ANCHOR-GUIDED DIMENSION REDUCTION METHODS FOR COHORT DATA VISUALISATION

A THESIS SUBMITTED TO THE UNIVERSITY OF MANCHESTER
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
IN THE FACULTY OF SCIENCE AND ENGINEERING

2022

Rui Qin

School of Engineering, Department of Computer Science

Contents

Word Count: 26963

List of Tables

List of Figures

Notations

Unbolded x represents a single number, boldface $\mathbf{x}$ represents a vector, and capital boldface $\mathbf{X}$ represents a matrix. An individual element of a vector is denoted with a subscript and without boldface. For example, the ith element of a vector $\mathbf{x}$ is x_i. A bold lower-case letter with an index such as $\mathbf{x}_j$ represents a particular row of matrix $\mathbf{X}$.

Symbol	Description
$\mathbf{0}_n$	n-dimensional column vectors with all elements equal to 0.
$\mathbf{0}_{n\times m}$	The zero matrix.
$\mathbf{1}_n$	n-dimensional column vectors with all elements equal to 0 and 1.
$\mathbf{1}_{n\times m}$	The one matrix.
$a_c \in \mathbb{R}$	The scaling parameter.
b	The batch size of the dataset.
C	The number of cohorts.
$\mathbf{C}_i$	The i-th cohort containing all data points whose label $y = i$
D	Dimensionality of original dataset in the high-dimensional space.
d	Dimensionality of the target low-dimensional space, usually $d = 2$ or 3.
$D(\cdot,\cdot)$	The distance or similarity between two data samples in the original space.
$\hat{D}(\cdot,\cdot)$	The distance in the embedded space.
$D_C(i,j)$	The distance between two data cohorts.
$\mathbf{I}_n$	The identity matrix of size n.
k	The number of the neighbourhood points.
m	The number of anchor points.
$\max(\cdot,\cdot)$	It returns the larger one between the two input numbers.
n	The number of the data samples in the original dataset.

Symbol	Description
p	The anchor sparsity parameter, controlling the number of anchor points.
$p_{j\|i}$	The conditional probability.
p_{ij}	The pairwise similarity between data points in the high-dimensional space, the symmetrized conditional probability
P_{l}	The local neighbourhood preservation score.
P_{g}	The global structure preservation score.
P_{s}	The separability score.
P	The overall evaluation score.
par	The stochastic triplet selection parameter.
$Perp(P_i)$	The perplexity.
q_{ij}	The pairwise similarity between data points in $2,3$-D space, the Student t-distribution with one degree of freedom.
r	The local vs. global control parameter.
$\mathbf{R}_c \in \mathbb{R}^{d \times d}$	The rotation matrix.
t	The number of nearest anchor points used to do the data reconstruction.
$\mathbf{u}_i$	The anchor point vector.
$\mathbf{U}$	The anchor point matrix.
$\mathbf{W} = \{\mathbf{w}_i\}$	The reconstruction matrix containing reconstruction weights.
$\mathbf{X} \in \mathbb{R}^{n \times D}$	The original dataset in the high-dimensional space.
$\mathbf{x}$	The high-dimensional data sample vector.
$\hat{\mathbf{X}} \in \mathbb{R}^{n \times d}$	The set of low-dimensional embedded points.
$\hat{\mathbf{x}}$	The low-dimensional embedded points.
y	The label notation.
$\delta > 0$	The hyper-parameter controls the embedding scale.
$\hat{\delta}$	The relaxation parameter used in local neighbourhood preservation calculation, the neighbourhood margin.
Γ^{SOE}	The ordinal constraint set, the SOE can be replaced by s-OE, LOE and triOE, referring different ordinal constraint sets.
$\overline{\kappa}$	The maximum of the neighbourhood size considered.
λ	The cohort separation control parameter.
$\left\{\hat{\boldsymbol{\upsilon}}_i \in \mathbb{R}^d\right\}_{i=1}^{C}$	The cohort embedding points.
$\boldsymbol{\zeta}_{\mathbf{u}_i}$	The rankings of the other anchor points in terms of their closeness to $\mathbf{u}_i$ based on the distance measure D.
$\boldsymbol{\zeta}_{\mathbf{u}_i}(\mathbf{u}_j)$	The particular ranking of $\mathbf{u}_j$.

Acronyms

2, 3-D	2-D or 3-D
k-NN	k-nearest neighbours
t-NAP	t-nearest anchor points
ANGEL	ANchor GuidEd Local (algorithm)
ANGEL-LR	ANGEL with Loop-Replicate optimisation
ANGEL-warm	ANGEL with warm-start optimisation
DR	dimension reduction
i-ANGEL	incremental ANGEL
Isomap	isometric feature mapping
KL	Kullback-Leibler
L2-pixel distance	the Euclidean distance between images
LAE	local anchor embedding
LE	Laplacian Eigenmap
LLE	locally linear embedding
LOE	local ordinal embedding
LTSA	local tangent space alignment
MDS	multi-dimensional scaling
NLDR	non-linear dimension reduction (method)
p-ANGEL	progressive ANGEL
PaCMAP	pairwise controlled manifold approximation projection
s-ANGEL	stochastic ANGEL
s-UMAP	supervised UMAP
t-SNE (tSNE)	t-distributed stochastic neighbour embedding
UMAP	uniform manifold approximation and projection

Abstract

Data visualisation is the first stage of analysing data and developing data-driven solutions. A visual understanding of the data can help analysts quickly identify key data patterns and intrinsic structural information, facilitating data scientists in many domains and applications. An effective way of “looking at” high-dimensional data is to embed the patterns into 2,3-D spaces by using dimension reduction techniques. The main goal of generating a meaningful visualisation for cohort datasets, which are datasets that contain natural clusters or can be classified by clustering technique, is to preserve the essential aspects of the intrinsic data information, e.g., local neighbourhood of data points, the internal structure of the cohort, positioning, and separation of data cohorts. It is still an open question on how to well balance between all these aspects, and the evaluation of the cohort positioning is mostly done qualitatively by plotting out the embeddings and assessing the plot manually.

This thesis focuses on improving cohort data visualisation and its evaluation. The first contribution is the ANchor GuidEd Local (ANGEL) algorithm with its variations, proposed to balance local neighbourhood preservation, cohort positioning, and cohort separation. The second contribution is the new evaluation approach designed to quantitatively measure all these three aspects. ANGEL is evaluated and compared with a series of state-of-the-art approaches using several benchmark datasets. Results show that it can effectively generate more informative visualisation. The third contribution is the incremental extensions of ANGEL. The simple but intuitive p-ANGEL and i-ANGEL algorithms are proposed to incrementally embed new data points in a batch embedding setup. It also saves memory cost and accelerates the computational speed, more applicable to large-scale real-world cohort data visualisations. Overall, this PhD research pushes the state-of-the-art of cohort data visualisation forward.

Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright

Publication

Journal paper under preparation:
Rui Qin, Xenofon Evangelopoulos, John Yannis Goulermas, and Tingting Mu. Comprehensive Improvement of Cohort Data Visualization via Injecting Anchor-guided Global Distances in Locality, 2022

Acknowledgements

First of all, I would like to give my greatest thanks to my supervisor Dr Tingting Mu. It is my great fortune to be her PhD student. I enjoyed working with her. She is talented and always shares her expertise in the area with me. She also gave me great help during the pandemic in research and daily life.

I sincerely thank my co-supervisor, Prof. Gavin Brown, for his support and guidance, especially during the viva preparation period.

I would like to thank my advisors, Prof. Pierre Olivier and Dr Giles Reger, for their guidance and support in my PhD study. They are very experienced advisors and provide a lot of helpful advice.

I would like to thank my collaborators Prof. Yannis Goulermas, Dr Xenofon Evangelopoulos, and Andrei Aleksanian, for their great help in making my project more convincing and applicable.

Many thanks to our team Tingting, Alessio, Arvid, Huw, Mirantha, Yian, Yusurf, and Xia, for all the support during my PhD study. The group meetings have been constructive in my research.

I would like to thank my friends Xueqi, Jing, Haoruo, Heng, Yu, Yuan, Hiris, Dongjiao, Yichi, Huyan, Xutong, Yaqi, and Shuodi for their support and encouragement. Without them, it would be hard to complete the study and research.

I give my great thanks to Xia Cui and Yuan Chai for their excellent help with professional proofreading.

I would like to thank Dr Ke Chen and Dr Haiping Lu for their immense guidance and help in my viva and the revision of my thesis.

Finally, I would express my deepest thanks to my father, Mr Zhenping Qin and my mother, Prof. Hongxia Guo. Thank you for standing by my side, supporting me, encouraging me whenever I meet difficulties and their continuous support for my study in the UK.

Chapter 1

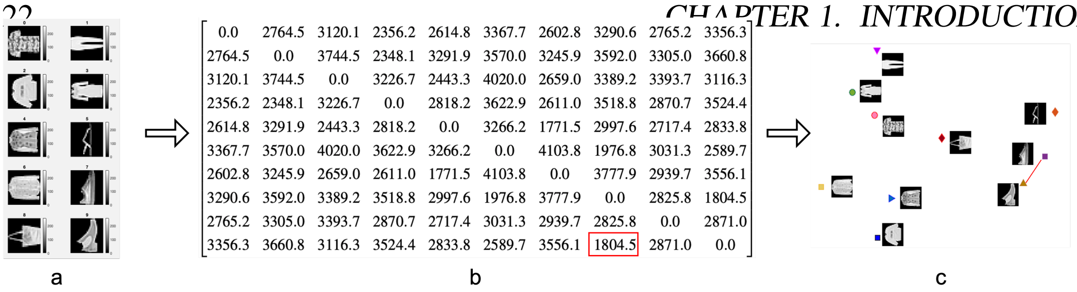
Introduction

1.1 Motivation

1.1.1 Cohort Data Visualisation

Visual analysis of data is an essential pre-processing step in many studies on machine learning applications, data mining, and pattern recognition [40, 84, 85]. A visual understanding of the data can help analysts to quickly identify and interpret critical data patterns and guide the selection of data processing, prediction, and pattern recognition algorithms, facilitating data scientists in many domains and applications. In most real-world problems, the dataset fed into prediction and pattern recognition algorithms is high-dimensional. For example, image data can be represented as a high-dimensional vector containing thousands of pixels; or textual data, usually represented by high-dimensional word-count vectors [85].

As human viewers can only perceive 2D or 3D ($2,3$-D) images, various visualisation techniques such as scatterplot matrices [65, 134] and parallel coordinates [55, 59, 139] have been introduced to present the information of high-dimensional datasets in $2,3$-D spaces. However, these approaches make no assumptions about the intrinsic structure of the data. They simply place the relevant information obtained directly in graphics. In order to efficiently and effectively extract useful information from high-dimensional data, dimension reduction (DR) methods are widely applied [40, 94]. These methods have numerous applications in many fields such as biology [30, 33, 150], chemistry [43] and medicine [4, 26].



$$\begin{bmatrix}
0.0 & 2764.5 & 3120.1 & 2356.2 & 2614.8 & 3367.7 & 2602.8 & 3290.6 & 2765.2 & 3356.3 \\
2764.5 & 0.0 & 3744.5 & 2348.1 & 3291.9 & 3570.0 & 3245.9 & 3592.0 & 3305.0 & 3660.8 \\
3120.1 & 3744.5 & 0.0 & 3226.7 & 2443.3 & 4020.0 & 2659.0 & 3389.2 & 3393.7 & 3116.3 \\
2356.2 & 2348.1 & 3226.7 & 0.0 & 2818.2 & 3622.9 & 2611.0 & 3518.8 & 2870.7 & 3524.4 \\
2614.8 & 3291.9 & 2443.3 & 2818.2 & 0.0 & 3266.2 & 1771.5 & 2997.6 & 2717.4 & 2833.8 \\
3367.7 & 3570.0 & 4020.0 & 3622.9 & 3266.2 & 0.0 & 4103.8 & 1976.8 & 3031.3 & 2589.7 \\
2602.8 & 3245.9 & 2659.0 & 2611.0 & 1771.5 & 4103.8 & 0.0 & 3777.9 & 2939.7 & 3556.1 \\
3290.6 & 3592.0 & 3389.2 & 3518.8 & 2997.6 & 1976.8 & 3777.9 & 0.0 & 2825.8 & 1804.5 \\
2765.2 & 3305.0 & 3393.7 & 2870.7 & 2717.4 & 3031.3 & 2939.7 & 2825.8 & 0.0 & 2871.0 \\
3356.3 & 3660.8 & 3116.3 & 3524.4 & 2833.8 & 2589.7 & 3556.1 & \boxed{1804.5} & 2871.0 & 0.0
\end{bmatrix}$$

Figure 1.1: Example of DR approach (MDS) applied to a subset of FMNIST dataset. The Figure a shows 10 image examples selected from the FMNIST dataset. The 10 × 10 matrix (Figure b) illustrates the L2-pixel distance between FMNIST images. Each row shows the distance between the image represented by this row and all other images. The Figure c scatter plot shows the embedding result using MDS method. The number enclosed by the red box is the distance between the nearest image and the 10-th image (the sneaker image in Figure a), which is reflected in the scatter plot as the smallest distance between their two representative 2-D points, as shown by the red line.

DR methods aim to transform high-dimensional data samples into low-dimensional points while preserving most of the intrinsic structure in the original dataset [87]. They can uncover similarities or dissimilarities between data samples and then retain them by locating embedded points in the targeted space, usually the $2, 3$-D space, for data visualisation purposes. For example, the Figure 1.1a shows image examples selected from the Fashion-MNIST (FMNIST) dataset [136]. The Euclidean distance between images (L2-pixel distance) can be calculated, which shows the dissimilarities between data samples. The embedded points can be derived by preserving the same distance relationships between data points, which involves attracting points whose original images has lower L2-pixel distance and repelling points whose images are quite different from each other. The Figure 1.1c is the scatter plot of the embedded points obtained by using classical multi-dimensional scaling (MDS) [15]. It can be observed that the L2-pixel distance between image slipper and image sneaker is the smallest of all the distances between the other images and image sneaker, as shown in the red box. Therefore, the embedding points between image slipper and image sneaker are also the closest in the scatter plot, as shown by the red line. From this 2-D plot relationships between the high-dimensional data samples can be inferred, so that the data set can be further processed and analysed.

The main research question of this thesis is: **What properties can be expected from a good DR method for data visualisation?**

To make this question more specific, this thesis focuses on visualising the cohort

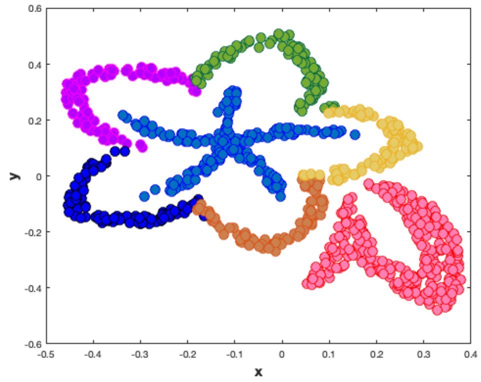
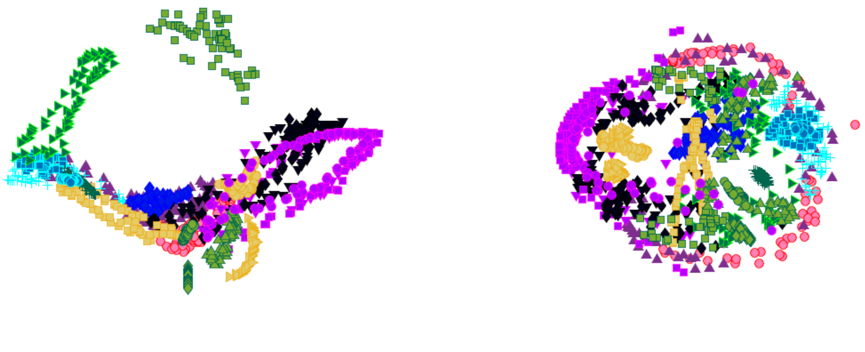


Figure 1.2: Example of synthetic 2-D flower dataset

data. The cohort data corresponds to the dataset containing either the natural clusters or clusters generated by clustering techniques. For instance, genome data includes data cohorts corresponding to different populations with different genetic variants [30], and the textual data 20-Newsgroup [16, 19, 20, 21] contains cohorts of news articles retrieved from different topic domains. Figure 1.1a shows that the FMNIST dataset comprises grayscale images of fashion products from ten categories, while Figure 1.2 displays a synthetic 2-D flower dataset that is manually assigned into 7 different cohorts.



(a) Isomap

(b) MDS

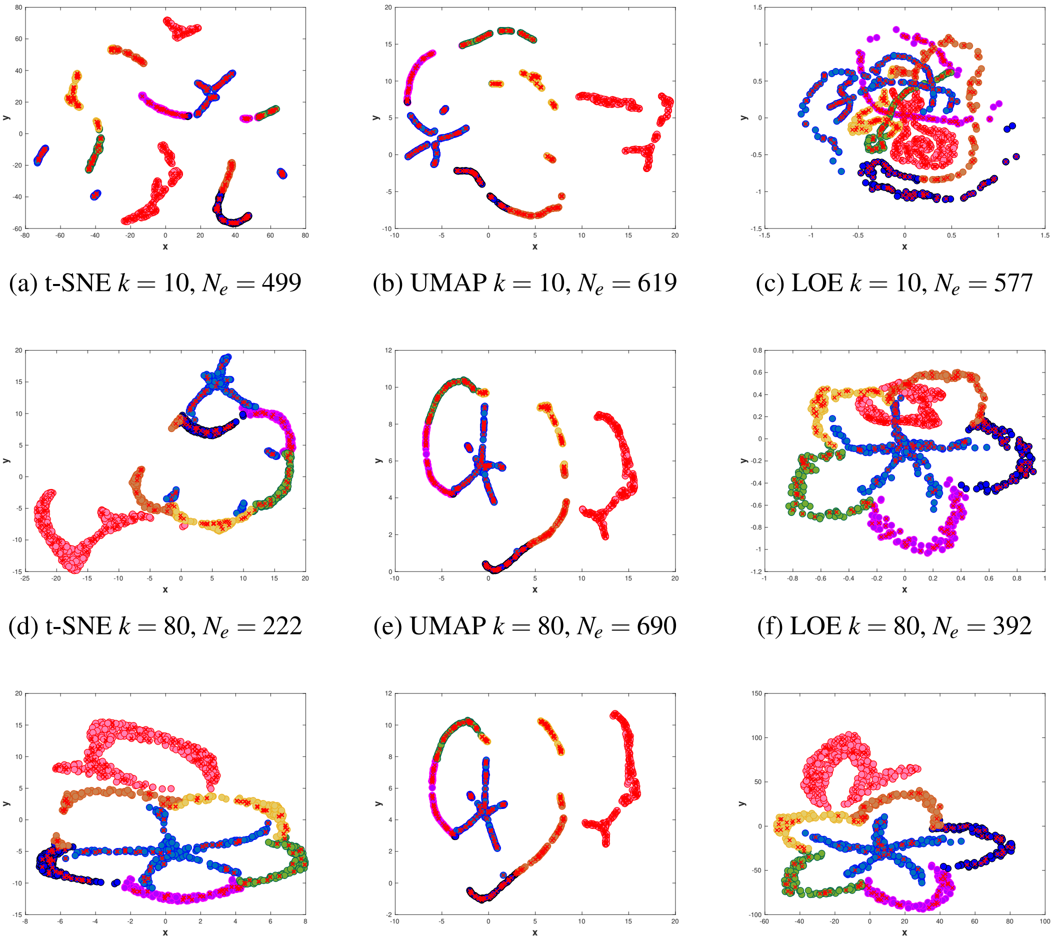
Figure 1.3: Embedding results of the Coil20 dataset using Isomap and MDS, respectively. The Coil20 dataset contains 20 different image data cohorts (introduced in Section 5.2.1). None of these two global metric approaches could successfully illustrate the structure of this cohort dataset.

1.1.2 Properties of a Good Visualisation

With regard the mechanism of the DR visualisation method, it can be straightforward to infer that the *first expected property* is that a *good* DR method for cohort data visualisation should obtain *good* point-based visualisation. This means, the user wish to construct an isometric DR mapping to embed data points so that the similarities or dissimilarities are perfectly unchanged after DR. However, in reality, it is not possible to isometrically embed any data with an intrinsic dimensionality more significant than 3 to a $2,3$-D space. Methods like MDS, non-metric MDS [3], soft ordinal embedding [116], Isomap and its variations [35, 41, 100, 115] try to retain all the distances or ordinal information of the dataset. However, as Figure 1.3 shows, for a fairly large number of data points with relatively high intrinsic dimensionality, the obtained results mostly do not reflect the original data structure. Compromise is unavoidable. Thus, a more detailed question related to the property of obtaining good visualisation can be proposed: **What are the main data patterns or structures to preserve in a 2,3-D spaces?**

Most mainstream approaches include these classical techniques of Laplacian eigenmap (LE) [12], t-distributed stochastic neighbour embedding (t-SNE) [121] and local ordinal embedding (LOE) [116], uniform manifold approximation and projection (UMAP) [90] attempt to preserve a local neighbourhood for each data point. The reason behind this is that the global topological structure of the data can be preserved if the correct local metric is preserved, which can be approximated by a suitable connectivity graph of k-nearest neighbours (k-NN). More specifically, the topological manifold $\mathcal{M}$ is defined as a second Hausdorff space where each point on the manifold has a small local Euclidean region [88]. The assumption of the approximate manifold $\mathcal{M}$ of the unstructured high-dimensional data is to be smooth and locally isometric, thus the connected graph or simplicial complex generated by k-NN is applied to reconstruct an equivalent topological representation in the low-dimensional space[60, 81, 88, 90]. Assuming that the reduced dimension is sufficient, there are theoretical guarantees provided for some local algorithms regarding their isometric mapping capability but under the assumption that a suitable choice of the neighbourhood size is given [109]. A guidance on neighbourhood size for LOE is larger than $O\left(\left(n^2 \log^d n\right)^{\frac{1}{d+2}}\right)$ if a set of n data points of dimension d is embedded [116].

To illustrate how sensitive this kind of the algorithms are to the neighbourhood



(a) t-SNE $k = 10$, $N_e = 499$

(b) UMAP $k = 10$, $N_e = 619$

(c) LOE $k = 10$, $N_e = 577$

(d) t-SNE $k = 80$, $N_e = 222$

(e) UMAP $k = 80$, $N_e = 690$

(f) LOE $k = 80$, $N_e = 392$

(g) t-SNE $k_{\min} = 120$, $N_e = 213$

(h) UMAP $k_{\min} = 60$, $N_e = 679$

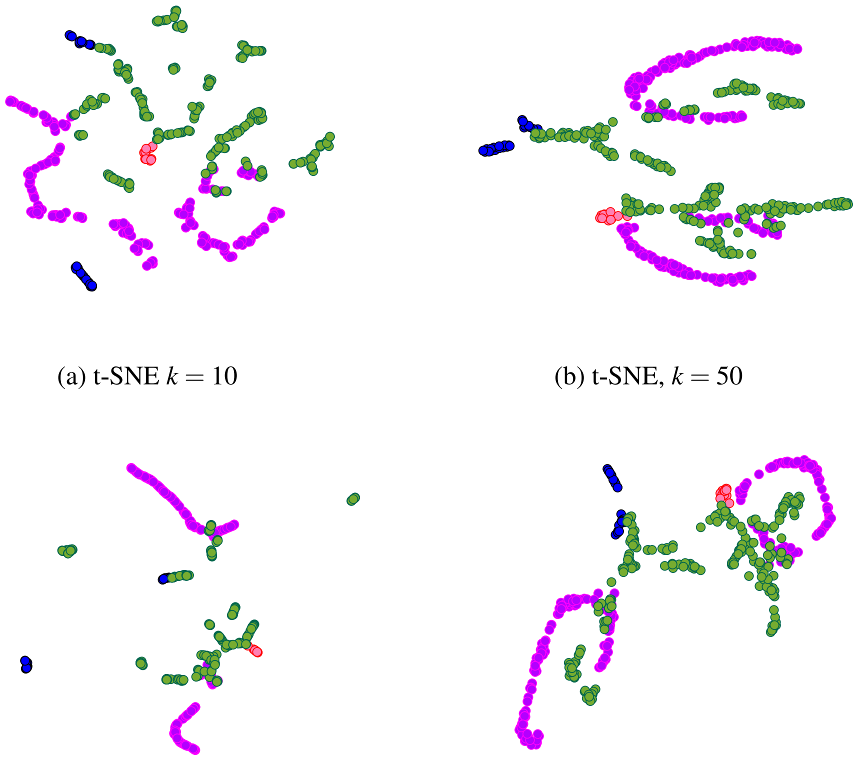
(i) LOE $k_{\min} = 150$, $N_e = 194$

Figure 1.4: Embed 2D Flower dataset in 2D space by t-SNE, UMAP, and LOE. The embedded points with at least one error neighbour in the local neighbourhood identified by k are highlighted by yellow crosses, and the number of such points N_e is reported as the 1-nearest neighbour preservation error.

size k, Figure 1.4 compares three representative techniques: t-SNE, UMAP, and LOE, using a set of $n = 1000$ data points of dimension $d = 2$ (flower dataset), for a small $k = 10$, a computed value by $k = \left\lfloor \left(n^2 \log^d n\right)^{\frac{1}{d+2}} \right\rfloor = 80$, and a tuned k by searching for the smallest integer that can mostly preserve the correct data shape, referred to as k_{min}. Here we embed 2-D data to 2-D dimension spaces, in order to illustrate more clearly the preservation of the structure and neighbourhood of different embedding techniques. That is because the unique property of the 2-D data that the figure of the original dataset can be easily visualised by analysts, which helps to make comparison and do analysis. Figure 1.4 shows that even using a small synthetic dataset, different embedding techniques have different k choices. Figure 1.5 also displays embedding results of 3-D bicycle dataset using t-SNE and UMAP under different k settings. These are only simple demonstrations for visualising synthetic data points that already show how hyper-parameter k influences embedding results. In high-dimensional data visualisation, it is difficult to know what the neighbourhood size should be to generate a reliable visualisation.

Instead of using the large neighbourhood size k to maintain the overall structure of the dataset, another trend of the visualisation approach is to utilise the property of the cohort data that each data point has its existing cohort label to display the cohort structure of the dataset. Many local algorithms have been extended to enhance cohort separability so that the between-cohort distance (dissimilarity) is maximised while the within-cohort distance (dissimilarity) is minimised in the reduced space, such as supervised LLE [143, 147], supervised t-SNE [27, 53, 91], and supervised UMAP (s-UMAP) [90]. As explained by Kobak et al. [72] and Mu et al. [92], the drawback of these algorithms is that the relative position between the embedded cohorts is rather random and it can be observed that they are also sensitive to the choice of the local neighbourhood size. This is illustrated by Figure 1.6, which presents results of s-UMAP using the same flower dataset as above but for two different runs with two different initialisations in their optimisation (Init1 and Init2) and different neighbourhood size (10, 80 and k_{min}).

It can be summarised that generating a *good* and meaningful visualisation for cohort data is not a trivial task. Ideally, it should demonstrate separation between data cohorts, preserve distances between not only individual data points but also cohorts, and illustrate some internal cohort structure. Most recent techniques like COVA [92], At-SNE [39] and PaCMAP [131] have been proposed to preserve both the local and



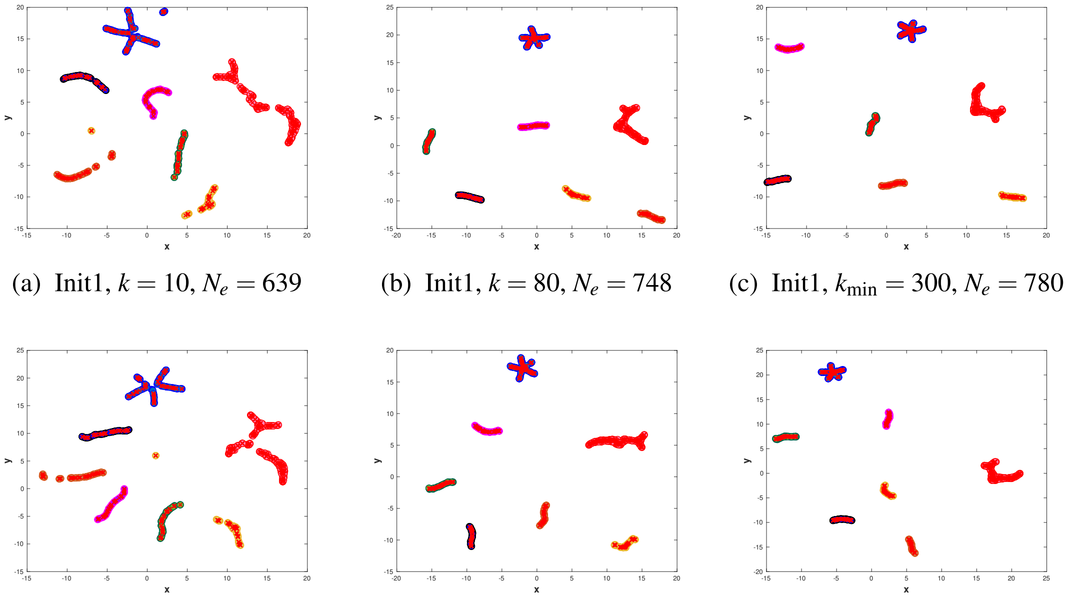
(a) t-SNE $k = 10$

(b) t-SNE, $k = 50$

(c) UMAP, $k = 10$

(d) UMAP, $k = 100$

Figure 1.5: The 3D bicycle dataset is embedded in 2D space by different k settings of t-SNE and UMAP.



(a) Init1, $k = 10$, $N_e = 639$ (b) Init1, $k = 80$, $N_e = 748$ (c) Init1, $k_{\text{min}} = 300$, $N_e = 780$

(d) Init2, $k = 10$, $N_e = 635$ (e) Init2, $k = 80$, $N_e = 774$ (f) Init2, $k_{\text{min}} = 300$, $N_e = 779$

Figure 1.6: Embed 2D Flower dataset in 2D space by s-UMAP. The embedded points with at least one error neighbour in the local neighbourhood identified by k are highlighted by red crosses, and the number of such points N_e is reported as the 1-nearest neighbour preservation error.

the global structure of the dataset. The common idea is to utilise the prototypes or anchor points or mid-near pairs of neighbours to first capture the cohort structure, and then refine local neighbouring points. However, COVA fails to maintain the internal cohort structure, while At-SNE and PaCMAP do not emphasise cohort separation for visualisation purpose.

To achieve the desired property of producing good visualisation of cohort data, we propose an ANchor GuidEd Local (ANGEL) algorithm. It is a multi-objective model that combines the advantage of unsupervised dimensionality reduction approach in preserving the local neighbourhood with the benefits of maintaining the internal cohort structure using anchor points. It is designed to provide a better balance regarding local neighbourhood preservation, cohort localisation and internal structure preservation, and adjustable cohort separation enhancement in a reduced space.

1.1.3 Visualisation Quality Evaluation

Once a DR algorithm can satisfy all targets of cohort separation, cohort positioning preservation, and local neighbourhood preservation, it is also essential to measure to what degree the method satisfies them. Classification accuracy is widely applied to measure the performance of visualisation algorithms [53, 90, 113]. Mu et al. [92] proposed the idea of using the 1-nearest neighbour classification rate to examine the compatibility between cohort memberships and the embedded result, which gives a reasonable cohort separability measurement. As the most local structure of the dataset is based on the local neighbourhood preservation, trustworthiness [66, 125, 126] and the S_N score proposed in [92] are introduced to measure the maintenance of k-NN from high-dimensional space to the target $2,3$-D space. However, for visualisation purposes, it is not necessary to keep every neighbouring point in the exact place, to retain the minor relaxation around the boundary.

As for the cohort distance preservation, which can also be treated as the global preservation of the dataset, most data visualisation approaches only discuss the visual effects of the embedded scatter plots. Only a few recent methods as [103, 131] put forward their measurements on global evaluation. However, they focus more on pairwise distance comparison over all data samples and do not emphasise the preservation of the cohort distance.

A new system of evaluation metrics needs to be designed that can take into account all three aspects (i.e., separability, local neighbourhood preservation and global cohort positioning preservation) simultaneously and give convincing results.

1.1.4 Incremental Visualisation

The vast majority of discussed DR visualisation methods operate in a batch mode, which means that they cannot handle incremental data points, and all the data samples are required during the embedding process [76]. Once the given data samples are all embedded in the targeted $2,3$-D space, extending the mapping to new data samples is challenging. Whenever a new data sample arrives sequentially, to obtain a new embedding result, most of the metric methods like t-SNE and UMAP mentioned above need to repeat the running of the "batch" version on the "new" dataset, which contains the newly added data. This process is time-consuming and wastefully discards the pre-generated embedding results [42, 79].

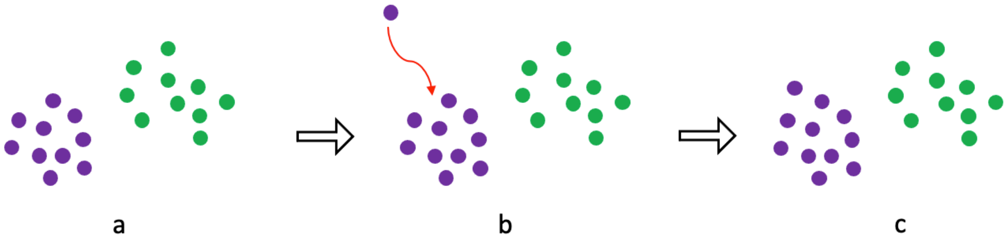


Figure 1.7: A brief explanation of incremental embedding. Image a shows the batch embedding result of two data cohorts, namely purple and green cohorts. A new data sample belonging to the purple cohort comes in Image b. Thus, Image c shows the results after embedding the new data sample.

A *good* DR method for cohort data visualisation is expected to possess the ability of incrementally embedding new data samples. Figure 1.7 shows a brief example of how incremental embedding works. COVA proposes the COVA-projection algorithms which map the newly added data sample to the visualisation space by learning parametric mapping functions. Similar to the idea of COVA-projection, kernel t-SNE [49] finds a model-based mapping matrix to embed out-of-sample data. The parametric t-SNE [118] and parametric UMAP [103] techniques utilise neural networks to train the batch mode and treat each incoming sample as test data. These extension methods locate new observations based on the assumption that the batch results are correct. They may fail if the batch dataset is not uniformly sampled [42].

Another group of incremental embedding approaches, including incremental LLE [73], incremental Isomap [76], and progressive UMAP [70], not only process the newly added data samples but also update the known batch embedding points, to provide more reliable results. However, as most of these incremental extensions are based on local metric algorithms, they usually inherit a majority of disadvantage from the original methods. Moreover, there is no current research on incremental ordinal embedding algorithms, which is a research gap that more research needs to be conducted to fill.

1.1.5 Summary

In this thesis, ANGEL algorithm is proposed to have a better balance in local neighbourhood protection, cohort positioning and internal structure protection, and to provide adjustable cohort separation enhancement in a reduced space. A novel evaluation approach is also proposed to measure the performance of DR algorithms according to these properties. The incremental extension of ANGEL is addressed by proposing a simple but intuitive p-ANGEL and i-ANGEL algorithms. They can save the memory cost while doing the embedding process and accelerate the computational speed, making ANGEL more applicable to real-world cohort data visualisations.

1.2 Research Questions, Hypothesis, and Objectives

Considering the research background and the motivations shown in Section 1.1, the general research question is proposed as "**What properties can be expected from a good DR method for cohort data visualisation?**". Since the strategy of DR algorithms for visualisation is to transform high-dimensional data samples to data points in $2,3$-D spaces while preserving the intrinsic structure of the dataset, two questions can be raised as follows: "**What are the main data patterns or structures to preserve in the $2,3$-D spaces?**" and "**What are limitations and unsolved problems of the recent DR methods for preserving the data patterns or structures?**".

A comprehensive analysis of the existing DR approaches is required, and their crucial strength and weaknesses of their performance in data structure preservation should be fully considered and discussed. Then, a question can be raised naturally as "**How could we design a DR method to avoid limitations and to achieve a better performance compared with state-of-the-art approaches?**". Another relevant and critical question is “**How could we measure the performance of a method in preserving the data patterns or structures?**”. To answer these questions, it is required to propose visualisation algorithms and corresponding measurements based on the properties of the algorithms. Moreover, the results obtained from the proposed measurement are expected to satisfy the characteristics of existing state-of-the-art methods at the same time.

In addition, the problem of big data is recognised as a challenging issue. In many cases, embedding large dataset is both memory and time consuming. Therefore, a related question has arisen, “**Is that possible to improve the previously proposed competing DR approach so that it has the potential to process big datasets?**”

Moreover, as most popular state-of-the-art methods cannot deal with the new coming data point once the embedding process has been finished, another related question arises, “**Is it possible to extend the previously proposed competing DR methods to make it potentially feasible to handle incoming data points consistently?**”

Based on these research questions, the hypothesis of this thesis can be proposed as the following:

- The embedded points in the $2,3$-D spaces for visualisation purposes should demonstrate separation between data cohorts, preserve distances between individual data points, retain cohort positioning and the cohort’s internal structure.
- The proposed evaluation metric should be able to measure the cohorts’ separation, distance preservation between data points, and distance preservation between cohorts.
- The further improved methods should be able to process the large dataset, reducing both time and memory consumption.
- The further improved methods should be able to process new data points consistently and have the new embedding results satisfy the same properties as the first hypothesis outlined.

Respectively, the research objectives of this thesis can be divided into the subobjectives stated below:

1. **Propose a strategy to retain cohort positioning.**

Similarities or dissimilarities between cohorts can be obtained by applying cohort proximity calculation approaches [110, 111] to the cohort data. The positions of cohorts in the visualisation space should be derived by a global-based embedding method that preserves most of the information regarding similarities or dissimilarities retrieved from the high-dimensional space. The generated cohort positions should be represented accurately in a position with the respect to the related cohorts, which is also the basis of preserving the internal structure of cohorts as shown in Section 4.2

2. **Propose a strategy to preserve internal structure of cohorts.**

   Based on the known cohort label information, anchor-based supervised learning algorithms [86, 129, 142], Landmark MDS [29], COVA, and At-SNE use the idea of utilising prototypes or anchor points as the skeleton of the cohorts' structure. Based on this property of anchor points, corresponding embedded points should also be generated in the $2,3$-D space. The embeddings should also have the ability to represent most corresponding similarities or dissimilarities between high-dimensional anchor points to preserve the global structure of datasets. Moreover, the embedded anchor points should also retain the positions of cohorts. Section 4.3 and 4.4 illustrate how we achieve this objective.

3. **Propose a strategy to enhance the separation between cohorts.**

   Supervised DR approaches for visualisation utilise the cohort label information to enhancing the separation between cohorts. The basic idea is to enlarge the dissimilarities of between-cohort data samples and emphasise the similarities of intra-cohort data samples. As anchor points are applied to control the global structure of the cohort data, the ability to adjust the degree of separation between anchor points belonging to different cohorts is required in the process of embedding. Section 4.4 provides a solution to this objective.

4. **Develop an algorithm to simultaneously to achieve the properties of enhancing separation between data cohorts, preserving distances between individual data points, and retaining cohort positioning and the cohort's internal structure.**

   Common local embedding algorithms [12, 57, 102, 121] can be employed to keep local neighbouring points around each data point. The goal is to simultaneously achieve all three aspects (cohort separation, cohort positioning preservation and local neighbourhood preservation) expected from the first hypothesis. Taking the advantage of the existing multi-objective algorithm from COVA and PaCMAP, this thesis aims to design an original algorithm to achieve this objective. In addition, less memory consumption, faster optimisation and accelerated approximation of the new embedding algorithm should be investigated and designed. Section 4.4.2, 4.5, and 6 show how we set up the model to achieve the goal and how we improve the model.

5. **Develop evaluation approaches to measure the performance of the embedding algorithms for visualisation.**

   As it is expected that the proposed DR algorithm can achieve three different aspects simultaneously, the performance of these three aspects should also be measured concurrently and quantitatively. The proposed evaluation approach should be able to balance three numerical scores based on the measurement of cohort separation, cohort positioning preservation and local neighbourhood preservation. Section 5 explains the idea of new evaluation techniques.

6. **Developing an incremental extension of the proposed DR algorithm for visualisation.**

   To incrementally embed newly added data samples, the extension of the proposed DR algorithm should be able to first estimate the similarities or dissimilarities between the new data sample and the batch dataset, as well as the relationship between the new data sample and the generated anchor points. Then, the embedding point along with the updated batch points should be computed. Moreover, the result of the incremental extension embedding approach should not deviate unduly from the result obtained by running the original algorithm on the whole dataset directly. Section 7 describes our model improvements and practical solutions.

1.3 Contributions

The main contributions of this thesis are outlined as follows:

1. The ANGEL algorithm, which is a bi-objective model that combines the local objective function with the global objective function, is proposed to cope with the limitations that most state-of-the-art DR approaches have: the inability to make the embedding results meet the desired expectations (local neighbourhood preservation, cohort positioning preservation, and cohort separability) simultaneously. The proposed algorithm addresses these issues by injecting the global distance structure into a local ordinal distance structure by controlling a set of easily generated anchor points for each cohort and the embedded points in $2, 3$-D spaces (Chapter 4).

2. A novel evaluation metric was proposed, which provides a quantitative measure of embedding results and allows comparison of the performance of different DR methods. Evaluation results confirmed intrinsic characteristics of the existing algorithms, and illustrated that ANGEL could obtain improved overall performance on local neighbourhood preservation, cohort positioning, and separation compared with state-of-the-art approaches (Chapter 5).

3. Variations of the ANGEL algorithm were proposed to obtain approximate embedding results via novel empirical faster optimisation approaches. In addition, another variation of ANGEL was also proposed, which substitute the local objective function with other local metric algorithms, illustrating the feasibility of applying the ANGEL framework to a popular metric embedding algorithm (Chapter 6).

4. Two extensions of the ANGEL algorithm were proposed in order to tackle the following two issues: reducing the memory consumption (p-ANGEL), reducing the time consumption and enabling the incremental embedding (i-ANGEL). The experimental results demonstrated the effectiveness of applying p-ANGEL and i-ANGEL on real-world cohort datasets for the visualisation purpose (Chapter 7).

