



LUDWIG-
MAXIMILIANS-
UNIVERSITÄT
MÜNCHEN

INSTITUT FÜR INFORMATIK
LEHRSTUHL FÜR DATENBANKSYSTEME
UND DATA MINING



Master's Thesis
in Computer Science

Order Preservation in Dimensionality Reduction: Structure-Based Embeddings in Real and Complex Space

Anna Maria Thum

Supervisor: Prof. Dr. Thomas Seidl
Advisor: Dr. Anna Beer
Submission Date: 6. October 2023

Declaration of Authorship

I hereby declare that the thesis submitted is my own unaided work. All direct or indirect sources used are acknowledged as references.

This paper was not previously presented to another examination board and has not been published.

München, 6. October 2023

.....
Anna Maria Thum

Abstract

In many real-world datasets, such as time series, the inherent order of data encodes valuable information. However, commonly used dimensionality reduction methods, like Principal Component Analysis (PCA), often cannot capture this order, potentially leading to a loss of crucial insights. In this work, we propose a new focus for dimensionality reduction methods: order-based distance preservation. Moreover, we introduce Principal Component Analysis for Structure-Aware Representation (PCA*) and variants of Complex Principal Components Analysis (CPCA). These methods are designed to incorporate sequential information into the dimensionality reduction process. By considering the inherent relationships in the data, PCA* aims to provide more meaningful and informative representations. This has the potential to enhance data analysis, visualisation, and model building in domains where order plays a critical role. Our experiments demonstrate promising results for order-based distance preservation with PCA* in comparison to other dimensionality reduction methods including PCA, random projection and LLE. To evaluate the quality of order-based distance preservation of dimensionality reduction methods, we employ different measures and conduct experiments on both synthetic and real-world data.

Contents

1	Introduction	3
2	Related Work	5
2.1	Global Structure Preserving Dimensionality Reduction Methods	5
2.1.1	Principal Component Analysis	5
2.1.2	Linear Discriminant Analysis	8
2.2	Local Structure Preserving Dimensionality Reduction Methods .	8
2.2.1	Locally Linear Embedding	9
2.2.2	t-Distributed Stochastic Neighbour Embedding	9
2.2.3	Uniform Manifold Approximation and Projection	10
2.2.4	Random Projections	10
2.2.5	Locality Preserving Projections	10
2.3	Dimensionality Reduction on Ordered Data	11
2.3.1	Time-Lagged Independent Component Analysis	11
2.3.2	Linear Sequence Discriminant Analysis	12
3	Dimensionality Reduction on Ordered Data	13
3.1	Ordered Data	13
3.2	Structure and Order Preservation	14
3.3	Variants of Principal Component Analysis	19
3.3.1	PCA with Ids	19
3.3.2	Complex PCA on Ordered Data	20
3.3.3	PCA*	24
4	Experiments	29
4.1	Datasets	29
4.2	Quality Measures for Dimensionality Reduction Algorithms . . .	31
4.3	Implementation Details	37
4.4	Experiments with CPCA on Synthetic Data	37
4.4.1	Expanding the Dimensionality of the Transformation . .	38
4.4.2	Exploration of Increasing Numbers of Neighbours	43

4.4.3	Analysis of Growing Numbers of Neighbours and Transformation Dimensions	45
4.5	Evaluation with PCA* on Synthetic Data	53
4.5.1	Increasing the Number of Dimensions of the Original Space	53
4.5.2	Expanding the Dimensionality of the Transformation . .	56
4.5.3	Exploration of Increasing Numbers of Neighbours	61
4.5.4	Analysis of Growing Numbers of Neighbours and Transformation Dimensions	63
4.5.5	Number of Lines Versus the Number of Neighbours . . .	71
4.6	Evaluation on Real-World and Simulated Data	78
4.6.1	Expanding the Dimensionality of the Transformation . .	78
4.6.2	Exploration of Increasing Numbers of Neighbours	83
4.6.3	Analysis of Growing Numbers of Neighbours and Transformation Dimensions	84
4.7	Limitations of Experiments	93
4.8	Experimental Results	93
5	Future Work	95
6	Conclusion	97
A	Technical Details on Experiments	98
	Bibliography	99

Chapter 1

Introduction

High-dimensional data poses many challenges and complexities that can impede data analysis, visualisation and interpretation. Among these challenges is the curse of dimensionality. It describes how the data becomes sparser while the dimensionality of the data space increases, which in turn affects distance measures. In addition to the curse of dimensionality, there are other geometric property phenomena. Consider the volume of the unit sphere, for instance. While the volume increases steadily from dimension one to five, it then subsequently decreases and eventually approximates zero as the dimensionality of the unit sphere reaches higher dimensions [50]. Therefore reducing dimensions is an important part of data analysis.

Since there are many challenges with high-dimensional data, there are numerous dimensionality reduction methods offering effective ways to distil valuable insights. For example, it is common to categorise dimensionality reduction techniques into local or global structure-preserving methods. Dimensionality reduction techniques with a high degree of local structure preservation maintain the local distances and relationships between neighbouring data points. Essentially they aim to map nearby points to nearby points in the lower-dimensional space [45]. Some of these techniques are Locally Linear Embedding (LLE) [19], t-Distributed Stochastic Neighbour Embedding (t-SNE) [4] and Uniform Manifold Approximation and Projection (UMAP) [32]. On the other hand, global structure-preserving methods aim to keep the overall relationships and trends in the data. Principal Component Analysis (PCA) emphasises dominant patterns of variability within a population, which can capture the overall patterns and trends across the dataset. In summary, PCA aims to capture the maximum amount of variance in the data by finding the directions of maximum variation through the principal components [46].

In the above-mentioned dimensionality reduction methods the primary focus is on capturing the spatial structure of the data. However, when dealing

with ordered data, which refers to data where the arrangement or sequence of data points holds significance, these methods disregard this aspect. Ordered data implies that the data points are organised in a specific sequence, and the relationships between adjacent points carry valuable information. For example, time series data are ordered, as they consist of observations collected over successive time intervals, and these observations have inherent temporal dependencies [6]. These dependencies require specialised analysis techniques to capture trends, patterns and seasonality accurately [6] [26]. Therefore, in addition to preserving local and global structures, we introduce another essential property: order-based distance preservation for dimensionality reduction methods designed specifically for ordered data. This property aims to ensure that the meaningful order and relationships between data points are maintained during the dimensionality reduction process. Similarly to local structure preservation which focuses on preserving the distances of the k -nearest neighbours, we want to preserve the distances to the predecessor and successor of each data point in the known order. For example, the order-preserving property ensures that in sequentially ordered data like time series the temporal relationships are accurately maintained in the lower-dimensional space. This can aid in understanding patterns, trends and changes over time, leading to more interpretable and actionable results.

We introduce the PCA variant Principal Component Analysis for Structure-Aware Representation (PCA*) which allows us to incorporate sequential order information into the dimensionality reduction process. By considering the vectors between successive data points in sequential data, PCA* captures the inherent relationships and dependencies within the dataset. This approach provides therefore a deterministic way to preserve order-based distances while reducing dimensions.

Furthermore, we compare our method with PCA as well as other unsupervised dimensionality reduction methods, like LLE, t-SNE and UMAP. Like PCA our proposed method is simple to implement and has roughly the same computational runtime as PCA.

This work is structured as follows: in Chapter 2 we give an overview of the PCA algorithm. We also explore other global and local structure-preserving dimensionality reduction methods. Chapter 3 introduces the concept of ordered data emphasising the importance of preserving various structures, particularly order-based distance preservation. Chapter 4 presents the practical aspect, including the datasets, quality measures and implementation details. We conduct experiments on synthetic and real-world data with PCA* and the variants of CPCA. Chapter 5 outlines potential future research directions in the field of order-based distance preservation and PCA*. To conclude this work, we summarise the key findings in the last Chapter 6.

Chapter 2

Related Work

As the volume of collected data continues to expand, driven by ongoing improvements in technology and the complexity of real-world problems, dimensionality reduction has become a fundamental part of modern research [50]. Therefore, there is a growing demand for dimensionality reduction methods with different functionalities. To answer these needs there are diverse dimensionality reduction methods, each tailored to specific data characteristics and analytical goals. In the following sections, we explore various dimensionality reduction methods, from global or local structure-preserving to dimensionality reduction techniques for ordered data.

2.1 Global Structure Preserving Dimensionality Reduction Methods

Global structure-preserving dimensionality reduction methods offer a profound means to retain essential information from high-dimensional data while retaining the inherent relationships and patterns within the data. We start by examining Principal Component Analysis (PCA) and linear discriminant analysis (LDA). Since PCA* is based on PCA, we delve deeper into the theory behind PCA. For other dimensionality reduction methods, we provide a brief description of their core principles.

2.1.1 Principal Component Analysis

Principal Component Analysis (PCA) is one of the most often used dimensionality reduction methods due to its simplicity and non-parametricity [26]. It aims to capture the most essential information within a high-dimensional dataset by transforming the data into a new lower-dimensional space with its

principal components. Principal components are the orthogonal directions in the data space accounting for the maximum variance and thus the transformation results in a new set of axes or dimensions that capture the most significant variations. This makes PCA a powerful method for dimensionality reduction [44] [26]. In this section, we explore how PCA works and where its strengths and weaknesses lie.

As mentioned earlier, PCA is a widely used technique in data analysis and has been defined in various contexts. In this work, we establish our definition based on the work of [31]:

Definition 2.1.1 (Principal Component Analysis). *Let X be data in $\mathbb{R}^{n \times d}$ with each row representing a d -dimensional data point x_i with $i \in \{1, \dots, n\}$. PCA now aims to identify a set of k orthonormal unit vectors $\mathcal{V} = [\nu_1, \dots, \nu_k]$, where each column ν_i represents a principal component capturing the directions of maximum variance in the data space and $k \leq d$. The goal is to maximise the variance of the data projected onto these vectors. Therefore, PCA can be written as an optimisation problem:*

$$\max_{\mathcal{V}} \text{tr}(\mathcal{V}^T X^T X \mathcal{V})$$

such that

$$\mathcal{V}^T \mathcal{V} = \mathbb{1}_{k \times k},$$

where tr denotes the trace operation, $\mathbb{1}_{k \times k}$ is the $k \times k$ unit matrix, and X^T denotes the transposed matrix.

In simpler terms, PCA aims to maximise the variance of the transformed data values. The diagonal entries of the covariance matrix of the original data matrix represent these variances. Transforming the data results in the transformed covariance matrix $\mathcal{V}^T X^T X \mathcal{V}$ for the transformed data shown in the definition above. By maximising the trace of this matrix, we effectively maximise the variance within the transformed data.

When using PCA we are following assumptions [44]:

- **Linearity:** PCA operates on linear combinations of features and therefore, does not capture non-linear relationships or patterns in the data effectively.
- **The larger the variance the more important:** PCA uses variance as a measure of spread and by maximising this, PCA ensures to capture as much information as possible about the data's spread and variability

- Independence: PCA uses the covariance matrix to find the principal components, but this technique assumes certain statistical properties, including independence of the variables [26].

There are multiple ways to compute PCA. One option is using the Singular Value Decomposition (SVD) of the data matrix. SVD is used in linear algebra to decompose a matrix into three separate matrices, which provide insights into the structure and properties of the original matrix.

For SVD, a commonly employed mathematical technique, we define it based on [26]:

Definition 2.1.2 (Singular Value Decomposition). *Suppose matrix X of dimension $n \times d$ is given. The singular value decomposition of X is as defined follows:*

$$X = U\Sigma V^T,$$

where

- U is an $n \times d$ orthonormal matrix which is often referred to as the left singular vectors,
- V is an $n \times d$ orthonormal matrix. It commonly is called the right singular matrix,
- Σ is an $d \times d$ diagonal matrix containing the singular values of X .

PCA involves finding the principal components of a dataset, which are the directions of maximum variance. These are essentially the eigenvectors of the covariance matrix of X . SVD helps to achieve this by decomposing the data such that its principal components are revealed.

First, for the dimensionality reduction of data with PCA, we prepare the data. Let X be the $n \times d$ -dimensional data. To compute the covariance matrix we need to center X . Centring data is important as without it, there might be translation effects and it is not clear if the results are statistically appropriate, see [11]. The covariance matrix is calculated as follows:

$$\text{Cov}(X_{\text{centred}}) = \frac{1}{n} X_{\text{centred}}^T X_{\text{centred}}.$$

We show that the matrix of the left singular vectors U contains the eigenvectors of the covariance matrix of X , by computing:

$$\text{Cov}(X_{\text{centred}}) = \frac{1}{n} X_{\text{centred}}^T X_{\text{centred}} \stackrel{\text{SVD}}{=} \frac{1}{n} (U\Sigma V^T)^T (U\Sigma V^T) = \frac{1}{n} U \Sigma^T \Sigma U^T.$$

Therefore contains $\Sigma^T \Sigma$ the eigenvalues of $\text{Cov}(X_{\text{centred}})$ as diagonal entries and V the eigenvectors, or, in the case of PCA, on X the principal components. To project the data to a lower dimensional space, we truncate V to $k \leq d$ columns and multiply X_{centred} with the truncated matrix. For this, we sort the eigenvalues in descending order. The eigenvectors corresponding to the largest eigenvalues are selected, as they represent the directions of the maximum variance in the data. We denote the truncated right singular vectors as V_k .

If we want to reconstruct the data from the lower-dimensional representation, we multiply the transformed data by the transpose of the principal component matrix V_k . The quality of reconstruction depends on how many principal components are retained.

As PCA captures the directions in which the data varies the most, PCA effectively preserves the overall spread and dispersion of the data and preserves therefore the global structure [46].

2.1.2 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is a supervised dimensionality reduction method, which can be used to classify data in machine learning and statistics when the groups are known. It is often used for feature extraction and reducing the number of dimensions of data while preserving class separability. LDA starts with a labelled dataset where each data point is associated with a class label. Then LDA can be broken down into three main steps: Separability calculation between different classes, calculation of the distance between the mean and the samples of each class, and projection of the data into a lower-dimensional embedding, such that the between-class variance is maximised and the within-class variance is minimised. Even though LDA is an often used technique, there is also a disadvantage: If the number of dimensions is much higher than the number of samples then LDA cannot find a lower-dimensional embedding for the data [49] [15].

2.2 Local Structure Preserving Dimensionality Reduction Methods

Apart from capturing variance and statistical independence, there are dimensionality reduction methods like Locally Linear Embedding (LLE), t-Distributed Stochastic Neighbour Embedding (t-SNE) and Uniform Manifold Approximation and Projection (UMAP) designed to preserve the local structure and relationships in high-dimensional data.

2.2.1 Locally Linear Embedding

LLE is a powerful non-linear dimensionality reduction method preserving the local structure of the data in the lower-dimensional embedding space. Its core idea is to find a representation of the data in the lower-dimensional space such that the local structure is optimally preserved. It first finds with the k -Nearest Neighbours (k -NN) Algorithm the pairwise Euclidean distances between data points. Then each data point is reconstructed as a weighted linear combination of its neighbours such that the neighbourhood of each data point is maintained while projecting it into a lower-dimensional space. In simpler terms, if points are close in the original dataset, then their respective transformed data points should be close in the projected dataset. As LLE only considers close data points found with the k -NN algorithm, there is no certainty given that far away points stay far away from each other in the embedding space. Additionally to this, LLe is also sensitive to noise and its own parameters [19]. One more aspect to consider is that there is no ensured one-on-one mapping from the higher-dimensional space to its projected lower-dimensional space [53]. This means for every new run LLE might produce different results.

2.2.2 t-Distributed Stochastic Neighbour Embedding

Like LLE, t-SNE is also an effective non-linear dimensionality reduction technique. It is often used to embed high-dimensional data in a lower-dimensional space to find clusters within the data by grouping similar data points closer. In its initial step, t-SNE defines pairwise similarities between each data point in the original higher-dimensional space. t-SNE then converts these similarities into conditional distributions such that closer points have higher probabilities. To transform the data to a lower-dimensional embedding space, t-SNE again computes pairwise similarities but with a different distribution. This different distribution is defined such that it matches best with the conditional probabilities from the higher-dimensional space. To find the best distribution that matches optimally the first probability distribution in the higher-dimensional space, t-SNE minimises the Kullback-Leibler divergence between these two distributions via gradient descent optimisation [4]. Similar to LLE, there are some aspects we have to acknowledge. t-SNE is sensitive to its hyperparameters, such as perplexity. Perplexity in t-SNE controls the balance between preserving the local and global structure. Therefore selecting hyperparameters can be a challenging task. Additionally, t-SNE is stochastic in nature, which means for every new run t-SNE may produce different results, even with the same parameters and input data [12].

2.2.3 Uniform Manifold Approximation and Projection

Uniform Manifold Approximation and Projection (UMAP) assumes the data points to be uniformly distributed and gives similar results to t-SNE but with more respect to the global structure of the source data. The UMAP algorithm consists of two main steps: construction of a weighted k -neighbour graph and computation of a representation of the data in the lower-dimensional embedding. To construct the specific weighted k -neighbour graph, UMAP finds the k -nearest neighbours under a predefined metric d . Then a weighted directed graph is defined such that its vertices are the data points and its edges are the prior computed k -nearest neighbours with a specific weight dependent on the distance between the considered points. UMAP then optimises the embedding of data points into a lower-dimensional space with a mathematical framework based on Riemannian geometry. As UMAP uses the stochastic k -nearest neighbour algorithm as well as the stochastic gradient descent algorithm, UMAP is not deterministic. Unlike for PCA, where the resulting dimensions represent the directions of the highest variance in the data, the dimensions in the lower-dimensional space lack specific meaning for the result of UMAP [32]. Thus, the interpretability of its results may be challenging.

2.2.4 Random Projections

Similar to PCA, random projection also uses linear projections to decrease the number of dimensions of high-dimensional data. In contrast to PCA however, the transformation matrix is not calculated by an eigenvalue decomposition, but rather chosen randomly with columns having unit-length, meaning each column has the length 1. Random projection is designed such that they approximately preserve certain pairwise distances between data points, despite their random nature. As the transformation matrix is chosen randomly, random projection is not deterministic resulting in different embeddings for the same data for multiple runs [9].

2.2.5 Locality Preserving Projections

The primary goal of Locality Preserving Projection (LPP) is to retain the local relationships or neighbourhood information among the data points. It begins by computing an affinity matrix, where each entry denotes the similarity between data points. This matrix quantifies how similar or dissimilar data points are to each other based on a chosen metric. Often the Euclidean distance or cosine similarity is used to calculate the similarities. The entries of the affinity matrix are then weighted by using heat kernels, resulting in the matrix W . LPP then constructs the graph Laplacian matrix \mathcal{L} from the affinity matrix

and then identifies a set of eigenvalues and their corresponding eigenvectors framework \mathcal{L} . The Laplacian matrix describes the graph's topology and connectivity by representing the difference between the diagonal matrix D and W , where D contains the sums of columns from X . Subsequently, the eigenvectors associated with the smallest eigenvalues are selected and used to project the dataset into a lower-dimensional space while preserving the local relationships [15].

2.3 Dimensionality Reduction on Ordered Data

This section focuses on dimensionality reduction methods especially designed to work on data which is ordered.

2.3.1 Time-Lagged Independent Component Analysis

A fundamental aspect of ordered data are time series. Time series data often possess unique characteristics and challenges that necessitate specialised dimensionality reduction techniques.

One of these methods is time-lagged ICA (tICA). tICA extends independent component analysis (ICA) to capture essential temporal patterns and slow dynamics in time series [34].

ICA, a widely used technique, assumes that observed data can be represented as linear combinations of hidden sources, and its primary objective is to find these underlying sources [47]. A key assumption of ICA is that these sources are statistically independent from one another, meaning that the probability distribution of one source should not depend on the probability distribution of another. ICA can be viewed as a dimensionality reduction method, as it aims to disentangle the distinct sources of variation within the data and by keeping only the sources with the highest amount of information dimensions are reduced. Essentially it aims to unmix the underlying sources of variation in the data. This is especially useful for scenarios where the observed data are mixed and overlapped, such as audio signals, images or biomedical data [47].

Similarly to ICA, tICA focuses on identifying independent components that represent meaningful temporal variations in the data. It does this by removing overall drift or translation of the data, by centring the data. Then it calculates the lagged covariance matrix of the centred data. The lag parameter determines how many time steps are considered when computing the time-lagged covariance. On this time-lagged covariance matrix, tICA performs an eigenvalue decomposition. As a dimensionality reduction method, tICA projects

the source data onto the dominant independent components, where the corresponding eigenvalues indicate the slowness of their respective process [34].

2.3.2 Linear Sequence Discriminant Analysis

Linear Sequence Discriminant Analysis (LSDA) aims to capture the optimal embedding of the high-dimensional sequential data in a lower-dimensional embedding, such that the separability of the sequence classes is maximised. Sequences are time series or multiple observations with an inherent order. LSDA can be seen as a more general version of LDA and can be described by the following steps: LSDA first computes statistical quantities like the mean sequence and the variance matrix to summarise and capture its essential characteristics. Thus it is assumed, that these first-order and second-order statistics explain a crucial part of differentiating the sequence classes. Once these statistics are computed, LSDA aims to reduce the dimensionality of the data. It does this by finding a linear transformation that maximises the separability between each different class of sequences. As different statistics can be used, there are various variants developed. However, this method aims to find a way to transform the data so that different sequences within the data become as distinct from each other as possible [48].

To our knowledge, there exists no unsupervised method for explicitly maintaining the order-based distances in ordered data.

Chapter 3

Dimensionality Reduction on Ordered Data

In this chapter, we define the nature of the data we work with in this thesis. We then explore three different preservation aspects of dimensionality reduction methods: local structure, global structure and order-based distance preservation. These focuses play a crucial role in dimensionality reduction methods as they help to obtain meaningful insights from data by preserving specific information while reducing dimensionality. Subsequently, we examine four variants of PCA: PCA with Ids and three different extensions of PCA using complex space. Finally, we propose PCA*, which extends PCAs functionality to sequential data.

3.1 Ordered Data

In the following, we explore the characteristics of ordered data. First, we define what we mean with ordered data and then look at examples like time series data. These highlight the importance of maintaining the inherent order of data while reducing dimensionality.

The following definition for sequential data is based on [6].

Definition 3.1.1 (Sequential Data). *Assume n data points $X = (x_1, \dots, x_n)$ in $\mathbb{R}^{n \times d}$. The dataset X is ordered sequentially if each element in X is associated with an index or position within the sequence. Formally, there exists a strict total order for all $x_i, x_j \in X$: $x_i <_{\text{order}} x_j$, for $i < j$ and $i, j \in \{1, \dots, n\}$.*

In the following, we refer to sequential data also as sequences.

Often, we encounter observations, where multiple sequences are concurrently detected. For instance, in financial markets, we may consider concurrent sequences of stock prices, trading volumes and bond yields, all originating

from the same trading session. Consequently, we regard not only individual sequences but also sets of multiple sequences:

Definition 3.1.2 (Set of Sequences). *A set of sequences is a collection of sequences, each consisting of more than one data point. Each of these sequences has the same dimensionality and the same number of elements*

One of the most prominent categories of sequences and sets of sequences are among others time series. Time series are a set of data points which are observed chronologically. They are used to analyse trends, patterns and behaviours that happen over time and can be found in various fields, like economics, finance, engineering, meteorology and more [36]. An example of time series are sensor data streams, as each measurement is taken at a specific moment. The order is important to understand how the measured entity changes over time. As we can have many sensors running at longer times these kinds of data streams can generate a high volume of high-dimensional data.

On the other hand, there are also data of which the order depends essentially on its geographic location, distances to other points [18] and relations between objects [13]. Biological and chemical sequences like chemical compounds [5] and genomic data like DNA-sequences [3] can be, due to their inherent structure and the way information is encoded represented as graphs [3]. A fundamental aspect of chemical compounds is their molecular and chemical structure because it describes how atoms are bonded together. For instance, isomers are molecules with the same number of atoms of each element but with a different arrangement or different spatial orientations. Isomers play a distinct role in chemistry, as their arrangement of atoms often leads to different chemical and physical properties [41]. In this context, our focus is on data conforming to the above definitions as either sequential or sets of sequences. Therefore, we need to ensure that the data is either totally ordered or totally ordered in one sequence.

There are also data displaying both time series and spatial order. Molecular dynamics, for example, are over time simulated trajectories of atoms and molecules where sequential positions and interactions are critical for understanding molecular behaviour [6]. However, we do not consider this type of data, as our focus is primarily on sequential data defined above and leave it to future research.

3.2 Structure and Order Preservation

When reducing dimensionality, it's important to acknowledge some level of information loss. This loss occurs when simplifying the representation of com-

plex data to a compressed lower-dimensional version. As a consequence, fine-grained relationships or nuances in the original data may not be fully kept [28]. There are different ways this information loss can appear and it is often assessed through various preservation metrics. We can categorise dimensionality reduction methods by the quality of their local or global structure preservation. In this section, we define both properties and provide arguments for their significance in ensuring meaningful data representations in lower-dimensional spaces as well as in their reconstructed space if possible. Subsequently, we define a new focus of dimensionality reduction methods: order-based distance preservation. The importance of this property is shown by an example of genetic mutations and the ordering of news.

We define the set of k -nearest neighbours (k NN) rooting on [14], which is often used in fields like regression, classification and data analysis [16].

Definition 3.2.1 (Set of k -Nearest Neighbours). *Let X be a dataset consisting of n datapoints (x_1, \dots, x_n) . The set of the k -nearest neighbours of a datapoint $x_i \in X$ is defined as the set of data points in X which have the k smallest distances to x_i , in terms of a distance metric.*

With the set of k -nearest neighbours defined, we define local structure preservation for dimensionality reduction techniques based on [51] and [45].

Definition 3.2.2 (Local Structure Preservation). *Let $X \in \mathbb{R}^{n \times d}$ represent the data in the original high-dimensional space, where n is the number of samples and d is the feature dimensionality. Assume $Y \in \mathbb{R}^{n \times l}$ being the data obtained after applying a dimensionality reduction method to X , where l is the reduced number of features, with $l < d$. We say the dimensionality reduction method preserves local structure, if it primarily maintains the relationships of the k -nearest neighbours for each point when reducing to lower dimensions, that is*

$$\forall x_i \in X \ \forall x_j \in N_k(x_i) : \|x_i - x_j\| \approx \|y_i - y_j\|,$$

where $N_k(x_i)$ denotes the set of k -nearest neighbours of x , $\|\cdot\|$ the distance measured applied and $\cdot \approx \cdot$ defines the approximation relationship between two values.

Dimensionality reduction methods focusing on local structure preservation aim to capture the local structure or local relationships among pairwise similar data points. They ensure that nearby data points in the higher-dimensional space should also be near each other in the transformed lower-dimensional space. The closeness of two data points is typically measured by an appropriate distance or similarity metric. As we see in often used dimensionality reduction methods like LLE [19] and t-SNE [4] the number of nearest neighbours that

is used is variable and should be chosen depending on the data. If we extend the local structure preservation to all data points, we capture relationships on a broader scale. Similarly to local structure preservation, global structure preservation is a common concept in literature for dimensionality reduction methods, we base our definition on [45] and [52].

Definition 3.2.3 (Global Structure Preservation). *Let X be the original high-dimensional data in $\mathbb{R}^{n \times d}$, where n is the number of samples and d is the number of features. Let Y be the result of applying a dimensionality reduction method on X , where each data point y_i is a l -dimensional vector with $l < d$. Global structure preservation ensures that pairwise distances between data points are preserved in the reduced space. In other words, for any data points x_i and x_j in X and their respective lower-dimensional representations y_i and y_j in Y follows:*

$$\forall i, j \in \{1, \dots, n\} : \|x_i - x_j\| \approx \|y_i - y_j\|,$$

where $\cdot \approx \cdot$ denotes the approximation between two values.

Dimensionality reduction methods preserving global structure aim at maintaining the overall distance from every data point to every other data point in the lower-dimensional space.

As a result, dimensionality reduction methods having both local and global structure-preserving properties, offer numerous advantages. As mentioned above, the local structure-preserving property ensures that the fine-grained relationships within the data are captured in the lower-dimensional space. Clusters, neighbourhoods or other local structures are therefore precisely represented. On the other hand, dimensionality reduction methods with a high degree of global structure preservation maintain the overall relationships, dependencies and coherence of the data. Thus the underlying data distribution is retained which helps with interpretability and visualisation. Therefore combining both properties in a dimensionality reduction method enhances the representation of local details along with the global structure.

Figure 3.1 shows the result of different dimensionality reduction methods on the same dataset. This is an often-used example to visualise global and local structure preservation. Here, the data is 3-dimensional and forms a S-shape. As PCA reduces dimensionality by aiming at linear projections which maximise the global variance of the transformed data, it only considers the global structure but not the local structure [52] [46]. Therefore the starting of the S is squashed with the middle part of the shape. On the other hand, we see that LLE preserves the local structure, as only the nearest neighbours are close to each other in the transformation, while the distances to other data points are ignored.

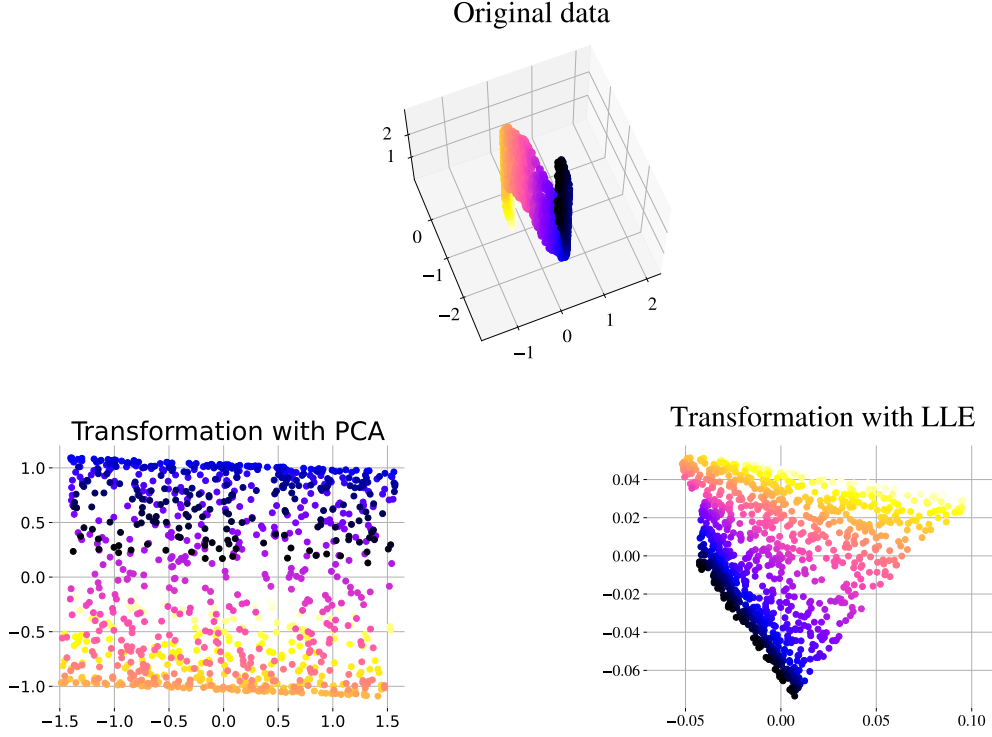


Figure 3.1: Transformation results with different dimensionality reduction methods

Additionally to the above-mentioned structure preservations, we propose a new focus for dimensionality reduction methods. For instance for ordered data, like time series, it may be of importance that the distance between each data point and its sequential data point is maintained in the projected space.

