VISUAL EXPLORATION OF HIGH-DIMENSIONAL DATA USING DIMENSIONALITY REDUCTION with Applications in Astronomy

YOUNGJOO KIM
Cover: Abstract illustration of the sharpening method proposed in this thesis. Trajectories of various objects shifting towards a dense area are shown. Images were created using DeepAI and edited by Youngjoo Kim.

Visual Exploration of High-dimensional Data Using Dimensionality Reduction with Applications in Astronomy

Youngjoo Kim
PhD Thesis

This thesis is the result of the Data Science and Systems Complexity (DSSC) research training program developed at the University of Groningen and is part of the Marie Sklodowska-Curie COFUND action. This work is co-funded by the DSSC program and the European Commission (DSSC 754315) and was conducted at the Scientific Visualization and Computer Graphics (SVCG) research group and Kapteyn Institute.
Visual exploration of high-dimensional data using dimensionality reduction

with Applications in Astronomy

PhD thesis

to obtain the degree of PhD at the
University of Groningen
on the authority of the
Rector Magnificus Prof. C. Wijmenga
and in accordance with
the decision by the College of Deans.

This thesis will be defended in public on
Tuesday 4 July 2023 at 09.00 hours

by

Youngjoo Kim

born on 21 February 1992
in Seoul, Republic of Korea (South Korea)
Supervisors
Prof. A. C. Telea
Prof. S. C. Trager
Prof. J. B. T. M. Roerdink

Assessment committee
Dr. M. H. F. Wilkinson
Prof. J. M. van der Hulst
Prof. C. Hurter
This Work is dedicated
By a humble native of Flatland
In the hope that
Even as he was initiated into the mysteries
Of three dimensions
Having been previously conversant
With only two
So the citizens of that celestial region
May aspire yet higher and higher
To the secrets of four five or even six dimensions
Thereby contributing
To the enlargement of the imagination
And the possible development
Of that most rare and excellent gift of modesty
Among the superior races
Of solid humanity.

— Edwin A. Abbott,
Flatland: A Romance of Many Dimensions
Intuitively analyzing multidimensional data for exploratory purposes is challenging. Multidimensional data visualization is used to tackle this challenge. In the field of multidimensional data visualization, dimensionality reduction (DR) provides a lower-embedding of the original high-dimensional data so that the data are more accessible by visualization. Furthermore, DR is the preferred method to find visual clusters of points that represent the clusters in the original data. However, finding visually well-separated clusters using conventional DR methods is challenging, as there are numerous DR methods applicable to a wide range of data sets. Therefore, this thesis focuses on using a preconditioning step of DR by sharpening the multidimensional data in the high-dimensional space prior to dimensionality reduction so that clusters are also separated better in the lower-dimensional embedding. We improve this sharpened DR method in terms of dimensional scalability, computational scalability, ease-of-use, and stability. The method is also applicable to any real-valued multidimensional data set allowing applications in various fields. We show applications in various labeled data sets, such as human motion, WiFi, and banknote data sets and qualitatively and quantitatively assess the results. Furthermore, we apply our proposed sharpened DR on an unlabeled astronomical data set of stellar abundances called GALactic Archaeology with HERMES (GALAH), which includes attributes of the chemical composition of stars that can be used to ultimately understand the evolutionary history of the Milky Way. We find that sharpened DR assists the manual labeling of clusters, which is otherwise challenging using existing visualization methods with this stellar abundance data set. We further analyze the resulting clusters from sharpened DR to conclude that the clusters are associated with different sub-components of the Milky Way. In summary, we argue that this work is an important step towards finding interesting and meaningful visual clusters for the exploratory analysis of high-dimensional data in various fields.
Het intuïtief analyseren van multidimensionale gegevens voor exploratieve doeleinden is een uitdaging. Multidimensionale datavisualisatie wordt gebruikt om deze uitdaging aan te gaan. Op het gebied van multidimensionale datavisualisatie zorgt dimensionaliteitsreductie (DR) voor een lagere inbedding van de oorspronkelijke hoogdimensionale data, zodat de data toegankelijker worden door middel van visualisatie. Bovendien is de DR de methode bij uitstek geschikt om visuele clusters van punten te vinden die de clusters in de oorspronkelijke gegevens weergeven. Het vinden van visueel goed gescheiden clusters met behulp van conventionele DR-methoden is echter een uitdaging, aangezien er talrijke DR-methoden zijn die van toepassing zijn op een breed scala van datasets. Daarom richt dit proefschrift zich op het gebruik van een preconditioneringsstap van DR door het verscherpen van de multidimensionale gegevens in de hoogdimensionale ruimte voorafgaand aan de dimensionaliteitsreductie, zodat clusters ook beter worden gescheiden in de lagerdimensionale inbedding. We verbeteren deze verscherpte DR-methode in termen van dimensionale schaalbaarheid, computationele schaalbaarheid, gebruiksgemak en stabiliteit. De methode is ook toepasbaar op elke reële multidimensionale dataset, waardoor toepassingen op verschillende gebieden mogelijk zijn. We tonen toepassingen aan de hand van verschillende gelabelde datasets, zoals menselijke bewegingen, WiFi en datasets van bankbiljetten, en beoordelen de resultaten kwalitatief en kwantitatief. Verder passen we de door ons voorgestelde verscherpte DR toe op een nietgelabelde astronomische dataset van stellaire abundanties, genaamd GA-Lactic Archaeology with HERMES (GALAH), die attributen bevat van de chemische samenstelling van sterren die kunnen worden gebruikt om uiteindelijk de evolutionaire geschiedenis van de Melkweg te begrijpen. We concluderen dat verscherpte DR helpt bij het handmatig labelen van clusters, wat een uitdaging zou zijn met bestaande visualisatiemethoden voor deze dataset van stellaire abundanties. Verder analyseren we de clusters die uit de verscherpte DR naar voren komen en concluderen dat de clusters geassocieerd zijn met verschillende subcomponenten van de Melkweg. Samenvattend kunnen we stellen dat dit werk een belangrijke stap is naar het vinden van interessante en betekenisvolle visuele clusters voor de exploratieve analyse van hoogdimensionale gegevens op verschillende gebieden.
This thesis is the result of the following publications:


