offer a different taste of what a hierarchy is in data. Rather than applying uniform zoom, one can observe how the data fragments recursively under the stress of the mismatch in the distribution of the neighbours defined by NE.
It is difficult to define what a cluster is, in practice the definition is often tailored to a task or interpreted by a human. These experiments show that the capacity to control with which severity a NE will tear manifolds in their LD representations can help in the extraction of meaningful clusters across selected scales. The NE performs what one might describe as a tweakable "pre clustering" of the data, facilitating and directing the work of a following conventional clustering algorithm.



Figure 10: Hierarchical representation of the mouse brain dataset, with LD dimensionality of 6.

To further evaluate the soundness of higher dimensional NE of data, the rest of the section will consider a 32-dimensional NE of the ImageNet dataset obtained using the proposed method on the first 192 principal components of the 1280-dimensional latent representation of the images, as extracted by the vision encoder of the large vision-language model EVA [34] (images $\rightarrow$ 1280-dim. latent space $\rightarrow$ 192 PCs $\rightarrow$ 32-dim. NE). The first 192 principal components capture 71.1% of the variance of the 1280-dimensional dataset. The preliminary linear projection towards 192 dimensions is performed to reduce the runtime of the NE, which scales linearly with the area of the data matrix. Furthermore, the current implementation on GPU has a hardcoded limit on the dimensionality of the input data, this limit was set to facilitate development while keeping reasonably optimised memory access patterns and GPU kernel launch overheads. The ImageNet dataset is comprised of more than 1.2M natural images of 1000 distinct classes. The number of classes makes it a particularly challenging benchmark, especially in weakly supervised scenarios.

In order to evaluate the embedding, a simple 1-nearest neighbour classifier using Euclidean distances is trained in different learning contexts and on different representations of ImageNet, the

accuracies are presented in Table 2. In the one-shot leaning scenario, the mean accuracy of 100 trials is shown; a trial consist of revealing the label of a random point of each class, and classifying all the other points.

	1280, EVA	192, PCA	32, PCA	32, NE
one-shot (top-1)	47.3%	45.9%	20.6%	**76.2%**
one-shot (top-5)	67.3%	67.8%	39.4%	**92.0%**
crossval (top-1, train)	84.9%	83.2%	71.6%	**87.2%**
crossval (top-1, test)	87.2%	87.2%	71.8%	**88.7%**

Table 2: Comparison of classification performance in 3 representations of the 1.2M images of ImageNet, in transductive 1 shot learning and 10-fold cross-validation.

The performance in one shot learning and the tighter margin between train and test accuracies in cross-validation indicate that, despite being obtained in a unsupervised manner, the 32-dimensional NE exposes information in a way that facilitates generalisation regarding this supervised task. The embedding is publicly available at *https://github.com/PierreLambert3/Imagenet-tSNE-embedding.git*.

To shed some light on how the NE organises data in the 32-dimensional space, Figure 11 shows a 2-dimensional PCA projection of the embedding, along with a PCA projection of the initial EVA representation. The linear nature of PCA allows an incomplete but undistorted view of the structures. The NE seems to represent data in tighter, less diffuse groups of points than the original representation. An artifact common to PCA and to spectral clustering is apparent [22, 35], where only a few clusters are distinct and appearing like long spikes, while the others collapse towards the centre of the linear projection.

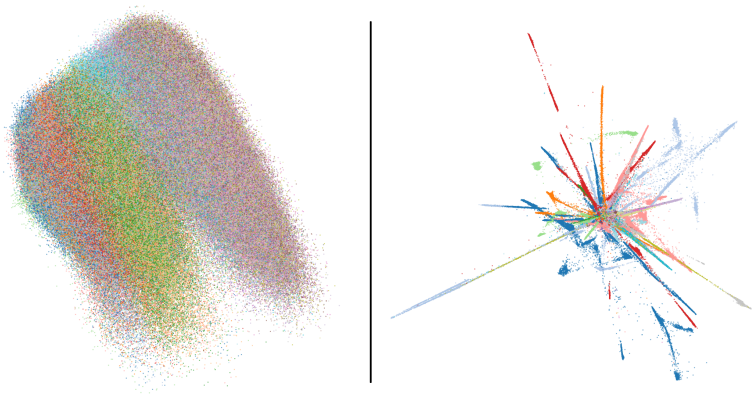


Figure 11: Two-dimensional PCA projections of the 1280-dimensional representation of ImageNet by EVA. Left: raw EVA features (1280 (EVA) → 2 (PCA vis.)); right: EVA features after embedding in 32D with the proposed method (1280 (EVA) → 192 (PCA) → 32 (t-SNE) → 2 (PCA vis.)).

5. Conclusion

This paper shows that good quality in neighbour embedding can be attained quickly and without upper limit on the dimensionality of the LD space, by combining the tracking of neighbours in the LD space with negative sampling. The proposed method relies on a novel extension of nearest neighbour descent, where the sets of neighbour around each point get refined iteratively concurrently with the embedding; this process benefits from the embedding in construction to refine neighbour sets faster, as a source of neighbour candidates that are likely to be eligible by design and often confirmed.

The method could supposedly be applicable to dynamical contexts with streams of data that are not known in advance, or data points with dynamical values. The virtually non-existent overhead of adding, removing, or modifying the data in the HD side results from the fact that HD similarities and neighbourhoods are estimated alongside the embedding, with a heuristic allowing to divert compute to the HD side or the embedding dynamically, depending on the need. This contrasts with other methods that typically run through several distinct and different algorithmic stages, like data preparation (e.g., conversion of the HD coordinates into a sparse neighbourhood graph, pre-computation of entropic affinities, etc.) before actually launching the NE iterations. This could broaden the applicability of NE to visualisation in continual learning, but also to contexts outside of strict visualisation for instance to pre-process data before some other computational tasks, like clustering or classification.

6. Acknowledgements

P.L. is a FRIA grantee of the F.R.S.-FNRS. E.C. is supported by a FSR grant (UCLouvain). JAL is a Research Director with the F.R.S.-FNRS.

References

[1] W. Dong, C. Moses, K. Li, Efficient k-nearest neighbor graph construction for generic similarity measures, in: Proceedings of the 20th International Conference on World Wide Web, WWW '11, Association for Computing Machinery, New York, NY, USA, 2011, p. 577–586. doi:10.1145/1963405.1963487.

[2] B. Tasic, Z. Yao, L. T. Graybuck, K. A. Smith, T. N. Nguyen, D. Bertagnolli, J. Goldy, E. Garren, M. N. Economo, S. Viswanathan, et al., Shared and distinct transcriptomic cell types across neocortical areas, Nature 563 (7729) (2018) 72–78.

[3] E.-A. Amir, K. Davis, M. Tadmor, E. Simonds, J. Levine, S. Bendall, D. Shenfeld, S. Krishnaswamy, G. Nolan, D. Pe'er, Visne enables visualization of high dimensional single-cell data and reveals phenotypic heterogeneity of leukemia, Nature biotechnology 31 (05 2013). doi:10.1038/nbt.2594.

[4] W. Yang, C. Sun, R. Huszár, T. Hainmueller, K. Kiselev, G. Buzsáki, Selection of experience for memory by hippocampal sharp wave ripples, Science 383 (6690) (2024) 1478–1483. arXiv:https://www.science.org/doi/pdf/10.1126/science.adk8261, doi:10.1126/science.adk8261.

[5] T.-T. Yu, C.-Y. Chen, T.-H. Wu, Y.-C. Chang, Application of high-dimensional uniform manifold approximation and projection (umap) to cluster existing landfills on the basis of geographical and environmental features, Science of The Total Environment 904 (2023) 167013. doi:https://doi.org/10.1016/j.scitotenv.2023.167013.

[6] K. P. F.R.S., Liii. on lines and planes of closest fit to systems of points in space, The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science 2 (11) (1901) 559–572. doi:10.1080/14786440109462720.

[7] L. van der Maaten, G. Hinton, Visualizing data using t-sne, Journal of Machine Learning Research 9 (86) (2008) 2579–2605.

[8] L. McInnes, J. Healy, J. Melville, Umap: Uniform manifold approximation and projection for dimension reduction (2020). arXiv:1802.03426.

[9] L. van der Maaten, Learning a parametric embedding by preserving local structure, in: D. van Dyk, M. Welling (Eds.), Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics, Vol. 5 of Proceedings of Machine Learning Research, PMLR, Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA, 2009, pp. 384–391.

[10] D. Kobak, G. Linderman, S. Steinerberger, Y. Kluger, P. Berens, Heavy-tailed kernels reveal a finer cluster structure in t-sne visualisations, in: U. Brefeld, E. Fromont, A. Hotho, A. Knobbe, M. Maathuis, C. Robardet (Eds.), Machine Learning and Knowledge Discovery in Databases, Springer International Publishing, Cham, 2020, pp. 124–139.

[11] P. Lambert, C. de Bodt, M. Verleysen, J. A. Lee, Squadmds: A lean stochastic quartet mds improving global structure preservation in neighbor embedding like t-sne and umap, Neurocomputing 503 (2022) 17–27. doi:https://doi.org/10.1016/j.neucom.2022.06.108.

[12] C. de Bodt, A. Diaz-Papkovich, M. Bleher, K. Bunte, C. Coupette, S. Damrich, E. F. Sanmartin, F. A. Hamprecht, E. Ágnes Horvát, D. Kohli, S. Krishnaswamy, J. A. Lee, B. P. F. Lelieveldt, L. McInnes, I. T. Nabney, M. Noichl, P. G. Poličar, B. Rieck, G. Wolf, G. Mishne, D. Kobak, Low-dimensional embeddings of high-dimensional data (2025). arXiv:2508.15929.

[13] J. de Leeuw, Modern multidimensional scaling: Theory and applications (second edition), Journal of Statistical Software 14 (10 2005). doi:10.18637/jss.v014.b04.

[14] D. Francois, V. Wertz, M. Verleysen, The concentration of fractional distances, IEEE Trans. Knowl. Data Eng. 19 (7) (2007) 873–886.

[15] G. E. Hinton, S. Roweis, Stochastic neighbor embedding, in: S. Becker, S. Thrun, K. Obermayer (Eds.), Advances in Neural Information Processing Systems, Vol. 15, MIT Press, 2002.

[16] D. Kobak, P. Berens, The art of using t-sne for single-cell transcriptomics., Nat Commun 10, 5416 (2019).

[17] L. van der Maaten, Accelerating t-sne using tree-based algorithms, Journal of Machine Learning Research 15 (93) (2014) 3221–3245.

[18] G. C. Linderman, M. Rachh, J. G. Hoskins, S. Steinerberger, Y. Kluger, Fast interpolation-based t-sne for improved visualization of single-cell rna-seq data, Nature methods 16 (2019) 243 – 245.

[19] J. Tang, J. Liu, M. Zhang, Q. Mei, Visualizing large-scale and high-dimensional data, in: Proceedings of the 25th International Conference on World Wide Web, WWW '16, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 2016, p. 287–297. doi:10.1145/2872427.2883041.

[20] Z. Yang, I. King, Z. Xu, E. Oja, Heavy-tailed symmetric stochastic neighbor embedding, in: Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, A. Culotta (Eds.), Advances in Neural Information Processing Systems, Vol. 22, Curran Associates, Inc., 2009.

[21] M. Abe, J. Miyao, T. Kurita, q-SNE: Visualizing Data using q-Gaussian Distributed Stochastic Neighbor Embedding , in: 2020 25th International Conference on Pattern Recognition (ICPR), IEEE Computer Society, Los Alamitos, CA, USA, 2021, pp. 1051–1058. doi:10.1109/ICPR48806.2021.9412900.

[22] J. N. Böhm, P. Berens, D. Kobak, Attraction-repulsion spectrum in neighbor embeddings, J. Mach. Learn. Res. 23 (1) (Jan. 2022).

[23] J. A. Lee, D. H. Peluffo-Ordóñez, M. Verleysen, Multi-scale similarities in stochastic neighbour embedding: Reducing dimensionality while preserving both local and global structure, Neurocomputing 169 (2015) 246–261. doi:https://doi.org/10.1016/j.neucom.2014.12.095.

[24] L. van der Maaten, G. Hinton, Visualizing data using t-SNE, Journal of Machine Learning Research 9 (2008) 2579–2605.

[25] T. T. M. Consortium, A single-cell transcriptomic atlas characterizes ageing tissues in the mouse, Nature 583 (7817) (2020) 590–595. doi:10.1038/s41586-020-2496-1.

[26] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, KDD'96, AAAI Press, 1996, p. 226–231.

[27] I. Robinson, E. Pierce-Hoffman, Tree-sne: Hierarchical clustering and visualization using t-sne, arXiv preprint arXiv:2002.05687 (2020).

[28] F. V. Paulovich, A. Arleo, S. van den Elzen, When dimensionality reduction meets graph (drawing) theory: Introducing a common framework, challenges and opportunities, in: Computer Graphics Forum, Wiley Online Library, 2025, p. e70105.

[29] Tasic, et al., Shared and distinct transcriptomic cell types across neocortical areas, bioRxiv (2017). arXiv:https://www.biorxiv.org/content/early/2017/12/06/229542.full.pdf, doi:10.1101/229542.

[30] N. Pezzotti, B. P. F. Lelieveldt, L. v. d. Maaten, T. Höllt, E. Eisemann, A. Vilanova, Approximated and user steerable tsne for progressive visual analytics, IEEE Transactions on Visualization and Computer Graphics 23 (7) (2017) 1739–1752. doi:10.1109/TVCG.2016.2570755.

[31] N. Pezzotti, T. Höllt, B. Lelieveldt, E. Eisemann, A. Vilanova, Hierarchical stochastic neighbor embedding, Comput. Graph. Forum 35 (3) (2016) 21–30.

[32] W. E. Marcílio-Jr, D. M. Eler, F. V. Paulovich, R. M. Martins, Humap: Hierarchical uniform manifold approx-

imation and projection, IEEE Transactions on Visualization and Computer Graphics 31 (9) (2025) 5741–5753. doi:10.1109/TVCG.2024.3471181.

[33] C. de Bodt, D. Mulders, M. Verleysen, J. A. Lee, Fast multiscale neighbor embedding, IEEE Transactions on Neural Networks and Learning Systems (2020) 1–15doi:10.1109/TNNLS.2020.3042807.

[34] Y. Fang, W. Wang, B. Xie, Q. Sun, L. Wu, X. Wang, T. Huang, X. Wang, Y. Cao, Eva: Exploring the limits of masked visual representation learning at scale (11 2022). doi:10.48550/arXiv.2211.07636.

[35] M. Belkin, P. Niyogi, Laplacian eigenmaps and spectral techniques for embedding and clustering, in: T. Dietterich, S. Becker, Z. Ghahramani (Eds.), Advances in Neural Information Processing Systems, Vol. 14, MIT Press, 2001.

3.7 Neighbour embeddings: beyond visualisation

The capacity of Func-SNE to embed data into arbitrary dimensions, along with its theoretical ability to adapt to new data points without overhead, has sparked much thought about opening the field of NE to new applications, beyond visualisation. One of the first hints that NE could be used outside of visualisation was given in [55], where a parametric version of t-SNE was used as a feature extractor for classification tasks, with promising results. The method never really took off, in part due to the difficulty of training accurate parametric models for NE, possibly for reasons explained in Section 1.2. Being able to embed new data points accurately and quickly towards dimensionalities greater than 3 could revive interest in this direction. This work does not invent anything new, it rather aims to remind or show the audience that there is much untapped potential in NE methods, and that modern advances render NE methods ripe for renewed interest in broader ML contexts.

In this work, two examples on how to use NE directly for ML tasks are presented. One use case presented here is the use of NE for feature extraction, with spectacular results in contexts where supervised data is scarce, in line with the experiments of [55].
The second use case is hierarchical clustering, where NE in conjunction with DBSCAN [45] are used to extract increasingly fine-grained clusters from data, through the use of LD similarity distributions with increasingly heavy tails. The hierarchy of clusters is represented as a graph, which can then be used algorithmically or visualised using a force-directed layout.
In the course of writing this thesis, the work *Tree-SNE: Hierarchical clustering and visualization using t-SNE* [93] came to the author's attention. Although not peer-reviewed, it precedes the present work and proposes a hierarchical embedding of points conceptually similar to the one suggested here. The failure to cite this work is regrettable and unintentional. Tree-SNE is designed with visualisation in mind: the algorithm consists in stacking on top of each other many 1-dimensional embeddings of the data such that each embedding, or "level", has a heavier LD kernel tail weight than the previous level and a lower perplexity; each level is initialised with the previous one to keep a similar organisation. This procedure elegantly grows a 2-dimensional image showing a tree-like structure as clusters fragment into smaller components, without strictly necessitating the use of

other algorithms to explicitly extract clusters.
In contrast to Tree-SNE, the method suggested in the following paper purposefully embeds the data into "medium" dimensionalities (for instance, a 4-dimensional space), to give the moving points ample room to manoeuvre, allowing the representation of higher dimensional structures. This approach can extract more faithful clusters by reducing the chance of extracting artefacts arising from the low expressivity of 1- or 2- dimensional spaces. Such artefacts can appear for instance when two points are attracted to each other but separated by a mass of other points: the repelling mass of points can prevent the two from meeting in very LD spaces, but additional degrees of freedom in LD can allow the two points to circumvent the obstacle.
Arguments can be made in favour of both approaches, and further investigation on the impact of the LD space dimensionality in the extraction of clusters through NE seems like an important research direction if NE are to be used for other tasks than direct visualisation.

Additionally, in this work, the ImageNet dataset is embedded into 32 dimensions using Func-SNE. One can question the possibility of accurately embedding such a complex dataset into only 32 dimensions, Annex 5.1 studies the question and proposes yet another potential use of NE: estimating the intrinsic dimensionality of data. While the annex doesn't prove the possibility of fully capturing the nuances of ImageNet in 32 dimensions, preliminary results are encouraging and do not contradict this hypothesis. As a note for the reader, this annex was supposed to be a brief detour explaining basic intuition about intrinsic dimensionality, but the quick stroll has evolved into an odyssey, and the annex might deserve its own subsection next to the other contributions.

MLVis 2025: Machine Learning Methods in Visualisation for Big Data (2025)
D. Archambault, I. Nabney and J. Peltonen (Editors)

Neighbour Embeddings: Beyond Visualisation

Pierre Lambert[†1] , Edouard Couplet[1] , Michel Verleysen[1] , and John Aldo Lee[1,2]

[1]Université catholique de Louvain - ICTEAM/ELEN, Belgium
[2] Université catholique de Louvain - IREC/MIRO, Belgium

Abstract

Machine learning (ML) has brought powerful tools to the visualisation community, particularly through neighbour embeddings (NE). This family of algorithms enables the intuitive visualisation of high dimensional datasets, by representing these in 2- or 3-dimensional spaces. This paper argues that as NE algorithms have progressed and diversified within the visualisation domain, they have matured into powerful yet often simple methods whose potential remains largely underutilised in broader machine learning contexts. This argument is illustrated by showing through two use cases, clustering and data preprocessing before a supervised task, how NE can contribute meaningfully with little additional algorithmic effort.

1. Dimensionality reduction and neighbour embedding

Specialized by evolution to fit in a three-dimensional environment, the human brain excels at extracting and interpreting visual information. This helps for survival, but less so in data analytics, where high-dimensional (HD) data are common. To circumvent the issue and attempt to get the gist of HD data, methods of dimensionality reduction (DR) for visualisation were developed by the machine learning (ML) community.

Historically, one of the first DR algorithms was PCA [F.R01], which is a linear projection of the data such that the axes in the low dimensional (LD) space best capture the variance in the data. The method is a staple in ML, commonly used both for algorithmic processing (with the LD dimensionality generally higher than 2 or 3) and for data visualisation (LD dimensionality below 3). The linear nature of PCA limits the amount of information that can be captured on each axes, severely hampering its usefulness when used for visualisation. To counter this limitation, the family of neighbour embeddings (NE) was introduced in 2002 by G. Hinton, with an original method called stochastic neighbor embedding (SNE) [HR02]. In this new paradigm, the data is embedded in a LD space through iterative refinements of a non-linear, non-parametric transformation of data, with the aim to preserve neighbourhoods from HD to LD. This allows for much more expressive embeddings, as witnessed in Fig. 1. The figure shows the representation of 60.10^3 images of handwritten digits in 2 dimensions, using either PCA or *t*-SNE [vdMH08], a method of NE significantly improving on SNE. The NE reveals clearly separated clusters in the data, corresponding to the different 10 digits despite not using any supervised signals.

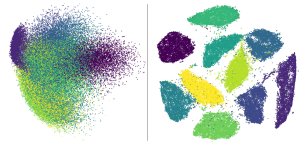


Figure 1: *PCA projection (left) and t-SNE embedding (right) of the MNIST database of handwritten digits.*

The impressive performance of NE for visualisation have propelled this family of algorithms to the centre of attention, both in applied domains [TYG*18, ADT*13, YSH*24] and in the research community. This surge in interest has led to the development of numerous methods of NE, each with unique characteristics, yet all sharing the common goal of data visualization. In short, recent advances in NE have increased their speed [MHM20, LRH*19, LCdBL24], offered more flexible ways to define neighbourhoods [KLS*20, AMK21], (perhaps inadvertently) allowed NE to embed in more than 3 dimensions [MHM20, LCdBL24], and permit new data points to be added to embeddings, either directly [LCdBL24] or in a post-hoc manner [KB19].

The purpose of this paper is to take a step back and realise that the advancements in the NE family now make them practical for direct use in a wider ML context, beyond the visualisation use case. This paper proposes two such use cases and shows that NE can be applied to these meaningfully and with minimal algorithmic modification. In section 2 we propose a novel, natural extension to NE that performs hierarchical clustering. In section 3, we show that preprocessing data using NE before a supervised task can produce embeddings which facilitate generalisation and greatly surpass other unsupervised methods, as was first hinted when parametric approximation was introduced to the community [vdM09].

† PL is a FRIA grantee of the Fonds de la Recherche Scientifique - FNRS. JAL is a Research Director with the F.R.S.-FNRS.

© 2025 The Authors.
Proceedings published by Eurographics - The European Association for Computer Graphics.
This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

DOI: 10.2312/mlvis.20251156

delivered by EUROGRAPHICS DIGITAL LIBRARY www.eg.org diglib.eg.org

In particular, the preprocessing application beats the state of the art one shot learning performance on ImageNet when starting from a latent representation of the images drawn from a vision-language transformer trained in a self-supervised manner.

2. Harnessing the inductive bias of NE towards clustering

Neighbour embeddings produce embeddings such that, for each point, neighbourhoods in the LD space match the neighbourhoods in the original HD space, using Euclidean distance in LD and a user-given distance metric in HD. The trick introduced by SNE allowing the formulation of this objective in derivable terms is to model neighbourhoods through a smooth kernel. Typically, the kernel is Gaussian in HD, and has heavier tails in LD. This purposeful mismatch in the HD and LD kernels is what essentially distinguishes t-SNE, as well as other recent variants of NE, from genuine SNE. Heavier tails in the LD kernel than in HD tends to shorten short-range distances, while they stretch mid- to long-range distances. This induced distance transformation, primarily meant to address issues of norm concentration [FWV07] and embedding space crowding [vdMH08], has the often observed after effect of also tightening clusters and magnifying gaps between them the LD embedding space. Broader gaps help data visualisation and ease interpretation, hypothesis formation or validation, as well as other similar exploratory tasks. Recent works [AMK21, KLS*20] have studied the effect of varying the kurtosis of the LD kernel and concluded that, the heavier the tail, the more easily the manifold will be torn in less dense regions, resulting in more clusters in the LD space. This tendency can be witnessed in Fig. 2, where embeddings of the MNIST digits datasets have been produced with increasing LD tail weights, from left to right. Looking at this figure,



Figure 2: *Embeddings of the MNIST dataset of handwritten digits with increasing LD kernel tails, from left to right.*

two observations stand out at a glance. First, the visually identifiable clusters are clearly separated in the embedding by broad strips of empty space. Second, the number of clusters in the LD space seems to relate monotonically to the weight of the LD kernel tails. From the first observation, one can intuit that a clustering algorithm running in an embedding produced by a NE algorithm would likely be less affected by the curse of dimensionality than in the original HD space, where the norms and distances tend to concentrate [FWV07]. This lead to works using NE to facilitate clustering in genomics [LSH15]. The second observation leads to the hypothesis that, if data has an underlying hierarchical structure, sweeping the kernel tail weight hyper-parameter from light to heavy tails could progressively reveal different levels of that hierarchy, with light tails showing large-scale groupings of points and heavier tails uncovering finer-grained relationships.