Definition 3.2.4 (Order-Based Distance Preservation). *Let $X = (x_1, \dots, x_n)$ be an ordered dataset as sequences or set of sequences, see Section 3.1. Furthermore, let $Y = (y_1, \dots, y_n)$ be its lower-dimensional projection by a dimensionality reduction method. We say the dimensionality reduction method aims to preserve the order-based distances if for every data-pair (x_i, x_{i+1}) for $i \in \{1, \dots, n\}$ in the ordered data-set X , its respective lower-dimensional data-pair (y_i, y_{i+1}) has approximately the same distance in terms of a distance metric.*

Dimensionality reduction methods focusing on preserving order-based distances ensure that distances between each consecutive pair of data points in a sequence are similar in the original higher-dimensional and lower-dimensional space. By preserving the distances in the order, the method can accurately

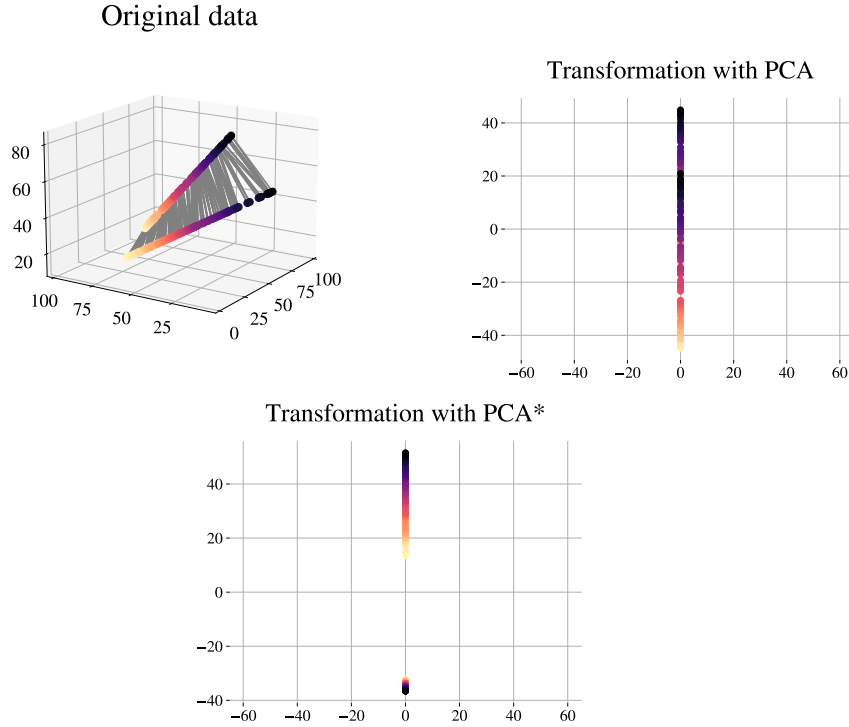


Figure 3.2: Transformation results with PCA and distance-order preserving variant PCA*

capture sequential relationships. This property is especially interesting for ordered data, where the order holds essential information like time series data. This may improve the overall interpretability of the result as we assume that variables that are far away in an order are also less dependent in contrast to those which are close in the order. See for example Figure 3.2. The data points appear in two lines and are ordered in a zigzag form, meaning every next data point lies on the opposite line. Transforming this data with PCA to a one-dimensional space, results in two lines but we see that the data points in the lines are not exactly as coloured as in the original space, whereas for the method PCA* they are. PCA* is a variant of PCA, which focuses on preserving order-based distances. For more information about PCA*, please refer to Section 3.3.3.

One real-world example can be seen in genetic mutations which when appearing close to each other on a chromosome have a higher likelihood to be co-inherited compared to mutations which are positioned at a greater distance from each other [54]. Another example was shown by Advani, Papotti and Asudeh [1] by demonstrating that among many other aspects, the order of

how news appear on a web page or broadcasting service, may affect the readers' opinions about current events or other topics. This can be manipulated in such a way that bias can be incorporated even though the correctness of the information is secured [21].

It is also important to note that achieving all three properties simultaneously can be challenging, as they sometimes conflict with each other. For instance, retaining the local structure in a time series with rapid and large fluctuations may lead to distortions in the temporal order of the data points. In other words, data points that were originally adjacent in time might have a different relationship with each other in the lower-dimensional representation to better satisfy the global distance preservation.

3.3 Variants of Principal Component Analysis

With the motivation to find a dimensionality reduction method, which preserves order-based distances we chose PCA as our fundamental dimensionality reduction method due to its computational efficiency, deterministic results as well as its common usage in several fields [33]. Thus, to incorporate sequential information into the dimensionality reduction technique we explore the following PCA variants: PCA with Ids, Complex PCA and PCA*.

3.3.1 PCA with Ids

To answer the question of how we can insert structural information into the data to keep this information in the lower-dimensional representation, a common suggestion is, to enumerate the data points. Assume X to be the data conforming to the definition 3.1.1 or definition 3.1.2. To enumerate the data points, we add a new dimension with its enumerated ID value to each data point in X :

$$X = \begin{pmatrix} x_{1,1} & \dots & x_{1,d} \\ \vdots & \ddots & \vdots \\ x_{n,1} & \dots & x_{n,d} \end{pmatrix} \rightarrow X_{\text{id}} = \begin{pmatrix} x_{1,1} & \dots & x_{1,d} & 1 \\ \vdots & \ddots & \vdots & \vdots \\ x_{n,1} & \dots & x_{n,d} & n \end{pmatrix}.$$

There are two distinct approaches for the application of PCA to X_{id} . The first one is applying PCA on X to compute the principal components, and then applying them to X_{id} on only the first d entries to reduce the dimensionality. This essentially gives the same representation to PCA with one dimensionality more, which contains the Id values. When plotting the data, we have to either remove the dimensionality containing the enumeration and treat it as metadata, or if not, we have to plot a dimensionality more. The other option

is to apply PCA to the X_{id} as if the new extended dimension were a normal feature of the dataset. In the subsequent, we explore some notable points why we have to be cautious when applying PCA on X_{id} to preserve order-based distances.

PCA on Discrete Data One assumption of PCA is that data follows a multivariate normal distribution, or at least, it operates under the assumption that the data is approximately normally distributed. Discrete data, on the contrary, follows non-normal probability distributions [30]. Therefore PCA treats the values in the Id dimension as numerical and may therefore attempt to find principal components that maximise variance in this dimension, which leads us to the next consequence.

Distortion of Order Distances As PCA aims to capture the most significant sources of variance, the presence of the enumeration dimension may lead PCA to emphasise variation that is unrelated to the order-based distances.

Loss of Interpretability The resulting principal components may not be easily interpretable, as the added dimension may dominate the principal components. Therefore, extracting meaningful insights about the original order relationships is made difficult.

3.3.2 Complex PCA on Ordered Data

To address the challenge of order-based distance preservation in PCA, we augment the data with complex vectors derived from consecutive data points in the sequentially ordered data set. Here we incorporate an additional layer of information that captures the sequential dependencies within the data in the complex part. We start by computing for each data point the vector to the next data point in the sequence. By adding the vectors as the complex part of the data, we aim to encode the sequential order effectively in the data itself. Therefore, the real part of the augmented data retains the original data and the complex part denotes the inter-point relationships. Applying principal components to the augmented data might thus lead to reduced information loss as the structure of the data is considered.

As described above, to incorporate the structure into the data, we exploit the complex space by adding imaginary values indicating the direction of the

next data point:

$$X = \begin{pmatrix} x_{1,1} & \dots & x_{1,d} \\ \vdots & \ddots & \vdots \\ x_{n,1} & \dots & x_{n,d} \end{pmatrix} \rightarrow$$

$$X_{\text{complex}} = \begin{pmatrix} x_{1,1} + ic_{1,1} & \dots & x_{1,d-1} + ic_{1,d-1} & x_{1,d} \\ \vdots & & \vdots & \vdots \\ x_{n,1} + ic_{n,1} & \dots & x_{n,d-1} + ic_{n,d-1} & x_{n,d} \end{pmatrix},$$

where each row of X is a data point in \mathbb{R}^d and $c_{k,l} = x_{k+1,l} - x_{k,l}$ for $k \in \{1, \dots, n\}$ and $l \in \{1, \dots, d-1\}$. When reducing the dimensionality of X_{complex} with PCA, we first have to centre the data. If we centre the data, not considering the complex part as vectors, the vectors of the centred X_{complex} do not have the correct length to indicate the position of the next point:

Example Assume the data to consist of three one-dimensional data points:

$$\begin{array}{ll} x_1 = 1 & x_{1,\text{comp.}} = 1 + 2i \\ x_2 = 3 & \rightarrow x_{2,\text{comp.}} = 3 + 4i \\ x_3 = 7 & x_{3,\text{comp.}} = 7. \end{array}$$

Centring yields:

$$\begin{aligned} x_{1,\text{comp.,centr.}} &= -\frac{8}{3} \\ x_{2,\text{comp.,centr.}} &= -\frac{2}{3} + 2i \\ x_{3,\text{comp.,centr.}} &= \frac{10}{3} - 2i. \end{aligned}$$

Thus we see, that the centred complex vectors do not have the correct length to accurately point to the next data point. Additionally, after centring, it becomes unclear where the sequence ends, since the last data point retains an imaginary part. This might be challenging when datasets consist of more than one sequence.

As our primary motivation is to enhance the information of the data by incorporating the sequential vector data in the complex part, centring hinders this approach. Hence we assume to only centre the data on the real part but leave the complex part. This partial centring leads to the question if the result of complex PCA (CPCA) is still usable. In [2] it is stated that centring data in SVD is optional and that not centring data minimises the influence

of a cluster's variance on another cluster. Cadima and Jolliffe [11] show that the transformed columns of the d -dimensional data $X_{\text{complex}} = X_{\text{real}} + iX_{\text{imag}}$ can be correlated and emphasise the difference of the principal components of centred and uncentred data: Let $X_{\text{complex}}\nu_i$ and $X_{\text{complex}}\nu_j$ be transformed data, with ν_k being the k -th principal component of its respective eigenvalue λ_k and $i, j, k \in \{1, \dots, d\}$, then:

$$\text{Cov}(X_{\text{complex}}\nu_i, X_{\text{complex}}\nu_j) = \nu_i^T C \nu_j = \lambda_i \nu_i^T \nu_j - (\nu_i^T c_m)(\nu_j^T c_m),$$

where C is the covariance matrix of X_{complex} and $c_m = \frac{1}{n} \text{Im}(X_{\text{imag}})^T \mathbb{1}_n$. This formula we already fit to our case with only the imaginary part not being centred. We see in the equation, that if the principal components are uncorrelated depends on how far away the connected data points are. Due to the resulting components potentially being correlated, they are generally dependent if the variables are Gaussian distributed. As PCA is not suitable for dependent or correlated data, the result of this dimensionality reduction method might not be mathematically or statistically significant [22]. On the other hand, Jolliffe states in [26] that uncentred PCA is a well-established method in chemistry, ecology and geology. They also mention that uncentred PCA performs eigenanalysis on matrices that are not covariance or correlation matrices but still can be seen as measures of similarity.

Apart from the above-shown consequences of partially centring data, applying PCA to X_{complex} introduces some other consequences. Applying PCA to complex-valued data results in complex principal components which then transform into a lower-dimensional complex space, introducing a level of intricacies and the consideration of how the real and imaginary parts are interdependent, as these components are intertwined, especially during the process of finding the principal components of the covariance matrix $\text{Cov}(X_{\text{complex}})$:

$$\text{Cov}(X_{\text{complex}}) \neq \text{Cov}(X_{\text{real}}) + \text{Cov}(X_{\text{imag}}),$$

where $X_{\text{complex}} = X_{\text{real}} + iX_{\text{imag}}$.

In the following, we concentrate on two main variants of CPCA, as these maintain correctly scaled vectors directing to the next consecutive data point, which can be seen as metadata:

CPCA-real: After obtaining the complex principal components, we choose to focus only on the real part.

CPCA-imag: To transform the complex data, we only use the imaginary part of the principal components.

Unlike PCA on real-valued data where the principal components are unique up to the sign, the principal components for PCA on complex data, or short

Complex PCA (CPCA), are unique up to a multiplication by a unit phase factor $e^{i\varphi}$. In the case of PCA, there are some commonly used options on how to ensure deterministic output: One is the positive sign approach. This simple and commonly used approach aligns principal components such that their signs are positive. For example, we scale the vector in such a way, such that its maximum element is positive [10]. Another option is the largest-absolute-value-based approach. This method ensures that the largest absolute value within each component is positive. This is also the approach we are using as `scikit` conforms to this as well, see [10] and [42].

The importance of choosing one convention to solve the sign arbitrariness can be seen in the comparison of results of multiple runs of PCA, see [10].

As the complex principal components are unique up to a scaling factor $e^{i\varphi}$, there are multiple ways to solve this ambiguity. In the following, we explore two options:

- **Align to PCA** To promote stability in the outcome of CPCA, we scale the complex principal components such that they are as close as possible to the principal components derived from applying PCA to the original data. The motivation behind this strategy is based on the desire to maintain the essence of the original data while leveraging the additional insights offered by the complex information. By aligning with the original principal components, we effectively 'ground' the complex principal components in a familiar, interpretable context, making them more accessible. Therefore, this alignment facilitates the interpretation of the result of CPCA in the context of the original data.

As the transformation with the resulting principal components of CPCA is complex, we may obtain the real part of the lower-dimensional representation. Therefore, to not lose as much information, we explore the second option:

- **Align to smallest imaginary part** By minimising the magnitude of the complex parts of the principal components, we effectively 'dampen' the influence of the phase information. This reduction in the complex part's magnitude aligns complex principal components more closely to their real counterparts. Therefore it promotes a deterministic and stable output, facilitating comparisons across different runs of complex PCA.

In the following theorem we show that phase-shifting a complex vector may not go as far as producing the exact same vector it has been aligned to:

Theorem 3.3.1 (Unit Phase Factor Alignment). *The real part of a complex-valued vector cannot take any arbitrary value solely by unit phase factor alignment.*

Proof. Assume x to be a complex vector. Let $\hat{x} = e^{i\varphi}x$ be the aligned vector obtained by aligning x with a unit phase factor $e^{i\varphi}$. The real part of the aligned vector is given by $\text{Re}(\hat{x}) = \text{Re}(x) \cos(\varphi) - \text{Im}(x) \sin(\varphi)$ by using Euler's formula. Now assume there exists $Z \in \mathbb{R}$ arbitrarily chosen such that $\text{Re}(\hat{x}) = Z$. With substituting this for $\text{Re}(\hat{x})$, follows $\text{Re}(x) \cos(\varphi) - \text{Im}(x) \sin(\varphi) = Z$. This equation cannot be directly solved alone for φ as it is also dependent on $\text{Im}(x)$, $\cos(\varphi)$ and $\sin(\varphi)$. To prove this assume $x = 1$. Above equation is now $\cos(\varphi) = z$. Since $\cos(\varphi)$ is bounded between -1 and 1 , z can only take values between -1 and 1 . Now consider $x = i$ which is purely complex. This makes above equation $\text{Re}(0) \cos(\varphi) - \text{Im}(i) \sin(\varphi) = Z$. Therefore $-\sin(\varphi) = Z$. As $\sin(\varphi)$ is bounded, Z cannot be arbitrarily chosen in \mathbb{R} . \square

In summary, we see that incorporating the sequential information into the data as the complex part to maximise the degree of how well PCA maintains the order-based distances within the data, has some disadvantages, like resulting dependence in data, phase aligning and poorer interpretability.

3.3.3 PCA*

This chapter introduces a variant of Principal Component Analysis that leverages the concept of vector differences between consecutive points in the dataset, allowing for the analysis of sequential changes and trends in the data. Principal Component Analysis for Structure-Aware Representation (PCA*) focuses essentially on the connections of each pair of consecutive data points in the sequentially ordered dataset.

Definition 3.3.2 (PCA*). *Assume a high-dimensional dataset $X \subset \mathbb{R}^{n \times d}$ consisting of n d -dimensional data points $\{x_1, \dots, x_n\}$. To transform the data with PCA* following steps have to be taken:*

1. *Data Centering: Data centring is done by subtracting the mean of each column from the corresponding column of the data matrix X . Let μ_X be the set of all means in X and X_{centred} be the computed centred dataset. This step is important as without it the result may be shifted, see the general case in Theorem 3.3.3.*
2. *Vector Computation: Compute the connections of each consecutive pair of data points in X . Let $V = \{v_1, \dots, v_{n-1}\}$ be the set of all these vector differences, with*

$$v_i = x_{i+1} - x_i \quad \forall i \in \{1, \dots, n-1\}.$$

3. *Vector Centring:* As we have seen in the previous subsection, centring data before applying SVD is crucial. Therefore, V needs to be centred similarly to step 1. Denote V_{centred} as the centred V .
4. *SVD Decomposition:* Next perform Singular Value Decomposition (SVD) on V_{centred} . SVD decomposes V_{centred} into three matrices:

$$V_{\text{centred}} = U\Sigma Z^T,$$

where

- U contains the left singular vectors
 - Σ is a diagonal matrix containing singular values
 - Z^T is a transposed matrix consisting of right singular vectors.
5. *Selecting Principal Components:* We sort the singular values in Σ in descending order. The principal components corresponding to the largest singular values capture the most variance in the data V . Thus, we choose the top k principal components, where k denotes the desired number of dimensions for the lower-dimensional space. Let PC_k be the matrix containing the chosen principal components as columns.
 6. *Projection:* Using the principal components from the step earlier, we project X_{centred} into the lower-dimensional space:

$$X_{\text{transf}} = X_{\text{centred}}PC_k.$$

7. *Reconstruction [Optional]:* We reconstruct the lower-dimensional representation of X back into the original high-dimensional space:

$$X_{\text{reconstr}} = X_{\text{transf}}PC_k^T + \mu_X,$$

where PC_k^T denotes the transposed matrix PC_k and μ_X be the column-wise mean of X of step 1.

By focusing on the connections between sequential data points and by applying PCA to them, PCA* aims to capture the most important directional changes in the data while reducing dimensionality.

Theorem 3.3.3. *Let X be the original high-dimensional source data and let PC_k be the top k principal components corresponding to the largest eigenvalues of the SVD of X_{centred} , where X_{centred} denotes the centred data X , like in the Definition 3.3.2 of PCA* in step 1. Not centring the original data when applying the principal components PC_k may result in shifted reconstructed data for PCA and PCA*.*

Proof. Assume X to be the source data. We now want to show that if we compute PCA on the centred data X_{centred} and then apply the principal components to X , the resulting data is shifted. Suppose $PC = \{pc_1, \dots, pc_d\}$ are the principal components calculated by applying PCA to the centred data $X_{\text{centred}} = X - \mu_X$, where μ_X the column-wise mean denotes. We now transform both X and X_{centred} with a subset of PC of vectors corresponding to the biggest eigenvalues PC_k , where k denotes the number of dimensions of the lower-dimensional space. Originally PCA applies the principal components onto X_{centred} :

$$X_{\text{centred,transf}} = X_{\text{centred}}PC_k.$$

We now want to check applying the principal components on X :

$$X_{\text{transf}} = XPC_k.$$

We now can substitute X with X_{centred} :

$$X_{\text{transf}} = XPC_k = (X_{\text{centred}} + \mu_X)PC_k.$$

Therefore the transformation does not eliminate the mean contribution from the data and the resulting transformed data may not have a mean of zero. When reconstructing the data we see the shift even more:

$$\begin{aligned} X_{\text{reconstr}} &= X_{\text{transf}}PC_k^T = (X_{\text{centred}} + \mu_X)PC_kPC_k^T \\ &= X_{\text{centred}}PC_kPC_k^T + \mu_XPC_kPC_k^T \end{aligned}$$

whereas for the centred data follows:

$$X_{\text{centr,reconstr}} = X_{\text{centr,transf}}PC_k^T = X_{\text{centred}}PC_kPC_k^T,$$

where then PCA adds μ_X to shift the reconstructed data to the centre of X , in contrast to the uncentred case where $\mu_XPC_kPC_k^T$ is added but this is not the correct centre of X . Thus the reconstruction of the uncentred data appears shifted. \square

As PCA and PCA* might look similar we now explore their main differences:

1. Data Representation:

- PCA: PCA operates on tabular data, where each row represents an observation and each column a feature. It aims to find orthogonal linear combinations of the original features to project the data into a new lower-dimensional space while still maximising the variance.

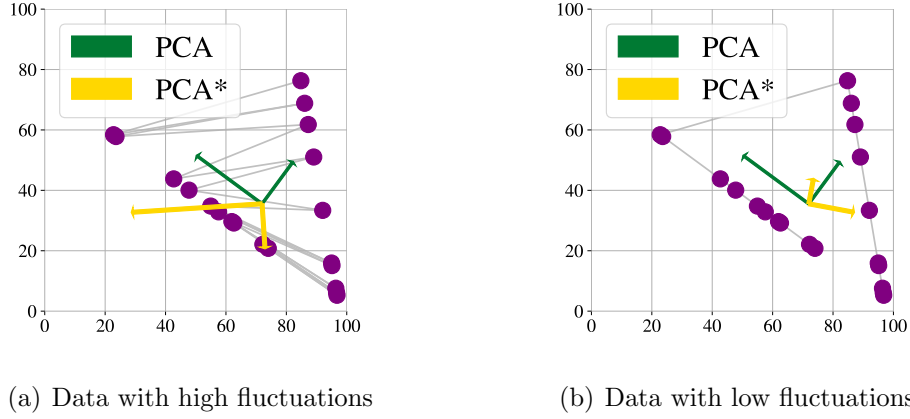


Figure 3.3: Principal components for PCA and PCA* on differently ordered datasets

- PCA*: PCA* is specifically designed for sequential data, where each data point is a vector observed over time or multiple sequences of vectors observed over time. It focuses on preserving the sequential relationships within the data while reducing dimensionality.

2. Dimensionality Reduction Goal:

- PCA: PCA aims to reduce the dimensionality of data by finding a new set of orthogonal axes that capture the maximum variance. It assumes the data points to be independent and does not consider any inherent time or sequential order. PCA treats each data point as a separate entity.
- PCA*: On the other hand, the primary goal of PCA* is to reduce dimensionality while still preserving the temporal or sequential relationships within the data. To accomplish this PCA* looks for a lower-dimensional representation that maintains the underlying patterns and dependencies in the data. It therefore might happen that the same data just differently ordered results in different lower-dimensional representations, see Figure 3.3.

3. Applications:

- PCA: This is an often used dimensionality reduction method for various tasks, like feature selection, noise-reduction and visualisation of high-dimensional data.

- PCA*: PCA* is a specifically tailored algorithm for tasks like data analysis, where the sequential order of observations is crucial.

Figure 3.3 emphasises the different results of PCA* and PCA. While PCA finds for both orders the same principal components, PCA*'s principal components depend on the order. Thus it provides a more flexible approach to dimensionality reduction, as it acknowledges that different orders and sequences within the data may reveal distinct patterns.

To summarise: PCA is a general-purpose dimensionality reduction method, whereas PCA* focuses on sequential data analysis, enhancing the quality of sequential dependencies within the data points.

Chapter 4

Experiments

In this chapter, we present a series of experiments to evaluate PCA* and the CPCA variants. We begin by introducing the data used for these experiments, which include both synthetic, generated data and real-world data such as stock closing prices. To assess the results, we use quality measures outlined in the second section of this chapter. While there are several variants of PCA discussed in Section 3.3, our primary focus in this section lies on PCA*. However, we also explore some of the primary experiment settings involving these CPCA variants. The measures are used to compare PCA* to some of the most often used dimensionality reduction methods described in Chapter 2, namely PCA, LLE, UMAP, t-SNE.

4.1 Datasets

To evaluate the dimensionality reduction methods, we explore multiple generated and real-world datasets. We first describe how we generate data and subsequently examine datasets of stock closing prices, air pollution in Japan and generated flight data (MTS).

Synthetic Data These data we constructed to assess and compare the performance of PCA*, Id PCA and the CPCA variants against other common dimensionality reduction methods on data where we have control over their complexity and order. The data generation process involves constructing data points that lie consecutively on l lines within a high-dimensional space. These lines are then sorted, so that the starting and ending points of close lines are near in proximity. Each line consists of an equal number of data points. To obtain sequentially ordered data, as defined in Section 3.1, we shuffle the data points to resemble specific orderings. These can be categorised as zigzag and smoothly ordered data as well as data ordered in separate lines.

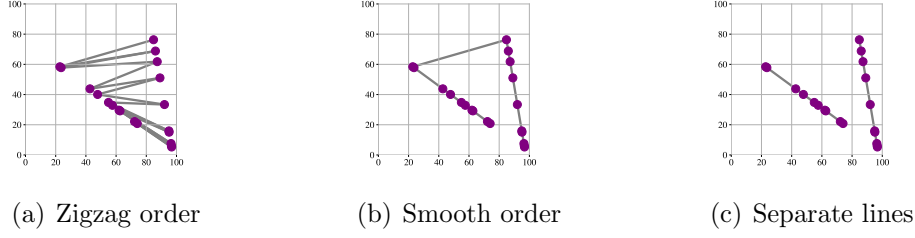


Figure 4.1: Figures show two-dimensional data generated in two lines.

If the data is arranged in zigzag order, we order the data points by their position within the lines. Suppose a list of lines. We then collect iteratively the i th points of each line: we start by taking the first points of each line, then the second until all points are collected. This creates a zigzag pattern as seen in Figure 4.1(a).

However, if the order of the data is smooth, then we connect each separate line by a continuous line. If we desire the data to remain in separate lines, we refrain from connecting them. An example is given in Figure 3.1, where 4.1(a) how the data is ordered in a zigzag form, 4.1(b) shows the same data as one-line ordered data and finally 4.1(c) the data as separate lines. We selected these orderings to simulate different data behaviours. Zigzag-ordered data represent data which have rapid and high fluctuations, while smoothly-ordered data simulates datasets where consecutive data points change only gradually. To cover a range of scenarios, we explore data ordered in separate lines. All these data orders consist of the same data points, allowing us to evaluate the effect of different ordering patterns on the performance of various dimensionality reduction methods.

We also introduce a level of randomness by adding a random value within a range determined by a jitter-bound to each dimensionality in the generated data points. This introduces slight variations in the data, making it more representative of real-world scenarios where data may not be perfectly precise but can exhibit small fluctuations.

Simulated and Real-World Data As the synthetic data consists of data points laying on more or less straight lines, we also explore PCA* on bigger datasets, and compare the results with the outcomes with PCA, LLE, t-SNE and UMAP:

- **Russell2000** The Russell2000 data consists of daily adjusted close prices of 1794 stocks like the CCBG (Capital City Bank Group, Inc.) and HCT (Holistia Colltech Limited) for the time of January 1st, 2013 to

2014 [37]. For simplicity, we excluded stocks containing missing data and omitted the time dimension since the order of the data points is represented by their input sequence. Thus, our analysis is conducted on a 1597-dimensional dataset, with each dimension containing 504 numerical observations.

- **MTS** We utilise the data set sourced from NASA [8], containing 10 samples of flight data. The dataset, although valuable for our purposes, presents certain limitations. Specifically, apart from basic information such as data sizes and dimensions for each flight time series, no additional contextual details were provided. However, these limitations do not hinder our primary research objective, which focuses on testing and evaluating dimensionality reduction techniques with respect to order-based distance preservation. Each of the given flight data contains 30 dimensions of which one denotes the time, which we removed similarly to the Russell2000 data. The number of observations varies between 4660 and 4900. To make them all equally long, we shortened each time series to 4660 data points.
- **Air Pollution in Japan** This air pollution dataset consists of observations of the nationwide sensor network surveilling air pollution in Japan, called the Atmospheric Environmental Regional Observation System (AEROS), set up by the Ministry of Environment Japan. There are hourly recordings of more than a thousand sensors from 2018 to 2023. Missing values are substituted with zeroes as recommended by the authors. Overall the dataset contains 1832 dimensions [38]. In this thesis, we use the data from the first half year which contains 4380 observations.

As a result, these datasets exhibit diverse characteristics: One dataset features more dimensions than observations, another has low dimensions but a substantial number of observations and is ordered in separate flights, and the final dataset combines high dimensions with a large number of observations.

4.2 Quality Measures for Dimensionality Reduction Algorithms

In order to evaluate the effectiveness of dimensionality reduction techniques, we need some measures for datasets to quantify the dissimilarity between the transformed data and the original dataset. We first discuss two fundamental metrics Mean Absolute Error (MAE) and Root Mean Square Error (RMSE).

These metrics provide insights into the average distances between corresponding data points in two datasets. Based on these two metrics, we explore two new metrics which are used to assess the performance of dimensionality reduction methods. Finally, we explore the Dynamic-Time Warping Algorithm and how we can use it to compare patterns in two data sequences.

In this section, we assume data to be sequential if not other stated. The next definition is based on [23].

Definition 4.2.1 (Mean Absolute Error). *The mean absolute error (MAE) measures the average absolute distances between two equally large sets of data X and Y . It is computed by summing up the absolute distance between each dataset and then dividing it by the total number of values. More precisely:*

$$\text{MAE}(X, Y) = \frac{1}{n} \sum_{i=1}^n \|x_i - y_i\|,$$

where $\|\cdot\|$ is the L_2 -norm and for $x_i \in X$ and $y_i \in Y$ for $i \in \{1, \dots, n\}$. n is the number of observations.

The mean absolute error is a metric used to quantify the average magnitude of errors between two sets of values. In simpler terms, MAE measures the average distance between data points of X and Y , with no consideration of the direction of the errors. Lower MAE values indicate higher similarity as they imply a smaller average deviation between each observation. The following definition is based on [23].

Definition 4.2.2 (Root Mean Square Error). *The root mean square error (RMSE) is a measure to compute the distance of two datasets X and Y with n entries. It is calculated by taking the square root of the average of the squared distances between X and Y :*

$$\text{RMSE}(X, Y) = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - y_i)^2},$$

where $x_i \in X$ and $y_i \in Y$ for $i \in \{1, \dots, n\}$ and n is the number of observations.

RMSE measures the square root of the average of the squared errors, giving more weight to larger errors compared to MAE. Similarly to MAE, lower RMSE values indicate more similar datasets.

The next measure was proposed by Hajizadeh, Aghagolzadeh and Ezoji (2020) [20]. Unlike suggested in the paper, we do not scale with the proposed scaling factor to compare the higher- and lower-dimensional spaces,

as the lower-dimensional representation is directly projected onto the higher-dimensional space without any change in the length of the data vectors, see following theorem:

Theorem 4.2.3. *Let X be a d -dimensional data with n data points and let $X_{\text{transf.}}$ be its PCA-transformed lower-dimensional representation. Then points in the transformed dataset have the same length as their respective points in the reconstructed embedding.*

Proof. Assume $y = (y_1, \dots, y_k) \in \mathbb{R}^k$ to be the transformed version of a data point in X where $k \leq d$. We can rewrite the inverse-transformed y as follows:

$$\begin{aligned} (y_1, \dots, y_k) \begin{pmatrix} \nu_{1,1} & \dots & \nu_{1,d} \\ \vdots & \ddots & \vdots \\ \nu_{k,1} & \dots & \nu_{k,d} \end{pmatrix} &= \left(\sum_{i=1}^k y_i \nu_{i,1}, \dots, \sum_{i=1}^k y_i \nu_{i,d} \right) \\ &= \sum_{i=1}^k y_i (\nu_{i,1}, \dots, \nu_{i,d}) \\ &= \sum_{i=1}^k y_i \nu_i, \end{aligned}$$

where $\nu_i = (\nu_{i,1}, \dots, \nu_{i,d}) \in \mathbb{R}^d$ denote the principal components with $i \in \{1, \dots, k\}$. With this, we show that the reconstructed point has the same length:

$$\begin{aligned} \left\| \sum_{i=1}^k y_i \nu_i \right\|^2 &= \left\langle \sum_{i=1}^k y_i \nu_i, \sum_{i=1}^k y_i \nu_i \right\rangle \\ &= \sum_{i=1}^k \sum_{j=1}^k y_i y_j \underbrace{\langle \nu_i, \nu_j \rangle}_{\substack{=0, \text{ if } i \neq j \\ =1, \text{ if } i=j}} \\ &= \sum_{i=1}^k y_i^2 \\ &= \|(y_1, \dots, y_k)\|^2, \end{aligned}$$

where we used $\|\cdot\|^2 = \langle \cdot, \cdot \rangle$ with the scalar product $\langle \cdot, \cdot \rangle$. \square

In other words, if we initially had data in a 100-dimensional space and reduced it to a 2-dimensional representation using PCA, the reconstructed data points form a 2-dimensional plane in the 100-dimensional space.