- **Chapter 3:** J. Heo*, Y. Kim*, and J. B. T. M. Roerdink. Human activity data using sharpened dimensionality reduction and clustering. International Conference on Biomedical and Health Informatics (ICBHI2021), 2021. [89]

- **Chapter 4:**

The work of the following chapter is in preparation to be submitted:

- **Chapter 5:** Y. Kim, S. C. Trager, A. C. Telea, and J. B. T. M. Roerdink. Analysis of Stellar Abundance Space of GALAH+ DR3 Using Sharpened Dimensionality Reduction.

* Equal contribution.
CONTENTS

1 INTRODUCTION 1
  1.1 Multidimensional data visualization for exploratory analysis 2
    1.1.1 Table lenses and small multiples 3
    1.1.2 Parallel Coordinate Plots (PCPs) 5
    1.1.3 Scatterplots 6
    1.1.4 Dimensionality Reduction 7
    1.1.5 Exploring Projections 9
    1.1.6 Visual cluster separation and projection methods 11
      1.1.6.1 Distance preservation methods 11
      1.1.6.2 Neighborhood preservation methods 13
      1.1.6.3 Labeling 15
  1.2 Outline of the Sharpened Dimensionality Reduction Method 16
  1.3 Applications in astronomy 18
    1.3.1 Clustering in astronomy 19
    1.3.2 Multidimensional data visualization in astronomy 20
    1.3.3 DR in astronomy 21
    1.3.4 Application goal 22
  1.4 This thesis 22

2 VISUAL CLUSTER SEPARATION BY HIGH-DIMENSIONAL SDR 27
  2.1 Introduction 27
  2.2 Related Work 29
    2.2.1 Dimensionality Reduction and Cluster Separation 30
    2.2.2 Dimensionality reduction for labeling 31
    2.2.3 DR in astronomy 32
  2.3 Proposed Method 33
    2.3.1 Local Gradient Clustering 33
    2.3.2 Dimensionality Reduction Candidates for HD-SDR 37
2.4 Results 39
  2.4.1 Synthetic data: qualitative evaluation 39
  2.4.2 Real-world data: qualitative evaluation 42
    2.4.2.1 Data sets and their traits 42
    2.4.2.2 HD-SDR applied to real-world data sets 43
  2.4.3 Quantitative Evaluation 44
    2.4.3.1 Evaluation of LGC 46
    2.4.3.2 Evaluation of HD-SDR 46
  2.5 Application to astronomical data 50
  2.6 Discussion 57
    2.6.1 Scalability 57
      2.6.1.1 Speed 57
      2.6.1.2 High-dimensional data 59
    2.6.2 Data distortion 60
    2.6.3 Relation to clustering 61
    2.6.4 Preservation of outliers 61
    2.6.5 Limitations and future work 63
      2.6.5.1 Undersegmentation and oversegmentation 63
      2.6.5.2 Noise-free data 64
      2.6.5.3 Parameter setting 64
      2.6.5.4 Post-processing in dimensionality reduction 65
      2.6.5.5 LGC for t-SNE and UMAP 66
      2.6.5.6 Selection of baseline DR method 66
  2.7 Conclusion 66

3 Human motion detection using SDR and clustering 79
  3.1 Introduction 79
  3.2 Method 80
    3.2.1 Clustering methods 80
    3.2.2 Evaluation metrics 81
  3.3 Results 82
    3.3.1 Synthetic data 82
    3.3.2 Real-world data 84
    3.3.3 Qualitative evaluation 85
    3.3.4 Quantitative evaluation 86
  3.4 Discussion 86
  3.5 Conclusion 87
The recent advancement of technology has made it possible to store large volumes of data from disparate sources and the data accumulate even as we speak. Different types and forms of data (variety) are produced with high speed (velocity) and in large volumes (volume) and the quality of data may also vary resulting in uncertainty of the data (veracity). Data with these 4V challenges – variety, velocity, volume, and veracity – require advanced technology for storage, maintenance, and analysis and are often known under the generic name of big data. The big data paradigm is present in various fields, such as astronomy, genetics, healthcare, socioeconomics, and education. While each of these fields produces different types of data, some of the questions and challenges related to the analysis, processing, and interpretation of the data are common to all fields.