From these observations, one can devise a hierarchical clustering procedure. Considering a data embedding starting with light LD kernel tails, which progressively evolves as the LD kernel tails grow heavier, as if transitioning from left to right in Fig. 2. Let us take snapshots of the changing embedding at given intervals, the saved embeddings can be written $\mathcal{X}^\ell$, the level ℓ indexes the advancement in growing kernel tail weight. On each embedding $\mathcal{X}^\ell$, a clustering algorithm is performed to extract all the clusters $C^{(\ell)}$. Building on the observation that the number of clusters tends to increase when the LD tails get heavier, one can now build a graph capturing the fragmentation of large clusters into smaller ones by considering each cluster as a node, and building edges e_{ij} between $C_i^{(g)}$ and $C_j^{(h)}$ such that

$$e_{ij} = \alpha \cdot \frac{|C_i^{(g)} \cap C_j^{(h)}|}{\min\left(|C_i^{(g)}|,\ |C_j^{(h)}|\right)}, \qquad \alpha = \begin{cases} 1 & \text{if } |h-g| = 1 \\ 0 & \text{else} \end{cases} .$$

Once the graph is obtained, one can utilise the results algorithmically or visualise them using a graph embedding algorithm, like in Figures 3 and 4, which use a force directed layout. For a cluster C_i^ℓ, the node displayed in the figures has a size proportional to $\sqrt{|C_i^\ell|}$, with bounds. The colour of the node is proper to the level ℓ of its corresponding cluster, transitioning from blue for light LD tails, to pink for heavy tails. A central node corresponding to the whole dataset has been artificially added, for aesthetic purposes. The NE was performed using an accelerated and flexible method introduced in [LCdBL24]. Concerning the hyper-parameters, the NE uses a perplexity of 35, an Euclidean distance metric in HD, and an embedding dimensionality of 4. Clustering was carried out with DBSCAN [EKSX96]. The hyper-parameters of DBSCAN were tuned manually, and this task was not as tedious as it can be when working directly with HD data, likely due to the lowered dimensionality and to the tendency of NE to overemphasise the clusters by magnifying the gaps between them.

One might observe that, given the speed of the DBSCAN algorithm along with the NE algorithm (with a complexity of $\mathcal{O}(N)$ per iteration, modern GPU implementations can embed the MNIST dataset in seconds), the proposed algorithm can be reasonably fast. Moreover it may be fully automatised by using a heuristic to drive the hyper-parameter tuning of the DBSCAN algorithm, for instance by requiring the number of clusters on level ℓ to be larger than the number of clusters found on level $\ell - 1$, ℓ increasing with the kernel tails.

Figure 3 shows a global overview of MNIST by looking at 3 levels of LD kernel tail weights, plus the central root node of the tree. The representation concurs with the 101 basics of graphology. For instance, the visually similar digits 5, 3, and 8 distinguish themselves from the rest on the upper levels of the hierarchy, before speciating into their own sub-graphs. The digits 1 with a short dash on top are closely related to poorly-written 9, and the digits 1 with a wineglass foot are considered as a mixture between 2 and 1, which makes subjective sense. Figure 4 gathers a few zoom-ins from another graph embedding with an additional level of depth, unveiling in a structured manner the myriad ways digits can be portrayed. The next section proposes another application of NE to ML: reducing the dimensionality of the data for downstream automatic processing.

3. Neighbour embedding as a pre-processing step

When using data for an automated ML task, it is common to reduce data dimensionality first, generally using PCA or sometimes autoencoders, in order to get a tractable number of meaningful features. Often-times, a good initial dimensionality reduction step can also

© 2025 The Authors.
Proceedings published by Eurographics - The European Association for Computer Graphics.

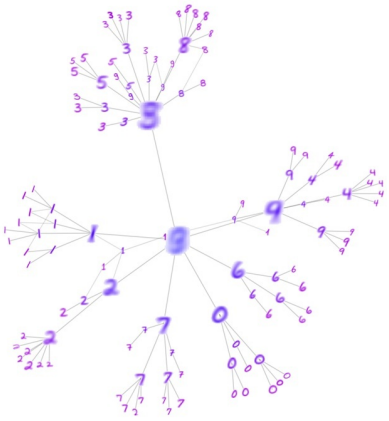


Figure 3: *Hierarchical representation of the MNIST data set of handwritten digits, using the introduced methodology of clustering.*

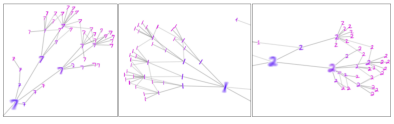


Figure 4: *Close up views of the computed graph, with one more level of depth.*

de-noise the data, improving performance in downstream tasks. A parametric approximation of t-SNE introduced in [vdM09] hints that NE can generate representations that facilitate generalisability. The capacity of modern NE to natively produce embeddings of dimensionality greater that 3 and to allow new points to be added to the embedding leads to this section, which revisits the use case of data preprocessing using non-parametric NE across 6 datasets, in Fig. 5 and Table. 1. The datasets investigated in these experiments are staples in supervised ML tasks, freely available online. For each dataset, the quality of different LD embeddings are evaluated in a one-shot learning context as well as in the more traditional 10-fold cross validation scenario. In the one shot scenario, the mean accuracy of 100 trials is reported; in each trial, one random point per label has its class revealed, and the remaining data points have their labels inferred. Every featured NE embedding is performed using an Euclidean metric in the HD space, and a arbitrarily set perplexity hyper-parameter (or the analogue `n_neighbours` in UMAP) of 35. The auto-encoder embeddings are performed using convolutional auto-encoders for MNIST and COIL-20, and standard feed-forward auto-encoders for the rest, each having one hidden layer between the latent space and the input or output layers. In the following experiments, we use XGBoost [CG16] for the 10-fold cross-validation and a 1-NN classifier for one-shot learning.

In Fig. 5, the first column shows the accuracy as a function of the embedding dimensionality in the 10-fold cross validation scenario, and the second column in the one-shot learning scenario. The dashed black line shows the performance if fitting the classification models in the original HD space. The advantage of NE is evident, as the accuracies using NE are systematically superior to those using auto-encoder or PCA, with significantly less dimensions. The gain in accuracy is impressive in one-shot learning scenarios.

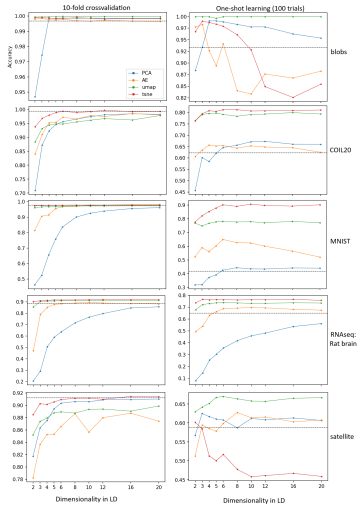


Figure 5: *Comparison of classification accuracies on embeddings produced with different DR methods.*

In addition to the 5 datasets that are reported in Fig. 5, the protocol is tested on the latent representation of the ImageNet dataset pulled from a large vision-language model called EVA [FWX*22]. In this latent space, the 1.2 million images are represented as vectors in a 1280-dimensional space. The ImageNet dataset has 1000 different classes, making high top-1 accuracies challenging to obtain. Three representations of the data are evaluated in a similar manner to the trials of Fig. 5: the bare latent representation, a PCA projection keeping the first 192 principal components with explained variance of 71.1%, and a t-SNE embedding with 32 dimensions in the latent space. These dimensionalities are chosen arbitrarily, and t-SNE ran from the 192 dimensional space of PCA rather than the initial EVA representation. Neighbour embedding was carried out using increased attractive forces and learning rate, as customary on very large data sets [KB19]. Classification performance in a 10-fold cross-validation for this dataset involved a simple multilayer perceptron with one hidden layer, for the sake of short running times. The one-shot learning evaluation involved a 1-NN classifier using Euclidean distances, as for the experiments in Fig. 5. The results of these experiments are reported in the last 4 rows of Table 1. The first 3 columns correspond to classifiers working in the EVA latent space, the PCA space, and the t-SNE space. The last column refers to a mixture of experts (MoE) [RPM*21] and shows the current state-of-

© 2025 The Authors.
Proceedings published by Eurographics - The European Association for Computer Graphics.

	1280, EVA	192, PCA	32, NE	MoE
zero-shot (top-1)	**84.0%**	0.0%	0.0%	
one-shot (top-1)	47.3%	45.9%	**76.2%**	68.7%
one-shot (top-5)	67.3%	67.8%	**92.0%**	
crossval (top-1, train)	84.9%	83.2%	**87.2%**	
crossval (top-1, test)	87.2%	87.2%	**88.7%**	

Table 1: *Comparison of classification performance on ImageNet across various scenarios.*

the-art top-1 one-shot accuracy using all 1000 classes on ImageNet, as reported in their paper. These experiments show that the 32-dimensional t-SNE representation enables significant improvements over the state of the art despite using a simple 1-NN classifier. The one-shot learning performance and the tighter train and test accuracies spread in the cross validation scenario indicates that the 32-dimensional t-SNE representation is a stalwart embedding and that NE may yet possess untapped potential by generating data representations that facilitate generalisation. The 32-dimensional embedding is publicly available at `https://github.com/PierreLambert3/Imagenet-tSNE-embedding`.

The elephant in the room concerns the 'zero-shot' accuracies in Table 1. Vision-language transformers consist of a text encoder and an image encoder, sharing the same latent space. They are generally trained using a contrastive loss as illustrated in Fig. 6. The shared latent space allows for so-called 'zero-shot' scenarios, where representations of inexistent images are generated by prompts like *'an image of __'* and used as artificial points with a known label, from which classification of true images can be initiated. The first row of Table 1 shows the zero-shot performance on ImageNet, using the same protocol as for the one-shot case, with the exception that the PCA projection and t-SNE embedding were performed on a dataset concatenating the representations of the actual ImageNet data points, which constitute the test set, and 1000 prototypes, one for each class label, which are generated by the text encoder. The same prototypes as in [ZA24] are used. The impressive zero-shot performance in the raw EVA space (84%) aligns with other works, while the abysmal accuracies of PCA and t-SNE in that first row call for further inquiry. A t-SNE embedding of the dataset is shown in the right panel of Fig. 6, the embedding hints that the text encoder generated points laying outside of the manifold of encoded natural images. This is confirmed by computing 'inter-label, intra-modularity' Euclidean distances in the EVA space between generated representations and comparing them to "intra-label, inter-modularity" distances between representations of true images and their generated prototypes. These distances average to 0.9 and 1.3, respectively. In other words, if ImageNet featured ducks and witches, the representations generated for the prompts 'An image of a witch' and 'An image of a duck' would lie closer to each other in the EVA space than the representations of real duck images are to their own prototype (i.e., the data point generated with 'An image of a duck'). From these observations, one can conclude that the text and image encoders have accorded themselves to produce representations featuring meaningful Euclidean distances in the context of classification using generated prototypes —this explains the high zero-shot accuracy— but the encoders have not learned to align manifolds in the latent space. In an artificial setting where prototypes for each label are generated by taking the centroid of each class in the vision encoder space (red points in 6), the top-1 accuracy in the EVA space rises from 84% for the zero-shot scenario to 85.8%. This shows that a 'better' alignment between the vision and text encoders could lead to enhanced zero-shot performances. In this example of zero-shot learning, the train set and the test set lie in separated manifolds, which is uncommon in data encountered in ML, the proposed method of pre-processing data with t-SNE to enhance generalisability fails in such cases.

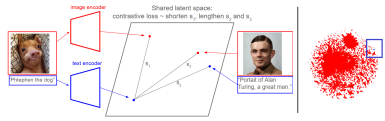


Figure 6: *Schematic explanation of vision-language models (left), and, (right) t-SNE embedding of the dataset built in the zero-shot case (red: representations of ImageNet, blue: prototypes generated from text).*

Figure 7 shows the power of the inductive bias of NE towards clustering, without the constraint of embedding in only two dimensions for visualisation. To illustrate that effect visually without further bias or complicated nonlinear distortions, 2D PCA projections are shown, starting from either the raw EVA 1280-dimensional features directly down to two dimensions or the 32D t-SNE embedding of the EVA features. The clusters clearly drift away like galaxies in an expanding universe. Notice the typical artifact of PCA projection, quite common too in spectral clustering [VL07,LS19,BN03], where only a few clusters are visually separated in the two available dimensions, appearing like long spikes, while the others collapse towards the centre of the linear projection.

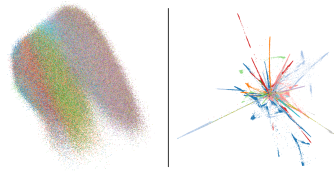


Figure 7: *Two-dimensional PCA projections of the 1280-dimensional representation of ImageNet by EVA. Left: raw EVA features (1280 (EVA) → 2 (PCA vis.)); right: EVA features after embedding in 32D with t-SNE (1280 (EVA) → 192 (PCA init.) → 32 (t-SNE) → 2 (PCA vis.)) .*

4. Conclusion

Experiments show that NE in new use cases, namely, clustering and data pre processing, can lead to meaningful representations. In particular, NE with dimensionality higher than 3 can lead to representations that generalise better, yielding ground-breaking results in scenarios of one-shot learning with simple classifiers. An exciting perspective would be to take inspirations from the capacity of NE to represent HD data robustly to devise new loss functions or regularisation terms in deep learning approaches.

© 2025 The Authors.
Proceedings published by Eurographics - The European Association for Computer Graphics.

References

[ADT*13] AMIR E.-A., DAVIS K., TADMOR M., SIMONDS E., LEVINE J., BENDALL S., SHENFELD D., KRISHNASWAMY S., NOLAN G., PE'ER D.: Visne enables visualization of high dimensional single-cell data and reveals phenotypic heterogeneity of leukemia. *Nature biotechnology 31* (05 2013). doi:10.1038/nbt.2594. 1

[AMK21] ABE M., MIYAO J., KURITA T.: q-SNE: Visualizing Data using q-Gaussian Distributed Stochastic Neighbor Embedding . In *2020 25th International Conference on Pattern Recognition (ICPR)* (Los Alamitos, CA, USA, Jan. 2021), IEEE Computer Society, pp. 1051–1058. URL: https://doi.ieeecomputersociety.org/10.1109/ICPR48806.2021.9412900, doi:10.1109/ICPR48806.2021.9412900. 1, 2

[BN03] BELKIN M., NIYOGI P.: Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation 15*, 6 (2003), 1373–1396. doi:10.1162/089976603321780317. 4

[CG16] CHEN T., GUESTRIN C.: Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Aug. 2016), KDD '16, ACM, p. 785–794. URL: http://dx.doi.org/10.1145/2939672.2939785, doi:10.1145/2939672.2939785. 3

[EKSX96] ESTER M., KRIEGEL H.-P., SANDER J., XU X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining* (1996), KDD'96, AAAI Press, p. 226–231. 2

[F.R01] F.R.S. K. P.: Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science 2*, 11 (1901), 559–572. doi:10.1080/14786440109462720. 1

[FWV07] FRANCOIS D., WERTZ V., VERLEYSEN M.: The concentration of fractional distances. *IEEE Trans. Knowl. Data Eng. 19*, 7 (2007), 873–886. 2

[FWX*22] FANG Y., WANG W., XIE B., SUN Q., WU L., WANG X., HUANG T., WANG X., CAO Y.: Eva: Exploring the limits of masked visual representation learning at scale, 11 2022. doi:10.48550/arXiv.2211.07636. 3

[HR02] HINTON G. E., ROWEIS S.: Stochastic neighbor embedding. In *Advances in Neural Information Processing Systems* (2002), Becker S., Thrun S., Obermayer K., (Eds.), vol. 15, MIT Press. URL: https://proceedings.neurips.cc/paper_files/paper/2002/file/6150ccc6069bea6b5716254057a194ef-Paper.pdf. 1

[KB19] KOBAK D., BERENS P.: The art of using t-sne for single-cell transcriptomics. *Nat Commun 10, 5416* (2019). 1, 3

[KLS*20] KOBAK D., LINDERMAN G., STEINERBERGER S., KLUGER Y., BERENS P.: Heavy-tailed kernels reveal a finer cluster structure in t-sne visualisations. In *Machine Learning and Knowledge Discovery in Databases* (Cham, 2020), Brefeld U., Fromont E., Hotho A., Knobbe A., Maathuis M., Robardet C., (Eds.), Springer International Publishing, pp. 124–139. 1, 2

[LCdBL24] LAMBERT P., COUPLET E., DE BODT C., LEE J. A.: Estimated neighbour sets and smoothed sampled global interactions are sufficient for a fast approximate tsne. In *32nd European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN 2024, Bruges, Belgium, October 9-11, 2024* (2024). URL: https://doi.org/10.14428/esann/2024.ES2024-203, doi:10.14428/ESANN/2024.ES2024-203. 1, 2

[LRH*19] LINDERMAN G. C., RACHH M., HOSKINS J. G., STEINERBERGER S., KLUGER Y.: Fast interpolation-based t-sne for improved visualization of single-cell rna-seq data. *Nature methods 16* (2019), 243 – 245. 1

[LS19] LINDERMAN G. C., STEINERBERGER S.: Clustering with t-sne, provably. *SIAM Journal on Mathematics of Data Science 1*, 2 (2019), 313–332. URL: https://doi.org/10.1137/18M1216134, arXiv:https://doi.org/10.1137/18M1216134, doi:10.1137/18M1216134. 4

[LSH15] LUX M., SCZYRBA A., HAMMER B.: Automatic discovery of metagenomic structure. In *2015 International Joint Conference on Neural Networks (IJCNN)* (2015), pp. 1–8. doi:10.1109/IJCNN.2015.7280500. 2

[MHM20] MCINNES L., HEALY J., MELVILLE J.: Umap: Uniform manifold approximation and projection for dimension reduction, 2020. arXiv:1802.03426. 1

[RPM*21] RIQUELME C., PUIGCERVER J., MUSTAFA B., NEUMANN M., JENATTON R., PINTO A. S., KEYSERS D., HOULSBY N.: Scaling vision with sparse mixture of experts. In *Proceedings of the 35th International Conference on Neural Information Processing Systems* (Red Hook, NY, USA, 2021), NIPS '21, Curran Associates Inc. 3

[TYG*18] TASIC B., YAO Z., GRAYBUCK L. T., SMITH K. A., NGUYEN T. N., BERTAGNOLLI D., GOLDY J., GARREN E., ECONOMO M. N., VISWANATHAN S., ET AL.: Shared and distinct transcriptomic cell types across neocortical areas. *Nature 563*, 7729 (2018), 72–78. 1

[vdM09] VAN DER MAATEN L.: Learning a parametric embedding by preserving local structure. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics* (Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA, 16–18 Apr 2009), van Dyk D., Welling M., (Eds.), vol. 5 of *Proceedings of Machine Learning Research*, PMLR, pp. 384–391. URL: https://proceedings.mlr.press/v5/maaten09a.html. 1, 3

[vdMH08] VAN DER MAATEN L., HINTON G.: Visualizing data using t-sne. *Journal of Machine Learning Research 9*, 86 (2008), 2579–2605. URL: http://jmlr.org/papers/v9/vandermaaten08a.html. 1, 2

[VL07] VON LUXBURG U.: A tutorial on spectral clustering. *Statistics and computing 17*, 4 (2007), 395–416. 4

[YSH*24] YANG W., SUN C., HUSZÁR R., HAINMUELLER T., KISELEV K., BUZSÁKI G.: Selection of experience for memory by hippocampal sharp wave ripples. *Science 383*, 6690 (2024), 1478–1483. URL: https://www.science.org/doi/abs/10.1126/science.adk8261, arXiv:https://www.science.org/doi/pdf/10.1126/science.adk8261, doi:10.1126/science.adk8261. 1

[ZA24] ZANELLA M., AYED I. B.: On the test-time zero-shot generalization of vision-language models: Do we really need prompt learning?, 2024. URL: https://arxiv.org/abs/2405.02266, arXiv:2405.02266. 4

© 2025 The Authors.
Proceedings published by Eurographics - The European Association for Computer Graphics.

3.8 Enhancing label diffusion with lessons from neighbour embeddings

The performance of ML is incontestable when models are trained in environments rich in usable information; this is the case for large language models, as the internet offers virtually unlimited text data directly usable by their self-supervised training. The performance of ML models tends to drop catastrophically when the data doesn't abundantly cover pairs of inputs and expected outputs.
In classification, this happens when there are many unlabelled observations, but only few labelled examples per class. This situation occurs when labelling data is costly or when classes or concepts appear on the fly, which happens in so called *zero-shot learning*. More importantly in the grand scheme of things, the question of learning from an abundance of unlabelled data and only few labelled examples occurs naturally in biological brains and in *embodiment learning*. In embodiment learning, an agent interacts with its environment and receives in return fresh sensory inputs and sometimes feedbacks such as pain or pleasure. An example on why learning quickly from few labels is critical there can be found in the case of an agent touching fire for the first time. The agent will receive a painful feedback, which can be considered as a label "avoid this situation"; it is then critical for the agent to solidly associate this label to the internal representation of recent events in order to avoid touching fire again in the future.
In the literature, scenarios with abundant unlabelled data and few labelled examples take several forms which will not be detailed here; terms such as *few-shot learning*, *one-shot learning*, *semi-supervised learning*, *transductive learning* are variations on the theme. It is however important to note that in scenarios of few-shot and one-shot learning, a feature extractor is often pre-trained using densely labelled data with a set of classes A, and then the task is to adapt a classifier for a new set of classes B with only few labelled examples in this new set [94, 95]. This generally leads to better performances, as the feature extractor has richer information, but this scenario is not always applicable in practice. Recent results in "Rethinking Few-Shot Image Classification: A Good Embedding is All You Need?" [96] suggest that extracting features with supervised signals is not necessarily required to reach good performances for classifying with scarcely labelled data, as they achieve competitive results by training simple KNN or logistic regression classifiers on top of features extracted with self-supervised

methods such as a deep neural network trained by MoCo [97].

To conclude this journey on an opening rather than on a closing, this work learns from NE to devise a new algorithm applicable in transductive learning contexts with very few labelled examples. In transductive learning, unlabelled points which are encountered during training are also the points that need to be classified at test time. Rather than explaining the final algorithm directly, this Section retraces the thought process and iterative developments that led to the final method; this allows a better understanding of the impact of the individual properties of NE which are transposed to the new algorithm. Section 3.8.1 introduces a recurring example used throughout the development of the method, in which NE displays remarkable properties that supervised methods struggle to replicate. Section 3.8.2 details the first component of NE which is used in the algorithm. Section 3.8.3 details the second component of NE used in the algorithm. Finally, Section 3.8.4 presents preliminary results comparing the method to some existing approaches; a more thorough evaluation and comparison to the state-of-the-art will be performed in a dedicated paper.