1.4 Structure of the Thesis

Here is a summary of the rest of this thesis:

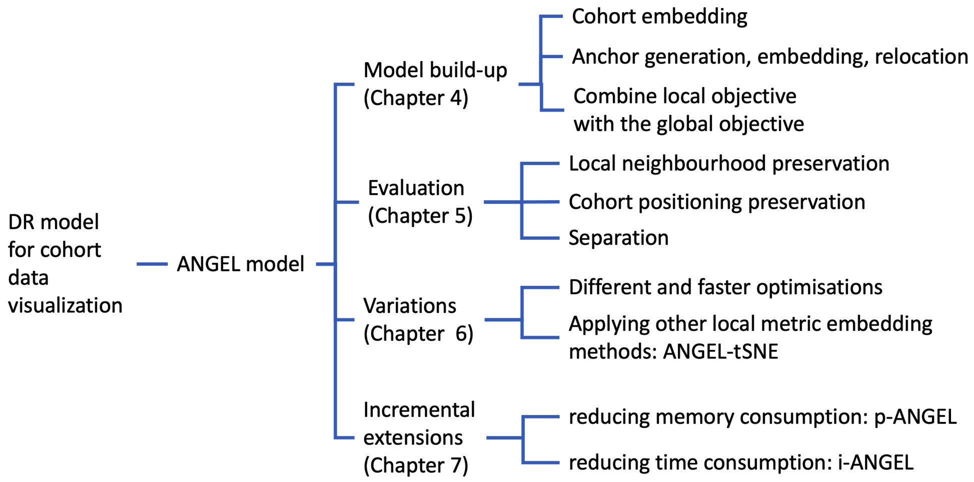

DR model for cohort data visualization
ANGEL model
Model build-up (Chapter 4)
Cohort embedding
Anchor generation, embedding, relocation
Combine local objective with the global objective
Evaluation (Chapter 5)
Local neighbourhood preservation
Cohort positioning preservation
Separation
Variations (Chapter 6)
Different and faster optimisations
Applying other local metric embedding methods: ANGEL-tSNE
Incremental extensions (Chapter 7)
reducing memory consumption: p-ANGEL
reducing time consumption: i-ANGEL

Figure 1.8: Illustration of anchor embedding using the FMNIST dataset.

Chapter 2 reviews the various prior works on DR approaches for data visualisation. DR methods can be categorised into several groups based on different purposes: local neighbourhood preservation, global information preservation, and multi-objective preservation. Furthermore, a critical discussion on the limitations of the existing works is provided. The rest part of this chapter gives a systematic review of incremental DR methods.

Chapter 3 provides a more detailed technical background which facilitates the understanding of the model proposed in this thesis. In addition, the embedding results of some classical and popular DR methods are examined here, and the explanations are given regarding why such methods are used in the proposed model.

Chapter 4 introduces the construction process of the proposed ANGEL algorithm. It starts with the cohort embedding process to locate the cohort positioning in $2,3$-D spaces. Then anchor points are generated, embedded, and relocated to maintain the intrinsic structure of the cohort while preserving the cohort position. Finally, data points are embedded regarding the relations between data samples and anchor points while maintaining the local neighbourhood of each data sample, leading to the construction of the bi-objective cost function. Different optimisation approaches are also discussed in solving this bi-objective cost function.

Following from Chapter 4, Chapter 5 proposes a novel evaluation metric to measure embedding results and compare the performance of different DR methods. Experiments are conducted to compare ANGEL with the state-of-the-art algorithms. Furthermore, a parameter study of the ANGEL model is also conducted and discussed in this chapter.

Chapter 6 presents variations of the ANGEL model. This chapter is divided into 2 parts. The first part focuses on applying different novel empirical optimisation approaches to the ANGEL algorithm to obtain fast approximation results. Experiments are conducted and the results show the advantage of these proposed optimisation methods. The second part illustrates the feasibility of applying the ANGEL framework to a popular metric embedding algorithm. The local objective function is substituted with another local embedding cost function as ANGEL-tSNE. Experiments are conducted to compare the results obtained by ANGEL-tSNE and ANGEL.

In Chapter 7, the proposed ANGEL algorithm is extended to reduce memory consumption, time consumption, and make it can continuously process the new coming data samples. The p-ANGEL algorithm targets at reducing memory consumption. The i-ANGEL is the incremental DR algorithm, reducing the time consumption. Experimental results show the effectiveness of applying p-ANGEL and i-ANGEL to real-world cohort datasets for visualisation.

Conclusions and future works are presented in Chapter 8.

Chapter 2

Literature Review

Dimension reduction (DR) is a widely used technique in the machine learning field that reduces the dimensionality of high-dimensional data to a meaningful target space with minimal information loss. The main focus of this chapter is to review recent research advances related to DR algorithms for data visualisation.

For data visualisation purposes, the target space is often a 2-D or 3-D $(2,3\text{-D})$ spaces. To form the question, a high-dimensional data $\mathbf{X} \in \mathbb{R}^{n\times D}$ is considered with n observations and D features (dimensions), and $\hat{\mathbf{X}} \in \mathbb{R}^{n\times d}$ represents a set of low-dimensional points where $d = 2$ or 3. Each $\hat{\mathbf{x}}_i$ can be treated as the corresponding embedding point of the data sample $\mathbf{x}_i$ obtained by applying DR methods. The general goal is to find the best $\hat{\mathbf{X}}$ that retains most of the given dataset $\mathbf{X}$'s information.

The manifold learning discusses the DR from another perspective, which assumes the observed high-dimensional data $\mathbf{X}$ to lie on an embedded sub-manifold of the high-dimensional space [109, 149]. Intuitively, the DR approaches finds the underlying low-dimensional space of the sub-manifold and recovers the manifold structure of the $\mathbf{X}$ in that low-dimensional space. The classic DR methods focus on linear dimension reduction and deal with data with the assumption of lying on a linear manifold [76, 88]. The manifold can be visualised as a line, a plane, or a hyperplane, depending on the number of dimensions involved [88]. Examples include Principal component analysis (PCA) [135] and multidimensional scaling (MDS) [15]. However, such methods often fail when dealing with the dataset containing a non-linear structure [114].

Some proposed non-linear dimension reduction methods (NLDR) focus on non-linear manifold learning problems. The classical NLDR approach, like isometric feature mapping (Isomap) [115] mainly keeps the global geometry of the non-linear manifold, while locally linear embedding (LLE) [102] seeks to map neighbourhood points on manifolds to neighbourhood points in lower dimensional spaces [109]. Thus, based on a similar categorisation idea, we can classify the popular methods into two categories, often referred to as local and global approaches. The crucial difference between the two categories is whether the algorithm aims to preserve the local or global structure of the high-dimensional dataset.

In general, approaches that focus on local structure preservation seek to preserve the *neighbours* of each point in a high-dimensional dataset when embedding the dataset in a $2,3$-D space, and this can be achieved with the preservation of distance information or ordinal information among neighbours. To supplement this, neighbourhood points are a group of points that are small in distance or ordinal ranking from the target point and are all within a given range. As the approach focuses on global structure preservation, it will place equal importance on its purpose of preserving distances or ordinal relationships between the target point and more distant points from the target point. In addition, for the visualisation of cohort data, the global approach will focus more on preserving distances between cohorts instead of distances on points.

There are many DR approaches to conducting research in these two directions, and both have led to awe-inspiring achievements. Recent research works will also address both directions, implementing as much functionality as possible. This chapter reviews each of them in Section 2.1, 2.2 and 2.3. Since our proposed DR algorithm does not give the preliminary manifold assumption of the $\mathbf{X}$, we will not involve mathematical terminology of the manifold learning. However, this will be involved in the future direction of our theoretical analysis of our proposed approach.

The methods will be discussed in both directions have a common disadvantage: they are necessarily only applicable to the batch mode. This means that these methods only work for a fixed set of data samples but do not support progressively increase the novel data samples. There have been many attempts to develop incremental algorithms based on their batch mode. Section 2.4 offers an overview of incremental DR methods. These studies offer great theoretical support and assistance.

Category	Metric Embedding	Ordinal Embedding	Supervised Embedding
Description	Preserving metric distances between data samples in the target $2,3$-D spaces	Preserving ordinal information derived from data samples in the target $2,3$-D spaces	Utilising cohort label information to help to enhance the separation of the embedding results
Strength	Straightforward and easy to implement	Do not rely on explicit input representation or similarity calculation	Enhance cohort separation, make cohorts to be better identified
Weakness	Parameters and initialisation need to be carefully selected, loss of global structure		Loss of local neighbourhood preservation
Methods	LLE [102], LE [12], LMDS [24], SNE [57], t-SNE [121], EE [22], LargeVis [113], UMAP [11], LDP [54], etc.	STE [123], t-STE [123], LOE [116], t-ETE [5], etc.	Supervised LLE [128, 143, 147], supervised t-SNE [27, 53, 68], Supervised UMAP, SLE-ML [112], etc.

Table 2.1: Overview of approaches focusing on local structure preservation

2.1 DR Methods for Local Structure Preservation

One primary approach of DR for data visualisation is the preservation of the local structure of the dataset. The most straightforward way is to maintain the relations between the target data sample $\mathbf{x}_i$ and its nearby data samples. Such relations can be categorised in the following directions: the distance and the ordinal ranking between data samples, as displayed in Table 2.1. The main difference between these two directions is whether they utilise an explicit similarity calculation during the embedding process. The metric DR methods will be reviewed in Section 2.1.1 and ordinal DR methods in Section 2.1.2.

Since DR approaches reduce the dimensionality of the high-dimensional dataset sharply to the $2,3$-D spaces, attempting to represent the neighbourhood relationships in $2,3$-D spaces that one desires to retain may result in mutual coverage between data cohorts. Supervised DR methods are proposed to enhance cohort separability and identify data cohorts better. Section 2.1.3 reviews the supervised extensions developed based on the popular DR visualisation method.

2.1.1 Local Metric Preservation Approaches

A straightforward way to preserve the local metric relationships between data samples is to preserve the coordinate relations between the target data sample $\mathbf{x}_i$ and its surrounding neighbourhood points. Locally linear embedding (LLE) [102] analysed the local symmetries of the dataset to discover nonlinear structure in high dimensional data and proposed a linear reconstruction idea for constructing $2,3$-D embedded points. The strategy of this algorithm is to reconstruct every data sample as a linear combination of its nearest neighbours. Thus, each embedded point $\hat{\mathbf{x}}_i$ is found to be the best $2,3$-D representations retaining the linear combination weights derived from the high-dimensional space. However, it may suffer from ill-conditioned eigen-problems and is sensitive to noise and control parameters [23].

The modified locally linear embedding (MLLE) [145] was proposed as an extension of LLE, which tries to overcome these drawbacks. It works by introducing multiple linearly independent weight vectors for each point in the reconstruction process. Hessian Eigenmaps [32] is also a variant of the LLE algorithm. Hessian Eigenmaps applies the LLE strategy but used the Hessian matrices.

Laplacian Eigenmaps retain the local data structure by keeping the similarities between neighbouring data samples $\mathbf{x}_i$ and $\mathbf{x}_j$. The similarity can be the Euclidean distance calculated between the high-dimensional data samples and weighted by kernel functions that emphasise the similarity of close samples. Then, the embedded $\hat{\mathbf{x}}_i$ will be constructed to maintain the relative position of neighbouring points based on these weighted similarities. However, it emphasises samples in the local vicinity and pays less attention to distant samples. Recently, Local linear laplacian Eigenmaps (LLLE) [83] combined the idea of the LLE and LE to enhance the robustness and improve the local structure preservation. Local distances preserving (LDP) [54] is also proposed, aiming at preserving local distances inspired by LE and LLE.

Local multidimensional scaling (LMDS) [24] was proposed as a local extension of the classical multidimensional scaling (MDS) approach, which is a global method that will be discussed in Section 2.2.1. The LMDS algorithm restricts the stress function to neighbouring points which share small distances. It also inherits the generality of the MDS that the metric distance will be directly preserved by embedded points in the $2,3$-D space. However, when dealing with the real-world dataset, keeping the actual distance calculated in the high-dimensional spaces is difficult to achieve at the same time.

Stochastic neighbourhood embedding (SNE) [57] and t-distributed stochastic neighbor embedding (t-SNE) [121] converts metric distances between samples into probability distributions. The goal is to create the $2,3$-D representations by minimising the similarity of the probability distributions of the high and low dimensional space using the Kullback-Leibler convergence function. SNE applies Gaussian distribution in both high-dimensional space and $2,3$-D spaces. At the same time, t-SNE utilise the Gaussian distribution in the high-dimensional space and the student-t distribution in the $2,3$-D spaces. SNE suffers from asymmetric loss function and crowding problems, and t-SNE can be seen as a successful extension of SNE, which handles these problems.

The t-SNE approach has inspired numerous further studies and discussions. Arora et al. [8], Kobak et al. [71], and Wattenberg et al. [133] gives discussions on the effective usage of t-SNE, and also gives an analysis of the intrinsic strategy of the algorithm. Elastic embedding (EE) [22], which combined the idea of SNE with the idea of LE, was proposed to enhance the robustness of the SNE embedding results. LargeVis [113], viSNE [7], BH-SNE [120], and FIt-SNE [82] have further improved t-SNE to accelerate the convergence and offer a better fit to larger datasets.

Uniform manifold approximation and projection (UMAP) is another popular local metric DR approach proposed in recent years. It use a similar strategy as t-SNE, as it first constructs a weighted graph of nearest samples where weights denote probability distributions. It then optimises the graph layout in the low-dimensional space based on the known graph derived from the high-dimensional space. UMAP is proved to perform well on biometric data visualisation [10, 11, 33]. However, both t-SNE and UMAP have their disadvantages in sensitivity to hyper-parameter selection and initialisation of the algorithm [72].

In general, these DR approaches focus on local structure preservation. They can be applied to the cohort dataset with label information and unlabelled data samples. Section 3.1 gives a technical description of the t-SNE method, which facilitates the proposed ANGEL variations on other local metric embedding strategies.

2.1.2 Local Ordinal Preservation Approaches

Ordinal embedding is a kind of DR problem without an explicit input representation of given data samples, or a similarity function between pairs of items [124]. For example, for multimedia data, it is difficult for people to give a mathematical description of music in order to state whether they think two songs are similar or not. Despite, they can provide their inference that song A is more similar to song B than song C. Such comparison formulates the basis of the ordinal embedding; that is, given a set of triplets (i, j, l), each refers to the comparison relationship that $\mathbf{x}_i$ is closer to $\mathbf{x}_j$ than to $\mathbf{x}_l$, the goal is to find a set of $2,3$-D points which satisfies as many of these triplets as possible.

Stochastic triplet embedding (STE) [123] and t-distributed stochastic triplet embedding (t-STE) [123] are proposed to handle the ordinal embedding problem with reference to the strategies of SNE and t-SNE. They introduce the probability of measuring whether the distance between the embedded points satisfies the corresponding triplet and then maximising the sum of log probability over all triplets. Violations of nearby points are emphasised during this process. STE adopts the Gaussian distribution, and the t-STE utilise the heavy-tailed student-t distribution.

The t-exponential triplet embedding (t-ETE) [5] inherits the heavy-tailed property of t-STE but proposes a transformation part to reduce the sensitivity to noisy triplets. The recently proposed Trimap [6] assigns weights to target triplets to reflect the relative similarities using the embedded points. Trimap also introduces a triplet selection process where triplets with closer point $\mathbf{x}_j$ belong to the set of nearest-neighbours of the point $\mathbf{x}_i$ and the distant point $\mathbf{x}_k$ is among the farthest points of $\mathbf{x}_i$ and chosen uniformly at random. This selection process also shows that it places more priority on preserving neighbourhoods.

Local ordinal embedding (LOE) [116] was proposed to make comparisons between data samples much easier and focuses more on local neighbour preservation. It simply interprets the triplet (i, j, l) as $\mathbf{x}_j$ belongs to the neighbourhood of $\mathbf{x}_i$ while $\mathbf{x}_l$ is not. The number of triplets is reduced, which accelerates the optimisation process. Moreover, the neighbourhood preservation strategy gives a more accurate view of the neighbourhood graph as local metric DR approaches, which is easier to construct and compare with metric DR methods.

The technical detail of the LOE method is given in Section 3.3, which facilitates the proposed model of the local metric preservation. In addition, Section 4 will explain the strategy of applying the LOE algorithm to the ANGEL model.

2.1.3 Supervised DR approaches

Traditional DR methods for visualisation usually process the data without considering the label information. However, in many real-world cases, these popular approaches may not be beneficial because of the overlapping between different data cohorts when the dimensionality of the data samples has been significantly decreased. Therefore, introducing label information to DR methods can improve the separation between data cohorts, making the embedding results easier to identify. Moreover, because of the enhanced separability, the supervised DR method also improves the classification accuracy, leading to more applications in the machine learning field [53].

The main difference between supervised and unsupervised DR techniques is whether to conduct the step of deriving similarity or dissimilarity between data samples. Instead of directly using distance measurements like Euclidean distance, dissimilarity measures applied in supervised DR approaches enforce the same class data samples to be close and different class data samples to be far away [52]. After obtaining the dissimilarities, the rest of the embedding process is the same for both unsupervised and supervised DR methods.

There are many supervised locally linear embedding (SLLE) [128, 143, 147] proposed as extensions of LLE aiming at classification problems, as well as the Supervised Laplacian Eigenmaps for multi-label datasets (SLE-ML) [112]. The goal is to find the embedding separating a within-cohort structure from a between-cohort structure [143]. The core idea is to introduce a scale parameter that controls how faithfully the data sample $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to the same cohort. In other words, if $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to a different cohort, the distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ would not change. However, if $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to the same cohort, then the similarity is enhanced by multiplying a scaling parameter.

A similar idea can be applied to the t-SNE algorithm, developing a set of supervised t-SNE approaches [27, 53, 68]. The supervised UMAP[1] also aims at capturing the structure of the high-dimensional dataset. However, enhancing the intra-cohort data relations will change the original similarity between data samples. Therefore, the embedding result derived by a supervised version of the local DR method could not

[1] https://umap-learn.readthedocs.io/en/latest/supervised.html

Category	Metric Embedding	Ordinal Embedding
Description	Aim to preserve metric distances between *all* data samples in the target $2,3$-D spaces	Aim to preserve *all* ordinal information derived from data samples in the target $2,3$-D spaces
Strength	Retain the overall structure of the high-dimensional dataset	
Weakness		Huge Time consumption
	Low local neighbourhood preservation, overlapping between cohorts	
Methods	MDS [15], Isomap [115], LTSA [146], PySef [95], etc.	GNMDS [3], SPE [106], Partial Order embedding [89], SOE [116], etc.

Table 2.2: Overview of approaches focusing on global structure preservation

represent the real structure of the dataset. Doubly supervised t-SNE [68] was proposed to highlight both the pre-given class information and the original intrinsic structure of the cohort.

The supervised DR approach focuses more on enhancing the separability between cohorts, so the cohort's position is somewhat random. Moreover, local neighbourhood preservation also drops as a sacrifice since it changes the similarity between data samples.

2.2 DR Methods for Global Structure Preservation

Compared to the local preservation approach, which only preserves neighbourhood information, the global preservation approach prefers to maintain the relationships between *all* points in the dataset. The global metric techniques aim to keep all the metric distances, while the global ordinal approach holds all ordinal information. These methods are reviewed in the following sections and summarised in Table 2.2.

One classical DR approach which has not been mentioned in Table 2.2 is principal component analysis (PCA) [135]. It is a linear mapping technique that projects all data samples along the axis with the largest variance of the dataset. However, real-world datasets such as Coil20 [17, 18] have complex intrinsic structures, leading to the failure of PCA to project data samples in $2,3$-D spaces. In most cases for data visualisation, PCA is often applied as a pre-processing step to obtain the initialisation of the local

DR methods [72].

2.2.1 Global Metric Preservation Approaches

Multidimensional scaling (MDS) refers to a group of classical dimension reduction techniques which are based on the aim of distance preservation [77]. Classical MDS [15] preserves the original Euclidean distance between all pair-wise high-dimensional data samples in the target $2,3$-D spaces. Variants for MDS are often based on different metric distance calculations. To reduce the matrix computation complexity, landmark MDS [29] is one of the early works that apply landmark points (anchor points) on embedding problems. This two-phase approach utilises landmarks to process large-scale datasets. It first embeds the landmarks and then embeds data points based on the transformation between data samples and landmarks. However, as a linear approach, MDS has limited capacity to capture the structure of the nonlinear datasets.

As for topology considerations, Isomap [115] replaces the Euclidean distance in MDS with a geodesic distance to reveal the global structure of the dataset, trying to assemble local geometry properties learned from the data. In other words, pair-wise geodesic distances between data samples are calculated by applying the shortest path algorithm through the k-Nearest neighbour graph constructed using Euclidean distance. Further studies on Isomap [9, 35, 41, 100] discuss and enhance the topology preservation in the low-dimensional space. Many supervised Isomap extensions introduced in [44] are also proposed to enhance the cohort separability by enlarging between-cohort dissimilarity compared with within-cohort dissimilarity. Li et al. [78] improves the supervised Isomap by reducing the sensitivity to noise data, and Zhang et al. [144] introduces labelled data with unlabelled data as a semi-supervised extension of Isomap.

Local tangent space alignment (LTSA) [146] proposed that a local tangent space is constructed for each data point, and a global low-dimensional embedding is obtained by affine transformation of the local tangent space. It starts with the local information extraction but ultimately preserves the local geometry as much as possible in the global coordinates of the embedded points.

PySef [95] further learns the dataset's global structure through a similarity matrix, which is formulated using data annotations. It defines the general similarity framework and can derive many existing DR approaches like MDS or generate a novel DR approach by appropriately setting the target similarity matrix.

In general, global metric approaches preserving pairwise distances or similarities between all data samples. However, they have a significant loss in terms of neighbourhood preservation. Moreover, they do not handle high data dimensionalities and a large number of data points well when drastic dimension reduction (to 2,3 D) is required.

2.2.2 Global Ordinal Preservation Approaches

Instead of retaining all metric distances between data samples, global ordinal preservation approaches preserve all ordinal (ranking) information derived from the high-dimensional dataset. As described in Section 2.1.2, ordinal DR methods aim to find a $2,3$-D embedding that satisfies the known triplets or quadruples that stores relations between data samples.

The optimisation problem of the general OE method was first proposed by [75, 107, 108]. Generalized non-metric multi-dimensional scaling (GNMDS) [3] proposes a semi-definite program to solve the ordinal embedding problem based on the given set of triplets. The embedding result is optimised over the generated Gram matrix. However, it could not handle large datasets due to the computational complexity.

Structure preserving embedding (SPE) [106] focuses on preserving the global topology structure of the high-dimensional dataset. If connectivity algorithms, such as k-nearest neighbours, can easily recover the edges of the input graph from the coordinates of the nodes only after embedding, then the topology is preserved. Thus, SPE captures the connectivity of the graph of the input dataset, and finds the $2,3$-D points to reveal the same properties. More applications such as [37] and [137] utilise the network nodes to learn the global topology, which also offers inspirations for topology preservation.

Partial order embedding [89] utilises the set of quadruples to formulate the relations between data samples. Similarly to the triplet discussed in Section 2.1.2, quadruples can be constructed as (i, j, k, l) where sample $\mathbf{x}_i$ and $\mathbf{x}_j$ are more similar than sample $\mathbf{x}_k$ and $\mathbf{x}_l$. The term "Partial Order" means that this set of quadruples satisfies the properties of a partial order: transitivity and antisymmetry.

Following the quadruple construction, soft ordinal embedding (SOE) [116] is used to preserve the global structure of the data. The technical detail of the SOE algorithm will be discussed in Section 3.2.

Global ordinal approaches suffer from vast time consumption when applied to large real-world datasets. Applying them to a smaller number of data points is more appropriate and gives promising results. The SOE takes advantage of retaining the cohort relations in the target $2,3$ space, which is explained in detail in Section 3.2 and 3.3.

2.3 Multi-objective DR Methods

Instead of solely considering the dataset's local or global structure, several studies explore different methods for multi-objective purposes. For example, one direction of the multi-objective DR problem, which will be discussed in this section, aims at preserving the local neighbour structure and the cohort relations simultaneously during the embedding process. The expected embedding result should possess both properties. However, in most multi-objective cases, a trade-off is unavoidable. Therefore, specific parameters can be adjusted, then the final result will satisfy the expectation.

Several DR approaches adopt the multi-objective property to improve information extraction from the dataset. For example, Jiang et al. [63] combines PCA and LE to achieve robustness of correct manifold information. Abeo et al. [1] incorporates local graphs with PCA to preserve global and local structures. Another work proposed by Abeo et al. [2] employs multiple manifold embedding methods and a noise-free penalty weight PCA approach. However, they are not targeted for data visualisation purposes.

COVA [92] is among the leading approaches that combine local graph embedding with consideration of global cohorts. It uses prototypes to represent cohorts to control the cohort separability and arrangement in the low-dimensional space. First, the distance between data samples and the closest prototype can be calculated. Then, each data sample mapped in the target $2,3$-D space is distributed approximately around the corresponding prototype to preserve the data-prototype distance. This forms the first objective of the COVA algorithm, which refers to the global structure preservation of the cohort data. The other aim is to retain the local neighbourhood of each data sample. The control parameters serve to govern the trade-off between the two objectives.

Anchor-t-SNE (AtSNE) [39] is a recent approach that attempts to balance the preservation of the local and global data structures for visualisation. It utilises cluster

Category	Out-of-sample Mapping Methods	Progressively Incremental methods	Neural network methods
Description	Use Kernel functions to generalise the embedding of the new sample	Embed new points along with updating the known results	Apply the neural network to learn the mapping weights
Strength	Straightforward and easily for calculation	Past embedding results could be updated	Stronger learning capacity
Weakness	Not accurate embedding result	Time-consuming to process many new samples	Training time, design of the neural network
Methods	Out-of-sample extensions for LLE[13, 148], ISOMAP [13, 25], LE [13], Kernel t-SNE [48, 49], bi-Kernel t-SNE [141]	Incremental LE [62], Incremental LLE [73], Incremental Isomap [42, 76], Incremental LTSA [87], Progressive UMAP [70]	Parametric t-SNE [118], Parametric UMAP[103]

Table 2.3: Overview of Incremental DR methods

centres to pull the data points around them to have control of the global data structure. However, it does not emphasise cohort separation, and is also time-consuming that requires GPU support.

Pairwise controlled manifold approximation projection (PaCMAP) [131] optimises the embedding points using three kinds of pairs of data samples: neighbour pairs, mid-near pairs, and farther pairs. PaCMAP utilises the attractive force on mid-near points and repulsion force on farther pairs to preserve the global structure of the dataset, and it restores the local neighbourhood via neighbour pairs. Weight parameters were applied to balance three loss functions built on these three groups of pairs.

Inspired by these algorithms, the multi-objective model ANGEL is proposed in Section 4. It also applies the anchor points strategy to maintain the global and local structure simultaneously. Variations of the ANGEL model are shown in Section 6.

2.4 Incremental DR Methods

The DR methods discussed above share a common drawback: they are essentially only applicable in batch mode. This means that these methods are only feasible for a fixed set of data samples but do not support the gradual arrival of new data samples. For example, when a new data sample $\mathbf{x}_{n+1}$ arrives, the Isomap approach needs first to reconstruct the distance matrix based on the new dataset $\mathbf{X}_{n+1} \in \mathbb{R}^{(n+1)\times D}$, and then process the embedding points using the new distance matrix. Running the entire algorithm repeatedly for newly arrived data samples is both time-consuming and discards the previous results [42, 79].

The incremental DR methods are developed to handle the streaming data [76], with an additional ability to visualise the gradual change of the embedding result [79]. Table 2.3 presents summary of some recent popular incremental DR methods. Incremental algorithms can be roughly categorised into three groups [42, 79]: out-of-sample mapping methods, progressively incremental methods, and the neural network method. The out-of-sample mapping methods and neural network methods belong to the parametric embedding category, while the progressively incremental methods are non-parametric [49].

The parametric approach relies on an explicit mapping function to project the high-dimensional data to the target $2,3$-D space [46]. Classical parametric includes PCA [135], Kernel PCA [104], and Auto-encoder network [58]. Section 2.4.1 will review out-of-sample mapping methods with kernel tricks. Neural network methods, which are related to the auto-encoder application, will be reviewed in Section 2.4.3.

The non-parametric approach takes a cost function approach to optimise the low-dimensional data directly. Many popular algorithms belong to this category, such as LLE, t-SNE, and UMAP enjoying larger flexibility of embedding [46]. Section 2.4.2 will review the non-parametric progressively incremental methods.

2.4.1 Out-of-sample Mapping Methods

The main assumption of the out-of-sample mapping methods is that all the known results are correct [42]. Thus, a kernel trick can generate an approximate projection function based on the known results.

Bengio et al. [13] is an early piece of work discussing the out-of-sample extension for MDS, LE, Isomap, and LLE. These non-parametric approaches are redefined in the Kernel PCA framework [105] and the Nystrom approximation is employed [97, 122]. Similar extensions for Isomap are also proposed in [25].

Kernel t-SNE [48, 49] proposed that an explicit solution can be derived from a simple interpolation given a fixed sample set of projected points. The mapping function is solved by minimising the distance between the projection of the newly coming data sample and all existing embedded points. Furthermore, Fisher Kernel [47, 96] is also applied to improve the projection result while considering the cohort label. Bi-kernel t-SNE [141] is proposed based on kernel t-SNE and PCA to overcome the drawback of kernel t-SNE regarding outlier projection.

Although the out-of-sample mapping method can process the new data sample in a linear time [48, 49, 141], it fails to improve the existing results if the new sample could change the neighbourhood of some samples, or even the structure of the dataset.

2.4.2 Progressively Incremental methods

The main idea of progressively incremental methods is to embed the new points and update the existing result simultaneously. Similar to the batch approach, relations between the new sample $\mathbf{x}_{n+1}$ and the batch set $\mathbf{X}$ should be derived first, and then the overall embedding can be updated based on the new relations. However, progressively incremental methods do not optimise all the points, but update points that the new point has influenced, as well as the new point itself.

Incremental LE [62], Incremental LLE [73], Incremental Isomap [42, 76], Incremental LTSA [87] and Progressive UMAP [70] all begin with the update of the k-NN graph. Points whose neighbours have been changed due to the new sample can also be observed during this process. Then the optimisation function can be constructed based on different embedding strategies to optimise the position of the new point and the updated points. Compared with the out-of-sample mapping methods, the progressively incremental methods can give more credible results [42].

However, progressively incremental methods could only process one new point per running. If the number of new points increases, the total cost of the time complexity will be high as the process needs to be repeated many times. Moreover, noisy data may destroy the overall structure of the embedding result.

2.4.3 Neural Network Methods

It is a significant branch of DR methods that applies auto-encoder neural networks to obtain the low-dimensional embeddings. Neural network DR methods are widely used in feature extraction and classification fields [26, 69, 127, 130, 138].

Parametric t-SNE [118] and Parametric UMAP [103] utilise the auto-encoder neural network to process the newly coming data samples. They rely on the basic property of the neural network that the projection of the new data sample can be treated as a test result, putting the new data sample as the test sample of the network. The weight of the network is trained by the batch dataset and embedding result of the batch dataset.

Although neural networks are widely applied in real-world applications, they do not tend to produce accurate visualisations of the high-dimensional dataset [131]. Moreover, training the network requires amounts of time and many data samples, which may not suitable for simply visualising the data. This thesis will not involve neural network applications, but it is still an attractive future direction to investigate.

2.5 Chapter Summary

This chapter presents an overview of research topics related to DR approaches for data visualisation. Each section summarises a group of DR methods with a similar embedding strategy. Section 2.1 and Section 2.2 describes approaches targeting at preserving local and global structure, respectively. Section 2.3 introduces multi-objective approaches for satisfying several purposes simultaneously. Section 2.4 provides an outline of incremental DR methods. Brief analyses of the research gaps of the existing DR works for cohort data visualisation are also presented in this chapter.

Chapter 3

Technical Background

In this chapter we provide the technical background to the remainder of this dissertation to facilitate readers' understanding of chapters that follow. We provide information on several data visualisation approaches that form the basis of our proposed algorithms.

Following the previous definition of the problem discussed in Section 2, we use $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\} \in \mathbb{R}^{n \times D}$ to denote the original dataset, for which we aim to find an embedding matrix $\hat{\mathbf{X}} = \{\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, \ldots, \hat{\mathbf{x}}_n\} \subset \mathbb{R}^{n \times d}$, where $d = 2, 3$. Each $\mathbf{x}_i = [x_{i1}, \ldots, x_{iD}]^T$ corresponds to a data pattern (or called object, sample) characterised as a point in a high-dimensional space $\mathbb{R}^D$; each $\hat{\mathbf{x}}_i = [\hat{x}_{i1}, \hat{x}_{i2}, \ldots, \hat{x}_{id}]^T$ stores the coordinates of an embedded point; n denotes the number of points. We use $\mathbf{0}_n$ and $\mathbf{1}_n$ to denote the n-dimensional column vectors with all elements equal to 0 and 1, respectively, while $\mathbf{0}_{n \times m}$ and $\mathbf{1}_{n \times m}$ the zero and one matrices with n rows and m columns. $\mathbf{I}_n$ denotes an identity matrix of size n.

3.1 t-SNE Algorithm

The t-distributed stochastic neighbour embedding (t-SNE) algorithm [121] converts a high-dimensional dataset into $2, 3$-D data points, where nearby points refer to similar objects, and distant points represent dissimilar samples. The primary strategy is to define a probability distribution P to measure the similarity among samples in the high-dimensional space. Then, a similar probability distribution Q is constructed over data points in the $2, 3$-D space that the Kullback-Leibler (KL) divergences between q_{ij} and

p_{ij} over all data points are minimised. Each p_{ij} and q_{ij} refer to the pairwise similarity between data points in the high-dimensional space and $2,3$-D space, respectively.

Mathematically, the conditional probability $p_{j|i}$ is defined as

$$p_{j|i} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma_i^2)}{\sum_{k\neq i}\exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2/2\sigma_i^2)}, \tag{3.1}$$

which can be explained as a Gaussian distribution centred at sample $\mathbf{x}_i$. Each $p_{j|i}$ is the probability that sample $\mathbf{x}_i$ would pick sample $\mathbf{x}_j$ as a neighbour if neighbours are picked in proportion to the probability density. Note that $p_{i|i} = 0$ and $\sum_j p_{j|i} = 1$. Define the joint probability p_{ij} to be the symmetrized conditional probability as

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}, \tag{3.2}$$

where $p_{ii} = 0$, $p_{ij} = p_{ji}$, and $\sum_{i,j} p_{ij} = 1$. Thus, the probability distribution P can be seen as the joint probability distribution over pairs of data points in the high-dimensional space. The variance of the Gaussian distribution σ_i can computed by a binary search on a defined perplexity function

$$Perp(P_i) = 2^{-\sum_j p_{j|i}\log_2 p_{j|i}}. \tag{3.3}$$

Another choice of the σ_i value can be a user-given fixed number [57]. The user-defined perplexity $Perp(P_i)$ can be interpreted as a measure of neighbours of the sample $\mathbf{x}_i$. A smaller perplexity indicates the probability concentrated heavily on the nearest neighbours of samples [119].

In the $2,3$-D space, t-SNE employs the Student t-distribution with one degree of freedom as

$$q_{ij} = \frac{(1 + \|\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j\|^2)^{-1}}{\sum_{k\neq l}(1 + \|\hat{\mathbf{x}}_k - \hat{\mathbf{x}}_l\|)^{-1}} \tag{3.4}$$

and set $q_{ii} = 0$. The cost function that minimises the KL divergence between P and the student-t based joint probability distribution Q is given as

$$C = KL(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}, \tag{3.5}$$

where the gradient of the KL divergence is given by

$$\frac{\partial C}{\partial \hat{\mathbf{x}}_i} = 4\sum_j (p_{ij} - q_{ij})(\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j)(1 + \|\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j\|^2)^{-1}. \tag{3.6}$$

The embedded points $\{\hat{\mathbf{x}}_i\}_{i=1}^n$ can be solved by applying gradient descent method. Every iteration is

$$\begin{aligned} \Delta\hat{\mathbf{x}}_i^{(t)} &:= -\eta\frac{\partial C}{\partial \hat{\mathbf{x}}_i} + \alpha(t)\Delta\hat{\mathbf{x}}_i^{(t-1)}, \\ \hat{\mathbf{x}}_i^{(t)} &:= \hat{\mathbf{x}}_i^{(t-1)} + \Delta\hat{\mathbf{x}}_i^{(t)}, \end{aligned} \tag{3.7}$$

where the momentum $\alpha(t)$ is used for faster and better convergence [98]. η is the learning rate that is a positive small constant ($\eta = 0.1$) and it can be updated during the optimisation process [61].

3.2 Soft Ordinal Embedding Algorithm

The SOE algorithm [116] is a typical and efficient ordinal embedding method. It has a nice feature of preserving both the ordinal structure and the density structure of the data. Moreover, comparing with other ordinal embedding approaches, SOE is among the best performing methods for preserving the local neighbourhood of the data [124]. The strategy is to construct the embedded data by enforcing ordinal constraints derived from the original data samples.

Given a quadruple of data samples $(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k, \mathbf{x}_l)$, the ordinal constraint of the quadruple is settled as

$$\text{If } D(\mathbf{x}_i, \mathbf{x}_j) < D(\mathbf{x}_k, \mathbf{x}_l), \text{ then } \hat{D}(\hat{\mathbf{x}}_i, \hat{\mathbf{x}}_j) < \hat{D}(\hat{\mathbf{x}}_k, \hat{\mathbf{x}}_l). \tag{3.8}$$

Here $D(\cdot,\cdot)$ denotes the distance or similarity between two data samples in the original space, while $\hat{D}(\cdot,\cdot)$ denotes the distance in the embedded space. As for visualisation, we consider using Euclidean distance in the embedded $2,3$-D space. However, the distance/similarity calculated in the original space can be measured by different techniques. In the non-metric embedding setting, the value of $D(\mathbf{x}_i, \mathbf{x}_j)$ is unknown but the distance order reflected by the left inequality of Eq. 3.8 is known. The quadruple set, as

$$\Gamma^{\mathrm{SOE}}(\mathbf{X}) = \left\{(i,j,k,l) | D(\mathbf{x}_i,\mathbf{x}_j) < D(\mathbf{x}_k,\mathbf{x}_l),\ i<j,\ k<l\right\}, \tag{3.9}$$

stores the full distance order obtained in the original space. SOE finds the embedding coordinates by solving the following optimisation problem:

$$\min_{\{\hat{\mathbf{x}}_i\}_{i=1}^n} \sum_{(i,j,k,l)\in\Gamma^{\mathrm{SOE}}(\mathbf{X})} \max{}^2\left[0, \hat{D}(\hat{\mathbf{x}}_i,\hat{\mathbf{x}}_j) + \delta - \hat{D}(\hat{\mathbf{x}}_k,\hat{\mathbf{x}}_l)\right], \tag{3.10}$$

where $\max(\cdot,\cdot)$ returns the larger one between the two input numbers, and $\delta > 0$ controls the embedding scale.

A simple way to obtain the optimum $\{\hat{\mathbf{x}}_i\}_{i=1}^n$ is the line search gradient descent. Since the objective function is built on a set of triplet, the update of each $\hat{\mathbf{x}}_i$ can be treated as a sum of the gradient of the main function $\max^2\left[0, \hat{D}(\hat{\mathbf{x}}_i,\hat{\mathbf{x}}_j) + \delta - \hat{D}(\hat{\mathbf{x}}_k,\hat{\mathbf{x}}_l)\right]$ over a subset of quadruples which contains $\mathbf{x}_i$.

Alternatively, as proposed by Terada et al. [116], the problem can be solved by minimising a majorizing function derived from the objective function in Eq. 3.10 [50]. However, compared with line search gradient, the majorizing function based optimisation for SOE requires calculation on huge matrices, which may cause unexpected memory and time consumption.

The potential number of ordinal constraints employed by SOE is of the order $O(n^4)$, and it becomes computationally very costly when the data size n increases.

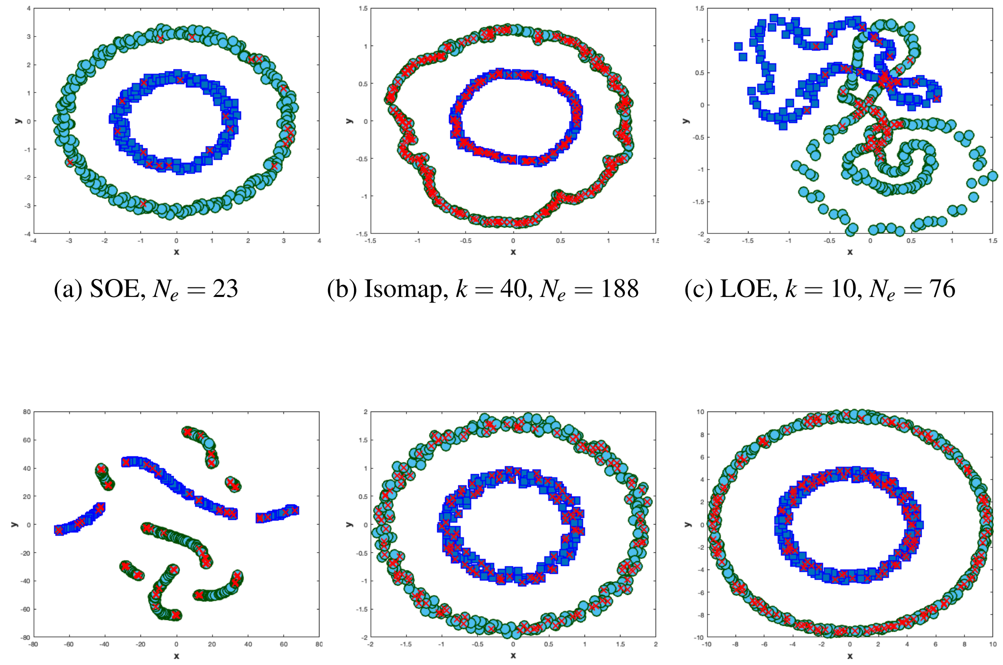
3.3 Local Ordinal Embedding Algorithm

Local ordinal embedding (LOE) [116] is a local variation of SOE, which considers only the k-NNs of each object to formulate the ordinal constraints:

$$\text{If } \mathbf{x}_j \in k\text{-NN}(\mathbf{x}_i) \text{ and } \mathbf{x}_l \notin k\text{-NN}(\mathbf{x}_i), \text{ then } \hat{D}(\hat{\mathbf{x}}_i,\hat{\mathbf{x}}_j) < \hat{D}(\hat{\mathbf{x}}_i,\hat{\mathbf{x}}_l). \tag{3.11}$$

The triplet set of ordinal constraints is settled as

$$\Gamma^{\mathrm{LOE}}(\mathbf{X}) = \left\{(i,j,l) | \mathbf{x}_j \in k\text{-NN}(\mathbf{x}_i), \mathbf{x}_l \notin k\text{-NN}(\mathbf{x}_i)\right\}. \tag{3.12}$$



(a) SOE, $N_e = 23$ (b) Isomap, $k = 40$, $N_e = 188$ (c) LOE, $k = 10$, $N_e = 76$

(d) t-SNE, $k = 10$, $N_e = 98$ (e) LOE, $k = 60$, $N_e = 188$ (f) t-SNE, $k = 60$, $N_e = 216$

Figure 3.1: Embed 2-D concentric circles in a 2-D space. The embedded points with at least one error neighbour in the local neighbourhood of size k are highlighted by red crosses, and the number of such points N_e is reported.