Definition 4.2.4 (Neighbour-Distance-Measure). *Let $X = (x_1, \dots, x_n)$ and $Y = (y_1, \dots, y_n)$ be two datasets that may not have the same dimensions. The Neighbour-Distance-Measure is the RMSE of distances between each successive point in X subtracted by the respective distance by each point in Y :*

$$q_{\text{neigh-distance}}(X, Y) = \frac{1}{nk} \sum_{i=1}^n \sum_{j=1}^k \left| \|x_i - x_j\|^2 - \|y_i - y_j\|^2 \right|,$$

where $\|\cdot\|$ is the L_2 -norm and k is the number of nearest neighbours of a point x in X , see Definition 3.2.1.

As we are not operating on a dataset, but rather a data sequence we are not considering the k -nearest spatial neighbours but rather the k -nearest neighbours on the sequence.

The Neighbour-Distance-Measure quantifies how similar pairwise distances between data points in two datasets are. This is done by comparing the squared distances between successive points in dataset X to the corresponding squared distances in dataset Y . Therefore, it is important to note that this measure relies on the method of how the neighbouring data points are sampled. A lower value of the Neighbour-Distance-Measure indicates a higher similarity in the datasets.

As the Neighbour-Distance-Measure compares how the pairwise distances in each data set differ from each other, we define our second measure to evaluate the relation between each data point to its neighbouring point in both datasets.

Definition 4.2.5 (Neighbour-Relation-Measure). *Consider the dataset $X = (x_1, \dots, x_n)$ in d -dimensional space and its projection $Y = (y_1, \dots, y_n)$ in lower-dimensional space with k dimensions where $k < d$. Let $V = (v_1, \dots, v_{n-1})$ represent the vectors connecting each data point to its successive data point:*

$$v_i = x_{i+1} - x_i \quad \forall i \in \{1, \dots, n-1\}.$$

Let $U = (u_1, \dots, u_{n-1})$ be the corresponding list of vectors for the projected data Y . The neighbour-relation measure quantifies the similarity between the relationships of vectors V and U as follows:

$$q_{\text{scal-prod}}(X, Y) = \frac{1}{n-2} \sum_{i=1}^{n-2} |\langle v_i, v_{i+1} \rangle - \langle u_i, u_{i+1} \rangle|,$$

where $\langle \cdot \rangle$ denotes the standard scalar product.

As this measure uses the scalar product of the connections between each data pair in a sequence, we may call it the scalar-product measure as well.

Similarly to the neighbour-distance measure above we can extend the neighbour-relation measure to multiple neighbours, to see if the relation up to the next k neighbours stays the same.

Definition 4.2.6 (Multiple-Neighbour-Relation-Measure). *Let $X = (x_1, \dots, x_n)$ be the data in a d -dimensional space and $Y = (y_1, \dots, y_n)$ its dimensionality reduced or reconstructed representation. To measure the relation up to k neighbours, we sum the neighbour-relation measure up for every k -nearest neighbour on the sequence and take the mean:*

$$q_{\text{mult-scal-prod}}(X, Y) = \frac{1}{\sum_{l=1}^k (n - 2l)} \sum_{j=1}^k \sum_{i=j+1}^{n-j} |\langle v_{i,j}^-, v_{i,j}^+ \rangle - \langle u_{i,j}^-, u_{i,j}^+ \rangle|,$$

where $v_{i,j}^-$ represents the vector from the i th point to its j th predecessor, while $v_{i,j}^+$ represents the vector between the i th point to its j th successor. Similarly, $u_{i,j}^-$ and $u_{i,j}^+$ denote the corresponding vectors in the original data. Moreover, $k < n/2$ denotes the number of considered nearest data points on the order of X and $\langle \cdot, \cdot \rangle$ the standard scalar product.

In essence, this measure computes the MAE of the differences between the pairwise scalar products of the vectors. Therefore we not only compare the length of each connection between the data-pairs in the order of the data, but also quantify the angle between them. A lower error implies that the dimensionality reduction technique maintained the essential structure and patterns in the data, as the Scalar-Product-Measure takes the angle between each connection into account as well as both lengths of the vectors between the two data points.

As time series data play in many data science and data mining fields an important role, the need to compare time series increased. One often used time series similarity measure is the Dynamic Time Warping (DTW) algorithm. DTW was introduced by Berndt and Clifford in 1994 to find similarities in time series, as it was earlier used in the speech recognition field [7]. It aims to capture patterns and similarities in two sequences by minimising the cumulative distance along an

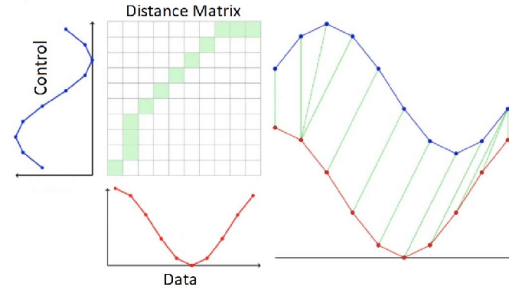


Figure 4.2: Dynamic Time Warping by Diab Mahmoud Diab, Basil AsSadhhan, Hamad Binsalleeh, S. Lambotharan, Konstantinos Kyriakopoulos, and Ibrahim Ghafir [17]

optimal alignment path that matches corresponding data points while keeping the influence of temporal shifting and distortion low. The next definition is sourced from [25] and [29].

Definition 4.2.7 (Dynamic Time Warping). *Let $X = (x_1, \dots, x_n)$ be a sequence of n d -dimensional data points and $Y = (y_1, \dots, y_m)$ be a second sequence of m d -dimensional data points. The first step to compute Dynamic Time Warping algorithm (DTW) is to calculate the $n \times m$ -dimensional distance matrix $\text{Dist}_{X,Y}$ of X and Y where the $(i^{\text{th}}, j^{\text{th}})$ entry of $\text{Dist}_{X,Y}$ equals the squared distance of x_i and y_j . $\text{Dist}_{X,Y}$ is also sometimes called path matrix as it is used to find the best warping path between these sequences which minimises the total cumulative distance between them. A warping path $P = (p_1, \dots, p_k)$, where $\max(m, n) \leq k \leq m + n - 1$ and p_i being an entry of $\text{Dist}_{X,Y}$, is a set of matrix entries defining a mapping between X and Y with following constraints:*

- *Endpoint constraint: the start- and endpoints of the warping path have to be the first and last data points of the path matrix, meaning $p_1 = (x_1, y_1)$ and $p_k = (x_n, y_m)$.*
- *Continuity constraint: The path cannot skip a step, i.e. the path takes step after step. This is when for the step $p_k = (x_i, y_j)$ and its following step $p_{k+1} = (x_{i+1}, y_{j+1})$ where $x_i - x_{i+1} \leq 1$ and $y_j - y_{j+1} \leq 1$.*
- *Monotonicity: the path is increasing, meaning the indices in the path move forward or stay the same while traversing the path.*

Let \mathcal{P} be the set of all warping paths P mapping between X and Y as described above. DTW aims to minimise the warping costs:

$$\text{DTW}(X, Y) = \min_{P \in \mathcal{P}} \left\{ \sum_{p \in P} p \right\}.$$

Two sequences are optimally matched with the lowest distance path.

DTW captures patterns and similarities in sequences which might have different lengths, speeds or temporal distortions.

In this work, we focus on the first two measures we explored: the Neighbour-Distance Measure and the Neighbour-Relation Measure, as these two are specifically created to quantify the order-based distance preservation of dimensionality reduction methods, whereas for DTW we want to maintain a reasonable level performance compared to other common dimensionality reduction methods.

4.3 Implementation Details

The following experiments were executed on a 64-bit ASUS computer with 7,6 GiB, Intel® Core™ i7-4720HQ CPU @ 2.60 GHz \times 8 with a disk capacity of 960,2 GB. Experiments in Subsections 4.5.1, 4.5.2, 4.5.4 and Section 4.6 were computed on Slurm, as these experiments took longer than 10 minutes each. Slurm operates on several Linux clusters. The machines connected to Slurm are the computers of the Institute of Computer Science of LMU with some exceptions. The experiments on SLURM took between half an hour and three days, with LLE being the method that most frequently required the longest processing time. Runtime-wise, PCA* is comparable to PCA.

We implemented PCA* with Python3, using `NUMPY` data structures and processed with `NUMPY` and `Pandas` operations. Principal components for PCA, PCA* as well as for the CPCA variants were computed using `Scipy`'s SVD function. To facilitate a comparison of PCA* results, we present the results alongside results from LLE, t-SNE, UMAP and random projection. For the latter three techniques, we made use of `Sklearn`'s implementations, while for UMAP, we used the implementation `umap-learn`[32].

To quantify the order-based distance preservation of the above-mentioned dimensionality reduction methods, we implemented the Neighbour-Distance measure using the distance matrix from `scipy`. As DTW is an often used measure, we used the `fastdtw` implementation [40]. Apart from that, we used overall `numpy` operations.

4.4 Experiments with CPCA on Synthetic Data

The following experiments aim to investigate the impact of the variants of CPCA and Id PCA in comparison with PCA on order-based distance preservation on sequential data. The first experiment assesses order-based distance preservation with varying dimensions to understand how these methods improve as more information is retained. The subsequent experiment transforms the data into two dimensions and examines the preservation of distances and relations with a varying number of predecessors and successors. The third experiment combines the first two, exploring dimensionality reduction and assessing distances and relations with an increasing number of neighbours. We measure order-based distance preservation using the Neighbour-Distance (ND) and relations with the Multiple-Neighbour-Relation (MNR), measure, comparing both transformations and reconstructions to the original data, see Definition 4.2.4 and 4.2.6. In this section, the nearest neighbours denote the data points which are the nearest in the order.

In the following experiments, we will call the CPCA variants:

- *CPCA real+scaled pca* We apply CPCA to the complex data and use only the real part of the resulting principal components. These components are phase-shifted to make them as similar as possible to the principal components of PCA.
- *CPCA-real+low imag* Similar to the the previous, we use the real part of the complex principal components but phase shift them to minimise their imaginary part.
- *CPCA-imag*: In this variant, we exclusively use the imaginary part of the complex principal components, scaling them to maximise their similarity to the principal components of real PCA.

Furthermore, we use the Id PCA variant, where we apply PCA to the whole dataset with Ids, as the other variant gives exactly the same results as PCA, see Section 3.3.1. In every experiment, we generate data with 10 different seeds, such that the results are not seed-dependent and show the average of all runs. The use of distinct seeds simulates various possible scenarios and the experiment results are thus comparable. Unless stated differently, we generate data distributed on 5 lines, each consisting of 100 data points. To each data point we add a jitter, such they are more representative of real-world data. If the data is ordered in separate lines, we apply the quality measures on each line separately to quantify the intra-line differences.

4.4.1 Expanding the Dimensionality of the Transformation

The choice of transformation dimensionality plays often a pivotal role in the quality of results of dimensionality reduction techniques. This is because selecting the most suitable configurations to aim for the best balance between reducing dimensionality and preservation of essential data characteristics can be challenging. To explore the quality of the variants of CPCA, we start by transforming the data multiple times, with different seeds for data generation.

Furthermore, we progressively increase the number of dimensions in each transformation. This step allows us to explore the impact of dimensionality on the quality of order-based distance and relation preservation.

In this experiment setting, Figure 4.3 presents the results of PCA, Id PCA and the CPCA variants applied to zigzag data. The outcomes for all three CPCA variants show poor performance in both the reconstructed and transformed representations, assessed with both measures. However, Id PCA exhibits a similar outcome to PCA while better preserving the cumulative distances to each predecessor and successor of every data point, as measured by

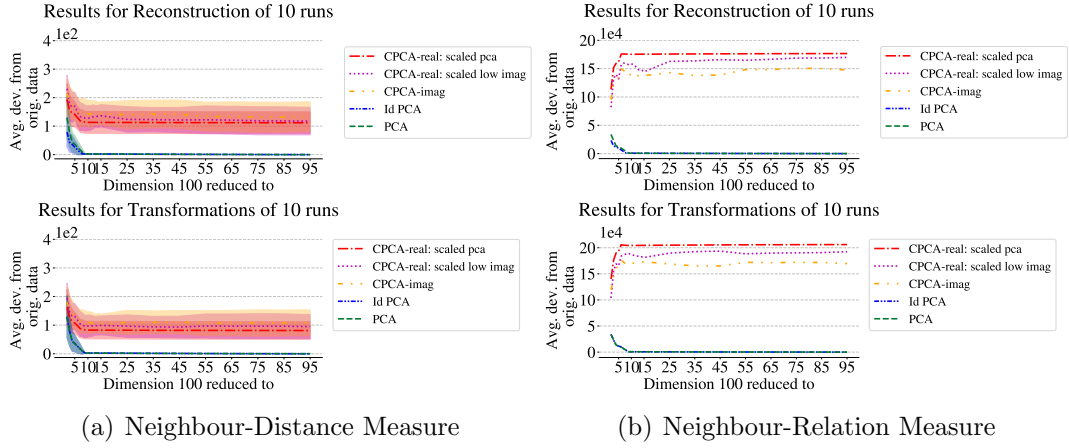


Figure 4.3: Results for zigzag data: comparing the summed distances between each predecessor and successor for data points in the original dataset and their reduced/reconstructed representations as dimensions increase

the ND metric. This is also shown in Figure 4.4, which focuses on dimensions between 2 and 10, a range often preferred for analysis.

Applying the dimensionality reduction methods to data that is smoothly ordered gives similar results to them applied to zigzag data, see Figure 4.5. Here the size of the variance of *CPCA-imag* strikes as it is multiple times bigger than the variances of the results of the other dimensionality reduction methods measured with the ND measure. Id PCA and PCA perform similarly well, with Id PCA maintaining the ND distances better for a lower number of dimensions but keeping the intra-point relations marginally worse in its reconstructed representation, see Figure 4.6.

All methods keep the order-based distances on separate lines similarly in both the reconstructed as well as the lower-dimensional embedding for both measures, see Figure 4.7 and 4.8.

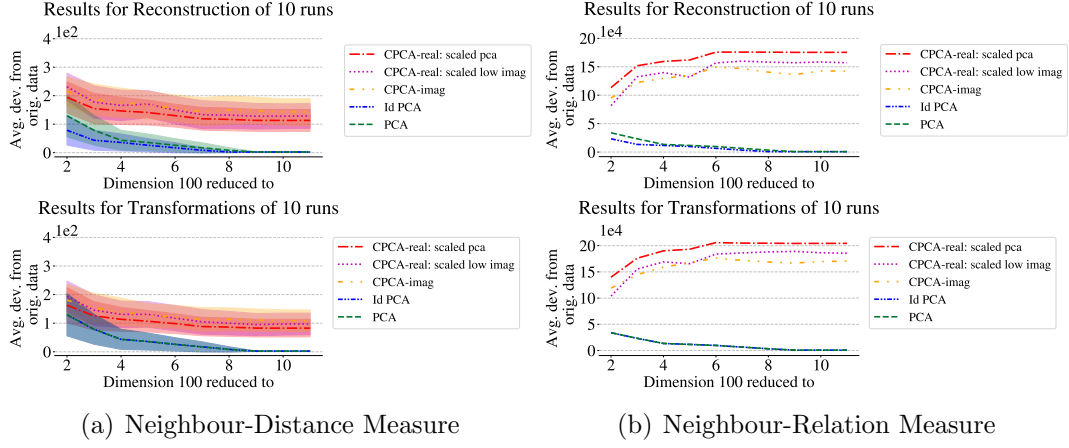


Figure 4.4: Results for zigzag data for transformation dimensions between 2 and 10: comparing the summed distances between each predecessor and successor for data points in the original dataset and their reduced/reconstructed representations as dimensions increase

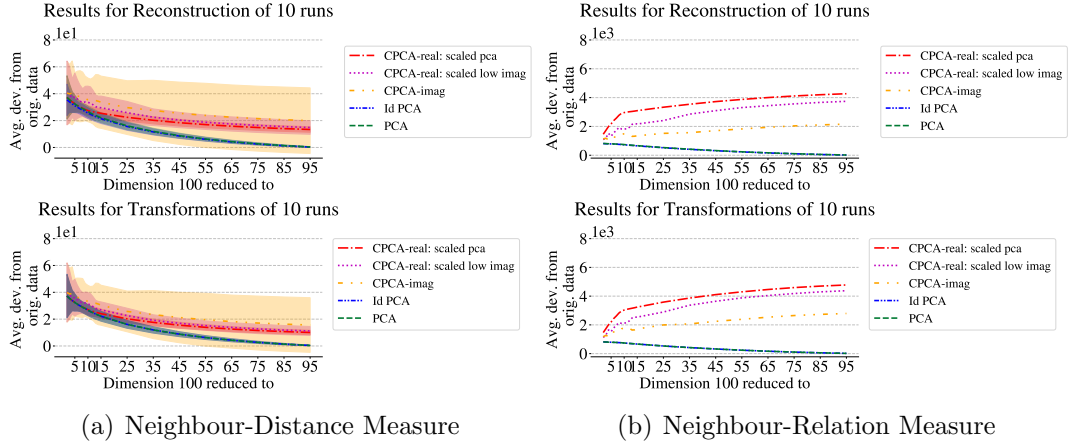


Figure 4.5: Results for smoothly ordered data: comparing the summed distances between each predecessor and successor for data points in the original dataset and their reduced/reconstructed representations as dimensions increase

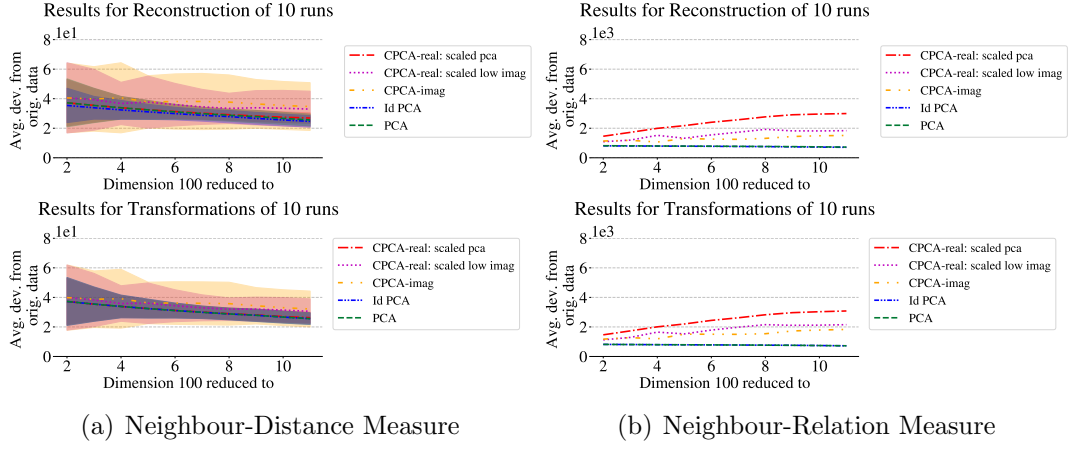


Figure 4.6: Results for smoothly ordered data for transformation dimensions between 2 and 10: comparing the summed distances between each predecessor and successor for data points in the original dataset and their reduced/reconstructed representations as dimensions increase

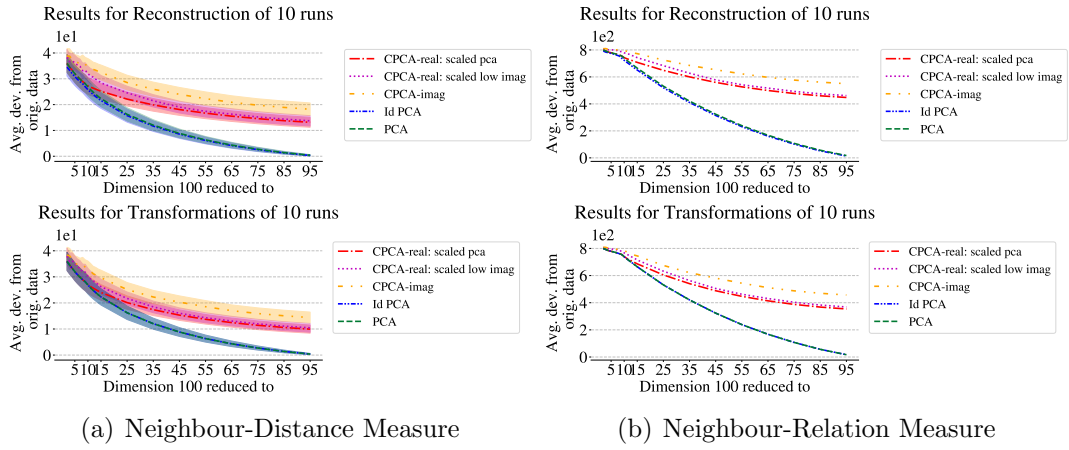


Figure 4.7: Results for data in separate lines: comparing the summed distances between each predecessor and successor for data points in the original dataset and their reduced/reconstructed representations as dimensions increase

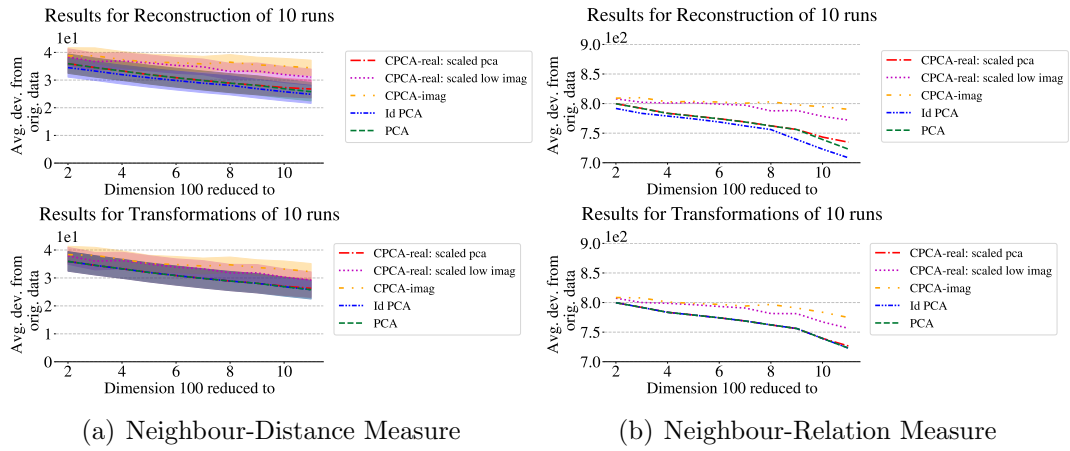


Figure 4.8: Results for data in separate lines for transformation dimensions between 2 and 10: comparing the summed distances between each predecessor and successor for data points in the original dataset and their reduced/reconstructed representations as dimensions increase

4.4.2 Exploration of Increasing Numbers of Neighbours

To understand how the dimensionality reduction techniques impact the ordering of data points in a sequential data set, we explore in this experiment setting the quality of these techniques respective to an increasing number of predecessors and successors. The data sets are reduced to dimensionality 2, as this dimensionality is often used to visualise data [27]. Furthermore, we quantify the distances and relationships between the reconstructed as well as the transformed representation with the distances and relationships in the original data with an increasing number of neighbours for each run and present the averaged result.

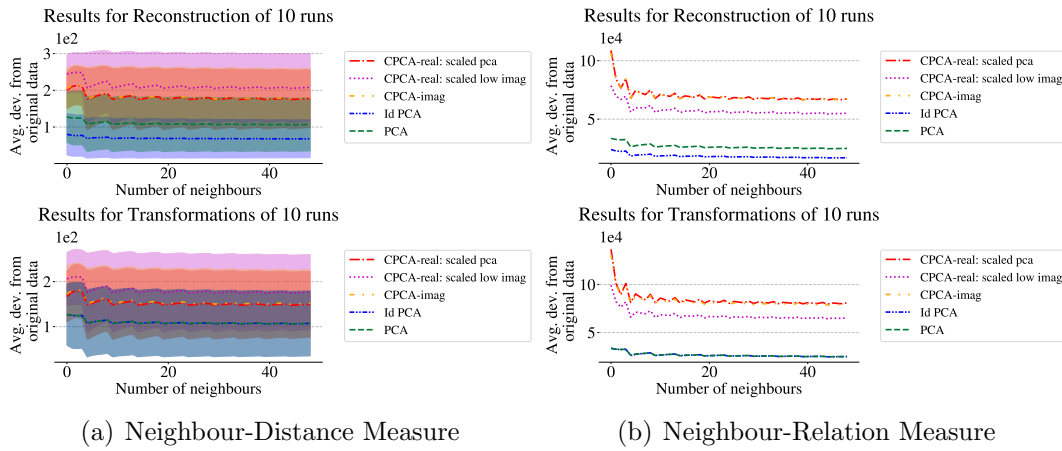


Figure 4.9: Results for 10 runs for zigzag data: comparing the summed distances of an increasing number of predecessors and successors

Similar to the previous experiment setting, the CPCA variants poorly maintain the relations and distances between each reconstructed and original as well as transformed and original data, see Figure 4.9. On the other hand, the result of Id PCA consistently outperforms PCA when comparing the distances and relationships between an increasing number of nearest neighbours in the sequence. It preserves the distances and relationships equally well as PCA in the lower-dimensional space. In Figure 4.10, we observe a high variance in the result for each variant for CPCA. However, the means for all CPCA variants are roughly to those of PCA and Id PCA. This trend also holds when comparing the relationships within the data using the MNR measure, where each dimensionality reduction method performs approximately similarly.

The same scenario is exhibited in Figure 4.11. The variants of CPCA show high variance, but overall all methods maintain similar relations and distances in the sequences of the dataset.

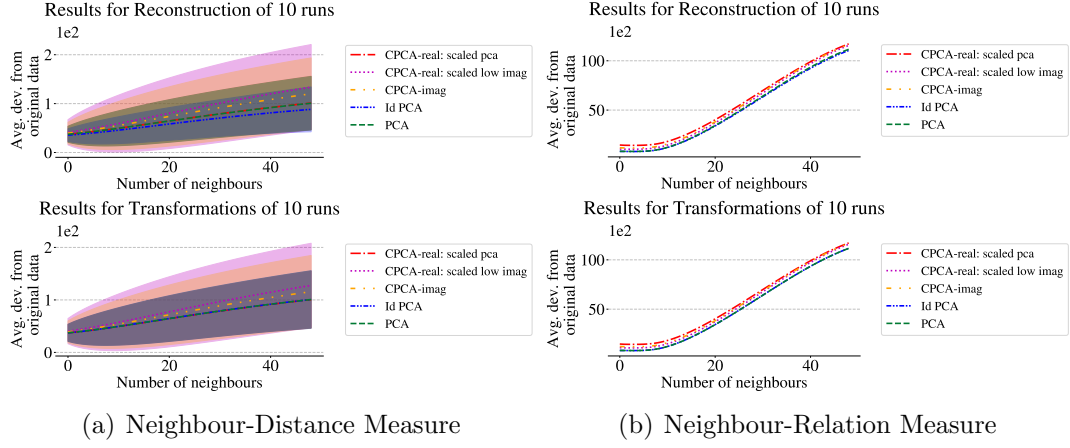


Figure 4.10: Results for 10 runs for smoothly ordered data: comparing the summed distances of an increasing number of predecessors and successors

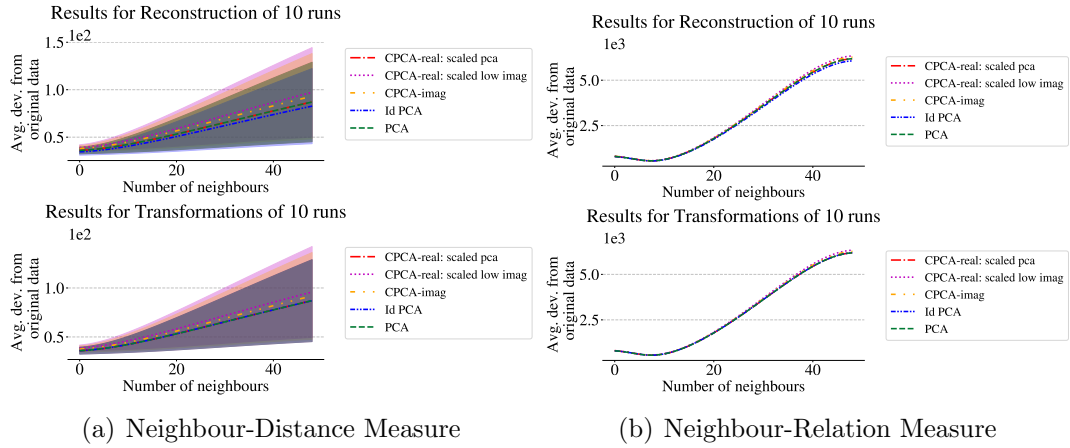


Figure 4.11: Results for data in separate lines: comparing the summed distances of an increasing number of predecessors and successors

4.4.3 Analysis of Growing Numbers of Neighbours and Transformation Dimensions

In this experiment, we combine the previous experiments to explore the effects of reducing dimensionality to a growing number of dimensions while also increasing the number of neighbours examined compared to the original dataset.

Applying Id PCA, PCA and the variants of CPCA to zigzag data in this setting yields the results visualised in Figure 4.12 quantified by the ND measure. We observe that while the results of the CPCA variants show slightly better order-based distance preservation in the transformational representation than the preservation in the reconstructed embedding, they are still inferior to the results of PCA and Id PCA. Furthermore, we see the impact of the zigzag order of the data. All CPCA variants maintain order-based distances somewhat poorly up to the four nearest predecessors and four nearest successors. These data points do not lie on the same lines. However, when comparing the five nearest neighbours in each direction of the sequence, we see an improvement. This is because every fifth predecessor and successor lie on the same line as data was generated on five lines. Analogously to the previous section, we observe that Id PCA performs slightly better than PCA in two dimensions in the reconstructed embedding.

We find a similar pattern when measuring the relations between a growing number of nearest neighbours in a sequence while transforming to an increasing number of dimensions, measured by MNR, see Figure 4.13. For each CPCA variant, we notice a marginally better result for dimensionality 2, indicating a higher quality of order-based relation preservation than in higher dimensions. Interestingly, all CPCA variants face significant challenges in maintaining order-based relations with the immediate predecessor and successor.

Applying the variants of CPCA to smoothly ordered data yields a similar result for all dimensionality reduction methods, with a worse result for *CPCA-real+low imag* and *CPCA-imag*, see Figure 4.14.

However, when evaluating the same transformations and reconstructions using the MNR measure, as shown in Figure 4.15, we observe different outcomes. Id PCA and PCA exhibit similar performance in maintaining relations within the data, whereas *CPCA-imag* and *CPCA-real+low imag* show slightly lower performances. *CPCA-real+scaled pca* underperforms in comparison to the other techniques. Across all variants of CPCA, we notice that they better maintain relations within the data when the number of transformed dimensions is lower, while the number of neighbours does not significantly impact their performance.

PCA, *CPCA real+aligned PCA* and Id PCA exhibit similar performance in preserving distances within consecutive data points in multiple sequences, as

shown in Figure 4.16. However, *CPCA real+low imag* and *CPCA imag* show comparatively less favourable results. All of the tested techniques improve their performance, as the number of dimensions used for data transformation increases, while the quality tends to decrease with a growing number of neighbours.