One of the central challenges are related to how one deals with the size of big data. A collection of data elements – typically called a data set – can be often seen as being structured as a table. Each row in this table represents a separate measurement of some type of phenomenon and corresponds to so-called samples, observations, or data points. Table columns correspond to so-called dimensions, attributes, or variables, which represent the different aspects of the phenomenon that are measured by its samples. Given a tabular structured data set, the size of a data set can be seen as the product of the number of samples $N$ and number of dimensions $n$. As either $N$ or $n$ increases, the data size increases making the analysis, processing, and interpretation of the data set more difficult.

At a high level data scientists study such data sets for a variety of reasons related to answering questions concerning the underlying phenomena which are captured or sampled by the respective data set. This desire to find answers from the vast amount of information from big data is just like archery. While the questions that are asked by domain experts or data scientists are the arrows to shoot, the technology enabling data science is the bow, and the data at hand are the (very) large target that lies in front of us. Following this analogy, the difficulty of answering such questions is a combination of both the size of the data set and the nature of the question. For instance, finding the maximum value of a numerical data set is
arguably a simple challenge even for a very large data set, as it involves a straightforward process of visiting all data values. Conversely, finding how certain dimensions (or ranges thereof) can predict the values of other dimensions is a very complex challenge even for a small data set, as the question itself is of complex nature.

Such challenges in analyzing big data relate to the relative contributions of the number of samples \( N \) and dimensions \( n \). In general, data sets having many dimensions are significantly harder to understand and to analyze when answering a given question (regardless of the number of samples) than data sets having many samples but fewer dimensions. The former data sets are known under the name of multidimensional data, or high-dimensional data as the dimension count increases. In contrast, data sets with just a few dimensions are called low-dimensional data.

There are several reasons why the dimension count \( n \) affects the difficulty of analysis more than the sample count \( N \). As previously explained, samples are (typically) measurements of the same phenomenon under comparable conditions and one can reduce the sample count \( N \) relatively easily by using, for example, aggregation techniques such as averaging or computing distributions of values over each dimension. In contrast, dimensions measure different aspects of a phenomenon, each of which can influence the phenomenon in various ways – hence, aggregating dimensions is significantly harder than aggregating samples. Moreover, explaining the meaning of a set of aggregated samples is relatively easier than explaining the meaning of a set of aggregated dimensions, as aggregation can occur independently per dimension. To target such challenges in high-dimensional data, visualization is one of the fundamental tools used by data scientists.

1.1 Multidimensional Data Visualization for Exploratory Analysis

In this section, the general concept of visualization is first introduced prior to explaining visualization targeted for multidimensional data. ‘Visualization’ aims to map a given data set (or parts thereof) to visual elements, most often resulting in 2D or 3D imagery. This mapping is then examined and ‘inverted’ to decode specific aspects of the data at hand by the user of the visualization [194]. One can further divide the field of visualization into two main branches [150, 194]: scientific and information visualization. Scientific visualization relates to the exploration of data sets with samples being measurements of a (physical) phenomenon that unfolds into Euclidean 2D or 3D space (or 4D space in the case of time-dependent data) –
meaning that the samples correspond to locations in such a space. Information visualization deals with the exploration of data sets with samples that come from so-called abstract data spaces. Hence, data sets in scientific visualization typically contain mainly quantitative attributes which represent the sampling of some continuous phenomenon. In contrast, data sets in information visualization contain a wider range of attribute types (e.g., quantitative, integral, ordinal, nominal, relational, text, and images). These attributes may or may not come from phenomena being governed by underlying continuity principles. A separate aspect common to both scientific and information visualization concerns the interaction with the visualization. By an iterative process of changing which parts of the data set are visualized and how the mapping occurs, users can answer complex questions on the data at hand more accurately. The process of interactive-and-iterative exploration of data by means of visualization to form, refine, and ultimately confirm (or disprove) hypotheses about the phenomena that the data represent is called visual analytics [147, 197, 228].