The similar optimisation problem as Eq. 3.10 is solved by LOE, but replacing the quadruple set Γ^{SOE} with the triplet set Γ^{LOE}

$$\min_{\{\hat{\mathbf{x}}_i\}_{i=1}^n} \sum_{(i,j,l)\in\Gamma^{\text{LOE}}(\mathbf{X})} \max{}^2\left[0, \hat{D}(\hat{\mathbf{x}}_i, \hat{\mathbf{x}}_j) + \delta - \hat{D}(\hat{\mathbf{x}}_i, \hat{\mathbf{x}}_l)\right]. \tag{3.13}$$

Terada et al. [116] also proposed the majorization algorithm to optimise the LOE update $\hat{\mathbf{x}}_i$. Similar line search gradient descent utilised in SOE can also be applied to LOE.

LOE significantly reduces the required number of ordinal constraints to $kn(n-k)$ and is computationally much cheaper.

In Figure 3.1, we compare LOE with SOE and Isomap both of which approximate isometric embeddings, and with t-SNE which is a local neighbourhood preservation algorithm like LOE. A set of $n = 500$ data points are sampled from 2-dimensional concentric circles (200 points from the inner circle and 300 from the outer one), and they are embedded into a 2-D space. SOE generated the best embedding but took over 2 hours to produce the result, while the other methods took seconds or minutes. Isomap requires $k \geq 35$ to construct a fully connected neighbour graph for geodesic distance estimation. LOE requires $k \geq 55$ and t-SNE $k \geq 60$ to recover the concentric circle shape. Regardless of whether LOE is able to recover the global structure of concentric circle, it is fairly robust in preserving a local neighbourhood structure, offering the lowest neighbour preservation error N_e (see figure caption for error definition). This encourages us to use LOE as a local component in ANGEL.

3.4 Local Anchor Embedding Algorithm

The usage of anchor points has been widespread in graph-based semi-supervise learning [86, 129, 142]. In data visualisation, Landmark MDS [29] is the early work using landmark points (anchor points) on embedding problems. This two-phase approach utilise landmarks to reduce the computation of the large-scale dataset. COVA [92], and AtSNE [39] are recent research focusing on applying anchor points as the global structure representation of the dataset.

The idea is to generate a small set of data points to roughly represent a data distribution or a graph structure, e.g., by using a clustering algorithm and computing the cluster centres. These points are referred to as the anchor points, each denoted by $\mathbf{u}_i \in \mathbb{R}^D$, and they are stored in anchor point matrix $\mathbf{U} = \{\mathbf{u}_1, \ldots, \mathbf{u}_m\} \subset \mathbb{R}^{m \times D}$.

The connection between anchor points and data points is the key point to be constructed. Inspired by LLE [102], local anchor embedding (LAE) [86] attempts to approximate each observed data point $\mathbf{x}_i$ by a convex combination of its t-nearest anchor points (t-NAP). Let $\mathbf{w}_i = [w_{i1}, w_{i2}, \ldots, w_{im}]^T$ denote the combination weights for reconstructing the i-th data point, and $\mathbf{W} = \{\mathbf{w}_i\}$ is the reconstruction matrix. It is optimised by solving

$$\begin{gathered}\mathbf{w}_i^* = \arg \min_{\mathbf{w}_i \in \mathbb{R}^m} \frac{1}{2} \left\| \mathbf{x}_i - \mathbf{U}^T \mathbf{w}_i \right\|_2^2, \qquad (3.14)\\ \text{subject to } \mathbf{1}^T \mathbf{w}_i = 1,\ w_{ij} \geq 0, w_{ij} = 0 \text{ if } \mathbf{u}_j \notin t\text{-NAP}(\mathbf{x}_i),\end{gathered}$$

where $\|\cdot\|_2$ denotes the l_2 norm. The projected gradient [34] is applied to solve

Eq. 3.14. It projects the updated point to a multinomial simplex expressed by constraints in Eq. 3.14 [86]. The Nesterov's method is used to accelerate the optimisation process [86, 93]. It alternates between gradient update and proper extrapolation for acceleration purpose. Ultimately, LAE is optimised with a total time complexity $O(tmn + t^2Tn)$, where T is the iteration of optimisation process.

Another related concept is about dictionary learning. Treating each original data sample $\mathbf{x}_i$ as a signal in $\mathbb{R}^D$, the dictionary $\mathcal{Dic} = \{\phi_1, \ldots, \phi_m\} \subset \mathbb{R}^{D\times m}$ refers to a small set of basic independent signals sampled from the high-dimensional space of the original dataset $\mathbf{X}$ where each $\mathbf{x}_i$ can be seen as a linear combination of $\mathcal{Dic}$. The dictionary is over-complete if $m > D$, making the set span the high-dimensional space [74, 117]. Setting $\mathbf{a}_i \in \mathbb{R}^m$ to be the coefficient vector containing the reconstruction weight, the goal is to find $\mathcal{Dic}$ and $\mathbf{a}_i$ that minimise the following equation

$$\arg\min_{\mathcal{Dic},\mathbf{a}_i} \frac{1}{2}\|\mathbf{x}_i - \mathcal{Dic}\cdot\mathbf{a}_i\|_2^2 + \lambda\|\mathbf{a}_i\|_0,$$

where λ here is the regularisation parameter. However, since the dictionary learning requires to solve both the $\mathcal{Dic}$ and $\mathbf{a}_i$, the learning representation computation is computationally expensive [99].

3.5 Progressive UMAP

UMAP [90] is one of the most advanced methods for DR and data visualisation. The main idea of UMAP is to generate fuzzy topological representations for both high and low dimensional data points and to change the low dimensional embedding so that the fuzzy topological representation is similar to that of the high dimensional dataset [45]. In this thesis, however, the theory of UMAP is not the focus of our work. The method mentioned in Progressive UMAP [70] on how to make UMAP, which cannot handle newly arrived data points, continuously process newly coming data points is the theoretical support for the incremental methods that will be proposed in the following chapters.

Briefly, the UMAP algorithm consists of two primary phases: graph construction and layout optimisation. The graph construction starts by generating k-NN graph that describes the distance between data samples in the high-dimensional space. Then for each data sample, a non-zeros distance from $\mathbf{x}_i$ and its nearest neighbour can be observed as ρ_i. Thus, the scaling parameter of σ_i for each $\mathbf{x}_i$ can be obtained by solving the equation

$$\log_2(k) = \sum_{\mathbf{x}_j \in k\text{-NN}(\mathbf{x}_i)} \exp\left(\frac{-\max\left\{0, D(\mathbf{x}_i, \mathbf{x}_j) - \rho_i\right\}}{\sigma_i}\right). \tag{3.15}$$

Using ρ_i and σ_i, the graph edge between two data samples $\mathbf{x}_i$ and $\mathbf{x}_j$ can be computed as

$$v_{j|i} = \exp\left(\frac{-\max\left(0, D\left(x_i, x_j\right) - \rho_i\right)}{\sigma_i}\right). \tag{3.16}$$

Then, the symmetric weight can be computed as

$$v_{ij} = v_{j|i} + v_{i|j} - v_{j|i} \cdot v_{i|j}, \tag{3.17}$$

which also indicates the probability distribution of data samples in the high-dimensional space. As the low-dimensional points are expected to has the similar probability distribution, the cost function of UMAP is designed as

$$C_{UMAP} = \sum_{i \neq j}\left[v_{ij} \cdot \log\left(\frac{v_{ij}}{w_{ij}}\right) - \left(1 - v_{ij}\right) \cdot \log\left(\frac{1 - v_{ij}}{1 - w_{ij}}\right)\right], \tag{3.18}$$

where the $w_{ij} = \left(1 + a\left\|\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j\right\|_2^{2b}\right)^{-1}$ represents the similarity between embedding points $\hat{\mathbf{x}}_i$ and $\hat{\mathbf{x}}_j$.

The layout optimisation phase can be viewed as a stochastic gradient descent on individual observation, however, no further detailed descriptions will be discussed with this thesis. The optimisation details can be found in the original paper of UMAP [45].

The algorithm of progressive UMAP is shown is Algorithm 1. The most important part is to update the k-NN graph each time when new data point arrives. Denote the new data point as $\mathbf{x}_{new}$, the new k-NN graph can be updated applying Progressive Approximate k-NEarest NEighbors (PANENE) [64]. Thus, data points whose neighbouring points are changed can also be observed from this new k-NN graph, denoting as $\mathbf{X}_{update}$. Then, both $\mathbf{x}_{new}$ and the subset of the dataset $\mathbf{X}_{update}$ will be optimised by

Algorithm 1 Progressive UMAP algorithm

```
Input: $D$-dimensional batch data points $\{\mathbf{x}_i\}_{i=1}^n$.
User-adjust hyperparameters: Reduced dimension $d$ (2 or 3), local neighbourhood size $k$;
Construct $k$-NN graph of the batch dataset.
if $\mathbf{x}_{new}$ arrives then
    Update the new $k$-NN graph using PANENE.
    Obtain $\mathbf{X}_{update}$.
    Embedding $\mathbf{x}_{new}$ and $\mathbf{X}_{update}$ using UMAP algorithm.
end if
Output: Embedded data points $\hat{\mathbf{x}}_{new}$ and updated subset of points $\hat{\mathbf{X}}_{update}$ in 2 (or 3)-D space.
```

UMAP algorithm.

This progressive UMAP algorithm is not a particularly complicated improvement on UMAP, but it is very effective in embedding the newly arrived data points into the $2,3$-D space. This has been a great inspiration to our incremental algorithm which will be discussed in Chapter 7.

3.6 Chapter Summary

This chapter provides a more detailed technical background of t-SNE, SOE, LOE, LAE, and progressive UMAP, which help readers in understanding the models presented in this thesis. Furthermore, it is explained here why these methods were used in the proposed model presented in Chapter 4 - Chapter 7.

Chapter 4

ANchor GuidEd Local (ANGEL) Model

4.1 Motivation

As stated in the previous chapters, generating a meaningful visualisation of the cohort dataset is not a trivial task. A *good* visualisation approach should demonstrate separation between data cohorts, preserve distances between individual data points, and retain cohort positioning and the cohort's internal structure.

Most popular embedding methods (Section 2.1) focus on local neighbourhood preservation while constructing low-dimensional embeddings. As we focus on cohort data visualisation, some methods can keep the cohort neighbour relationships if a large neighbourhood is carefully selected. However, in high-dimensional data visualisation, it is difficult to know what the neighbourhood size should be to generate a reliable visualisation.

Some global embedding methods (Section 2.2) keep all information from the dataset in the low-dimensional space. However, they have a significant loss in terms of neighbourhood preservation. Moreover, they do not handle high data dimensions and a high number of data points well when drastic dimension reduction (to 2,3 D) is required.

As shown in Section 2.3, only a limited number of current data visualisation research studies discuss an effective way to demonstrate distinct cohorts by preserving cohort relationships and keeping the distance between data points. They emphasise

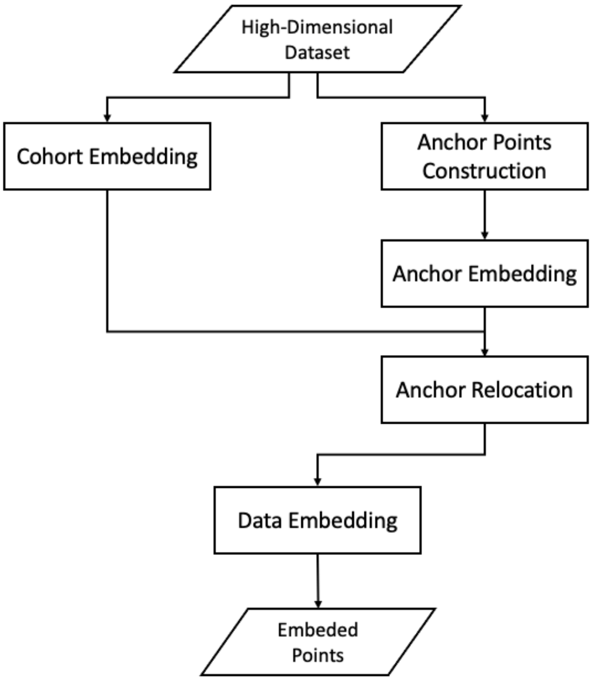


Figure 4.1: The general framework of the ANGEL algorithm.

the utilisation of prototype or anchor points to represent each cohort and sketch the cohort relationships. However, the limitation of representing the complex internal cohort structure and weak cohort separation disturbs analysts from visualising the high-dimensional dataset.

This chapter introduces a novel data visualisation model called ANchor GuidEd Local (ANGEL), which provides an improved balance on local neighbourhood preservation, cohort positioning, internal structure preservation, and adjustable cohort separation enhancement in the reduced space. The proposed method contains five processing stages, as shown in Figure 4.1:

1. Generate cohort positions in a $2,3$-D using the global ordinal embedding method SOE [116] to keep cohort positions and relations in Section 4.2
2. Generate anchor points to summarise the overall data distribution or structure, and capture the relationships between the anchor points and the data points in the original space by LAE [86] in Section 4.3.
3. Embed the anchor points in a $2,3$-D space using a triplet ordinal embedding

method derived based on the SOE strategy, then adjust the internal cohort structure and the cohort separation using a proposed transformation algorithm in Section 4.4.1.

4. Relocate embedded anchor points according to the cohort positions in Section 4.4.2.

5. Embed the data points around the anchor points to maintain the same global structure as modelled in the first stage while enforcing an additional local neighbouring structure. As for this last step, we introduce several different approaches to optimise the proposed cost function in Section 4.5, and each of them has different advantages for data visualisation purposes.

4.2 Cohort Position Embedding

Facing a given cohort dataset, where each data point is associated with a cohort label $y_i \in \{1, \dots, C\}$, the first step is to identify different cohorts. The relative positions of each cohort belonging to the cohort dataset are expected to convey important information to the viewers [92], and it can be referred to as the global structure of the cohort dataset. For example, as shown in Figure 1.1, the FMNIST image dataset contains 10 different cohorts. To visualise this dataset in $2,3$-D space, embedded points relating to the cohort “Sneaker” are expected to be placed close to points belonging to the cohort “Ankle boot”, rather than the points of cohort “Top”. When applying the DR method to a cohort dataset for visualisation, it is desirable to preserve as much as possible the relationships between all cohorts. Since the number of cohorts is relatively small compared with the number of data points, it is suitable to utilise global DR methods in Section 2.2 to obtain the cohort position embedding.

We first compute the distance between two data cohorts. Measures like single-linkage distance Eq. 4.1 [110], complete-linkage distance Eq. 4.2 [110, 111], average-linkage distance Eq. 4.3 [110], and Hausdorff distance Eq. 4.4 [101] can be used.

$$\text{Single-Linkage: } D_C(i,j) = \min_{\mathbf{x}_p \in \mathbf{C}_i, \mathbf{x}_q \in \mathbf{C}_j} D(\mathbf{x}_p, \mathbf{x}_q), \tag{4.1}$$

$$\text{Complete-Linkage: } D_C(i,j) = \max_{\mathbf{x}_p \in \mathbf{C}_i, \mathbf{x}_q \in \mathbf{C}_j} D(\mathbf{x}_p, \mathbf{x}_q), \tag{4.2}$$

$$\text{Average-Linkage: } D_C(i,j) = \frac{1}{n_i n_j} \sum_{y_p=i} \sum_{y_q=j} D(\mathbf{x}_p, \mathbf{x}_q), \tag{4.3}$$

$$\text{Hausdorff distance: } D_C(i,j) = \min_{\mathbf{x}_p \in \mathbf{C}_i} \left\{ \max_{\mathbf{x}_q \in \mathbf{C}_j} D(\mathbf{x}_p, \mathbf{x}_q) \right\}, \tag{4.4}$$

where $i, j \in \{1, 2, ...C\}$ are cohort indices, n_i denotes the number of data points in the i-th cohort, $\mathbf{C}_i$ denotes the i-th cohort containing all data points whose label $y = i$. We use Euclidean distance-based average-linkage distance according to practical experience.

In most cases, real-world data sets often contain noisy and outlier data points. In order to obtain more accurate distances between cohorts, these noisy and outlier data points are preferably excluded from the cohort distance computation. We assume that each data cohort can be considered as a Gaussian distribution and that the mean and the standard deviation of the Gaussian distribution can be obtained statistically. Thus, we can calculate cohort distance only using data samples within *thred* standard deviations from the mean. A larger *thred* means more relaxation on data selection; however, $thred = 0$ refers to no data selection process. The recommend setting is $thred = 2$ or 3.

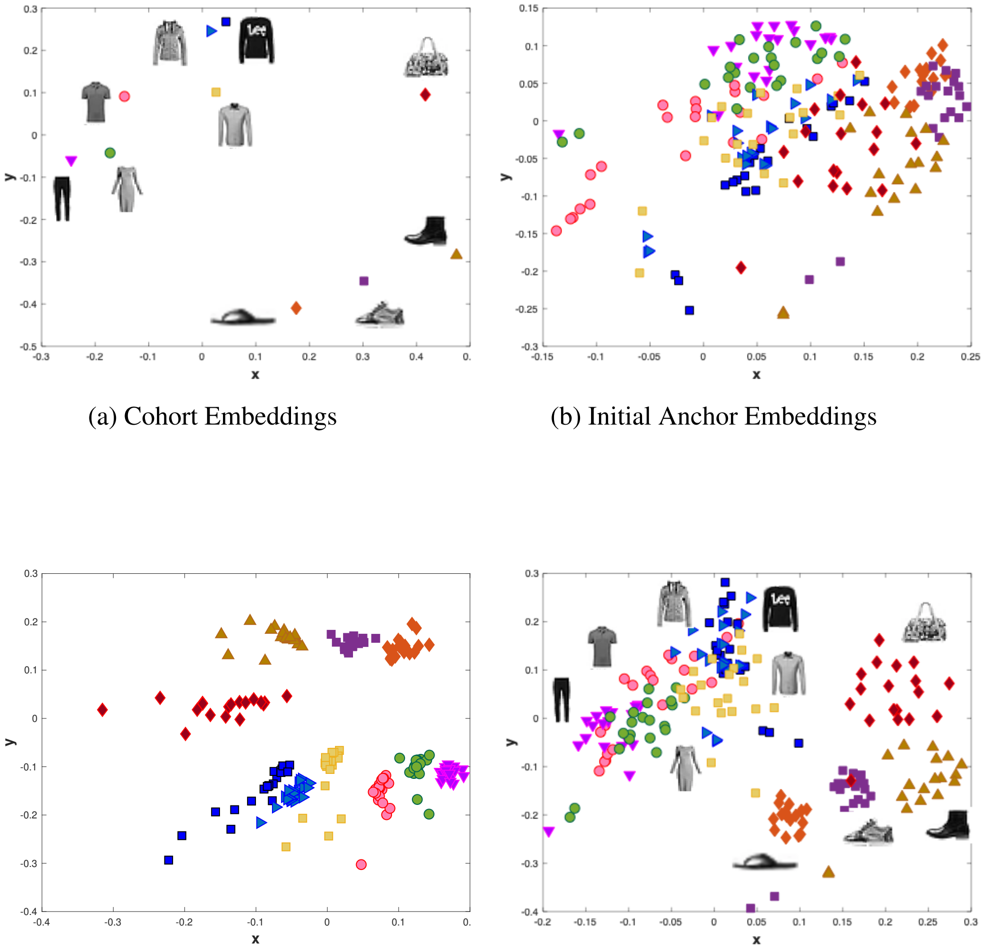
After obtaining the distances between C cohorts, we generate a complete set of ordinal triplets

$$\Gamma_C^{\text{SOE}} = \{(i,j,l) | D_C(i,j) < D_C(i,l)\}, \tag{4.5}$$

which is used to compute the C cohort embeddings $\left\{\hat{\mathbf{\upsilon}}_i \in \mathbb{R}^d\right\}_{i=1}^{C}$ by SOE following Eq. 3.10. We use these cohort embeddings to decide where the cohorts should be in the reduced space (see Figure 4.2a as an example).

4.3 Anchor Points Construction

The next step is to construct anchor points, also known as prototypes, in the original space $\mathbb{R}^D$ to outline the structure of each cohort. In order to represent the intrinsic characteristics of each cohort, anchor points are generated by considering the density of data points in each cohort. For cohorts that have more data points, more anchor points are generated, and vice versa. To implement this, the K-means algorithm [80]



(a) Cohort Embeddings

(b) Initial Anchor Embeddings

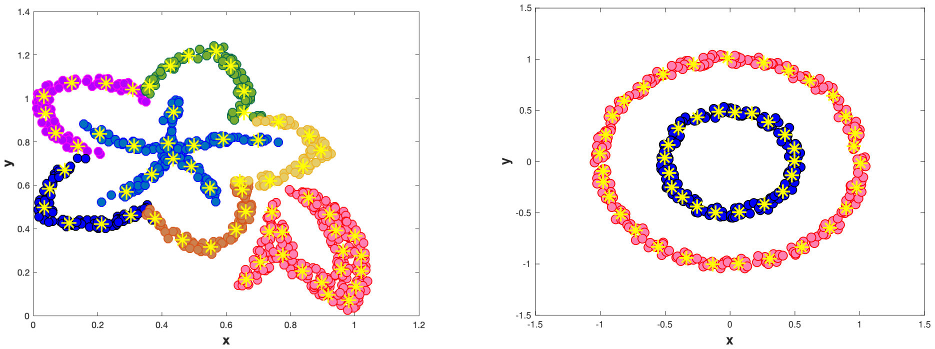
(c) Enhanced Anchor Embeddings

(d) Relocated Anchor Embeddings

Figure 4.2: Illustration of anchor embedding using the FMNIST dataset.

is used to partition each data cohort into several smaller clusters. Each anchor point is taken as the centroid point of each small cluster.

We use an anchor sparsity parameter $0 < p \ll 1$ to control the anchor number, and normally, set $p = 0.05$. These $m = \lfloor pn \rfloor \ll n$ anchor points serve as an informative summary of the internal cohort structure. Figure 4.3 shows two examples of the generated anchor points, and, as seen, they sketch the cohort positioning and shape. We denote these anchor points as $\mathbf{U} = \{\mathbf{u}_i\}_{i=1}^{m}$, with $c(\mathbf{u}_i) \in \{1, 2, \ldots, C\}$ denotes the cohort membership of each anchor point. As for some extremely imbalanced datasets with some large cohorts with tiny ones, a very small p will be selected, but we will



(a) Flower

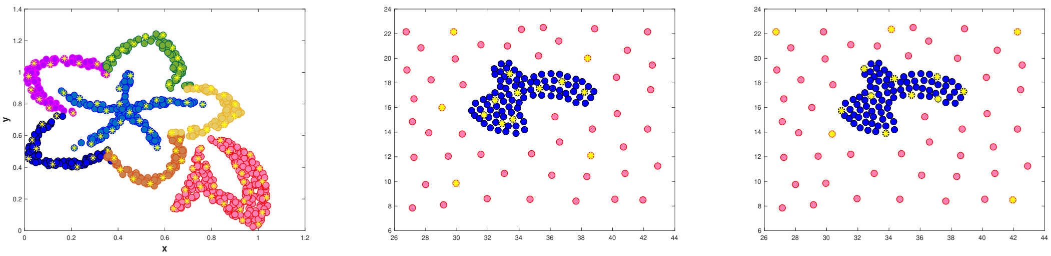
(b) Circle

Figure 4.3: The original flower and circle datasets, along with their generated anchor points. Circles in different colours correspond to the original data points from different cohorts, while stars in yellow to generated anchor points.

force those tiny cohorts to have at least 3 anchor points to support the cohort structure. Meanwhile, we still support selecting a given number of anchor points in each cohort, which leads to more freedom to construct appropriate anchor points.

Another way to roughly sketch the dataset is trying to find the vertices of each cohort simplex. Convex-hull method finds the convex-hull of each cohort and treat vertices of the convex-hull as anchor points [14]. This method focuses on finding the boundary of each cohort. With all points of the cohort in the convex-hull, each data point can be easily represented as the convex combination of vertices of the convex-hull. The problem is that if the number of the dimension of the dataset is bigger than the number of the data points in the dataset, the convex-hull cannot be constructed. Thus, the convex-hull method is not applied in our algorithm.

Other than finding the boundary, the Landmark method uses the sequential maxmin method [28, 31, 132] to find not only the vertices but also the supporting data points inner each cohorts. The main idea of this landmark selecting method is to choose points that are as far apart as possible. To be more specific, the algorithm aims to find the $\mathbf{u}_{i+1}$ from the data set $\mathbf{X}$ which maximise the $D(\mathbf{x}, \mathbf{u}_i)$. That is, suppose $\{\mathbf{u}_1, \dots, \mathbf{u}_i\}$ have been chosen as landmarks, the next landmark $\mathbf{u}_{i+1})$ is selected based on



(a) Flower, Landmark (b) Cloud points, K-means (c) Cloud points, Landmark

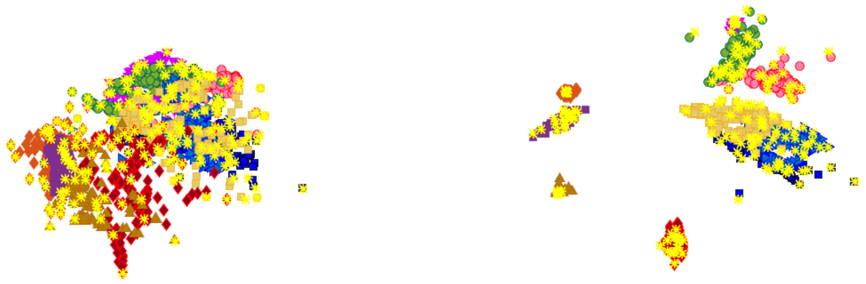
Figure 4.4: Anchor points generated by applying different approaches.

$$\mathbf{u}_{i+1} = \arg\max_{\mathbf{x}} \{F(\mathbf{x})\},$$

where $F(\mathbf{x}) = \min D(\mathbf{x}, \mathbf{u}_1), ..., d(\mathbf{x}, \mathbf{u}_i)$.

Anchor points generated by the Landmark method can cover the origin dataset. However, a disadvantage is that the landmark method still prefers to initially select boundary points as anchor points. Figure 4.4 illustrates anchor points generated by using different approaches. As the flower dataset sketches the shape of the flower, anchor points generated by applying the Landmark algorithm (Figure 4.4a) are similar to the anchor points generated by using K-means algorithm (Figure 4.3a). However, it can be observed from anchor points generated for the “leaf” part of the flower shape and anchor points generated for the Cloud points dataset (Figures 4.4b and 4.4c) that the Landmark approach tends to choose the boundary points as anchor points. As for the real dataset, take the FMNIST dataset as an example (See Section 5.2.1 for more detail about the dataset), Figure 4.5 shows embedded anchor points of FMNIST, where the high-dimensional anchor points are generated by different approaches (details of the anchor embedding algorithms can be found in Section 4.4). Although the dimension reduction process loses information of original data samples and anchor points, it can be found in the images that the landmark method outlines the boundary of cohorts while the K-means method covers more inner structure of cohorts. Thus, the K-means method is chosen for the ANGEL algorithm. In addition, however, we provide landmark choice for specific purposes.

The reconstruction operation of LAE is used to link these anchor points with the original data points. Since we aim at visualising a cohort dataset and anchor points are generated based on each cohort, we reconstruct each data point $\mathbf{x}_i$ with label $y_i = c$



(a) FMNIST, Landmark (b) FMNIST, K-means

Figure 4.5: Embedded anchor points of FMNIST dataset using anchor embedding and algorithms introduced in Section 4.4. The high-dimensional anchor points of FMNIST dataset are generated by applying different approach.

based on anchor points whose $c(\mathbf{u}_i) = y_i$. That is, we modified the LAE algorithm as

$$\begin{aligned} \mathbf{w}_i^* = \arg \min_{\mathbf{w}_i \in \mathbb{R}^m} \ & \frac{1}{2} \left\| \mathbf{x}_i - \mathbf{U}^T \mathbf{w}_i \right\|_2^2 . \\ \text{subject to } \ & \mathbf{1}^T \mathbf{w}_i = 1, \ w_{ij} \geq 0, \\ & w_{ij} = 0 \text{ if } \mathbf{u}_j \notin t\text{-NAP}(\mathbf{x}_i) \text{ and } c(\mathbf{u}) \neq y_i, \end{aligned} \tag{4.6}$$

The optimised reconstruction weights $\mathbf{w}_i^*$ will later help to bridge embedded anchor points with data embeddings to keep the cohort structure of the dataset. However, since we only select 3-nearest anchor points for each data point, the weight matrix $\mathbf{W}$ containing all reconstruction weights is very sparse. That means we do not need to acquire large memory to store the global information of the dataset. The pseudo-code of the anchor generation algorithm is shown as Algorithm 2.

4.4 Anchor Points Embedding

The main driving force of ANGEL is to preserve distance orders, that is, to preserve the ordinal information of the dataset. When the data has a high intrinsic dimension in

Algorithm 2 Anchor Generation Algorithm

```
Input: $D$-dimensional data points $\{\mathbf{x}_i\}_{i=1}^n$ and their cohort labels $\{y_i\}_{i=1}^n$ .
Fixed hyperparameters: anchor point density $p = 0.1$; reconstruction anchor number $t = 3$;
Set initial anchor number: $m = 0$
for Each cohort $\mathbf{C}_i$, $i \in \{1, 2, \dots C\}$ do
    Calculate the cluster number $K_i = \lfloor pn_i \rfloor$, apply K-means clustering to points from cohort $i$, obtain anchor points by taking the centroid point of each cluster.
    Compute anchor reconstruction weight $\{\mathbf{w}_i^*\}_{i=1}^n$ by Eq. 4.6 for each data point.
    Update anchor number by $m \leftarrow m + K_i$.
end for
Output: $D$-dimensional anchor point $\{\mathbf{u}_i\}_{i=1}^m$, anchor labels $\{c(\mathbf{u}_i)\}_{i=1}^m$, and data reconstruction weights $\{\mathbf{w}_i^*\}_{i=1}^n$,
```

$\mathbb{R}^D$, it is easier to manipulate distance orders for a small set of anchor points instead of a complete set of data points. This is also computationally much cheaper than embedding the whole set of data samples. In our algorithm design, we take into account the following concerns to embed anchor points:

1. Distance order between data cohorts should be kept by embedded anchor points.
2. Distance order between anchor points should be preserved.
3. Control of enhance the separation between data cohorts.

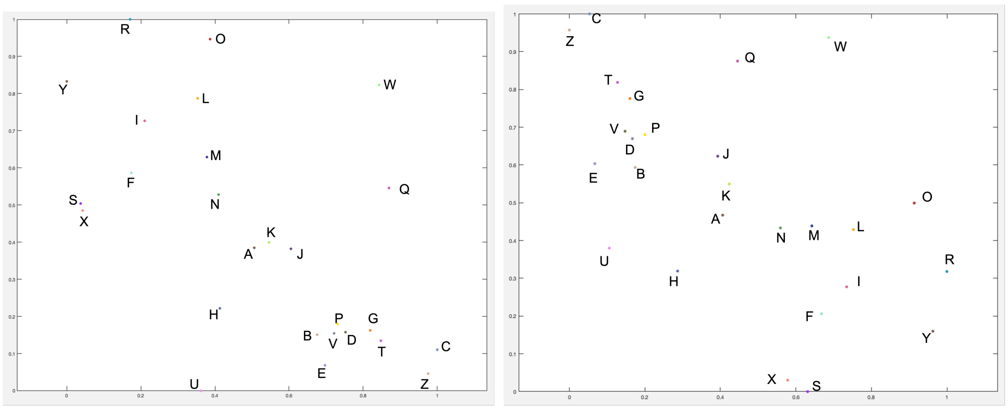
4.4.1 Anchor Embedding Process

We have discussed the cohort position embedding process in Section 4.2. The next step is to embed anchor points. The starting point is to simply preserve the complete distance orders between all the anchor points by the SOE algorithm [116] as we did in the cohort position embedding process. However, since the number of anchor points is much larger than the number of cohorts, applying the SOE algorithm is time-consuming.

We propose a triplet ordinal embedding algorithm as an approximation of the SOE algorithm, which leading to the following set of ordinal triplets:

$$\Gamma^{\text{triOE}}(\mathbf{U}) = \left\{(i,j,l) | \|\mathbf{u}_i - \mathbf{u}_j\|_2 < \|\mathbf{u}_i - \mathbf{u}_l\|_2 \right\}, \tag{4.7}$$

and optimise the equation:



(a) Isolet, SOE

(b) Isolet tri-OE

Figure 4.6: Embedding results of a subset of Isolet dataset (introduced in Section 5.2.1). It can be observed from the image that both embedding results shares similar relative positions of points.

$$\min_{\{\hat{\mathbf{u}}_i\}_{i=1}^{m}} \sum_{(i,j,l)\in\Gamma^{\text{triOE}}(\mathbf{U})} \max^2\left(0, \hat{D}(\hat{\mathbf{u}}_i,\hat{\mathbf{u}}_j)+\delta-\hat{D}(\hat{\mathbf{u}}_i,\hat{\mathbf{u}}_l)\right). \tag{4.8}$$

Compared with SOE (Section 3.2 Eq. 3.10 and 3.9), this modified triplet ordinal embedding reduces the number of constraints from $m(m-1)(m-2)(m-3)/4$ to $m(m-1)(m-2)/6$, which accelerate the embedding process. Figure 4.6a and 4.6b shows that there is little difference between results obtained by SOE and tri-OE respectively, and Figure 4.7 explains that SOE costs much more time than the proposed tri-OE does as the number of data sample n increases.

However, the direct embedding process of high-dimensional anchor points will always lead to a mingled $2,3$-D result, which is hard to identify cohort membership of each embedded anchor points. To fix the problem, we define a cohort-based distance measure between two anchor points:

$$D^c(\mathbf{u}_i,\mathbf{u}_j) = D\left(c(\mathbf{u}_i), c(\mathbf{u}_j)\right). \tag{4.9}$$

When there exists overlapping between data cohorts, there will be a certain degree of mismatch between $D^c(\mathbf{u}_i,\mathbf{u}_j)$ and $D(\mathbf{u}_i,\mathbf{u}_j) = \|\mathbf{u}_i - \mathbf{u}_j\|_2$. Our algorithm provides the option to enhance the separation between cohorts by selectively letting $D^c(\mathbf{u}_i,\mathbf{u}_j)$

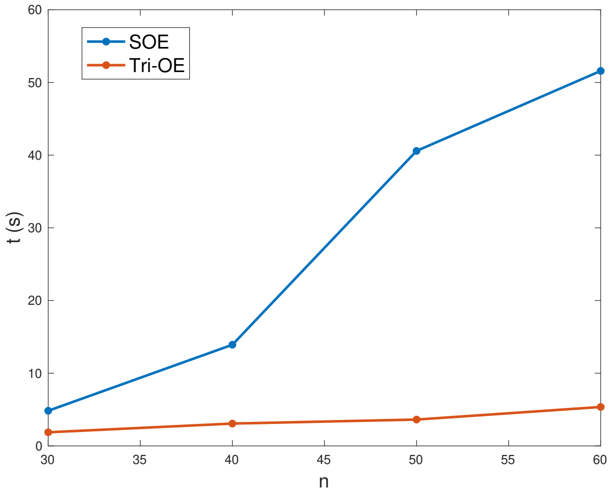


Figure 4.7: Time recording of SOE and tri-OE as the size of dataset n increases.

override $D(\mathbf{u}_i, \mathbf{u}_j)$ for anchor points that are neighbours according to $D(\mathbf{u}_i, \mathbf{u}_j)$ but not to $D^c(\mathbf{u}_i, \mathbf{u}_j)$. This is implemented by modifying the original set of ordinal triplets $\Gamma^{\text{triOE}}(\mathbf{U})$ to

$$\Gamma^{\text{s-OE}}(\mathbf{U}) = \left(\Gamma^{\text{triOE}}(\mathbf{U}) \setminus \Gamma^{\text{R}}(\mathbf{U})\right) \cup \Gamma^{\text{A}}(\mathbf{U}), \tag{4.10}$$

where

$$\Gamma^{\text{R}}(\mathbf{U}) = \left\{(i,j,l) | \mathbf{u}_j, \mathbf{u}_l \in \lfloor \lambda m \rfloor\text{-NN}(\mathbf{u}_i), \boldsymbol{\zeta}_{\mathbf{u}_i}(\mathbf{u}_j) < \boldsymbol{\zeta}_{\mathbf{u}_i}(\mathbf{u}_l), \boldsymbol{\zeta}^c_{\mathbf{u}_i}(\mathbf{u}_j) > \boldsymbol{\zeta}^c_{\mathbf{u}_i}(\mathbf{u}_l)\right\},$$
$$\Gamma^{\text{A}}(\mathbf{U}) = \left\{(i,l,j) | \mathbf{u}_j, \mathbf{u}_l \in \lfloor \lambda m \rfloor\text{-NN}(\mathbf{u}_i), \boldsymbol{\zeta}_{\mathbf{u}_i}(\mathbf{u}_j) < \boldsymbol{\zeta}_{\mathbf{u}_i}(\mathbf{u}_l), \boldsymbol{\zeta}^c_{\mathbf{u}_i}(\mathbf{u}_j) > \boldsymbol{\zeta}^c_{\mathbf{u}_i}(\mathbf{u}_l)\right\},$$

with $\lfloor \cdot \rfloor$ denoting the floor operation. We use the parameter $0 \leq \lambda \leq 1$ to control how many ordinal triplets to modify. The vector $\boldsymbol{\zeta}_{\mathbf{u}_i}$ stores the rankings of the other anchor points in terms of their closeness to $\mathbf{u}_i$ based on the distance measure D, and $\boldsymbol{\zeta}_{\mathbf{u}_i}(\mathbf{u}_j)$ denotes the particular ranking of $\mathbf{u}_j$. The smaller $\boldsymbol{\zeta}_{\mathbf{u}_i}(\mathbf{u}_j)$ is, the closer $\mathbf{u}_j$ is to $\mathbf{u}_i$. The same definition applies to $\boldsymbol{\zeta}^c_{\mathbf{u}_i}$ but based on D^c. This new set $\Gamma^{\text{s-OE}}(\mathbf{U})$ removes the incompatible distance orders between D^c and D for each anchor and its $\lfloor \lambda m \rfloor$-NNs, and adds the amended ones following D^c. When $\lambda = 1$, all the incompatible triplets are modified. When $\lambda = 0$, there is no modification and $\Gamma^{\text{R}}(\mathbf{U}) = \Gamma^{\text{A}}(\mathbf{U}) = \emptyset$.

Anchor embeddings computed by solving the Eq. 4.8 using $\Gamma^{\text{s-OE}}(\mathbf{U})$ are denoted as $\left\{\hat{\mathbf{u}}_i^{(0)} \in \mathbb{R}^d\right\}_{i=1}^m$. One should bear in mind that, for data with comparatively weak cohort separation in the high-dimensional space, an overly strong enforcement of cohort separation will change the true data structure, particularly resulting in less reliable preservation of local distance structure. Figures 4.2b and 4.2c illustrate the computed anchor embeddings at this stage.

After the cohort separation adjustment, the embedded anchor points $\left\{\hat{\mathbf{u}}_i^{(0)}\right\}_{i=1}^{m}$ may result in a cohort positioning arrangement incompatible to what is directly gathered by $\{\hat{\mathbf{v}}_i\}_{i=1}^{C}$. For example, Figure 4.2a shows a cohort embedding result of the FMNIST dataset, while Figure 4.2b presents the anchor embedding result with the parameter $\lambda = 0.1$, which is also referred as the initial anchor embeddings. It can be seen very explicitly that the distribution of initial anchor embedding points does not coincide with the positions of the data cohorts. Furthermore, initial anchor embedding points are mixed up and hard to distinguish from each other. Figure 4.2c shows the separability enhanced anchor embeddings with the parameter $\lambda = 0.9$. Although the anchor embedding algorithm's large λ value is capable of putting anchor points into accurate locations according to the cohort embedding result, it lost inherent cohort structure since anchor points are embedded to very close points. Thus, it is preferred to relocate initial anchor points to the cohort positions in order to keep the cohort relations of the dataset.

4.4.2 Anchor Relocation Process

The anchor embedding process could not demonstrate the cohort structure of the data successfully. Figure 4.2a shows the cohort embedding result of the FMNIST dataset. Figures 4.2b and 4.2c demonstrate two anchor embedding results of the same FMNIST dataset using different hyper-parameter λ settings. It can be clearly seen that the cohort structure of both two anchor embedding results do not correspond to the cohort relationship obtained by cohort embedding result. Therefore, we propose a mild adjustment to previously embedded results $\left\{\hat{\mathbf{u}}_i^{(0)}\right\}_{i=1}^{m}$ so that they align with $\{\hat{\mathbf{v}}_i\}_{i=1}^{C}$, but without changing the internal cohort structure. This is referred to as *anchor relocation process*.