Interestingly, both *CPCA real+low imag* and *CPCA imag* exhibit superior preservation of the Multiple-Neighbour-Relation (MNR) measure as the number of neighbours increases, in contrast to PCA, Id PCA and *CPCA real+scaled PCA*, as depicted in Figure 4.17. This suggests that although the Euclidean distances are not as well preserved in comparison to PCA, the CPCA variants that retain a substantial part of their imaginary component, excel in preserving relations between up to 10 neighbours better than the other methods. PCA, Id PCA and *CPCA real+scaled PCA* follow a similar pattern: they exhibit suboptimal preservation of relations for up to four dimensions and for more than seven neighbouring data points in both directions of the sequences. It is logical that these methods perform better in higher dimensions, as higher-dimensional transformations allow for the retention of more information but it is surprising that this stops the more neighbours we examine.

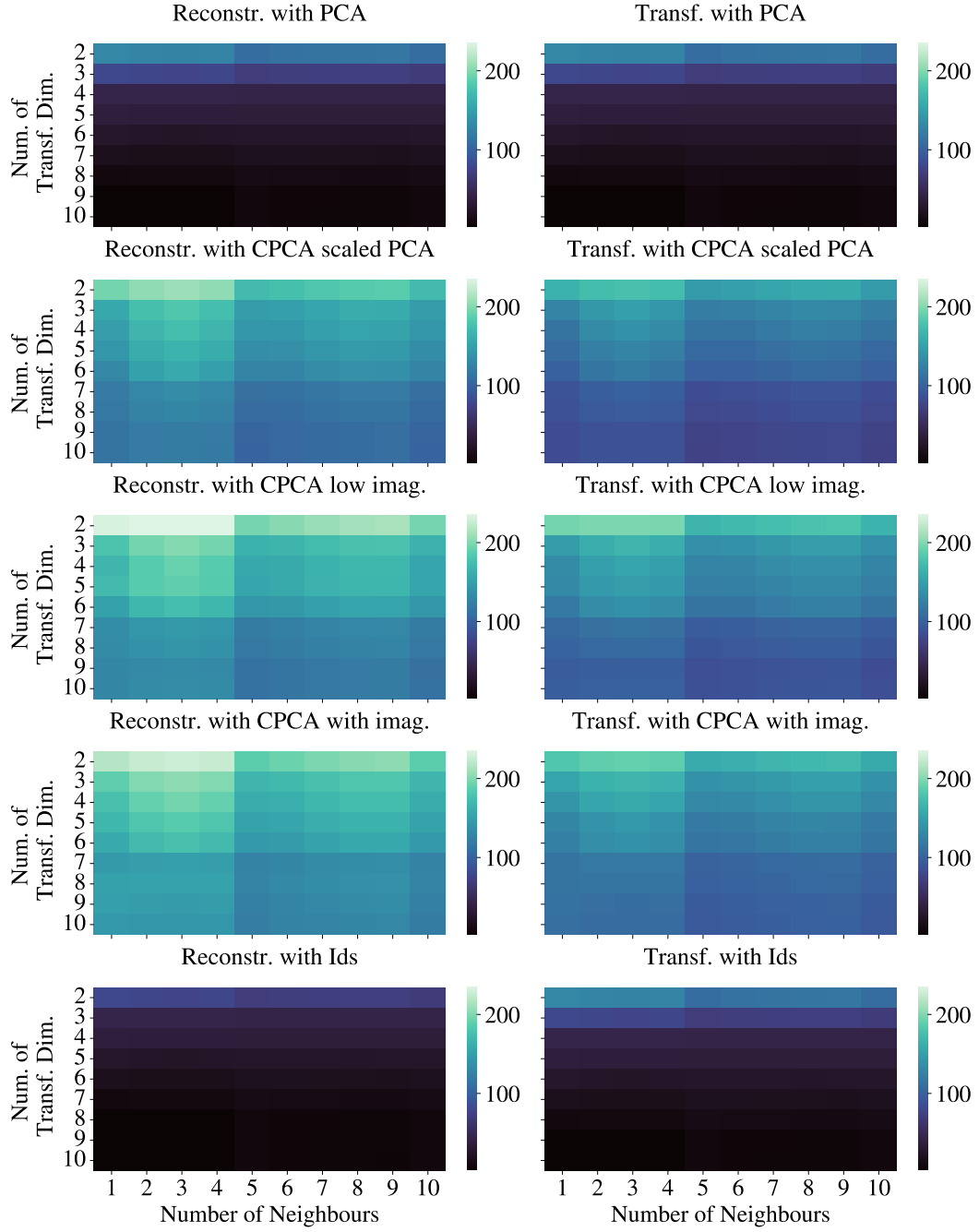


Figure 4.12: Neighbour-Distance measure on zigzag data: comparing the results for increasing numbers of neighbours and dimensionality of the transformed space

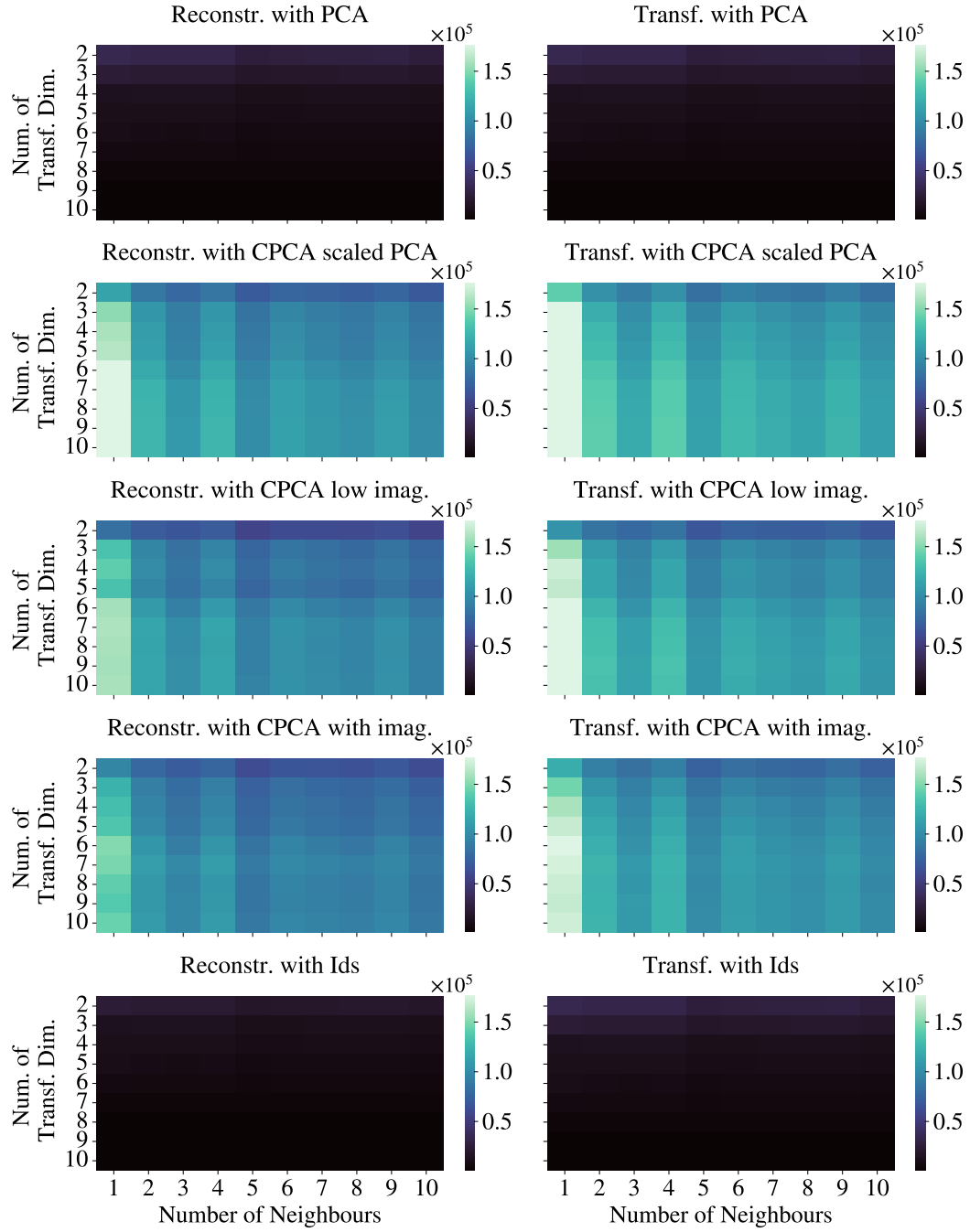


Figure 4.13: Multiple-Neighbour-Relations measure on zigzag data: comparing the results for increasing numbers of neighbours and dimensionality of the transformed space

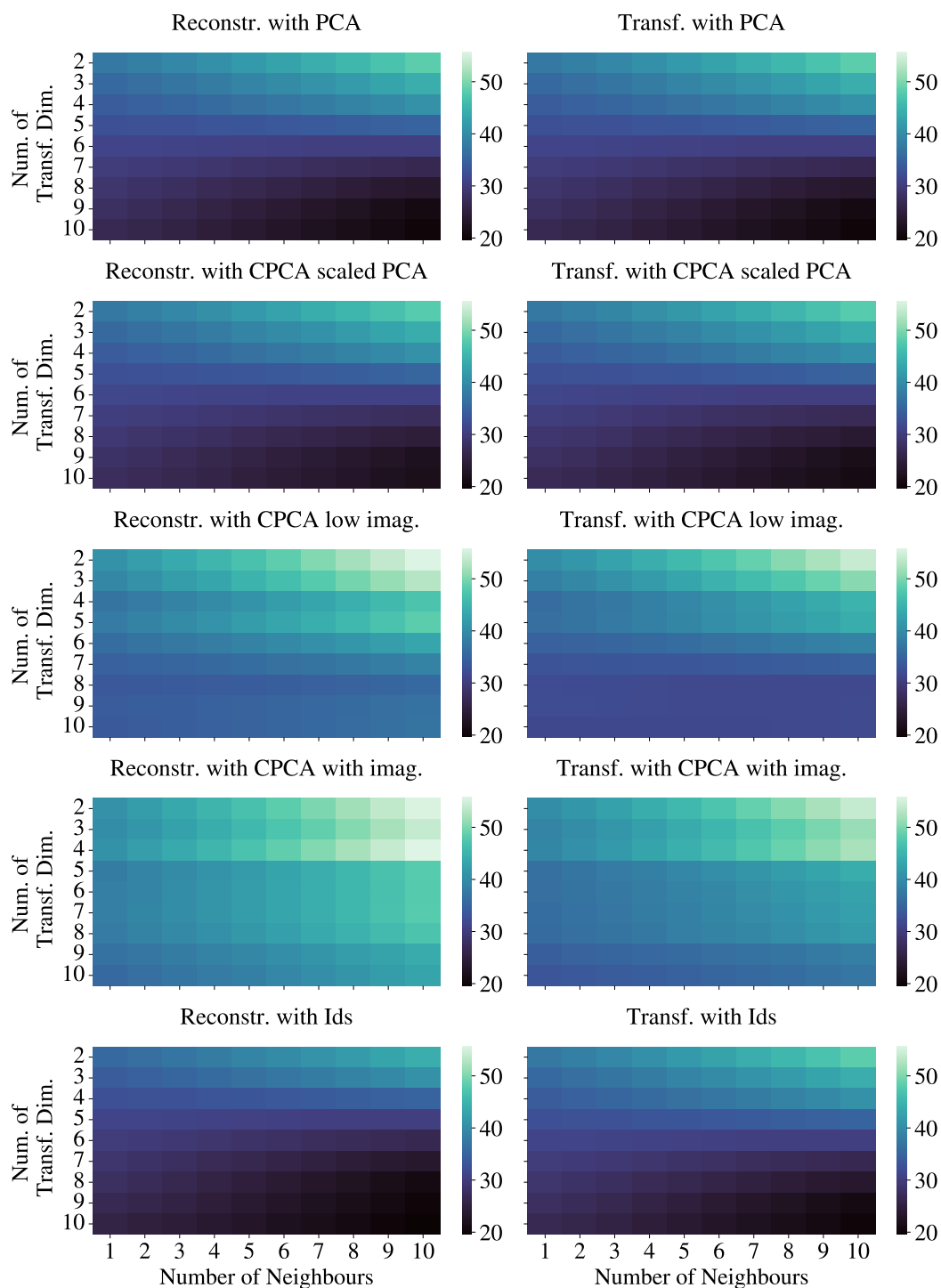


Figure 4.14: Neighbour-Distance measure on smoothly ordered data: comparing the results for increasing numbers of neighbours and dimensionality of the transformed space

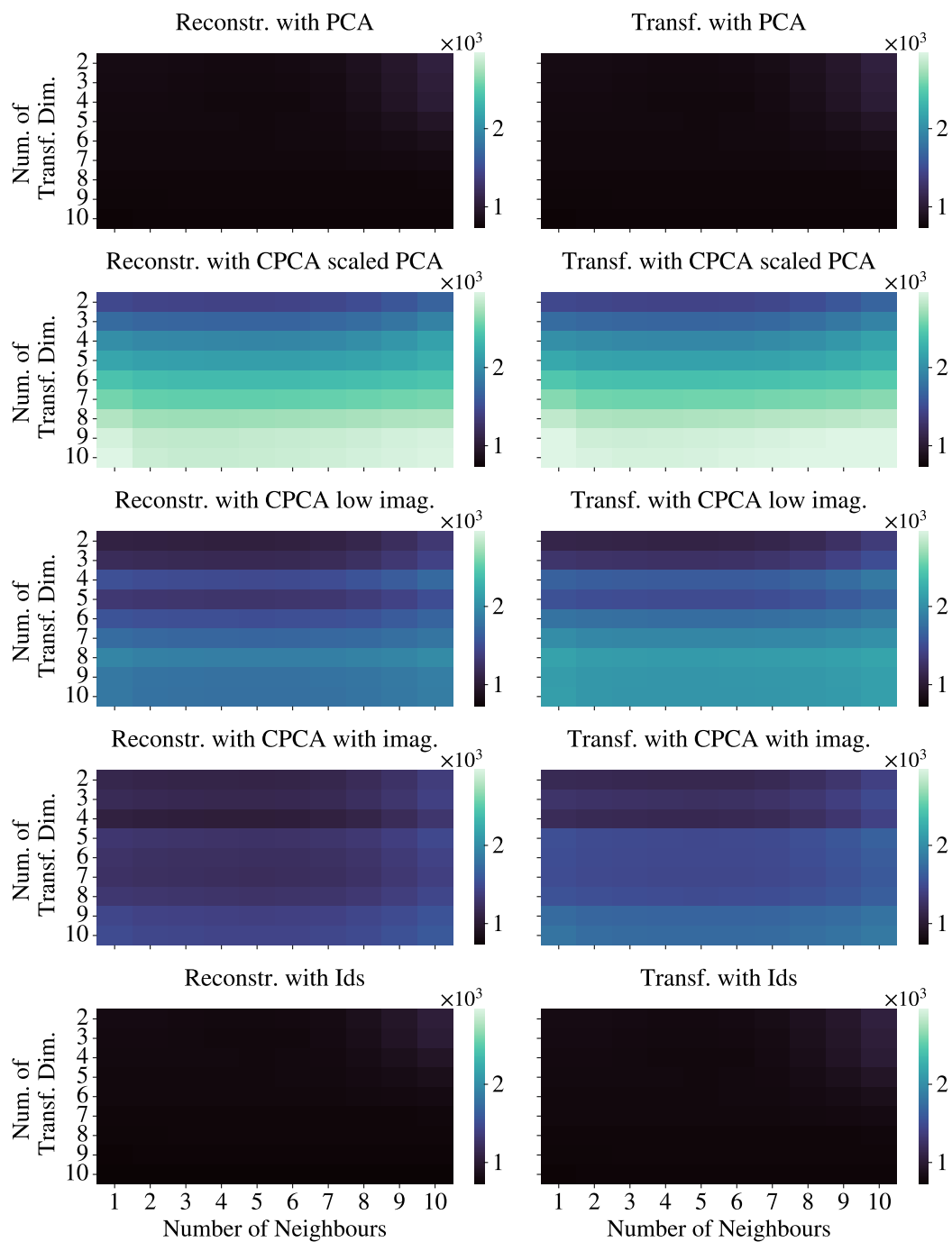


Figure 4.15: Multiple-Neighbour-Relation measure on smoothly ordered data: comparing the results for increasing numbers of neighbours and dimensionality of the transformed space

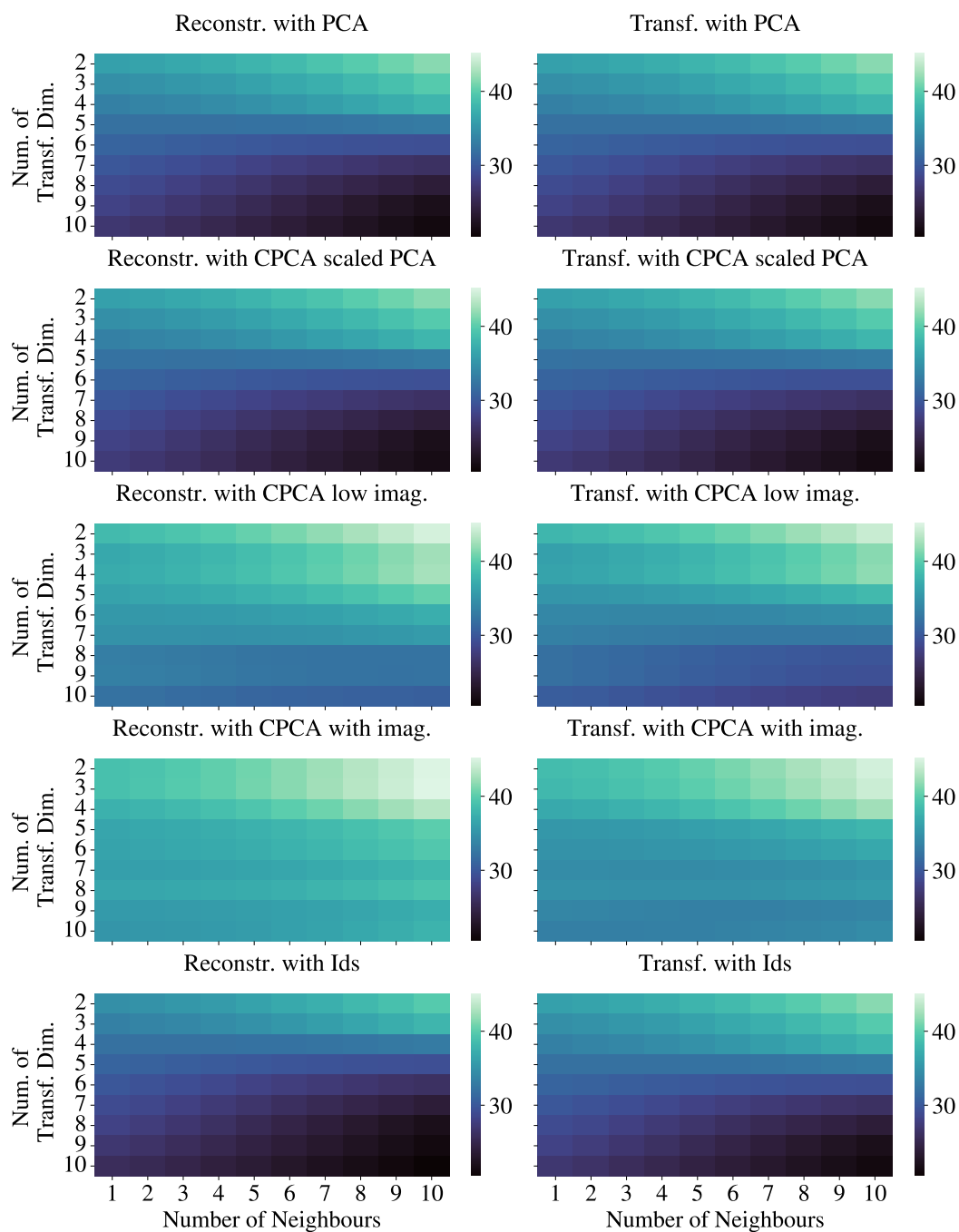


Figure 4.16: Neighbour-Distance measure on data on separate lines: comparing the results for increasing numbers of neighbours and dimensionality of the transformed space

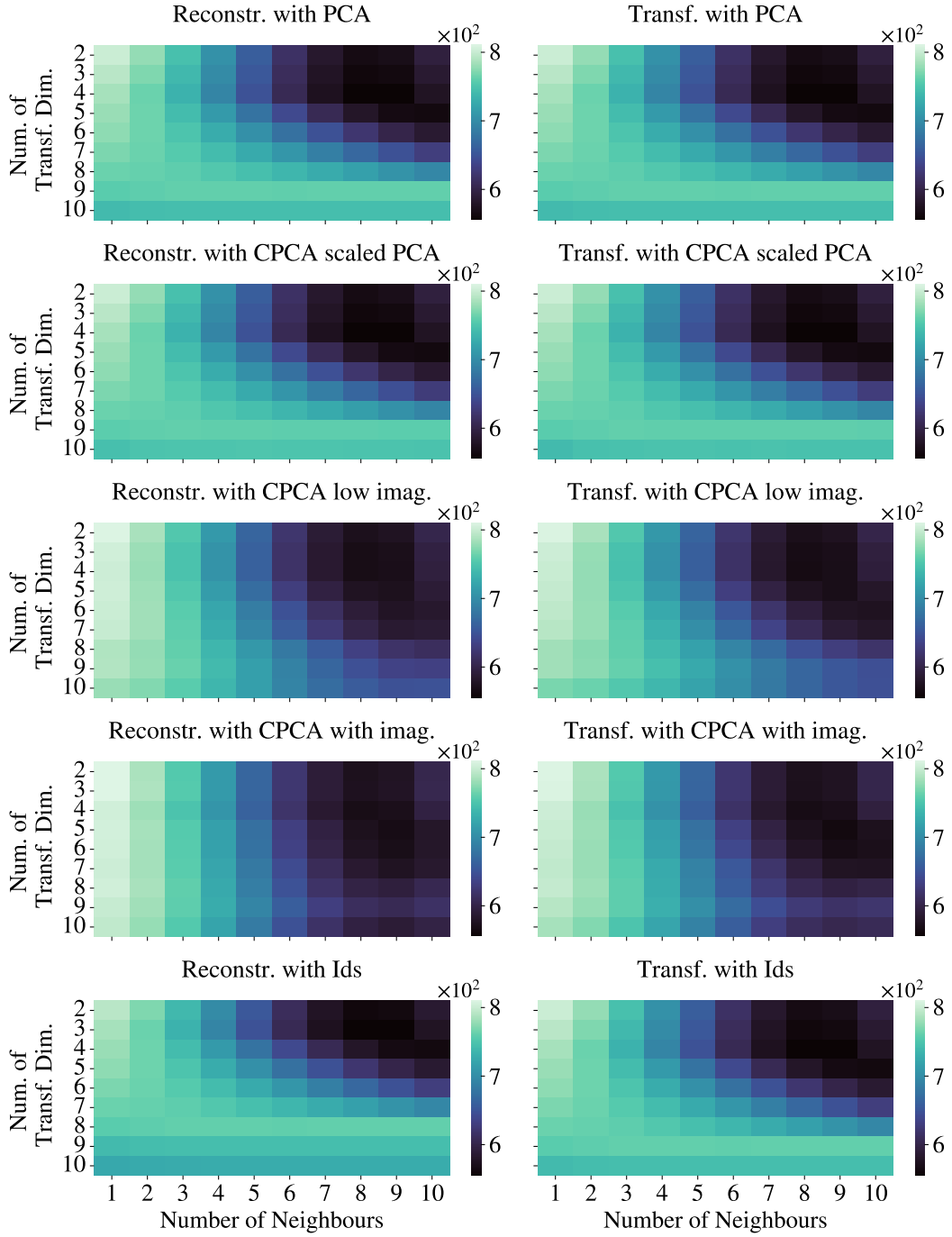


Figure 4.17: Multiple-Neighbour-Relation measure on data in separate lines: comparing the results for increasing numbers of neighbours and dimensionality of the transformed space

4.5 Evaluation with PCA* on Synthetic Data

In this section, we directly compare PCA* with other commonly used dimensionality reduction techniques. Based on the promising results obtained from initial experiments with PCA*, we have chosen to prioritise it as the primary dimensionality reduction method for our extended analyses. We evaluate PCA* alongside other methods across four distinct kinds of experiments using the generated data, outlined in Section 4.1.

Analogously to the previous experiments, we generate 10 different datasets consisting of 5 lines, each entailing 100 data points with jitter. These datasets strongly depend on the seed with which they are generated. The original space resides in a 100-dimensional space unless otherwise specified. For visibility reasons, we omit to display the variance of the Neighbour-Relation (NR) and Multiple-Neighbour-Relation (MNR) measures since it remained equally large for each observation across all experiments. When the data is organised in separate lines, we evaluate each line individually. In the following experiments, we present a side-by-side comparison of how well distances are preserved using the Neighbour-Distance (ND) measure and how well relationships are maintained using either the NR or MNR measure, see Definitions 4.2.4 and 4.2.6. In every evaluation, we compare the distances and relations in both, transformed and reconstructed embeddings, with the distances and relations within the original data.

Furthermore, k -nearest neighbours denote in this setting, the k -nearest neighbours on the sequence in both directions resulting in $2k$ neighbours when using the ND measure.

4.5.1 Increasing the Number of Dimensions of the Original Space

We explore the average deviation between the original and transformed distances while the number of dimensions of the original higher-dimensional space increases from two onward. Therefore, by enlarging the dimensionality of the original space, we control the complexity of the data. Then we measure the quality of the dimensionality reduction methods applied with the quality measures described in Section 4.2. We compare the relations between each preceding and succeeding data point for each element in the dataset in the original space and to its representation for the reconstructed data and respectively its lower-dimensional embedding with different metrics.

Figure 4.18 presents the outcomes of dimensionality reduction methods applied to zigzag data. Notably, PCA* stands out by preserving both distance and relations with respect to its first neighbours on the sequence more effec-

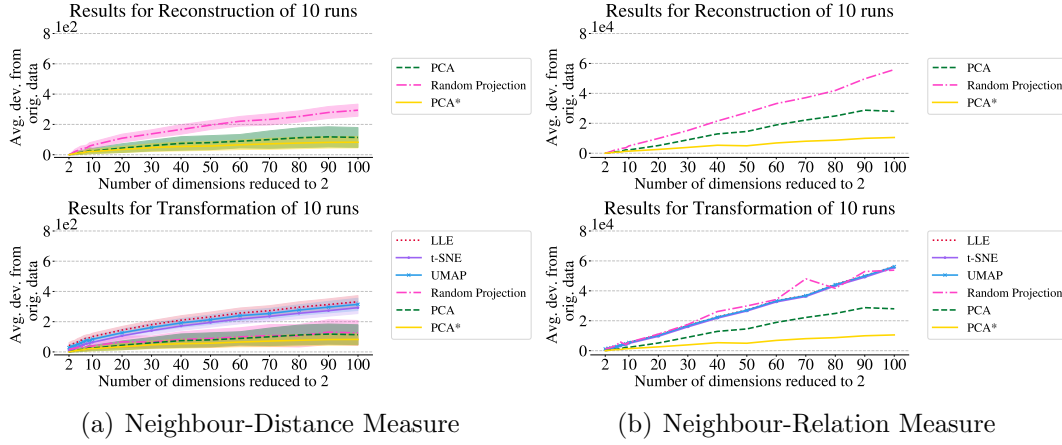


Figure 4.18: Results zigzag data: increasing the number of dimensions of the original data

tively than other methods, as evidenced by its lower results for both measures. In contrast, random projection lags behind PCA and PCA* and is similar to LLE, t-SNE and UMAP. LLE, t-SNE, and UMAP exhibit comparable results

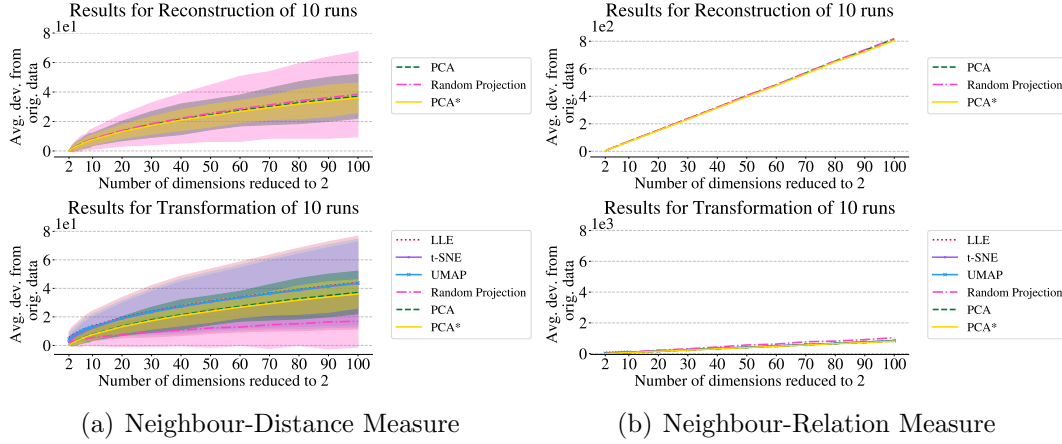


Figure 4.19: Results for smoothly-ordered data: increasing the number of dimensions of the original data

Comparing these results to the outcomes of dimensionality reduction methods applied to data connected by a smooth line, we see that PCA* yields a mean result similar to PCA but with lower variance, as illustrated in Figure 4.19. Interestingly, random projection performs well in the lower-dimensional representation, as measured with the ND measure. This is because random

projection falls under the category of local structure-preserving dimensionality reduction techniques, and when the data consists of roughly straight lines, the local structure aligns with the order-based distances. However, it is worth noting that due to its random nature, random projection exhibits significantly higher variance than PCA*. When assessing its results with the NR measure, it becomes evident that random projection does not preserve angular relations as effectively as PCA*, LLE, and other methods in the lower-dimensional space. Overall all methods keep the order-based relations similarly well.

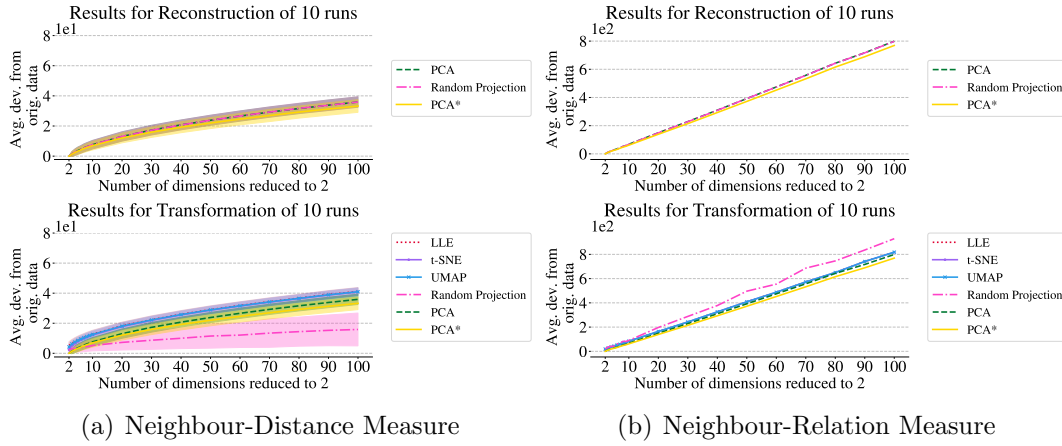


Figure 4.20: Results for data in 5 separate lines: increasing the number of dimensions of the original data

Figure 4.20 presents similar results for data arranged in separate lines. When compared to PCA, LLE, t-SNE and UMAP, PCA* shows marginally lower mean results for each measure. As mentioned earlier, random projection prioritises local structure preservation, making it better at maintaining distances compared to PCA*, given that the order of the data is similar to their spatial distances. However, as seen in previous results, random projection falls short in preserving relations in the transformed space compared to other observed methods. Additionally, its random nature is reflected in the transformed space, with higher variance than any other dimensionality reduction method.