When the data does not have a spatial nature, i.e., is not representing measurements that are sampled over a compact subset of the 2D or 3D continuous, physical, space, multidimensional data visualization methods typically fall into the realm of information visualization research. Multidimensional data visualization focuses on visually representing data that has more dimensions \( n \) than we can straightforwardly map using classical encoding methods present in scientific visualization [35, 194]. In detail, such classical encoding methods map every data dimension to one of the so-called visual channels or visual variables such as 2D or 3D coordinates or color. There are only a limited number of visual variables – roughly 3 to 6 – that can be used simultaneously if we want to allow users to decode each of them independently [206, 207]. Hence, visualizing data sets that have more dimensions requires other methods than the aforementioned straightforward mapping. A compact overview of the best known multidimensional data visualization methods including their relative advantages and limitations is given below (see also Fig. 1.1 for a taxonomy of these methods). This section focuses particularly on the earlier identified challenge of visualizing high-dimensional data sets, which is defined next as data sets having tens to hundreds of dimensions \( n \).

1.1.1 Table lenses and small multiples

Arguably one of the simplest visual encodings to use to depict a table is to draw the table itself, as done by many spreadsheet applications. Besides using a layout which is immediately familiar to a wide range of users, this
Figure 1.1: A taxonomy of widely used multidimensional data visualization techniques. This thesis focuses on the highlighted branch of dimensionality reduction.

Visualization also allows users to inspect every value (of every row across every column) in detail. However, this option does not scale well for many samples $N$. To alleviate this, table lenses [167] use the same Cartesian layout but reduce every cell to a single horizontal pixel-line that encodes the cell’s value via the line’s length and/or color (see Fig. 1.2(a)). This allows scaling the visualization, in the first instance, to an $N$-value equal to the number of vertical pixel lines available on a screen. Enhanced methods use further sampling and aggregation to allow displaying tables with $N$ exceeding 100K samples [193]. Grouping similar-valued samples further allows creating even more aggregated visualizations from such tables to the point where each screen pixel represents a data sample [213].

A Small Multiples visualization is a related visualization technique which organizes the data in a tabular fashion [140, 194]. The created visualization resembles a table in which rows and columns indicate the grouping of data samples in terms of the value ranges and/or dimensions that every ‘cell’ in the table depicts. Next, each cell uses the same type of visualization (e.g., barchart, timeline, scatterplot, or any other user-chosen technique) to
1.1 Multidimensional Data Visualization for Exploratory Analysis

Figure 1.2: Examples of different visualization methods. (a) Table lens. (b) PCP. (c) Scatterplot. (d) SPM.

depict the subset of data assigned to it. Small multiple visualizations are at the core of many dashboard and data analytic tools, such as Tableau [27]. While table visualizations are scalable in the sample count $N$ as previously described, they are limited in the number of dimensions $n$ that they can display typically up to 20, as each dimension requires an explicit display space to be mapped to.

1.1.2 Parallel Coordinate Plots (PCPs)

Parallel coordinate plots (PCPs) [180, 214, 226] also use a Cartesian layout (Fig. 1.2(b)). All $n$ dimensions are displayed as vertical axes stacked
horizontally with some space in between. Each of the $N$ samples is then mapped to a polyline with one control point per dimension. The $x$- and $y$-coordinates of these points are given by the position of the dimension along the set of stacked axes, respectively the value of the sample along that dimension. PCPs allow one to see data patterns such as correlations between dimensions or dense or sparse regions or outliers on each axis. However, PCPs require axis ordering for studying the correlations of specific dimensions. They also cannot clearly show patterns that involve more than two or three dimensions. Extended versions of PCPs require extensive user interaction to generate the desired optimal views. Finally, while they scale as well as table lenses with the sample count $N$, they have the same limitation with respect to the dimension count $n$ given by the fact that each dimension requires separate display space.

1.1.3 Scatterplots

Scatterplots are arguably one of the oldest and still most used visualization methods for multidimensional data [28, 208]. The simplest form of scatterplots display the values of the $N$ samples of a data set along two selected dimensions from a total of $n$ dimensions – every sample being plotted as a point (see Fig. 1.2(c)). A few additional data dimensions can be added to a scatterplot visualization by using a 3D spatial representation and/or extra visual variables such as hue, brightness, shape, or size of the plotted points. Scatterplots allow one to easily see the distribution of samples over the value ranges of the depicted dimensions. Consequently, more refined observations can be made such as the detection of (direct or inverse) correlations, clustering of samples, or presence of outlier data points.

Scatterplot matrices (SPMs) generalize the idea of scatterplots to display a higher number of dimensions than those allowed by a single scatterplot [12]. SPMs use a small multiples design – for a data set with $n$ dimensions, a matrix of $n \times n$ scatterplots is shown, where scatterplot $(i, j)$ shows the data dimensions $i$ and $j$. Figure 1.2(d) illustrates this for a data set of $N = 50$ observations each having $n = 6$ dimensions (A–F). The columns and rows each indicate the attributes A–F they visualize. Extended versions of SPMs may contain extra information to be laid out in either of the off-diagonal scatterplots in the matrix – note that this matrix is symmetric. Other extensions include additional information on each dimension on the diagonal entries, such as the histograms shown in Fig. 1.2(d). Like PCPs and basic scatterplots, SPMs also scale well with the number of samples $N$. However, they require screen space which is quadratic in the
number of dimensions \( n \) – note that, in contrast, table lenses and PCPs require space that is linear in this count. Moreover, when \( n \) increases to roughly over a dozen dimensions, SPMs become too complex for one to gain insight from the data, as one has to go through many scatterplots while remembering and correlating the seen patterns [140, 180]. Density plots are another option to reduce clutter and show more insight in the density of all regions of a scatterplot [140].