Taking $\left\{\hat{\mathbf{u}}_i^{(0)}\right\}_{i=1}^{m}$ as an example. Simplifying the notation $c(\mathbf{u}_i)$ as c_i, we use the following rotation and scaling based transformation for the relocation:

$$\hat{\mathbf{u}}_i = a_c \left(\hat{\mathbf{u}}_i^{(0)} - \frac{1}{n_i} \sum_{c_p=c_i} \hat{\mathbf{u}}_p^{(0)} \right) \mathbf{R}_{c_i} + \hat{\boldsymbol{\upsilon}}_c, \tag{4.11}$$

where $\frac{1}{n_i} \sum_{c_p=c_i} \hat{\mathbf{u}}_p^{(0)}$ is the centroid of the initially embedded anchor points that belong to the same cohort as $\hat{\mathbf{u}}_i^{(0)}$. The rotation matrix $\mathbf{R}_c \in \mathbb{R}^{d \times d}$ and the scaling parameter $a_c \in \mathbb{R}$ control the transformation for anchor points from cohort $c \in \{1, 2, \ldots C\}$. We restrict the scaling parameter $0 \leq a_c \leq 1$, allowing only to shrink the cohort coverage in this relocation process. In the case of $d = 2$, the rotation matrix is constructed as

$$\mathbf{R}_c = \begin{bmatrix} \cos\theta_c & \sin\theta_c \\ -\sin\theta_c & \cos\theta_c \end{bmatrix}, \tag{4.12}$$

controlled by the angle parameter $-\pi \leq \theta_c \leq \pi$. Take the 2-D visualisation as an example, the transformation parameters are optimised by incorporating Eq. 4.11 into the following objective function:

$$\min_{\substack{\{-\pi \leq \theta_c \leq \pi\}_{c=1}^{C}, \\ \{0 \leq a_c \leq 1\}_{c=1}^{C}}} \sum_{(i,j,l) \in \Gamma^{\text{s-OE}}(\mathbf{U})} \max^2 \left(0, \hat{D}(\hat{\mathbf{u}}_i, \hat{\mathbf{u}}_j) + \delta - \hat{D}(\hat{\mathbf{u}}_i, \hat{\mathbf{u}}_l) \right), \tag{4.13}$$

A simple gradient descent approach is sufficient for optimisation. Figure 4.2d illustrates the final relocated anchor embeddings which has compatible cohort arrangement to what is captured by $\{\hat{\boldsymbol{\upsilon}}_i\}_{i=1}^{C}$ and meanwhile preserves the internal cohort structure. For 3-D visualisation, exactly the same approach is used, but with one more angle parameter added for each cohort.

4.5 Multi-objective Model Construction

To embed the data objects, the goal is to enable the data points to distribute appropriately around the anchor points following the same global structure underpinned by these anchors, and meanwhile to preserve reasonably well the local neighbourhood of each data object. We use the LAE reconstruction weights computed by Eq. 3.14 to bridge the embedded data and anchor points, and minimise an embedding reconstruction error together with a distance order preservation error. With the generated anchor points, we preserve the combination relationship between data samples and anchor points in the low-dimensional space as

$$\min_{\{\hat{\mathbf{x}}_i\}_{i=1}^n} O_{LAE} = \frac{1}{2}\sum_i \|\hat{\mathbf{x}}_i - \mathbf{w}_i\hat{U}\|^2. \tag{4.14}$$

This proposed objective function O_{LAE} can be seen as the global objective function. The optimum result can be solved out quickly as when each $\hat{\mathbf{x}}_i^{(0)} = \mathbf{w}_i\hat{U}$, the cost function O_{LAE} would achieve the minimum value 0. We denote $\hat{\mathbf{X}}^{(0)} = \{\hat{\mathbf{x}}_i^{(0)}\}_{i=1}^n$. However, reconstruction error is unavoidable, which will leading to a loss of the local neighbourhood preservation.

The LOE method has the privilege of preserving the neighbourhood information. Figure 3.1 in Section 3.3 suggests that LOE can reach the higher neighbour preservation score compared with other classical embedding methods. We treat Eq. 4.5 as the O_{LOE} to be target local objective function, and restate the equation here as

$$\min_{\{\hat{\mathbf{x}}_i\}_{i=1}^n} \sum_{(i,j,l)\in\Gamma^{\mathrm{LOE}}(\mathbf{X})} \max^2\left[0, \hat{D}(\hat{\mathbf{x}}_i,\hat{\mathbf{x}}_j) + \delta - \hat{D}(\hat{\mathbf{x}}_i,\hat{\mathbf{x}}_l)\right].$$

There are multiple ways to construct and optimise the multi-model of ANGEL. In this section we introduce two classical strategies of multi-model to set up their normal and straightforward optimisation process. We specify the *ANGEL algorithm* as the *r*-constrained model with the direct optimisation. In Section 6.1, we will propose several approximate accelerate optimisation process of ANGEL, and each of the optimisation process will have its own unique superscript or subscript to distinguish it.

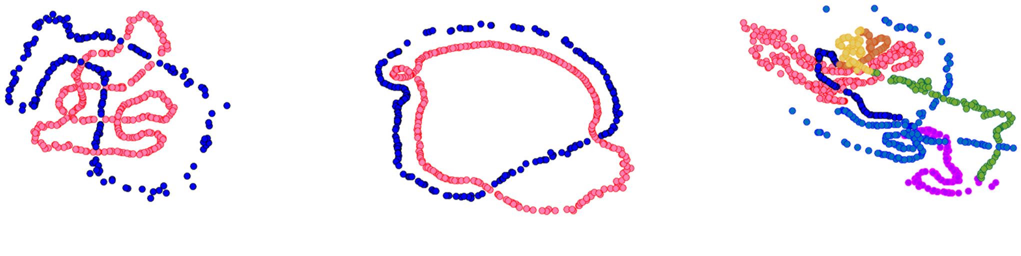
4.5.1 Weighted-sum Method

The weighted-sum method [140] depends on the importance of each sub-single objective problems.

$$\begin{aligned}
\min_{\{\hat{\mathbf{x}}_i\}_{i=1}^n} & \alpha O_{LAE} + (1-\alpha)O_{LOE} \\
&= \alpha\frac{1}{2}\sum_i \|\hat{\mathbf{x}}_i - \mathbf{w}_i\hat{U}\|^2 + \\
&(1-\alpha) \sum_{(i,j,l)\in\Gamma^{\mathrm{LOE}}(\mathbf{X})} \max^2\left[0, \hat{D}(\hat{\mathbf{x}}_i,\hat{\mathbf{x}}_j) + \delta - \hat{D}(\hat{\mathbf{x}}_i,\hat{\mathbf{x}}_l)\right]
\end{aligned} \tag{4.15}$$

If the α is chosen to be a large value, the feasible solution would be more focus on the global structure preservation, and vice versa.

The optimal solution $\hat{\mathbf{X}}^*$ can be found by solving the minimisation problem. A simple gradient descent method can be directly applied. Since two sub-single objective problems focus on different targets, they may have variables with different orders of magnitude. The scaling problem can be avoided by normalising the objective functions. However, if the magnitude difference is enormous, the normalising process has little effect. Figure 4.8 shows the embedding result of Circle dataset (which contains 500 data points) and Flower dataset (which contains 1000 data points) optimised by the weighted-sum model. The original data image can refer to Figure 4.3. Figure 4.8b shows that $\alpha = 0.9$ emphasises the global structure preservation. It could almost reveal the concentric circle shape. However, when the number of datapoint increases, Figure 4.8c cannot keep the shape of the flower when $\alpha = 0.9$. Furthermore, setting one weight as an extremely large value like 0.9 means sacrificing the other target hugely. Finding an optimal weight parameter α could be one of the challenging problems of the weighted-sum method.



(a) Circle, $\alpha = 0.1$ (b) Circle, $\alpha = 0.9$ (c) Flower, $\alpha = 0.9$

Figure 4.8: Embedding results of Circle and Flower dataset using the weighted-sum model.

4.5.2 r-constrained Model

Another popular way to formulate the multi-model is to construct a constrained problem [51]. The strategy is to optimise one objective function while the other objective function is limited to a constrained value [56] . Here, we convert the global objective function O_{LAE} into constraints, and use the radius parameter $r > 0$ to adjust the

Algorithm 3 ANGEL Algorithm

```
**Input**: $D$-dimensional data points $\{\mathbf{x}_i\}_{i=1}^n$ and their cohort labels $\{y_i\}_{i=1}^n$ .
**User-adjust hyperparameters**: Reduced dimension $d$ (2 or 3), cohort separation control
    $0 \leq \lambda \leq 1$; local neighbourhood size $k$; global vs local control $r < 1$.
**Fixed hyperparameters**: data selection threshold $thred = 2$, anchor point density $p = 0.1$;
    reconstruction anchor number $t = 3$; embedding scale $\delta = 0.1$.
Construct cohort ordinal quadruples $\Gamma_C^{\text{OE}}$ by Eq. 3.9 applying the *thred* data selection
    process, and use $\Gamma_C^{\text{OE}}$ to compute cohort embeddings $\{\hat{\mathbf{\upsilon}}_i\}_{i=1}^C$ by Eq. 3.10.
Obtain $D$-dimensional anchor point $\{\mathbf{u}_i\}_{i=1}^m$, anchor labels $\{c(\mathbf{u}_i)\}_{i=1}^m$, and data reconstruction
    weights $\{\mathbf{w}_i^*\}_{i=1}^n$ by applying Algorithm 2.
Construct anchor ordinal triplets $\Gamma^{\text{s-OE}}$ following Eq. 4.10, and use $\Gamma^{\text{s-OE}}$ to compute initial
    anchor embeddings $\left\{\hat{\mathbf{u}}_i^{(0)}\right\}_{i=1}^m$ by Eq. 4.8.
Compute relocated anchor embeddings by Eq. 4.11 and optimise Eq. 4.13 using $\left\{\hat{\mathbf{u}}_i^{(0)}\right\}_{i=1}^m$
    and $\{\hat{\mathbf{\upsilon}}_i\}_{i=1}^C$ to obtain the embedded anchor points $\{\hat{\mathbf{u}}_i\}_{i=1}^m$.
Construct data ordinal triplets $\Gamma^{\text{LOE}}(\mathbf{X})$ using Eq. 3.12, and compute data embeddings
    $\{\hat{\mathbf{x}}_i\}_{i=1}^n$ by optimising Eq.(4.16).
**Output**: Embedded data points $\{\hat{\mathbf{x}}_i\}_{i=1}^n$ in 2 (or 3)-D space.
```

preference of global preservation over the local preservation,

$$\begin{aligned} \min_{\{\hat{\mathbf{x}}_i\}_{i=1}^n} \quad & \sum_{(i,j,l)\in\Gamma^{\text{LOE}}(\mathbf{X})} \max{}^2\left[0, D(\hat{\mathbf{x}}_i, \hat{\mathbf{x}}_j) + \delta - D(\hat{\mathbf{x}}_i, \hat{\mathbf{x}}_l)\right], \\ \text{subject to} \quad & \left|\left|\hat{\mathbf{x}}_i - \hat{\mathbf{U}}^T \mathbf{w}_i^*\right|\right|_2^2 < r, i = 1, 2, \ldots, n, \end{aligned} \tag{4.16}$$

A suggested setting is $r \approx 0.05$ according to our empirical experience. The interior-point approach is used for optimisation. As for gradient calculation, the similar strategy which SOE applies as mentioned in Section 3.2 is adopted. In general, we use $t = 3$ anchors points to form the convex combination so that the reconstruction weights bear richer structural information as compared to weights computed using large t. We provide pseudo-code of the complete ANGLE implementation in Algorithm 3.

4.6 Complexity Analysis

ANGEL is a multi-step algorithm, for which we perform a brief complexity analysis:

- To generate anchor points by K-means clustering, the complexity is $O(n^2Dp/C)$.

This corresponds to running K-means once for each of the C cohorts. Each run partitions approximately $\frac{n}{C}$ data points in the D-dimensional space into $K \approx \frac{n}{C} \times p$ clusters.

- The LAE weight computation requires a computational cost of $O(t^2 n N_{\text{LAE}+tmn})$, where N_{LAE} denotes the number of iterations in LAE optimisation.
- The computational complexity of SOE and tri-OE is mainly dominated by the number of ordinal quadruples or triplets. In each optimisation iteration, the complexity is $O\left(C(C-1)(C-2)(C-3)\right)$ for cohort embedding, $O\left(n(n-1)(n-2)p^3/6\right)$ for anchor embedding, and $O\left(n(C-1)^2 n^2 p^3/6C^2\right)$ for anchor relocation.
- The computational complexity of the data embedding step is mainly dominated by the number of local ordinal constraints used by LOE, which results in a complexity of $O(kn(n-k))$ in each optimisation iteration.

In most data analysis practice, the cohort number C is within an acceptable range of $5-30$. The setting $p = 0.05$ is adopted and the iteration number in optimisation is usually between 30 and 200. This positions ANGEL as an algorithm with complexity between $O(n^2)$ and $O(n^3)$ with respect to the data point number n.

4.7 Chapter Summary

This chapter describes the construction process of the proposed ANGEL algorithm. It starts with the cohort embedding process, using the SOE method to locate the cohort position in 2,3-D space. Then anchor points are generated by K-means algorithm to represent each cohort. The proposed tri-OE method is then used to embed generated anchor points to maintain the intrinsic structure of the cohort, and a relocation method is used to allow the cohort positions to be retained concurrently. Finally, a multi-objective cost function is constructed by embedding the data points through the relationship between the data samples and the anchor points while maintaining the local neighbourhood of each data sample. Different optimisation methods such as weighted-sum methods and r-constrained method are also discussed in solving this multi-objective cost function. The ANGEL algorithm adopts the r-constrained method as the optimisation approach.

Chapter 5

Evaluation and Result

The previous chapter discussed the formulation of the ANGEL model thoroughly. The proposed multi-objective model ensures the balance between local neighbourhood preservation and global cohort relationship retention, and it also offers an option for cohort separability adjustment. The visual effect of the embedding result is an important evaluation of the data visualisation approaches. However, how to quantitatively measure the performance of the embedding result for cohort data visualisation is still an open question.

Embedding methods focusing on local structure preservation such as [118] and [141] introduce Trustworthiness [66, 125, 126], which is a measurement score for analysts to trust the embedding result to represent the actual local pattern of the dataset. To make it simpler, it measures how well the *k*-NN of each data sample have been retained around the corresponding embedded points in $2,3$-D space. However, for cohort data visualisation purpose, we can tolerate such an error when a point on the boundary of a neighbourhood in a high-dimensional space is embedded in the target $2,3$-D spaces, the embedded point is still on the boundary but not exactly in the corresponding neighbourhood, as Figure 5.1 shows. Thus, it is worth improving the local measurement to consider a slight relaxation on neighbourhood preservation and giving a more reasonable evaluation score.

McInnes et al. [90], Tang et al. [113], and Hajderanj et al. [53] utilise a *k*-NN classifier to evaluate how well the embedding result works for the supervised classification. *k* is a usually set as a small integer for the number of nearest neighbours in the classification, the *k*-NN classifier can also be used to evaluate how faithfully

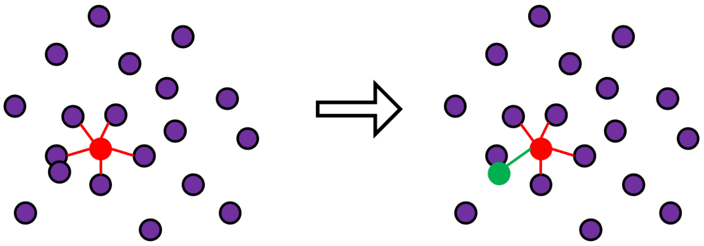


Figure 5.1: A brief explanation of the relaxation on neighbourhood preservation. The image on the left shows the 5-nearest neighbourhood points to the red dot. However, such an error is acceptable if the green points marked in the right image have been taken as neighbourhood points.

the reduced-dimensional data preserves the neighbourhood structure of the original data. However, in contrast to the the previously mentioned neighbourhood evaluation approach, the k-NN classifier does not provide a clear view of whether the high-dimensional neighbouring points are retained in the neighbourhood of the embedded points. Mu et al. [92] proposed 1-NN classification rate could be used to examine the separability between the embedded cohort cohorts. A higher classification accuracy indicates a stronger separability between cohorts.

As for the cohort distance preservation, which can also be treated as the global preservation of the dataset, most data visualisation approaches only discuss the visual effects of the embedded scatter plots. Only a few recent approaches have presented the evaluation metrics for the preservation of global structure. Sainburg et al. [103] introduces the Pearson correlation calculation based on pairwise distances comparisons between data points in high-dimensional space and embedding space. However, it does not emphasise on the preservation of cohort relations. Wang et al. [131] proposes Data Triplet Accuracy to measure how embedded points preserve data triplets randomly generated over original data samples, which can be treated as a comparison between the distance ranking of data points of the high-dimensional dataset and the embedded datasets. Inspired by the distance ranking comparison, we propose our global evaluation measure for the cohort structure preservation.

This chapter introduces the proposed evaluation metrics to measure the performance of the ANGEL. The experimental results show that we can perform better even compared to existing approaches in that both local and global targets can be maintained and balanced.

5.1 Evaluation

A classical and simple way for assessing embeddings is to compare for each point the neighbour match in the original and embedded spaces over a changing range of neighbourhood sizes (from local to global):

$$P(\overline{\kappa}) = \frac{1}{n\overline{\kappa}} \sum_{\kappa=1}^{\overline{\kappa}} \sum_{i=1}^{n} \frac{|\kappa\text{-NN}\,(\mathbf{x}_i) \cap \kappa\text{-NN}\,(\hat{\mathbf{x}}_i)|}{\kappa}, \tag{5.1}$$

where $\overline{\kappa}$ represents the maximum of the neighbourhood size considered. For instance, $\overline{\kappa} = \lfloor 0.9n \rfloor$ almost covers the whole data region, where n is the number of high-dimensional data samples. The κ-NN in this context returns the indices of the neighbouring points to enable a comparison between spaces. However, such a score is highly error sensitive, counting almost every single distance mistake. Given the very low reduced dimension $d = 2,3$, a low score is expected thus it shows the visualisation is not very informative in this case. It would be useful to employ measures that highlight more significant errors.

5.1.1 Local Evaluation on Neighbourhood Preservation

To examine how well a local neighbourhood is preserved, we adopt the main idea as above, alternatively substitute the average calculation based on the range of $\overline{\kappa}$ with the exact local neighbourhood preservation calculation on $\kappa = k_0$, where k_0 a hyper-parameter to determine the local neighbourhood preservation. Moreover, in order to filter out minor errors, we tolerate neighbourhood points that are wrongly placed but still quite close to the neighbourhood boundary by introducing a neighbouring relaxation strategy. This motivates the following score:

$$P_l = \sum_{i=1}^{n} \frac{\left|\kappa\text{-NN}\,(\mathbf{x}_i) \cap \kappa\text{-NN}_{\hat{\delta}}\,(\hat{\mathbf{x}}_i)\right|}{\kappa}, \tag{5.2}$$

A relaxed NN finder is applied in the embedded space:

$$\kappa\text{-NN}_{\hat{\delta}}\,(\hat{\mathbf{x}}_i) = \left\{ \hat{\mathbf{x}}_j \;|\, \|\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j\|_2 \leq \|\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_i^{e_1}(\kappa)\|_2 + \hat{\delta} \left\|\hat{\mathbf{x}}_i^{e_1}(\kappa) - \hat{\mathbf{x}}_i^{e_2}(\kappa)\right\|_2 \right\}, \tag{5.3}$$

FMNIST	cat-SNE	s-UMAP	Pysef	PaCMAP	COVA-1
Trustworthiness	0.9804	0.9783	0.8972	0.9739	0.8776
P_l	0.4307	0.3916	0.1745	0.3378	0.2440
MNIST					
Trustworthiness	0.9472	0.9211	0.7547	0.9278	0.7038
P_l	0.5008	0.3840	0.1887	0.3593	0.1861

Table 5.1: Comparison between trustworthiness score and proposed P_l score of MNIST and FMNIST embedding results generated by existing embedding techniques (details can be found in Section 5.2). The relative discrepancies of the local neighbourhood preservation results obtained by the two evaluation methods for different DR methods are the same, but P_l has the greater amount of difference among results.

where $\hat{\mathbf{x}}_i^{e1}(\kappa)$ and $\hat{\mathbf{x}}_i^{e2}(\kappa)$ denote the first and second farthest neighbours of $\hat{\mathbf{x}}_i$ within the k-NN region in the embedded spaces, the relaxation parameter $0 \leq \hat{\delta} < 1$ allows the placement of neighbouring points outside the target neighbourhood, but not too far away. When $\hat{\delta} = 0$, $\kappa\text{-NN}_{\hat{\delta}}(\hat{\mathbf{x}}_i) = \kappa\text{-NN}(\hat{\mathbf{x}}_i)$. A recommended setting according to empirical experience is $\hat{\delta} = 0.1$. A higher value of P_l indicates a better preservation of the local distance structure with the local region examined of a half size of the minimum cohort.

Table 5.1 shows the comparison between the existing trustworthiness score and the proposed P_l score of FMNIST embedding results generated by 5 existing embedding techniques (details can be found in Section 5.2. It is convincing that cat-SNE and s-UMAP share a high local neighbourhood preservation score while Pysef and COVA-1 suffer from loss of local information. However, P_l has more variation than trustworthiness. A larger amount of variation makes the comparison of the results of the different methods more evident and distinguishable, and the difference is not obliterated when averaged with the other two evaluation methods targeting global structure preservation and separability. Thus, the proposed P_l score is preferred.

5.1.2 Global Evaluation on Cohort Distance Preservation

To investigate the global level of cohort structure preservation, we compare the distance orders of between data cohorts in the original and embedded spaces. The following score is used based on the Spearman's rank correlation coefficient:

$$P_{\mathrm{g}} = \frac{1}{C}\sum_{c=1}^{C} \mathrm{spearman}(\boldsymbol{\pi}_c, \hat{\boldsymbol{\pi}}_c) = 1 - \frac{6}{C}\sum_{c=1}^{C} \frac{(\boldsymbol{\pi}_c - \hat{\boldsymbol{\pi}}_c)^2}{n(n^2-1)}, \tag{5.4}$$

where the $(C-1)$-dimensional vector $\boldsymbol{\zeta}_c$ stores the cohort rankings in terms of the closeness between the data cohort c and the remaining $C-1$ cohorts in the original space $\mathbb{R}^D$, while $\hat{\boldsymbol{\zeta}}_c$ is defined in the same way but in the embedded space $\mathbb{R}^d$. In order to be consistent with the previous cohort embedding process, we also use the Euclidean distance based average-linkage distance for obtaining $\boldsymbol{\zeta}_c$ and $\hat{\boldsymbol{\zeta}}_c$. Data selection step is also applied to compute the cohort rankings with the fixed $thred = 2$, in order to remove the influence of outlier points.

To note that, for both P_{l} and P_{g}, when computing distances between individual points, Euclidean distance is used for obtaining κ-NN$_{\hat{\delta}}$ and $\hat{\boldsymbol{\zeta}}_c$ to suit particularly the visualisation purpose. While for κ-NN and $\boldsymbol{\zeta}_c$ in the original space, other distance measures could be considered to suit better the nature of the data.

5.1.3 Cohort Separation Evaluation

Additionally, we report a cohort separation score P_{s} based on 1-NN classification following the standard cohort separation evaluation strategy [92]. We divide the embedded data points into μ partitions and compute the averaged 1-NN classification accuracy by apply μ-fold cross validation, using the embedded points and their cohort labels. Here, a default setting is $\mu = 5$ for simplicity. Table 5.2 shows cohort separation evaluation results of six real-world datasets (Section 5.2.1 provides details of these datasets) applying UMAP and s-UMAP algorithms. Both UMAP and s-UMAP have the same neighbourhood settings, and we keep other hyper-parameters from the original work. Images of embedding results can be found in Figures 5.8 - 5.13. It is clear that s-UMAP has a higher separation score than UMAP has, which is sufficient to confirm the accuracy of our cohort separation evaluation P_{s} for measuring the degree of separation.

5.1.4 Embedding Evaluation Score

Overall, a good visualisation is expected to possess high P_{l}, P_{g} and P_{s} in a balanced manner. Thus, we handle the embedding evaluation score as

Method		MNIST	FMNIST	Coil20	CIFAR	Isolet	Reuters
UMAP	P_s	0.8160	0.6910	0.9118	0.5700	0.7782	0.8620
s-UMAP	P_s	**0.9760**	**0.9660**	**1.0000**	**0.9410**	**0.9987**	**0.9749**

Table 5.2: P_s scores for UMAP and s-UMAP. The bold numbers highlight results which are higher.

$$P = \omega_l P_\text{l} + \omega_g P_\text{g} + \omega_s P_\text{s} = \frac{1}{3}P_\text{l} + \frac{1}{3}P_\text{g} + \frac{1}{3}P_\text{s}. \tag{5.5}$$

However, The coefficients for each score P_l, P_g, and P_s can be adjusted for different purposes. For example, if the intention of the algorithm is to retain more of the distance between each cohort, and the preservation of the neighbourhood near individual points is not of significant importance, then we can adjust the parameters so that the weight of P_g increases and the weight of P_l decreases. The freely adjustable weights give our metrics more flexibility and application scenarios.

5.2 Experiments and Results

In this section, we report on the results of a series of experiments conducted to assess the quality of ANGEL on several datasets, and compare it against leading DR methods. The qualitative measures show directly the graphical results obtained by the different methods, and they give us a very intuitive conclusion about what kind of visualisation effect we prefer to achieve. The quantitative results are based on calculations from previous measurement methods and show the strengths and weaknesses of the different methods and allow for more accurate comparisons. In addition, a study of the ANGEL parameters is reported to illustrate how different parameters affect the embedding results.

5.2.1 Datasets

Synthetic datasets: In order to see whether the ANGEL can work well on preserving the internal cohort structure, two 2-D synthetic datasets were annotated as in Figures 5.5a, 5.6a which we refer to circle and flower, and one 3-D synthetic dataset was annotated as in Figure 5.7a which we refer to bicycle. These synthetic datasets all have a relatively complex structure and have been artificially grouped into different point clusters.

- **Concentric circle (Circle)**: 2-D dataset which contains 500 data points belonging to 2 data cohorts [38].
- **Flower**: 2-D synthetic dataset which contains 1000 data points classified into 7 classes.
- **Bicycle**: 3-D synthetic dataset which contains 500 data points classified into 4 classes.

Real-world cohort datasets: Both qualitative measurements and quantitative evaluations are conducted on the real-world dataset.

- **MNIST**: 784-D dataset [136] consists of randomly selected 1000 images from 10 classes, with 100 images for each class. Examples of each class is shown is Figure 5.2a.
- **FMIST**: 784-D dataset [136] consists of random selected 2500 images of 10 classes, with 250 images for each class. Examples of each class is shown is Figure 5.2b.
- **Coil20**: 1024-D dataset [18, 17] consists of 1440 images of 20 objects. The images of each objects were taken 5 degrees apart as the object is rotated on a turntable and each object has 72 images. Examples of each class is shown is Figure 5.3.
- **CIFAR**: 4096-D CIFAR-10 dataset [36] consists of 1000 images of 10 classes. Each image was processed through a pre-trained CNN and extracted the activations in the first fully connected layer as the features. The pre-trained neural network applied VGG16 architecture that was trained on ImageNet data. Examples of each class is shown is Figure 5.4.
- **Isolet**: 1560-D dataset [18, 17] consists of 1560 signal vectors, each vector represent the voice recording of each letter of the alphabet, leading to 26 different classes.

(a) MNIST

(b) FMNIST

Figure 5.2: Images examples of MNIST dataset and FMNSIT dataset respectively. As for FMNIST, the label of each image from the top left to the right bottom are: 0-Top, 1-Trouse, 2-Pullover, 3-Dress, 4-Coat, 5-Scandal, 6-Shirt, 7-Sneaker, 8-Bag, 9-Ankle boot.

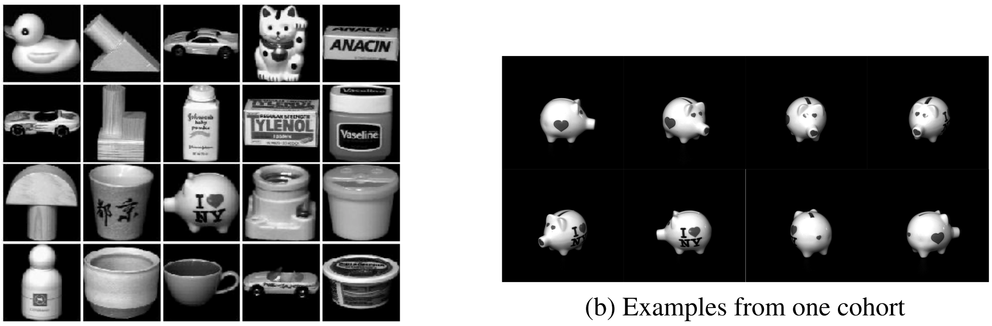
- **Reuters**: 5685-D Reuters-21578 corpus [16, 19, 20, 21] consists of 877 documents sampled from 10 classes: trade, ship, sugar, money-supply, cpi, cocoa, reserves, iron-steel, veg-oil, livestock.

As for synthetic and image datasets, feature distances were calculated by L2-pixel distance, while for textual datasets, calculation of cosine distances were applied.

5.2.2 Experimental Setup

ANGEL is compared with the six different state-of-the-art DR approaches on both qualitative measurements and quantitative evaluations. The cat-SNE [27] and the s-UMAP [90] are supervised extension of two popular data visualisation techniques, tSNE and UMAP. They are used to compare the advantages and disadvantages between the ANGEL algorithm and relatively traditional but popular local objective methods. PySef is a recent proposed supervised global objective algorithm which helps to test the global structure preservation performance of ANGEL. PaCMAP and COVA are both multi-objective models. We compare the advantages of ANGEL over these two methods for the same multi-objective purposes.

For competing algorithms with neighbourhood parameters, two types of neighbourhood size are examined for the competing local algorithms, including a small one $k = 10$ (the default setting of ANGEL), and a large one k close to $\left\lfloor \left(n^2 \log^d n\right)^{\frac{1}{d+2}} \right\rfloor$,



(a) Coil20

(b) Examples from one cohort

Figure 5.3: Images examples of Coil20 dataset. Figure 5.3a shows 20 cohorts and numbered them from 1 to 20, from the top left to the right bottom. Figure 5.3b presents examples selected from one cohort.

which is supposed to retain the global information of the high-dimensional dataset.

Other hyper-parameter settings of each algorithm is presented as the following:

- **ANGEL**: all of its fixed hyper-parameters are set as values reported in Algorithm 3. Its user-adjust hyper-parameters include $d = 2$ for 2-D visualisation; $\lambda = 0$ and $r = 0.05$ was set for the 3 synthetic datasets to illustrate the original data shapes; as for real-world datasets a suitable λ and r was chosen to obtain a better evaluation score $P = \frac{1}{3}(P_\text{l} + P_\text{g} + P_\text{s})$; and $k = 10$ for a small fixed neighbourhood size.
- **cat-SNE** [27]: as the supervised extension of tSNE, it introduced the threshold $\theta \in [0.5, 1]$ as the perplexity parameter. The suitable $theta$ is chosen by the maximum evaluation score P.
- **s-UMAP** [90][1]: keep the hyper-parameters from the original work.
- **PySef** [95]: LinearSEF model was applied with no specific hyper-parameters.
- **PaCMAP** [131]: the ratio setting for mid-near pair of points and further-pair of points were the default value.
- **COVA-1** [92]: it utilised the cohort position obtained by ANGEL as the position of its prototypes. The cohort membership parameter $\lambda = 0$. The balancing

[1] Manual for Supervised UMAP is located at https://readthedocs.org/projects/umap-learn/downloads/pdf/ latest/

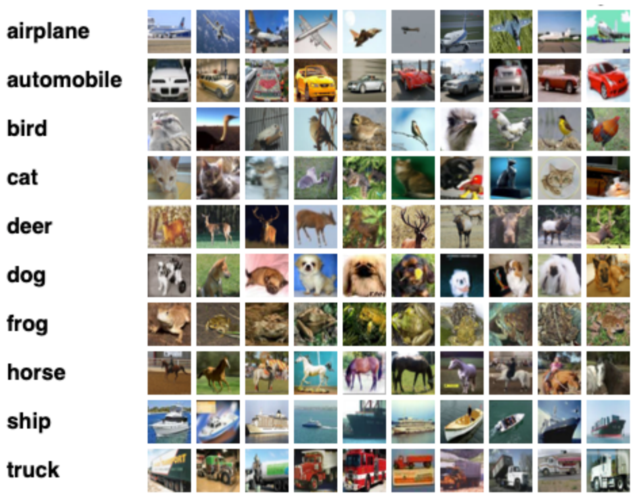


Figure 5.4: Images examples of Cifar dataset, which is obtained from https://www.cs.toronto.edu/ kriz/cifar.html.

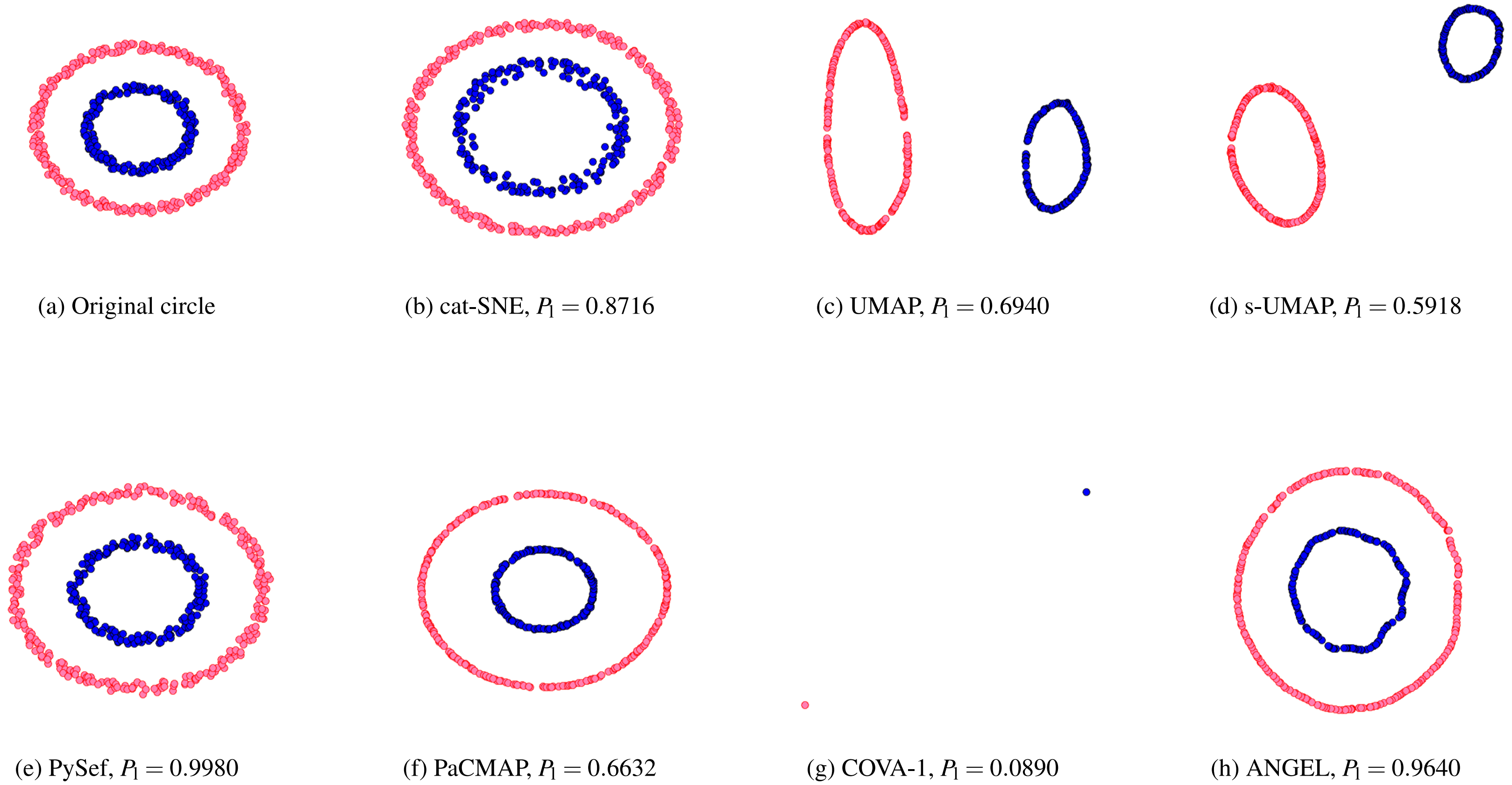
parameter α varies from 0.1 to 0.9 to find the best one with the maximum evaluation score P.

ANGEL is also compared with its transformation models including the ANGEL with warm-starting optimisation and weighted-sum model with loop-repetitive optimisation. The hyper-parameter study experiments results will be illustrated at the end.

All the experiments are conducted on CSF3 system supported by the University of Manchester. The CSF3 is a High Performance Computing (HPC) cluster (8,644 cores + 100 GPUs). The OS on CSF3 nodes is CentOS Linux release 7.9.2009 (Core). We adopted the standard Ivybridge cores: 38 nodes of 2×8-core Intel Xeon E5-2650 v2, 2.60GHz + 64GB RAM. ANGEL and COVA-1 was conducted based on MATLAB R2018a and other approaches were running on python.

5.2.3 Visualisation Assessment

Results of the 2-D and 3-D synthetic datasets are directly compared in Figures 5.5, 5.6, and 5.7. In order to make a comparative visualisation effects, neighbouring parameter



(a) Original circle

(b) cat-SNE, $P_{\mathrm{l}} = 0.8716$

(c) UMAP, $P_{\mathrm{l}} = 0.6940$

(d) s-UMAP, $P_{\mathrm{l}} = 0.5918$

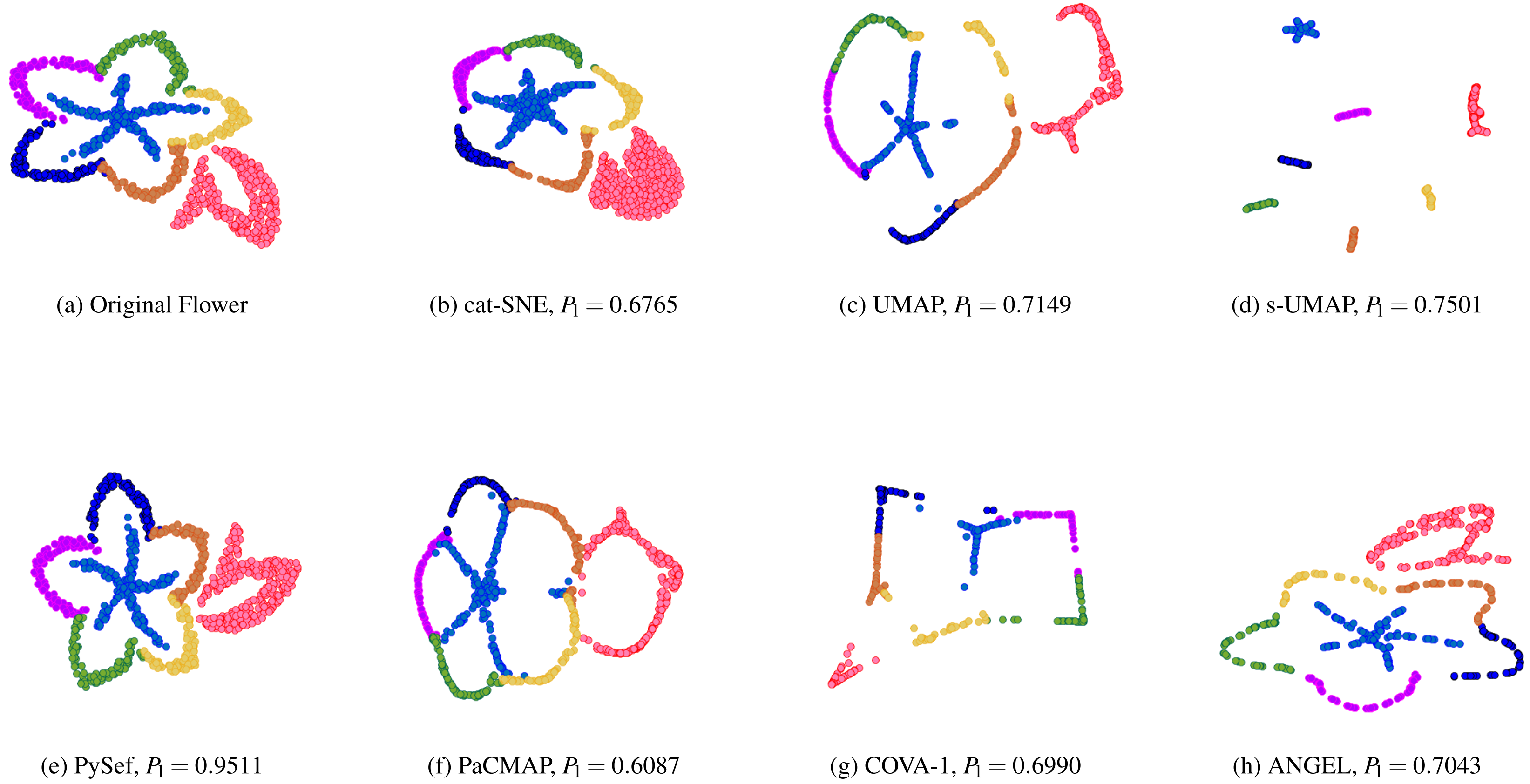
(e) PySef, $P_{\mathrm{l}} = 0.9980$

(f) PaCMAP, $P_{\mathrm{l}} = 0.6632$

(g) COVA-1, $P_{\mathrm{l}} = 0.0890$

(h) ANGEL, $P_{\mathrm{l}} = 0.9640$

Figure 5.5: Comparison of 2D Concentric Circle data embeddings. Different colours distinguish different data cohorts.