3.8.1 First observations, and the Label Spreading baseline

Throughout this thesis, the remarkable capacity of NE to nearly perfectly separate classes in MNIST without supervision while working directly on Euclidean distances in HD has been a constant source of wonder. This motivated an investigation on the possibility to extract the principles behind this behaviour and graft them into a self-contained classification algorithm. In order to illustrate the performance of other methods and the iterative progress towards the final algorithm, a particular configuration of MNIST is used throughout this work. The configuration consists of a subset of MNIST containing only the digits 3, 5, and 8, with only one labelled example per class. Multiple random seeds were tested, and a seed offering particularly bad performances using a KNN classifier in HD was selected to devise a challenging scenario. Figure 3.1 shows embeddings of this restricted MNIST dataset using various DR methods, with the one prototype per class shown as larger points. There, the NE method on the right which features LD similarities with larger tails clearly separates the classes visually. The PCA projection and MDS embeddings show the typical blurry embedding which arises in datasets with concentrating distances in HD.

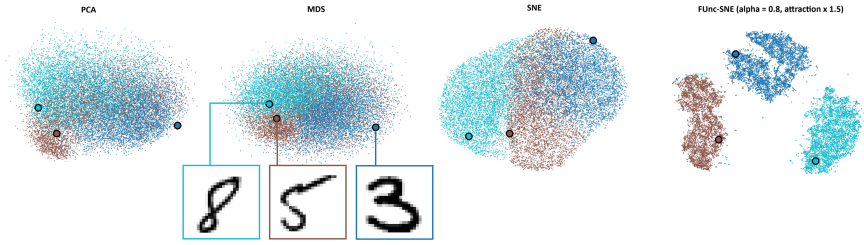


Figure 3.1: Embeddings of the restricted MNIST dataset, figuring only the digits 3, 5, and 8. The prototypes for each class, which are the points of known labels, are shown as larger points. The seed used to determine the three prototypes was chosen to be challenging.

In order to have a baseline for comparison, several standard methods used in semi-supervised learning were tested on this configuration; these are all trained on the HD data. The results are shown in Figure 3.2, featuring a global accuracy score for each method and the distribution of misclassified points across a NE representation of the data. The performance of these methods is quite poor, with accuracies ranging from 43% to 74%, showing the difficulty of the task.

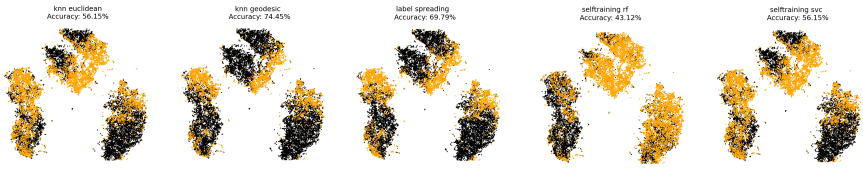


Figure 3.2: Classification performances on the restricted MNIST configuration of Figure 3.1, using various methods. From left to right: KNN classifier using Euclidean distances, KNN classifier using Geodesics, Label Spreading [98], random forest classifier [29] in semi-supervised mode, support vector machine classifier [99] in semi-supervised mode. The semi-supervised mode refers to scikit-learn's round based semi-supervised setting. All methods with meaningful hyperparameters were tuned using a grid search, the best results are reported. The pointwise errors are reported in the rightmost NE of Figure 3.1, but all classifications are performed in HD; yellow points are misclassified.

The proposed method takes inspiration from a family of algorithms called label propagation [100], a paradigm naturally suited for settings where points of known labels are rare. In these methods, the data is represented as a graph in which state vectors are diffused from point to point, if points have a known label, the state vector can be hardcoded or more softly influenced to reflect this known label. One widespread method of this family is *label spreading* [98], which is detailed here as it serves as a basis for the proposed method.
In label spreading, a graph is first built from the data using sparse KNN

sets or non-adaptive Gaussian kernels around each point. The graph, with edges stored in a square matrix $\mathbf{W}$, is then normalised symmetrically to produce the matrix $\mathbf{S}$ such that $\mathbf{S}_{ij} = \mathbf{W}_{ij}(\sqrt{\mathbf{D}_{ii}\mathbf{D}_{jj}})^{-1}$, where $\mathbf{D}$ is the diagonal degree matrix. One should notice that this normalisation preserves the relative importance of nodes when considering degrees. Once the graph is constructed, labels are encoded as vectors associated to each point, points that have a known label receive a one-hot encoding of their class, while unlabelled points receive zero vectors; the matrix $\mathbf{Y}$ contains these label vectors as rows. Another matrix $\mathbf{V}$ is initialised as a copy of $\mathbf{Y}$. Then, the labels are propagated through the graph by iterating:

$$\mathbf{V} \leftarrow \alpha\mathbf{S}\mathbf{V} + (1-\alpha)\mathbf{Y}, \tag{3.1}$$

where α is a hyperparameter between 0 and 1 controlling the trade-off between propagation and label retention. The paper [98] provides theoretical guarantees of convergence; once converged, the labels are predicted by taking the class corresponding to the largest value in each row of $\mathbf{V}$.

3.8.2 The first component: attractive forces

This Section details the first component of NE that is transposed to the family of label diffusion algorithms.

The graph used in the label spreading algorithm preserves the relative importance of nodes when considering their degrees. This is often a desirable property when working with graphs, but one can question its usefulness when diffusing labels, especially when known labels are few. Let us consider the following case: there are only two classes, and one point from each class has its label revealed; what happens if one of these points is a hub, with many attached edges, and another is a poorly connected point with a degree much lower than the hub? There, the label revealed on the first point will have a competitive advantage over the label of the non-hub point. When working on classification, common practice is to put all classes on an equal footing, for instance by using equidistant representations (such as one-hot encoding), or taking active measures to counter class imbalance.

This consideration is all the more important when considering the tendency of points in high-dimensional spaces to exhibit a wide range of degrees, with some recurring hubs [101]. As demonstrated in Figure 3.3, this behaviour is observed in the restricted MNIST dataset.

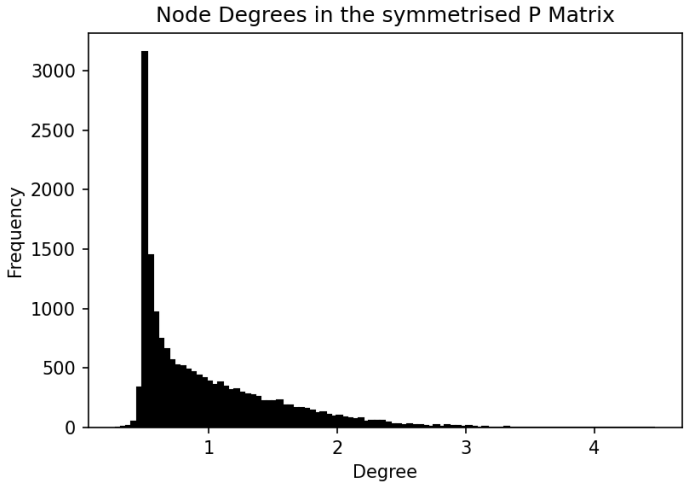


Figure 3.3: Distribution of the degree of each point according to the symmetrised $\mathbf{P}$ matrix as defined by t-SNE, in the restricted MNIST dataset. The perplexity is set to 7.

Following this intuition and staying true to the objective of using principles from NE, the first component of the proposed method is to devise a label diffusion process inspired from label spreading, which uses a degree-equalised matrix $\mathring{\mathbf{P}}$ corresponding to the symmetrised $\mathbf{P}$ matrix of t-SNE, normalised such that each row and each column sums to 1. This matrix is obtained by applying Sinkhorn-Knopp normalisation [102] on $\mathbf{P}$. Using this matrix, a diffusion process is defined as such:

$$\frac{d\mathbf{V}_i}{dt} \;=\; \mathrm{dt}\left(\alpha \sum_{j \in \mathrm{KNN}_i} \mathring{\mathbf{P}}_{ij} \left(\mathbf{V}_j - \mathbf{V}_i\right) \;+\; (1-\alpha)\, \frac{\mathbf{M}_i}{N_r(\mathbf{Y}_i)}\, \mathbf{Y}_i\right), \tag{3.2}$$

$$\begin{cases} \mathbf{M}_i = 1 & \text{if } i \text{ has a label,} \\ \mathbf{M}_i = 0 & \text{otherwise.} \end{cases} \tag{3.3}$$

With $N_r(\mathbf{Y}_i)$ being the number of revealed labels for the label assigned to point i; this should allow class imbalance to be taken into account.
In essence, points that do not have a revealed label diffuse their state $\mathbf{V}_\cdot$ towards their neighbours, similarly to how charges propagate in a circuit where resistances would be given by the inverse of $\mathring{\mathbf{P}}_{ij}$, and all capacitances would be equal. Points that have a revealed label are subject to the same dynamics and also receive a constant injection of "label current". This current encourages an infinite growth along the direction given by the one-hot

encoded vector of the known label; in practice, the left part of the equation pulls them back towards the mass of points and an equilibrium is reached. Figure 3.4 shows the errors and distribution of uncertainty across a NE of the restricted MNIST dataset, using the proposed method as it stands at this stage of the design process. There, comparing to label spreading, the accuracy has raised from 70% to 83%. However, when looking at the distribution of the errors, the property of NE to distinguish clearly between the three classes is far from being fully captured.

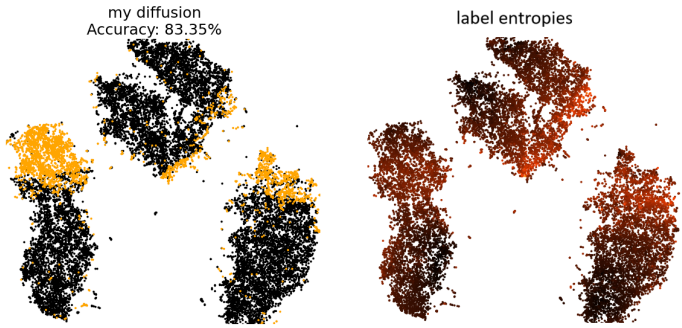


Figure 3.4: Errors and distribution of uncertainty across a NE of the restricted MNIST dataset, using the proposed method at this stage of development. Uncertainty is shown as the min-max normalised entropy of the vectors in $\mathbf{V}$, hot colours indicating high values.

It has been observed in Figure 1.8 that neighbours in HD can actually be placed quite far apart in LD, and it is quite common the have neighbours spanning multiple classes. The "unreliability" of some neighbours in HD regarding this classification objective could explain why the wrong labels seem to diffuse easily between what NE show as distinct clusters in the embedding. Therefore, the next step in the design of the method is to modify the diffusion process to increase the influence of "good" neighbours, and reduce that of "bad" neighbours.

One way to distinguish good from bad neighbours is to look one degree further: if two points are neighbours but share few mutual neighbours, then they can be considered as weakly related. To implement this idea, before equalising the degrees of $\mathbf{P}$, each value P_{ij} is multiplied by a *mutuality score* defined as the normalised intersection of the KNN sets of points i and j. The normalisation is performed by dividing the intersection by K, the neighbourhood size which is set to $\lceil 3\,\text{perplexity} \rceil$. This value for K is a common choice in accelerated NE methods [32], which comes from the realisation that the similarities in HD will have decayed enough at the

$\lceil 3\,\text{perplexity} \rceil^{th}$ neighbour to consider further points as having a negligible influence. The idea of using higher degree relationships in the paradigm of label diffusion is not entirely new: in [103], the authors explore the use of hyper-graphs to perform label diffusion, showing accuracy gains over standard label spreading.
After adding this soft filtering of neighbours, the accuracy rises above 91%, as shown in Figure 3.5. While the same dataset is used throughout this section, each iteration was tested on multiple datasets and random seeds and the progress witnessed on the restricted MNIST dataset is representative of the general trend.

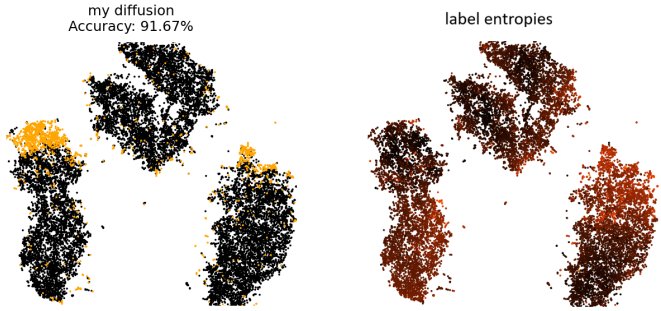


Figure 3.5: Errors and distribution of uncertainty across a NE of the restricted MNIST dataset, using the proposed method at this stage of development.

3.8.3 The second component: repulsive forces

An central element to NE are the repulsive forces; these enable the methods to cut through the diffuse structures of HD data and separate them clearly in LD. Examples have been shown throughout this thesis and most recently in Figure 3.1, where the SNE method doesn't separate the classes as well as the other NE method, which uses mismatched similarities with heavier tails in LD. The attraction-based label diffusion process devised so far still fails to separate the classes as well as NE methods naturally do, the next step is thus to try and transpose the repulsive forces of NE into the label diffusion process.

To the best of this humble scribe's knowledge, all methods in the family of label diffusion rely solely on attractive forces to propagate labels. This is somewhat surprising in retrospect, given that the analogy with electrical circuits was made explicit in works such as [104] and more generally

in *Random walks and electric networks* [105]. If we push the analogy further and consider a graph consisting of 2 isolated sub-graphs, charging one sub-graph positively should induce a negative charge in the other, as an electric field would. Coming from the NE standpoint, this analogy is even more striking, as the electric field scales inversely with distance, very close to how the repulsive forces in NE scale.

Adding the repulsive component to the diffusion process has been particularly challenging, the vast majority of attempts led to failure, but, eventually, a successful approach was found. As diverse as the failed attempts were, it seems in hindsight that they were all plagued by the same issue: repulsive forces in NE are well suited for visualisation and purely unsupervised contexts, but they need to be adapted for the needs of label diffusion where some points are more important than others. Intuition on that is summarised hereafter.

In NE, the fact that each point strongly repels points in the near proximity encourages point to have a distinct location in LD, allowing the fine grained details to be captured. However, the purpose of label diffusion is to group points sharing the same label together: ideally, all points of a given class should share the coordinates in LD, therefore one does not want local repulsions to overwhelm the attractive forces pulling together points of identical label.

Another reason why NE's repulsive forces cannot be used directly here, is that the global all-to-all far ranging nature of the repulsion in NE can often overwhelm the attractive forces, leading to situations where points are driven more by repulsion from the local occupancy of the LD space at a given time than by the HD similarities. While this contributes to the capacity of NE to disentangle complex structures in HD, it is detrimental to label diffusion where some points are more important than others: points of known label are expected to stay along a certain direction or close to a predefined location in LD, and in any case they should be driving the dynamics, not the other way around.

A solution to this last point would be to only apply repulsive forces on points that do not have a known label, immunising them to the strong currents of repulsion. However, another problem arises there: the points of known label are expected to be much less numerous than the unlabelled points and the dynamics of NE tend to follow Spock's mantra of "the needs of the many outweigh the needs of the few". Indeed, the dynamics of NE prioritise the satisfaction of the global cost function even if it means sacri-

ficing the preservation of the neighbourhood of individual points, this was witnessed in Fig. 1.8 where HD neighbours can be located far apart in LD. Restricting the movement of points with known labels leads in practice to these points being ignored by the mass of unlabelled points, defeating the purpose of the diffusion process. Attempts to counter this effect by adding regularisation terms forcing points of known labels to be significant to all other points have been tried, but there the NE inspired diffusion tends to become too constrained and converges to mediocre solutions, both on the standpoint of accuracy and on the preservation of structures.

These considerations led to a rethinking of how to apply repulsive forces in the diffusion process. In order to keep the dynamics mostly driven by the attractive forces, the aim is for the repulsive forces to act as gentle nudges rather than as an overwhelming current, accompanying the diffusion process rather than driving it. More concretely, these forces will no longer be strong enough to prevent points from sharing coordinates in LD, they will will no longer be applied in a N-to-N fashion, and they will also be shorter ranged.
In that spirit, for each point, the set of neighbours of neighbours in HD is determined, the set is called KNNofKNN$_i$ for the i^{th} point. Repulsive forces will only be applied between a point and these neighbours of neighbours, taking inspiration from the Mexican hat kernels sometimes used in self-organising maps [106] or in signal processing, and which mimic the principles of lateral inhibition [74] in biological neural networks.
Shadowing the neighbours of neighbours set, a square matrix $\mathbf{B}$ is initialised with all values set to 0 for points that are not neighbours of neighbours, and 1 for neighbours of neighbours; this matrix will scale the repulsive forces. Points in a row of KNNofKNN that occur multiple times have their associated values in $\mathbf{B}_{\cdot}$ set to 0, such that a neighbour of neighbour only contributes once. Once this binary filtering is done, for the i^{th} point, all the values in $\mathbf{B}_i$ corresponding to the neighbours of the first degree neighbour j are multiplied by $1 - m^s_{ij}$ where m^s_{ij} is the mutuality score that was used in Section 3.8.2 to scale attractive forces between i and j. In essence, this means that neighbours of strong neighbours will have a reduced repulsive influence, while neighbours of weak neighbours will have an increased repulsive influence. Once this is done, the $\mathring{\mathbf{B}}$ matrix is obtained by normalising $\mathbf{B}$ as was done for $\mathring{\mathbf{P}}$. In contrast to the $\mathbf{Q}$ matrix of NE, which is recomputed at each iteration from the current state of the embedding, $\mathbf{B}$ will remain constant throughout the diffusion process.

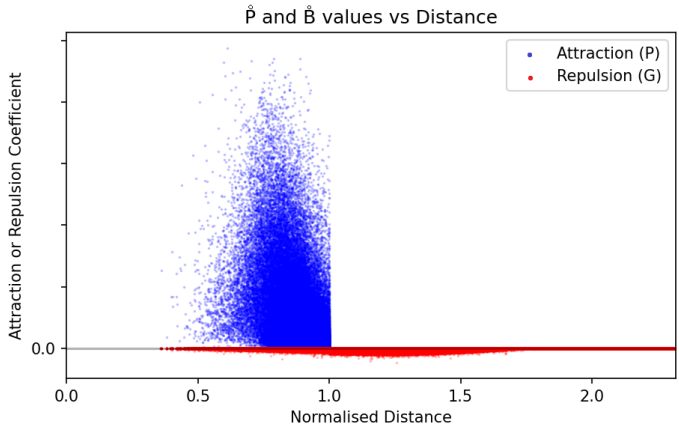


Figure 3.6: Distribution of the values of $\mathring{\mathbf{B}}$ and $\mathring{\mathbf{P}}$ on the restricted MNIST dataset. The distance to a neighbour or neighbour of neighbour on the x-axis is normalised by dividing by the distance to the farthest neighbour, which explains why all values of $\mathring{\mathbf{P}}$ are shown on x values smaller than 1. Some values of $\mathring{\mathbf{B}}$ are shown on x values smaller than 1, this indicates that neighbours of neighbours can also be first degree neighbours: the current algorithm doesn't filter these out, but this should be considered for further works. The "Mexican hat" shape of the attraction-repulsion profile is apparent.

Figure 3.6 shows the distribution of the values of $\mathring{\mathbf{B}}$ and $\mathring{\mathbf{P}}$ on the restricted MNIST dataset, plotted against the normalised distance to the neighbour or to the neighbour of neighbour. This figure shows that the idea of using a Mexican hat-like profile based on neighbourhood relationships also translates to distance relationships, as neighbours of neighbours tend to be located at a distance larger than first degree neighbours. The diffusion process with added repulsive forces is defined as such:

$$\frac{d\mathbf{V}_i}{dt} = \mathrm{dt}\left(\alpha \sum_{j \in \mathrm{KNN}_i} \mathring{\mathbf{P}}_{ij}\left(\mathbf{V}_j - \mathbf{V}_i\right) + (1-\alpha)\,\frac{\mathbf{M}_i}{N_r(\mathbf{Y}_i)}\,\mathbf{Y}_i \right.$$

$$\left. - \alpha\,\beta \sum_{l \in \mathrm{KNNofKNN}_i} \mathcal{K}_{il}\,\mathring{\mathbf{B}}_{il}\left(\mathbf{V}_l - \mathbf{V}_i\right)\right) \quad ,$$

$$\mathcal{K}_{il} = \frac{1}{1 + \lambda\,\|\mathbf{V}_l - \mathbf{V}_i\|} \quad .$$

Where β is a hyperparameter controlling the strength of the repulsive forces, it is set to 0.1 throughout the experiments; this value was determined through

trial and error. The term $\mathcal{K}_{il}$ is analogous to the kernel used in NE to scale interactions according to distance in LD; here, however, it scales only the repulsive forces. This change contributes to having dynamics driven mostly by attraction. Another contributing factor is the hyperparameter λ, which controls the range of the repulsive forces; it is set to 400 throughout all the experiments, decaying the repulsion rapidly with distance. Early experiments show that the value for λ can impact the performances, and a monotonic relationship between accuracy and λ hasn't been noticed. It appears critical that further works explore strategies to adapt λ according to the dataset properties (number of points, number of labels), and to the other hyperparameters (number of neighbours of neighbours, strength of repulsion β).

Contrary to NE, this method doesn't necessitate a choice on the dimensionality of the embedding space, as this dimensionality is equal to the number of classes. In theory, instead of using one-hot encoded vectors for the label directions, one could use vectors with equal pairwise angles, allowing to reduce the dimensionality of the LD space by 1 (in 2-dimensions, we can build an equilateral triangle, in 3-dimensions a regular tetrahedron, etc.). Another vanishing hyperparameter is the initialisation, as the points without a revealed label are all initialised to the origin, and the others to their corresponding one-hot vector. The computational complexity of the method is $\mathcal{O}(N(K^2)M)$, with N the number of points, K the number of neighbours, and M the dimensionality of the label vectors, which is equal to the number of classes.

The one-shot accuracy of the completed method of the restricted MNIST scenario is shown in Figure 3.7, the accuracy has been raised to 97%. A clear correlation appears when comparing the distribution of errors to that of entropies across the NE, showing potential for active learning scenarios where the model choses which supervised information is needed next, allowing label efficient learning. It is worth noticing that the high entropies are mostly located at the boundaries of the clusters identified by NE, hinting that the proposed label diffusion method is indeed sharing principles with NE in practice. The bottom view of Figure 3.7 shows that the method appears to be robust to class imbalance, as increasing the number of revealed labels for one class doesn't appear to impact negatively the accuracy.

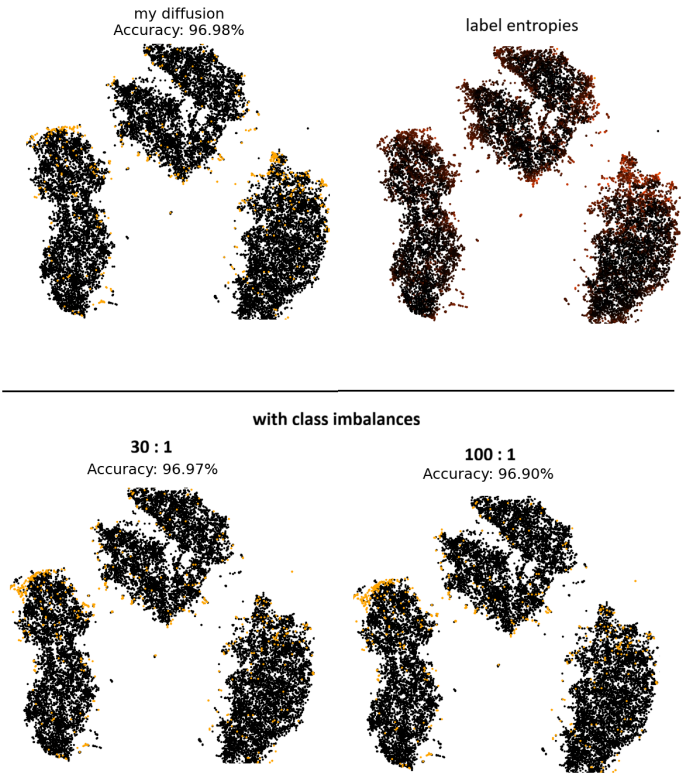


Figure 3.7: Errors and distribution of uncertainty across a NE of the restricted MNIST dataset, using the proposed method in its final form. Bottom view: the label "3" has 30 (left) or 100 (right) labelled prototypes, the other classes keep their lone labelled prototype.