1.1.4 Dimensionality Reduction

The multidimensional visualization methods described above – table lenses, SPMs, and PCPs – scale relatively well in the sample count \( N \) but poorly in the dimension count \( n \). More specific limitations exist too: SPMs strongly rely on the visual memory of the user [12, 206], while PCPs, table lenses, and small multiples are also limited by the width of the screen. All these methods (except plain scatterplots) require careful ordering of the dimensions [41, 214, 226]. In summary, these methods are not applicable to high-dimensional data exceeding tens of dimensions.

One common aspect of all these methods which makes them scale poorly with the dimension count is that they all allocate explicit screen space to depict each of the dimensions they aim to visualize. While this is arguably desirable when one needs to study the specific values of samples in all these dimensions, there are cases where a trade-off between how easily we can ‘read’ the dimension values and how many dimensions we can handle is desirable. In this sense, we can roughly classify multidimensional visualization methods into two types [41, 140]:

- **observation-centric**: These methods focus on depicting relations between the data points, e.g., clusters of similar points or outlier points. They focus less on depicting dimension values;

- **attribute-centric**: These methods support users in reasoning about explicit values of all the depicted dimensions. They focus less on comparing data samples; Table lenses and small multiples, SPMs, and PCPs discussed so far are more attribute-centric than observation-centric.

SPMs and small multiples are not categorized as observation-centric methods, as these methods do not plot the observations in a single visualization, but rather in several smaller visualizations, which make it difficult to reason about clusters as a whole.
Dimensionality Reduction (DR) methods, also known as projections or low-dimensional embeddings\(^1\) are more observation-centric than attribute-centric. The name ‘projection’ can be most intuitively understood by following the commonly known meaning of the word in geometry. Imagine how a 2D living polygon confined to a 2D planar world understands the concept of a 3D sphere while the sphere passes through this 2D plane as depicted in Edwin A. Abbott’s book, ‘Flatland’ [1]. When a 3D sphere passes through Flatland, the 2D occupant of Flatland will sense the passing sphere only in terms of a 1D curved line segment whose length changes from all possible viewing directions as illustrated in Fig. 1.3. Luckily, we humans can see the full view of a 2D circle resulting from the plane-sphere intersection, which is the ‘projection’ of the sphere onto the plane (more precisely, the projection of the sphere’s surface). Likewise, one is able to gain intuition on the high-dimensional structure being projected using DR.

More formally, a DR method can be modeled by a function \(P : \mathbb{R}^n \rightarrow \mathbb{R}^s\), with \(s \ll n\). For visualization purposes, one uses \(s \in \{2, 3\}\), which leads to 2D and 3D projections, respectively (see Fig. 1.2(c)). We focus on \(s = 2\) for the remainder of this thesis\(^2\). Given a data set \(D \{x_i\} \subset \mathbb{R}^n\), let \(P(D) = \{P(x_i)\}|x_i \in D, i = 1, 2, \ldots, N\}, P(D) \subset \mathbb{R}^s\) be its projection by the function \(f\). The resulting projection \(P(D)\) is most typically visualized as an \(s\)-dimensional scatterplot with one point depicting each sample in \(P(D)\). Projection functions can be seen as optimizers which aim

---

1 The terms DR, projections, and low-dimensional embedding are used interchangeably for the remainder of this thesis.

2 Although 3D projections \((s = 3)\) can capture more data structures compared with 2D projections \((s = 2)\), 3D projections are known to have little added-value over 2D projections based on an extensive quantitative and qualitative comparison of 2D versus 3D projections from Tian et al. [198]. Hence, this thesis focuses on applying and analyzing 2D projections.
to keep the similarity of points in $P(D)$ close to that of their counterparts in $D$. Given a similarity function $d_n : \mathbb{R}^n \rightarrow \mathbb{R}^+$ for $D$ and another similarity function $d_s : \mathbb{R}^s \rightarrow \mathbb{R}^+$, the projection function $P$ aims to achieve $d_s(P(x_i), P(x_j)) \approx d_n(x_i, x_j)$ for as many – ideally all – point pairs $(x_i, x_j) \in D \times D$. The sign $\approx$ denotes that the two distances should be as close to each other as possible. Obtaining a perfect so-called ‘distance preservation’ between $D$ and $P(D)$ is rarely possible with real-world data.