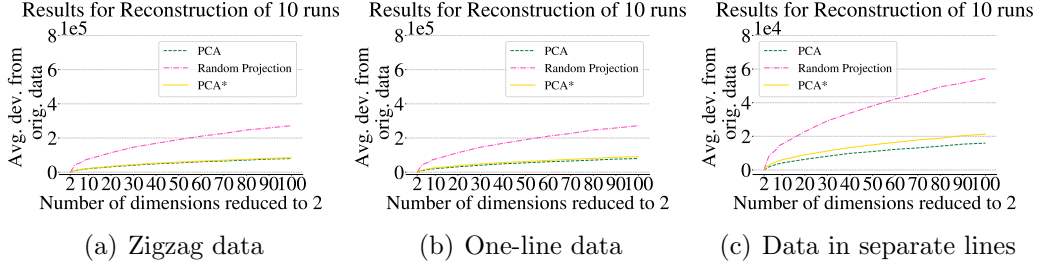


Figure 4.21: Results for data in different orders measured with DTW: increasing the number of dimensions of the original data

Finally, we compare the results of the dimensionality reduction methods with DTW, see Figure 4.21. For each order, PCA* shows similar results as PCA whereas random projection exhibits substantially worse results. This may be because random projection transforms data onto lower-dimensional space by applying random linear transformation without taking into account the intrinsic structure of the data.

4.5.2 Expanding the Dimensionality of the Transformation

In this experiment, we apply dimensionality reduction methods to the 100-dimensional dataset while systematically increasing the number of dimensions in the transformed space. We start with a reduced dimensionality of two and gradually increase it up to the original number of dimensions. To assess the quality of the dimensionality reduction techniques, we conduct a comparison between the transformed and original data, and analogously the reconstructed and original data. This evaluation involves computing the average deviation for each summed distance between each point and its predecessor and successor, employing the quality measures in Section 4.2. With this experiment, we determine the optimal number of dimensions to which the dataset should be transformed. By varying the dimensionality, we gain insights into how dimensionality reduction impacts the dataset’s overall quality and structure.

When we compare all dimensionality reduction techniques applied to zigzag data, we observe that random projection performs less effectively than both PCA and PCA*, as shown in Figure 4.22. Interestingly, the results of LLE, t-SNE and UMAP do exhibit significant variations. This could be attributed to the fact that we did not fine-tune the parameters of these techniques and instead relied on default values. For instance, sklearn’s implementation of t-SNE uses a default perplexity value of 30, which is associated with the number of the

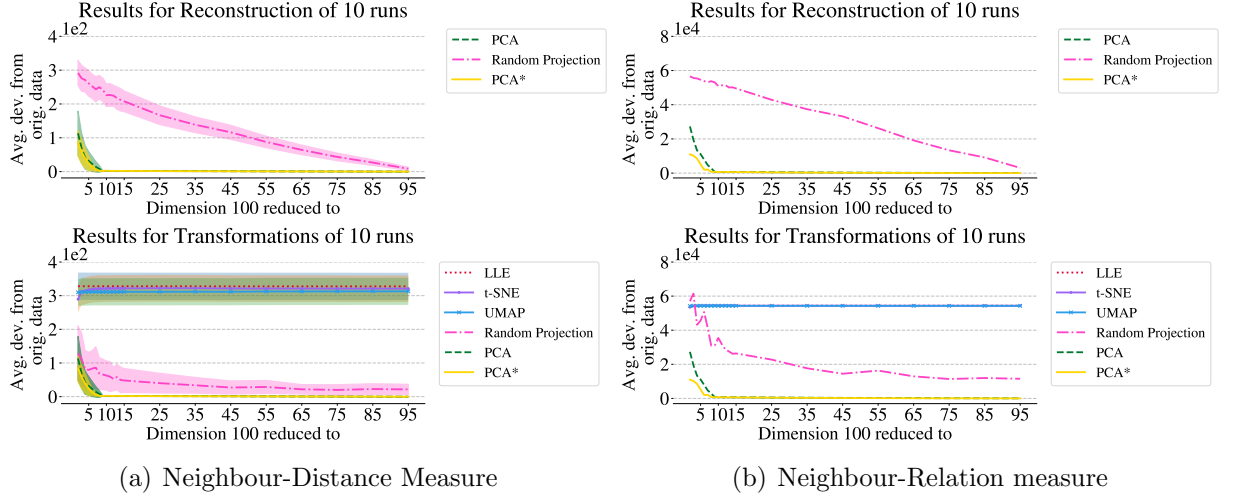


Figure 4.22: Results for zigzag data: increasing the number of dimensions of the lower-dimensional space

spatial nearest neighbours considered in other dimensionality reduction methods [43]. Given that LLE, t-SNE and UMAP are local structure-preserving methods, they tend to maintain distances to spatially nearest neighbours. In other words, these methods try to maintain the distance to the spatially nearest neighbours, they aim to keep the same distance for all lower dimensionalities. Therefore the results remain constant for each number of dimensions to which the data is transformed.

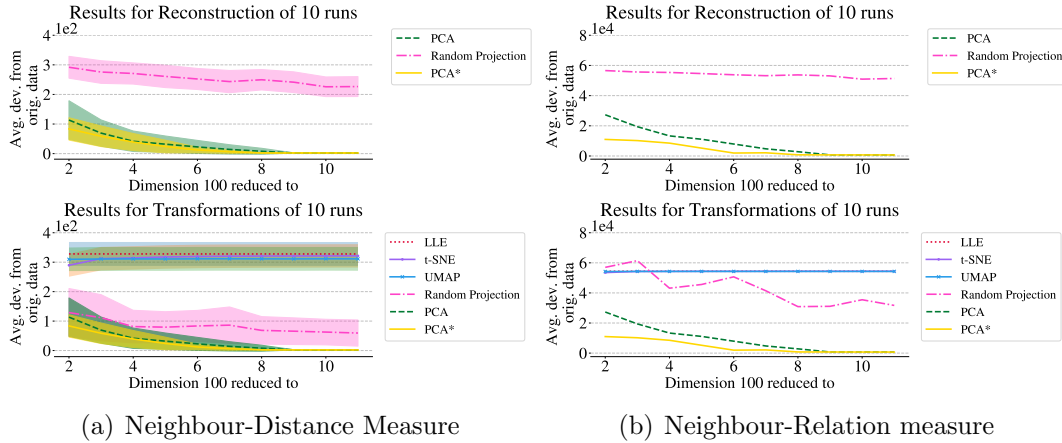


Figure 4.23: Results for zigzag data: increasing the number of dimensions between 2 and 10 for the lower-dimensional space

As data visualisation is only feasible in lower than 3 dimensions, we explore the range from 2 to 10 in Figure 4.23. PCA* outperforms PCA and random projection for lower dimensions, exhibiting lower variance in the ND measure. This holds for both measures, especially for lower dimensions up to approximately eight.

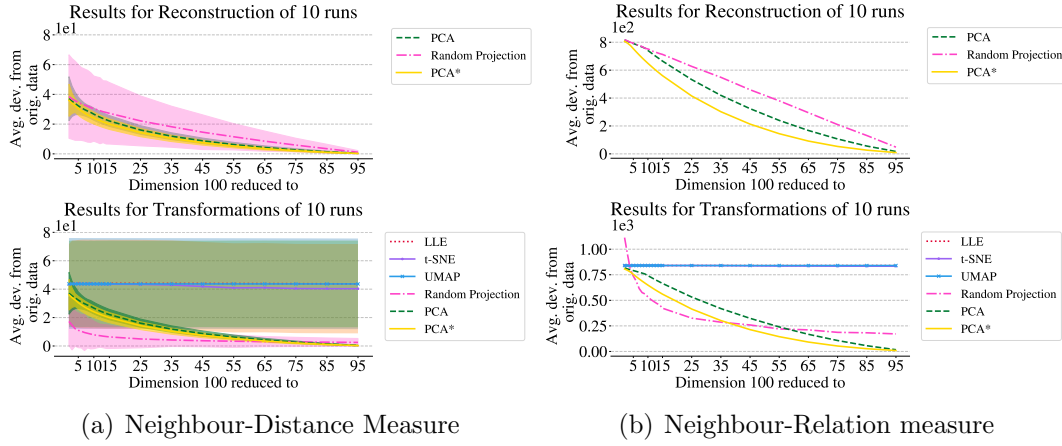


Figure 4.24: Results for smoothly ordered data: increasing the number of dimensions of the lower-dimensional space

Comparing the results of the dimensionality reduction methods applied to smoothly ordered data, we observe that LLE, t-SNE and UMAP are again stable but with a lower variable than the ones on zigzag data, as shown in Figure 4.24. This improvement may be attributed to the smoother data order, which is more akin to the spatial structure. Similarly, the reconstruction of data transformed by random projection yields results comparable with PCA and PCA*. However, concerning the transformed data, random projection outperforms PCA*. PCA*'s variance is nevertheless the smallest to all other compared techniques. Moreover, it shows for both spaces improving results and measured with MNR, we see PCA* outperforming PCA and random projection for their reconstructed representation.

Zooming into the first 10 dimensions, we see in Figure 4.25 PCA* displaying enhanced results compared to the other dimensionality reduction methods for both tested measures as well as for both reconstructed and transformed embeddings. In this case, the random projection transformed data maintains the averaged distances better in the mean.

These results are comparable to the outcomes of applying the dimensionality reduction methods to data in separate lines, as illustrated in Figure 4.26. The performance of PCA* is overall enhanced, apart from the case of com-

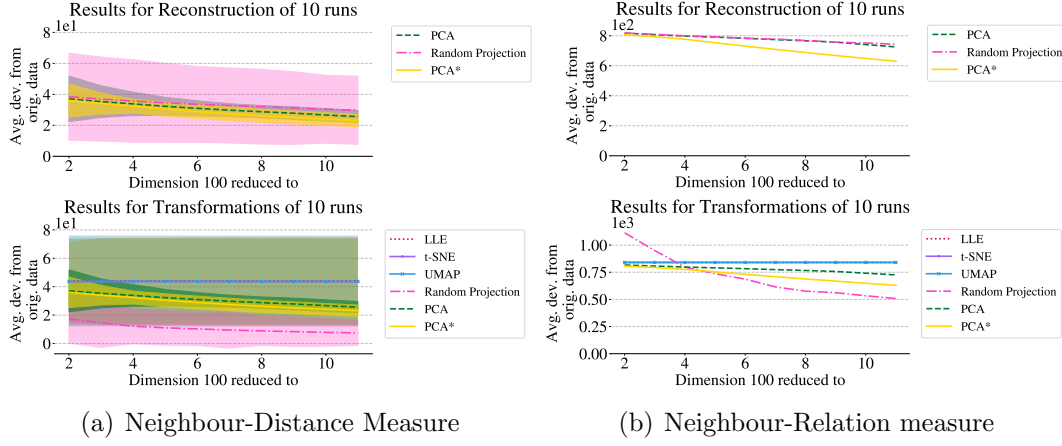


Figure 4.25: Results for one-line data: increasing the number of dimensions between 2 and 10 for the lower-dimensional space

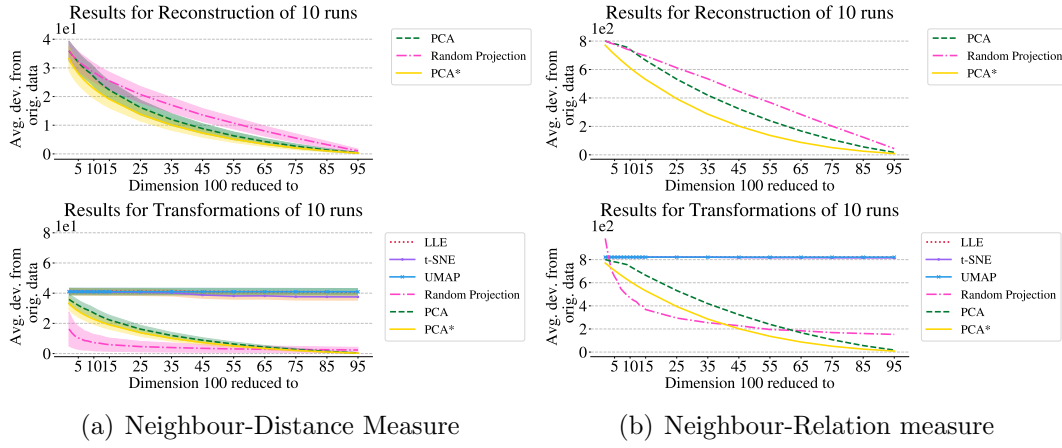


Figure 4.26: Results for data in 5 separate lines: increasing the number of dimensions of the lower-dimensional space

paring respective distances between consecutive data points of the transformed data with the original data. Here random projection outperforms with a higher variance.

Exploring a lower number of dimensions for each transformation, we see PCA* maintaining the order-based distances better than PCA, LLE, t-SNE and UMAP for the reconstruction and transformation measured with both measures ND and MNR. Compared to random projection, we see PCA* having lower results when quantifying the distances and relations of consecutive data points in the order of the reconstructed data, as depicted in Figure 4.27. In the

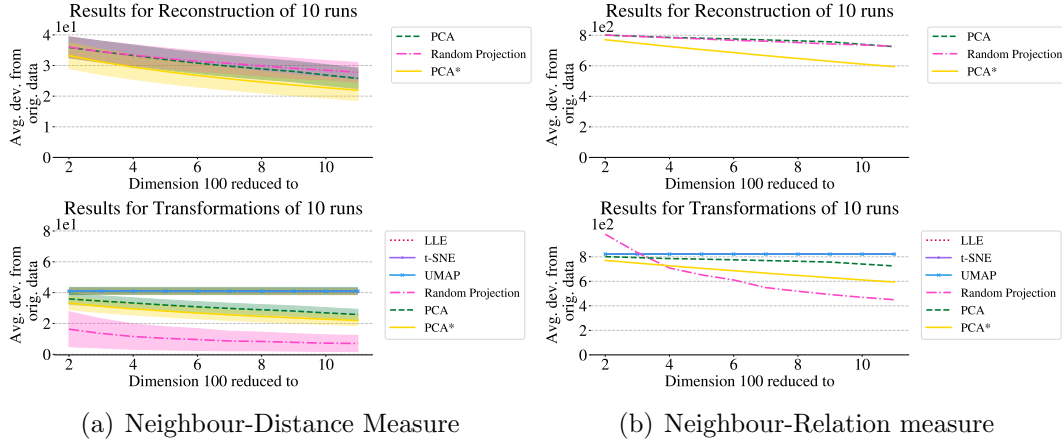


Figure 4.27: Results for data in 5 separate lines: increasing the number of dimensions between 2 and 10 for the lower-dimensional space

case of transformed data, the representation of random projection maintains the pairwise distances computed with ND measure better overall, with a higher variance. If we compare the results calculated by the MNR measure, we see PCA* keeping the relations between consecutive data points in dimensions lower than four better than random projection. After that, random projection outperforms all of the other dimensionality reduction methods.

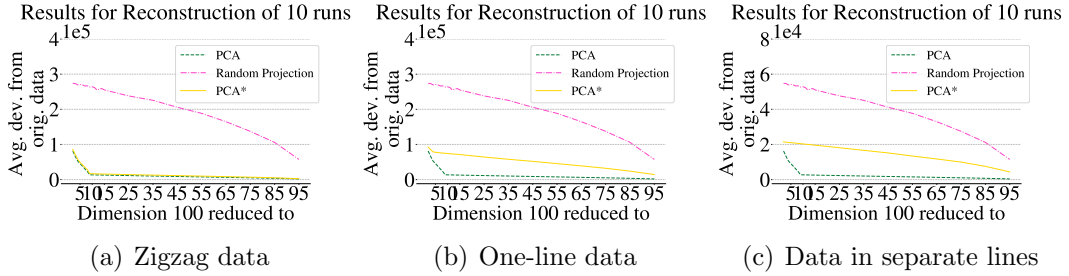


Figure 4.28: Results for data in different orders measured with DTW

Figure 4.28 visualises the results for reconstructions of PCA*, PCA and random projection computed with DTW. We see that similarly to the previous section, PCA and PCA* have comparable results for the reconstruction of zigzag data, whereas the reconstruction of data smoothly ordered or ordered in separate lines shows poorer results than PCA. However, random projection does not keep the DTW distances as well as PCA and PCA*.

4.5.3 Exploration of Increasing Numbers of Neighbours

As dimensionality reduction methods with the local-structure preserving property rely on the concept of nearest neighbours, we evaluate the effect of dimensionality reduction on the data with an increasing number of observed neighbours. We first apply dimensionality reduction methods to the 100-dimensional data to reduce the dimensions to two. We then measure the quality of the transformations for an increasing number of neighbours by quantifying how different the transformed data is compared to the original data. Thus this scenario helps us understand how well the dimensionality reduction method preserves the sequential structure within the data. As our focus lies on order-based distance preservation, we define neighbours as data points which are close to each other in the sequential order of the data.

In the following context, when we refer to observing k neighbours, we are specifying that we are considering k predecessors and k successors that appear within the sequence.

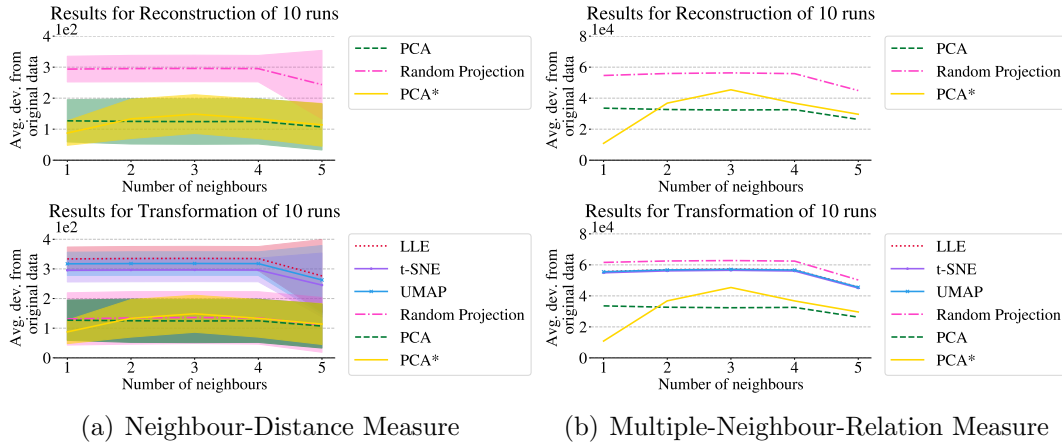


Figure 4.29: Results for data in zigzag order: increasing the number of observed neighbours

Comparing the average deviations for an increasing number of neighbours on zigzag data between its reconstructed and original data and respectively transformed representation with the original data, highlights PCA*'s ability to maintain the distances and relations to its predecessor and successor, see Figure 4.29. For more than two neighbours, the results of PCA and PCA* become more comparable. Random projection, however, shows similar results to the other local structure-preserving methods LLE, t-SNE and UMAP. Each of them performs stable for up to four neighbours and then increases in quality.

Now, applying the same analysis to the data connected in a smooth line, we

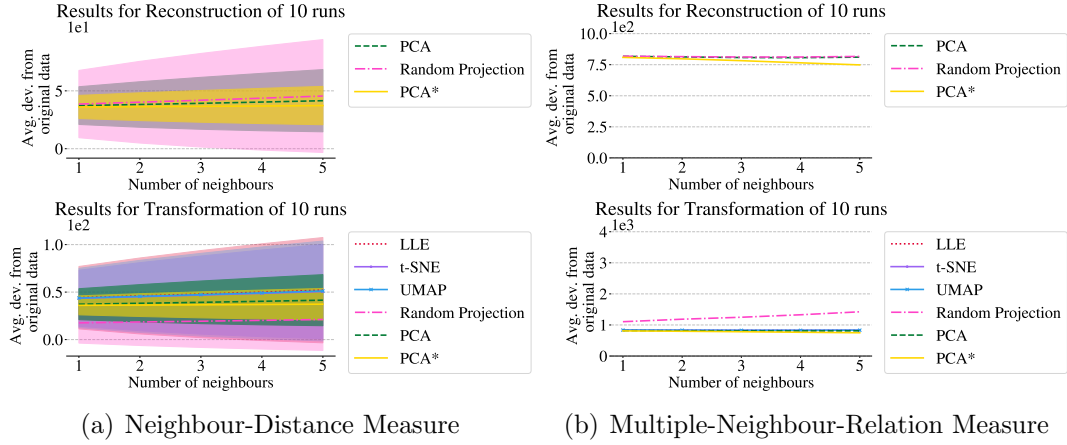


Figure 4.30: Results for smoothly ordered data: increasing the number of observed neighbours

observe the means of PCA and PCA* being similar, as shown in Figure 4.30. Random projection outperforms PCA* mean-wise in the lower-dimensional projection. However, PCA* exhibits a much lower variance compared to the other methods included in the comparison. When quantifying the results using the MNR measure, we notice a significant improvement for PCA* over PCA, the more neighbours we compare in the original dimensional space. In the lower-dimensional embedding, the results of random projection are, compared to the other methods, strongly increasing, implying that random projection does not maintain the angular relations within the data as well. This observation aligns with findings from previous experiments.

Moving onto the analysis of dimensionality reduction methods applied to data ordered in separate lines, as illustrated in Figure 4.31. In this scenario, we observe a consistently strong performance by PCA* relative to other dimensionality reduction methods for both measures, with one exception. In the lower-dimensional representation, random projection excels at preserving the distances between consecutive points compared to PCA*, for example. Nevertheless, when we examine the relations with the MNR measure, we once again notice that random projection struggles to maintain the connections between these data pairs effectively.

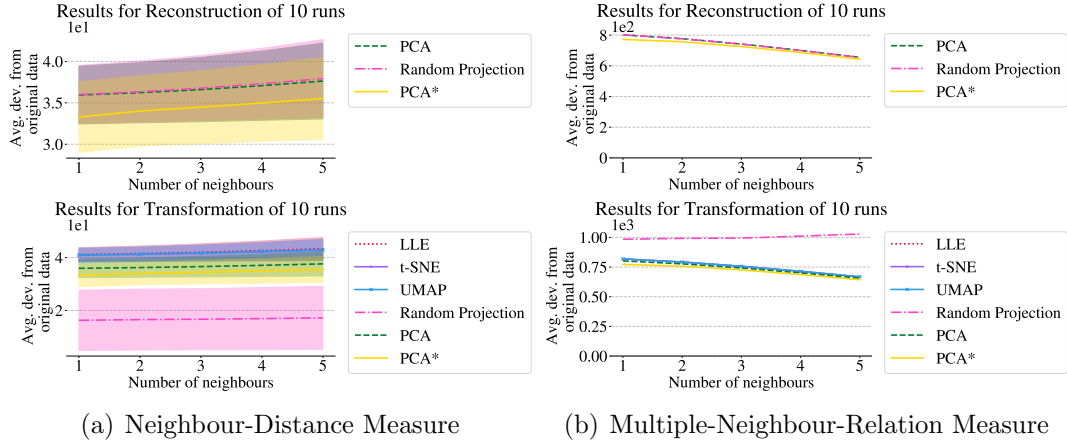


Figure 4.31: Results for data in separate lines: increasing the number of observed neighbours

4.5.4 Analysis of Growing Numbers of Neighbours and Transformation Dimensions

In this scenario, we apply dimensionality reduction methods to a 100-dimensional dataset. We systematically increase the number of dimensions for the transformation while observing a growing number of neighbours. The process involves reducing the dimensionality of the data and then comparing the average deviation in the relationships between nearby data points in both the original and transformed spaces. Essentially, this scenario combines elements from the second and third settings. To quantify the results, we use the two quality measures defined in Section 4.2: Neighbour-Distance measure (ND measure) and Multiple-Neighbour-Relation measure (MNR measure).

We begin by comparing the results of applying dimensionality reduction methods to zigzag-ordered data and quantifying them using the ND measure, as shown in figure 4.32. In the higher-dimensional space, we observe that PCA and PCA* exhibit similar performance, with PCA* outperforming PCA when reducing the data to two dimensions and comparing the first predecessor and successor of each data point. In contrast, random projection demonstrates poor results in the higher-dimensional space and marginally poorer performance in the lower-dimensional space than PCA* and PCA. Thus, each of these methods keeps the distance to the nearest five neighbours comparably low, except from random projection in higher-dimensional space. However, LLE, t-SNE, and UMAP encounter challenges in preserving the order-based distances within the data. Due to their focus on local structure preservation, we see an improved result for comparing the results up to the fifth predecessor and successor.

Similar to these results, PCA and PCA* produce roughly equal results for the MNR measure, but PCA* outperforms PCA for only one predecessor and successor while underperforming for up to three neighbours on both sides, as shown in Figure 4.33. In higher dimensionality, the quality of results for random projection is poorer compared to PCA and PCA*. Random projection, LLE, t-SNE and UMAP exhibit their local structure preservation effect, similar to the case before. However, random projection starts with worse performance in lower dimensions for the transformations and shows improved performance for more than 55 dimensions. In contrast, LLE, t-SNE and UMAP maintain consistently similar results across different dimensions.

Figure 4.34 displays the performance of dimensionality reduction methods applied to smoothly ordered data, with results quantified using the ND measure. We see again that PCA and PCA* yield similar outcomes, with PCA* exhibiting a slight advantage, particularly in lower dimensions. On the other hand, random projection performs worse in its reconstructed form but excels in lower-dimensional settings compared to PCA, PCA*, LLE, t-SNE, and UMAP. In the context of smoothly ordered data, LLE, t-SNE, and UMAP demonstrate poor results in preserving distances for various numbers of neighbours.

Assessing the quality of dimensionality reduction techniques using the MNR measure on smoothly ordered data reveals slightly better results for PCA* compared to PCA and random projection in higher dimensionality settings. However, on transformed dimensionality, the result of random projection slightly outperforms PCA*. In contrast, LLE, t-SNE, and UMAP consistently produce similar results across different dimensions and numbers of neighbours, as shown in Figure 4.35.

Similarly, the results for dimensionality reduction methods applied to data ordered on separate lines, show for PCA and PCA* similar results, with PCA* showing slight improvements, particularly when considering the first neighbours and lower dimensions. Random projection, on the other hand, performs slightly worse than PCA and PCA* in the reconstructed embedding but surpasses them in the transformed representation. In contrast, LLE, t-SNE, and UMAP consistently underperform when compared to PCA, PCA*, and random projection, as depicted in Figure 4.36.

Similar to the previous figures, Figure 4.37 reveals a marginal improvement for PCA* for the reconstructed data when compared to PCA and random projection. PCA and random projection exhibit similar reconstruction quality. When comparing the transformations, we see random projection outperforming PCA*. LLE, t-SNE, and UMAP consistently produce poorer outcomes compared to PCA.

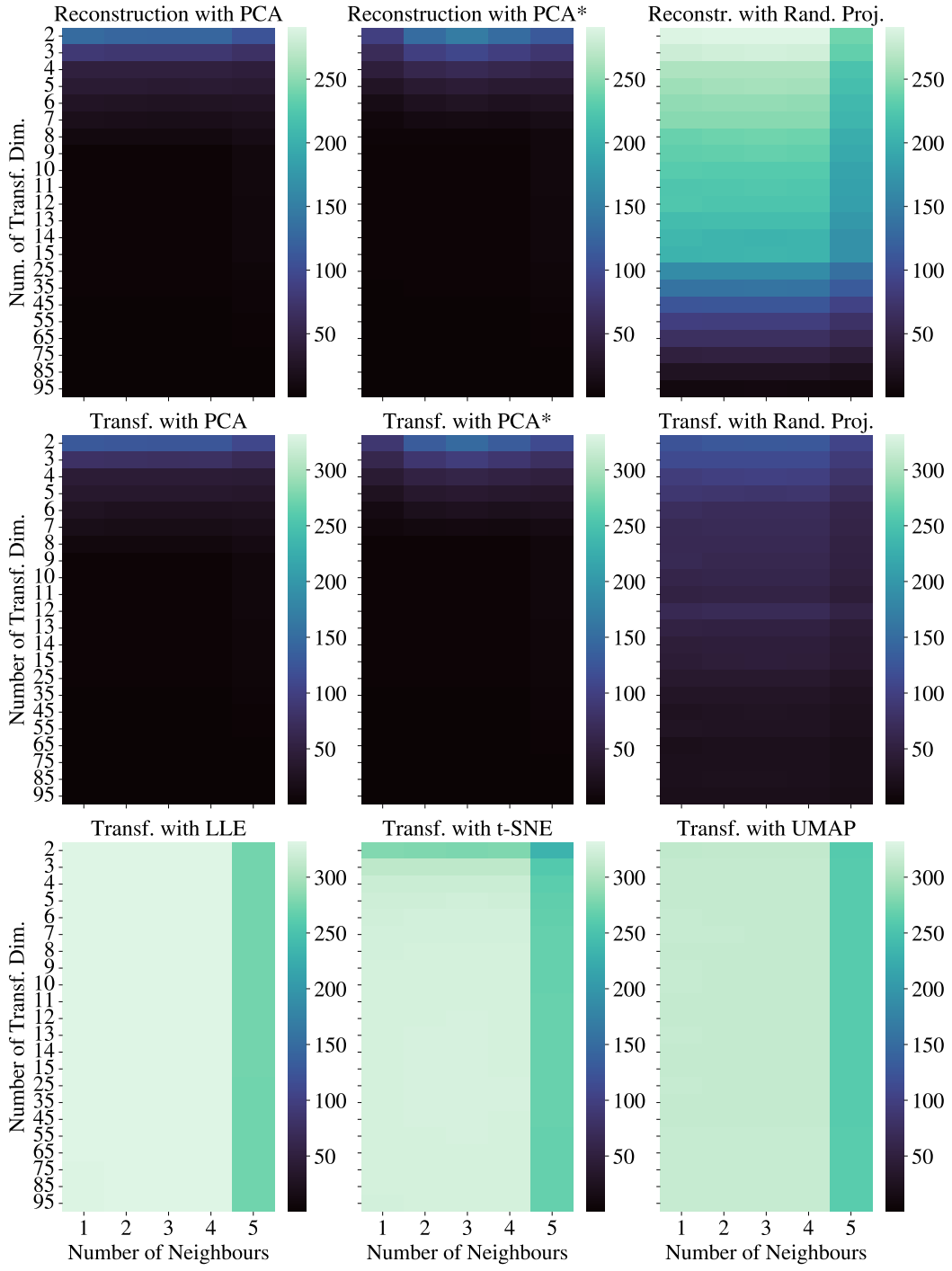


Figure 4.32: ND measure for zigzag data: Comparing the results of different dimensionality reduction methods for increasing numbers of neighbours and dimensionalities of the transformed space