The next section explores further results and brings comparisons to existing methods.

3.8.4 Results

As mentioned earlier, the horizon of methods allowing classification from few labels is vast and also diverse. Since this method has only been finalised recently, a thorough comparison to the state-of-the-art is yet to be performed; this section presents preliminary results comparing the proposed method to some existing approaches. The hyperparameters of the method are kept to their default values throughout this section: perplexity

is set to 7, $\alpha = 0.9$, $\beta = 0.1$ and $\lambda = 400$. While a rigorous approach would require a careful selection of the learning rate dt from the properties of the graphs to prevent oscillations, the implementation used here sets it to 0.2 and decays it to 0.01 over 18000 iterations. One exception is made for the COIL-20 dataset, as will be detailed later.

The first comparison will look at one-shot classification accuracies on the full MNIST dataset. The proposed *Mexican hat classifier* is designed for a transductive setting; this setting involves no train/test separation, many unlabelled points and few labelled points, and does not involve pre-training on a disjoint set of classes. Some of the compared methods are not designed specifically for this setting, but they all share in common the objective of classifying with very little labelled data.
Matching networks [107] and *Siamese networks* [108] are two deep learning approaches developed to tackle few-shot learning scenarios. When evaluated on MNIST with 1 labelled example per class, these works report accuracies of 72% and 70% respectively.
In [109], the authors propose a method which classifies images based on a learned dictionary of image patches; for instance, a small horizontal bar can be learned from MNIST as it is a recurring pattern in the numbers "2" and "5", they report 69% accuracy in one-shot on MNIST.
Another work recently published in *Nature artificial intelligence* called *Learning from one and only one shot* [110] proposes an original representation learning approach. In short, a representation is built by looking at how similarly two images can be elastically deformed into one another, for instance, a "1" can be deformed into a "7" by "pulling" the top pixels to the left, but deforming a "1" into a "2" would require more complex transformations, putting them further apart in the learned representation. Approximations are made to allow non-homeomorphic objects to be compared: the method allows for instance a 1 to loop on itself to almost form a 0. In this paper, the authors do not report an experiment on MNIST directly comparable to the one-shot transductive learning setting, because they use a subset of MNIST where all encountered points have a known label during training, however, they heavily restrict the number of points in total. Our transductive learning setting uses less label information as they do but (much) more unlabelled information, while they use a limited but equal amount of labelled and unlabelled data. Nevertheless, their report is informative for the sake of comparison; they plot the accuracy obtained on MNIST as a function of the number of labelled examples per class, reach-

ing 90% accuracy with 4 to 6 labelled examples per class.
Another work, this one perfectly comparable to the current setting, is *Nonlinear Higher-Order Label Spreading* [111]. There, the label spreading algorithm is extended to hyper-graphs, allowing higher order relationships between points to be explicitly taken into account, which has a computational cost. On MNIST, they report an accuracy a bit under 92% when revealing 6 labels per class on a PCA representation of MNIST.

The proposed Mexican hat classifier reaches an accuracy of 88.8% when revealing only 1 label per class on MNIST; the data is represented using the same number of principal components as in [111]. The accuracy raises to 95.3% when revealing 5 labels per class, dividing the error of [111] by two while using less labelled instances. The accuracies are reported as the mean of 30 runs using different random seeds to select the labelled prototypes. The difference in one shot accuracy between the full MNIST dataset and the restricted MNIST example, which was specifically designed to be challenging, could be attributed to the hyperparameter choices. This further motivates the necessity to automate the selection of hyperparameters in future works.
It is particularly encouraging to notice that the old fashioned machine learning principles used in the Mexican hat classifier suggests competitive or perhaps even better performances than recent representation learning approaches specifically designed for few-shot learning, without tuning the hyperparameters. While the audaciously named paper *Rethinking Few-Shot Image Classification: A Good Embedding is All You Need?* [96] does show the importance of having a good feature extractor, the Mexican hat classifier's performance using only PCA for feature extraction might bring nuance to the statement.

In terms of effective scaling, the Mexican hat classifier took a couple of minutes when running on the MNIST dataset using a laptop GPU, it is implemented in the low level language CUDA, to exploit the intrinsically parallel nature of the diffusion process.
It is interesting to note that applying a KNN classifier directly on a 2-dimensional NE of MNIST reaches an accuracy of 95% in one-shot (this was reported in Table 1.1), which is still above the Mexican hat classifier. Apart from the good hopes of having higher accuracies with hyperparameter tuning, applying a transductive learning method directly to the HD space can be more advantageous in practice. First of all, an initial reduction

to LD with NE requires additional choices of hyperparameters for both the NE and the KNN. Second, and more importantly, working directly in HD can open the potential to integrate the diffusion to other ML algorithms, for instance using it to regularise the training of a neural network, or to derive gradients.

The following tables compare the Mexican hat classifier to other methods suited for transductive few shot learning, in several settings. The methods featured in the comparisons are: label spreading using 3 values for K ($K = 7$, $K = 17$, and $K = 27$), an Euclidean KNN classifier using $K = 1$, a geodesic KNN classifier using $K = 1$ for the classification from neighbours and $K_{\text{graph}} = 7$, $K_{\text{graph}} = 17$, and $K_{\text{graph}} = 27$ for the graph construction, a support vector classifier in the self-supervised setting proposed by scikit-learn [51], using default hyperparameters, and a random forest classifier in the self-supervised setting proposed by scikit-learn, using default hyperparameters.
These results confirm the good performance of the Mexican hat classifier in various settings and overall robustness to class imbalance, with some exceptions like the performance on the *Anuran* dataset.

Another potential application for this method not featured here is data visualisation. Since the propagation happens in HD, it could be used as a tool with which the user can explore the data, introducing the possibility to superimpose task-driven information on top of the unsupervised structure extraction performed by NE.
For instance, in an interactive software, the user might be visually exploring a NE of the data; on the side, a view could offer a selection of colours, which will correspond to classes from the point of view of the Mexican hat label propagation. The user could then click on a colour and then in the embedding, revealing the "label" of the selected points, launching a diffusion. Then, another colour could be selected, and other points painted. The user could then see in real time how the colours diffuse through the data, helping understand the relationships between points and clusters relevant to the structures being painted by the user. For instance, in Table 3.6, two different labels are considered in the mouse brain dataset, the Mexican hat diffusion performs well in both cases; this shows that choosing different points from which to diffuse can allow the process to reveal different structures, relevant to the selection.

Method	1-shot	5-shot	unbalanced
KNN: Euclidean	0.7043 ± 0.0741	0.8933 ± 0.0277	$\mathbf{0.8001} \pm 0.0455$
KNN: geodesic K=7	0.1002 ± 0.003	0.4283 ± 0.4240	0.2014 ± 0.2876
KNN: geodesic K=17	0.1006 ± 0.003	0.4269 ± 0.4211	0.1997 ± 0.2883
KNN: geodesic K=27	$\mathbf{0.7468} \pm 0.0671$	$\mathbf{0.8972} \pm 0.0272$	0.7989 ± 0.0667
SelfTraining: SVC	0.7048 ± 0.0749	0.8676 ± 0.0167	0.2204 ± 0.1110
SelfTraining: RF	0.4090 ± 0.0970	0.6045 ± 0.1344	0.2047 ± 0.1523
Label spreading K=7	0.2633 ± 0.0832	0.6201 ± 1099	0.4352 ± 0.1019
Label spreading K=17	0.5028 ± 0.1510	0.8495 ± 0.0540	0.7136 ± 0.0395
Label spreading K=27	0.6158 ± 0.1380	0.8881 ± 0.0292	0.7656 ± 0.0282
Mexican hat	0.7044 ± 0.0563	0.8586 ± 0.0301	0.7761 ± 0.0581

Table 3.1: Accuracy of the tested methods on the *Anuran* dataset, in different settings. The reported mean values and deviations are computed from 10 trials. All methods were confronted to the same trials, allowing fair comparison. In the unbalanced scenario, 50% of the classes had 1 point of known label, and the other half of the classes had 1 to 8 points of known label, chosen randomly.

Method	1-shot	5-shot	unbalanced
KNN: Euclidean	0.6640 ± 0.0725	0.7409 ± 0.0432	0.6459 ± 0.0438
KNN: geodesic K=7	0.7313 ± 0.0501	0.7590 ± 0.0427	0.6944 ± 0.0380
KNN: geodesic K=17	0.7332 ± 0.0542	0.7660 ± 0.0389	0.6898 ± 0.0392
KNN: geodesic K=27	0.7284 ± 0.0515	0.7610 ± 0.0416	0.6884 ± 0.0368
SelfTraining: SVC	0.6640 ± 0.0725	0.7938 ± 0.1607	0.4337 ± 0.0559
SelfTraining: RF	0.5137 ± 0.0822	0.4833 ± 0.0726	0.3272 ± 0.0701
Label spreading K=7	0.5639 ± 0.1087	0.7123 ± 0.0536	0.5830 ± 0.0646
Label spreading K=17	0.6651 ± 0.0870	0.7654 ± 0.0347	0.6736 ± 0.0623
Label spreading K=27	0.6976 ± 0.0984	0.7799 ± 0.0344	0.6779 ± 0.0567
Mexican hat	$\mathbf{0.8966} \pm 0.0105$	$\mathbf{0.9006} \pm 0.0088$	$\mathbf{0.8666} \pm 0.0504$

Table 3.2: Accuracy of the tested methods on the *blobs* dataset, in different settings.

Method	1-shot	5-shot	unbalanced
KNN: Euclidean	0.6438 ± 0.0181	0.8522 ± 0.0136	0.7159 ± 0.0206
KNN: geodesic K=7	0.8217 ± 0.0166	0.9070 ± 0.0100	0.8371 ± 0.0066
KNN: geodesic K=17	0.7361 ± 0.0212	0.8723 ± 0.0102	0.7686 ± 0.0212
KNN: geodesic K=27	0.7077 ± 0.0125	0.8593 ± 0.0137	0.7469 ± 0.0184
SelfTraining: SVC	0.6438 ± 0.0181	0.8406 ± 0.0143	0.3560 ± 0.0439
SelfTraining: RF	0.4680 ± 0.0150	0.8115 ± 0.0187	0.5259 ± 0.0423
Label spreading K=7	0.5618 ± 0.0397	0.9222 ± 0.0118	0.7115 ± 0.0215
Label spreading K=17	0.7621 ± 0.0176	0.8752 ± 0.0126	0.8061 ± 0.0290
Label spreading K=27	0.7351 ± 0.0164	0.8465 ± 0.0095	0.7672 ± 0.0223
Mexican hat	0.9002 ± 0.0066	$\mathbf{0.9413} \pm 0.0141$	$\mathbf{0.9082} \pm 0.0116$
Mexican hat PP=4	$\mathbf{0.9554} \pm 0.0138$		

Table 3.3: Accuracy of the tested methods on the *COIL-20* dataset, in different settings. On this dataset, the proposed method is also evaluated using another perplexity of value 4 instead of 7, showing that hyperparameter tuning can bring better results.

Method	1-shot	5-shot	unbalanced
KNN: Euclidean	0.6379 ± 0.0614	0.8531 ± 0.0276	0.7305 ± 0.0310
KNN: geodesic K=7	0.8025 ± 0.0589	0.9123 ± 0.0303	0.8658 ± 0.0241
KNN: geodesic K=17	0.7703 ± 0.0564	0.8929 ± 0.0286	0.8266 ± 0.0273
KNN: geodesic K=27	0.7557 ± 0.0644	0.8806 ± 0.0288	0.8157 ± 0.0288
SelfTraining: SVC	0.6380 ± 0.0612	0.8344 ± 0.0359	0.3517 ± 0.0402
SelfTraining: RF	0.5457 ± 0.0557	0.6812 ± 0.0385	0.2528 ± 0.0590
Label spreading K=7	0.7110 ± 0.0894	0.9104 ± 0.0241	0.8232 ± 0.0468
Label spreading K=17	0.7701 ± 0.0634	0.9049 ± 0.0274	0.8097 ± 0.0477
Label spreading K=27	0.7654 ± 0.0515	0.8961 ± 0.0308	0.8044 ± 0.0328
Mexican hat	$\mathbf{0.8889} \pm 0.0406$	$\mathbf{0.9449} \pm 0.0251$	$\mathbf{0.9172} \pm 0.0283$

Table 3.4: Accuracy of the tested methods on the *digits* dataset, in different settings. Digits is a smaller alternative of MNIST featuring 8-by-8 images.

Method	value
KNN: Euclidean	0.7142 ± 0.0955
KNN: geodesic K=7	0.7984 ± 0.0805
KNN: geodesic K=17	0.7984 ± 0.0819
KNN: geodesic K=27	0.8030 ± 0.0812
SelfTraining: SVC	0.7142 ± 0.0955
SelfTraining: RF	0.4551 ± 0.0684
Label spreading K=7	0.4790 ± 0.1665
Label spreading K=17	0.6980 ± 0.1311
Label spreading K=27	0.7512 ± 0.1205
Mexican hat	$\mathbf{0.8682} \pm 0.0867$

Table 3.5: Accuracy of the tested methods in a 5-way, one-shot setting on the *miniImageNet* dataset with features extracted by a ResNet architecture [112] trained by the MoCo [97] self-supervised scheme. The miniImageNet dataset is a downsampled version of ImageNet, keeping the high complexity of natural images, but requiring less computational resources to use.

Method	cell type	brain region
KNN: Euclidean	0.7624 ± 0.0141	0.6227 ± 0.0074
KNN: geodesic K=7	0.7564 ± 0.0168	0.5028 ± 0.0030
KNN: geodesic K=17	0.7618 ± 0.0145	0.4999 ± 0.0044
KNN: geodesic K=27	0.7617 ± 0.0130	0.4989 ± 0.0017
SelfTraining: SVC	$\mathbf{0.8006} \pm 0.0289$	0.5843 ± 0.1027
SelfTraining: RF	0.7567 ± 0.0343	0.5745 ± 0.0377
Label spreading K=7	0.7276 ± 0.0252	0.4669 ± 0.0379
Label spreading K=17	0.7568 ± 0.0201	0.5646 ± 0.0452
Label spreading K=27	0.7683 ± 0.0199	0.5780 ± 0.0226
Mexican hat	0.7922 ± 0.0239	$\mathbf{0.6386} \pm 0.0475$

Table 3.6: Accuracy of the tested methods on the *mouse brain* dataset in a one-shot setting, using cell type and brain region as labels.

Chapter 4

Discussion and Perspectives

This section serves as a conclusion to the thesis, reflecting on the insights gained throughout the 4 years of research on NE methods, and discussing the potential future directions of the field. Section 4.1 brings additional insights and intuitions that are too general to fit elsewhere, focusing on the global picture. Section 4.2 closes the thesis with a succinct summary of the contributions and a discussion on future research directions.

4.1 General insights

There seems to be a trend to apply NE methods to larger and larger datasets, with new limits often overtly publicised. Arguments can be made that this tendency does not make much sense in the current state of NE, and resembles more a GPU-measuring contest than a meaningful scientific endeavour.
Indeed, as the number of points increases, the difficulty of the embedding task can only increase or remain the same, as adding points cannot reduce the intrinsic dimensionality of the structures. Moreover, when embedding millions or even more points, can the human interpreter really make sense of the intricacies of the embedding? There, wouldn't the global picture be more meaningful than the details, which NE specialise in and which are defined by small groups of points? This point is further amplified by the fact that NE methods tend to lose their capacity to preserve intricacies as the number of points increases. This was witnessed in Annex 5.1, where the 1.2 million points ImageNet are embedded into 2 dimensions with FUnc-SNE and then a subsample of only the first 30 classes, totalling 39000 points, is embedded again with the same method, revealing details that were not apparent in the full dataset embedding.
This tendency of NE to lose details when the number of points increases is likely due to an internal mechanism of NE rather than the increased intrinsic dimensionality of the data, since the local structures described by the first 30 classes of ImageNet likely don't dramatically change if the points are taken as a subsample or within the full dataset. One mechanism that might be the cause of the loss of details comes from the far ranging nature of the repulsive forces in NE. While the repulsive forces of NE as considered by methods descending from SNE are normalised by the sum of pairwise repulsive interactions, the actual nature of the repulsions taking effect on a point changes when N increases. Indeed, when considering that LD similarities are defined with heavy tails along with the fact that a volume scales super-linearly with length, one realises that the repulsion affecting a given point is more and more the result of coarse, large ranging interactions than from local, detail-defining interactions. In the current state of NE, this might suggest that as N increases, the methods that precisely model local repulsive interactions such as FIt-SNE, BH-t-SNE, and FUnc-SNE might lose some of their advantage over the coarser approximations used for instance by UMAP and LargeVis.

If justified demands for very large embeddings were to arise, a redesign of the shape of repulsions might be necessary to maintain the capacity of NE to preserve local details by diminishing or perhaps even zeroing out the far ranging repulsive interactions. There seems to be however a challenge to overcome, since it is precisely the heavy tails of the LD similarities that is believed to allow NE to so effectively preserve local structures. Perhaps this capacity, which manifests with high kernel tails, is actually not originating from the tails but from the shape of the LD similarity distribution at near and medium ranges, which happens to correlate with tail heaviness when defined as they usually are? This consideration aligns with the observation that changing the heaviness of the tails has an effect on the local structures by splitting groups of points "along the cracks" of the probability distribution in HD, as was shown in the FUnc-SNE paper [42] and re-pictured in Figure 4.1. It is unlikely that this splitting of masses of points originates from far-ranging interactions, since these interactions would tend to apply forces that are more or less aligned throughout the mass of points.

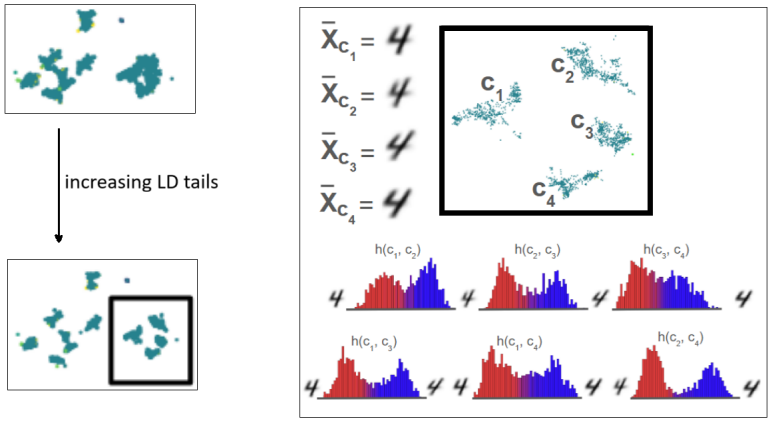


Figure 4.1: Effect of kernel tails on a cluster of number "4" points in a NE of MNIST, figure taken from the FUnc-SNE paper. Increasing the tails of the kernel further splits the cluster into its subcomponents. On the right panel, the histograms show the distribution of points in HD from pairs of clusters when the points are projected onto the line connecting their centroids. The subcomponents appear to have been split along "cracks", where the points were less sampled in HD.

This figure and the observations made in the *FUnc-SNE* paper and in the *beyond visualisation* paper reveal the crucial role that the LD distribution shape has on the structures being extracted by NE. This has already

been observed in other works which inspired the conception of FUnc-SNE [11, 12]; the importance of the attraction-repulsion balance has also been put to light in publications like [54]. However, using FUnc-SNE interactively reveals that the spectrum of tail heaviness in LD and the attraction-repulsion balance might be better understood when taken together, as a two-dimensional plane of possible configurations. Since the non-linear dynamics of NE quickly overwhelm simple intuitions on the direct effect of the hyperparameters, perhaps the phenomenon is best approached through analogies and empiricism.
Figure 4.2 proposes an comparison between the effect of the pair of hyperparameters controlling the LD kernel shape and the attraction-repulsion balance in NE, with the effect of kernel shape and of a subsequent thresholding in a 1-dimensional convolution. On the left side, a 2-dimensional dataset is composed of aligned Gaussian balls, with a hierarchical arrangement obtained by a binary-tree like structure. They are shown on the very top of the column as small black dots. Just below these dots, another image shows what the data looks like when fed to the NE algorithm: the finest grained level of the hierarchy is not discernible to the eye. From top to bottom in the left column, multiple NE of the data are shown. The first three embeddings are produced by increasing the heaviness of the LD kernel tails, each one distinguishing a finer level of the hierarchy. The final embedding increases the repulsion strength, allowing the finest grained level of the hierarchy to be extracted. Moving only one of the hyperparameters in an interactive setting does not allow the extraction of the full hierarchy.
On the right side of the figure, the same dataset in philosophy is considered, but using a 1-dimensional signal with impulsions happening along the same hierarchical arrangement as the centres of the Gaussian balls of the left side. Convolutions with different kernels are performed, and a binary output is shown in orange, obtained from thresholding of the convolution results. All the kernels are obtained by adding a shared positive Gaussian, and another negatively signed Gaussian which can vary in bandwidth; the resulting kernel is re-scaled such that the peak has a constant value throughout the experiments. The topmost result shows that a (almost) purely positive kernel only captures the coarser levels of the hierarchy with its continuous output ("convolved signal" in the figure). Finer levels of the hierarchy can be captured in the continuous outputs (2^{nd} and 3^{rd} rows) when the Mexican-hat like shape gets more and more pronounced, effectively sharpening the continuous output. However, it is only when moving the threshold that the finest grained level of the hierar-

chy is extracted in the binary output.
While this comparison has its limits, it might help understand more intuitively why considering these two hyperparameters together is more judicious than independently: one appears to be mostly controlling the regions capable of being discerned (the shape of the continuous output in the convolution), while the other tends to controls what acts as a threshold that permits the separation of the "pre-cut" regions. Again, the non-linear dynamics involved there and the fact that probability distributions have a constant integral adds nuance to this hypothesis, which attempts to describe a trend more than a relation set in stone.

These considerations naturally lead to questioning what a cluster really is, the definition in effect seems leave place to interpretation. After all, the data points, which are the enumerable manifestation of a distribution laying in a continuous space, cannot be separated into distinct groups without some notion of thresholding and choice, for instance in a metric. Perhaps one can consider NE as performing clustering, albeit by yielding results in an unconventional format, and using the continuously valued hyperparameters controlling the attraction-repulsion balance and LD kernel shape as implicit thresholding mechanisms.

To conclude this section, it seems that the visual extraction of meaningful structures from HD data is fated to involve compromises and choices. When observing the data as-is in HD, through means like PCA or MDS, the structures are diffuse and largely overlapping. Making sense of these require deformations and thus taking choices, this manifests in NE by the inductive biases involving purposefully mismatched similarity distributions in HD and LD, creating tensions in a system of points which is best alleviated by ripping apart the data along failure lines.
When the purpose of structure extraction is a supervised task, the compromises and choices can be guided by a top-down signal, but in visual data exploration, the first task of NE, such guidance is unavailable, thereby necessitating interaction or multi-scale approaches to fully explore the structures present in data.