Dozens of different projection methods have been created in the last decades (for recent surveys refer to [62, 153]). These methods can be generally categorized into two subcategories (see also Fig. 1.1):

- **Variable selection methods**: These methods aim to select $s$ dimensions from the original $n$ dimensions to create $P$. Dimension selection can be done manually – as in scatterplots – or by using various ranking mechanisms which tell which dimensions are the most relevant or interesting for a given exploration scenario. For example, Ferdosi et al. finds subspaces showing clustering using max-trees of density estimates [68, 69]. Dimension selection is related closely to the concept of feature selection in machine learning [2, 80]. While manual dimension selection imposes a significant burden on the user, it is arguable that no subset of $s$ dimensions is ideal for depicting a given $n$-dimensional real-world data set. Hence, variable selection methods are not explored further in this thesis.

- **Variable synthesis methods**: These methods generate the $s$ dimensions of $P(D)$ by analyzing the $n$ dimensions of the input data set to optimize the similarity preservation between the aforementioned $D$ and $P(D)$. Such methods are typically fully automatic, i.e., do not require users to (pre)select which of the $n$ dimensions they want to use (or not) for projection. More importantly, these methods can preserve more structure of the high-dimensional data set in the projection by creating combinations (syntheses) of the original $n$ dimensions rather than just selecting a subset thereof. Hence, the remainder of this thesis focuses on variable synthesis methods.

### 1.1.5 Exploring Projections

With DR serving as the bow in the earlier archery analogy, this section focuses on the role of arrows: finding answers to interesting questions that address high-dimensional data. DR has multiple goals, such as data compression [10], feature extraction [20], and exploratory analysis [153]. The focus of this thesis is on the latter use-case, exploratory analysis.
At a high level, exploratory data analysis can take two extreme forms: first, one has precise, targeted questions to be answered on a data set $D$ using its visualization $P(D)$ (e.g., finding out whether a given labeled data set can be easily classified by a given machine learning technique [169] or how many distinct groups of samples exist in a data set [153]); and second, one favors open exploration (i.e., aims to study the projection to find interesting, previously unknown and potentially useful, insights on a data set or phenomenon, such as how a given deep learning model performs [168]). Common to both above scenarios is the need for the user to interpret the projection – more precisely, to perform the ‘inverse mapping’ from structures visible in the projection $P(D)$ to aspects present in the data $D$, as mentioned earlier. In the remainder of this thesis, the use of projections for both types of scenarios are addressed.

Figure 1.4: Example showing the shadow (2D projection) of a 3D penguin.

The inverse mapping process introduced above can be illustrated by a (necessarily) simple example. Imagine looking at a shadow $P(D)$ of an animal $D$ on the wall not knowing which animal it is. The shadow looks like an oval-like figure with two sharp elongated lumps to the side and two lumps of what looks like wide feet as in Fig. 1.4. The $s = 2$-dimensional projection (shadow) suggests that the animal is in fact a penguin. Likewise, when taking a glimpse of a higher-dimensional data set, one can take a glimpse of certain structural properties of the data set by exploring its projection, e.g., the size and number of clusters, neighboring relations between clusters, intermixing of clusters having different data properties, and outlier points. Such properties are collectively known as the data structure. Likewise, projections are also characterized as operations which aim to preserve the data structure [105, 153].

As previously explained, projections aim to put similar samples (from $D$) close to each other (in $P(D)$). The visual structures created by projections will be an effect of the similarity of the data points and how well the projection function $P$ preserves this similarity. For instance, imagine a data
set $D$ where the data is distributed in two well-separated clusters of points. A good-quality projection technique (in terms of similarity preservation) will create a projection $P(D)$ in which we see two well-separated point groups. From this projection, one can infer the cluster structure of the original data set $D$. Likewise, the ability of a projection to create such well-separated structures of points is key to its quality and suitability for visualization purposes. In other words, a good projection technique $P$ is one where the data cluster separation present in $D$ is reflected by the visual cluster separation in the projection $P(D)$ [168, 169].

The (visual) cluster separation can be defined more formally, as this concept is crucial to characterizing a projection technique. Let $H : S \rightarrow X$ be any metric or tool that captures the clusters present in a data set $S$. Let $H(S)$ denote the application of $H$ to all points in $S$. When $S = D$, $H$ captures the data cluster separation. Likewise, $H$ captures visual cluster separation (VCS) when $S = P(D)$. Examples of $H(D)$ include classifiers that label data points so that points in the same cluster are assigned the same label $x \in X$; and clustering techniques that assign a label (cluster ID) to similar data points ($X \subseteq \mathbb{N}$) or count the number of clusters in a data set (in this case, $X = \mathbb{N}$). Several instances of $H(P(D))$ have been proposed to measure VCS in visualization research [62]. In this context, a projection $P$ is said to have ‘good VCS’ when $H(P(D))$ and $H(D)$ are alike, i.e., $P$ should ideally represent in the 2D visual space the same cluster structures that the metric $H$ captures in the original data space.