(a) Original Flower

(b) cat-SNE, $P_1 = 0.6765$

(c) UMAP, $P_1 = 0.7149$

(d) s-UMAP, $P_1 = 0.7501$

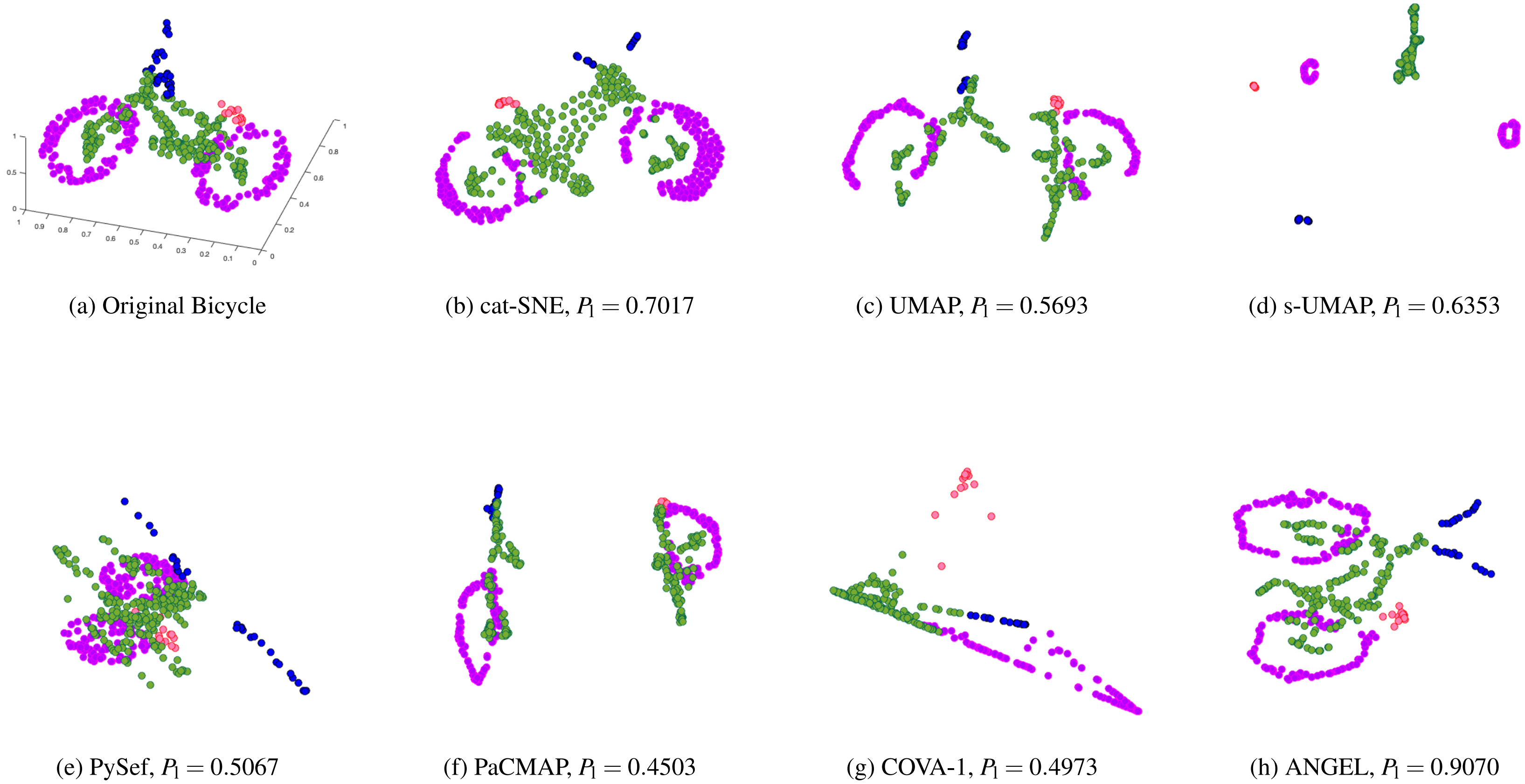
(e) PySef, $P_1 = 0.9511$

(f) PaCMAP, $P_1 = 0.6087$

(g) COVA-1, $P_1 = 0.6990$

(h) ANGEL, $P_1 = 0.7043$

Figure 5.6: Comparison of 2D Flower data embeddings. Different colours distinguish different data cohorts.



(a) Original Bicycle

(b) cat-SNE, $P_\mathrm{l} = 0.7017$

(c) UMAP, $P_\mathrm{l} = 0.5693$

(d) s-UMAP, $P_\mathrm{l} = 0.6353$

(e) PySef, $P_\mathrm{l} = 0.5067$

(f) PaCMAP, $P_\mathrm{l} = 0.4503$

(g) COVA-1, $P_\mathrm{l} = 0.4973$

(h) ANGEL, $P_\mathrm{l} = 0.9070$

Figure 5.7: Comparison of 2D bicycle data embeddings. Different colours distinguish different data cohorts.

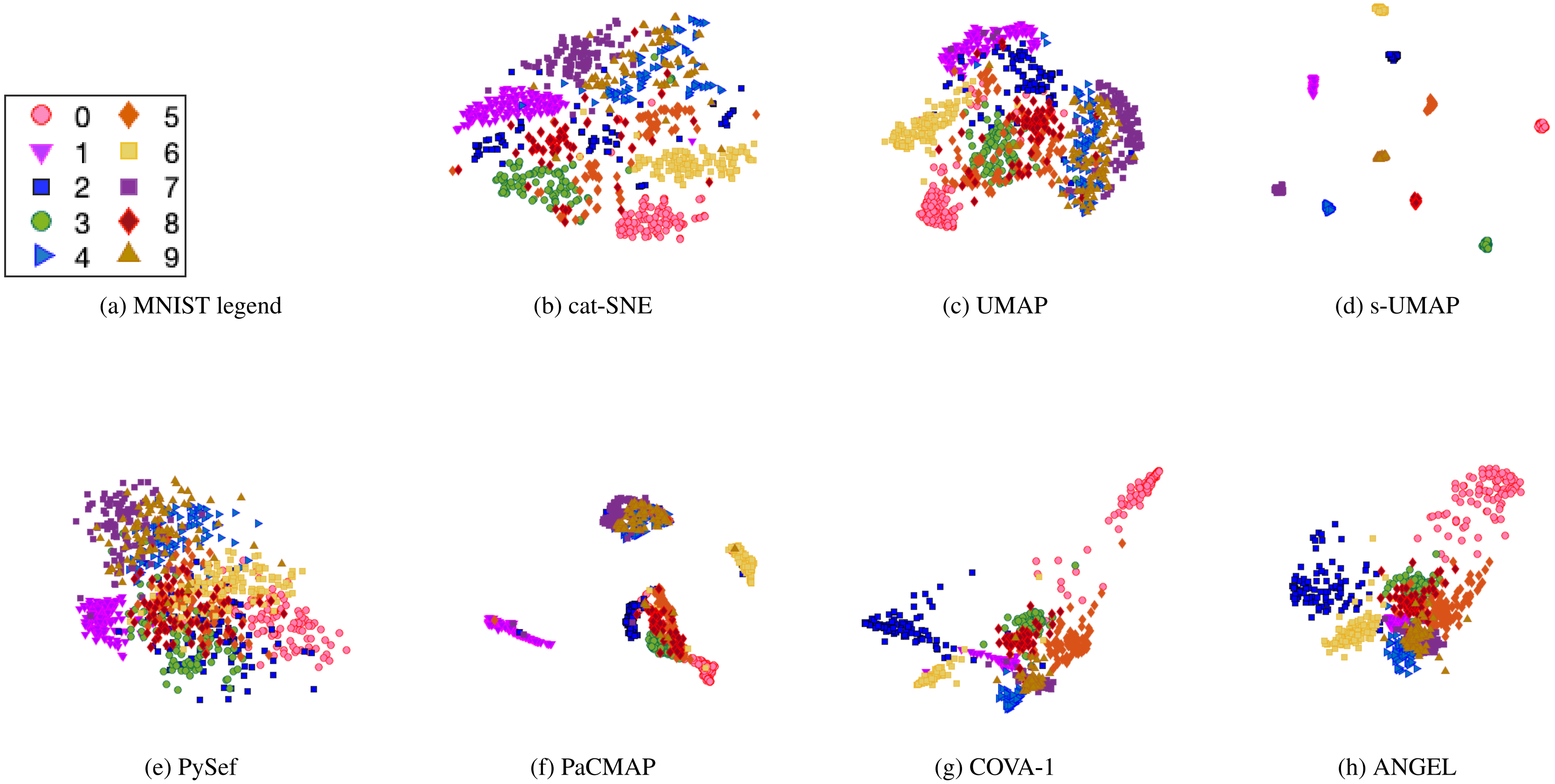
for each approaches (except for ANGEL k=10) are chosen to be the large neighbourhood $k = \left\lfloor \left(n^2 \log^d n\right)^{\frac{1}{d+2}} \right\rfloor$. It can be seen from Figure 5.5e and Figure 5.6e that PySef is particularly good with preserving both the correct data shape and the correct local neighbourhood (high P_l score) for the ideal 2-D to 2-D mapping case. However, as for a simple 3-D to 2-D case Figure 5.7e, it tries to project the bicycle from the top side leading to a mixed projections which is hard to identify cohorts. PySef also does not provide any option to change the direction of the projection for visualisation purpose. Moreover, as shown in the later results in Table 5.3, its performance drops significantly where being used to visualise high-dimensional data when information loss is unavoidable in the reduced space.

None of the UMAP or the s-UMAP can ideally recover the shape of the synthetic dataset. COVA-1 is good at positioning cohorts in an appropriate place but fails to display the internal cohort structure, particularly for the concentric circle and bicycle cases. The cat-SNE method performs better than PySef since it can capture the global structure but with a lower local neighbour preservation score. ANGEL can recover the original data shape as well as PySef, but also with a bit lower local neighbour preservation score.

Then, we visually compare the results of the 6 high-dimensional real-world datasets in Figures 5.8-5.13. For image datasets MNSIT and FMNSIT, most algorithms are able to match their cohort closeness with the image content based similarity, e.g., numbers like 4 and 9, 1 and 7 possess similar shapes, while numbers like 3, 5, and 8 are more similar to each other. Similarly, “Ankle boots”, “Sandal”, and “Sneaker” all belong to the shoes category, while “Top”, “Trouser”, and “Pullover” are clothes with close shape. Overall, cat-SNE and ANGEL offer better cohort separation and identification than other methods and display more meaningful internal cohort structures than s-UMAP.

For the third image dataset Coil20, we manually assign the same colour to cohorts whose items share a similar shape; for example, the number 3 cohort, the number 6 cohort, and the number 9 cohort are all cars; thus we assign the purple colour to these three cohorts. This helps us identify the cohort relations preservation in a much more convenient way. PySef, PaCMAP, COVA and ANGEL can keep the global structure well according to Figure 5.10. However, only ANGEL can capture most of the internal cohort structure as a circle.

The results of Cifar dataset are shown in Figure 5.11. We compare the image



(a) MNIST legend

(b) cat-SNE

(c) UMAP

(d) s-UMAP

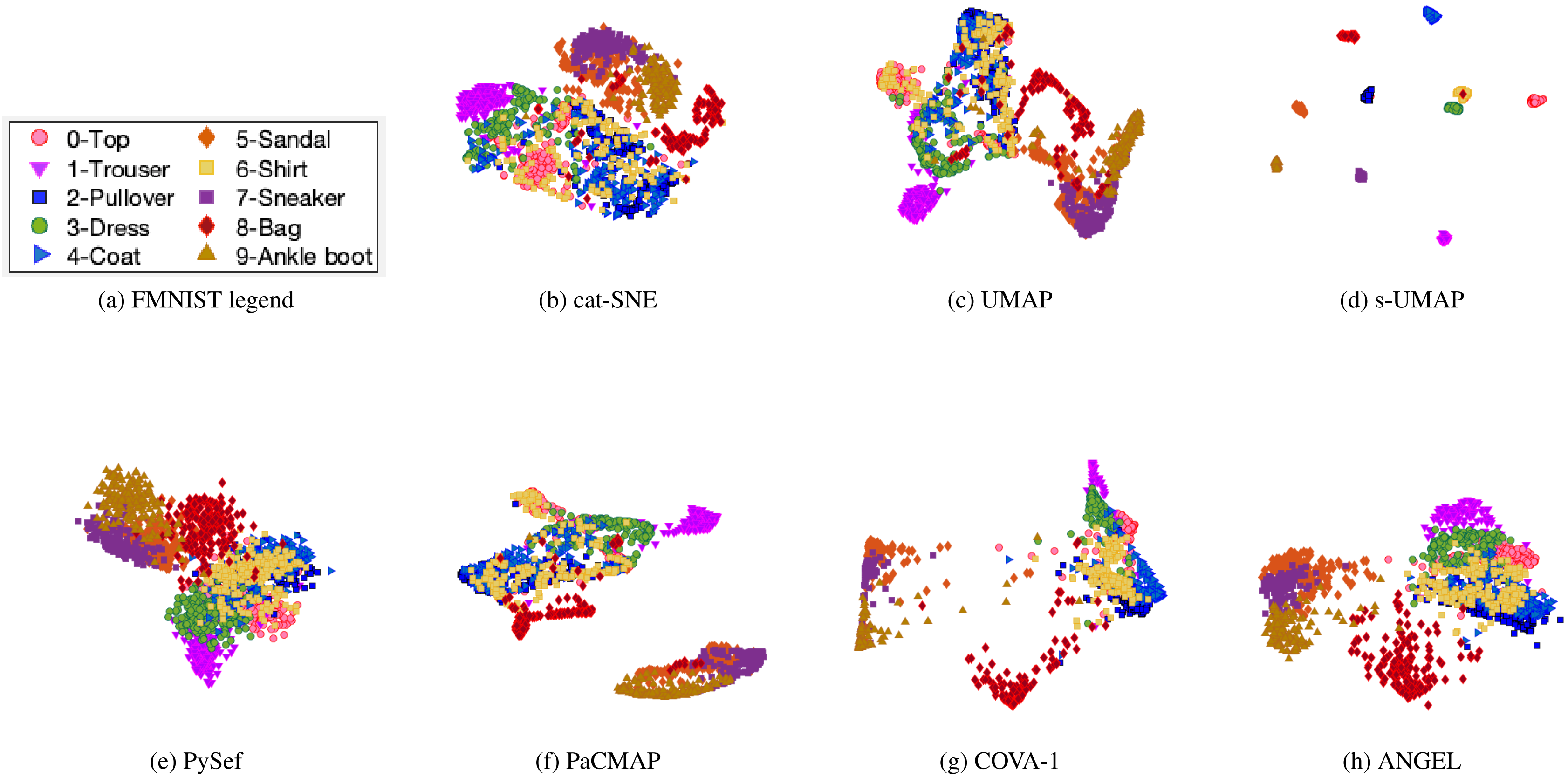
(e) PySef

(f) PaCMAP

(g) COVA-1

(h) ANGEL

Figure 5.8: Comparison of MNIST data embeddings. Different colours distinguish different data cohorts.



(a) FMNIST legend

(b) cat-SNE

(c) UMAP

(d) s-UMAP

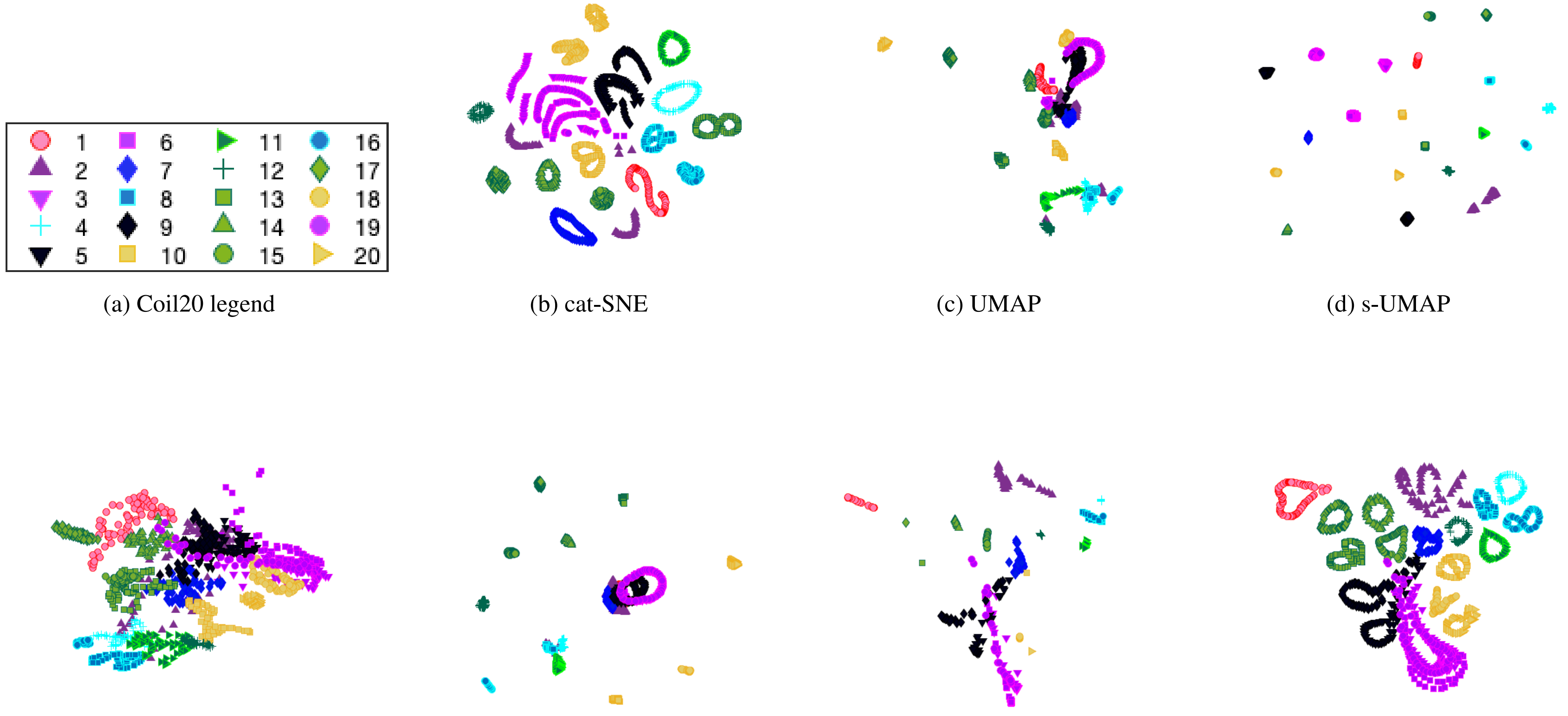
(e) PySef

(f) PaCMAP

(g) COVA-1

(h) ANGEL

Figure 5.9: Comparison of FMNIST data embeddings. Different colours distinguish different data cohorts.



(a) Coil20 legend

(b) cat-SNE

(c) UMAP

(d) s-UMAP

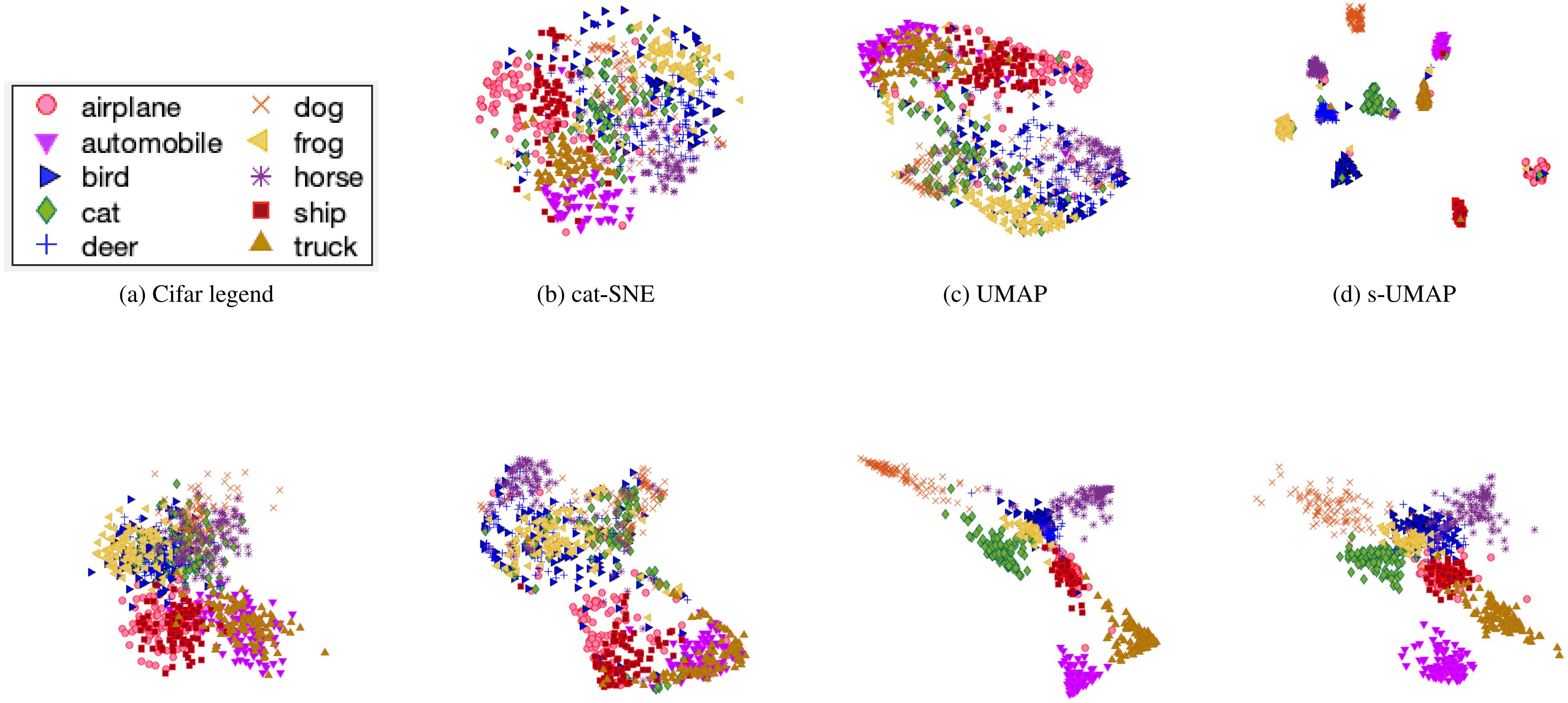
(e) PySef

(f) PaCMAP

(g) COVA-1

(h) ANGEL

Figure 5.10: Comparison of Coil20 data embeddings. Different colours distinguish different data cohorts.



(a) Cifar legend

(b) cat-SNE

(c) UMAP

(d) s-UMAP

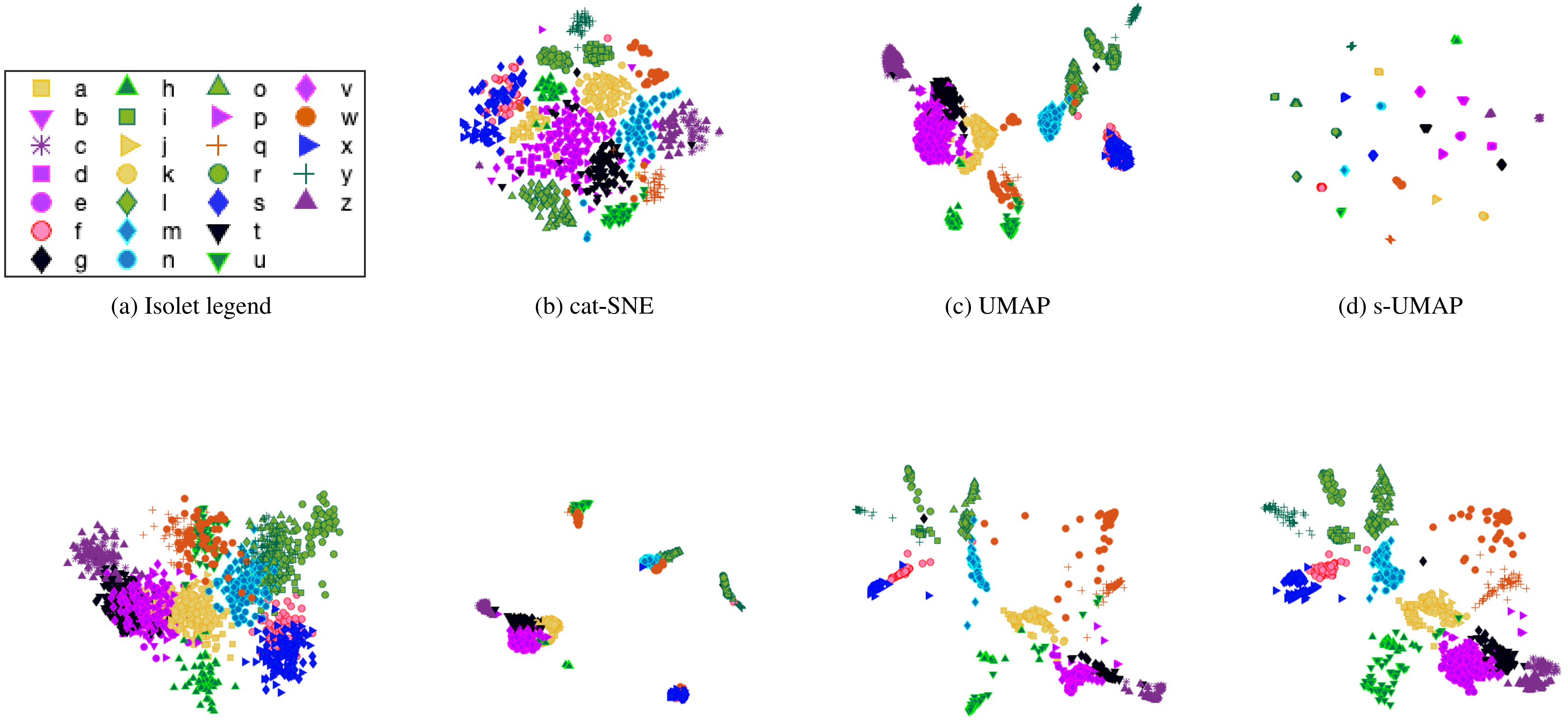
(e) PySef

(f) PaCMAP

(g) COVA-1

(h) ANGEL

Figure 5.11: Comparison of Cifar data embeddings. Different colours distinguish different data cohorts.



(a) Isolet legend

(b) cat-SNE

(c) UMAP

(d) s-UMAP

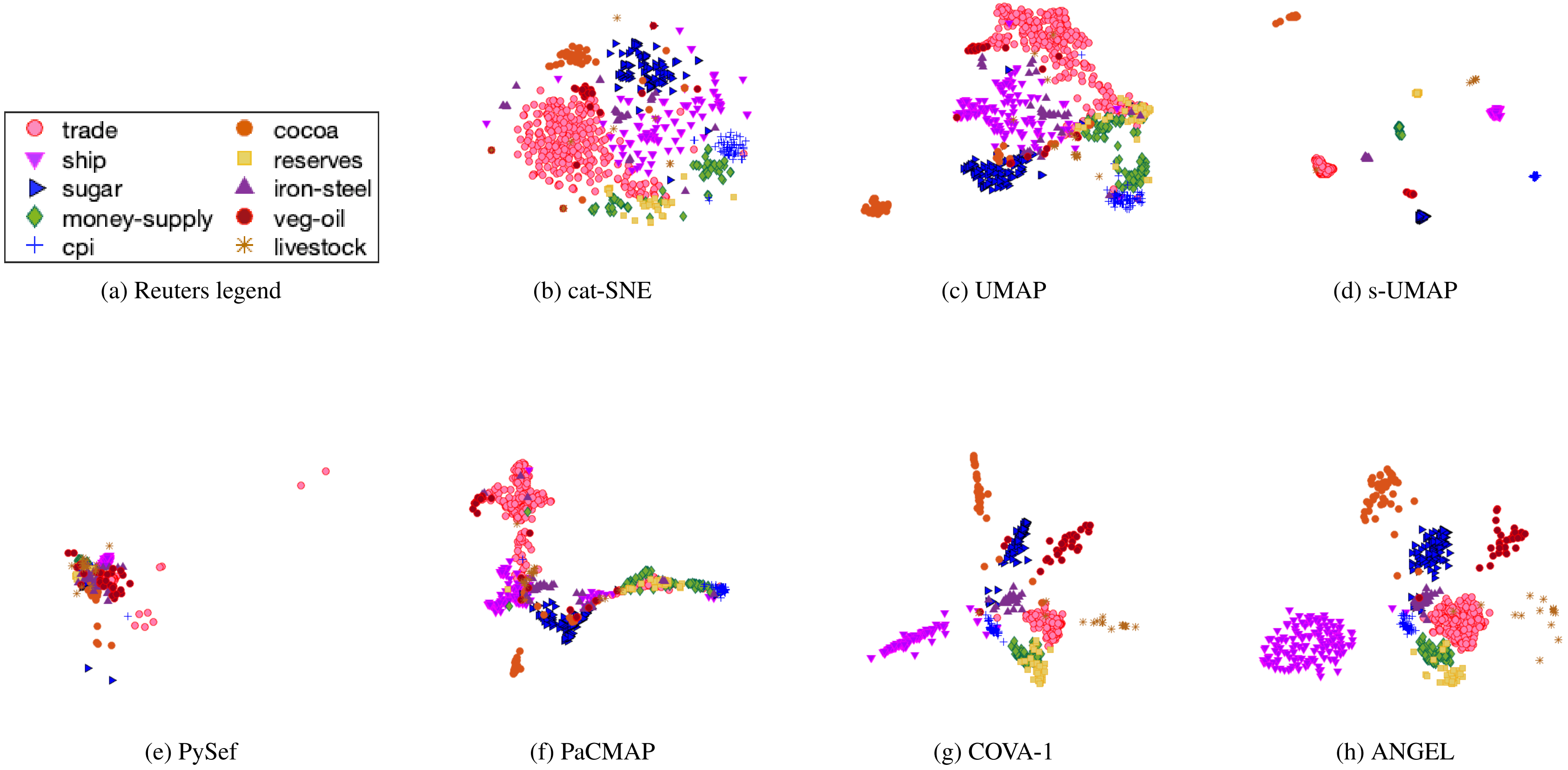
(e) PySef

(f) PaCMAP

(g) COVA-1

(h) ANGEL

Figure 5.12: Comparison of Isolet data embeddings. Different colours distinguish different data cohorts.



(a) Reuters legend

(b) cat-SNE

(c) UMAP

(d) s-UMAP

(e) PySef

(f) PaCMAP

(g) COVA-1

(h) ANGEL

Figure 5.13: Comparison of Reuters data embeddings. Different colours distinguish different data cohorts.

content based similarity via their cohort closeness, e.g., automobile and truck cohorts are closer, and so are cat and dog cohorts. Results of cat-SNE, UMAP, and PySef can present the expected relations of these image cohorts, but points are more often mixed. COVA and ANGEL are able to present distinguished cohorts with a good cohort relationship preservation.

Figure 5.12 shows results of Isolet dataset. Similar to the Coil dataset, we manually assign the same colour to the alphabet whose pronunciation is similar, such as "b" and "d", "m" and "n". UMAP, Cova, and ANGEL are capable of identifying each cohort and revealing their cohort relations. PySef and PaCMAP almost capture the global structure of the cohort data; however, PySef suffers from mixed points and cohorts while PaCMAP lost most of the internal structure of the cohort.

As for the text dataset Reuters (Figure 5.13), cat-SNE, UMAP, COVA-1, and ANGEL can match their cohort arrangement with the text content based similarity, e.g., the money-supply cohort is closer to cpi and reserves, veg-oil to cocoa. Although PaCMAP can barely retain cohort relations, it is hard to identify different cohorts.

Overall, most methods can result in a relatively good visual effect if a well-designed parameter can be found. This is in line with the discussion we put forward earlier in Section 1.1. ANGEL and COVA are among the best algorithms that can consistently obtain good visualisation results. At the same time, other methods are more or less likely to show less than satisfactory results in some datasets. ANGEL is the only one who can successfully show the inherent cohort structure of the Coil20 dataset.

The following section will compare and discuss the evaluation results among all approaches.

5.2.4 Comparative Analysis

We have mentioned in Section 1.1 that, many existing visualisation techniques are sensitive to its neighbourhood size, which is often difficult to choose. One advantage of ANGEL is that it is sufficient to use a small neighbourhood size. We first compare ANGEL with the competing methods under the same neighbourhood size setting of $k = 10$. This is shown by Table 5.3, which reports the P_l, P_g and P_s scores for the 5 high-dimensional real-world datasets. The cat-SNE, s-UMAP, PaCMAP have relatively high local preservation score P_l as expected, while COVA and PySef is much better at cohort positioning with higher P_g. We notice that the approaches of cat-SNE

Method		cat-SNE	s-UMAP	PySef	PaCMAP	COVA-1	ANGEL
MNIST	P_l	**0.5008** (1)	0.3840 (2)	0.1887 (5)	0.3593 (3)	0.1861(6)	0.2617 (4)
	P_g	0.1794 (5)	0.1648 (6)	0.4261 (3)	0.2885 (4)	0.5697 (2)	**0.6170** (1)
	P_s	0.9020 (2)	**0.9760** (1)	0.5160 (6)	0.7450 (5)	0.8870 (3)	0.8610 (4)
	P	0.5274(3)	0.5083 (4)	0.3169 (6)	0.4643 (5)	0.5476 (2)	**0.5799** (1)
FMNIST	P_l	**0.4307** (1)	0.3916 (2)	0.1745 (6)	0.3378(3)	0.2440 (5)	0.3211 (4)
	P_g	0.5224 (4)	0.2800 (6)	0.6000 (3)	0.5200(5)	0.7321 (2)	**0.7988** (1)
	P_s	0.7985 (3)	**0.9660** (1)	0.5765(6)	0.6585 (5)	0.8750 (2)	0.7620 (4)
	P	0.5081 (4)	0.5459 (3)	0.4503 (6)	0.5054 (5)	0.6170 (2)	**0.6276** 1)
Coil20	P_l	**0.6506** (1)	0.5642 (2)	0.3006 (6)	0.5385 (4)	0.4219(5)	0.5426(3)
	P_g	0.0071 (5)	0.0634 (4)	0.1041 (3)	-0.0726 (6)	0.2125(2)	**0.2250**(1)
	P_s	0.9988 (2)	**1.0000** (1)	0.8104(6)	0.8937 (5)	0.9785 (3)	0.9312(4)
	P	0.5522 (2)	0.5425 (3)	0.4051 (6)	0.4532 (5)	0.5376 (4)	**0.5663** (1)
CIFAR	P_l	0.2606 (2)	0.2349 (3)	0.1006 (6)	**0.2739** (1)	0.1378 (5)	0.2258 (4)
	P_g	0.2070 (4)	0.0461 (6)	0.0533 (5)	0.2805 (2)	**0.2873** (1)	0.2752 (3)
	P_s	0.7560(4)	**0.9410** (1)	0.3580 (6)	0.4860 (5)	0.8480 (2)	0.7810 (3)
	P	0.4079 (3)	0.4073 (4)	0.1706 (6)	0.3468 (5)	0.4244 (2)	**0.4273** (1)
Isolet	P_l	**0.4425** (1)	0.3243 (2)	0.1918 (5)	0.3235 (4)	0.0592 (6)	0.3212 (3)
	P_g	0.0436 (6)	0.1375 (5)	0.2237 (3)	0.1578 (4)	**0.4630** (1)	0.2458 (2)
	P_s	0.9270 (4)	**0.9987** (1)	0.4263 (6)	0.6936 (5)	0.9453 (3)	0.9897(2)
	P	0.4711 (4)	0.4868(3)	0.2246 (6)	0.3916 (5)	0.4892 (2)	**0.5189** (1)
Reuters	P_l	**0.3146** (2)	0.4390 (1)	0.1213 (5)	0.3876(3)	0.0545 (6)	0.2458 (4)
	P_g	0.0909 (6)	0.0521 (5)	0.1661 (4)	0.3030 (3)	**0.5103** (1)	0.3212 (2)
	P_s	0.9327 (3)	0.9749 (2)	0.8164(6)	0.8472 (5)	0.9225 (4)	**0.9897** (1)
	P	0.4904 (4)	0.4887 (5)	0.3667(6)	0.5126 (2)	0.4957 (3)	**0.5189** (1)

Table 5.3: Visualisation scores of different methods with the best bold and the second underlined. Relative ranks are displayed in brackets. To make the result comparable, all the neighbourhood parameter $k = 10$. Other hyper-parameter settings followed the experimental setup.

and s-UMAP have very poor P_g scores. Such result is not surprising as the neighbourhood parameter is set to a small number. Unlike COVA-1 and ANGEL, they are not designed to match the distances between the embedded cohorts to a desired set of between-cohort distances in the original space. PaCMAP and Pysef suffers from low separation score P_s. PySef and COVA also has a relatively low neighbour preservation score comparing with ANGEL.

Overall, ANGEL achieves the highest $P = \frac{1}{3}(P_\mathrm{l} + P_\mathrm{g} + P_\mathrm{s})$ among all the compared approaches and for all the 5 datasets. Local neighbourhood preservation score P_l have been sacrificed as expected, but not that significantly. ANGEL can best preserve the positioning relationships between point clouds, as well as the relatively good separation between point clouds.

Method		MNIST	FMNIST	Coil20	CIFAR	Isolet	Reuters
cat-SNE	ΔP_g	-0.2097	-0.0679	**0.0980**	**0.0863**	**0.0871**	**0.2485**
s-UMAP	ΔP_g	-0.0627	**0.1830**	**0.0088**	**0.0654**	-0.0800	-0.0412
PySef	ΔP_g	-0.2116	-0.1794	**0.0617**	**0.1346**	**0.0156**	-0.1394
PaCMAP	ΔP_g	-0.0715	-0.1394	**0.3079**	-0.1072	-0.1102	-0.2206

Table 5.4: ΔP_g scores for s-tSNE, s-UMAP, PySef and PaCMAP, which is the difference between P_g the using single-linkage distance to calculate between-cohort distances and P_g using average distance to do so. The bold numbers highlight the positive results which infer that the P_g scores have been improved compared with 5.3 after change the cohort distance measurement.

Method		MNIST	FMNIST	Coil20	CIFAR	Isolet	Reuters
cat-SNE	ΔP_g	-0.6473	-0.3443	-0.1199	**0.0181**	-0.1151	**0.0182**
s-UMAP	ΔP_g	-0.5152	-0.3358	-0.1528	-0.1637	-0.1883	-0.3103
PySef	ΔP_g	-0.4025	-0.3782	-0.0592	-0.0873	-0.0065	-0.2945
PaCMAP	ΔP_g	-0.4000	-0.4182	**0.0103**	-0.1019	-0.1982	-0.2388

Table 5.5: ΔP_g scores for s-tSNE, s-UMAP, PySef and PaCMAP, which is the difference between P_g the using single-linkage distance to calculate between-cohort distances and P_g of ANGEL algorithm using average distance to do so. The bold numbers highlight the positive results which infer that the P_g scores of SOTA methods higher than ANGEL algorithm after change the cohort distance measurement.

In case these methods, by any chance, preserve certain cohort structural information that may not be compatible to the average-linkage distances that we used in evaluation, we further experiment with multiple between-cohort distance measures and choose the best one, which is the single-linkage distance, to recalculate their P_g scores. Table 5.4 reports the ΔP_g, which is the difference between P_g the using single-linkage distance to calculate between-cohort distances and P_g using average distance to do so. The scores are improved in many cases, but overall, they are still not satisfactory since only few of the P_g results are comparable to results of ANGEL, which has been reported in Table 5.5.

Additionally, we report performance for the ocal method s-UMAP using a larger neighbourhood size $k = \left\lfloor \left(n^2 \log^d n\right)^{\frac{1}{d+2}} \right\rfloor$. Although PaCMAP and COVA-1 are not local methods, they have the neighbourhood parameter choice that can be adjusted. Table 5.6 reports the evaluation score of the large k.

It can be seen that the overall evaluation scores do not improve much as the k

Method		MNIST	FMNIST	Coil20	CIFAR	Isolet	Reuters
		$k=55$	$k=81$	$k=55$	$k=55$	$k=50$	$k=70$
s-UMAP	P_l	$\underline{0.4114}$*	$\underline{0.3746}$	0.5218	$\underline{0.2349}$	0.3029	$\underline{0.3414}$
	P_g	0.2085*	0.3261*	0.0071	0.0461	0.0604	0.0097
	P_s	$\underline{1.0000}$*	$\underline{0.9950}$	$\underline{1.0000}$	$\underline{0.9400}$	$\underline{0.9994}$	$\underline{0.9989}$*
	P	0.5400*	0.5652*	0.5096	0.4070	0.4542	0.4500
PaCMAP	P_l	$\underline{0.3593}$	0.3077	0.5368	$\underline{0.2805}$*	0.2922	$\underline{0.2624}$
	P_g	0.2885	0.5624*	0.0867*	0.2739	0.1525	0.2097
	P_s	0.7450	0.6455	0.8396	0.4740	0.7481*	0.7879
	P	0.4643	0.5052	0.4877*	0.3428	0.3976*	0.4200
COVA-1	P_l	0.2212*	0.2262	0.4378*	0.1620*	0.2526*	0.0545
	P_g	0.5818*	0.7442*	0.1676	0.1442	$\underline{0.2896}$	$\underline{0.5103}$
	P_s	0.8420	$\underline{0.8485}$	$\underline{0.9757}$	$\underline{0.7920}$	0.9026	0.9236*
	P	0.5483	0.6063	0.5270	0.3661	0.4807	0.4961*

Table 5.6: Visualisation scores for s-UMAP, PaCMAP and COVA with larger k which is calculated by the given k which is close to the $\left\lfloor \left(n^2 \log^d n\right)^{\frac{1}{d+2}} \right\rfloor$ based on each dataset. The * highlights the improving score after using the larger k compared with Table 5.3. The underlined number refers to the score which is higher than score of ANGEL obtained in Table 5.3.

becomes larger. None of the new P is higher than the P of ANGEL results. As for their global score P_g is not very sensitive to the choice of the neighbourhood size for the high-dimensional datasets, but the local neighbour preservation score P_l can drop with larger k in some cases. Moreover, increasing k may even reduce the cohort separability P_s. The overall preservation score P is still lower than the result obtained by ANGEL in Table 5.3.

In summary, our evaluation approach successfully proved that cat-SNE and s-UMAP are good at preserving local neighbourhood while PySef and COVA-1 are capable of maintaining the cohort position as a global structure preservation. These results prove the reliability of our evaluation tools. Therefore, we can then claim that our ANGEL model outperforms all other methods based on the overall P score, as it balances all the three aspects: local neighbourhood preservation, cohort position preservation, and cohort separability.

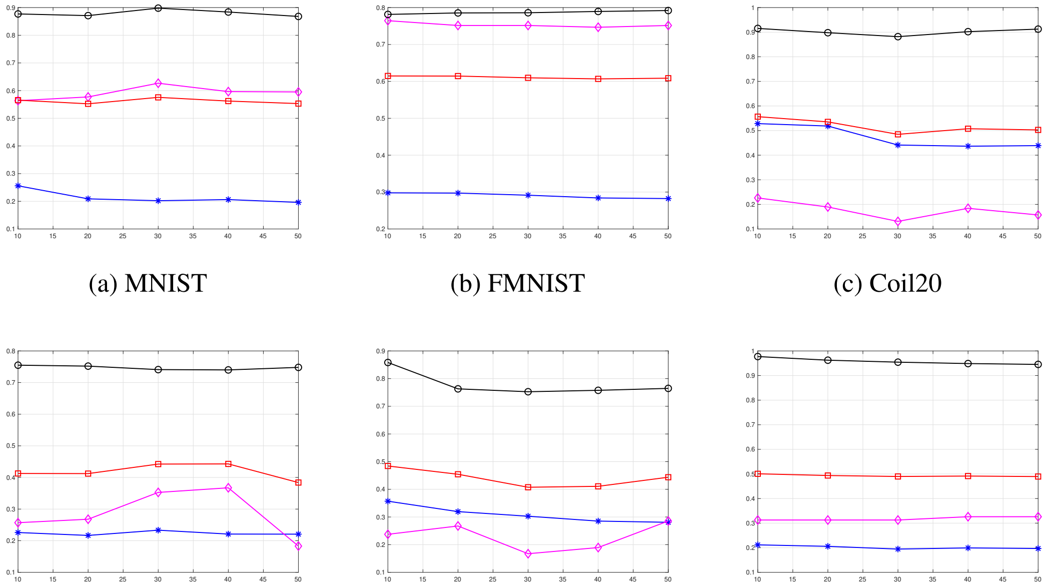
5.2.5 ANGEL's Parameter Study

As shown in Algorithm 3, there are some important hyper-parameters: the local neighbourhood size k, the cohort separation control λ, and the global vs local control r.