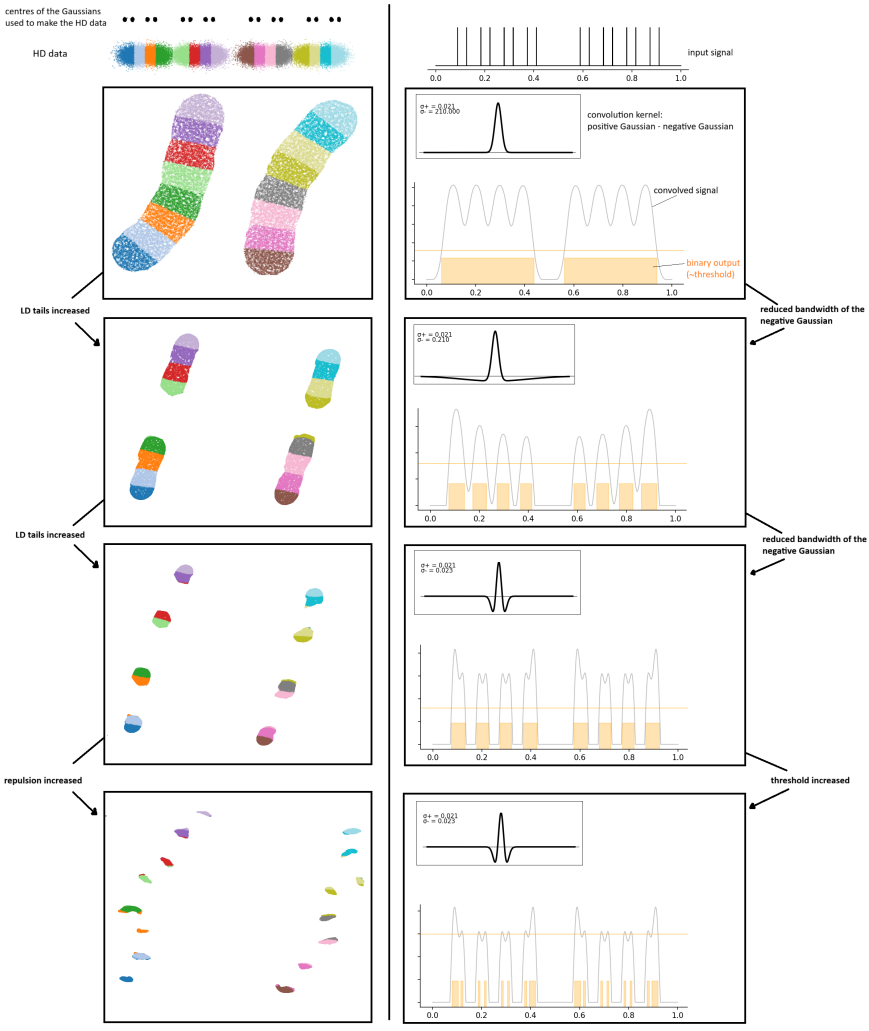


Figure 4.2: Comparison between the effect of modifying the LD kernel tails and attraction-repulsion balance in NE, with the effect of changing the kernel shape an threshold used to determine the binary output from a thresholded convolution.

4.2 Conclusion and Future Directions

In short, this thesis has contributed to the field of NE and visualisation by proposing fast and accurate methods which allow the extraction of global structures through SQuaD-MDS [64], local structures with FUnc-SNE [42], and both local and global structures with the hybrid MDS-NE method [41]. Developments focusing on the optimisation of NE methods [67] and a rethinking of how to approach NE enable FUnc-SNE to reach unparalleled levels of responsiveness, allowing interactive visualisations of large datasets and opening new applicative horizons. Accompanying tools have been developed [65] to introduce explanations to NE, or are being developed (Section 3.8) to bring further interaction through colour propagation. Realisation that NE's potential are not limited to visualisation led to works exploring new applications, in the hope to inspire new research directions [69] which could lead to consequential improvements in ML tasks, as demonstrated by the Mexican hat classifier. In some way, there is beauty in the idea that NE, which were brought to the visualisation field by ML, have matured there and are now ready to bring back insights to ML.

Machine learning researchers seem to be modern-day Sisyphus, for there lingers a sense that more doors were opened than closed. The following paragraphs pose some considerations which are sometimes very prospective and speculative, but seem worthy of rumination.

The hybrid MDS-NE method empirically suggests that multi-scale NE can be achieved with good quality and a linear complexity in N. However, if considering multi-scale neighbourhood preservation as a variant of a sorting problem, then a $\log N$ factor is expected. Is this factor hidden in the number of iterations needed for convergence, but not yet manifest in practice due to the limited size of datasets used in experiments? Does the N-to-N relationships between the points sharing a LD space constrain the "sorting" enough to effectively reduce the overall difficulty of the problem? Such questions might not see great practical implications, but pondering on these could help develop links between systems of particles in continuous space and discrete problems. The convenient positive feedback loop between NE iterations and KNN refinement exploited in FUnc-SNE could be another hint that non-obvious mechanisms are at play there.

On an even more nebulous standpoint, when observing the optimisation of NE, groups of points appear to form coherent masses moving together as ensembles rather than as individual points. These masses of points are rendered rigid by their configurations which locally minimise the energy of the densely coupled system of points. Smaller groups moving with relative high velocity towards such masses are repelled and get deformed by a cushioned impact; the smaller group then either gets assimilated into the larger mass, or reforms and flies away on another trajectory. If assimilated, the larger group either largely retains its shape, or finds a new stable configuration; if bouncing off, the larger group is generally unaffected by the interaction. On rare occasions, the impact causes a noticeable re-arrangement of the larger mass, as the projectile provided enough energy to overcome a barrier between two stable configurations.
Intuitively, this kind of behaviour seems desirable in internal representations of learning systems, as it would allow consolidated concepts to resist perturbations from newer, less established concepts. Finding a way to transpose some of the properties of NE to loss functions or perhaps local regularisations in deep learning could be an interesting research direction.
Machine learning algorithms can be placed on a spectrum reflecting a trade-off between data transformation and memory. For instance, a KNN approach is in the memory extreme, while a model based on cellular automata would be on the other extreme. Most deep learning approaches tend to be on the transformation side of the spectrum, local regularisations based on attraction-repulsion dynamics inspired by NE could add a memory component helping fight catastrophic forgetting and allowing an easy way to estimate surprise, a key component in continual learning and active inference.
If applying principles of NE to deep learning, insights from NE might also help devise new contrastive learning losses, which share similarities with negative sampling based NE by the way positive pairs are encouraged to get closer, and negative pairs are pushed apart.

Another direction worthy of consideration is to explore just how much NE methods can compress information. The paper of Section 3.7 represents what is initially a 1280-dimensional representation of the ImageNet dataset into only 32-dimensions, the embedding seems to allow good generalisation. Annex 5.1 shows that this dataset might indeed express low enough intrinsic dimensionality to be represented in such a space. Compressing all the information of the data would require a multi-scale approach, but

if heavy compression is possible without losing information, perhaps deep learning models could benefit from NE in this standpoint as well.
However, it is possible that deep learning models would struggle on such low dimensional, non-redundant representations, as they often rely on expanding the data into a very high dimensional space to disentangle structures. However, compact representations driven be NE-like loss functions might have their uses as regularising bottlenecks or as parallel specialised representations after bifurcations within the networks.

On a more practical standpoint, most published applications of NE on the monitoring of industrial processes focus on static datasets: the data is collected during a process, and then, as a subsequent step, visualised using NE. However, the theoretical capability of FUnc-SNE to process data streams without overhead might open new opportunities for online monitoring. One could imagine a system where sensors continuously feed data to a online NE visualisation, allowing operators to detect anomalous movements in the process in real-time. In this example, one point would correspond to one sensor, but one could also consider a scenario where one point would represent a full observation from multiple sensors, and where embeddings would feature historical points. This would allow the tracking of the process over time and possibly detect drifts or anomalies, be it an anomalous instantaneous state, or an anomalous movement of the state. Acolytes of Nostradamus would predict that monitoring of automated processes will become increasingly important in the future, in part due to the advent of what is sometimes called "AI agents", referring to large language models given the capacity to call tools. If this prophecy comes true, the adoption of NE for online monitoring purposes might become all the more important, as the role of human operators would shift from direct control to supervision. Hopefully, our ingrained need to understand and make sense of our environment will prevent us from devolving into mere button-pushers, and NE could be one of the tools that help us maintain a meaningful connection to the processes we oversee.

The Mexican hat diffusion developed in this thesis weights its repulsive force by a function of distance in the LD space, as NE do, but also from relationships in HD space. In the context of label diffusion, this formulation allows the disentanglement of cloudy HD structures as NE does. One cannot help but wonder if adding information from the HD side of the data to repulsions could further improve the discerning power of NE methods

too. Perhaps such information could help alleviate the inaccuracies in the estimation of repulsions in the fast methods.

Multi-scale neighbour embeddings are shown to preserve both local an global structures in data, where the notion of locality is defined on rank based relationships in some methods, or on both ranks and distances in the hybrid MDS-NE approach. Perhaps another notion of scale could reflect coordinates in a graph plotting the attraction-repulsion balance on one axis, and the LD kernel tailed-ness on another axis. From there, one could imagine another take on multi-scale NE, where the different clusters that are extracted when increasing the kernel tails and repulsion during interactive exploration could be combined into a single embedding, showing a hierarchy of structures directly.

Bibliography

[1] D H Hubel and T N Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *J Physiol*, 160(1):106–154, January 1962.

[2] Bruno A Olshausen and David J Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, June 1996.

[3] Daniel L. K. Yamins, Ha Hong, Charles F. Cadieu, Ethan A. Solomon, Darren Seibert, and James J. DiCarlo. Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the National Academy of Sciences*, 111(23):8619–8624, 2014.

[4] Maximilian Riesenhuber and Tomaso Poggio. Hierarchical models of object recognition in cortex. *Nature Neuroscience*, 2(11):1019–1025, November 1999.

[5] Jakob Hohwy. *The Predictive Mind*. Oxford University Press, 11 2013.

[6] Geoffrey E Hinton and Sam Roweis. Stochastic neighbor embedding. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems*, volume 15. MIT Press, 2002.

[7] Tasic et. al. Shared and distinct transcriptomic cell types across neocortical areas. *Nature*, 563(7729):72–78, November 2018.

[8] Dmitry Kobak and Philipp Berens. The art of using t-SNE for single-cell transcriptomics. *Nature Communications*, 10(1):5416, November 2019.

[9] Karl Pearson. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[10] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using *t*-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.

[11] Motoshi Abe, Junichi Miyao, and Takio Kurita. q-SNE: Visualizing Data using q-Gaussian Distributed Stochastic Neighbor Embedding. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 1051–1058, Los Alamitos, CA, USA, January 2021. IEEE Computer Society.

[12] Dmitry Kobak, George Linderman, Stefan Steinerberger, Yuval Kluger, and Philipp Berens. Heavy-tailed kernels reveal a finer cluster structure in t-SNE visualisations. *Mach Learn Knowl Discov Databases*, 11906:124–139, April 2020.

[13] El-Ad David Amir, Kara L Davis, Michelle D Tadmor, Erin F Simonds, Jacob H Levine, Sean C Bendall, Daniel K Shenfeld, Smita Krishnaswamy, Garry P Nolan, and Dana Pe'er. viSNE enables visualization of high dimensional single-cell data and reveals phenotypic heterogeneity of leukemia. *Nat. Biotechnol.*, 31(6):545–552, June 2013.

[14] Steffen Schneider, Jin Hwa Lee, and Mackenzie Weygandt Mathis. Learnable latent embeddings for joint behavioural and neural analysis. *Nature*, 617(7960):360–368, May 2023.

[15] Yuhan Hao, Stephanie Hao, Erica Andersen-Nissen, William M Mauck, 3rd, Shiwei Zheng, Andrew Butler, Maddie J Lee, Aaron J Wilk, Charlotte Darby, Michael Zager, Paul Hoffman, Marlon Stoeckius, Efthymia Papalexi, Eleni P Mimitou, Jaison Jain, Avi Srivastava, Tim Stuart, Lamar M Fleming, Bertrand Yeung, Angela J Rogers, Juliana M McElrath, Catherine A Blish, Raphael Gottardo, Peter Smibert, and Rahul Satija. Integrated analysis of multimodal single-cell data. *Cell*, 184(13):3573–3587.e29, May 2021.

[16] Ashwin Narayan, Bonnie Berger, and Hyunghoon Cho. Assessing single-cell transcriptomic variability through density-preserving data visualization. *Nature Biotechnology*, 39(6):765–774, June 2021.

[17] Haiyang Huang, Yingfan Wang, Cynthia Rudin, and Edward P Browne. Towards a comprehensive evaluation of dimension reduction methods for transcriptomic data visualization. *Communications Biology*, 5(1):719, July 2022.

[18] Kaiwen Wang, Yuqiu Yang, Fangjiang Wu, Bing Song, Xinlei Wang, and Tao Wang. Comparative analysis of dimension reduction methods for cytometry by time-of-flight data. *Nature Communications*, 14(1):1836, April 2023.

[19] Mikhail Andronov, Maxim V Fedorov, and Sergey Sosnin. Exploring chemical reaction space with reaction difference fingerprints and parametric t-SNE. *ACS Omega*, 6(45):30743–30751, November 2021.

[20] Francesco Trozzi, Xinlei Wang, and Peng Tao. UMAP as a dimensionality reduction tool for molecular dynamics simulations of biomacromolecules: A comparison study. *J Phys Chem B*, 125(19):5022–5034, May 2021.

[21] Christopher R. Hoyt and Art B. Owen. Probing neural networks with t-sne, class-specific projections and a guided tour. *ArXiv*, abs/2107.12547, 2021.

[22] Runyu Jing, Li Xue, Menglong Li, Lezheng Yu, and Jiesi Luo. layerumap: A tool for visualizing and understanding deep learning models in biological sequence classification using umap. *iScience*, 25(12):105530, 2022.

[23] Scott Gigante, Smita Krishnaswamy, {Adam S.} Charles, and Gal Mishne. Visualizing the phate of neural networks. *Advances in Neural Information Processing Systems*, 32, 2019. Publisher Copyright: © 2019 Neural information processing systems foundation. All rights reserved.; 33rd Annual Conference on Neural Information Processing Systems, NeurIPS 2019 ; Conference date: 08-12-2019 Through 14-12-2019.

[24] Jeremie Bogaert, Jean Emmanuel, Cyril de Bodt, and François-Xavier Standaert. Fine-tuning is not (always) overfitting artifacts. In *ESANN 2023*, pages 157–162, 01 2023.

[25] Shakeel Ahmed, Nimra Waqar, Asad Raza, Amaad Khalil, and Muhammad Adil. Cheese net: A feature-optimized hybrid learning

model. *International Journal of Innovations in Science and Technology*, 7:218–231, 05 2025.

[26] Wenhui Song, Xin Zhang, Guozhu Yang, Yijin Chen, Lianchao Wang, and Hanghang Xu. A study on dimensionality reduction and parameters for hyperspectral imagery based on manifold learning. *Sensors*, 24(7), 2024.

[27] Tulsi Patel, Mark W. Jones, and Thomas Redfern. Manifold explorer: Satellite image labelling and clustering tool with using deep convolutional autoencoders. *Algorithms*, 16(10), 2023.

[28] Xiaoyu Du, Cheng Cheng, Yujing Wang, and Zhijie Han. Research on network attack traffic detection hybridalgorithm based on umap-rf. *Algorithms*, 15(7), 2022.

[29] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, October 2001.

[30] Weipeng Lu and Xuefeng Yan. Visual high dimensional industrial process monitoring based on deep discriminant features and t-SNE. *Multidimensional Systems and Signal Processing*, 32(2):767–789, April 2021.

[31] Gardiola Oktorifo, Maulud Abdulhalim Shah, Nawaz Muhammad, and Jamaludin Nabila Farhana. Enhancing recursive-pca-based process monitoring using t-sne visualization. *Materials Research Proceedings*, 53:458–463, 2025.

[32] Laurens van der Maaten. Accelerating t-sne using tree-based algorithms. *Journal of Machine Learning Research*, 15(93):3221–3245, 2014.

[33] George C Linderman, Manas Rachh, Jeremy G Hoskins, Stefan Steinerberger, and Yuval Kluger. Fast interpolation-based t-SNE for improved visualization of single-cell RNA-seq data. *Nat. Methods*, 16(3):243–245, March 2019.

[34] Andreja Bubic, D Yves von Cramon, and Ricarda I Schubotz. Prediction, cognition and the brain. *Front. Hum. Neurosci.*, 4:25, March 2010.

[35] Masataka Watanabe. *From biological to artificial consciousness*. Frontiers Collection. Springer International Publishing, Cham, 2022.

[36] Thomas Parr, Giovanni Pezzulo, and Karl J Friston. *Active inference*. The MIT Press, 2022.

[37] Justin Lin and Julia Fukuyama. Calibrating dimension reduction hyperparameters in the presence of noise. *PLoS Comput. Biol.*, 20(9):e1012427, September 2024.

[38] Quynh Nguyen Phuong VU, Paula Lago, and Sozo Inoue. Improving noise robustness of single sensor data in human activity recognition with umap and additional data. In *Proceedings of the 2022 ACM International Symposium on Wearable Computers*, ISWC '22, page 109–111, New York, NY, USA, 2022. Association for Computing Machinery.

[39] John A. Lee and Michel Verleysen. Quality assessment of dimensionality reduction: Rank-based criteria. *Neurocomputing*, 72(7):1431–1443, 2009. Advances in Machine Learning and Computational Intelligence.

[40] Leland McInnes, John Healy, and James Melville. UMAP: Uniform manifold approximation and projection for dimension reduction. February 2018.

[41] Pierre Lambert, Cyril de Bodt, Michel Verleysen, and John A Lee. SQuadMDS: A lean stochastic quartet MDS improving global structure preservation in neighbor embedding like t-SNE and UMAP. *Neurocomputing*, 503:17–27, September 2022.

[42] Pierre Lambert, Edouard Couplet, Michel Verleysen, and John Aldo Lee. FUnc-SNE: A flexible, fast, and unconstrained algorithm for neighbour embeddings. 2025.

[43] Cyril de Bodt, Alex Diaz-Papkovich, Michael Bleher, Kerstin Bunte, Corinna Coupette, Sebastian Damrich, Enrique Fita Sanmartin, Fred A Hamprecht, Emőke-Ágnes Horvát, Dhruv Kohli, Smita Krishnaswamy, John A Lee, Boudewijn P F Lelieveldt, Leland McInnes, Ian T Nabney, Maximilian Noichl, Pavlin G Poličar, Bastian Rieck, Guy Wolf, Gal Mishne, and Dmitry Kobak. Low-dimensional embeddings of high-dimensional data. 2025.

[44] Ignacio Díaz, José M Enguita, Ana González, Diego García, Abel A Cuadrado, María D Chiara, and Nuria Valdés. Morphing projections: a new visual technique for fast and interactive large-scale analysis of biomedical datasets. *Bioinformatics*, 37(11):1571–1580, 11 2020.

[45] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, page 226–231. AAAI Press, 1996.

[46] Charu C. Aggarwal, Alexander Hinneburg, and Daniel A. Keim. On the surprising behavior of distance metrics in high dimensional space. In Jan Van den Bussche and Victor Vianu, editors, *Database Theory — ICDT 2001*, pages 420–434, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[47] Damien Francois, Vincent Wertz, and Michel Verleysen. The concentration of fractional distances. *IEEE Transactions on Knowledge and Data Engineering*, 19(7):873–886, 2007.

[48] Michel Verleysen and Damien François. The curse of dimensionality in data mining and time series prediction. In Joan Cabestany, Alberto Prieto, and Francisco Sandoval, editors, *Computational Intelligence and Bioinspired Systems*, pages 758–770, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[49] Mebarka Allaoui, Mohammed Lamine Kherfi, and Abdelhakim Cheriet. Considerably improving clustering algorithms using umap dimensionality reduction technique: A comparative study. In Abderrahim El Moataz, Driss Mammass, Alamin Mansouri, and Fathallah Nouboud, editors, *Image and Signal Processing*, pages 317–325, Cham, 2020. Springer International Publishing.

[50] Clément Pealat, Guillaume Bouleux, and Vincent Cheutet. Improved time-series clustering with umap dimension reduction method. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 5658–5665, 2021.

[51] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[52] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[53] George C Linderman and Stefan Steinerberger. Clustering with t-SNE, provably. *SIAM J Math Data Sci*, 1(2):313–332, May 2019.

[54] Jan Niklas Böhm, Philipp Berens, and Dmitry Kobak. Attraction-repulsion spectrum in neighbor embeddings. *J. Mach. Learn. Res.*, 23(1), January 2022.

[55] Laurens van der Maaten. Learning a parametric embedding by preserving local structure. In David van Dyk and Max Welling, editors, *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, volume 5 of *Proceedings of Machine Learning Research*, pages 384–391, Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA, 16–18 Apr 2009. PMLR.

[56] Tim Sainburg, Leland McInnes, and Timothy Q Gentner. Parametric UMAP embeddings for representation and semisupervised learning. *Neural Comput*, 33(11):2881–2907, October 2021.

[57] David Novak, Cyril de Bodt, Pierre Lambert, John A Lee, Sofie Van Gassen, and Yvan Saeys. Interpretable models for scRNA-seq data embedding with multi-scale structure preservation. November 2023.

[58] Michael McCloskey and Neal J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation*, 24:109–165, 1989.

[59] R M French. Catastrophic forgetting in connectionist networks. *Trends Cogn. Sci.*, 3(4):128–135, April 1999.

[60] Cyril de Bodt, Dounia Mulders, Michel Verleysen, and John Aldo Lee. Fast multiscale neighbor embedding. *IEEE Trans. Neural Netw. Learn. Syst.*, 33(4):1546–1560, April 2022.

[61] John A. Lee, Diego H. Peluffo-Ordóñez, and Michel Verleysen. Multi-scale similarities in stochastic neighbour embedding: Reducing dimensionality while preserving both local and global structure.

Neurocomputing, 169:246–261, 2015. Learning for Visual Semantic Understanding in Big Data ESANN 2014 Industrial Data Processing and Analysis.

[62] Pierre Lambert, Lee John, Michel Verleysen, and Cyril de Bodt. Impact of data subsamplings in fast Multi-Scale neighbor embedding. In *ESANN 2021 proceedings*, Louvain-la-Neuve (Belgium), 2021. Ciaco - i6doc.com.

[63] Jan de Leeuw and Patrick Mair. Multidimensional scaling using majorization: SMACOF inr. *J. Stat. Softw.*, 31(3), 2009.

[64] Pierre Lambert, Cyril De Bodt, Michel Verleysen, and John Aldo Lee. Stochastic quartet approach for fast multidimensional scaling. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2021.

[65] Pierre Lambert, Rebecca Marion, Julien Albert, Emmanuel Jean, Sacha Corbugy, and Cyril De Bodt. Globally local and fast explanations of t-SNE-like nonlinear embeddings. In *31st ACM International Conference on Information and Knowledge Management (CIKM)*, 2022.

[66] Yurii Nesterov. A method for solving the convex programming problem with convergence rate $o(1/k^2)$. *Proceedingsof theUSSRAcademyof Sciences*, 269 : $543 - -547$, 1983.

[67] Pierre Lambert, Lee John, Edouard Couplet, and Cyril de Bodt. Nesterov momentum and gradient normalization to improve t-SNE convergence and neighborhood preservation, without early exaggeration. In *ESANN 2023 proceesdings*, Louvain-la-Neuve (Belgium), 2023. Ciaco - i6doc.com.

[68] Pierre Lambert, Edouard Couplet, Cyril de Bodt, and John Aldo Lee. Estimated neighbour sets and smoothed sampled global interactions are sufficient for a fast approximate t-SNE. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2024.

[69] Pierre Lambert, Edouard Couplet, Michel Verleysen, and John Aldo Lee. Neighbour Embeddings: Beyond Visualisation. In Daniel Archambault, Ian Nabney, and Jaakko Peltonen, editors, *Machine Learning Methods in Visualisation for Big Data*. The Eurographics Association, 2025.

[70] Wulfram Gerstner, Werner M Kistler, Richard Naud, and Liam Paninski. *Neuronal dynamics*. Cambridge University Press, Cambridge, England, August 2014.