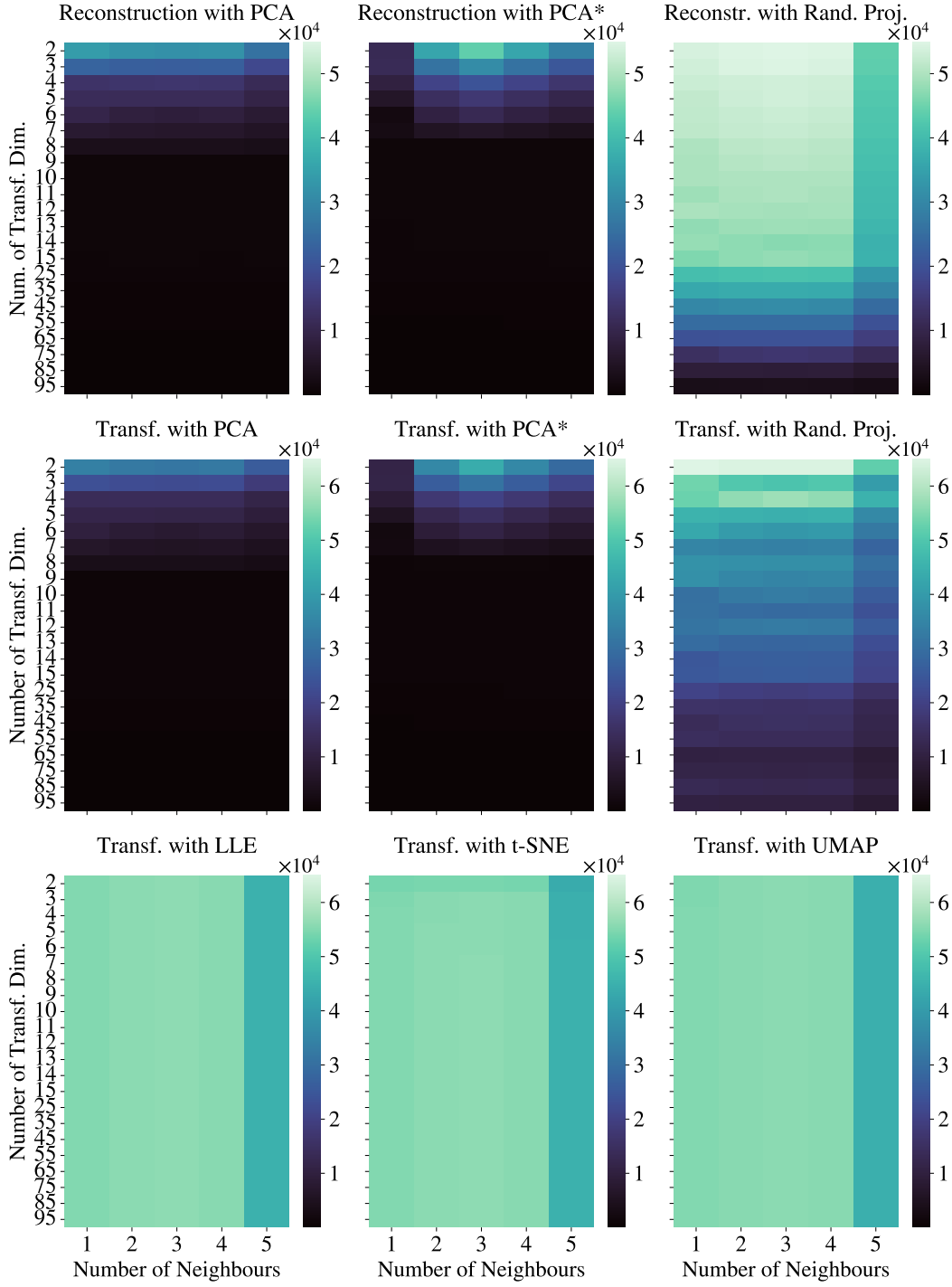


Figure 4.33: MNR measure for zigzag data: Comparing the results of different dimensionality reduction methods for increasing numbers of neighbours and dimensionalities of the transformed space

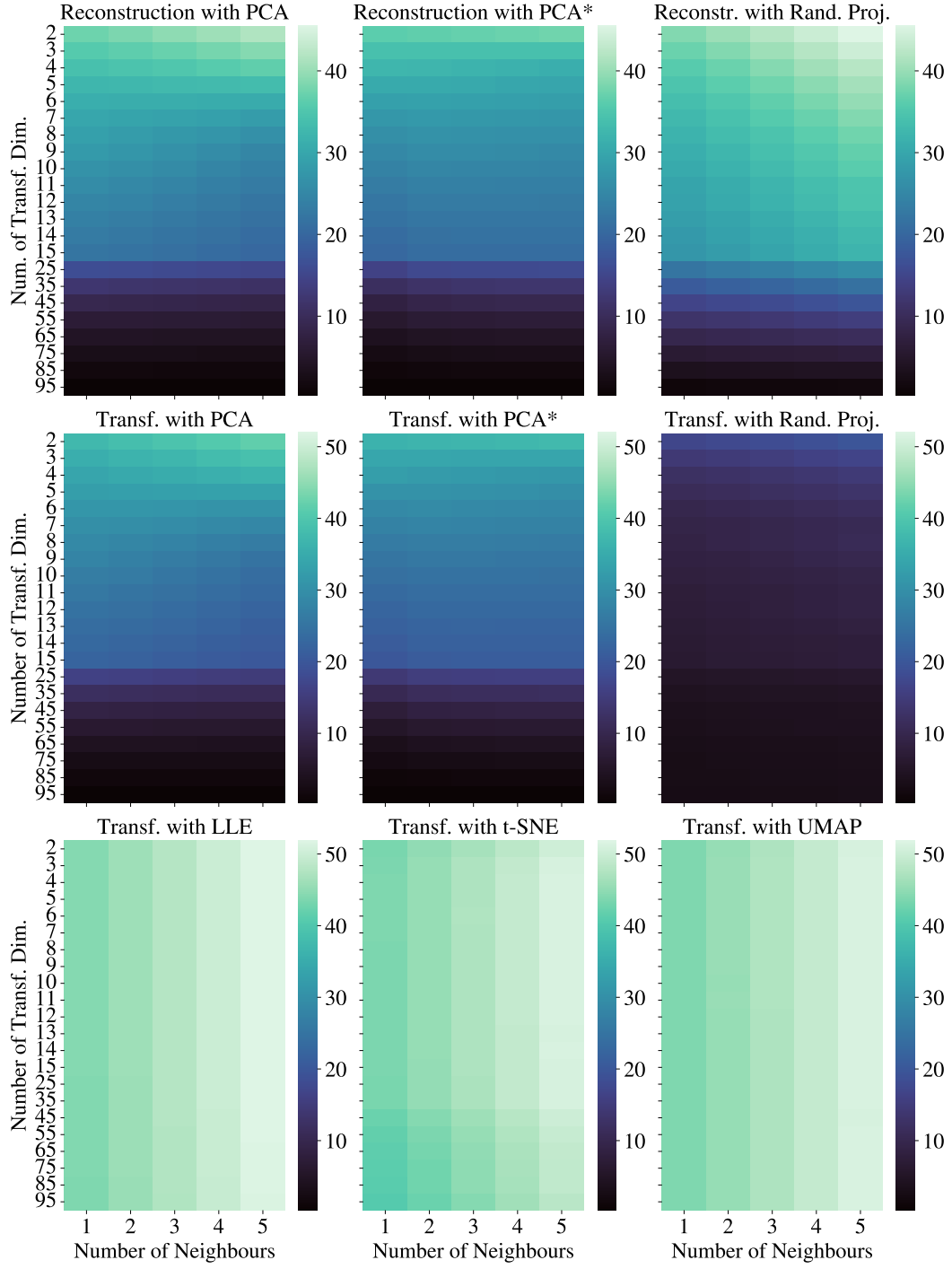


Figure 4.34: ND measure for smoothly ordered data: Comparing the results of different dimensionality reduction methods for increasing numbers of neighbours and dimensionalities of the transformed space

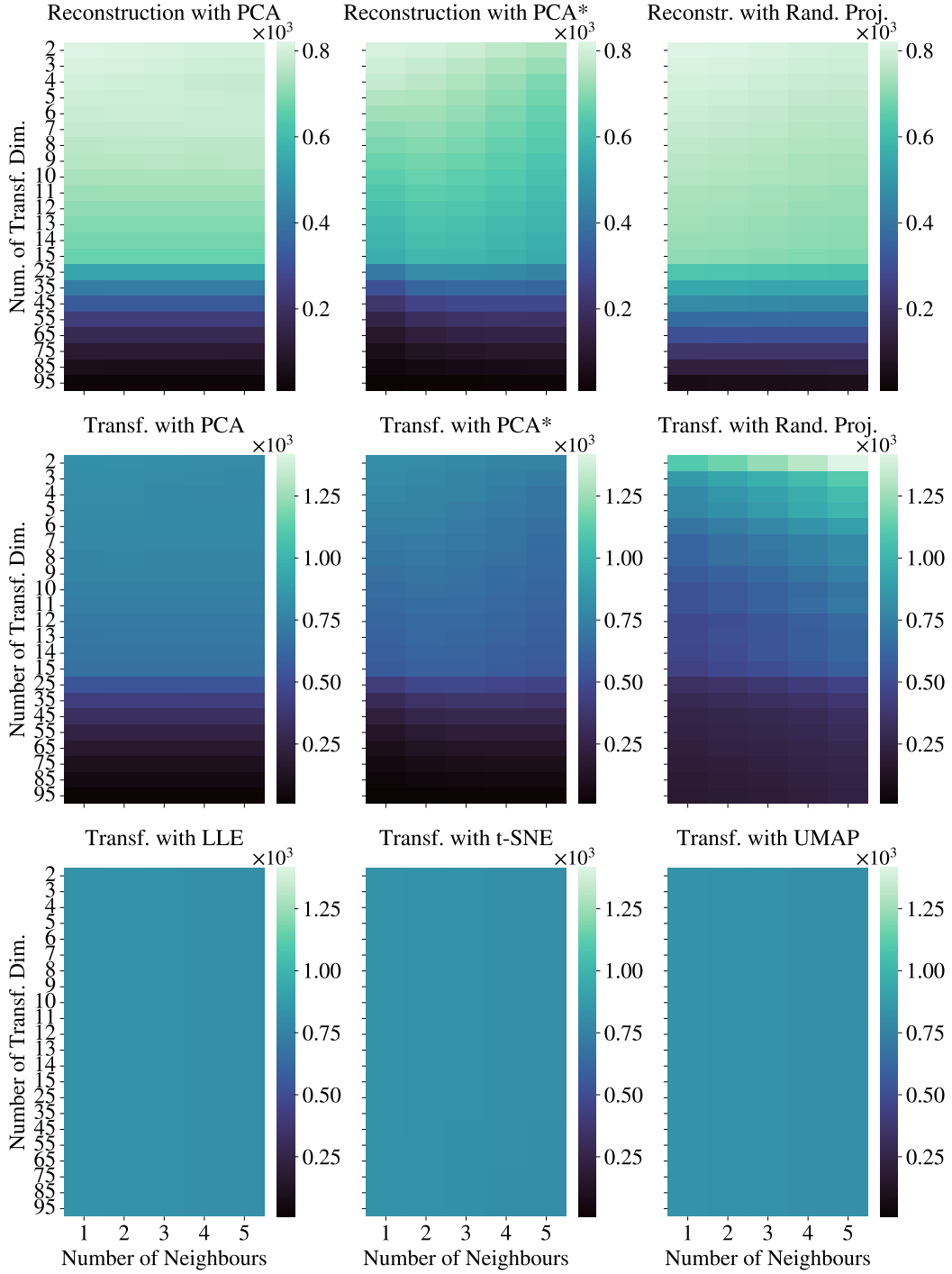


Figure 4.35: MNR measure for smoothly ordered data: Comparing the results of different dimensionality reduction methods for increasing numbers of neighbours and dimensionalities of the transformed space

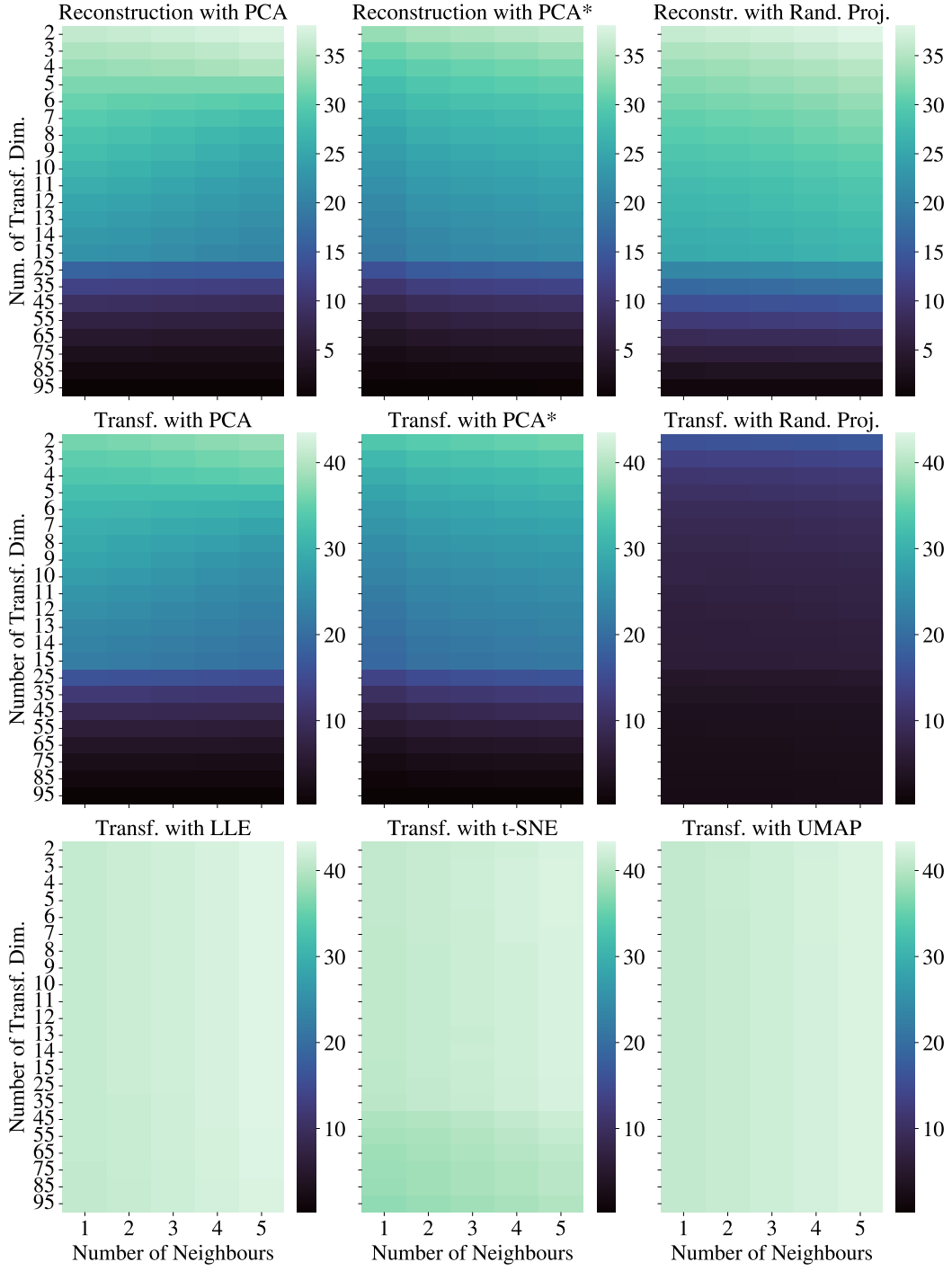


Figure 4.36: ND measure for data in separate lines: Comparing the results of different dimensionality reduction methods for increasing numbers of neighbours and dimensionalities of the transformed space

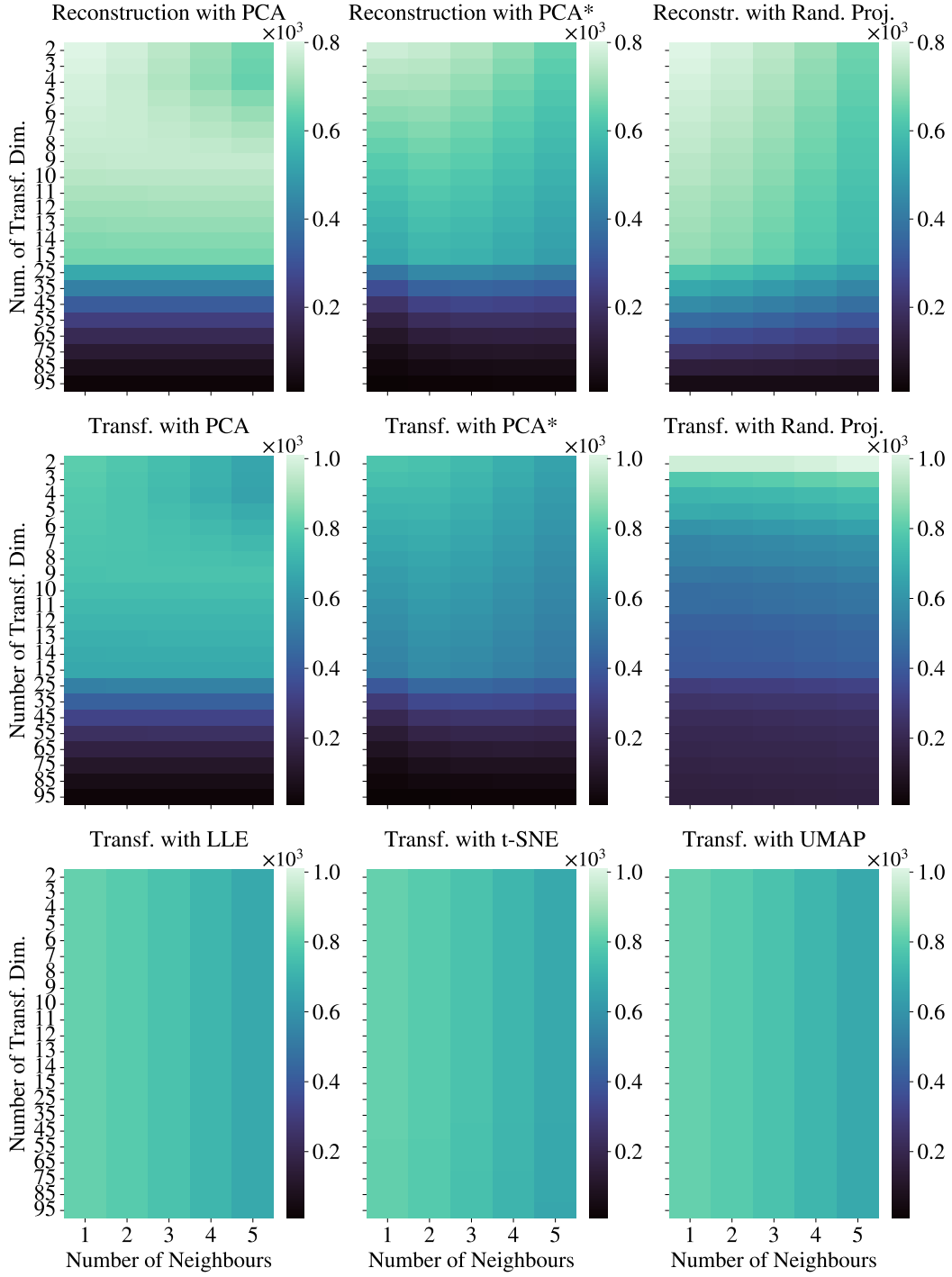


Figure 4.37: MNR measure for data in separate lines: Comparing the results of different dimensionality reduction methods for increasing numbers of neighbours and dimensionalities of the transformed space

4.5.5 Number of Lines Versus the Number of Neighbours

To explore the impact of data complexity on the performance of dimensionality reduction methods, we generate datasets comprising 100-dimensional data points organised on a gradually increasing number of lines. As the number of lines increases, the data's complexity also grows, allowing us to examine how different dimensionality reduction methods cope with varying levels of data intricacy. We then apply various dimensionality reduction techniques to these datasets and assess their performance using the quality measures outlined in Section 4.2: Neighbour-Distance (ND) and Multiple-Neighbour-Relation (MNR). Additionally, we analyze the results for different numbers of neighbours. This experimental setup enables us to identify parameter combinations that provide insights into which dimensionality reduction method best preserves order-based distances and relations for an increasing number of neighbours in increasingly complex scenarios.

We begin our analysis by examining the results of dimensionality reduction techniques applied to zigzag data using the ND measure. The results are visualised in Figure 4.38. It becomes evident that PCA* demonstrates superior performance in preserving distances between each data point and its predecessor and successor when compared to PCA, random projection as well as LLE, t-SNE, and UMAP. Apart from this, PCA* has a similar result to PCA. Random projection struggles with order-based distance preservation in the reconstructed space but displays similar results to PCA in its transformation. Additionally, Figure 4.38 illustrates that t-SNE yields better results than both LLE and UMAP but remains unable to match the order-based distance preservation achieved by PCA and PCA*. This observation aligns with the local structure-preserving nature of these techniques, as they aim to maintain distances that reflect the local context.

Figure 4.39 presents results for the dimensionality reduction methods to zigzag data, using the MNR measure to assess order-based relation preservation across varying numbers of neighbours and lines in both the reconstructed and transformed representations. PCA* outperforms PCA in retaining relationships with the first neighbours on the order for both directions, while exhibiting similar overall performance. In contrast, random projection fares less favourably, showing reduced preservation of order-based distances in both higher and lower-dimensional representations compared to PCA and PCA*. Random projection, LLE, t-SNE, and UMAP perform well when data is distributed along a single line but struggle to maintain relationships in the presence of complex zigzag patterns. This is consistent with their strengths in preserving local distances.

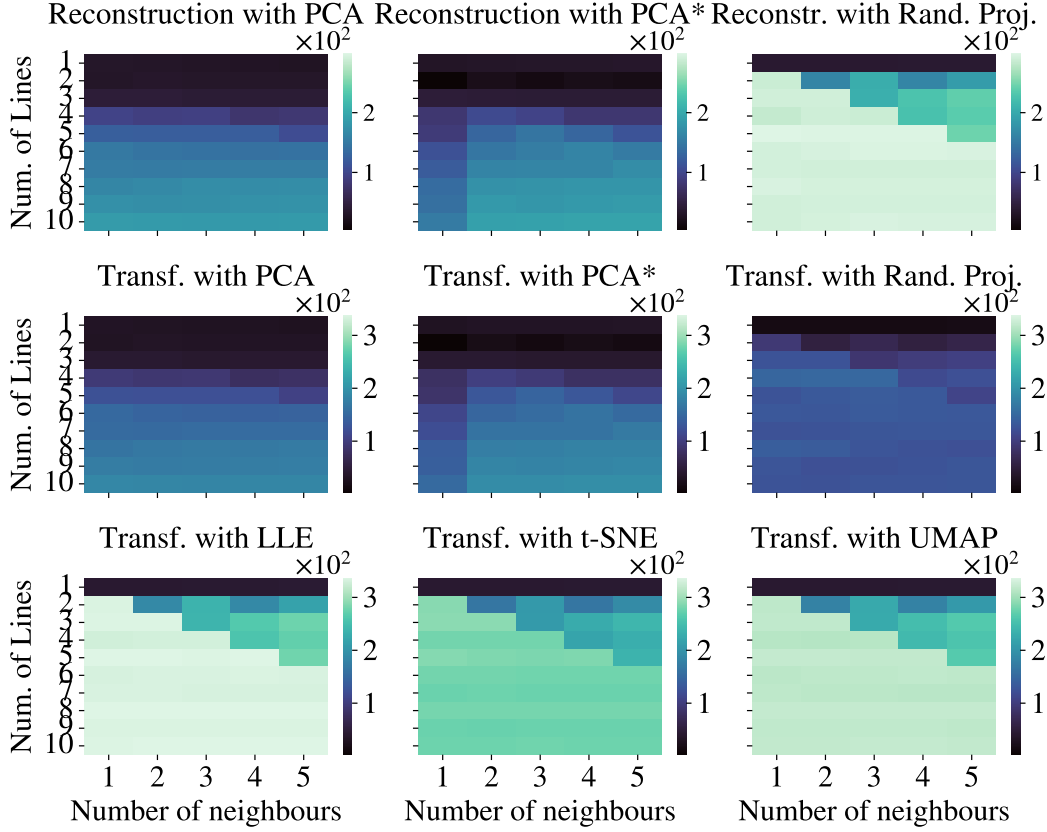


Figure 4.38: Zigzag data with ND measure: comparing the results of different dimensionality reduction methods for varying numbers of neighbours and numbers of lines

Applying our analysis to data ordered connected with a smooth line, we observe generally better results for PCA* when there are more than two lines, as shown in Figure 4.40. PCA outperforms PCA* for one and two lines. However, as the number of lines increases, PCA's performance declines. In contrast, random projection maintains the order-based distances less effectively in its reconstructed form but excels in its lower-dimensional embedding, a pattern consistent with other settings. LLE, t-SNE, and UMAP exhibit poor performance in preserving order-based distances as the number of lines and neighbours grows.

Figure 4.41 reveals a significant difference between PCA, PCA* and random projection. While random projection's results are roughly comparable to PCA* in the higher-dimensional space, its respective transformation shows poor results compared to PCA and PCA*. Comparing the results of PCA and

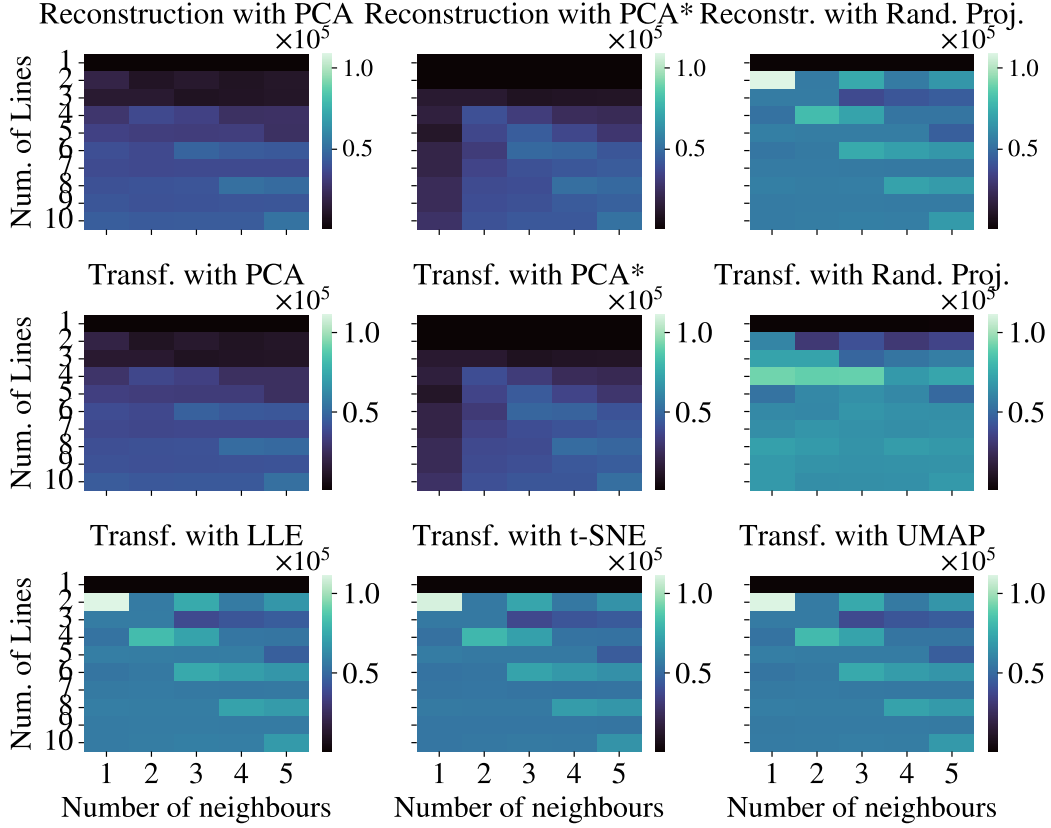


Figure 4.39: Zigzag data with MNR measure: comparing the results of different dimensionality reduction methods for varying numbers of neighbours and numbers of lines

PCA*, we see that PCA* outperforms PCA for one line and more neighbours but resembles more and more the result of PCA as the number of lines and neighbours increases. Similarly, LLE, t-SNE and UMAP also perform well in this experiment and are thereby comparable to PCA*.

Similar to the results of ND measure on data which is connected with one line, the results of applying various dimensionality reduction methods on data which is ordered on separate lines, show that PCA* generally better maintains the order-based distances than PCA, see Figure 4.42, except from data being in only one line. Random projection, as previously, exhibits worse results in the original space with its reconstructed data, but in the lower-dimensional space, it performs well. LLE, t-SNE, and UMAP also display equally poor results compared to PCA, PCA*, and random projection.

For the same experimental setting but using the MNR measure, we observe

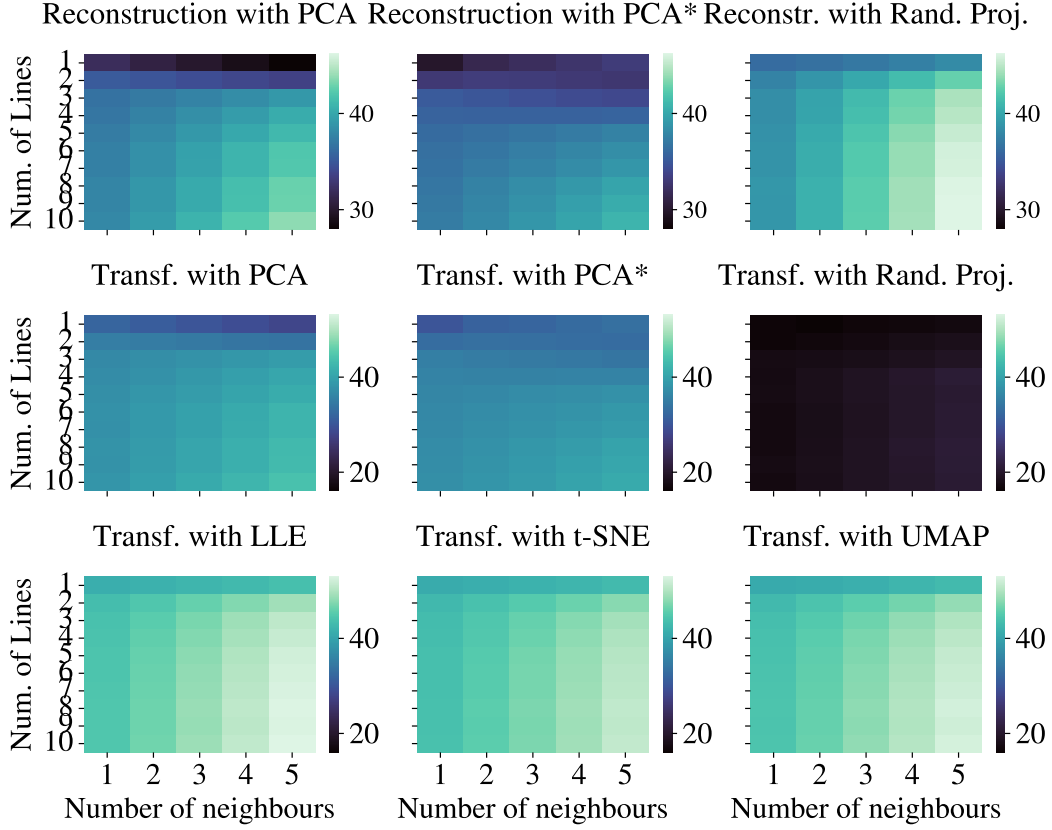


Figure 4.40: Smoothly ordered data with ND measure: comparing the results of different dimensionality reduction methods for varying numbers of neighbours and numbers of lines

marginally better results for PCA* compared to PCA for 3 or more lines, with PCA performing better on one and two lines, as seen in Figure 4.43. Random projection demonstrates similar results in the original space to PCA but performs better for one or two lines. In the transformed space, it exhibits poor results, as it fails to retain the relations between connections as effectively as PCA and PCA*. Additionally, each of the tested methods performs slightly better for a lower number of neighbours.