1.1.6 Visual cluster separation and projection methods

A variety of projection methods achieve different amounts of visual cluster separation and preserve different properties of the input data. From this perspective, projection methods can be grouped into distance preservation and neighborhood preservation methods [153]. Both types of methods and their relation to visual cluster separation are introduced next.

1.1.6.1 Distance preservation methods

Our general definition of projection methods states that such methods aim to preserve a similarity of points between the high-dimensional and low-dimensional (projection) spaces (see Sec. 1.1.4). Distance preservation methods achieve this preservation by measuring the similarity using distance metrics $d_n$ and $d_2$ defined in the two spaces [192]. In most cases, $d_n$ and $d_2$ revert to the classical Euclidean distance, although variants such as the cosine distance can also be used. Given the above, computing $P$
amounts to minimizing a cost (or stress) function of these two distances, e.g., $\sum_{i \neq j} [d_n(x_i, x_j) - d_2(P(x_i), P(x_j))]^2$ [39]. We now introduce a few well-known distance preservation methods.

Principal Component Analysis (PCA), a widely used distance preservation method, is known to preserve ‘large’ pair-wise distances by maximizing the variance of the data [132]. PCA first computes the eigenvectors of the covariance matrix of the data $D$ – in other words, PCA finds the principal components that are linear combinations of the original variables – and projects the data on the top $s$ principal components (which have largest variance) to create the lower-dimensional embedding $P(D))$ [98, 227]. PCA is easy to use and can be applied to any real-valued data set, and is relatively stable for newly added data – a desirable property also known as out-of-sample (OOS) support. Independent Component Analysis (ICA) is similar to PCA but is not not widely used for DR purposes and instead aims to find independent sources of the data (blind source separation), such as solving the well-known cocktail party problem [100]. Another method similar to PCA is Multidimensional Scaling (MDS) [203]. MDS uses the distance matrix of all pair-wise point distances in the data and aims to minimize the aforementioned stress function. Other linear methods include Locality Preserving Projection (LPP) [83], Neighborhood Preserving Embedding (NPE) [84], Linear Discriminant Analysis (LDA) [37], Neighborhood Components Analysis (NCA) [78], and Maximally Collapsing Metric Learning (MCML) [77]$^3$. Methods such as Factor Analysis also have a linear nature and thus generally produce poor quality in terms of visual cluster separation.

While ‘linear’ methods like PCA and MDS are simple to implement and fast to compute, they do not account well for the non-linearity of the real-world data and their high dimensionality [8] and thus are not suitable for high-quality visual cluster separation. Intuitively, consider a 3D data set consisting of four point clusters located at the vertices of an equilateral tetrahedron. Arguably, a projection with good visual cluster separation will show four point clusters which are (a) well separated from each other and (b) at more or less equal distances from each other. However, creating a 2D linear projection which respects (a) and (b) perfectly is not possible – hence, (a) is favored. A linear projection will always map the same distance $d_n$ from the data space to the same value $d_2$ in the projection space – thus will not be able to fully satisfy (a) or (b).

---

$^3$ Note that the last three methods – LDA, NCA, and MCML – are supervised learning methods, which require labeled data sets.
Landmark MDS (LMDS) is a variant of MDS, where the computational scalability (linear in the sample count) is achieved by first embedding a small subset $L$ called ‘landmarks’ from the original data $D$ using classical MDS, around which the remaining samples $D \setminus L$ are then placed using a fast triangulation procedure [181]. LMDS is also a distance-preservation method, but aims to preserve distances locally rather than the global approach of PCA or MDS. Local Affine Multidimensional Projection (LAMP) and Least Square Projection (LSP) also use such landmarks (or representatives) [104, 157]. These landmark-based methods compute the landmarks using an expensive (but accurate) projection method and then project the remaining points using a fast (but less accurate) method. The computational scalability is not only affected by the number of observations $N$ and dimensions $d$ but also the number of selected landmarks $|L|$. Selecting a large fraction of the data as landmarks would typically result in a better projection quality than selecting a few, but a trade-off exists between the quality and computing time.

1.1.6.2 Neighborhood preservation methods

In contrast to distance preservation methods, neighborhood preservation methods do not directly use distances between the points during their computation of $P$, but the neighborhood relations thereof. These methods aim to keep points which are neighbors of each other in $D$ as neighbors of each other also in $P(D)$. More formally, if $N_k(x_i)$ is a set of $k$-nearest neighbors of a given point $x_i \in D$, then neighborhood preservation methods aim to acquire the same set of $k$-nearest neighbors for the projected point, $P(x_i)$. Preserving neighborhoods typically achieves a better cluster separation in the 2D projection space, which is the key added value of many unsupervised learning applications [94, 133]. The intuition behind the aforementioned cluster separation quality relates to the following. While distance preservation methods need to consider the actual distances between points in their optimization process, neighborhood preservation methods have more flexibility – as long as points stay as neighbors of each other in 2D as they are in nD, the actual distances between the former are free to be optimized for.