[71] Rodrigo Quian Quiroga. Rodrigo quian quiroga. *Curr. Biol.*, 21(19):R798–R799, October 2011.

[72] D O Hebb. *The organization of behavior*. Psychology Press, Philadelphia, PA, May 2002.

[73] Erkki Oja. Simplified neuron model as a principal component analyzer. *J. Math. Biol.*, 15(3):267–273, November 1982.

[74] S Amari. Dynamics of pattern formation in lateral-inhibition type neural fields. *Biol. Cybern.*, 27(2):77–87, August 1977.

[75] Rüdiger W Brause. A symmetrical lateral inhibition network for PCA and feature decorrelation. In *ICANN '93*, pages 486–489. Springer London, London, 1993.

[76] J B Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, March 1964.

[77] J W Sammon. A nonlinear mapping for data structure analysis. *IEEE Trans. Comput.*, C-18(5):401–409, May 1969.

[78] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

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

[80] Rainer Storn and Kenneth Price. *J. Glob. Optim.*, 11(4):341–359, 1997.

[81] Emre O. Neftci, Hesham Mostafa, and Friedemann Zenke. Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Processing Magazine*, 36(6):51–63, 2019.

[82] Wei Dong, Charikar Moses, and Kai Li. Efficient k-nearest neighbor graph construction for generic similarity measures. In *Proceedings of the 20th international conference on World wide web*, pages 577–586, 2011.

[83] Jian Tang, Jingzhou Liu, Ming Zhang, and Qiaozhu Mei. Visualizing large-scale and high-dimensional data. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, page 287–297, Republic and Canton of Geneva, CHE, 2016. International World Wide Web Conferences Steering Committee.

[84] Sebastian Damrich, Niklas Böhm, Fred A Hamprecht, and Dmitry Kobak. From t-SNE to UMAP with contrastive learning. In *The Eleventh International Conference on Learning Representations*, 2023.

[85] Aleksandr Artemenkov and Maxim Panov. NCVis: Noise contrastive approach for scalable visualization. In *Proceedings of The Web Conference 2020*, volume 15, pages 2941–2947, New York, NY, USA, April 2020. ACM.

[86] Dimitris K Agrafiotis. Stochastic proximity embedding. *J. Comput. Chem.*, 24(10):1215–1221, July 2003.

[87] Jonathan X. Zheng, Samraat Pawar, and Dan F. M. Goodman. Graph drawing by stochastic gradient descent. *IEEE Transactions on Visualization and Computer Graphics*, 25(9):2738–2748, 2019.

[88] Edouard Couplet, Pierre Lambert, Michel Verleysen, Dounia Mulders, John Aldo Lee, and Cyril De Bodt. Natively interpretable t-SNE. In *ECML PKDD*, volume 1, page 1, 2023.

[89] Chaofan Chen, Oscar Li, Chaofan Tao, Alina Jade Barnett, Jonathan Su, and Cynthia Rudin. *This looks like that: deep learning for interpretable image recognition*. Curran Associates Inc., Red Hook, NY, USA, 2019.

[90] Jarkko Venna, Jaakko Peltonen, Kristian Nybo, Helena Aidos, and Samuel Kaski. Information retrieval perspective to nonlinear dimensionality reduction for data visualization. *Journal of Machine Learning Research*, 11(13):451–490, 2010.

[91] John A Lee, Edouard Couplet, Pierre H Lambert, Pierre Merveille, Ludovic Journaux, Dounia Mulders, Cyril de Bodt, and Michel Verleysen. Improving on early exaggeration in t -SNE: Early hierarchization better preserves global structure. *Neurocomputing*, 660(131882):131882, January 2026.

[92] Jaakko Peltonen, Wen Xu, Timo Nummenmaa, and Jyrki Nummenmaa. Fair neighbor embedding. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett,

editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 27564–27584. PMLR, 23–29 Jul 2023.

[93] Isaac Robinson and Emma Pierce-Hoffman. Tree-sne: Hierarchical clustering and visualization using t-sne. *arXiv preprint arXiv:2002.05687*, 2020.

[94] Malik Boudiaf, Imtiaz Ziko, Jérôme Rony, Jose Dolz, Pablo Piantanida, and Ismail Ben Ayed. Information maximization for few-shot learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 2445–2457. Curran Associates, Inc., 2020.

[95] Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B Tenenbaum, Hugo Larochelle, and Richard S Zemel. Meta-learning for semi-supervised few-shot classification. *arXiv preprint arXiv:1803.00676*, 2018.

[96] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B. Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: A good embedding is all you need? In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 266–282, Cham, 2020. Springer International Publishing.

[97] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[98] Dengyong Zhou, Olivier Bousquet, Thomas Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In S Thrun, L Saul, and B Schölkopf, editors, *Advances in Neural Information Processing Systems*, volume 16. MIT Press, 2003.

[99] M A Hearst, S T Dumais, E Osuna, J Platt, and B Scholkopf. Support vector machines. *IEEE Intell. Syst.*, 13(4):18–28, July 1998.

[100] Xiaojin Zhu and Zoubin Ghahramani. Learning from labeled and unlabeled data with label propagation. *ProQuest number: information to all users*, 2002.

[101] Miloš Radovanovi263;, Alexandros Nanopoulos, and Mirjana Ivanovi263;. Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research*, 11(86):2487–2531, 2010.

[102] Philip A Knight. The Sinkhorn–Knopp algorithm: Convergence and applications. *SIAM J. Matrix Anal. Appl.*, 30(1):261–275, January 2008.

[103] Francesco Tudisco, Austin R Benson, and Konstantin Prokopchik. Nonlinear higher-order label spreading. In *Proceedings of the Web Conference 2021*, New York, NY, USA, April 2021. ACM.

[104] Tal Wagner, Sudipto Guha, Shiva Kasiviswanathan, and Nina Mishra. Semi-supervised learning on data streams via temporal label propagation. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5095–5104. PMLR, 10–15 Jul 2018.

[105] Peter Doyle and J Snell. *Random walks and electric networks*. American Mathematical Society, Providence, Rhode Island, 1984.

[106] Markus Hagenbuchner and Ah Chung Tsoi. A supervised training algorithm for self-organizing maps for structures. *Pattern Recognit. Lett.*, 26(12):1874–1884, September 2005.

[107] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, koray kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[108] Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2, pages 1–30. Lille, 2015.

[109] Alex Wong and Alan Yuille. One shot learning via compositions of meaningful patches. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1197–1205, 2015.

[110] Haizi Yu, Igor Mineyev, Lav R Varshney, and James A Evans. Learning from one and only one shot. 1(1), July 2025.

[111] Francesco Tudisco, Austin R. Benson, and Konstantin Prokopchik. Nonlinear higher-order label spreading. In *Proceedings of the Web Conference*

2021, WWW '21, page 2402–2413, New York, NY, USA, 2021. Association for Computing Machinery.

[112] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[113] Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva: Exploring the limits of masked visual representation learning at scale. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19358–19369, 2023.

[114] K Fukunaga and D R Olsen. An algorithm for finding intrinsic dimensionality of data. *IEEE Trans. Comput.*, C-20(2):176–183, February 1971.

[115] Elizaveta Levina and Peter Bickel. Maximum likelihood estimation of intrinsic dimension. In L Saul, Y Weiss, and L Bottou, editors, *Advances in Neural Information Processing Systems*, volume 17. MIT Press, 2004.

[116] Elena Facco, Maria d'Errico, Alex Rodriguez, and Alessandro Laio. Estimating the intrinsic dimension of datasets by a minimal neighborhood information. *Sci. Rep.*, 7(1):12140, September 2017.

[117] Christoph Stockl, Yukun Yang, and Wolfgang Maass. Local prediction-learning in high-dimensional spaces enables neural networks to plan. *Nat. Commun.*, 15(1):2344, March 2024.

Chapter 5

Annexes

5.1 Intuition on intrinsic dimensionality

This Annex brings a brief intuition on the concept of intrinsic dimensionality (IDIM). A way to estimate it and relations with NE are discussed in Section 5.1.1, as well as the challenges that arise when estimating high IDIM values. In Section 5.1.2, the possibility to use NE to estimate IDIM is explored superficially, using the example of the EVA [113] representation of ImageNet, the dataset used and reduced to 32 dimensions in Section 3.7 using the NE method developed in Section 3.6.

5.1.1 Intuition on intrinsic dimensionality and its estimation

If measurements of the observable world would naturally be modelled by uniform, amorphous HD distributions of variables, then existence would be quite bland. Fortunately, data exhibits structure at multiple scales. There are many ways to formally describe this, but in short, some variables can be predicted from others. It is because data is structured that our brains exist in the first place (else there would be nothing to make sense of), and that machine learning is applicable to real world problems. This annex doesn't aim to bring a thorough nor a rigorous explanation on what IDIM is, rather it aims to provide some intuition on the concept, and in particular why this concept is relevant when studying NE methods.

Intrinsic dimensionality is tightly tied to this notion of structure in data as it aims to quantify how many variables are truly necessary to describe the data. For instance, if two distinct 127862368-dimensional vectors were the only observations sampled from a distribution, then the IDIM of the dataset would be 1, because the structure formed by the two points can be perfectly represented on a single axis pointing from one point to the other in the ambient 127862368-dimensional space. From this example, two upper bounds for the IDIM of a sample of N distinct points can be intuited: it cannot be larger that $N - 1$, and it cannot be larger than the ambient dimensionality which was used to describe the dataset in the first place. It is important to understand that the structures that IDIM aims to capture are the geometry described by the data points, with no interest in the values taken by the variables in HD. Without further elaborating in this direction, this means that the information necessary to determine the IDIM of a dataset is captured without loss by the Euclidean distance matrix which

MDS attempts to preserve. An additional property worth noting is that sampling new points can never reduce the IDIM of a dataset, it can only increase the IDIM or not affect it.

Estimating the IDIM of a dataset is useful to understand the complexity of the structures present in the data. In the case of data visualisation, it is a good indicator on the feasibility of representing the data in very LD spaces. Multiple approaches exist to estimate the IDIM of a dataset. In short, these can take two directions: models based on linear projections, where the IDIM is estimated by counting the number of eigenvalues larger than a certain threshold, like [114], and geometric methods as in [115,116]. An intuitive geometric approach is described by the following algorithm: Considering a point in a dataset of ambient dimensionality M around which we are interested in determining the IDIM of the data. One can grow a hypersphere centred on the point and count how many neighbours are contained within it as its radius increases, as illustrated in Figure 5.1. If the sampled points in the surrounding area were uniformly distributed in a M'-dimensional subspace of the ambient space, then the number of neighbours within the hypersphere would grow proportionally to $r^{M'}$, where r is the radius of the hypersphere, rather than growing with a rate depending on the ambient dimensionality M. By estimating how the number of neighbours grows with the radius, one can fit a curve to estimate the IDIM M' at a given scale bounded by the maximum radius of the hypersphere or by the number of neighbours after which the process is stopped. Estimators of IDIM generally rely on an assumption of a certain property in the data, here, the assumption is that within the M'-dimensional subspace, the points are uniformly distributed.

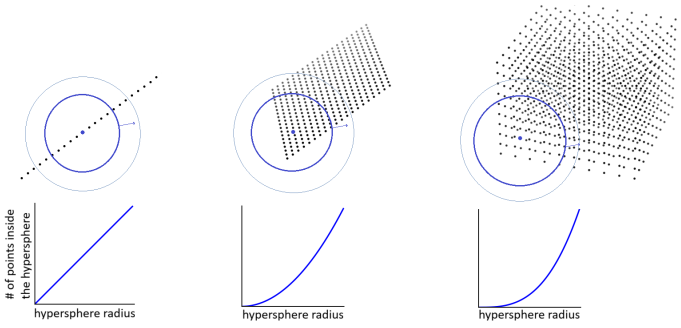
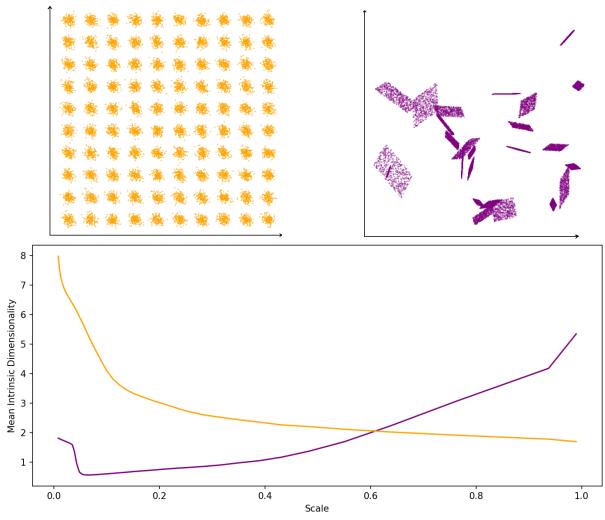


Figure 5.1: Illustration of the growing hypersphere method for estimating pointwise IDIM. For each point, a growing hypersphere centred on it grows, and the rate at which the number of neighbours within the hypersphere increases is used to estimate the IDIM.

The algorithm described above brings forth an important consideration when discussing IDIM: the notion of scale. Let us consider a dataset of ambient dimensionality 3, where the data points are densely sampled from a distribution of the shape of an uncooked fusilli lunghi, which are spiral pasta resembling very long spaghetti that would have been wrapped around a thin cylinder before drying, the shape is similar to a telephone cord.
If the growing hypersphere method was applied on a point situated in the mass of a single fusilli lunghi using a tiny radius, then the number of points captured by the sphere would grow isotropically, yielding an estimated IDIM of 3. If one was to look a little further, the locally linear structure of the fusilli lunghi would start to appear, and the estimated IDIM would drop to approach 1. Going a bit further with the hypersphere radius, the linear structure would start to curve significantly, raising the estimated IDIM back towards 2. Finally, if the radius was increased enough to encompass the entire structure, the estimated IDIM would approach 1 again, as the length of the fusilli lunghi would dominate over the small diameter of its coils.
It should be understood that the notion of scale isn't an artefact of the estimator used to measure IDIM, but rather a property of the data itself.

When using NE on actual data, it can be informative to plot the evolution of the estimated IDIM along scales, as shown in Figure 5.2. There, in the top view, a linear projection of two synthetic datasets are shown. On the left (orange), the data is designed to have a low global IDIM and a high local IDIM, whereas on the right (purple), the data is designed to have a high global IDIM and a low local IDIM. Under these two scatterplots, a plot of the mean estimated pointwise IDIM at different scales is shown for both datasets. The general allure of these two curves shows that the estimated IDIM reflects the intended design of the datasets.
When looking at these curves and at the linear projections, one can intuit which dataset is more suited for observation through NE. As explained in Section 1.2, NE methods tend to preserve local structures at the detriment of global structures. Therefore, if the global landscape of a dataset is composed of large, intrinsically HD structures, but that these structures are themselves composed of simpler LD parts, as is the case in the purple

dataset, then NE methods should be able faithfully capture some of the information contained in the data. In the opposite case, when the data is locally HD but globally LD (orange dataset), NE methods will struggle to capture any meaningful structure, since these methods don't prioritise the preservation of global structures. In that case, global methods like PCA or MDS are better suited.
This is verified in the bottom part of Figure 5.2, which shows the NE of both datasets. The points are coloured by how well they integrate in the true local structures (top row) and in the global organisation of the data (bottom row). Hot colours indicate poor quality, and the mean quality scores are indicated under each embedding. Visually, in both datasets, the separation of the data manifolds into distinct clusters is well captured, however, the organisation of these clusters in both datasets is poorly preserved, as expected and confirmed by the global quality scores which are sensibly the same across both datasets. However, when considering local structures, the purple dataset fares significantly better than the orange dataset.


Mean Intrinsic Dimensionality
Scale
0.0
0.2
0.4
0.6
0.8
1.0
1
2
3
4
5
6
7
8

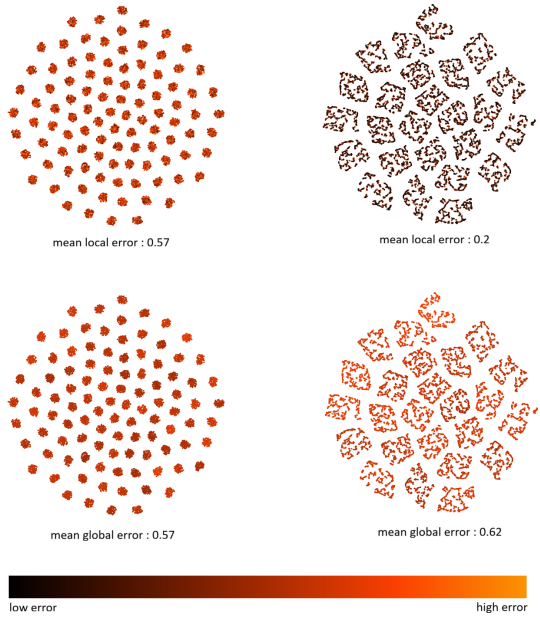

mean local error : 0.57
mean local error : 0.2
mean global error : 0.57
mean global error : 0.62
low error
high error

Figure 5.2: Illustration of the notion of scale-dependent intrinsic dimensionality.
Top view: Left and orange: 20-dimensional Gaussian blobs of 150 points each organised on a 2-dimensional grid. Top row: PCA embeddings of both datasets, bottom row: mean estimated intrinsic dimensionality at different scales. To keep computations low, the number of neighbours captured by the growing hypersphere are kept constant throughout the scales changes, but the dataset is subsampled to simulate larger scales, a principle borrowed from [60].
Bottom view: Neighbour embeddings of both datasets, keeping the left-right organisation of the top view. The points are coloured by their local error ($1 - \mathrm{Q}^i_{NX}(15)$, the Q_{NX} measure is explained atop Fig. 1.3) in the top row, and by one minus the Pearson correlation between pairwise distances in HD and LD (global structure preservation) in the bottom row.

In Figure 5.2, the local intrinsic dimensionality of the orange dataset's *distribution* is 20, and the global intrinsic dimensionality of the purple dataset's distribution is 24. However, when looking at the multi-scale IDIM curves estimated on *samples* of these distributions, one can see that the maximal IDIMs of both curves are far below their theoretical values.
The underestimations can have two origins: the first possibility is that the sampling is unlucky or too sparse to capture the complexity of the distributions, leading to truly lower IDIMs in the samples. The second possibility is that the estimator underestimates the true IDIMs regardless of the representativeness of the samples.

K	64	128	256	512	1024
IDIM	22.1 ± 3.4	21.8 ± 2.5	21.3 ± 1.8	20.7 ± 1.4	19.9 ± 1.2

Table 5.1: Intrinsic dimension estimation of $50k$ observations sampled from a Gaussian blob in 30 dimensions (the IDIM should be close to the ambient dimensionality), as a function of the number of neighbours K used for the estimation.

In Table 5.1, the IDIM of a sample of 50000 points drawn from a Gaussian blob in 30 dimensions is estimated using the growing hypersphere method, with varying numbers of neighbours K for the estimation. The scale should not significantly affect the estimation of the IDIM in this case due to the lack of structure. This experiment aims to see if having richer samples can help the estimator with the measurement. The IDIM estimations did not vary largely with the addition of points used by the estimator, indicating that the underestimations of the IDIM in Fig. 5.2 likely do not stem from a lack of representativeness in the samples, but rather from limitations of the estimator itself.

As a sanity check, the same experiment was performed with uniform distributions in 30 dimensions and with another estimator based on similar principles, Two-NN [116], both cases yielded similar results. As dimensions grow, an unintuitive phenomenon arises where data points tend to lie near the "surface" of the distribution rather than in the "bulk"; in order to remove the possibility that this phenomenon could affect the estimations, the IDIM was also estimated by growing a hypersphere from the centre of the distributions rather than from the data points themselves, but the results stayed significantly lower than the IDIM of the distribution, sometimes going as far down as 11 for very large values of K.

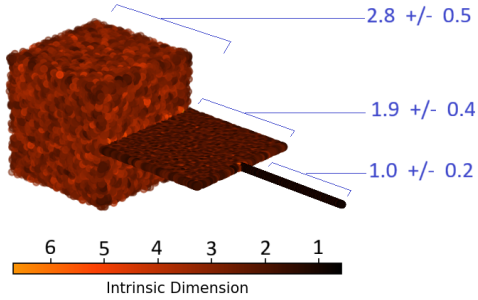


Figure 5.3: Estimated pointwise intrinsic dimensionalities on a dataset consisting of a cube, a plane, and a line, in an ambient space of 3 dimensions. The estimations are performed using the growing hypersphere method. In blue: the mean estimated intrinsic dimensionality for each of the 3 structures.

To better understand the limits of the IDIM estimator, Figure 5.3 shows the pointwise estimated IDIMs on a dataset in an ambient 3-dimensional space, with points sampled from a cube, a plane, and a line. There, as the IDIM of a sample grows, the variance of the pointwise estimations also grows, and the mean estimations there also tend to be biased downwards. Similar results were obtained with Two-NN [116]. The large variance in the estimations did not tamper down when using larger values of K, the number of neighbours used for the estimation. It is possible that the high variance in the estimations stems from instabilities when fitting the curves to estimate the IDIM, as the dimension M' appears as an exponent in the model. Table 5.2 demonstrates that the estimator can indeed be quite sensitive numerically: there, a minimal dataset of 31 equidistant points has it's IDIM evaluated under different levels of Gaussian noise. The true IDIM of the dataset is 30, but the estimated IDIM varies widely depending on the

noise level. Instabilities arise in this scenario because, at low noise levels, the growing hypersphere goes from encompassing 0 points to 30 points with a tiny change in radius.

Noise level	0.001	0.01	0.1
Intrinsic dimension ($\pm$ std)	641.1 ± 99.7	63.5 ± 9.4	9.4 ± 1.8

Table 5.2: Estimation of the intrinsic dimension of a dataset of 31 equidistant points, with an actual IDIM 30. Different levels of Gaussian noise are added to the dataset before estimating the intrinsic dimension by the growing hypersphere method using $K = 30$.

5.1.2 Can ImageNet fit in 32 dimensions?

In Section 3.7, the ImageNet dataset as represented by the EVA large vision-language model [113] is embedded to 32 dimensions using the NE method developed in Section 3.6. The embedding allows for remarkable performance and generalisability when performing a supervised task, when compared to using the raw 1280-dimensional EVA features. This section aims to give plausible intervals to the IDIM of the ImageNet dataset as represented by EVA, in order to understand if most of the structures can be captured in spaces of the order of 32 dimensions. As seen previously, estimating large IDIMs is difficult, this section explores if NE methods can be used in conjunction to traditional estimators for this purpose.

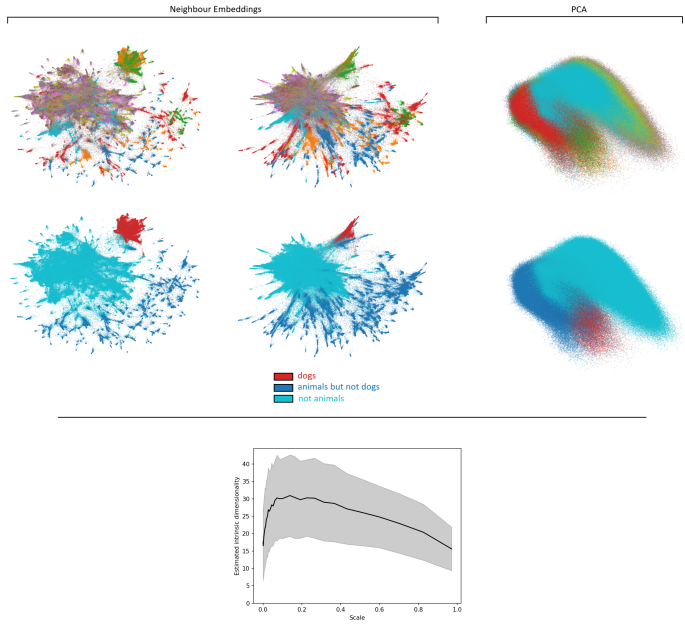


Figure 5.4: Embeddings of the ImageNet dataset as represented internally by the EVA large vision-language model [113], where each observation lies in a 1280-dimensional space. Top: Neighbour embeddings and PCA of the ImageNet dataset, the NE are built using the FUnc-SNE method developed in Section 3.6 and were built after an initial PCA reduction to 70 dimensions. The leftmost NE have sightly stronger repulsive forces and larger LD kernel tails than the central NE. Both were obtained during the same interactive visualisation session, explaining the similar global structure. The top row is coloured using all the 1000 classes, the bottom row only shows 3 artificial classes. Bottom: Estimated intrinsic dimensionality of the ImageNet dataset across all scales, performed on the 1280-dimensional EVA features [113].