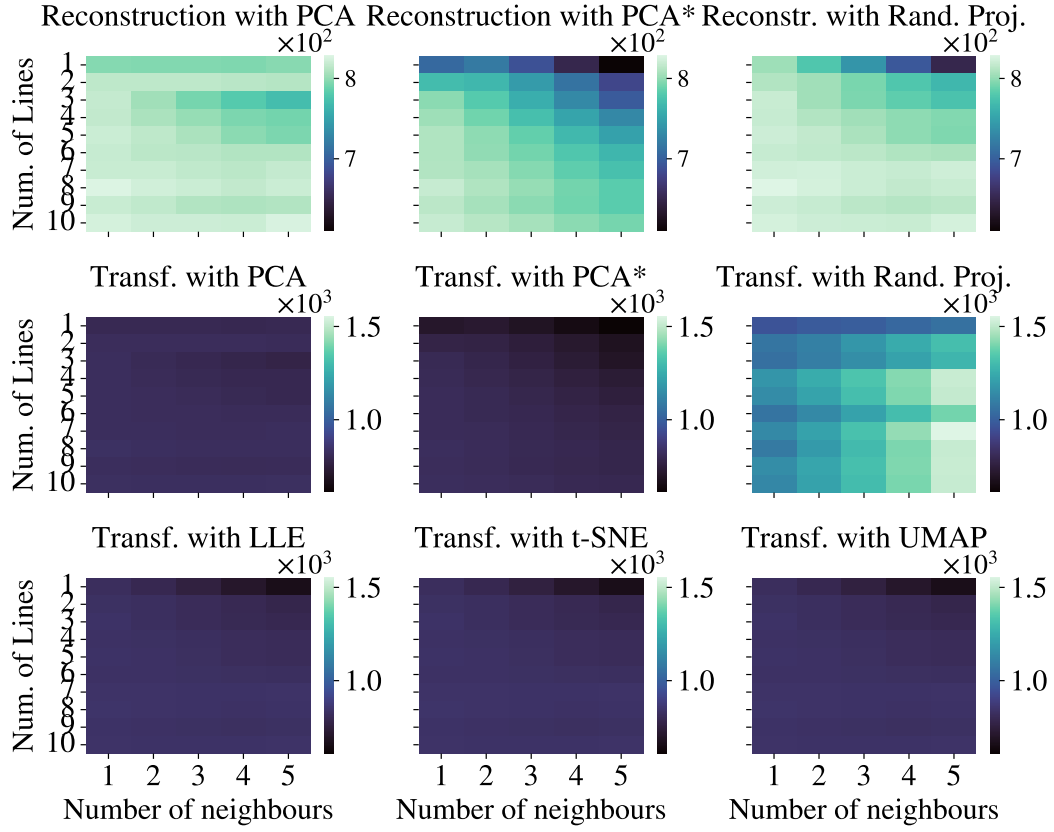


Figure 4.41: Smoothly ordered data with MNR measure: comparing the results of different dimensionality reduction methods for varying numbers of neighbours and numbers of lines

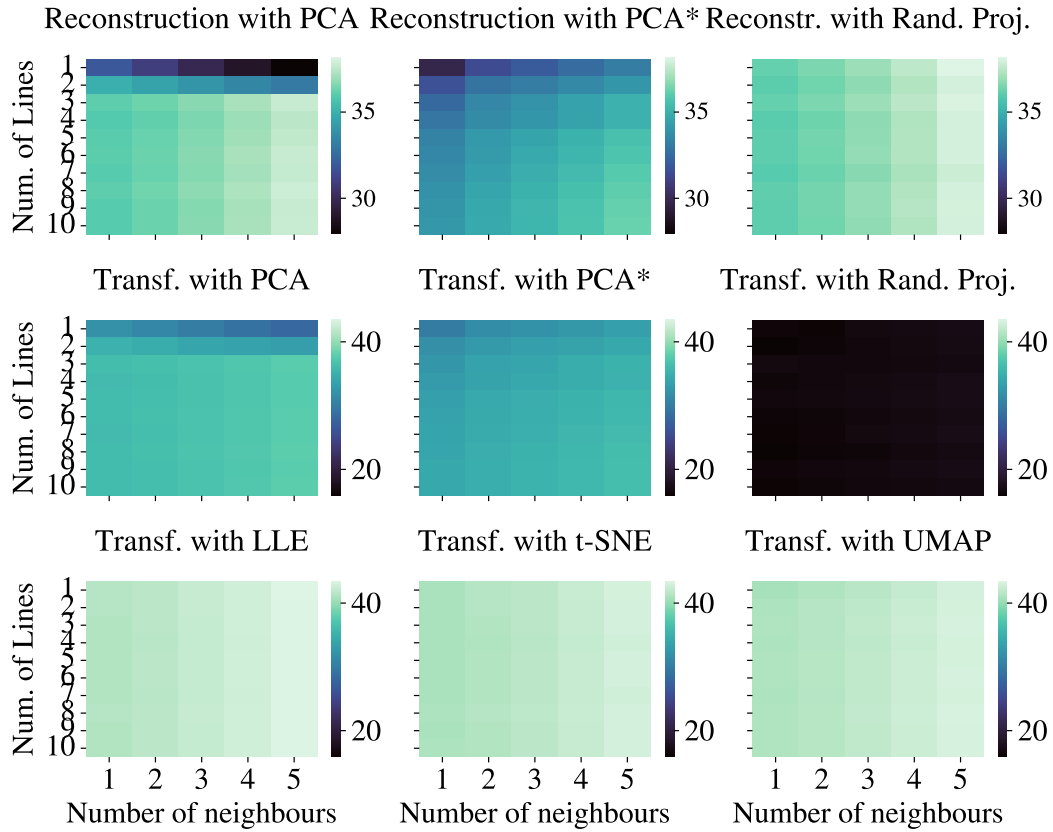


Figure 4.42: Separate lines data with ND measure: comparing the results of different dimensionality reduction methods for varying numbers of neighbours and numbers of lines

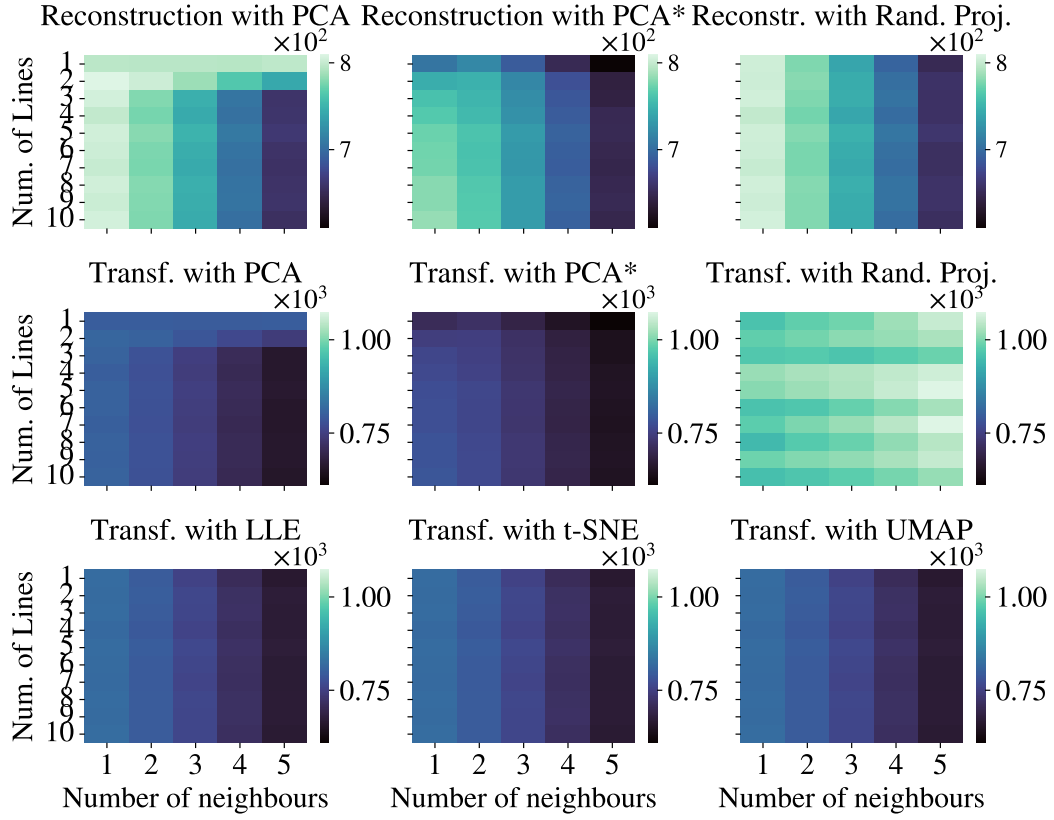


Figure 4.43: Separate lines data with MNR measure: comparing the results of different dimensionality reduction methods for varying numbers of neighbours and numbers of lines

4.6 Evaluation on Real-World and Simulated Data

Due to noise, missing data points or outliers, real-world and simulated data may often be more complex than the generated data. In this section, we expand our analysis to three real-world datasets: Russell2000, MTS Flights and air pollution data in Japan. These datasets vary in dimensionality, ranging from very high dimensional (over 1000 dimensions) to lower dimensions (29). For this analysis, we will primarily focus on using PCA* as it has shown promising results in previous experiments, outperforming both PCA and the variants of CPCA.

4.6.1 Expanding the Dimensionality of the Transformation

In the following, the average deviation from each reconstruction and transformation of real-world and simulated data in comparison to the original data. Average deviation is defined as the mean of all summed differences between each data point and its predecessor and successor. As we have seen in Subsection 4.5.2, the results of LLE, t-SNE and UMAP remained constant. Therefore, we omit these techniques in this experiment.

In Figure 4.44, we observe a notable difference in the variances and mean values between random projection, PCA and PCA*. Random projection exhibits a significant overall spread, indicating a wider range of variability in the results. Both embeddings of random projection diverge much more on average from the original data than the results of PCA and PCA*. When we focus on the comparison between PCA and PCA*, we observe that PCA* performs marginally better than PCA. This implies in both terms of mean and variance suggesting that PCA* maintains slightly better pairwise distances within the data. On the other hand, comparing the results of the methods with the NR measure we see PCA* demonstrating improvement over PCA in ranges 2 to roughly 480. Beyond this range, PCA continues to excel in maintaining relationships among data points. In all scenarios within this experiment, it is evident that both PCA and PCA* exhibit a sharp improvement as the dimensionality increases up to 400. This is due to the dataset having more dimensions than data points and consisting of 504 observations, enabling PCA and PCA* to effectively preserve order-based distances well. Interestingly, the outcome of random projections only exhibits a gradual decrease, even when transforming to a higher dimension than the dataset contains data points.

When applying the dimensionality reduction methods to the MTS dataset,

we observe remarkably high performances achieved by both PCA and PCA*. The exceptional results may be due to the inherent limitations of this dataset, see the paragraph on real-world and simulated data in Section 4.1. Notably, PCA* constantly outperforms PCA, suggesting a slightly better preservation for both distances and relations among data points. In contrast, random projection exhibits consistently suboptimal results across all dimensions to which the data was reduced, as illustrated in Figure 4.45.

When we transform the air pollution data to an increasing number of dimensions, we observe that random projection consistently produces poor results when comparing the relations of its reconstructed representation to the original data, see Figure 4.46. However, when comparing the transformed data, we find that it yields better results than PCA and PCA*. This scenario resembles the setting in which we applied the dimensionality reduction methods to smoothly ordered data. This similarity suggests that this dataset may exhibit lower fluctuations. Across all settings, PCA* consistently outperforms PCA in preserving relationships and distances between consecutive data points.

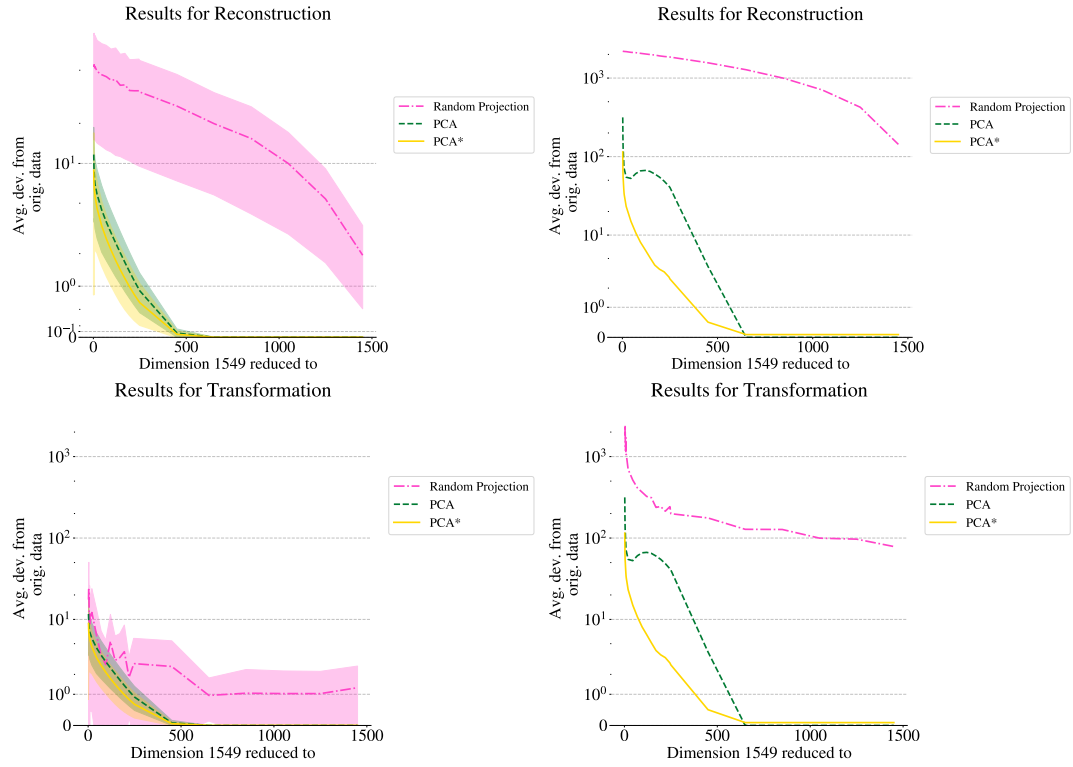


Figure 4.44: Russell2000 data: Comparing the results of different dimensional-ity reduction methods for an increasing number of transformation dimension-ality

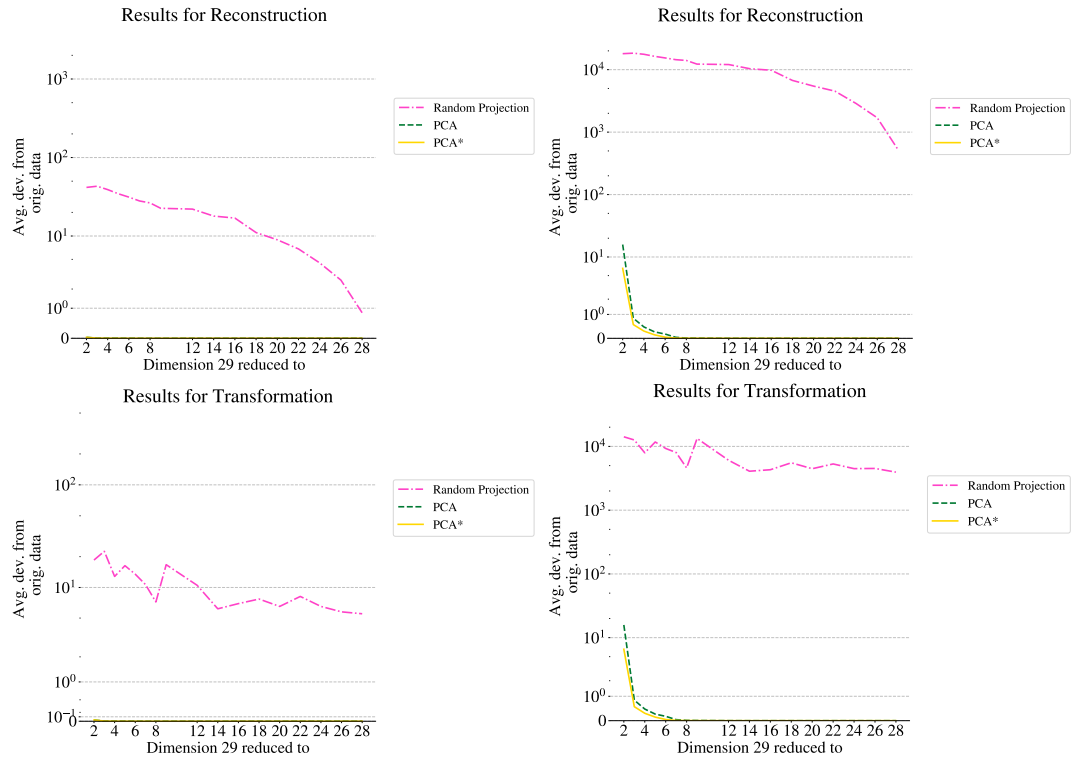


Figure 4.45: MTS flights data: Comparing the results of different dimensionality reduction methods for an increasing number of transformation dimensionality. Variance is omitted due to visibility reasons.

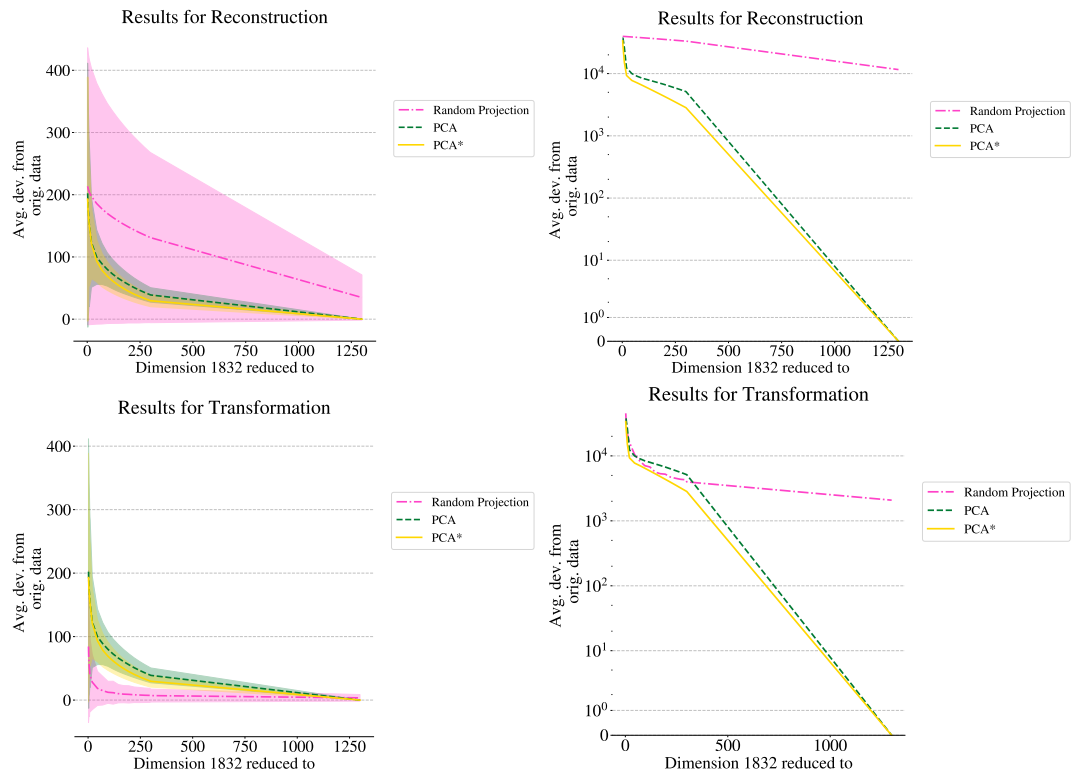


Figure 4.46: Air-pollution data: Comparing the results of different dimensionality reduction methods for an increasing number of transformation dimensionality

4.6.2 Exploration of Increasing Numbers of Neighbours

Similar to Subsection 4.5.3, we now investigate how PCA, PCA*, and random projection reduce the dimensionality of real-world data from their original high dimensionalities to 2 dimensions. We then quantify, for an increasing number of neighbours in each sequence direction, how effectively these methods preserve both distances and relations. We use the Neighbour-Distance (ND) to measure how well distances are kept and the Multiple-Neighbour Relation (MNR) for assessing the preservation of relationships in this analysis. These measures are described in Section 4.2.

Analogously to the separate lines, we measure the distances and relations within each flight of the MTS-flights data separately.

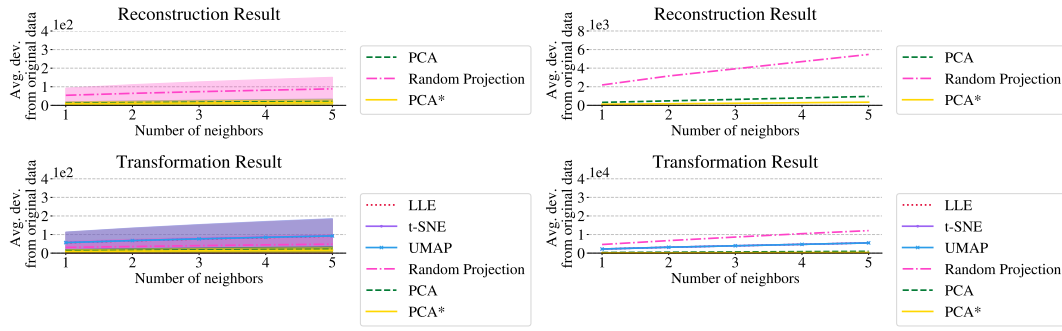


Figure 4.47: Russell2000 data: Comparing the results of different dimensionality reduction methods for varying numbers of neighbours

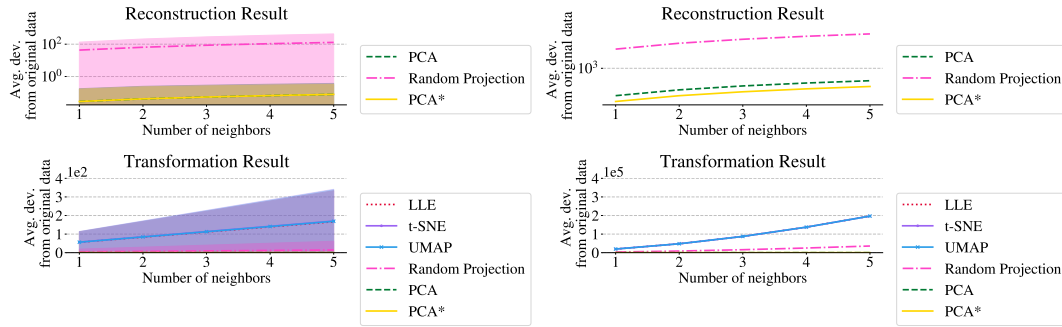


Figure 4.48: MTS flights: Comparing the results of different dimensionality reduction methods for varying numbers of neighbours

When applying dimensionality reduction methods to both the Russell2000 and the MTS flight data, we observe similar outcomes, as depicted in Fig-

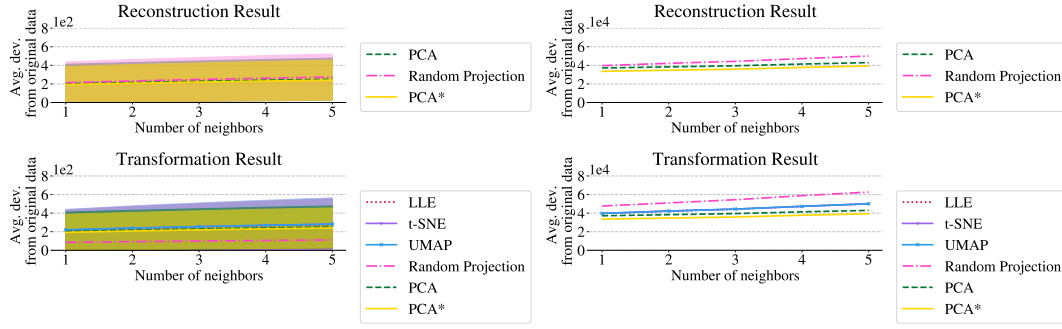


Figure 4.49: Air-pollution data: Comparing the results of different dimensionality reduction methods for varying numbers of neighbours

ure 4.47 and Figure 4.48. Notably, random projection exhibits poor performance in preserving distances and relations between different numbers of successors and predecessors, in contrast to PCA and PCA*. Interestingly, LLE, t-SNE and UMAP yield even worse results than random projection. However, PCA* demonstrates a marginal improvement over PCA in preserving distances and significantly outperforms both PCA and other methods in maintaining relations within the data.

On the air pollution data, every dimensionality reduction method maintains the distances within the data similarly, apart from the lower-dimensional transformation of random projection, which keeps distances almost constantly low for all increasing neighbours, as illustrated in Figure 4.49. On the other hand, comparing the relations within both embeddings with the relations within the original data shows that PCA* is the best dimensionality reduction method and random projection the worst.

4.6.3 Analysis of Growing Numbers of Neighbours and Transformation Dimensions

Analogous to previous subsections, we explore the effects of changing two key parameters in the transformation and reconstruction evaluation: the number of successors and predecessors, and the number of dimensions used to transform the data. In every dimensionality reduction step, we quantify the order-based distance and relation preservation for a growing number of neighbours on the sequence with the Neighbour-Distance (ND) and Multiple-Neighbour-Relation (MNR) measure compared to the distances and relations in the original data, see Section 4.2. By comparing the results with the original data, we gain insights into the trade-offs between order-based distance and relation preser-

vation when applying dimensionality reduction techniques to sequential data.

Comparing the quality of the results of dimensionality reduction methods that transformed the Russell2000 data to an increasing number of dimensions juxtaposed to the number of neighbours examined, we see similar results measured with ND and MNR measures, see Figure 4.50 and 4.51. PCA* marginally improves the order-based distances and relations within the data of PCA, see figures 4.52(a) and 4.52(c).

Analogously to previous experiments, random projection maintains order-based distances and relations better than its reconstructed counterpart. LLE, t-SNE and UMAP are comparably suboptimal, as they maintain the distances and relations to the first neighbours better and then decrease in quality the more neighbours we observe.

When we apply dimensionality reduction methods to the MTS flight data, we observe similar results for both PCA and PCA* when assessed with ND and MNR measures, as illustrated in figures 4.53 and 4.54. For a more detailed comparison between PCA and PCA* figures 4.52(b) and 4.52(d) depict their results with a separate scale. With the ND measure, both PCA and PCA* perform similarly. However, when using the MNR measure, we observe that PCA* outperforms PCA, especially for transformations to two dimensions. The reconstruction of random projects underperformed compared to PCA and PCA*. However, its transformations, while not as good as those of PCA and PCA*, are still better than the results achieved by LLE, t-SNE, and UMAP. It's worth noting that t-SNE encountered issues with the flight dataset, particularly when the dataset contained rows with a standard deviation of zero. This limitation highlights one of the disadvantages of t-SNE, as other dimensionality reduction methods were able to handle this dataset successfully.

Figure 4.55 shows comparable results for PCA and PCA*, with PCA* performing slightly better for the first two neighbours. The overall pattern is consistent with the observations made in the Russell2000 data scenario.

PCA* outperforms all other dimensionality reduction methods in all settings when applied to the air pollution data assessed with the MNR measure, see Figure 4.56.

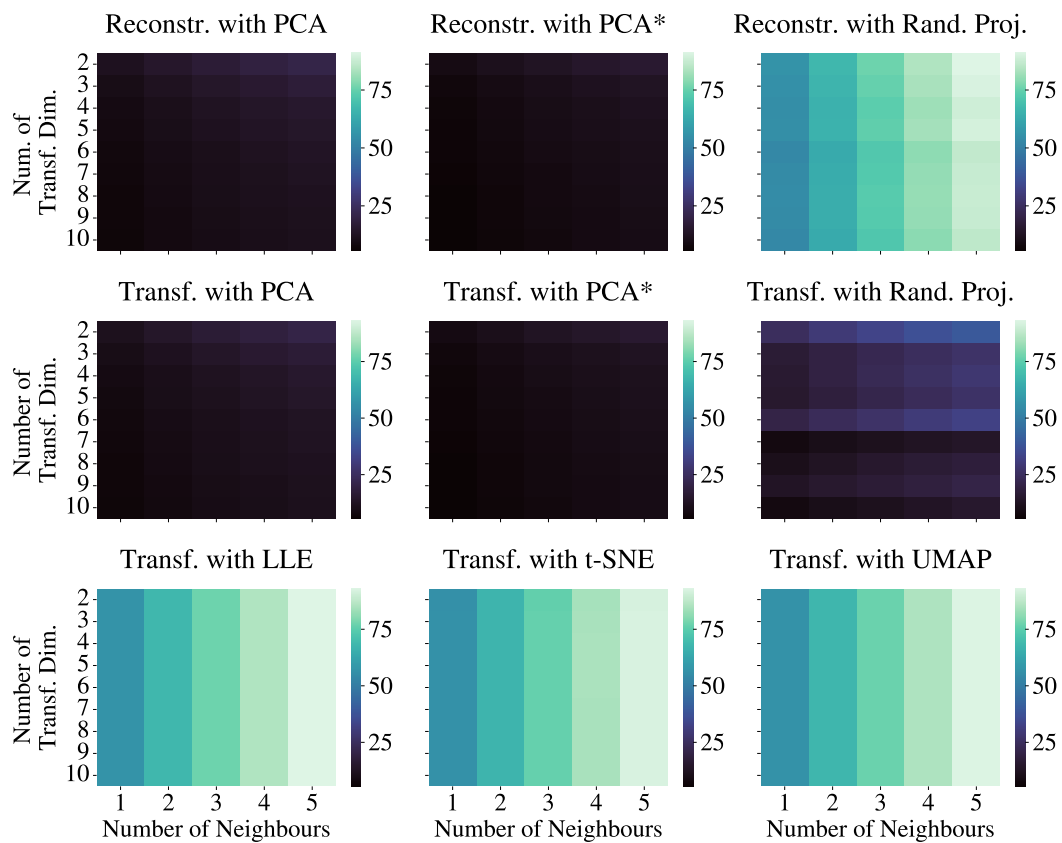


Figure 4.50: Neighbour-Distance measure on Russell2000 data: Comparing the results of different dimensionality reduction methods for increasing numbers of neighbours and dimensionality of the transformed space with the original data

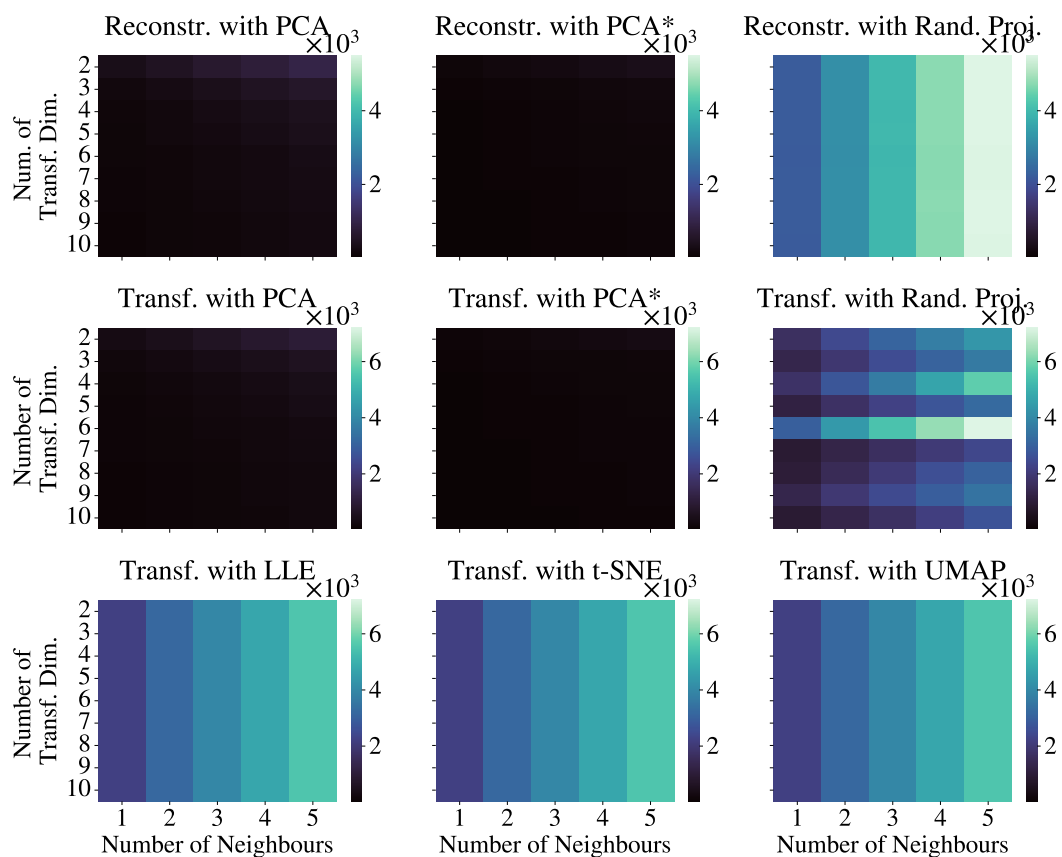


Figure 4.51: Multiple-Neighbour-Relation measure on Russell2000 data: Comparing the results of different dimensionality reduction methods for increasing numbers of neighbours and dimensionality of the transformed space with the original data