In this section, we will compare ANGEL results for different values of these hyper-parameters. The target is to investigate how this hyper-parameters will influence the embedding result of ANGEL.

First, we change the local neighbourhood size k from 10 to 50. Since this hyper-parameter focuses on local neighbourhood preservation, we are not intend to set k to a very large value. Figure 5.14 shows the trend line of the evaluation scores of all six real-world dataset. It can be deducted that increasing k does not leading to a significant change on the preservation score for all these dataset. When $k = 50$, some P_l, P_g, and P even have a sizeable decline. Figure 5.15 also shows that changing neighbourhood size k would not change much on visualisation. A smaller k may even produce a better evaluation score and visualisation image.

Thus we can conclude that our ANGEL algorithm is insensitive to the neighbourhood choice, which is much more convenient for analysts to apply.



(a) MNIST (b) FMNIST (c) Coil20

(d) Cifar (e) Isolet (f) Reuters

Figure 5.14: Preservation scores of ANGEL embedding results of the different datasets applying different $k = 10, 20, 30, 40, 50$. The x axis of each sub-figure denotes different k value, the y axis of each sub-figure denotes the preservation value. Different colours refer to different preservation score respectively: blue: P_l, pink: P_g, black: P_s, red: P.

Then we set the cohort separation control λ to different values, and Figure 5.17

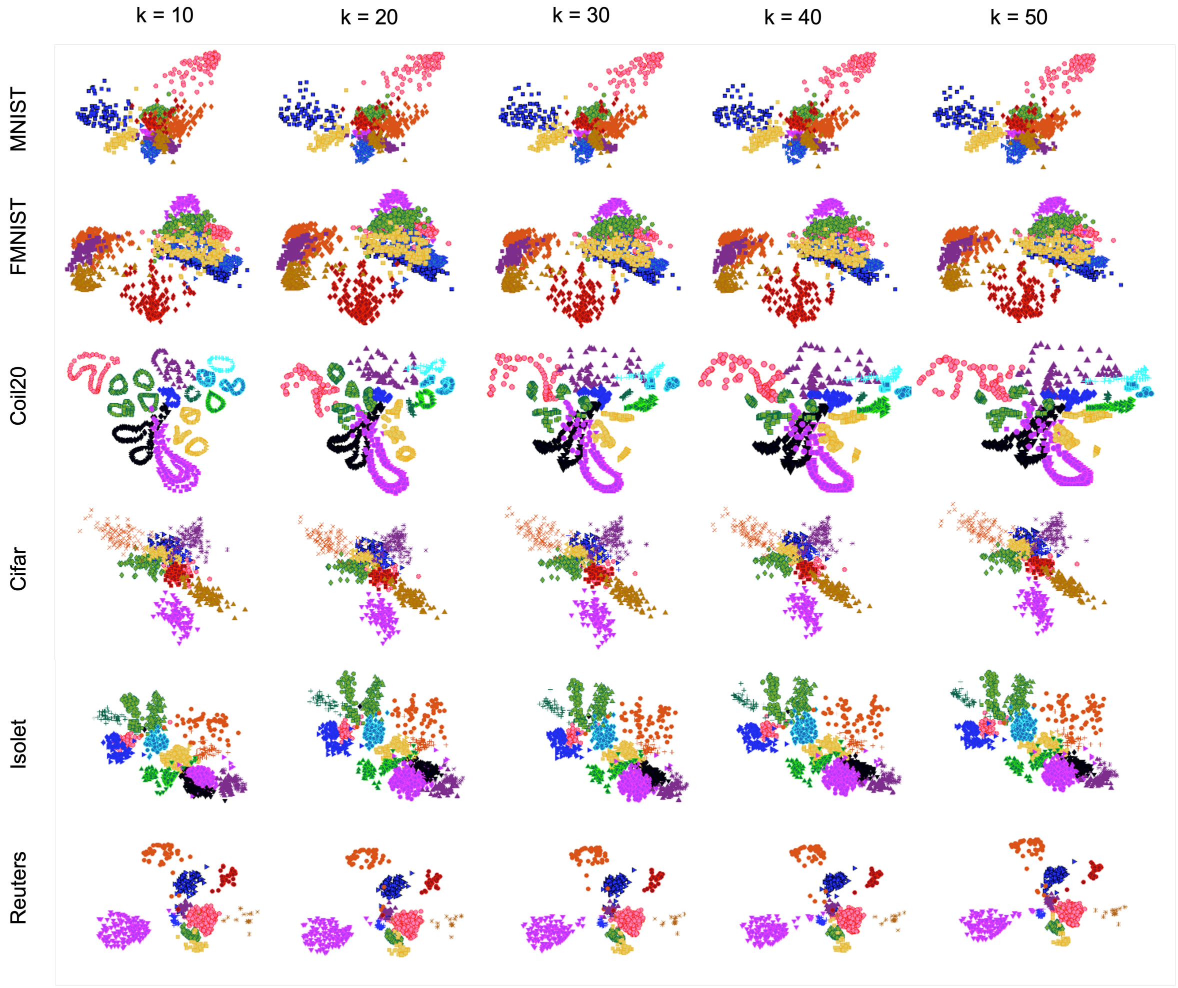
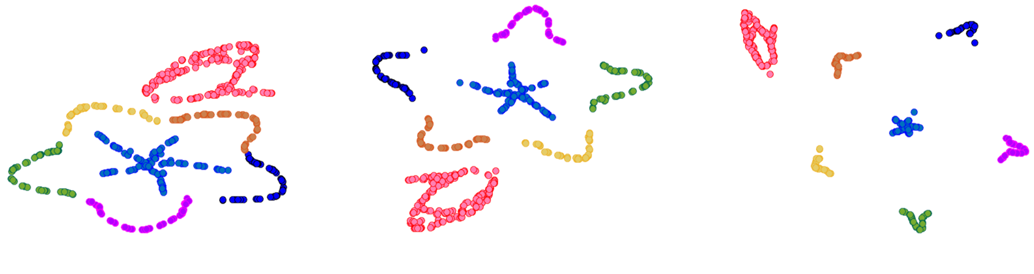


Figure 5.15: Embedding results of all real-world datasets with neighbourhood size k changes from 10 - 50.

shows the trend line of the evaluation score while Figures 5.16, 5.18 shows the embedding results. Figure 5.16 shows how λ changes the separation between cohorts in synthetic flower dataset. However, such changes are not very obvious in real-world data visualisation images. From Figure 5.18, we can barely notice that some cohorts are more compact and the separation between cohorts has been enhanced, as the λ increases from 0.1 to 0.9. Furthermore, the P_s in Figure 5.17 also shows that λ would not significantly change the cohort separation score.

Thus, λ is a mild separation control for real-world dataset, however, it performs well on low-dimensional dataset. Thus, the value of λ could be set any $\lambda < 0.5$ as an easier choice for Algorithm application.

Finally, we focus on the global vs local control r. We compare different r from 0.1, 0.05, to 0.01 with the result $\hat{\mathbf{X}}^{(0)}$, which is the optimisation result of the global cost

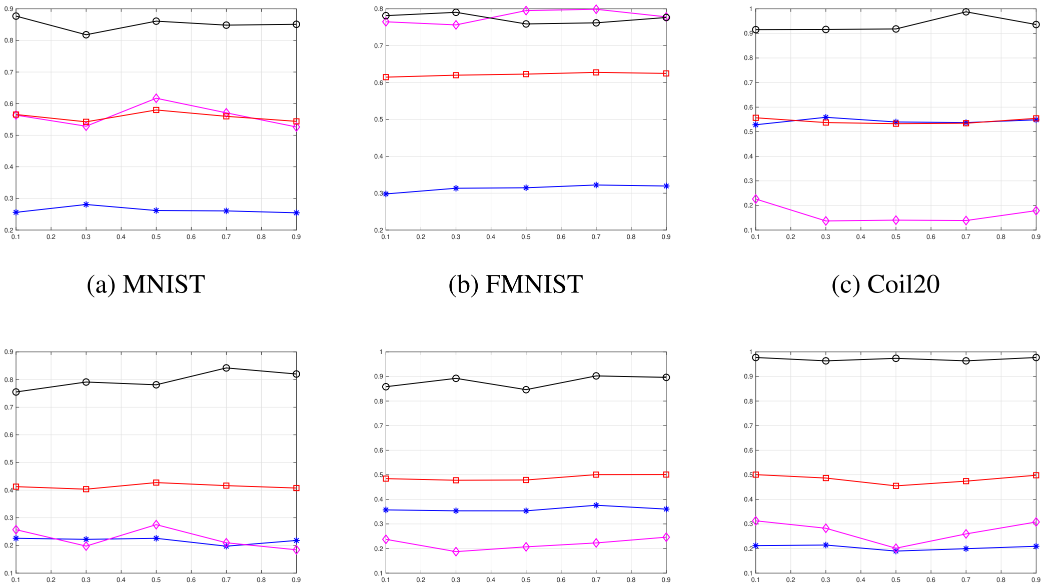


(a) Flower, $\lambda = 0$

(b) $\lambda = 0.5$

(c) $\lambda = 0.9$

Figure 5.16: Embedding results of flower dataset with λ changes from 0 - 0.9.



(a) MNIST

(b) FMNIST

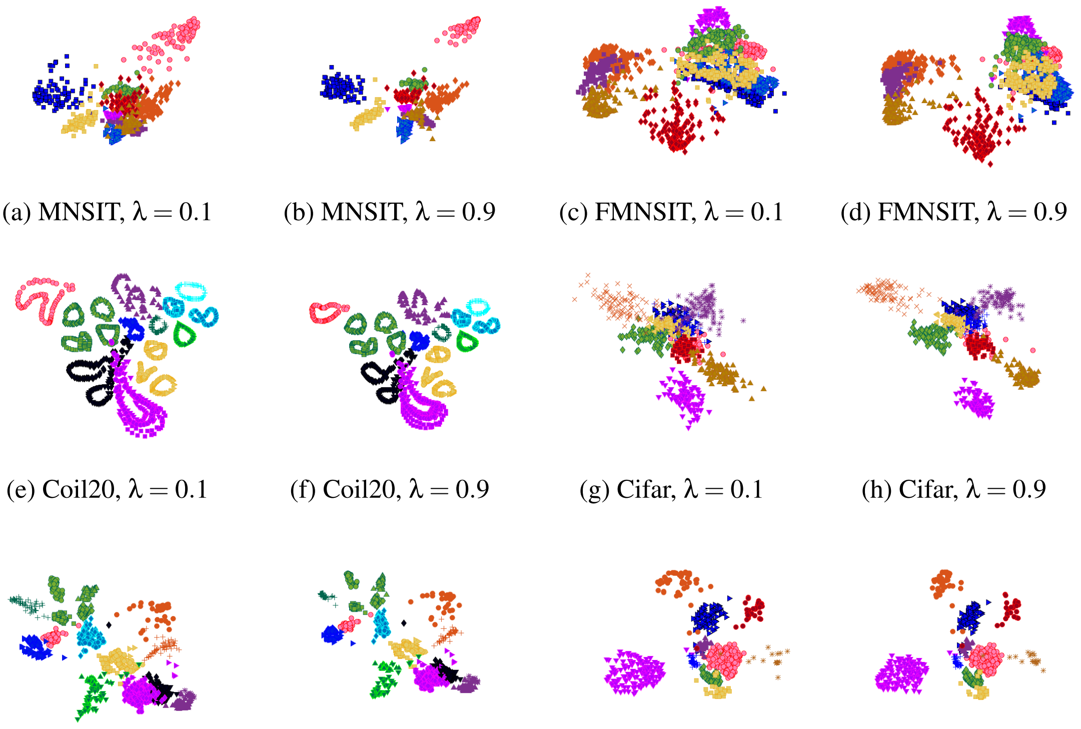
(c) Coil20

(d) Cifar

(e) Isolet

(f) Reuters

Figure 5.17: Preservation scores of ANGEL embedding results of the different datasets applying different $\lambda = 0.1, 0.3, 0.5, 0.7, 0.9$. The x axis of each sub-figure denotes different λ value, the y axis of each sub-figure denotes the preservation value. Different colours refer to different preservation score respectively: blue: P_l, pink: P_g, black: P_s, red: P.



(a) MNSIT, $\lambda = 0.1$ (b) MNSIT, $\lambda = 0.9$ (c) FMNSIT, $\lambda = 0.1$ (d) FMNSIT, $\lambda = 0.9$

(e) Coil20, $\lambda = 0.1$ (f) Coil20, $\lambda = 0.9$ (g) Cifar, $\lambda = 0.1$ (h) Cifar, $\lambda = 0.9$

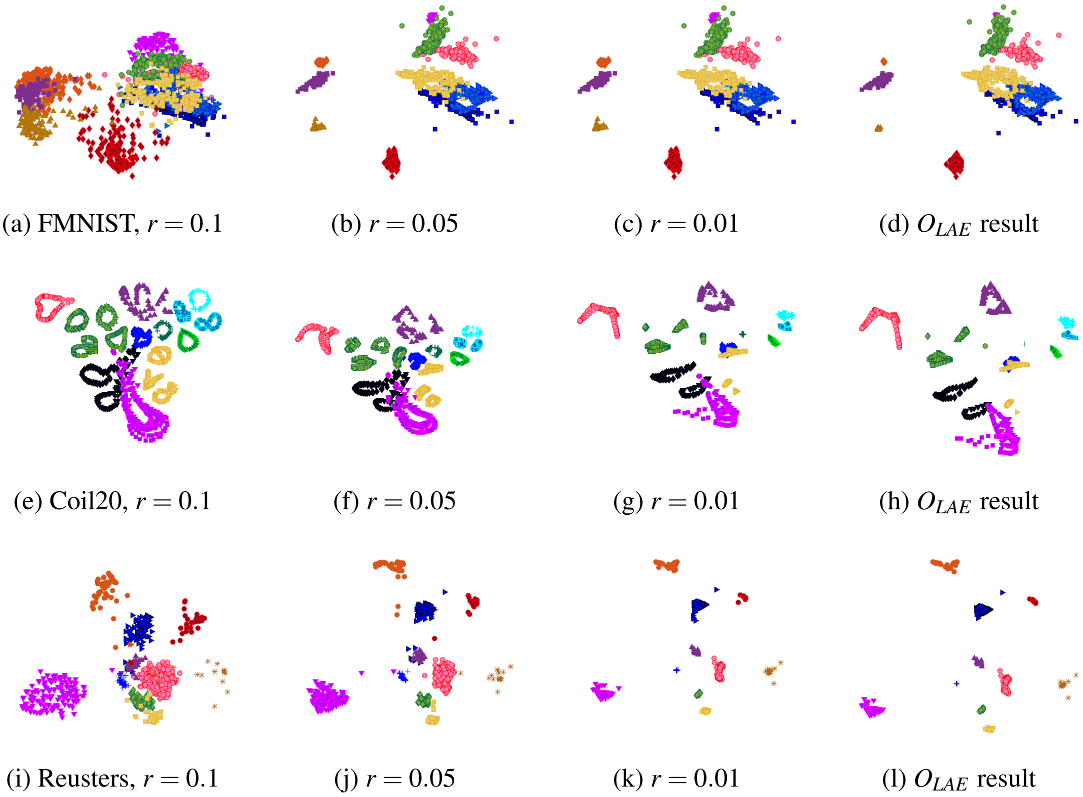
(i) Isolet, $\lambda = 0.1$ (j) Isolet, $\lambda = 0.9$ (k) Reuters, $\lambda = 0.1$ (l) Reuters, $\lambda = 0.9$

Figure 5.18: Embedding results of real-world datasets under different settings of the cohort separation parameter λ.

function O_{LAE} (Eq. 4.14) in Section 4.5. $\hat{\mathbf{X}}^{(0)}$ can be treated as a special case when $r = 0$.

From Figure 5.19, as r decreases, the embedding results collapse to the $\hat{\mathbf{X}}^{(0)}$. This can also be considered as an enhanced cohort separation process, making each cohort more recognisable. P_{s} score presented in Figure 5.20 also confirms this inference. However, except for MNSIT, r with other meaningful values can achieve a higher local neighbourhood preservation score P_{l} than the score obtained by $\hat{\mathbf{X}}^{(0)}$.

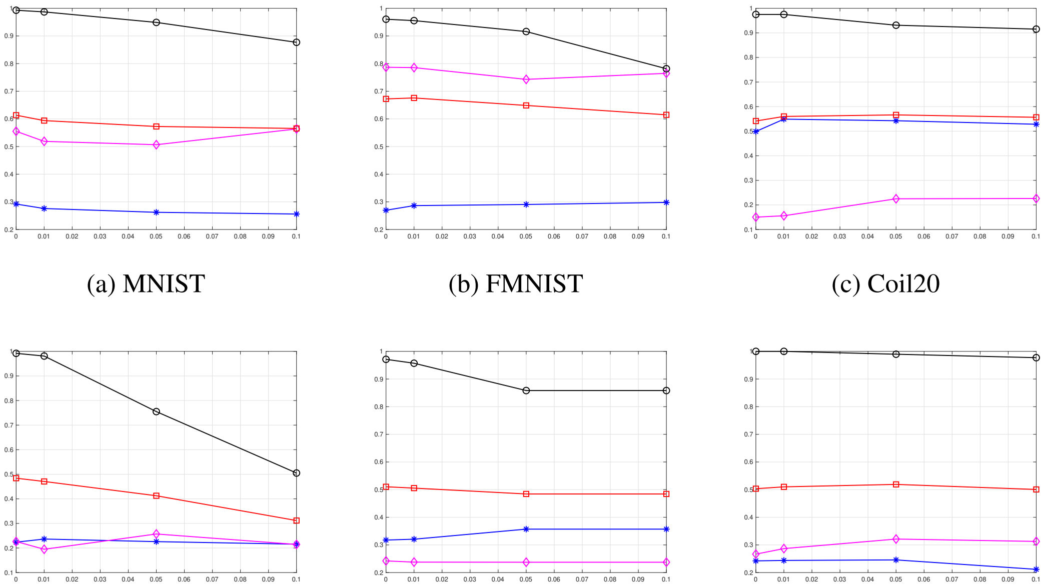
Thus, the choice of r is relatively important compared with other hyper-parameters k and λ. Users need to make trade-offs between the separation and the local neighbourhood preservation.


(a) FMNIST, $r = 0.1$ (b) $r = 0.05$ (c) $r = 0.01$ (d) O_{LAE} result

(e) Coil20, $r = 0.1$ (f) $r = 0.05$ (g) $r = 0.01$ (h) O_{LAE} result

(i) Reusters, $r = 0.1$ (j) $r = 0.05$ (k) $r = 0.01$ (l) O_{LAE} result

Figure 5.19: Embedding results of real-world datasets under different settings of the global vs local control parameter r.



(a) MNIST

(b) FMNIST

(c) Coil20

(d) Cifar

(e) Isolet

(f) Reuters

Figure 5.20: Preservation scores of ANGEL embedding results of the different datasets applying different $r = 0.1, 0.05, 0.01, 0$. The x axis of each sub-figure denotes different r value, the y axis of each sub-figure denotes the preservation value. Different colours refer to different preservation score respectively: blue: P_l, pink: P_g, black: P_s, red: P.

5.3 Chapter Summary

In this chapter, a new evaluation metric is proposed to measure embedding results and compare performance of different DR methods. Experiments are conducted to compare ANGEL with state-of-the-art algorithms. The experimental results show that ANGEL is the only method that can represent both point cohort position management and the internal structure of each cohort in $2,3$-D spaces. It is also able to achieve the highest preservation score among all DR approaches. In addition, a parametric study of the ANGEL model is presented and discussed in this chapter. It is concluded that the neighbourhood size k essentially does not affect the DR results too much. The λ has a strong separation for synthetic data points and a visual separation for high-dimensional data points. The r parameter enhances the degree of separation between data but affects local neighbourhood preservation and visual effects.

Chapter 6

ANGEL Variations

The previous chapters illustrate the proposed ANGEL model and demonstrate experimental results on how ANGEL works as a visualisation tool.

This chapter introduces the variations of the ANGEL algorithm. First three variants for ANGEL focus on algorithm acceleration. The warm-start optimisation method is based on the ε-constrained optimisation algorithm. It gives a fast approximation within only a few iterations of running time. The Loop-Replicate (LR) optimisation is developed based on the weighted sum optimisation approach. It iteratively changes the weight values in circular iterations, allowing a reasonably good approximation to be obtained in a relatively short period of time. The stochastic ordinal constraints selection is adapted to reduce the number of ordinal constraints to reduce the time consumption of the local ordinal embedding. However, it does not introduce severely negatively impact the local neighbourhood preservation.

It should be highlighted that none of these three optimisation methods proposed for ANGEL algorithm are currently theoretically supported, and only experimental results are available to prove that they are effective and accessible.

The last extension shows the feasibility of applying the ANGEL framework to a popular metric embedding algorithm such as t-SNE, where ANGEL construction is more generalised for different applications.

Algorithm 4 Warm-start optimisation for ANGEL Step 8

```
Input: Reconstruction weights $\{\mathbf{w}_i^*\}_{i=1}^n$; anchor embeddings $\{\hat{\mathbf{u}}_i\}_{i=1}^m$.
Hyperparameter: Small iteration number $N_{\text{iter}} = 10$.
Compute warm-start data embeddings $\left\{\hat{\mathbf{x}}_i^{(0)}\right\}_{i=1}^n$ by Eq. 6.1.
Construct data ordinal triplets $\Gamma^{\text{LOE}}(\mathbf{X})$ following Eq. 3.12, and use $\Gamma^{\text{LOE}}(\mathbf{X})$ to compute data embeddings $\{\hat{\mathbf{x}}_i\}_{i=1}^n$ by Eq. 4.5, where gradient descent updates are conducted for $N_{\text{iter}}$ iterations using $\left\{\hat{\mathbf{x}}_i^{(0)}\right\}_{i=1}^n$ as the initialisation.
Record results obtained from each iteration, and set the one with best $P$ as the embedding result.
Output: Embedded data points $\{\hat{\mathbf{x}}_i\}_{i=1}^n$ in 2 (or 3)-D space.
```

6.1 Different Optimisations for ANGEL

6.1.1 The Warm-start (Warm) Optimisation Method

Given large-scale dataset, solving Eq. 4.16 can be costly. We propose an alternative way to find a quick and fairly good solution to Eq. 4.16. Taking the LAE reconstruction weights $\{\mathbf{w}_i^*\}_{i=1}^n$ and the computed anchor embedding $\{\hat{\mathbf{u}}_i\}_{i=1}^m$ as the input, we approximate the data embeddings by directly solve the global objective function O_{LAE} (Eq. 4.14) and obtain

$$\hat{\mathbf{x}}_i^{(0)} = \sum_{i=1}^{m} w_i^* \hat{\mathbf{u}}_i. \tag{6.1}$$

From Figures 5.20 and 5.19, it can be observed that $\left\{\hat{\mathbf{x}}_i^{(0)}\right\}_{i=1}^n$ has the advantage of keeping the cohort structure of the original dataset, however, it suffers from a local neighbourhood preservation loss comparing with results obtained using ANGEL algorithm.

We then use $\left\{\hat{\mathbf{x}}_i^{(0)}\right\}_{i=1}^n$ as a warm-start initialisation of the LOE optimisation in Eq. 4.5, using the triplet set $\Gamma^{\text{LOE}}(\mathbf{X})$. Only a smaller number of iteration is performed. According to our empirical experience, less than 10 iterations for optimising Eq. 4.5 is sufficient. Such an setting of early stopping moves the data points $\left\{\hat{\mathbf{x}}_i^{(0)}\right\}_{i=1}^n$ towards more correct local neighbourhood but without distorting much the global structure carried by $\left\{\hat{\mathbf{x}}_i^{(0)}\right\}_{i=1}^n$. Pseudo-code of the proposed acceleration is provided in Algorithm 4. We denote the ANGEL model with this warm-start optimisation as ANGEL-warm, in order to avoid misunderstanding.

Algorithm 6 Loop-Replicate optimisation for ANGEL Step 8

```
Input: Initialisation $\hat{\mathbf{X}}_0$; Initial parameter $\alpha_0 = 0$ or 1; Maximum iteration $Iter$; The batch loop length $T$; Critical time point $t_\alpha$.
Set the iteration counter $t = 0$. Set $\alpha = \alpha_0$ for the Eq. 4.15
while $t < Iter$ do
    if $t == t_\alpha$ or $t == nT,\ n = 1,2,\ldots$ then
        Set $\alpha = 1 - \alpha$.
    end if
    Compute the Euclidean gradient of the Eq. 4.15 with respect to $\alpha$ .
    Do the gradient descent optimisation process for one iteration to update $\hat{\mathbf{X}}_t$.
    $\hat{\mathbf{X}}_{t+1} \leftarrow$ updated $\hat{\mathbf{X}}_t$
    $t = t + 1$
end while
Output: Embedded data points $\{\hat{\mathbf{x}}_i\}_{i=1}^n$ in 2 (or 3)-D space.
```

6.1.2 Loop-Replicate (LR) Optimisation Method

In the weighted-sum method (Eq. 4.15), if the value of the weight parameter α is set to 1, then only the global structure preservation will be considered when finding the optimal solution. Also, if the value of the weight parameter α is set to 0, then converting a single-objective problem will only rely on the local neighbour characteristics.

The LR optimisation method is proposed that alters α from 0 to 1 within each small loop of iterations repetitively. The complete pseudo-code is presented in Algorithm 6.

The initialisation α is set to be 0 or 1, corresponding to the preference of which sub-single objective problem to be optimised first. The parameter $Iter$ restricts the maximum iterations used in the optimisation process. We divided $Iter$ into many small repetitive loops; each loop has the time length T. With each loop, we set the critical time point t_α to change the value of the α. Since the value of the α is restrict to be 0 or 1, changing the value in the optimisation process will influence the gradient of the multi-objective model (Eq. 4.15). In other words, we only optimise one sub-single objective function in the first t_α iterations, then optimise the other sub-single objective function at the next $T - t_\alpha$ iterations, and repeat this process until it reaches the maximum $Iter$ iterations. Figure 6.1 presents a general framework of the LR algorithm.

From another perspective, except for the first t_α iterations, the initialisation of the next period of the optimisation process is the result of the previous optimisation process. It can be seen as we repeat the process of reaching the first and second targets. The critical time point t_α is set to balance two sub-single objectives. $Iter$ is limited

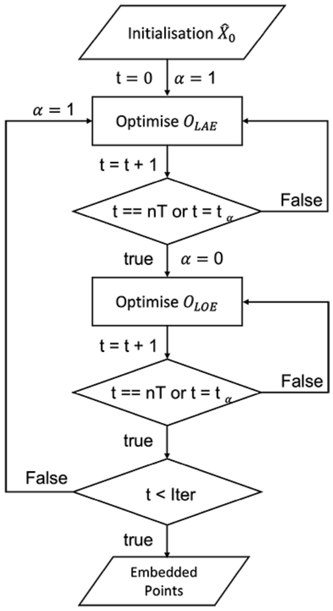


Figure 6.1: The general framework of the Loop-Replicate optimisation algorithm.

to be a non-integer value and multiplies the loop length T. The exact value of $Iter$ is computed by searching for the optimal endpoint to solve both sub-single objectives.

There is no analytical proof that this Loop-Replicate optimisation method can reach the optimal set of the multi-objective model. However, it improves the optimisation result of the weighted-sum model. The common setting is the $Iter = 1000$, $T = 100$, then set $t_\alpha = 90$ with $\alpha = 1$ at the beginning point, which optimises O_{LAE} first. We denote the ANGEL model with this LR optimisation as ANGEL-LR.

6.1.3 Stochastic Optimisation Method

The novel optimisation approaches proposed in Section 6.1 and 6.1.2 mainly focus on reducing the number of iterations to accelerate the optimisation process. In this section, we will discuss its capability to reduce the number of triplets when applied to the cost function for reducing the time consumption.

As discussed in the previous chapter, the ANGEL algorithm has a relatively high complexity between $O(n^2)$ and $O(n^3)$ concerning the data point number n. The main reason is that the size of the data ordinal constraints boosts rapidly when the number of data points increases. Take the concentric circle dataset as an example. The

Algorithm 7 Subset selection for ANGEL stochastic optimisation

```
Input: full ordinal triplet set $\Gamma^{\mathrm{LOE}}(\mathbf{X})$.
Hyperparameter: random sampling portion $par$.
Set the empty subset $\Gamma^{\mathrm{LOE}}_{sub}$
for each data point $\mathbf{x}_i$ do
    for each index of neighborhood point $j$ do
        get the set of ordinal triplets $\Gamma^{\mathrm{LOE}}_{i,j} = \{(i,j,l)\} \subset \Gamma^{\mathrm{LOE}}$ and the number of triplets in this set $n_{i,j}$.
        compute the $n_l = \lfloor n_{i,j} \times par \rfloor$ .
        randomly select $n_l$ ordinal triplets from $\Gamma^{\mathrm{LOE}}_{i,j}$ and stores them in $\Gamma^{\mathrm{LOE}}_{sub}$.
    end for
end for
Output: The subset $\Gamma^{\mathrm{LOE}}_{sub}$ used for each iteration of the optimisation process
```

dataset contains 500 data points that belongs to 2 different cohorts. Setting the k-nearest neighbourhood as $k = 10$, the total number of the ordinal constraints will be $500 \times 10 \times (500 - 1 - 10)$. For each data point $\mathbf{x}_i$, the triplet constraint (i, j, l) has only $k = 10$ choices of j, however, the l varies from $500 - 11 = 489$ different indices. If we use another flower dataset that contains 1000 data points belonging to 7 cohorts, under the same $k = 10$ setting, the choice of l will increase to $1000 - 11 = 989$ selections, which overwhelms the variation of the j. With the increasing size of the datasets, the number of the choice of l increases that leads to the increase of the triplets to be fed into the optimisation equation of ANGEL algorithm.

One straightforward idea is to reduce the number of ordinal triplets used in each gradient optimisation process by finding a fast approximation of the ANGEL embedding result. Inspired by the stochastic gradient descent, where each iteration measures the gradient based on a single randomly picked data sample, a subset of the ordinal triplets is randomly selected for each iteration to accelerate the optimisation process.

Algorithm 7 is proposed as the stochastic triplet selection. This selection process ensures that for each data point $\mathbf{x}_i$, all neighbouring points have been considered in every optimisation process but points not positioned in the neighbourhood of $\mathbf{x}_i$ are randomly optimised. We denote the ANGEL model with this stochastic optimisation as s-ANGEL.

6.1.4 Experiments and Results

Dataset: We use the same datasets as in Section 5.2.1.

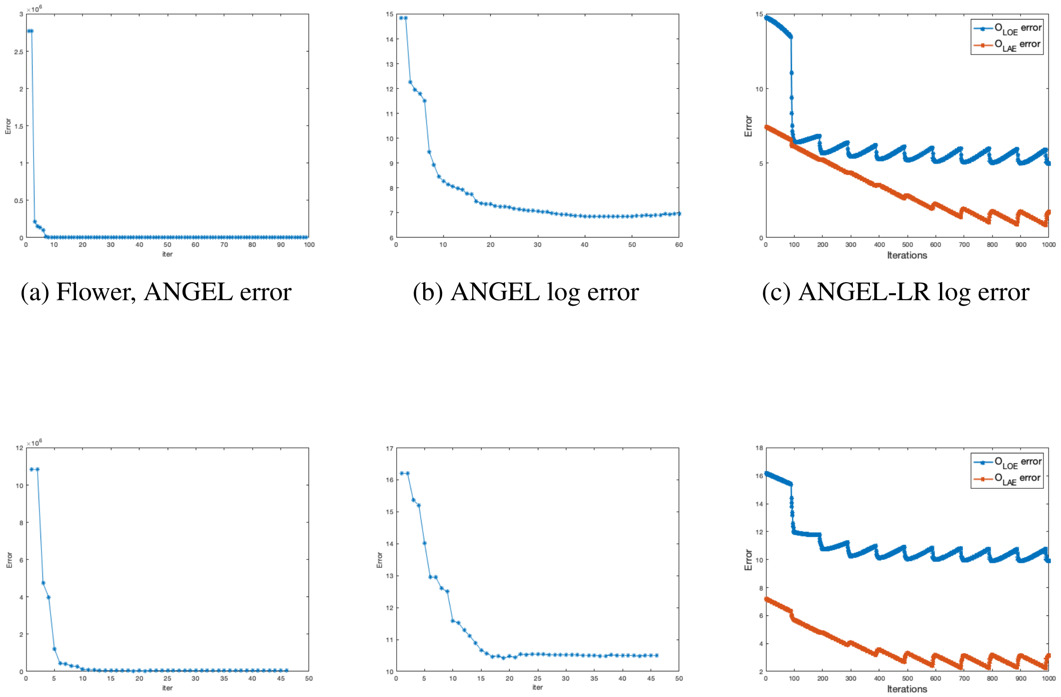
Experimental Setup:

- To make fair comparison, we adopt the same cohort position embeddings, anchor points, and anchor embeddings for the preliminary preparation. We only compare the data embedding result utilising different optimisation methods.
- As for ANGEL-warm, we set $N_{iter} = 10$, and present the best result from 10 iterations.
- As for ANGEL-LR, we set $\alpha = 0$ as the first weight, $Iter = 1000$, batch $T = 100$, and the critical time point $t_\alpha = 90$, leading to 90 times of O_{LAE} optimisation and 10 times of O_{LOE} optimisation in each batch T.
- As for s-ANGEL, we find $par < 0.5$ to accelerate the optimisation. Thus, we set par into 0.5 and 0.1 and compare embedding results.

First, we show the convergence of these algorithms in Figures 6.2 and 6.3. Since ANGEL-LR mainly applies the LOE approach in the optimisation process, its convergence is not concerned in this section. As shown in Figures 6.2a and 6.2d, the original ANGEL model converges really fast from a large error value. Thus, we apply the log error and obtained Figures 6.2b and 6.2e. Figures 6.2c and 6.2f shows the log(Error) of the LR optimisation method.

Comparing the ANGEL-LR with ANGEL, they both converge very fast during the first few iterations of the O_{LOE} optimisation process. (Note that the O_{LOE} optimisation begins at the 90-th iteration of ANGEL-LR). Then ANGEL smoothly achieves the optimal point, while ANGEL-LR oscillates with respect to optimisation of different objective functions. However, the cost error changes in a stable way. The user can stop the optimisation process at any time and obtain an approximation result. Figures 6.4 and 6.5 shows the visualisation results under different iterations of ANGEL, ANGEL-warm and ANGEL-LR.

As for the s-ANGEL, we compare the convergence under different settings of par. When $par = 1$, the original ANGEL is adopted. From Figure 6.3, the log error converges fast for all settings. s-ANGEL with $par = 0.1$ is the slowest one but still could converge after around 50 iterations. It appears the situation that the log error first becomes smaller, then larger and then smooths during the optimisation process. That is because we applied the log function and it amplifies normal oscillations of the gradient descent method. Figure 6.6 presents the image results with different par value. The



(a) Flower, ANGEL error

(b) ANGEL log error

(c) ANGEL-LR log error

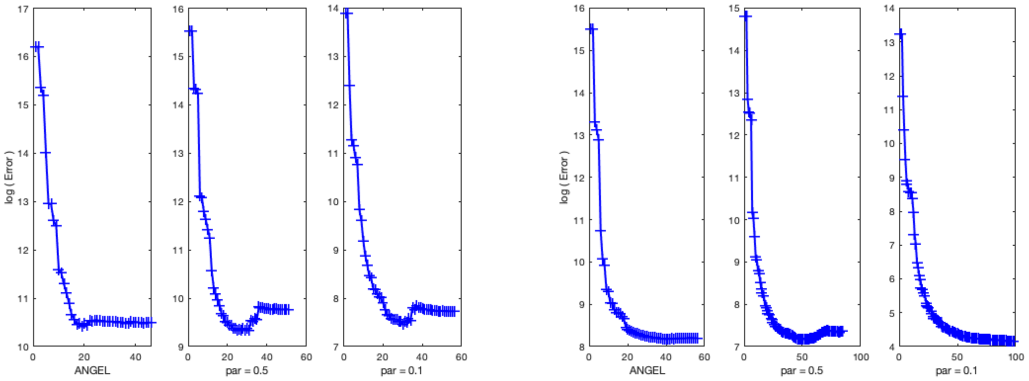
(d) FMNIST, ANGEL error

(e) ANGEL log error

(f) ANGEL-LR log error

Figure 6.2: Convergence of the optimisation process of the Flower and FMNIST datasets. The first row demonstrate Flower convergence, the second row shows convergence result of FMNIST.

result with $par = 0.1$ gives the most different result compared with the result of ANGEL as expected. However, if we wish to obtain a quick view of the ANGEL result, these images are all acceptable as an approximation.

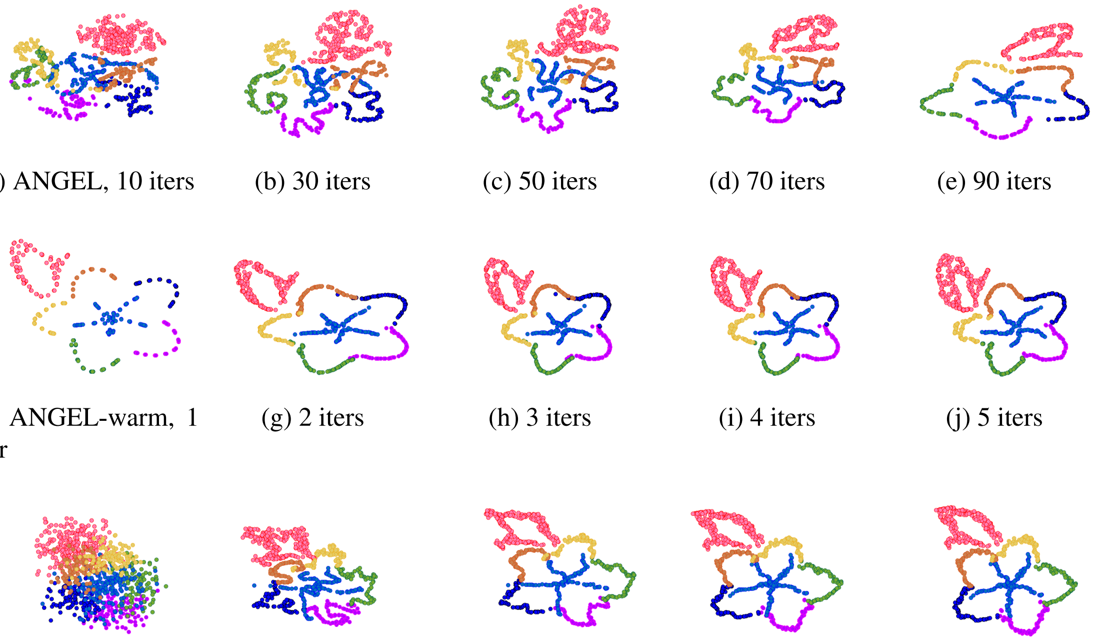


(a) FMNIST

(b) Coil20

Figure 6.3: Convergence of the optimisation process of the FMNIST and Coil20 datasets using s-ANGEL with different *par* settings. The log error is recorded.

Then we evaluate the performance of these fast-approximation results. Table 6.1 and 6.2 demonstrates preservation scores of each method applied to each dataset. Compared with the ANGEL (the columns with $par = 1$ in Table 6.2), ANGEL-warm achieves the worst performance on all three scores. Results of ANGEL-LR and s-AGNEL with $par = 0.5$ are relatively competitive, and it even achieves better results using some datasets. Taking account into the time-consumption recored in Table 6.3, ANGEL-warm and s-ANGEL with $par = 0.1$ is among the fastest ones to finish the optimisation, while those with better scores costs much more time than s-ANGEL with $par = 0.1$. However, none of the methods are as competitive as SOTA in terms of speed. Here, to guarantee consistency, the time measurement experiments are all conducted on CSF3 system supported by the University of Manchester. The CSF3 is a High Performance Computing (HPC) cluster (8,644 cores + 100 GPUs). The OS on CSF3 nodes is CentOS Linux release 7.9.2009 (Core). We adopted the standard Ivy-bridge cores: 38 nodes of 2×8-core Intel Xeon E5-2650 v2, 2.60GHz + 64GB RAM. Codes running on Matlab used the *tic* and *toc* commands, and algorithms running on Python used the *time* library.

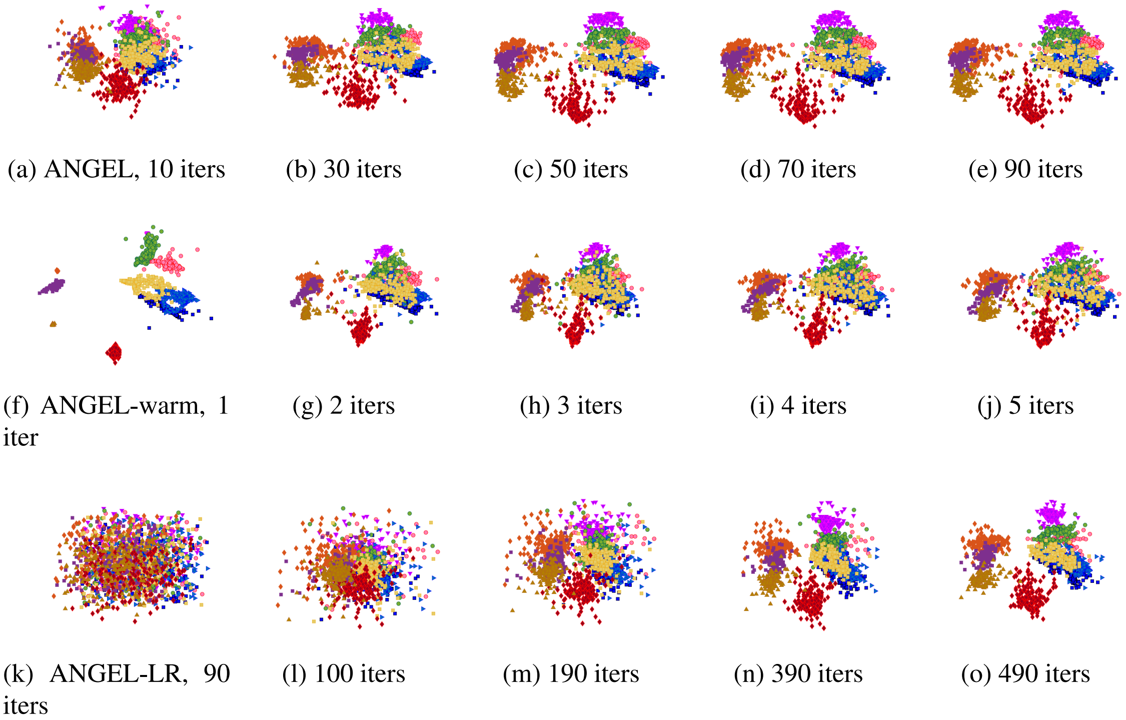


(a) ANGEL, 10 iters (b) 30 iters (c) 50 iters (d) 70 iters (e) 90 iters

(f) ANGEL-warm, 1 iter (g) 2 iters (h) 3 iters (i) 4 iters (j) 5 iters

(k) ANGEL-LR, 90 iters (l) 100 iters (m) 190 iters (n) 390 iters (o) 490 iters

Figure 6.4: Visualisations of the flower embedding during different ANGEL optimisation iterations. The first line demonstrates the ANGEL algorithm. The second line demonstrates the ANGEL warm approach. The last line shows the ANGEL-LR optimisation method. All the optimisation processes start with a random initialisation.