Embeddings of the ImageNet dataset are shown in the top part of Figure 5.4, along with the multi-scale IDIM estimation of the dataset in the bottom part. The multi-scale IDIM estimation is performed as explained in Section 5.1.1, using the growing hypersphere method on the 1280-dimensional dataset. The NEs show sharper structures than the PCA projections, but large parts of the dataset appear amorphous in the NEs. This stems from the high complexity of the dataset across scales, as hinted by the multi-scale IDIM curve. It is interesting to note that, despite generally not capturing global structures, the NE representations exhibit some global organisation, separating dog from other animals, and animals in general from non-animals. This may be due to a bias in the dataset or the learning scheme, inducing large differences between these categories in the way EVA represents the data, in turn allowing NE to capture these nuances.

As a first attempt to estimate the IDIM of the ImageNet dataset as represented by EVA, the correspondence between the true, controlled, IDIM of synthetic HD data and the estimated IDIM using the growing hypersphere method is given in Figure 5.5. In this setting, the sub-linear growth of the estimated IDIM with the true IDIM indicates that the larger the estimation, the less certain we can be of the actual IDIM of the data. The correspondence here is likely very flawed if used on actual data exhibiting structure, as the hypersphere method is subject to numerical instability and assumes uniform distributions in the local subspaces. Nevertheless, this curve gives a first idea of how to interpret the multi-scale IDIM curve of Figure 5.4. There, if considering the 1 standard deviation bands around the means, the estimated IDIMS vary between about 6 and 42, converting these values to true IDIMs using Figure 5.5 gives IDIMs ranging from 6 to 95, quite a spread.

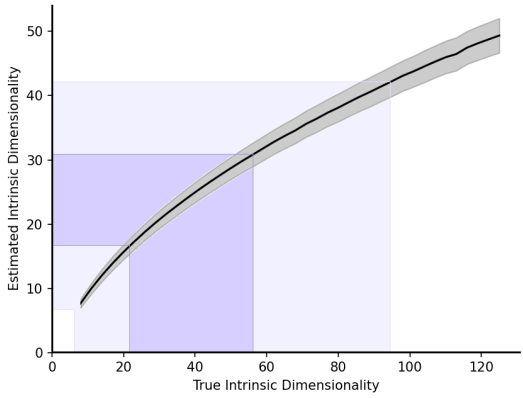


Figure 5.5: Empirical correspondence between true IDIM in HD data and estimated IDIM using the growing hypersphere method. The measures were done on synthetic datasets of uniform distributions of growing dimensionalities. The blue shades correspond to the spread of estimated IDIMs in the multi-scale curve of Fig. 5.4.

The estimated IDIMS obtained from the multi-scale IDIM curve of Figure 5.4 and the correspondence curve suggests that the 1280-dimensional EVA representation of ImageNet is redundant to some extent, but such a large spread makes it difficult to draw conclusions. Intuitively, given how expressive a 90-dimensional space is, it seems likely that the IDIM of the dataset is in the lower range of the spread of estimated values.
Let us attempt to tighten these bounds, but this time focusing on a subset of

the ImageNet dataset, for convenience. Lessons learned on a subset might bring insights on the full dataset later on. The first 30 classes of ImageNet are selected, for a total of 39000 points. If the NE of the full dataset shown in Figure 5.4 are of any indication, the points sharing a label should more or less cluster together, meaning that the local structures in the isolated subsample should not be too different from the local structures drawn by the selected points as they lay in the full dataset.

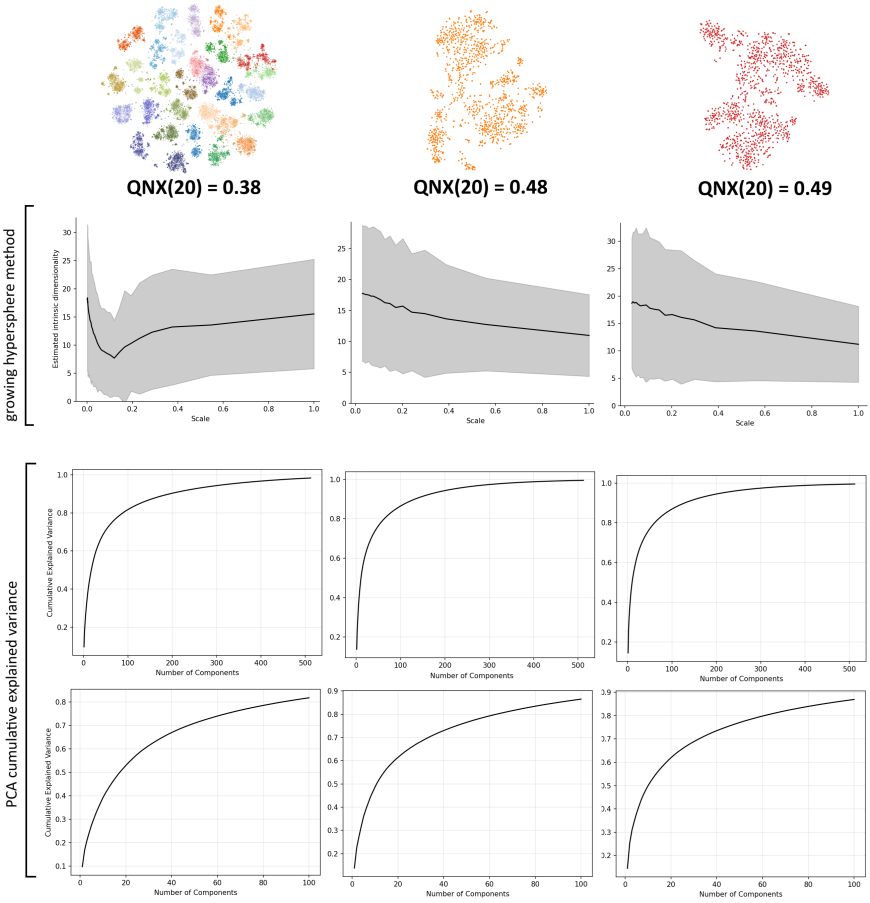


Figure 5.6: Top row: NE of subsets of ImageNet, the leftmost sample consists of the first 30 classes of ImageNet, the other two samples are the 2^{nd} and 6^{th} classes of the dataset. The local quality score ($Q_{NX}(20)$) of the embeddings using the subsampled dataset as reference HD data are indicated under each embedding, these scores were first introduced in [39] and are defined in Section 1.2, equation 1.1. The local quality scores figuring here are averaged across all points in the embedding. Second row: estimated intrinsic dimensionalities of the corresponding subsets, using the growing hypersphere method on the 1280-dimensional EVA features [113]. Bottom two rows: cumulative explained variance ratios of PCA performed on the subsets, up to the 500^{th} and the 100^{th} principal components.

The subsampled points are represented as NE in Figure 5.6, along with their multi-scale IDIM curves and PCA explained variance ratios. The local quality scores of the embeddings are indicated under each NE. The IDIM values estimated by the growing hypersphere method are in line with the values found in the leftmost part of the multi-scale IDIM curve of Figure 5.4; were they to be converted to "true" IDIMs using the correspondence of Figure 5.5, these values would span a large range (approximately from 10 to 40).
Some methods estimating IDIM use the notion of local linearity like [114], but these methods tend to overestimate the true IDIM when the data is organised in curving manifolds and be particularly sensitive to noise, outliers, border effects, and hyperparameter choices. When looking at the cumulative explained variance ratios of PCA performed on the subsets, it is difficult to place a clear threshold on the number of principal components to keep in order to capture most of the variance: the curves are smooth and don't exhibit clear knees.

Both the PCA approach and the growing hypersphere method would suggest IDIM values (much) larger than 10, but when looking at the local quality scores of the NE in Figure 5.6, these scores are quite high: around 0.4 and 0.5. This indicates that the neighbourhoods around the points can be quite well preserved in 2-dimensional NE, suggesting that the dimensionality of the local structures in HD are likely over estimated by the previous methods. Is a more accurate approach possible here?

Intrinsic dimensionality estimators based on local linearity or on local MDS estimate the IDIM by measuring how well the local structures from the HD data can be captured when represented in LD through a linear projection or through an embedding preserving pairwise distances. *Perhaps one can keep this philosophy, but use NE methods instead?* After all, NE methods are particularly well suited to "unroll" curving manifolds, reveal-

ing their underlying LD structure; moreover, as shown in Annex 5.2, NE methods are robust to noise precisely in the setting where the data lies in LD manifolds in HD spaces.

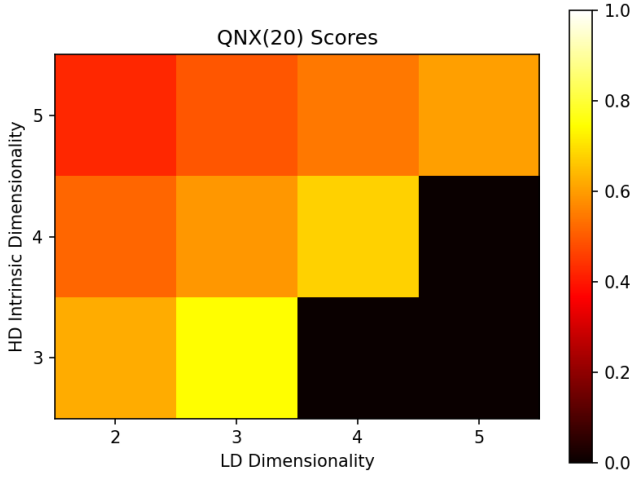


Figure 5.7: Relationship between the IDIM in the HD data and the local quality score $Q^i_{NX}(20)$ on NE of the data, through LD dimensionalities ranging from 2 to 5, visualised as a heat-map (black boxes indicate no data).

To first evaluate the possibility of using NE methods to estimate IDIM, Figure 5.7 shows how the $Q_{NX}(20)$ local quality score of NE evolve with the known IDIM of synthetic HD data and the dimensionality of the embedding space. The HD data is sampled from uniform distributions of a given dimensionality. The heat-map confirms the intuition that, for a fixed LD dimensionality, the local quality scores decreases when the IDIM of the data increases, and that for a fixed IDIM, the local quality scores increase when the LD dimensionality increases. The seemingly monotonic nature of the relationships might be usable to estimate IDIM.
Figure 5.8 shows how the $Q_{NX}(20)$ scores of 2-dimensional NE of uniformly distributed HD data correspond to different IDIMs. There, as in the correspondence curve of Figure 5.5, low $Q_{NX}(20)$ values (corresponding to high estimated IDIM in Fig. 5.5) yield a large spread of possible IDIMs. However, the values of $Q_{NX}(20)$ obtained on the NE of the ImageNet subsets in Figure 5.6 (0.4 and 0.5) correspond to a much narrower range of

"true" IDIM values when considering the relationship of Figure 5.8. Using the NE and $Q_{NX}(20)$ based estimator, the IDIM of the sample of 30 classes is estimated to be between 4 and 9, and the IDIMs of the two other samples are estimated to be between 2.5 and 6.5, as indicated by the thin blue lines in Figure 5.8. This brings much tighter bounds on the IDIM than the other tested methods. The estimation of IDIM across the whole ImageNet dataset across multiple scales is not performed in this annex, but the results on the subsample do not invalidate the possibility that the full dataset could be accurately represented in 32 dimensions using NE methods that preserve both local an global structures, as the method presented in Section 3.7.

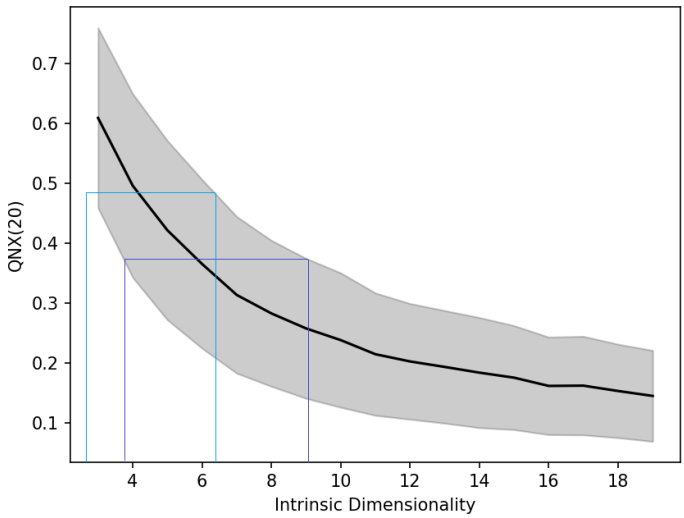


Figure 5.8: Relationship between the local quality score $Q^i_{NX}(20)$ on 2-dimensional *t*-SNE embeddings of uniformly distributed data points and their intrinsic dimensionality. The thin blue lines show how the local quality scores of the embeddings of Fig. 5.6 would correspond to different intrinsic dimensionalities if this relationship was to be used as an estimator.

This work will be pursued with more robust empirical evaluations in future work, aiming a publication. If the approach proves to be reliable, it could be another reason to use NE methods capable of good local quality preservation in LD spaces of intermediate dimensionalities, like the method developed in Section 3.6.

5.2 Are neighbour embeddings robust to noise?

This parenthesis is added to investigate the surprising behaviour observed in Figure 1.2 where slight, almost imperceptible noise in the input data gets severely amplified by the application of NE. This behaviour is unexpected, because NE methods are generally somewhat robust to noise when the hyperparameters are appropriate [37,38]. While the amount of papers explicitly pointing out the robustness of NE to noise is limited, its ubiquity in application involving the notoriously noisy single-cell datasets [8,13,15–18] is a testament to this property.

Studying the robustness of NE to noise is beyond the scope of this thesis and would warrant a thorough investigation involving different types of noise (structured or unstructured, with varying structures, distributions, and scales), applied to datasets with diverse characteristics such as the local intrinsic dimensionalities, cluster separability, and density variations. Such a study would also need to consider multiple NE methods or different hyperparameter configurations within flexible NE algorithms.
Instead, this short enquiry explains why, in some cases (isotropic noise around the points, locally low intrinsic dimensionality, high ambient dimensionality), NE can be resilient to the effects of noise. This might help understand why the example of Figure 1.2 was so strongly affected by noise compared to the more standard scenarios found in the literature and the applicative fields.

Since NE methods focus on preserving neighbourhoods around each points, one can wonder if it is this characteristic that allows them to be robust to noise in common applicative cases. Since visualising data with high local intrinsic dimensionalities is problematic (intuition on intrinsic dimensionality is given in 5.1), we will assume that common datasets used in NE exhibit LD manifolds laying in HD ambient spaces. Let us pose the following hypothesis: *If the data lies in manifolds of particularly low dimensionality relative to the ambient space, and if the noise is high dimensional and unstructured, then NE methods are robust to the noise because the neighbourhoods in HD will remain largely unchanged by the perturbations.*

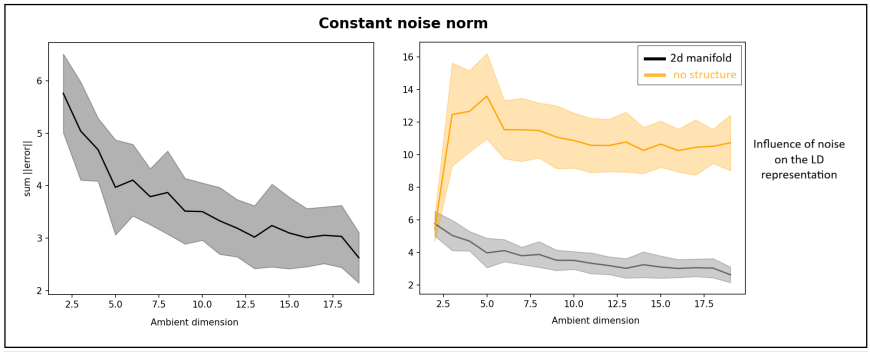
This hypothesis will not be formally proven true here, but strong arguments supporting it will be given. Figure 5.9 shows how NE representa-

tions of HD data suffer from noise applied in HD, as the ambient dimensionality in HD increases. Two types of datasets are considered: one where the data lies in a 2-dimensional manifold in HD (decreasing intrinsic dimensionality / ambient dimensionality fraction along the x-axis), and one where the data is sampled from a uniform distribution in HD (the intrinsic dimensionality is equal to the ambient dimensionality). More details on the experimental protocol are given in the caption.

Considering the case where the norm of the noise vectors applied to each point is constant throughout the growth in ambient dimensionality (top row), it is manifest that, as the intrinsic dimensionality of the data decreases relative to the ambient dimensionality, the embeddings become less and less affected by high dimensional noise.
When the intrinsic dimensionality is equal to the ambient dimensionality, the effect of noise on the embeddings worsens severely before slowly decreasing, without approaching the resilience observed in the first case in this range of ambient dimensionalities. The sharp increase followed by a slower decrease could suggest two competing phenomena. This study does not explain the initial increase, it is possible that the explanation is related to the way HD similarities are computed or to the non-linear nature of NE. The later tampering can be explained by the fact that the distances to neighbours in higher dimensions grow with the square root of the dimensionality whereas the noise remains of constant norm.
This consideration makes the increasing robustness to noise of the 2-dimensional manifold case particularly noteworthy, because there, the distances to neighbours remain the same throughout the different ambient dimensionalities. This might suggest that angles between noise vectors and the manifold could be responsible for resistance to noise more than scale.

In the second case, where the norm of the noise vectors is scaled by the square root of the ambient dimensionality (bottom view in Fig. 5.9), the effect of the growing noise on the embeddings of the low dimensional manifold appears to be constant throughout the growth in ambient dimensionality and noise strength, whereas the effect on the data with the intrinsic dimensionality equal to the ambient noise worsens monotonically throughout the growth.




Constant noise norm
sum ||error||
Ambient dimension
2d manifold
no structure
Influence of noise on the LD representation

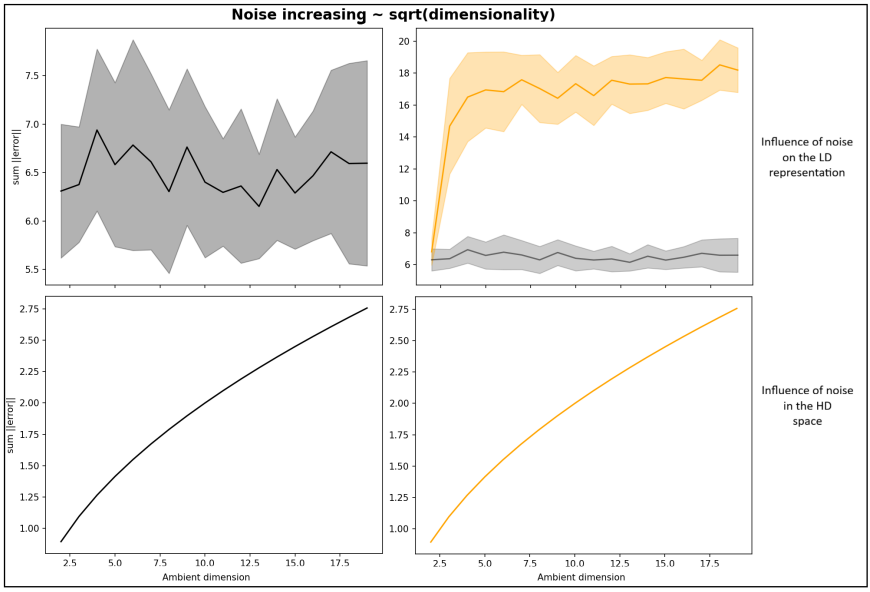

Noise increasing ~ sqrt(dimensionality)
sum ||error||
Ambient dimension
Influence of noise on the LD representation
Influence of noise in the HD space

Figure 5.9: Effect of adding isotropic noise of a given norm to all points in HD, with an increasing dimensionality. The noisy and raw datasets are embedded to 2-dimensions using t-SNE, the errors reported on the y-axis measure how much the points have moved between the embeddings of the raw and noisy data: a value of 0.0 would indicate that the application of noise in HD had no effect on the LD embedding.
The t-SNE runs share hyperparameters and initialisations, the errors on LD embeddings are computed after a normalisation step because the preservation of scale across runs is not guaranteed by t-SNE. The x-axis indicates the ambient dimensionality of the data. Black curves: the dataset consists of points laying in a 2-dimensional manifold in the ambient space. Orange curves: the dataset consists of points sampled from a uniform distribution of dimensionality identical to the ambient space. The reported errors are the mean of 30 independent runs.
Top view: the noise is of fixed norm throughout. Bottom view: the norm of the noise is scaled by $\sqrt{M}$, where M is the ambient dimensionality, to mimic the growth of lengths in high dimensions. The "Influence of noise in the HD space" figures plot the norm of the noise vectors applied to each point in HD, to help interpret the results.

To further investigate the robustness of NE to HD noise in the case where the data lies in a low dimensional manifold, Figure 5.10 grows the ambient dimensionality from 2 to 1000, while scaling the norm of the noise vectors by the square root of the ambient dimensionality. There, the effect of noise on the LD embeddings remains remarkably low considering the amplitude of the noise applied in HD, as shown by the rightmost plot.

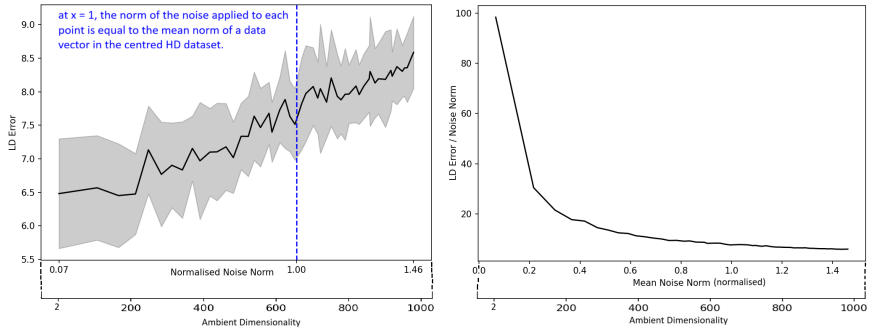


Figure 5.10: Evolution of the effect of noise in the LD representation of the uniform dataset from Figure 5.9. The noise increases with the square root of the ambient dimensionality, the ambient dimensionality grows from 2 to 1000. In both plots, the x-axis shows both the ambient dimensionality and the norm of the noise vectors applied to each point in HD. This norm is normalised for the plots such that at $x = 1.0$, the noise norm is equal to the average norm of an observation vector in HD, after centring the data.
Left: the y-axis shows the influence of the HD noise on the NE representation of the data, similarly to Figure 5.9.
Right: the y-axis plots the evolution of the ratio between the noise-induced deviation in LD and the norm of the noise vectors applied in HD.

One explanation for the observed robustness to noise of NE when the local intrinsic dimensionality of the data is low compared to the ambient

dimensionality is that, the higher the ambient dimensionality / intrinsic dimensionality ratio, "the more orthogonal" the noise vectors are to the manifold where the data lies. This relation is plotted in Figure 5.11, where the component of the noise aligned with the manifolds visibly decays with the dimensionality of the noise and ambient space.
This tendency stems from a well known natural property of HD spaces, where random vectors tend to be orthogonal to each other. In machine learning, this property can be both a blessing and a curse: it can be detrimental in gradient descent where it leads to inefficiencies, but on the other hand some models use this property to their advantage. This is the case in the fascinating *Local prediction-learning in high-dimensional spaces enables neural networks to plan* [117] paper, where the likely orthogonality of random vectors allow the organisation of the internal representation of models.