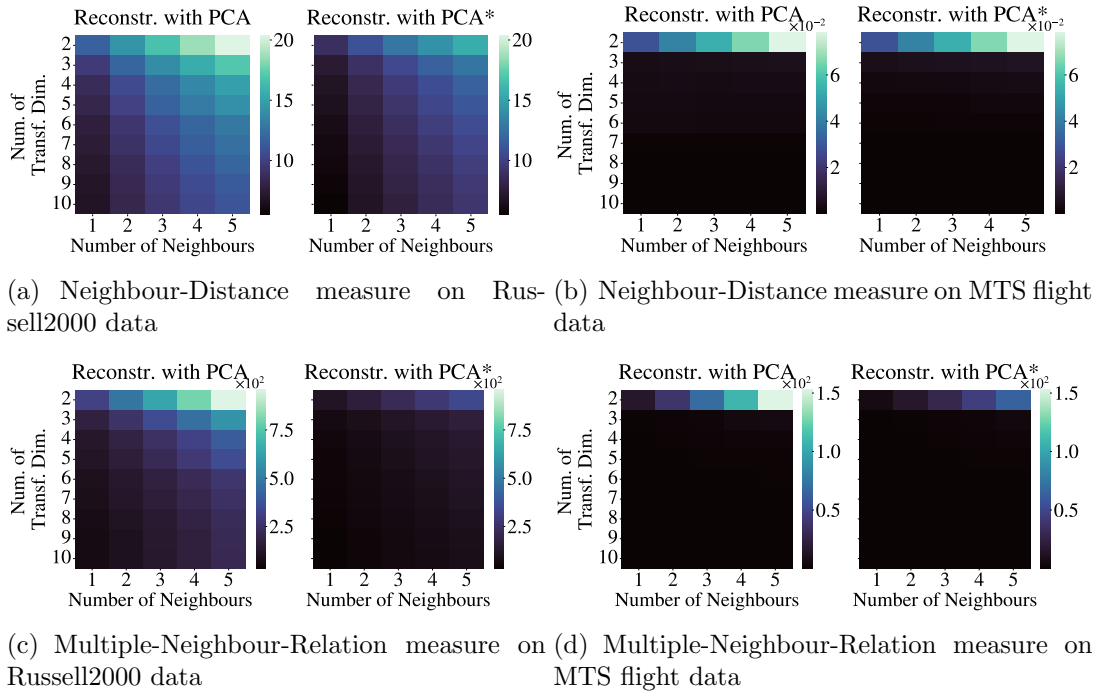


Figure 4.52: Comparing the results of PCA and PCA* separately for increasing numbers of neighbours and dimensionality of the transformed space with the original data

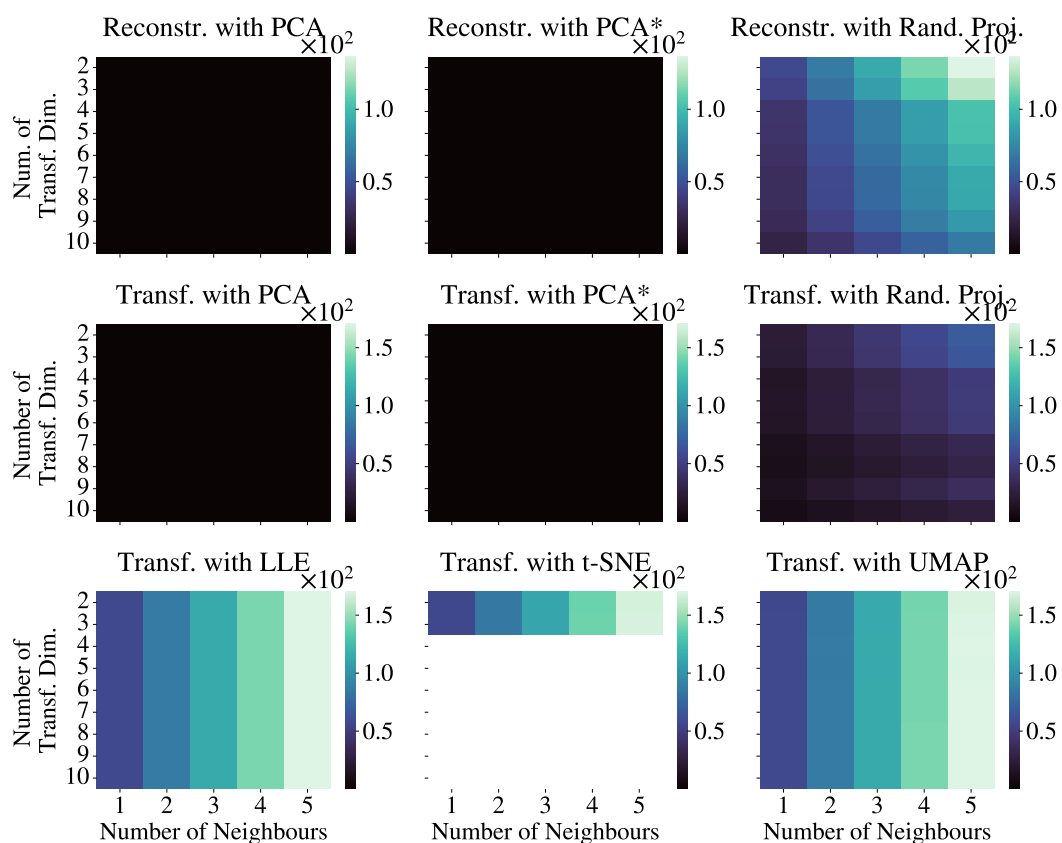


Figure 4.53: Neighbour-Distance measure on MTS flights data: Comparing the results of different dimensionality reduction methods for increasing numbers of neighbours and dimensionality of the transformed space with the original data

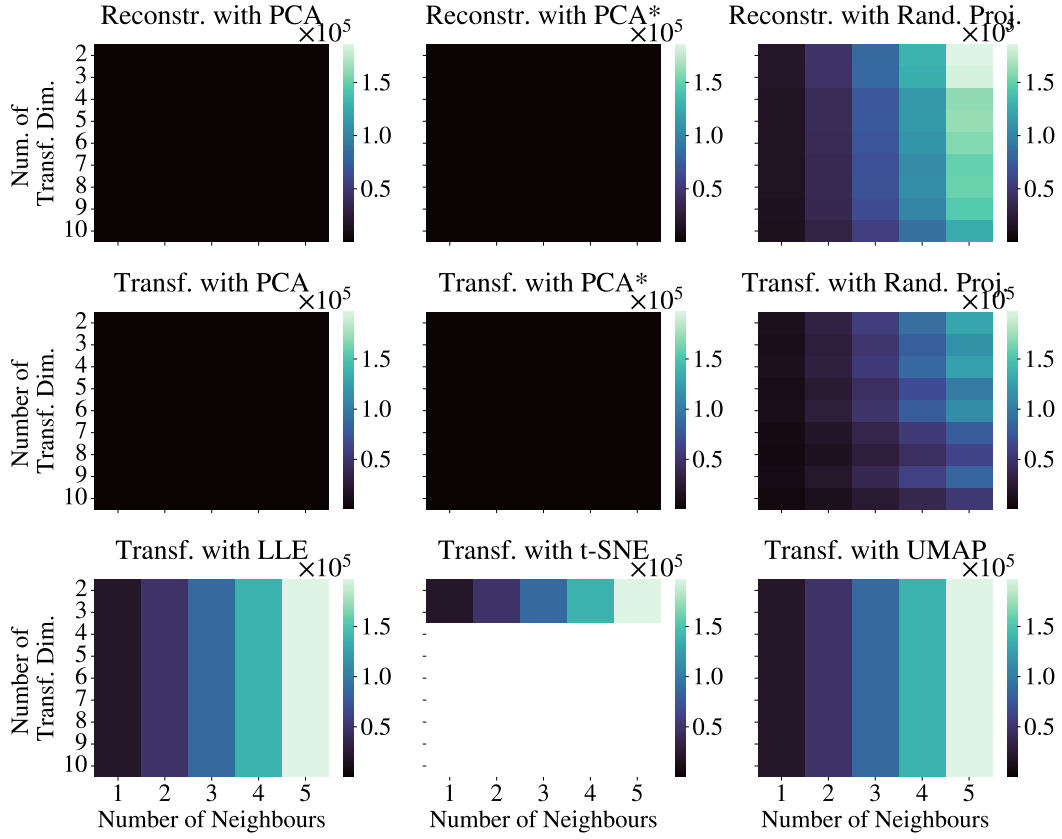


Figure 4.54: Multiple-Neighbour-Relation measure on MTS flights data: Comparing the results of different dimensionality reduction methods for increasing numbers of neighbours and dimensionality of the transformed space with the original data

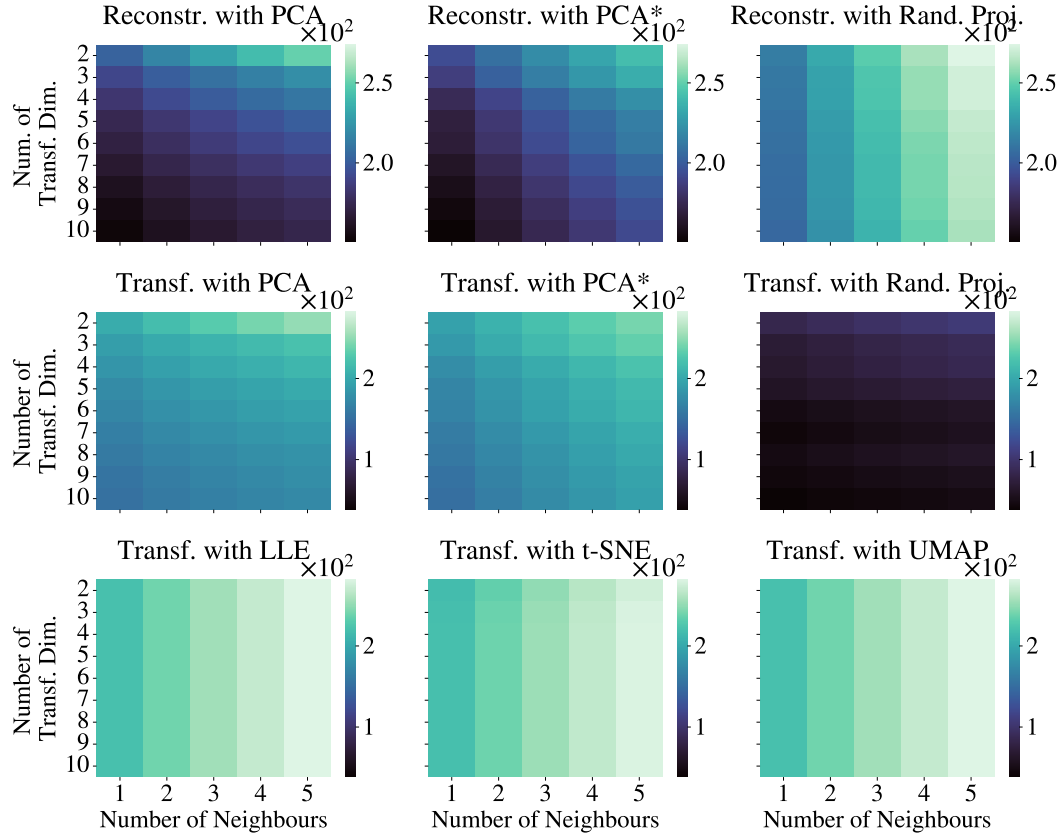


Figure 4.55: Neighbour-Distance measure on air-pollution data: Comparing the results of different dimensionality reduction methods for increasing numbers of neighbours and dimensionality of the transformed space with the original data

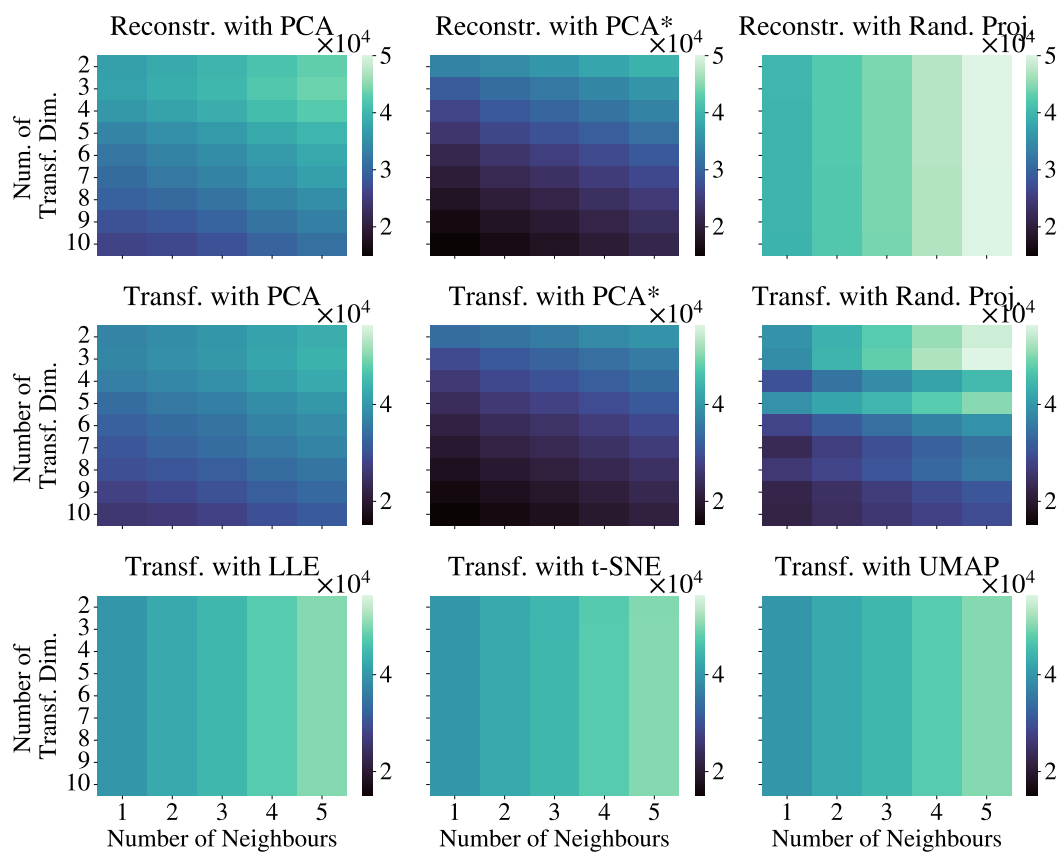


Figure 4.56: Multiple-Neighbour-Relation measure on air-pollution data: Comparing the results of different dimensionality reduction methods for increasing numbers of neighbours and dimensionality of the transformed space with the original data

4.7 Limitations of Experiments

In our research, we introduce a new setting focused on order-based distance preservation. It is important to note that as this field is new and broad, our experiments represent only a subset of the possible scenarios that can be tested in this setting.

Furthermore, we acknowledge that the toy examples used in our experiments have limitations. Specifically, they are designed to begin in a relatively simplified setting: consisting of 5 lines, each containing 100 data points.

It is also worth highlighting that in the real world, data may not always exhibit the same patterns as our toy example. For instance, the strict zigzag ordering or the smooth ordering in our experiments may not accurately reflect the complexity and diversity of real-world data distributions. They were used to simulate extreme orders like zigzag data simulating scenarios with high fluctuations, whereas smoothly ordered data represented data, which consists of the same data points, but without the rapid variations.

We also conducted tests on three real-world datasets. These datasets used to evaluate our methods exhibited diverse characteristics, like varying in terms of dimensionality. Furthermore, these datasets also had a varying number of observations, indicating different sample sizes. Additionally, one of the datasets was composed of multiple time series data, which were all measured separately. While this provides valuable insights, we recognize that the diversity of real-world data is vast, and these three datasets are only a small part of what exists. Nevertheless, they offer a promising outlook for the applicability of our methods in more complex and varied scenarios.

In the case of CPCA, where we augmented the data with vectors as the imaginary part to account for the sequential information, it is important to note that certain challenges arise. These challenges become evident when dealing with consecutive non-unique data points, as adding a 0-vector to the data may signal the end of a sequence. Given that our approach allows for the presence of multiple sequences with the data, handling such cases requires careful consideration and adaptation of the CPCA techniques.

4.8 Experimental Results

In the series of experiments, we explored dimensionality reduction techniques, with a particular focus on order-based distance preservation. Our investigations covered a spectrum of datasets, including synthetic and real-world data, of which the data varied in order and complexity. The primary focus was to assess the performance of PCA* and the variants of CPCA in terms of their ability to preserve order-based distances and relationships within the data.

Our initial experiments involved synthetic data, which allowed us to control the data's characteristics. We discovered that all variants of CPCA maintained the distances and relations suboptimal in comparison to PCA. PCA on data with Ids performed comparably to PCA. In our second experiment, we compared PCA* with various other dimensionality reduction methods. We saw that PCA* exhibited consistently a good preservation for data with high fluctuations, whereas the lower dimensional representation of random projection on smoothly ordered data and data in separate lines (because it did not fluctuate as much) maintained exceptionally well the compared distances. In contrast, its reconstruction is poor compared to the reconstructions of PCA* and PCA.

We applied our dimensionality reduction methods to real-world datasets, including financial data from the Russell2000, flight data from NASA MTS dataset and data on air pollution in Japan. Here, too PCA* emerged as a robust performer, consistently outshining PCA in preserving relationships and distances between consecutive data points. Random projection, although stable, lagged behind in terms of maintaining these order-based characteristics, apart from applying it to the air pollution data, where similarly to the smoothly ordered data, its transformation kept more order-based distances.

Chapter 5

Future Work

As the results of our experiments lay a foundation for future explorations in the field of dimensionality reduction and order-based distance preservation, we give some suggestions on what we believe PCA* and order-based distance preservation, in general, have the potential to enhance various data analysis tasks.

Clustering PCA can be used in clustering applications, see [39]. Our experiments have shown the promise of order-based distance preservation of PCA*. Future research may apply PCA* specifically for clustering tasks, where maintaining the intrinsic sequential order of data can be critical for effective clustering.

NLP and Text Analysis In the realm of Natural Language Processing (NLP), sentences are often represented as vectors [24]. Given that sentences inherently possess sequential order, the application of order-based distance preservation techniques holds the potential to enhance the accuracy and interpretability of NLP models.

Multiple Sequential Orders While our experiments focused on data with a single clear order, real-world scenarios often involve multiple orders within the same dataset. For example, molecular dynamics or pixel data, where both structural and temporal orders coexist. Future investigations may extend PCA* to accommodate and preserve these multiple orders effectively.

Partially Ordered Data We have primarily focused on data which is totally-ordered. In future work, it is interesting to investigate the application of PCA* to data that is not linearly ordered but exhibits partial order, such as a hierarchy or graphs. This exploration can give insight into how our order-based

distance-preserving technique performs in scenarios where the data's structure is more complex and includes branches. Therefore, testing PCA* on this scenario can provide more information about its versatility and effectiveness on a broader range of datasets.

Variable-Length Data Analysing the data with multiple sequences of varying lengths poses an interesting challenge. We primarily focus in this work on equally long sequences, but as the MTS data shows, real-world data may not always be so similar.

Further Explorations for CPCA While order-based distance preservation is not the primary focus for the variants of CPCA, there may be a specialised use case where its incorporation may be useful. Future research may uncover these applications and refine the methods for this scenario.

Order-Based Distance Preservation in Local Structure Preserving Methods As LLE, for example, employs the k -nearest neighbour algorithm, to find the spatially nearest neighbours of each point, it would be interesting to explore its adaptation to aiming for order-based distance preservation. Here we would not be taking the k -nearest neighbours in the space but the sequential k -nearest neighbours. This adaptation could lead to a representation of data which maintains the inherent order better than LLE currently.

Chapter 6

Conclusion

In this thesis, we investigate dimensionality reduction methods applied to ordered data. We introduce and analyse variants of PCA tailored to ordered data, including PCA with Ids, Complex PCA on ordered data and PCA*.

Our experiments, conducted with synthetic and real-world datasets, provide valuable insights into the performance of these techniques. We explore the impact of various aspects, like the increasing number of dimensionality or a growing number of observed neighbours.

Overall, our experiments provide evidence that PCA* is a valuable tool for dimensionality reduction, particularly when order-based distances and relationships need to be preserved in a dataset where the order shows high variability. However, the choice of method should be driven by the specific goals and characteristics of the data at hand. PCA* offers a valuable alternative when order-based information is crucial.

In future work, it is essential to further explore the generalisability of PCA*. For instance, the extension of PCA* to multiple sequential orders or the application on a partially ordered dataset is intriguing for future investigations.

Appendix A

Technical Details on Experiments

Computer Generation	CPU	Kernel/ Threads	RAM
Flüsse	i5-4590 @ 3.3GHz	4/4	32GB
Edelsteine	i7-8700 @ 3.2GHz	6/12	64GB
Gesteine A	i9-9900 CPU @ 3.10GHz	8/16	64GB
Gesteine B	i9-9900 CPU @ 3.10GHz	8/16	64GB

Table A.1: Generally available computing resources at the IFI [35]

Bibliography

- [1] Rishi Advani, Paolo Papotti, and Abolfazl Asudeh. Maximizing neutrality in news ordering. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, page 11–24, New York, NY, USA, 2023. Association for Computing Machinery.
- [2] Nikos Alexandris, Sandeep Gupta, and Nikos Koutsias. Remote sensing of burned areas via pca, part 1; centering, scaling and evd vs svd. *Open Geospatial Data, Software and Standards*, 2017.
- [3] Akbar Ali, Kinkar Das, Darko Dimitrov, and Boris Furtula. Atom-bond connectivity index of graphs: a review over extremal results and bounds. *Discrete Mathematics Letters*, 5, 03 2021.
- [4] Mehala Balamurali. t-distributed stochastic neighbor embedding. In B.S. Daya Sagar, Qiuming Cheng, Jennifer McKinley, and Frits Agterberg, editors, *Encyclopedia of Mathematical Geosciences*, pages 1–9, Cham, 2020. Springer International Publishing.
- [5] Igor I. Baskin, Vladimir A. Palyulin, and Nikolai S. Zefirov. A neural device for searching direct correlations between structures and properties of chemical compounds. *J. Chem. Inf. Comput. Sci.*, 37:715–721, 1997.
- [6] Anna Beer, Olivér Palotás, Andrea Maldonado, Andrew Draganov, and Ira Assent. Dropp: Structure-aware pca for ordered data - a general method and its applications in climate research and molecular dynamics. UNPUBLISHED, 2023.
- [7] Donald J. Berndt and James Clifford. Using dynamic time warping to find patterns in time series. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, AAAIWS'94, page 359–370. AAAI Press, 1994.
- [8] Kanishka Bhaduri. Multivariate time series search, 2011. Accessed August 4, 2023. <https://data.nasa.gov/dataset/Multivariate-Time-Series-Search/mrcc-b53g>.

- [9] Ella Bingham and Heikki Mannila. Random projection in dimensionality reduction: Applications to image and text data. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '01, page 245–250, New York, NY, USA, 2001. Association for Computing Machinery.
- [10] Rasmus Bro, Evrim Acar, and Tamara Kolda. Resolving the sign ambiguity in the singular value decomposition. *Journal of Chemometrics*, 22:135 – 140, 02 2008.
- [11] Jorge Cadima and Ian Jolliffe. On relationships between uncentred and column-centred principal component analysis. *Pakistan Joudnal of Statistics*, 25(4), 2009.
- [12] Yanshuai Cao and Luyu Wang. Automatic selection of t-sne perplexity, 2017.
- [13] D.J. Cook and L.B. Holder. *Mining Graph Data*. Wiley, 2006.
- [14] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.
- [15] John P. Cunningham and Zoubin Ghahramani. Linear dimensionality reduction: Survey, insights, and generalizations, 2016.
- [16] Subramaniam Dhanabal and Chandramathi SA. A review of various k-nearest neighbor query processing techniques. *Int. J. Comput. Appl.*, 3, 01 2011.
- [17] Diab M. Diab, Basil AsSadhan, Hamad Binsalleeh, Sangarapillai Lambotharan, Konstantinos G. Kyriakopoulos, and Ibrahim Ghafir. Anomaly detection using dynamic time warping. In *2019 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, pages 193–198, 2019.
- [18] M.M. Fischer and J. Wang. *Spatial Data Analysis: Models, Methods and Techniques*. SpringerBriefs in Regional Science. Springer Berlin Heidelberg, 2011.
- [19] Benyamin Ghojogh, Ali Ghodsi, Fakhri Karray, and Mark Crowley. Locally linear embedding and its variants: Tutorial and survey, 2020.

- [20] Rassoul Hajizadeh, A. Aghagolzadeh, and M. Ezoji. Local distances preserving based manifold learning. *Expert Systems with Applications*, 139:112860, 2020.
- [21] Felix Hamborg, Karsten Donnay, and Bela Gipp. Automated identification of media bias in news articles: an interdisciplinary literature review. *International Journal on Digital Libraries*, 20, 12 2019.
- [22] Fang Han and Han Liu. Principal component analysis on non-gaussian dependent data. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML’13, page I–240–I–248. JMLR.org, 2013.
- [23] Rob J. Hyndman and Anne B. Koehler. Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4):679–688, 2006.
- [24] Shoaib Jameel and Steven Schockaert. Modeling context words as regions: An ordinal regression approach to word embedding. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 123–133, Vancouver, Canada, August 2017. Association for Computational Linguistics.
- [25] Young-Seon Jeong, Myong K. Jeong, and Olufemi A. Omitaomu. Weighted dynamic time warping for time series classification. *Pattern Recognition*, 44(9):2231–2240, 2011. Computer Analysis of Images and Patterns.
- [26] I.T. Jolliffe. *Principal Component Analysis*. Springer Series in Statistics. Springer, 2002.
- [27] Hyeong Ryeol Kam, Sung-Ho Lee, Taejung Park, and Chang-Hun Kim. Rviz: a toolkit for real domain data visualization. *Telecommunication Systems*, 60(1572-9451), 2015.
- [28] Nandakishore Kambhatla and Todd K. Leen. Dimension reduction by local principal component analysis. *Neural Computation*, 9(7):1493–1516, 1997.
- [29] Eamonn J. Keogh and Michael J. Pazzani. Scaling up dynamic time warping for datamining applications. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’00, page 285–289, New York, NY, USA, 2000. Association for Computing Machinery.

- [30] Stanislav Kolenikov and Gustavo Angeles. Socioeconomic status measurement with discrete proxy variables: Is principal component analysis a reliable answer? *Review of Income and Wealth*, 55(1):128–165, 2009.
- [31] Yang Liu, Feiping Nie, Quanxue Gao, Xinbo Gao, Jungong Han, and Ling Shao. Flexible unsupervised feature extraction for image classification. *Neural Networks*, 115:65–71, 2019.
- [32] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Grossberger. Umap: Uniform manifold approximation and projection. *The Journal of Open Source Software*, 3(29):861, 2018.
- [33] Sergio Munoz-Romero, Vanessa Gomez-Verdejo, and Jeronimo Arenas-Garcia. Regularized multivariate analysis framework for interpretable high-dimensional variable selection. *IEEE Computational Intelligence Magazine*, 11(4):24–35, 2016.
- [34] Yusuke Naritomi and Sotaro Fuchigami. Slow dynamics in protein fluctuations revealed by time-structure based independent component analysis: The case of domain motions. *The Journal of Chemical Physics*, 134(6):065101, 02 2011.
- [35] RBG / IT of Institut für Informatik der LMU. General available compute resources at ifi. online, n.d. Accessed September 19, 2023. https://www.rz.ifi.lmu.de/infos/compute_en.html.
- [36] Emanuel Parzen. An Approach to Time Series Analysis. *The Annals of Mathematical Statistics*, 32(4):951 – 989, 1961.
- [37] Chi Seng Pun. Low- and high-dimensional stock price data, 2017.
- [38] Uday Kiran Rage. 5+ years of multiple time series data of hourly pm2.5 recordings gathered from various sensors located throughout japan (1-1-2018 to 25-4-2023), 2023. Accessed August 9, 2023. <https://data.mendeley.com/datasets/phgrnvkykmr/1>.
- [39] R. Remesan, M. Bray, and J. Mathew. Application of pca and clustering methods in input selection of hybrid runoff models. *Journal of Environmental Informatics*, 31(2), June 2018.
- [40] Stan Salvador and Philip Chan. Toward accurate dynamic time warping in linear time and space. *Intell. Data Anal.*, 11(5):561–580, oct 2007.
- [41] P.S.D. Sarker and L. Nahar. *Chemistry for Pharmacy Students: General, Organic and Natural Product Chemistry*. Wiley, 2013.

- [42] scikit-learn developers. scikit-learn, 2023. Accessed September 25, 2023. https://github.com/scikit-learn/scikit-learn/blob/55a65a2fa/sklearn/decomposition/_pca.py#L118.
- [43] scikit-learn developers. sklearn.manifold.tsne, 2023. Accessed September 23, 2023. <https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html>.
- [44] Jonathon Shlens. A tutorial on principal component analysis. *CoRR*, abs/1404.1100, 2014.
- [45] Vin Silva and Joshua Tenenbaum. Global versus local methods in nonlinear dimensionality reduction. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems*, volume 15. MIT Press, 2002.
- [46] Song, Yinglei, Li, Yongzhong, and Qu, Junfeng. Preserving global and local structures for supervised dimensionality reduction. *MATEC Web of Conferences*, 28:06003, 2015.
- [47] James V. Stone. *Independent Component Analysis: A Tutorial Introduction*. The MIT Press, 09 2004.
- [48] Bing Su, Xiaoqing Ding, Hao Wang, and Ying Wu. Discriminative dimensionality reduction for multi-dimensional sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(1):77–91, 2018.
- [49] Alaa Tharwat, Tarek Gaber, Abdelhameed Ibrahim, and Aboul Ella Hassanien. Linear discriminant analysis: A detailed tutorial. *AI Communications*, 30(2), 2017.
- [50] Michel Verleysen and Damien François. The curse of dimensionality in data mining and time series prediction. In Joan Cabestany, Alberto Prieto, and Francisco Sandoval, editors, *Computational Intelligence and Bioinspired Systems*, pages 758–770, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.
- [51] Jinping Wang, Ruisheng Ran, and Bin Fang. Global and local structure network for image classification. *IEEE Access*, 11:27963–27973, 2023.
- [52] 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. *J. Mach. Learn. Res.*, 22(1), jan 2021.

- [53] F.C. Wu and Z.Y. Hu. The lle and a linear mapping. *Pattern Recognition*, 39(9):1799–1804, 2006.
- [54] Guo Yu and Jacob Bien. Learning local dependence in ordered data. *Journal of Machine Learning Research*, 18(42):1–60, 2017.