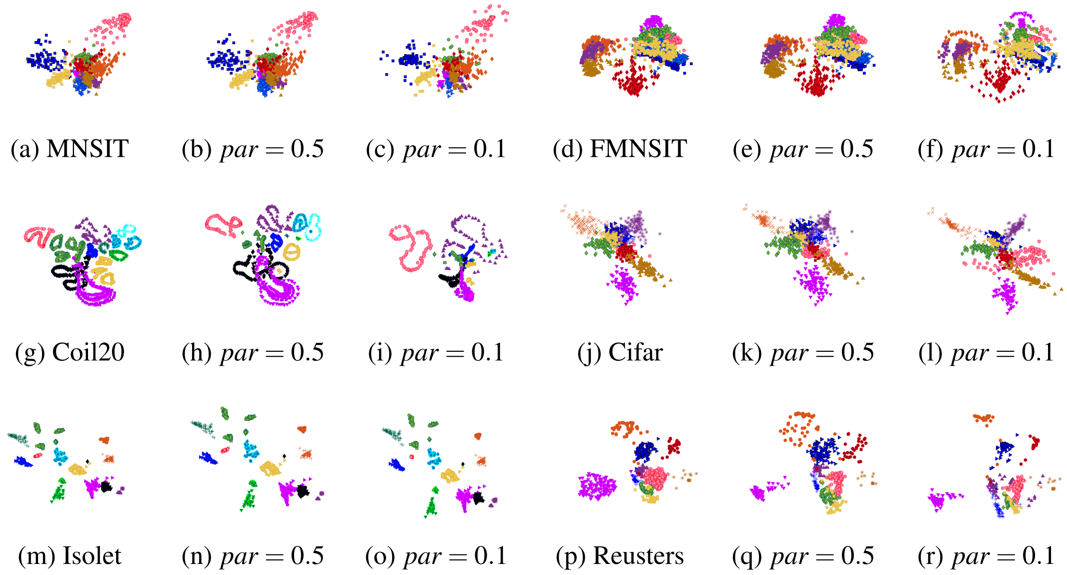


(a) ANGEL, 10 iters (b) 30 iters (c) 50 iters (d) 70 iters (e) 90 iters

(f) ANGEL-warm, 1 iter (g) 2 iters (h) 3 iters (i) 4 iters (j) 5 iters

(k) ANGEL-LR, 90 iters (l) 100 iters (m) 190 iters (n) 390 iters (o) 490 iters

Figure 6.5: Visualisations of the FMNIST embedding during different ANGEL optimisation iterations. The first row demonstrates the ANGEL algorithm. The second row demonstrates the ANGEL-warm approach. The last row shows the ANGEL-LR optimisation method. All the optimisation processes start with a random initialisation.



(a) MNSIT (b) *par* = 0.5 (c) *par* = 0.1 (d) FMNSIT (e) *par* = 0.5 (f) *par* = 0.1

(g) Coil20 (h) *par* = 0.5 (i) *par* = 0.1 (j) Cifar (k) *par* = 0.5 (l) *par* = 0.1

(m) Isolet (n) *par* = 0.5 (o) *par* = 0.1 (p) Reusters (q) *par* = 0.5 (r) *par* = 0.1

Figure 6.6: Embedding results using s-ANGEL with different *par* settings.

Method		MNIST	FMNIST	Coil20	CIFAR	Isolet	Reuters
ANGEL-warm	P_l	0.2485	0.2563	0.4628	0.1882	0.3006	0.1812
	P_g	0.5842	0.8436	0.2033	0.1273	0.2541	0.4061
	P_s	0.6380	0.6555	0.9125	0.4720	0.7397	0.8176
	P	0.4902	0.5851	0.5262	0.2625	0.4315	0.4683
ANGEL-LR	P_l	0.2682	0.2892	0.5712	0.1872	0.3292	0.3221
	P_g	0.5588	0.7600	0.1544	0.1576	0.2465	0.4206
	P_s	0.7620	0.6625	0.9167	0.4670	0.7635	0.8677
	P	0.5297	0.5705	0.5474	0.2706	0.4464	0.5368

Table 6.1: Performance difference between the ANGEL based on different optimisation approaches.

Concisely, all the fast optimisation approaches proposed in this section could successfully find an approximation result of ANGEL. Therefore, according to the performance of each algorithm, we offer suggestions based on the need of optimisation tool selection as follows:

- If the users concern the speed, ANGEL-warm and s-ANGEL ($par = 0.1$) are best choices.
- If the users concern the the balance between the speed and preservation score, s-ANGEL ($par = 0.1$) is the most suitable one.
- If the users concern the accuracy over the speed, ANGEL-LR and s-ANGEL ($par = 0.5$) are better choices. Moreover, *par* is a hyper-parameter for users to adjust to meet their own goal.

6.2 ANGEL-tSNE Model Construction

This section presents that instead of applying O_{LOE} as the local objective function, other popular local embedding method can also be adopted. Here, we introduce an ANGEL variation utilising t-SNE as the local objective function.

6.2.1 Method

We restate the t-SNE cost function (Eq. 3.5) as

		par = 1	par =0.5	par =0.1			par = 1	par =0.5	par =0.1
MNSIT	P_l	**0.2560**	0.2279	0.1803	FMNIST	P_l	**0.2979**	0.2793	0.2079
	P_g	0.5636	**0.5782**	0.5055		P_g	**0.7648**	0.7539	0.7479
	P_s	**0.8770**	0.8540	0.8550		P_s	**0.7815**	0.7770	0.7810
	P	**0.5655**	0.5534	0.5136		P	**0.6148**	0.6034	0.5778
Coil20	P_l	**0.5283**	0.5072	0.4349	Cifar	P_l	**0.2155**	0.1799	0.1621
	P_g	0.2262	0.2315	**0.2378**		P_g	**0.2145**	0.1624	0.1503
	P_s	0.9153	0.9361	**0.9528**		P_s	0.5050	0.7720	**0.8780**
	P	**0.5566**	0.5538	0.5418		P	0.3117	0.3714	**0.3968**
Isolet	P_l	**0.3571**	0.3144	0.3182	Reuters	P_l	0.2117	0.2171	**0.2301**
	P_g	**0.2374**	0.2089	0.2112		P_g	0.3127	0.3067	**0.3552**
	P_s	0.8583	**0.9647**	0.9622		P_s	**0.9772**	0.9567	0.9396
	P	0.4843	0.4960	**0.4972**		P	0.5006	0.4935	**0.5083**

Table 6.2: ANGEL performance change under different settings of the cohort separation parameter *par*. The bold number highlights the highest score obtained using different *par* settings.

Method	MNIST	FMNIST	Coil20	CIFAR	Isolet	Reuters
cat-SNE	91.4s	284.48s	154.8s	112.2s	204.1s	78.3s
s-UMAP	7.9s	16.9s	16.6s	15.2s	12.9s	10.7s
PaCMAP	2.63s	4.7s	3.1s	3.7s	4.1s	2.7s
ANGEL	339.9s	2336.4s	1525.3s	913.6s	1828.4s	846.0s
ANGEL-warm	131.7s	**389.5**s	**292.7**s	188.0s	441.8s	232.1s
ANGEL-LR	203.3s	1245.8s	328.2s	318.9s	489.8s	**134.2**s
s-ANGEL $par = 0.5$	205.3s	1368.4s	828.1s	406.0s	1355.3s	415.7s
s-ANGEL $par = 0.1$	**52.0**s	684.4s	320.5s	**131.3**s	**336.3**s	134.7s

Table 6.3: Time consumption of the ANGEL based on different optimisation approaches (only data embedding part). The bold number highlights the fastest approach among different ANGEL optimisations. The time of SOTA (cat-SNE, s-UMAP, PaCMAP) are also reported.

$$C = KL(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}},$$

where $p_{ij} = \frac{p_{j|i}+p_{i|j}}{2n}$, $p_{j|i} = \frac{\exp(-\|\mathbf{x}_i-\mathbf{x}_j\|^2/2\sigma_i^2)}{\sum_{k\neq i}\exp(-\|\mathbf{x}_i-\mathbf{x}_k\|^2/2\sigma_i^2)}$, and $q_{ij} = \frac{(1+\|\hat{\mathbf{x}}_i-\hat{\mathbf{x}}_j\|^2)^{-1}}{\sum_{k\neq l}(1+\|\hat{\mathbf{x}}_k-\hat{\mathbf{x}}_l\|)^{-1}}$.

The Barnes-Hut-SNE [119] introduces a sparse approximation of input similarities, emphasising nearest neighbours more. As the Gaussian distribution rapidly converges to zero when moving outwards from the mean, the probability $p_{j|i}$ of distant dissimilar data points $\mathbf{x}_i$ and $\mathbf{x}_j$ will be almost zero. Thus, we use a sparse approximation as Eq. 6.2 to substitute the original $p_{j|i}$ without materially adversely affecting the quality of the embedding.

$$p_{j|i} = \begin{cases} \dfrac{\exp\left(-D\left(\mathbf{x}_i, \mathbf{x}_j\right)^2 / 2\sigma_i^2\right)}{\sum_{\mathbf{x}_l \in k\text{-NN}(\mathbf{x}_i)} \exp\left(-D\left(\mathbf{x}_i, \mathbf{x}_l\right)^2 / 2\sigma_i^2\right)}, & \text{if } \mathbf{x}_j \in k\text{-NN}(\mathbf{x}_i) \\ 0, & \text{otherwise} \end{cases} \tag{6.2}$$

In [119], the value of k is set to be $\lfloor 3Perp(P_i) \rfloor$ where the perplexity $Perp(P_i)$ is defined as Eq. 3.3. Since all σ_i is derived from the user-given $Perp(P_i)$ by a simple binary search, we can easily convert the σ_i from the nearest neighbour number k directly. Thus, it is more straightforward for us to compare the sparse approximation of t-SNE with other k-nearest neighbour based embedding algorithms under the same k selection.

Similar to the construction of ANGEL, for ANGEL-tSNE, we set the same global objective function O_{LAE} as Eq. 4.14, whereas change the local objective function to the cost function of the sparse-tSNE. That is, we can implement the r-constrained model of the ANGEL-tSNE as

$$\begin{aligned} \min_{\{\hat{\mathbf{x}}_i\}_{i=1}^n} \quad & \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}, \\ \text{subject to} \quad & \left\|\hat{\mathbf{x}}_i - \hat{\mathbf{U}}^T \mathbf{w}_i^*\right\|_2^2 < r, i = 1, 2, \ldots, n, \end{aligned} \tag{6.3}$$

where $p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}$, $pj|i$ is computed using Eq. 6.2, and $q_{ij} = \frac{(1+\|\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j\|^2)^{-1}}{\sum_{k \neq l}(1+\|\hat{\mathbf{x}}_k - \hat{\mathbf{x}}_l\|)^{-1}}$.

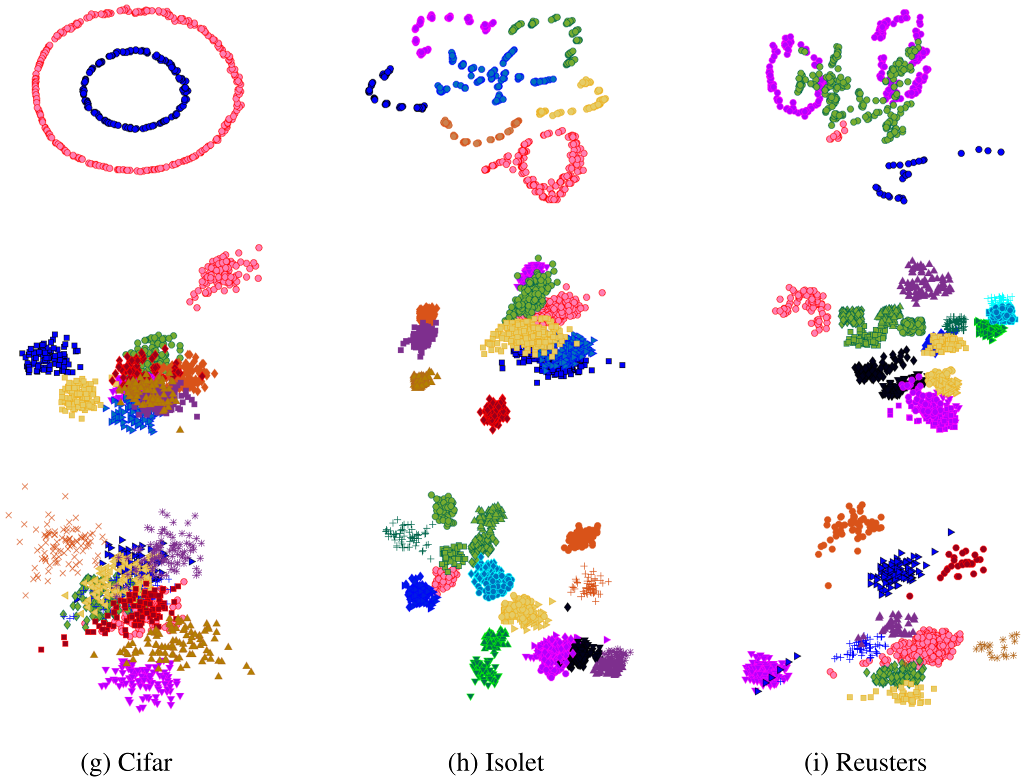
6.2.2 Experiment and Results

Dataset: We use the same data sets used in Section 5.2.1.

Experimental Setup:

- As for ANGEL-tSNE, we followed the parameter settings used in ANGEL.

Figure 6.7 shows the image embedding result of ANGEL-tSNE. Compared with results generated by ANGEL in Section 5.2, ANGEL-tSNE produces similar image results. This ensures that anchor points can well represent the overall structure of the


(g) Cifar

(h) Isolet

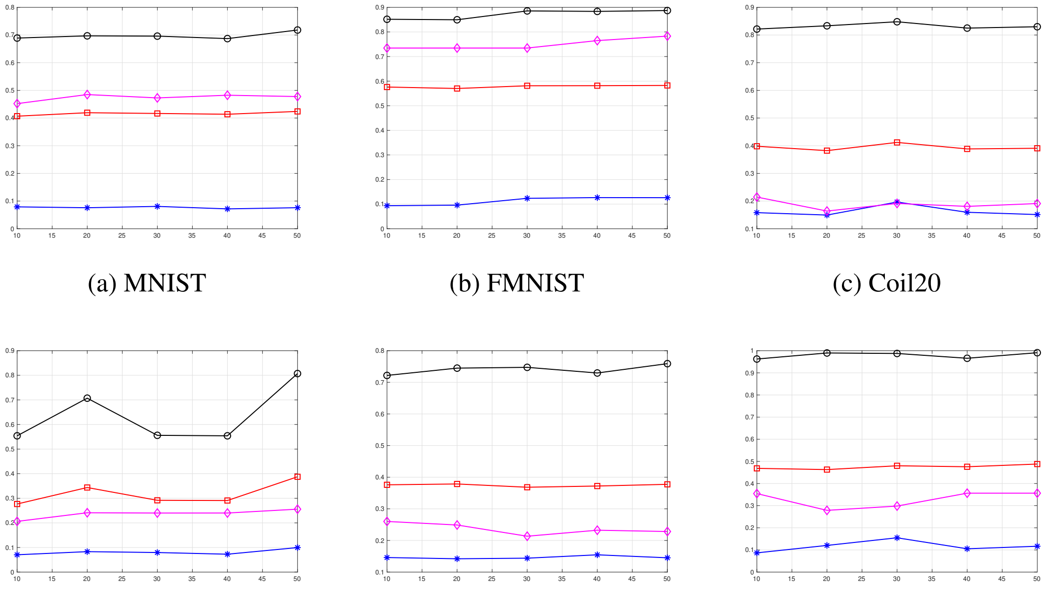
(i) Reusters

Figure 6.7: Embedding results of real-world datasets applying ANGEL-tSNE.

embedding points, while the local objective function is only utilised to modify the local neighbour preservation.

Table 6.4 compares evaluation score between ANGEL and ANGEL-tSNE. Although ANGEL-tSNE maintains the global score P_g and the separation score P_s, it suffers from the loss of the local preservation score P_l. However, it has the much more accelerated optimisation process than ANGEL has according to Table 6.5, which is also consistent with the fact that the t-SNE algorithms runs much faster than the LOE algorithm. It is obvious in Table 6.5 that AGNEL-tSNE achieves a comparable speed as t-SNE has. Here, to guarantee consistency, the time measurement experiments are all conducted on CSF3 system supported by the University of Manchester. The CSF3 is a High Performance Computing (HPC) cluster (8,644 cores + 100 GPUs). The OS on CSF3 nodes is CentOS Linux release 7.9.2009 (Core). We adopted the standard Ivybridge cores: 38 nodes of 2×8-core Intel Xeon E5-2650 v2, 2.60GHz + 64GB RAM. Codes running on Matlab used the *tic* and *toc* commands, and algorithms running on Python used the *time* library.

Finally, we will investigate the neighbourhood size k applied to the ANGEL-tSNE



(a) MNIST

(b) FMNIST

(c) Coil20

(d) Cifar

(e) Isolet

(f) Reuters

Figure 6.8: Preservation scores of ANGEL-tSNE embedding results of the different datasets applying different $k = 10, 20, 30, 40, 50$. The x axis of each sub-figure denotes different k value, the y axis of each sub-figure denotes the preservation value. Different colours refer to different preservation score respectively: blue: P_l, pink: P_g, black: P_s, red: P.

		ANGEL	ANGEL-tSNE			ANGEL	ANGEL-tSNE
MNSIT	P_{l}	**0.2560**	0.1463	FMNIST	P_{l}	**0.2979**	0.0935
	P_{g}	**0.5636**	0.4509		P_{g}	0.7648	**0.7830**
	P_{s}	0.8770	**0.9640**		P_{s}	0.7815	**0.8515**
	P	**0.5655**	0.5204		P	**0.6148**	0.5760
Coil20	P_{l}	**0.5283**	0.1581	Cifar	P_{l}	**0.2259**	0.1184
	P_{g}	**0.2262**	0.2144		P_{g}	**0.2570**	0.2315
	P_{s}	**0.9153**	0.8215		P_{s}	0.7550	**0.8220**
	P	**0.5566**	0.3980		P	**0.4216**	0.3906
Isolet	P_{l}	**0.3571**	0.1460	Reuters	P_{l}	**0.2117**	0.0868
	P_{g}	0.2374	**0.3601**		P_{g}	0.3127	**0.3546**
	P_{s}	**0.8583**	0.7218		P_{s}	**0.9772**	0.9624
	P	**0.4843**	0.3760		P	**0.5006**	0.4685

Table 6.4: Compare ANGEL with ANGEL-tSNE. The bold number reports the higher score compared between ANGEL and ANGEL-tSNE.

Method	MNIST	FMNIST	Coil20	CIFAR	Isolet	Reuters
t-SNE	7.3s	10.7s	6.3s	7.4s	7.8s	5.9s
ANGEL	339.9s	2336.4s	1525.3s	913.6s	1828.4s	846.0s
ANGEL-tSNE	13.0s	43.5s	23.4s	13.5s	32.5s	11.5s

Table 6.5: Time consumption of the t-SNE, ANGEL, and ANGEL-tSNE. As for ANGEL and ANGEL-tSNE, only data embedding part has been reported.

algorithm. Similarly, we change k from 10 to 50 and construct the t-SNE algorithm based on Eq. 6.2. Figure 6.8 shows the trend of the preservation scores of different datasets under different k settings. When k increases, P_{l} increases for some dataset but the change is very mild. The overall score is not affected by the change of the neighbourhood size, which proves that our construction of the algorithm can reduce the sensitivity of the local metric algorithm as t-SNE.

6.3 Chapter Summary

This chapter describes the variations of the ANGEL model. The chapter is organised into two parts. The first part focuses on applying different newly proposed optimisation methods to the ANGEL algorithm to obtain fast approximation results. Experimental results show the advantages of these optimisation methods in terms of algorithm speed-up and give recommendations on the choice of parameters for different scenarios. The second section illustrates the feasibility of applying the ANGEL framework to a popular embedding algorithm. The previously used local objective function is replaced by another local embedding cost function, namely ANGEL-t-SNE. Experimental results show that replacing the local objective function does not affect the global results but only for the neighbourhood preservation case.

Chapter 7

Incremental ANGEL

The vast majority of the DR visualisation methods operate in a batch mode, which means that they cannot process new coming data points [76]. When new data samples arrive sequentially, to obtain a new embedding result, most of the approaches such as LOE, t-SNE and UMAP, need to repeat running of the "batch" version on the "new" dataset, which is updated by adding the new data sample. This process is time-consuming and wastefully discards the pre-generated embedding results [42, 79].

The incremental DR methods are developed to handle the streaming data [76], with an additional ability to visualise the gradual change of the embedding result [79]. There are plenty of incremental methods developed based on metric embedding approaches as reviewed in Section 2.4. However, it is worth mentioning that there is no incremental extensions on ordinal embedding approaches.

We proposed two extensions of ANGEL targeting at dealing with new coming data samples. The first approach progressive ANGEL (p-ANGEL) (Section 7.1) aims at memory reduction, which treats portions of datasets as the new coming samples and embedded these "new" points progressively. Another incremental ANGEL (i-ANGEL) approach (Section 7.2) can process the real new data points, however, when applying this extension to the existing dataset, it aims at speeding up the acceleration. Experimental results shows their performance on 6 real-world datasets and compared ANGEL with these two approaches.

7.1 Memory Reduction: p-ANGEL

One of the limitations of ANGEL is the memory consumption, since it requires to compute the overall distance matrix among all data samples and embedded points while processing the data embedding. When coming across a dataset containing n data samples, an essential step is to construct an $n \times n$ matrix to store the distance between data samples. However, when n is larger than 20k, it is difficult for a normal computational node to handle the large matrix.

Inspired by progressive DR algorithms [42, 70, 76, 87], the progressive ANGEL (p-ANGEL) is proposed to reduce the size of the distance matrix. The idea is to separate the existing dataset into a batch dataset and a progressive dataset, and embedded data samples in the progressive dataset one by one as the new coming data sample. Thus, the distances between the "new" data sample and the batch dataset are computed to obtain the updated k-nearest neighbourhood. For each embedding process, the updated points along with "new" coming data sample will be optimised, which ideally reduce the memory cost compared with ANGEL. To be more specific, in Algorithm 8, the distance matrix calculation is replaced by line 8 where the maximum distance matrix has the size $(n-1) \times 1$. The trade-off is between the time and memory since we do the embedding of each data point $\mathbf{x}_i$ one by one in a loop instead of finding the embedding of the whole dataset.

The algorithm p-ANGEL starts with the same cohort embedding, anchor generation and embedding as the ANGEL model. Then it updates the ordinal triplets based on the updated k-nearest neighbourhood, as shown in Section 7.1.1. The data embedding process is based on the updated ordinal triplets to obtain the embedded points in $2,3$-D spaces. The full algorithm is stated in Section 7.1.2

7.1.1 Updating Ordinal Triplet Constraints

The construction of ordinal triplet constraints of data points is a critical step for the ANGEL algorithm, because it keeps the local neighbourhood information of each data point while doing the embedding process. When a new data point $\mathbf{x}_{n+1}$ arrives, it only influence directly the ordinal triplet constraints of batch data points that includes $\mathbf{x}_{n+1}$ in their k-nearest neighbours or belongs to the k-nearest neighbours of the new data point $\mathbf{x}_{n+1}$. We can update the set $\Gamma^{LOE}(\mathbf{X}_n)$ by calculating k-nearest neighbourhood among the new dataset $\mathbf{X}_{n+1}$ that holds the batch dataset $\mathbf{X}_n$ with the data point $\mathbf{x}_{n+1}$ as

$$\Gamma^{LOE}(\mathbf{X}_{n+1}) = \left\{(i,j,l)|\mathbf{x}_j \in k\text{-NN}(\mathbf{x}_i), \mathbf{x}_l \notin k\text{-NN}(\mathbf{x}_i)\right\}, \tag{7.1}$$

here $\mathbf{X}_{n+1} = [\mathbf{X}_n; \mathbf{x}_{n+1}]$. We use $\mathbf{X}_A$ where each $\mathbf{x}_i, i \in A$ whose ordinal triplet constraints are affected, and we denote $\mathbf{X}_{An} = [\mathbf{X}_A; \mathbf{x}_{n+1}]$ includes the new data points into the $\mathbf{X}_A$.

Unlike progressive metric embedding algorithms, we do not require updating the exact edges of neighbourhood graph as [42, 70] do. However, we only need to replace the data point that is not in the updated neighbourhood with the new coming data point $\mathbf{x}_{n+1}$. That is, taking $\mathbf{x}_{n+1}$ among the new k-nearest neighbour of the set $\mathbf{X}_A$, the dropped ordinal triplet constraint set is

$$\Gamma^{LOE-P}(\mathbf{x}_{n+1}) = \left\{(i,j,k)|\mathbf{x}_i \in \mathbf{X}_A, D(\mathbf{x}_i,\mathbf{x}_j) > D(\mathbf{x}_i,\mathbf{x}_{n+1}), \mathbf{x}_l \notin k\text{-NN}(\mathbf{x}_i)\right\}, \tag{7.2}$$

where the $\Gamma^{LOE-P} \subset \Gamma^{LOE}$, and the updated ordinal triplet constraint set is

$$\Gamma^{LOE-U}(\mathbf{x}_{n+1}) = \left\{(i,j,k)|\mathbf{x}_i \in \mathbf{X}_A, \mathbf{x}_j = \mathbf{x}_{n+1}, \mathbf{x}_l \notin k\text{-NN}(\mathbf{x}_i)\right\}. \tag{7.3}$$

Furthermore, the new generated ordinal constraint set according to the new data point $\mathbf{x}^*$ is

$$\Gamma^{LOE-N}(\mathbf{x}_{n+1}) = \left\{(i,j,k)|\mathbf{x}_i = \mathbf{x}_{n+1}, \mathbf{x}_j \in k\text{-NN}(\mathbf{x}_{n+1}), \mathbf{x}_l \notin k\text{-NN}(\mathbf{x}_{n+1})\right\}. \tag{7.4}$$

Then the full updated ordinal constraint set regarding to the $\mathbf{x}_{n+1}$ is given as

$$\Gamma^{LOE*}(\mathbf{x}_{n+1}) = \Gamma^{LOE-U}(\mathbf{x}_{n+1}) \cup \Gamma^{LOE-N}(\mathbf{x}_{n+1}). \tag{7.5}$$

The whole triplet set $\Gamma^{LOE}(\mathbf{X}_{n+1})$ can also be denoted by

$$\Gamma^{LOE}(\mathbf{X}_{n+1}) = \left(\Gamma^{LOE}(\mathbf{X}_n) \setminus \Gamma^{LOE-P}(\mathbf{x}_{n+1})\right) \cup \Gamma^{LOE*}(\mathbf{x}_{n+1}). \tag{7.6}$$

We will emphasize the $\Gamma^{LOE*}(\mathbf{x}_{n+1})$ in the following optimisation algorithm. The simple and straightforward computation leads to a memory reduction, as there is no need to find the updated triplets by recalculating the k-NN graph among all data samples using Eq. 7.1.

7.1.2 p-ANGEL Algorithm Implementation

The pseudo-code of the modified algorithm is shown in Algorithm 8. Similar to the constrained optimisation problem Eq. 4.16 of ANGEL algorithm, we can construct

$$\begin{aligned} &\min_{\{\hat{\mathbf{x}}_i\}, i\in An} \sum_{(i,j,l)\in\Gamma^{\mathrm{LOE}*}(\mathbf{x}_{n+1})} \max^2\left(0, D(\hat{\mathbf{x}}_i,\hat{\mathbf{x}}_j)+\delta-D(\hat{\mathbf{x}}_i,\hat{\mathbf{x}}_l)\right), \\ &\text{subject to} \quad \left\|\hat{\mathbf{x}}_i-\hat{\mathbf{U}}^T\mathbf{w}_i^*\right\|_2^2 < r,\ i\in An. \end{aligned} \tag{7.7}$$

A suggested setting is $\delta = 0.1$ and $r \approx 0.05$, which follows the suggested batch embedding setting. The final optimum embedding result $\hat{\mathbf{X}}_{n+1}$ is obtained by updating $\hat{\mathbf{X}}_A \subset \hat{\mathbf{X}}_n$ and including the new embedding $\hat{\mathbf{x}}_{n+1}$.

Comparing with the original ANGEL algorithm, this p-ANGEL algorithm reduces memory consumption from $n \times n$ to $(n-1) \times 1$, but suffers from the local neighbourhood preservation loss. It is hard to practically qualify and evaluate the usage memory cost in MATLAB under Linux or Mac OS. However, this will be an interesting future direction to be discussed in our potential works.

7.2 Speed Acceleration: i-ANGEL

As for incremental ANGEL (i-ANGEL), we start with the real batch dataset $\{\mathbf{x}_i\}_{i=1}^n$ and their cohort labels $\{y_i\}_{i=1}^n$. The first step is to embed these high-dimensional data points to the target $2,3$-D space applying ANGEL algorithm. The embedding of the new data sample will depends on embedded anchor points and data points of the batch dataset.

Following the the progressive incremental approaches discussed in Section 2.4.2, the incremental methods also begin with updating of the k-nearest neighbour graph, as proposed in Section 7.1.1. Then the optimisation function can be constructed based on to optimise the position of the new point and the updated points. Since we applied the

Algorithm 8 p-ANGEL Algorithm

```
**Input**: $D$-dimensional data samples $\{\mathbf{x}_i\}_{i=1}^n$ and their cohort labels $\{y_i\}_{i=1}^n$;
**User-adjust hyper-parameters**: Reduced dimension $d$ (2 or 3), cohort separation control $0 \leq \lambda \leq 1$; local neighbourhood size $k$; global vs local control $r \approx 0.05$; batch size $b$;
**Fixed hyper-parameters**: anchor point density $p = 0.1$; reconstruction anchor number $t = 3$; neighbour margin $\delta = 0.1$; stochastic triplet selection $par = 0.5$.
Step 4 - 7 of the ANGEL Algorithm 3
Sample a batch data set $\mathbf{X}_b = \{\mathbf{x}_i\}_{i=1}^b \subset \{\mathbf{x}_i\}_{i=1}^n$ with their cohort labels $\{y_i\}_{i=1}^b \subset \{y_i\}_{i=1}^n$, then we denote the rest of the data set as $\mathbf{X}_{n-b} = \{\mathbf{x}_i\}_{i=1}^{n-b}$ with $\{y_i\}_{i=1}^{n-b}$
Construct data ordinal triplets $\Gamma^{\mathrm{LOE}}(\mathbf{X}_b)$, and compute data embeddings $\{\hat{\mathbf{x}}_i\}_{i=1}^b$ by optimising Eq. 4.16.
**for** Each data point $\mathbf{x}_i \in \{\mathbf{x}_i\}_{i=1}^{n-b}$ **do**
    Compute distance between new data sample $\mathbf{x}_i$ and batch dataset $\mathbf{X}_b$.
    Obtain the set of affected data points $\mathbf{X}_A$ and the full updated ordinal triplet set $\Gamma^{LOE*}(\mathbf{x}_{b+1})$ by Eq. 7.5.
    Initialise $\hat{\mathbf{x}}_{b+1} \leftarrow \sum_{i=1}^m w_i^* \hat{\mathbf{u}}_i$ or initialise $\hat{\mathbf{x}}_{b+1}$ as a random coordiante.
    Compute the updated embeddings $\hat{\mathbf{X}}_A$ and $\hat{\mathbf{x}}_{b+1}$ by optimise Eq. 7.7.
    Update the batch embeddings using $\hat{\mathbf{X}}_A$ and add $\hat{\mathbf{x}}_{b+1}$ to batch embeddings to obtain $\{\hat{\mathbf{x}}_i\}_{i=1}^{b+1}$.
**end for**
**Output**: Embedded data points $\{\hat{\mathbf{x}}_i\}_{i=1}^n$ in 2 (or 3)-D space.
```

ordinal embedding method as the local objective function, we need to update the set of triplets, and then update the anchor-point reconstruction weights, and finally embed the data samples. We will discuss the detailed steps in the following sections.

7.2.1 Calculating Anchor Reconstruction Weights

Assuming we fix the anchor points $\mathbf{U} \in \mathbb{R}^{m \times D}$ derived from the batch dataset $\mathbf{X}_n$, the critical step of our proposed i-ANGEL algorithm is to derive the anchor reconstruction weights of the new coming data $\mathbf{x}_{n+1}$. Since anchor points sketch the global structure of the dataset, anchor reconstruction weights of the $\mathbf{x}_{n+1}$ ensures the embedded $\hat{\mathbf{x}}_{n+1}$ to allocate around the reasonable position.

If $\mathbf{x}_{n+1}$ comes with the corresponding cohort label y_{n+1}, then anchor reconstruction weights can be directly calculated using Eq. 3.14 based on anchor points belonging to that cohort. However, suppose we only know the coordinates of the new data point $\mathbf{x}_{n+1}$. In that case, we need to firstly obtain a cohort label to $\mathbf{x}_{n+1}$, then derive the anchor reconstruction weights simultaneously.

The question is, how to obtain the cohort label of the new data point $\mathbf{x}_{n+1}$. Since we

Algorithm 9 i-ANGEL Algorithm

```
**Input**: $D$-dimensional batch data points $\{\mathbf{x}_i\}_{i=1}^n$ and their cohort labels $\{y_i\}_{i=1}^n$; $D$-dimensional new data point $\mathbf{x}_{n+1}$;
**User-adjust hyper-parameters**: Reduced dimension $d$ (2 or 3), cohort separation control $0 \leq \lambda \leq 1$; local neighbourhood size $k$; global vs local control $r \approx 0.05$.
**Fixed hyper-parameters**: anchor point density $p = 0.1$; reconstruction anchor number $t = 3$; neighbour margin $\delta = 0.1$; stochastic triplet selection $par = 0.5$.
Embed batch data points using Algorithm 3 and obtained the batch embeddings $\{\hat{\mathbf{x}}_i\}_{i=1}^n$, $D$-dimensional anchor points $\{\mathbf{u}_i\}_{i=1}^m$, $d$-dimensional anchor embeddings $\{\hat{\mathbf{u}}_i\}_{i=1}^m$, the corresponding cohort membership of anchor points, anchor reconstruction weights of each data point $\{\mathbf{w}_i^*\}_{i=1}^n$.
Compute distance between new data sample $\mathbf{x}_{n+1}$ and batch dataset $\{\mathbf{x}_i\}_{i=1}^n$.
Obtain the set of affected data points $\mathbf{X}_A$ and the full updated ordinal triplet set $\Gamma^{LOE*}(\mathbf{x}_{n+1})$ by Eq. 7.5
**if** $\nexists$ $y_{n+1}$ **then**
    Compute the $y_{n+1}$ using $k$-NN classifier.
**end if**
Compute the new anchor reconstruction weight $\mathbf{w}_{n+1}^*$ by Eq. 4.6 for the new data point $\mathbf{x}_{n+1}$
Initialise $\hat{\mathbf{x}}_{n+1} \leftarrow \sum_{i=1}^m w_i^* \hat{\mathbf{u}}_i$.
Compute the updated embeddings $\hat{\mathbf{X}}_A$ and $\hat{\mathbf{x}}_{n+1}$ by optimise Eq. 7.7.
Update the batch embeddings using $\hat{\mathbf{X}}_A$ and add $\hat{\mathbf{x}}_{n+1}$ to batch embeddings $\{\hat{\mathbf{x}}_i\}_{i=1}^n$
**Output**: Embedded data points $\{\hat{\mathbf{x}}_i\}_{i=1}^{n+1}$ in 2 (or 3)-D space.
```

have already gathered k-nearest neighbours of $\mathbf{x}_{n+1}$ from the previous section, we can apply the k-NN classification method [67] directly to retrieve the cohort membership of $\mathbf{x}_{n+1}$. To make it simple, we use k-NN classifier in our algorithm to obtain the y_{n+1}, where k is exactly the number of the k-nearest neighbours used to generate ordinal constraint triplets. Then the Eq. 3.14 is utilised to derive the new anchor reconstruction weights $\mathbf{w}_{n+1}^*$.

7.2.2 i-ANGEL Algorithm Implementation

Holding the updated triplet set, $\Gamma^{LOE}(\mathbf{X}_{n+1})$, and the adjusted anchor reconstruction weights, $\mathbf{w}*_{n+1}$, the most direct way to get the new data embedding result is to optimise the Eq. 4.5. However, this means we need to go through all the ordinal triplet constraints and try to optimise the embedding of all data points, which is cumbersome and time-consuming.

We propose the i-ANGEL algorithm that only utilises the updated ordinal triplet constraints $\Gamma^{LOE*}(\mathbf{x}_{n+1})$ and optimise $\hat{\mathbf{X}}_{An} = \{\hat{\mathbf{x}}_i\}, i \in An$, which contains the embedding of data points $\mathbf{X}_A$ whose neighbourhood is affected by the new data point $\mathbf{x}_{n+1}$ and the embedding point of $\mathbf{x}_{n+1}$ itself.

The pseudo-code of i-ANGEL implementation is shown in Algorithm 9. This algorithm can continuously optimise the streaming out-of-sample data points by updating the batch embedding and cohort labels after each optimisation process.

7.3 Experiments and Results

Dataset: We apply the same data sets used in Section 5.2.1.

Experimental Setup:

- The batch size of the dataset is set to $b = \lfloor 0.3n \rfloor$ as a default setting. Rest of hyper-parameters are set as the same as the ANGEL does.

- The s-ANGEL optimisation will also be applied to optimise i-ANGEL and p-ANGEL respectively as an acceleration choice. In order to compete with the s-ANGEL with $par = 0.1$, we also set $par = 0.1$ for i-ANGEL and p-ANGEL optimisation.

Figure 7.1 shows the embedding image of the batch dataset and the embedding result of the whole dataset after applying p-ANGEL or i-ANGEL. It is clear that the incremental embedding approach can successfully embed new data to the relevant cohort and obtain a reliable visualisation result.

As shown in Table 7.1 we compare the overall evaluation score between s-ANGEL, p-ANGEL, and i-ANGEL with the batch size as the default value. For all these three methods, we set the stochastic parameter $par = 0.1$ for a fast implementation. It is obvious that for most datasets, all the embedding results obtained by three algorithms share the similar P_{g} and P_{s}, while i-ANGEL has a relatively low local neighbourhood preservation score P_{l}. That is mainly because that i-ANGEL rely on 1-NN classier to determine the label of the newly coming point, which may cause conflicts between anchor-point reconstruction and local neighbourhood preservation. As p-ANGEL adopts the same anchor points and anchor-data reconstruction weights of s-ANGEL, the overall score of p-ANGEL is more close to the score of s-ANGEL.

Method		MNIST	FMNIST	Coil20	CIFAR	Isolet	Reuters
	P_l	**0.1803**	**0.2089**	**0.4349**	**0.1621**	**0.3182**	**0.2301**
s-ANGEL	P_g	0.5055	**0.7479**	**0.2378**	0.1503	0.2112	0.3552
$par = 0.1$	P_s	**0.8550**	0.7810	**0.9528**	**0.8780**	**0.9622**	**0.9396**
	P	0.5136	**0.5778**	**0.5418**	**0.3968**	**0.4972**	**0.5083**
p-ANGEL	P_l	0.1729	0.0987	0.3478	0.1420	0.2647	0.1528
$par = 0.1$	P_g	**0.5939**	0.6933	0.1891	0.2230	0.2591	**0.3915**
$b = \lfloor 0.3n \rfloor$	P_s	0.8320	0.7005	0.9319	0.6820	0.8173	0.9373
	P	**0.5329**	0.4975	0.4896	0.3490	0.4470	0.4939
i-ANGEL	P_l	0.1607	0.1336	0.2603	0.1351	0.2092	0.1251
$par = 0.1$	P_g	0.5855	0.7042	0.1355	**0.2873**	**0.2842**	0.2267
$b = \lfloor 0.3n \rfloor$	P_s	0.8460	**0.8745**	0.8847	0.7320	0.7103	0.9122
	P	0.5307	0.5708	0.4268	0.3848	0.4012	0.4213

Table 7.1: Evaluation results comparison of s-ANGEL, p-ANGEL, and i-ANGEL. The bold number highlights the highest score among all ANGEL variations.

Considering the time consumption in Table 7.2, i-ANGEL has the shortest running time since it reduces the anchor embedding running time sharply. However, if we simply pay attention to the time cost of data embedding process, both i-ANGEL and p-ANGEL require much more time than s-ANGEL, with the same $par = 0.1$ setting. Moreover, the time consumed by i-ANGEL and p-ANGEL increases as the number of data points increases. Here, the motivation of comparing p-ANGEL and i-ANGEL with s-ANGEL is to reduce the experimental time consumption. We apply the same stochastic part to both p-ANGEL and i-ANGEL as setting of the s-ANGEL to ensure the consistency. To guarantee more experimental consistency, the time measurement experiments are all conducted on CSF3 system supported by the University of Manchester. The CSF3 is a High Performance Computing (HPC) cluster (8,644 cores + 100 GPUs). The OS on CSF3 nodes is CentOS Linux release 7.9.2009 (Core). We adopted the standard Ivybridge cores: 38 nodes of 2×8-core Intel Xeon E5-2650 v2, 2.60GHz + 64GB RAM. Codes running on Matlab used the *tic* and *toc* commands, and algorithms running on Python used the *time* library.