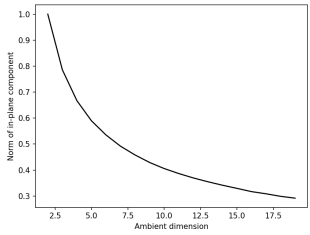


Figure 5.11: Evolution of the proportion of the noise vector that lies in the tangent space of the manifold (in-plane component) as the ambient dimensionality increases.

Intuition on how noise unaligned with the data manifold tends not to impact neighbourhoods significantly is presented in the drawing of figure 5.12.

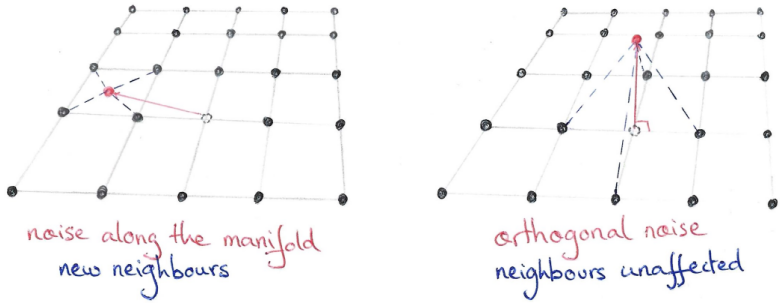


Figure 5.12: When noise is unaligned with the data manifold, the neighbourhoods tend to be unaffected.

Coming back to the example of Figure 1.2, the data lies in a 2-dimensional manifold in 2-dimensions, meaning that the intrinsic dimensionality / ambient dimensionality ratio is 1: the worst case scenario if the hypothesis holds true. In practice, datasets that show meaningful structures when scrutinised through the lenses of NE likely exhibit low local intrinsic dimensionalities, bringing them in more favourable conditions regarding robustness to noise.

5.3 Datasets

This section lists the datasets used in the thesis.

- **"S-curve"**: This toy dataset samples points from a curved 2-dimensional plane curving to for a "S" in a 3-dimensional ambient space.
- **"mouse brain"**: Refers to the single-cell RNA-sequencing dataset measuring the concentration of thousands of different types of messenger RNA in 23,822 cells from the neocortex of mice [7], specifically from the primary visual cortex (V1, or VISp) and the anterolateral motor cortex (ALM). Reading the presence of messenger RNA is a surrogate for the gene expression levels in the cells, which are difficult to measure directly. Biologists have distinguished 133 cell classes and assigned colours following some structure: at higher levels, the hot colours are assigned to GABAergic neurons (inhibitory), cold colours for glutaminergic neurons (excitatory), and grey-brow colours for cells that are not neurons, such as astrocytes and oligodendrocytes. The paper [7] comes with a useful dendrogram showing the hierarchical organisation of the cells according to biological knowledge, using the same colours as the ones shown throughout this thesis.
  When not specified otherwise, this dataset is preprocessed by feature selection and non-linear scaling as in [8], and then keeping the first 256 or 50 principal components of the scaled selection, depending on the experiment.
- **"COIL-20"**: The COIL-20 dataset consists of images of 20 different objects, rotating around their centre and pictured every 5 degrees, resulting in 72 images per object. The rotation naturally describes a 1-dimensional ring manifold in the high-dimensional space of images. The background is plain, allowing use without processing through computer vision methods.
- **"blobs"**: The blobs dataset is a synthetic dataset consisting Gaussian blobs in a high dimensional space. The number of points, number of centres, standard deviation of the blobs, and the ambient dimensionality depend on the experiment.
- **"abalone"**: The abalone dataset consists of physical measurements of abalones (a type of sea snail) the label to predict is the age of the abalone. The dataset contains 4177 samples described by 8 features.

- **"MNIST"**: The MNIST dataset consists of $70,000$ images of handwritten digits from 0 to 9, each image is 28×28 pixels, leading to a 784-dimensional space when flattening the images. The dataset is commonly used as a benchmark in machine learning and computer vision. If not specified otherwise in this thesis, only the train set of $60,000$ images is used.
- **"ImageNet"**: The ImageNet dataset is a large dataset of labelled images commonly used in machine learning and computer vision, the dataset features 1000 classes, and has 1.28 millions of observations. In this thesis, the raw image pixel values are not used but rather 1280-dimensional feature vectors extracted from a large vision-language model called EVA [113].
- **"Anuran"**: Contains 7195 observations consisting of extracted features from a collection of frog calls.

5.4 Paper : Impact of data subsampling on fast multi-scale neighbour embeddings

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

Impact of data subsamplings in Fast Multi-Scale Neighbor Embedding

Pierre Lambert[1], John A. Lee[1,2], Michel Verleysen[1], Cyril de Bodt[1]

1- UCLouvain.be - ICTEAM/ELEN
Place du Levant 3 L5.03.02, 1348 Louvain-la-Neuve - Belgium
2- UCLouvain.be - IREC/MIRO
Avenue Hippocrate 55 B1.54.07, 1200 Brussels - Belgium

Abstract. Fast multi-scale neighbor embedding (f-ms-NE) is an algorithm that maps high-dimensional data to a low-dimensional space by preserving the multi-scale data neighborhoods. To lower its time complexity, f-ms-NE uses random subsamplings to estimate the data properties at multiple scales. To improve this estimation and study the f-ms-NE sensitivity to randomness, this paper generalizes the f-ms-NE cost function by averaging several subsamplings. Experiments reveal that this can slightly improve the quality of the embeddings while maintaining reasonable computation times. Codes are available at `https://github.com/cdebodt/Fast_Multi-scale_NE`.

1 Introduction

Dimension reduction (DR) maps high-dimensional (HD) data to a low-dimensional (LD) space such that the LD embedding faithfully represents the HD data. Some information is typically lost in the DR process and, therefore, the faithfulness of a projection is considered with respect to some criterion. The main uses of DR are in machine learning, to curb the curse of dimensionality, and in visualisation.

When HD dimension is high, the Euclidean distances between points tend to concentrate towards similar values [1]. Because of this phenomenon, using distance preservation as a direct criterion for DR can become problematic. Neighbor embedding (NE) techniques such as stochastic neighbor embedding (SNE) [2] alleviate the effects of distance concentration by defining neighbor probability distributions in both spaces to embed points in LD [3]. These methods are frequently used in visualization as they tend to produce embeddings that show well separated, readable clusters and nicely preserve local neighborhoods [4].

Originally, NE algorithms required the user to specify a scale for the neighborhood preservation. By capturing the structures on a single scale, embeddings produced by these methods can lead the user to erroneous interpretations. More recently, multi-scale NE approaches were introduced by combining neighborhood probabilities tuned across various scales [5, 6]. Multi-scale approaches often yield better results by preserving the distant structures, while still ensuring good local neighborhood preservation. However, multi-scale NE methods have a time complexity of $\mathcal{O}(N^2 \log N)$ with N data points, restricting their use to data sets of moderate size. A fast version of multi-scale NE (f-ms-NE) [7] has a time complexity of $\mathcal{O}(N \log^2 N)$, with a slight decrease in neighbor preservation as

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

a trade-off. To achieve such an acceleration, f-ms-NE relies on random data subsamplings to capture both the local and global structures in the HD data. However, the stability of the f-ms-NE performances with respect to the number of these subsamplings has not been characterized previously.

This paper hence generalizes f-ms-NE by defining cost functions averaged over multiple random subsamplings, enabling to study the effect of their number on the LD embeddings. In addition to aiming at increasing DR quality while preserving manageable computation times, this work further assesses the sensitivity of f-ms-NE to randomness. Public codes are freely available at `https://github.com/cdebodt/Fast_Multi-scale_NE`.

This paper is structured as follows: Section 2 summarizes NE algorithms. Section 3 explains how f-ms-NE is adapted to consider multiple random subsamplings. Section 4 details experimental results by comparing the performances of the f-ms-NE algorithm across an increasing number of subsamplings, in both terms of DR quality and speed. Final conclusions are drawn in Section 5.

2 Neighbor embedding algorithms

The f-ms-NE algorithm accelerates multi-scale NE, which is based on single-scale NE. Section 2.1 first sketches single-scale NE, Section 2.2 then describes its multi-scale extension, and Section 2.3 finally details the f-ms-NE acceleration.

2.1 Single-scale neighbor embedding

Let $\mathbf{\Xi} = [\xi_i]_{i=1}^N$ denote a set of N points in a HD space with M features. Let $\mathbf{X} = [\mathbf{x}_i]_{i=1}^N$ represent these data in a P-dimensional LD space, $P \leq M$. The HD and LD distances between the i^{th} and j^{th} points are δ_{ij} and d_{ij}, respectively, for $i \in \mathcal{I} = \{1, ..., N\}$ and $j \in \mathcal{I} \setminus \{i\}$. SNE [2] aims at preserving pairwise similarities from HD to LD, which are respectively defined as

$$\sigma_{ij} = \frac{\exp(-\pi_i \delta_{ij}^2/2)}{\sum_{k \in \mathcal{I} \setminus \{i\}} \exp(-\pi_i \delta_{ik}^2/2)}, \;\; s_{ij} = \frac{\exp(-d_{ij}^2/2)}{\sum_{k \in \mathcal{I} \setminus \{i\}} \exp(-d_{ik}^2/2)}, \;\; \sigma_{ii} = s_{ii} = 0 \;\; . \quad (1)$$

Precision π_i is adapted to the local density in order to tune the HD similarities to a target scale. The scale is provided by the user in the form of a perplexity K for the distribution $[\sigma_{ij}; j \in \mathcal{I} \setminus \{i\}]$ such that $\log K = -\sum_{j \in \mathcal{I} \setminus \{i\}} \sigma_{ij} \log \sigma_{ij}$.

SNE interprets these normalized similarities as neighborhood probabilities around each point, in order to produce a LD embedding that minimises the sum of Kullback-Leibler (KL) divergences $C_{SNE} = \sum_{i \in \mathcal{I}, j \in \mathcal{I} \setminus \{i\}} \sigma_{ij} \log (\sigma_{ij}/s_{ij})$.

The t-SNE extension [4] symmetrizes the similarities and considers a Student t function with one degree of freedom in LD space, to cope with 'crowding' problems in LD. The HD similarities τ_{ij} and LD ones t_{ij} are now defined as

$$\tau_{ij} = (\sigma_{ij} + \sigma_{ji})/(2N), \;\; t_{ij} = \frac{(1 + d_{ij}^2)^{-1}}{\sum_{k \in \mathcal{I}, l \in \mathcal{I} \setminus \{k\}} (1 + d_{kl}^2)^{-1}}, \;\; \tau_{ii} = t_{ii} = 0 \;\; . \quad (2)$$

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

2.2 Multi-scale neighbor embedding

The multi-scale SNE method [5] combines SNE similarities computed with exponentially increasing perplexities. Single-scale similarities are indexed by a scale counter h for $i \in \mathcal{I}$ and $j \in \mathcal{I} \setminus \{i\}$:

$$\sigma_{ijh} = \frac{\exp(-\pi_{ih}\delta_{ij}^2/2)}{\sum_{k\in\mathcal{I}\setminus\{i\}} \exp(-\pi_{ih}\delta_{ik}^2/2)},\ s_{ijh} = \frac{\exp(-p_{ih}d_{ij}^2/2)}{\sum_{k\in\mathcal{I}\setminus\{i\}} \exp(-p_{ih}d_{ik}^2/2)},\ \sigma_{iih} = s_{iih} = 0\ .$$

HD precisions π_{ih} are set using perplexities growing as $K_h = 2^{h-1}K_1$, with a small base perplexity $K_1 = 2$ and $1 \leq h \leq H = \lfloor \log_2(N/K_1) \rceil$, where $\lfloor\cdot\rceil$ denotes rounding. LD precisions $p_{ih} = p_h = K_h^{-2/P}$ follow the exponential growth of the HD precisions. Multi-scale similarities average single-scale ones as

$$\sigma_{ij} = H^{-1}\sum\nolimits_{h=1}^{H} \sigma_{ijh},\ \ s_{ij} = H^{-1}\sum\nolimits_{h=1}^{H} s_{ijh}\ . \tag{3}$$

The authors recommend using L-BFGS [8] to optimise the cost function.

2.3 Fast multi-scale neighbor embedding

Single-scale and multi-scale NE methods have time complexities of $\mathcal{O}(N^2)$ and $\mathcal{O}(N^2 \log N)$, respectively. The typical smallness of perplexity K with respect to N enables single-scale methods to consider that sufficiently distant points have a null HD similarity. Sparse HD similarities can be computed by finding the sets of nearest neighbors of each point using vantage-point trees [9]. This reduces their computation time to $\mathcal{O}(KN \log N)$. On the LD side, a Barnes-Hut (BH) algorithm [10, 11] approximates efficiently the similarities by relying on tree structures to compound far-away points, dispensing with computing all pairwise interactions, reducing the time complexity to $\mathcal{O}(N \log N)$. The BH method uses a threshold parameter $\theta \in [0, 1]$ that determines whether to use the approximation or not, with higher values meaning rougher but faster estimations.

The largest scale used to compute multi-scale similarities has a perplexity K_H in the $\mathcal{O}(N)$ range. At such a scale, the above sparse HD similarity approximation would require neighbor sets of $\mathcal{O}(N)$ size, defeating the purpose of accelerating the computations. For this reason, reducing the time complexity of multi-scale NE requires a different approach for the HD side. In [7], the authors tackle this problem by defining small multi-scale neighbor sets $\widetilde{\mathcal{I}}_i$ for $i \in \mathcal{I}$.

To compute $\widetilde{\mathcal{I}}_i$, the authors define a hierarchy of subsampled HD data sets $\{\mathbf{\Xi}_h\}_{h=1}^H$, where $\mathbf{\Xi}_h$ is a random sample of $\mathbf{\Xi}$ with $\lfloor 2^{1-h}N \rceil$ elements drawn without replacement. A vantage-point tree is created on $\mathbf{\Xi}_h$ for each scale $h \in \{1, ..., H\}$; the trees are generated in $\mathcal{O}(N \log N)$ time. For each scale, the corresponding tree is used to compute the neighborhood $\widetilde{\mathcal{I}}_{ih}$ within the sample $\mathbf{\Xi}_h$ for $i \in \{1, ..., N\}$; this takes a total of $\mathcal{O}(N \log^2 N)$ time. By having samples $\{\mathbf{\Xi}_h\}_{h=1}^H$ that decrease in size when the scale grows, the accounted neighbors $\widetilde{\mathcal{I}}_{ih}$ are more and more dispersed in the data cloud, hence capturing larger scale properties. The multi-scale neighbor sets are then defined as $\widetilde{\mathcal{I}}_i := \cup_{h=1}^H \widetilde{\mathcal{I}}_{ih}$.

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

Like the non-accelerated version, f-ms-NE uses multi-scale similarities averaged over sparse single-scale similarities,

$$\widetilde{\sigma}_{ij} = H^{-1} \sum_{h=1}^{H} \widetilde{\sigma}_{ijh}, \quad \text{with} \quad \widetilde{\sigma}_{ijh} = \begin{cases} \frac{\exp(-\widetilde{\pi}_{ih}\delta_{ij}^2/2)}{\sum_{k\in\widetilde{\mathcal{I}}_i}\exp(-\widetilde{\pi}_{ih}\delta_{ik}^2/2)} & \text{if } j \in \widetilde{\mathcal{I}}_i \\ 0 \text{ otherwise} \end{cases} .$$

Precision $\widetilde{\pi}_{ih}$ is fixed such that

$$K_1 = -\sum_{j\in\widetilde{\mathcal{I}}_{ih}} \widetilde{\sigma}^{\pi}_{ijh} \log \widetilde{\sigma}^{\pi}_{ijh} \text{ where } \widetilde{\sigma}^{\pi}_{ijh} = \begin{cases} \frac{\exp(-\widetilde{\pi}_{ih}\delta_{ij}^2/2)}{\sum_{k\in\widetilde{\mathcal{I}}_{ih}}\exp(-\widetilde{\pi}_{ih}\delta_{ik}^2/2)} & \text{if } j \in \widetilde{\mathcal{I}}_{ih} \\ 0 \text{ otherwise} \end{cases} .$$

In the $t-$distributed version, the authors use symmetrised HD similarities $\widetilde{\tau}_{ij} = (\widetilde{\sigma}_{ij} + \widetilde{\sigma}_{ji})/(2N)$ and optimise the cost function $\widetilde{C}_t = -\sum_{i\in\mathcal{I}, j\in\mathcal{I}\setminus\{i\}} \widetilde{\tau}_{ij} \log t_{ij}$.

3 Multiple subsamplings in fast multi-scale NE

In f-ms-NE, the larger-scale structures of the HD data are captured by computing neighborhoods on samples of decreasing sizes. Therefore, f-ms-NE estimates the larger-scale data properties with some sensitivity to randomness, as an unlucky sampling could bypass some important structures within the data. We hence propose to adapt f-ms-NE by using multiple subsamplings of the HD data, aiming at reducing the impact of randomness when modeling large-scale structures.

The proposed adaptation is to perform the whole sampling process R times, to compute sets $\widetilde{\mathcal{I}}_{ir}$ for $i \in \{1, ..., N\}$ and $r \in \{1, ..., R\}$. The sparse similarities are computed independently for each batch and the cost function becomes

$$\widetilde{C}_t = R^{-1} \sum\nolimits_{r=1}^{R} \left(-\sum\nolimits_{i\in\mathcal{I}, j\in\mathcal{I}\setminus\{i\}} \widetilde{\tau}_{ijr} \log t_{ij} \right) . \tag{4}$$

As R is set to a small value independent of N, the time complexity with regards to N remains unchanged. But as the original cost function is computed R times, a computation time increase is expected. However, each repetition being independent, a parallel implementation with R threads can easily be done.

4 Experimental results and discussion

The method is tested on seven data sets of size N and dimensionality M [12]. Anuran: (N,M) = (7195, 22); Plant: (N,M) = (9568, 4); Gestures: (N,M) = (9901, 18); Satellite: (N,M) = (6434, 36); Waveform: (N,M) = (5000, 40); Theorem: (N,M) = (6118, 51); and Musk: (N,M) = (6598, 166). On each set and for each $R \in \{1, ..., 5\}$, f-ms-NE with multiple subsamplings is applied 30 times using a random initialisation, and 30 times with a PCA initialisation. The t-distributed version of the algorithm is used and the target dimension P is 2 for all experiments. Each optimization consists of 30 iterations of the L-BFGS algorithm and uses a BH threshold $\theta = 0.75$, as these values tend to produce good DR quality in reasonable time for the original f-ms-NE [7].

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

Figure 1 shows the quality of the resulting embeddings. As in [7], their quality is measured by the area under the curve (AUC) of the relative neighborhood preservation R_{NX} [13], with a logarithmic scale for the neighborhood size. The AUC of R_{NX} reaches 1 when the preservation is perfect for all data scales.

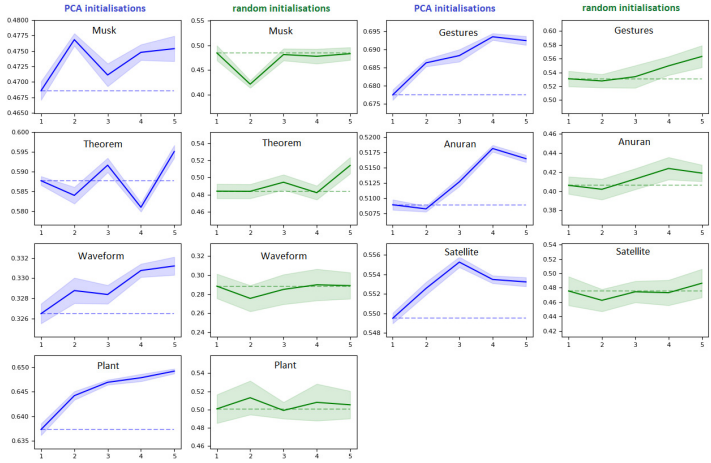


Fig. 1: Evolution of the AUC of the R_{NX} curves with the number of subsamplings. The y-axis is the AUC, and the x-axis is the number R of subsamplings. The solid lines are the mean values over 30 trials, the shaded areas correspond to one standard deviation on each side. The dashed line corresponds to the mean AUC for the original version of f-ms-NE, which uses one sampling.

We observe that subsampling multiple times has an effect on the DR quality of the embeddings that f-ms-NE yields. However, the change is often slight when compared to embeddings produced from a single sampling and, surprisingly, the change is not always positive. Using multiple subsamplings in conjunction with a PCA initialisation tends to produce a more consistent increase in quality, but the magnitude of the increase depends on the data set. The modest benefits of using multiple subsamplings can suggest that the construction of $\widetilde{\mathcal{I}}_i$ in the original f-ms-NE algorithm is already quite robust to randomness.

In terms of computation time, we obtain a consistent linear increase in time with respect to R. With a single thread implementation on the tested computer, using $R = 5$ requires approximately twice the time that the algorithm would take in its original version.

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

5 Conclusion

Experimental results indicate that using multiple subsamplings in f-ms-NE can provide improvements to the DR quality. The moderate nature of these improvements hints that the sampling process used by f-ms-NE enables to model the multi-scale structures of the HD data with a remarkable robustness with respect to the randomness.

To further study the influence of randomness on f-ms-NE, alternate subsampling strategies will be envisioned. For instance, using a biased sampling to increase the diversity of the sampled points in the hierarchy of subsampled HD data sets $\{\mathbf{\Xi}_h\}_{h=1}^{H}$ is likely to improve the estimation of the global structure of the HD data.

References

[1] D. François, V. Wertz, and M. Verleysen. The concentration of fractional distances. *IEEE Transactions on Knowledge and Data Engineering*, 19:873–886, 2007.

[2] Geoffrey Hinton and Sam Roweis. Stochastic neighbor embedding. *Advances in neural information processing systems*, 15:833–840, 2003.

[3] John A. Lee and Michel Verleysen. Shift-invariant similarities circumvent distance concentration in stochastic neighbor embedding and variants. *Procedia Computer Science*, 4:538–547, 2011.

[4] L. van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[5] John A. Lee, Diego H. Peluffo-Ordóñez, and Michel Verleysen. Multi-scale similarities in stochastic neighbour embedding: Reducing dimensionality while preserving both local and global structure. *Neurocomputing*, 169:246–261, 2015.

[6] C. de Bodt, D. Mulders, M. Verleysen, and J. A. Lee. Perplexity-free t-SNE and twice student tt-SNE. *ESANN*, pages 123–128, 2018.

[7] Cyril de Bodt, Dounia Mulders, Michel Verleysen, and John Aldo Lee. Fast multiscale neighbor embedding. *IEEE T NEUR NET LEAR*, pages 1–15, 2020.

[8] R. Byrd, Peihuang Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *SIAM J. Sci. Comput.*, 16:1190–1208, 1995.

[9] Peter Yianilos. Data structures and algorithms for nearest neighbor search in general metric spaces. volume 93, 01 1993.

[10] Laurens van der Maaten. Accelerating t-SNE using tree-based algorithms. *Journal of Machine Learning Research*, 15(93):3221–3245, 2014.

[11] Cyril de Bodt, Dounia Mulders, Michel Verleysen, and John A. Lee. Extensive assessment of Barnes-Hut t-SNE. In *ESANN*, pages 135–140, 2018.

[12] Moshe Lichman. UCI Machine Learning repository, 2013.

[13] John Lee and Michel Verleysen. Quality assessment of dimensionality reduction: Rank-based criteria. *Neurocomputing*, 72:1431–1443, 03 2009.