Finally, we focus on parameter study of the p-ANGEL algorithm. Table 7.3 reports evaluation results of p-ANGEL under different settings. The results confirmed that parameter *par* only influences P_l. A smaller *par* can significantly reduce the time consumption of data embedding process, but only suffers from a minor local neighbourhood preservation loss. Reducing the batch data size to $b = \lfloor 0.1n \rfloor$ does not ensure that optimisation times can be reduced. However, it may get a better evaluation score

Method		MNIST	FMNIST	Coil20	CIFAR	Isolet	Reuters
cat-SNE		91.4s	284.48s	154.8s	112.2s	204.1s	78.3s
s-UMAP		7.9s	16.9s	16.6s	15.2s	12.9s	10.7s
PaCMAP		2.63s	4.7s	3.1s	3.7s	4.1s	2.7s
ANGEL	Anchor embedding	484.3s	4864.9s	2429.1s	922.9s	2750.9s	770.2s
	Data embedding	339.9s	2336.4s	1525.3s	913.6s	1828.4s	846.0s
	Overall time	824.2s	7201.3s	3954.4s	1836.5s	4589.3s	1616.2s
p-ANGEL $par = 1$	Anchor embedding	484.3s	4864.9s	2429.1s	922.9s	2750.9s	770.2s
	Data embedding	1362.6s	4364.2s	2459.7s	3018.2s	2549.7s	1779.0s
	Overall time	1846.9s	9229.1s	4888.8s	3941.1s	5300.6s	25492s
s-ANGEL $par = 0.1$	Anchor embedding	484.3s	4864.9s	2429.1s	922.9s	2750.9s	770.2s
	Data embedding	52.0s	684.4s	320.5s	131.3s	1355.3s	137.4s
	Overall time	536.3s	5549.3s	2749.6s	1054.2s	4106.2s	**907.6s**
p-ANGEL $par = 0.1$	Anchor embedding	484.3s	4864.9s	2429.1s	922.9s	2750.9s	652.1s
	Data embedding	284.1s	1454.5s	835.8s	1052.6s	826.3s	1058.1s
	Overall time	768.4s	6319.4s	3264.9s	1975.5s	3577.2s	1710.2s
i-ANGEL $par = 0.1$	Anchor embedding	64.5s	55.6s	298.7s	63.9s	696.5s	61.9s
	Data embedding	264.0s	1626.0s	806.1s	984.6s	608.8s	861.9s
	Overall time	**328.5s**	**1681.6s**	**1104.8s**	**1048.5s**	**1305.3s**	923.8s

Table 7.2: Optimisation time difference between the ANGEL, p-ANGEL, s-ANGEL, and i-ANGEL. The bold number highlights the fastest approach among different ANGEL variations. The time of SOTA (cat-SNE, s-UMAP, PaCMAP) are also reported.

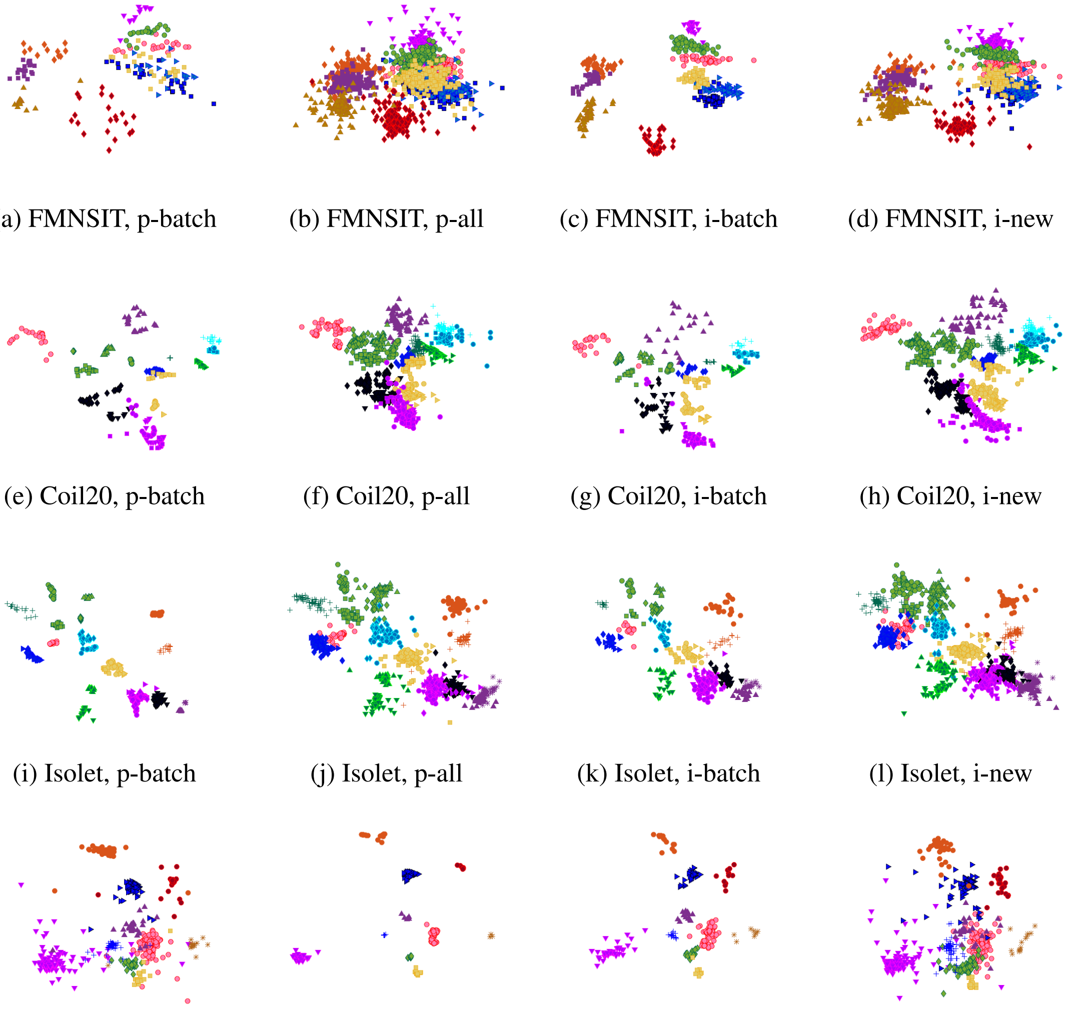
compared with results obtained by setting a larger batch data size $b = \lfloor 0.3n \rfloor$. Thus, users can choose the appropriate batch data size to cope with different sizes of datasets and expectations of results. As i-ANGEL shares a similar strategy and parameters of p-ANGEL, we do not do further research on the parameters of the i-ANGEL.

7.4 Chapter Summary

This chapter presents an extended algorithm for ANGEL to reduce memory consumption and time consumption and enable it to process new data samples continuously. The p-ANGEL aims to reduce memory consumption. The i-ANGEL is an incremental DR algorithm that can process truly new data samples and has the advantage of reduced time consumption. Experimental results demonstrate the effectiveness of applying p-ANGEL and i-ANGEL to real-world queueing datasets for visualisation.

Method		$b = \lfloor 0.3n \rfloor$	$b = \lfloor 0.3n \rfloor$	$b = \lfloor 0.1n \rfloor$	Method		$b = \lfloor 0.3n \rfloor$	$b = \lfloor 0.3n \rfloor$	$b = \lfloor 0.1n \rfloor$
		$par = 1$	$par = 0.5$	$par = 0.5$			$par = 1$	$par = 0.5$	$par = 0.5$
MNSIT	P_{l}	**0.1888**	0.1664	0.1811	FMNIST	P_{l}	**0.1535**	0.1326	0.1521
	P_{g}	0.5370	0.5370	**0.5588**		P_{g}	0.7624	0.7333	**0.7685**
	P_{s}	**0.8840**	0.8400	0.8520		P_{s}	**0.8265**	0.8215	0.8140
	P	**0.5366**	0.5145	0.5306		P	**0.5808**	0.5625	0.5782
	t	1362.6	**632.6**	772.2		t	4364.2	3049.0	**2184.5**
Coil20	P_{l}	**0.3540**	0.3171	0.3176	Cifar	P_{l}	**0.1449**	0.1400	0.1394
	P_{g}	0.2156	**0.2289**	0.2180		P_{g}	0.3018	**0.4655**	0.3442
	P_{s}	**0.8972**	0.8910	0.8826		P_{s}	**0.7790**	0.6710	0.7040
	P	**0.4889**	0.4790	0.4728		P	0.4086	**0.4225**	0.3959
	t	2459.7	993.8	**954.0**		t	3018.2	**1219.6**	1323.7
Isolet	P_{l}	**0.2063**	0.2029	0.1980	Reuters	P_{l}	**0.2009**	0.1746	0.1838
	P_{g}	**0.2525**	0.2061	0.2115		P_{g}	0.2667	0.2630	**0.3382**
	P_{s}	**0.7558**	0.6929	0.7071		P_{s}	**0.9567**	0.9532	0.9327
	P	**0.4048**	0.3673	0.3722		P	0.4747	0.4636	**0.4849**
	t	2549.7	1012.8	**958.7**		t	1779.0	1150.2	**1089.2**

Table 7.3: Evaluation results of p-ANGEL under different parameter settings. t refers to the time of the data embedding process. The bold number highlights the best score: highest preservation score and the lowest t.



(a) FMNSIT, p-batch (b) FMNSIT, p-all (c) FMNSIT, i-batch (d) FMNSIT, i-new

(e) Coil20, p-batch (f) Coil20, p-all (g) Coil20, i-batch (h) Coil20, i-new

(i) Isolet, p-batch (j) Isolet, p-all (k) Isolet, i-batch (l) Isolet, i-new

(m) Reusters, p-batch (n) Reusters, p-all (o) Reusters, i-batch (p) Reusters, i-new

Figure 7.1: Embedding results of real-world datasets using p-ANGEL and i-ANGEL. The p-batch and i-batch refers to the embedding results of the batch dataset, while the p-all refers to the embedding results of all data samples, i-new refers to the embedding results containing all new coming samples.

Chapter 8

Conclusions and Future Direction

This thesis discusses dimension reduction methods applied to data visualisation tasks, i.e. transforming a high-dimensional dataset into $2, 3$-D representation points that retain the inherent structure of the dataset. The ANGEL algorithm was proposed as a response to a limitation of most state-of-the-art DR methods: the incapability of making the embedding achieve the desired properties (local neighbourhood preservation, cohort positioning preservation, and cohort separability) simultaneously. A novel evaluation approach was also presented to give more reliable quantitative measurements of embedding results, allowing ANGEL to compete with other state-of-the-art approaches. Variations and extensions of the ANGEL algorithm were discussed to handle the time and memory combustion problems of ANGEL. Moreover, incremental ANGEL (i-ANGEL) was proposed to continuously deal with the new coming data.

In this chapter, the contributions and findings of the project are summarised, and an overview of future directions is presented.

8.1 Conclusions

This section summarises the work that have been carried out and presented in Chapter 4, 5, 6 and 7. It also outlines how the project met the objectives stated in Section 1.2.

1. **Propose a strategy to retain cohort positioning**

   Similarities/dissimilarities between cohorts can be obtained by applying cohort proximity calculation approaches [110, 111] to the cohort dataset. The average-linkage distance is utilised to calculate the distance between cohorts. Then, the SOE algorithm is applied to embed each cohort to $2,3$-D space to preserve the relative cohort relations based on the cohorts' distances. The obtained embedding points retain the positions of cohorts due to the advantage of the SOE algorithm. Main procedures can be found in Section 4.2. The minor contribution is to apply the **data selection process based on the Gaussian distribution assumption**, in order to make each cohort more identifiable and make cohort relations more accurate.

2. **Propose a strategy to preserve cohorts' internal structure**

   In order to represent the intrinsic characteristics of each cohort, anchor points are generated by considering the density of data points in each cohort. The K-means algorithm [80] is used to partition each data cohort into several smaller clusters. Each anchor point is taken as the centroid point of each small cluster.A **novel tri-OE approach** is proposed to embed the generated anchor points to the target $2,3$-D space, preserving as much ordinal information of anchor points as possible but use less ordinal constraints than the original soft ordinal embedding. The proposed method reduces time consumption but does not reduce the embedding quality. This is a minor contribution and has been stated in Section 4.4.

3. **Propose strategy to enhance the separation between cohorts**

   The basic idea of enhancing the separability between cohorts is to enlarge the dissimilarities of between-cohort data samples and emphasise similarities between intra-cohort samples. As anchor points represent each cohort, we proposed **the supervised tri-OE algorithm** in Section 4.4. The label information is introduced to the proposed tri-OE approach in order to manually adjust the ordinal triplets to enhance the separability of between-cohort anchor points. The modification process we proposed is controlled by the parameter λ, which will also be studied in Section 5.2. Moreover, in order to maintain the cohort positioning, an **anchor relocation algorithm** is proposed to adjust the locations of anchor points according to the position of cohorts in $2,3$-D space in Section 4.4.2. These are major contributions of the thesis.

4. **Develop an algorithm to simultaneously to achieve the property of enhancing separation between data cohorts, preserving distances between individual data points, and retaining cohort positioning and the cohort's internal structure**

   A multi-objective function is proposed to achieve the objectives simultaneously, which is the last part of the **ANGEL model**. The cost function combines the local objective function and the global objective function. The proposed objective function O_{LAE} (Eq.4.14) is the global objective function proposed based on LAE approach (Section 3.4). The O_{LOE} (Eq.4.5) is set to be local objective function (Section 4.5). Another local objective function t-SNE is also discussed in Section 6.2. Two different ways of constructing and optimising the multi-objective model have been discussed: the weighted-sum optimisation method and the r-constrained optimisation method. The latter approach is adopted as the optimisation method used in the ANGEL model (Section 4.5). Moreover, **variations of ANGEL** are proposed in Section 6 to obtain approximate embedding results via novel empirical faster optimisation approaches. To reduce the memory cost of the ANGEL model, **p-ANGEL** is proposed in Section 7.1, which also leads to the construction of the incremental extension of ANGEL. This part is the major contribution of the thesis.

5. **Develop evaluation approaches to measure the performance of the embedding algorithm for visualisation**

   A **novel evaluation metric** was proposed, which provides a quantitative measure of embedding results and compares the performance of different DR methods. It averages the proposed approximated local neighbourhood preservation score, the global cohort positioning preservation score, and the cohort separability score. Evaluation results confirmed the intrinsic characteristics of the existing algorithms and illustrated that ANGEL could obtain improved overall performance compared with state-of-the-art approaches. This is a major contribution and is stated in Chapter 5.

6. **Developing an incremental extension of the proposed DR algorithm for visualisation**

   The **incremental extension of ANGEL (i-ANGEL)** is proposed to process the new coming data samples, which is a major contribution described in Section

7.2. The nearest neighbourhood is first updated based on the coming data sample. Then, the influenced points and the new data sample are updated via the data embedding process of the ANGEL model. Experimental results illustrate that the i-ANGEL is able to preserve the properties as the ANGEL does but suffers a local neighbourhood preservation loss.

8.2 Future Direction

Even though some progress on generating a *good* cohort data visualisation has been made, these achievements are only the tip of the iceberg. There are still many problems that are expected to be investigated in the future. This section gives an overview of potential future directions in DR techniques for data visualisation:

- ANGEL is proposed to simultaneously enhance the separation between data cohorts, preserve distances between individual data points, and retain cohort positioning and the cohort's internal structure. However, since the construction of ANGEL is based on the LOE methods, the time consumption is one of the major limitations of applying ANGEL to the large real-world dataset. In addition, although many acceleration algorithms have been proposed in this project, it is still difficult to obtain fast embedding results for large datasets. As a result, further works can be done in order to deal with this time-consuming problem.

- Various optimisation approaches are proposed to solve the ANGEL model. However, most of them are based on the empirical experimental observations but no theoretical support. For example, the stochastic optimisation of ANGEL (s-ANGEL) results show the convergence of the s-ANGEL, but no mathematical support is given. Therefore, further work can focus on giving theoretical proof and explanations of proposed optimisation approaches.

- The proposed evaluation metric adopts Spearman's rank correlation coefficient to measure the preservation of the cohorts' positioning. However, the measurement of the preservation of the global structure of the embedded points is still an open question. It is an important topic that can also lead to the development of global structure preservation approaches. For this reason, developing a better evaluation metic can be a direction of future research.

- Another research gap lies in the visualisation of the multi-label data. Classical approaches such as tSNE, UMAP, and recently proposed methods such as COVA and PaCAMP are all unable to deal with the multi-label dataset. ANGEL also shares this limitation. The SLE-ML [112] and KSLE-ML [112] handle this issue, and it would be meaningful to develop further extensions of ANGEL to deal with the multi-label dataset.

Bibliography

[1] Timothy Apasiba Abeo, Xiang-Jun Shen, Bing-Kun Bao, Zheng-Jun Zha, and Jianping Fan. A generalized multi-dictionary least squares framework regularized with multi-graph embeddings. *Pattern Recognition*, 90:1–11, 2019.

[2] Timothy Apasiba Abeo, Xiang-Jun Shen, Ernest Domanaanmwi Ganaa, Qian Zhu, Bing-Kun Bao, and Zheng-Jun Zha. Manifold alignment via global and local structures preserving pca framework. *IEEE Access*, 7:38123–38134, 2019.

[3] Sameer Agarwal, Josh Wills, Lawrence Cayton, Gert Lanckriet, David Kriegman, and Serge Belongie. Generalized non-metric multidimensional scaling. In *Artificial Intelligence and Statistics*, pages 11–18, 2007.

[4] William E Allen, Han Altae-Tran, James Briggs, Xin Jin, Glen McGee, Andy Shi, Rumya Raghavan, Mireille Kamariza, Nicole Nova, Albert Pereta, et al. Population-scale longitudinal mapping of covid-19 symptoms, behaviour and testing. *Nature Human Behaviour*, 4(9):972–982, 2020.

[5] Ehsan Amid, Nikos Vlassis, and Manfred K Warmuth. Low-dimensional data embedding via robust ranking. *arXiv preprint arXiv:1611.09957*, 2016.

[6] Ehsan Amid and Manfred K Warmuth. Trimap: Large-scale dimensionality reduction using triplets. *arXiv preprint arXiv:1910.00204*, 2019.

[7] 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. *Nature biotechnology*, 31(6):545–552, 2013.

[8] Sanjeev Arora, Wei Hu, and Pravesh K Kothari. An analysis of the t-sne algorithm for data visualization. In *Conference On Learning Theory*, pages 1455–1462. PMLR, 2018.

[9] Mukund Balasubramanian and Eric L Schwartz. The isomap algorithm and topological stability. *Science*, 295(5552):7–7, 2002.

[10] Etienne Becht, Charles-Antoine Dutertre, Immanuel WH Kwok, Lai Guan Ng, Florent Ginhoux, and Evan W Newell. Evaluation of umap as an alternative to t-sne for single-cell data. *BioRxiv*, page 298430, 2018.

[11] Etienne Becht, Leland McInnes, John Healy, Charles-Antoine Dutertre, Immanuel WH Kwok, Lai Guan Ng, Florent Ginhoux, and Evan W Newell. Dimensionality reduction for visualizing single-cell data using umap. *Nature biotechnology*, 37(1):38–44, 2019.

[12] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.

[13] Yoshua Bengio, Jean-françcois Paiement, Pascal Vincent, Olivier Delalleau, Nicolas Roux, and Marie Ouimet. Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering. *Advances in neural information processing systems*, 16, 2003.

[14] Pardis Birzhandi and Hee Yong Youn. Cbch (clustering-based convex hull) for reducing training time of support vector machine. *The Journal of Supercomputing*, 75(8):5261–5279, 2019.

[15] Ingwer Borg and Patrick Groenen. Modern multidimensional scaling: Theory and applications. *Journal of Educational Measurement*, 40(3):277–280, 2003.

[16] Deng Cai, Xiaofei He, and Jiawei Han. Document clustering using locality preserving indexing. *IEEE Transactions on Knowledge and Data Engineering*, 17(12):1624–1637, December 2005.

[17] Deng Cai, Xiaofei He, and Jiawei Han. Speed up kernel discriminant analysis. *The VLDB Journal*, 20(1):21–33, 2011.

[18] Deng Cai, Xiaofei He, Jiawei Han, and Thomas S Huang. Graph regularized nonnegative matrix factorization for data representation. *IEEE transactions on pattern analysis and machine intelligence*, 33(8):1548–1560, 2010.

[19] Deng Cai, Xiaofei He, Wei Vivian Zhang, and Jiawei Han. Regularized locality preserving indexing via spectral regression. In *Proceedings of the 16th ACM conference on Conference on information and knowledge management (CIKM'07)*, pages 741–750, 2007.

[20] Deng Cai, Qiaozhu Mei, Jiawei Han, and Chengxiang Zhai. Modeling hidden topics on document manifold. In *Proceeding of the 17th ACM conference on Information and knowledge management (CIKM'08)*, pages 911–920, 2008.

[21] Deng Cai, Xuanhui Wang, and Xiaofei He. Probabilistic dyadic data analysis with local and global consistency. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML'09)*, pages 105–112, 2009.

[22] Miguel A Carreira-Perpinán. The elastic embedding algorithm for dimensionality reduction. In *ICML*, volume 10, pages 167–174, 2010.

[23] Jing Chen and Zhengming Ma. Locally linear embedding: a review. *International Journal of Pattern Recognition and Artificial Intelligence*, 25(07):985–1008, 2011.

[24] Lisha Chen and Andreas Buja. Local multidimensional scaling for nonlinear dimension reduction, graph drawing, and proximity analysis. *Journal of the American Statistical Association*, 104(485):209–219, 2009.

[25] Heeyoul Choi and Seungjin Choi. Robust kernel isomap. *Pattern recognition*, 40(3):853–862, 2007.

[26] Yuanfei Dai, Chenhao Guo, Wenzhong Guo, and Carsten Eickhoff. Drug–drug interaction prediction with wasserstein adversarial autoencoder-based knowledge graph embeddings. *Briefings in Bioinformatics*, 2020.

[27] C. de Bodt, D. Mulders, D. L
'opez-S
'anchez, M. Verleysen, and J. A. Lee. Class-aware t-SNE: cat-SNE. In *ESANN*, pages 409–414, 2019.

[28] Vin De Silva and Gunnar E Carlsson. Topological estimation using witness complexes. In *PBG*, pages 157–166, 2004.

[29] Vin De Silva and Joshua B Tenenbaum. Sparse multidimensional scaling using landmark points. Technical report, Technical report, Stanford University, 2004.

[30] Alex Diaz-Papkovich, Luke Anderson-Trocmé, and Simon Gravel. A review of umap in population genetics. *Journal of Human Genetics*, pages 1–7, 2020.

[31] Paweł Dłotko. Ball mapper: A shape summary for topological data analysis. *arXiv preprint arXiv:1901.07410*, 2019.

[32] David L Donoho and Carrie Grimes. Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *Proceedings of the National Academy of Sciences*, 100(10):5591–5596, 2003.

[33] Michael W Dorrity, Lauren M Saunders, Christine Queitsch, Stanley Fields, and Cole Trapnell. Dimensionality reduction by umap to visualize physical and genetic interactions. *Nature communications*, 11(1):1–6, 2020.

[34] John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. Efficient projections onto the l 1-ball for learning in high dimensions. In *Proceedings of the 25th international conference on Machine learning*, pages 272–279, 2008.

[35] Mohammed Elhenawy, Mahmoud Masoud, Sebastien Glaser, and Andry Rakotonirainy. A new approach to improve the topological stability in non-linear dimensionality reduction. *IEEE Access*, 8:33898–33908, 2020.

[36] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer vision and Image understanding*, 106(1):59–70, 2007.

[37] Min Feng, Jianxin Liao, Jingyu Wang, Sude Qing, and Qi Qi. Topology-aware virtual network embedding based on multiple characteristics. In *2014 IEEE International Conference on Communications (ICC)*, pages 2956–2962. IEEE, 2014.

[38] Pasi Fränti and Sami Sieranoja. K-means properties on six clustering benchmark datasets, 2018.

[39] Cong Fu, Yonghui Zhang, Deng Cai, and Xiang Ren. AtSNE: Efficient and robust visualization on GPU through hierarchical optimization. In *Proceedings*

of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pages 176–186, 2019.

[40] Takanori Fujiwara. *Advancing Visual Analytics Using Dimensionality Reduction*. PhD thesis, University of California, Davis, 2021.

[41] Kelum Gajamannage, Randy Paffenroth, and Erik M Bollt. A nonlinear dimensionality reduction framework using smooth geodesics. *Pattern Recognition*, 87:226–236, 2019.

[42] Xiaofang Gao and Jiye Liang. An improved incremental nonlinear dimensionality reduction for isometric data embedding. *Information Processing Letters*, 115(4):492–501, 2015.

[43] Wil Gardner, Ruqaya Maliki, Suzanne M Cutts, Benjamin W Muir, Davide Ballabio, David A Winkler, and Paul J Pigram. Self-organizing map and relational perspective mapping for the accurate visualization of high-dimensional hyperspectral data. *Analytical Chemistry*, 92(15):10450–10459, 2020.

[44] Xin Geng, De-Chuan Zhan, and Zhi-Hua Zhou. Supervised nonlinear dimensionality reduction for visualization and classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 35(6):1098–1107, 2005.

[45] Benyamin Ghojogh, Ali Ghodsi, Fakhri Karray, and Mark Crowley. Uniform manifold approximation and projection (UMAP) and its variants: Tutorial and survey. *arXiv preprint arXiv:2109.02508*, 2021.

[46] Andrej Gisbrecht and Barbara Hammer. Data visualization by nonlinear dimensionality reduction. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 5(2):51–73, 2015.

[47] Andrej Gisbrecht, Daniela Hofmann, and Barbara Hammer. Discriminative dimensionality reduction mappings. In *International Symposium on Intelligent Data Analysis*, pages 126–138. Springer, 2012.

[48] Andrej Gisbrecht, Wouter Lueks, Bassam Mokbel, Barbara Hammer, et al. Out-of-sample kernel extensions for nonparametric dimensionality reduction. In *ESANN*, volume 2012, pages 531–536, 2012.

[49] Andrej Gisbrecht, Alexander Schulz, and Barbara Hammer. Parametric nonlinear dimensionality reduction using kernel t-sne. *Neurocomputing*, 147:71–82, 2015.

[50] Patrick JF Groenen, Suzanne Winsberg, O Rodriguez, and Edwin Diday. I-scal: Multidimensional scaling of interval dissimilarities. *Computational Statistics & Data Analysis*, 51(1):360–378, 2006.

[51] Yacov Haimes. On a bicriterion formulation of the problems of integrated system identification and system optimization. *IEEE transactions on systems, man, and cybernetics*, 1(3):296–297, 1971.

[52] Laureta Hajderanj, Daqing Chen, and Isakh Weheliye. The impact of supervised manifold learning on structure preserving and classification error: a theoretical study. *IEEE Access*, 9:43909–43922, 2021.

[53] Laureta Hajderanj, Isakh Weheliye, and Daqing Chen. A new supervised t-sne with dissimilarity measure for effective data visualization and classification. In *Proceedings of the 2019 8th International Conference on Software and Information Engineering*, pages 232–236. ACM, 2019.

[54] Rassoul Hajizadeh, Ali Aghagolzadeh, and Mehdi Ezoji. Local distances preserving based manifold learning. *Expert Systems with Applications*, 139:112860, 2020.

[55] John A Hartigan. Printer graphics for clustering. *Journal of Statistical Computation and Simulation*, 4(3):187–213, 1975.

[56] Maike Hennen, Philip Voll, and André Bardow. An adaptive normal constraint method for bi-objective optimal synthesis of energy systems. In *Computer Aided Chemical Engineering*, volume 33, pages 1279–1284. Elsevier, 2014.

[57] Geoffrey E Hinton and Sam T Roweis. Stochastic neighbor embedding. In *Advances in neural information processing systems*, pages 857–864, 2003.

[58] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

[59] Alfred Inselberg and Bernard Dimsdale. Parallel coordinates for visualizing multi-dimensional geometry. In *Computer Graphics 1987*, pages 25–44. Springer, 1987.

[60] Alan Julian Izenman. Introduction to manifold learning. *Wiley Interdisciplinary Reviews: Computational Statistics*, 4(5):439–446, 2012.

[61] Robert A Jacobs. Increased rates of convergence through learning rate adaptation. *Neural networks*, 1(4):295–307, 1988.

[62] Peng Jia, Junsong Yin, Xinsheng Huang, and Dewen Hu. Incremental laplacian eigenmaps by preserving adjacent information between data points. *Pattern Recognition Letters*, 30(16):1457–1463, 2009.

[63] Bo Jiang, Chris Ding, Bio Luo, and Jin Tang. Graph-laplacian pca: Closed-form solution and robustness. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3492–3498, 2013.

[64] Jaemin Jo, Jinwook Seo, and Jean-Daniel Fekete. Panene: A progressive algorithm for indexing and querying approximate k-nearest neighbors. *IEEE transactions on visualization and computer graphics*, 26(2):1347–1360, 2018.

[65] Sara Johansson and Jimmy Johansson. Interactive dimensionality reduction through user-defined combinations of quality metrics. *IEEE transactions on visualization and computer graphics*, 15(6):993–1000, 2009.

[66] Samuel Kaski, Janne Nikkilä, Merja Oja, Jarkko Venna, Petri Törönen, and Eero Castrén. Trustworthiness and metrics in visualizing similarity of gene expression. *BMC bioinformatics*, 4(1):1–13, 2003.

[67] James M Keller, Michael R Gray, and James A Givens. A fuzzy k-nearest neighbor algorithm. *IEEE transactions on systems, man, and cybernetics*, (4):580–585, 1985.

[68] Hannah Kim, Jaegul Choo, Chandan K Reddy, and Haesun Park. Doubly supervised embedding based on class labels and intrinsic clusters for high-dimensional data visualization. *Neurocomputing*, 150:570–582, 2015.

[69] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[70] Hyung-Kwon Ko, Jaemin Jo, and Jinwook Seo. Progressive uniform manifold approximation and projection. In *EuroVis (Short Papers)*, pages 133–137, 2020.

[71] Dmitry Kobak, George Linderman, Stefan Steinerberger, Yuval Kluger, and Philipp Berens. Heavy-tailed kernels reveal a finer cluster structure in t-sne visualisations. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 124–139. Springer, 2019.

[72] Dmitry Kobak and George C Linderman. Initialization is critical for preserving global data structure in both t-sne and umap. *Nature biotechnology*, 39(2):156–157, 2021.

[73] Olga Kouropteva, Oleg Okun, and Matti Pietikäinen. Incremental locally linear embedding. *Pattern recognition*, 38(10):1764–1767, 2005.

[74] Kenneth Kreutz-Delgado, Joseph F Murray, Bhaskar D Rao, Kjersti Engan, Te-Won Lee, and Terrence J Sejnowski. Dictionary learning algorithms for sparse representation. *Neural computation*, 15(2):349–396, 2003.

[75] Joseph B Kruskal. Nonmetric multidimensional scaling: a numerical method. *Psychometrika*, 29(2):115–129, 1964.

[76] Martin HC Law and Anil K Jain. Incremental nonlinear dimensionality reduction by manifold learning. *IEEE transactions on pattern analysis and machine intelligence*, 28(3):377–391, 2006.

[77] John A Lee and Michel Verleysen. *Nonlinear dimensionality reduction*, volume 1. Springer, 2007.

[78] Chun-Guang Li and Jun Guo. Supervised isomap with explicit mapping. In *First International Conference on Innovative Computing, Information and Control-Volume I (ICICIC'06)*, volume 3, pages 345–348. IEEE, 2006.

[79] Housen Li, Hao Jiang, Roberto Barrio, Xiangke Liao, Lizhi Cheng, and Fang Su. Incremental manifold learning by spectral embedding methods. *Pattern Recognition Letters*, 32(10):1447–1455, 2011.

[80] Aristidis Likas, Nikos Vlassis, and Jakob J Verbeek. The global k-means clustering algorithm. *Pattern recognition*, 36(2):451–461, 2003.

[81] Tony Lin, Hongbin Zha, and Sang Uk Lee. Riemannian manifold learning for nonlinear dimensionality reduction. In *European Conference on Computer Vision*, pages 44–55. Springer, 2006.

[82] 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(3):243–245, 2019.

[83] Feng Liu, Weijie Zhang, and Suicheng Gu. Local linear laplacian eigenmaps: A direct extension of lle. *Pattern Recognition Letters*, 75:30–35, 2016.

[84] Shixia Liu, Weiwei Cui, Yingcai Wu, and Mengchen Liu. A survey on information visualization: recent advances and challenges. *The Visual Computer*, 30(12):1373–1393, 2014.

[85] Shusen Liu, Dan Maljovec, Bei Wang, Peer-Timo Bremer, and Valerio Pascucci. Visualizing high-dimensional data: Advances in the past decade. *IEEE transactions on visualization and computer graphics*, 23(3):1249–1268, 2016.

[86] Wei Liu, Junfeng He, and Shih-Fu Chang. Large graph construction for scalable semi-supervised learning. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, pages 679–686, 2010.

[87] Xiaoming Liu, Jianwei Yin, Zhilin Feng, and Jinxiang Dong. Incremental manifold learning via tangent space alignment. In *IAPR Workshop on Artificial Neural Networks in Pattern Recognition*, pages 107–121. Springer, 2006.

[88] Yunqian Ma and Yun Fu. *Manifold learning theory and applications*, volume 434. CRC press Boca Raton, 2012.

[89] Brian McFee and Gert Lanckriet. Partial order embedding with multiple kernels. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 721–728. ACM, 2009.

[90] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[91] Martin R Min, Laurens Maaten, Zineng Yuan, Anthony J Bonner, and Zhaolei Zhang. Deep supervised t-distributed embedding. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 791–798, 2010.

[92] Tingting Mu, John Yannis Goulermas, and Sophia Ananiadou. Data visualization with structural control of global cohort and local data neighborhoods. *IEEE*

transactions on pattern analysis and machine intelligence, 40(6):1323–1337, 2017.

[93] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2003.

[94] Luis Gustavo Nonato and Michael Aupetit. Multidimensional projection for visual analytics: Linking techniques with distortions, tasks, and layout enrichment. *IEEE Transactions on Visualization and Computer Graphics*, 25(8):2650–2673, 2018.

[95] Nikolaos Passalis and Anastasios Tefas. Pysef: A python library for similarity-based dimensionality reduction. *Knowledge-Based Systems*, 2018.

[96] Jaakko Peltonen, Arto Klami, and Samuel Kaski. Improved learning of riemannian metrics for exploratory analysis. *Neural Networks*, 17(8-9):1087–1100, 2004.

[97] John Platt. Fastmap, metricmap, and landmark mds are all nyström algorithms. In *International Workshop on Artificial Intelligence and Statistics*, pages 261–268. PMLR, 2005.

[98] Ning Qian. On the momentum term in gradient descent learning algorithms. *Neural networks*, 12(1):145–151, 1999.

[99] Ignacio Ramirez, Pablo Sprechmann, and Guillermo Sapiro. Classification and clustering via dictionary learning with structured incoherence and shared features. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3501–3508. IEEE, 2010.

[100] Guy Rosman, Michael M Bronstein, Alexander M Bronstein, and Ron Kimmel. Nonlinear dimensionality reduction by topologically constrained isometric embedding. *International Journal of Computer Vision*, 89(1):56–68, 2010.

[101] Günter Rote. Computing the minimum hausdorff distance between two point sets on a line under translation. *Information Processing Letters*, 38(3):123–127, 1991.

[102] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326, 2000.

[103] Tim Sainburg, Leland McInnes, and Timothy Q Gentner. Parametric umap embeddings for representation and semi-supervised learning. *arXiv preprint arXiv:2009.12981*, 2020.

[104] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Kernel principal component analysis. In *International conference on artificial neural networks*, pages 583–588. Springer, 1997.

[105] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10(5):1299–1319, 1998.

[106] Blake Shaw and Tony Jebara. Structure preserving embedding. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 937–944, 2009.

[107] Roger N Shepard. The analysis of proximities: multidimensional scaling with an unknown distance function. i. *Psychometrika*, 27(2):125–140, 1962.

[108] Roger N Shepard. The analysis of proximities: Multidimensional scaling with an unknown distance function. ii. *Psychometrika*, 27(3):219–246, 1962.

[109] Vin D Silva and Joshua B Tenenbaum. Global versus local methods in nonlinear dimensionality reduction. In *Advances in neural information processing systems*, pages 721–728, 2003.

[110] Robert R Sokal. Numerical taxonomy. *Scientific American*, 215(6):106–117, 1966.

[111] Gabor J Szekely and Maria L Rizzo. Hierarchical clustering via joint between-within distances: Extending ward's minimum variance method. *Journal of classification*, 22(2):151–183, 2005.

[112] Mariko Tai and Mineichi Kudo. A supervised laplacian eigenmap algorithm for visualization of multi-label data: Sle-ml. In *Iberoamerican Congress on Pattern Recognition*, pages 525–534. Springer, 2019.

[113] 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*, pages 287–297, 2016.

[114] Joshua Tenenbaum. Mapping a manifold of perceptual observations. *Advances in neural information processing systems*, 10, 1997.

[115] Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

[116] Yoshikazu Terada and Ulrike Luxburg. Local ordinal embedding. In *International Conference on Machine Learning*, pages 847–855. PMLR, 2014.

[117] Ivana Tosic and Pascal Frossard. Dictionary learning. *IEEE Signal Processing Magazine*, 28(2):27–38, 2011.

[118] Laurens Van Der Maaten. Learning a parametric embedding by preserving local structure. In *Artificial Intelligence and Statistics*, pages 384–391, 2009.

[119] Laurens Van Der Maaten. Barnes-hut-sne. *arXiv preprint arXiv:1301.3342*, 2013.

[120] Laurens Van Der Maaten. Accelerating t-sne using tree-based algorithms. *The journal of machine learning research*, 15(1):3221–3245, 2014.

[121] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[122] Laurens Van Der Maaten, Eric Postma, and Jaap Van den Herik. Dimensionality reduction: a comparative. *J Mach Learn Res*, 10(66-71):13, 2009.

[123] Laurens Van Der Maaten and Kilian Weinberger. Stochastic triplet embedding. In *2012 IEEE International Workshop on Machine Learning for Signal Processing*, pages 1–6. IEEE, 2012.

[124] Leena Chennuru Vankadara, Siavash Haghiri, Michael Lohaus, Faiz Ul Wahab, and Ulrike von Luxburg. Insights into ordinal embedding algorithms: A systematic evaluation. *arXiv preprint arXiv:1912.01666*, 2019.

[125] Jarkko Venna and Samuel Kaski. Neighborhood preservation in nonlinear projection methods: An experimental study. In *International Conference on Artificial Neural Networks*, pages 485–491. Springer, 2001.

[126] Jarkko Venna and Samuel Kaski. Local multidimensional scaling. *Neural Networks*, 19(6-7):889–899, 2006.

[127] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008.

[128] Michail Vlachos, Carlotta Domeniconi, Dimitrios Gunopulos, George Kollios, and Nick Koudas. Non-linear dimensionality reduction techniques for classification and visualization. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 645–651, 2002.

[129] Meng Wang, Weijie Fu, Shijie Hao, Dacheng Tao, and Xindong Wu. Scalable semi-supervised learning by efficient anchor graph regularization. *IEEE Transactions on Knowledge and Data Engineering*, 28(7):1864–1877, 2016.

[130] Yasi Wang, Hongxun Yao, and Sicheng Zhao. Auto-encoder based dimensionality reduction. *Neurocomputing*, 184:232–242, 2016.

[131] Yingfan Wang, Haiyang Huang, Cynthia Rudin, and Yaron Shaposhnik. Understanding how dimension reduction tools work: An empirical approach to deciphering t-sne, umap, trimap, and pacmap for data visualization. *Journal of Machine Learning Research*, 22(201):1–73, 2021.

[132] Larry Wasserman. Topological data analysis. *Annual Review of Statistics and Its Application*, 5:501–532, 2018.

[133] Martin Wattenberg, Fernanda Viégas, and Ian Johnson. How to use t-sne effectively. *Distill*, 1(10):e2, 2016.

[134] Leland Wilkinson, Anushka Anand, and Robert Grossman. Graph-theoretic scagnostics. In *Information Visualization, IEEE Symposium on*, pages 21–21. IEEE Computer Society, 2005.

[135] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

[136] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

[137] Yu Xie, Maoguo Gong, A Kai Qin, Zedong Tang, and Xiaolong Fan. Tpne: Topology preserving network embedding. *Information Sciences*, 504:20–31, 2019.

[138] Yimin Yang, QM Jonathan Wu, and Yaonan Wang. Autoencoder with invertible functions for dimension reduction and image reconstruction. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 48(7):1065–1079, 2016.

[139] Xiaoru Yuan, Peihong Guo, He Xiao, Hong Zhou, and Huamin Qu. Scattering points in parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1001–1008, 2009.

[140] Lofti Zadeh. Optimality and non-scalar-valued performance criteria. *IEEE transactions on Automatic Control*, 8(1):59–60, 1963.

[141] Haili Zhang, Pu Wang, Xuejin Gao, Yongsheng Qi, and Huihui Gao. Out-of-sample data visualization using bi-kernel t-sne. *Information Visualization*, 20(1):20–34, 2021.

[142] Kai Zhang, James T Kwok, and Bahram Parvin. Prototype vector machine for large scale semi-supervised learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1233–1240. ACM, 2009.

[143] Shi-qing Zhang. Enhanced supervised locally linear embedding. *Pattern Recognition Letters*, 30(13):1208–1218, 2009.

[144] Yan Zhang, Zhao Zhang, Jie Qin, Li Zhang, Bing Li, and Fanzhang Li. Semi-supervised local multi-manifold isomap by linear embedding for feature extraction. *Pattern Recognition*, 76:662–678, 2018.

[145] Zhenyue Zhang and Jing Wang. Mlle: Modified locally linear embedding using multiple weights. *Advances in neural information processing systems*, 19, 2006.

[146] Zhenyue Zhang and Hongyuan Zha. Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *SIAM journal on scientific computing*, 26(1):313–338, 2004.

[147] Lingxiao Zhao and Zhenyue Zhang. Supervised locally linear embedding with probability-based distance for classification. *Computers & Mathematics with Applications*, 57(6):919–926, 2009.

[148] Xiaoming Zhao and Shiqing Zhang. Facial expression recognition using local binary patterns and discriminant kernel locally linear embedding. *EURASIP journal on Advances in signal processing*, 2012(1):1–9, 2012.

[149] Nanning Zheng and Jianru Xue. *Statistical learning and pattern analysis for image and video processing*. Springer Science & Business Media, 2009.

[150] Christina E Zielinski. Meeting the challenges of high-dimensional single-cell data analysis in immunology. *Frontiers in immunology*, 10:1515, 2019.