Visual cluster separation is, arguably, a property aimed for (and used to determine the quality of) neighborhood preservation projections even more so than for distance preservation methods. Indeed, for the latter, one could determine the projection quality by computing how well the nD distances are approximated by 2D distances, i.e., by evaluating the stress function. The same cannot be done for neighborhood preservation methods as these do not aim to explicitly preserve distances. Instead, the key
quality-determining factor relates to how well groups of points close in $D$ stay close in $P(D)$.

Two of the most well-known methods in this family of neighborhood preservation techniques are $t$-SNE and UMAP. $t$-SNE has gained its popularity due to its strong visual cluster separation ability compared with other DR methods [133]. $t$-SNE aims to minimize a cost function (i.e., the Kullback-Leibler divergence) between the $nD$ and 2D data distributions. The distribution of points in $nD$ has the form of pair-wise conditional probabilities indicating how likely the two data points are neighbors of each other. The distribution of points in 2D is modeled using a Student $t$-distribution, which is selected to ‘spread out’ the points from the high-dimensional space in the projection so that the neighborhood relations are preserved. $t$-SNE next computes the 2D locations of the points by stochastically minimizing the Kullback-Leibler divergences by gradient descent starting from a randomly-initialized configuration of 2D points.

While $t$-SNE is able to compute projections with strong visual cluster separation, it also has several drawbacks. The computational complexity is quadratic in the number of observations $N$, due to the calculation of SNE gradients that use $N \times N$ pairs of points [210]. This is a major issue for analyzing data sets with a large number of observations (and dimensions) like astronomical surveys, which will be introduced in the next section. Separately, the projection results of $t$-SNE are difficult to predict and replicate, due to the stochastic nature of $t$-SNE [133]. Setting the various hyperparameters of the method is also tricky: small changes in their values can lead to significant changes in the projection and determining the optimal parameter values for a given data set is not (yet) known. Moreover, $t$-SNE does not have the aforementioned OOS support and is furthermore sensitive to small changes in its input data. These issues for $t$-SNE and also other DR methods are discussed more in detail in Chapter 4.

A more-recent neighborhood preservation method is Uniform Manifold Approximation and Projection (UMAP) [145]. UMAP also aims to preserve neighborhoods computed from of a graph extracted from the high-dimensional data observations. UMAP is faster than $t$-SNE, has parameters that are easier to set, and incorporates the OOS ability. However, typical UMAP projections show too-strong clustering of similar observations, often to the level where a large amount of the available whitespace in the projection is not used to depict data. This issue can further cause problems with the visual exploration of the projection as a large number of points are overplotted (drawn atop of each other).
1.1.6.3 Labeling

Clustering and labeling are two closely related concepts in data science, machine learning, and interactive visualization, and obtaining projections with a high visual cluster separation supports data labeling. We now introduce the relation between clustering and data labeling.

Clustering, a branch of unsupervised learning, aims to group observations using a measure of similarity and proximity among data points without (pre-)labeled training data [53]. This serves multiple purposes such as partitioning a given data set into a (small) set of subsets and simplifying a data set by extracting a few representatives from each of its clusters. Additional applications of clustering algorithms include identifying natural modes or types of samples in distributions, classification, data aggregation, simplification, and data visualization. Although clustering algorithms do not require predefined labels, they still need a priori knowledge of the data. For example, $k$-means clustering requires the number of clusters to be specified explicitly [134], hierarchical clustering requires defining a similarity threshold [103], and DBSCAN requires specifying the minimum number of neighborhoods required to form a dense region [16, 125]. In contrast to these methods, the work in this thesis does not aim to create a ‘hard’ partitioning of a data set (or its projection) into perfectly separated clusters. Instead, it aims to create projections in which the apparent visual clusters (seen in terms of local agglomerations of points) reflect the existing distribution of data points as much as possible in the high dimensional space.

Labeling is the process of assigning (usually categorical) values to the points in a data set. These labels typically reflect properties which can be inferred in some way from the existing attributes or dimensions already present in the data. In classical supervised learning, classification algorithms are trained to create models that can assign labels (as observed from an existing training set) to new, unseen or unlabeled data points. However, one typically needs a large collection of already-labeled data to train such models and this collection should be large enough so that the distribution of values to be encountered in the application at hand is sampled sufficiently. When such rich labeled data sets are not available, one can use semi-supervised learning algorithms, which attempt to use a small labeled data set to propagate labels to unlabeled points prior to training a conventional classifier [15, 38, 223].

Besides the aforementioned methods that automatically propagate labels to data points, there also exist methods that involve the user in this process. At a high level, given a data set $D$, such methods present a visualiza-
tion $V(D)$ to the user in which all relevant data dimensions are depicted, e.g., a projection $P(D)$. Next, the user explores this visualization interactively and decides which labels can be assigned to certain data points. Next, such labels can be used by machine learning algorithms, e.g., active learning (AL) methods [17]. Conversely, the underlying machine learning algorithms can inform and assist the user about which samples they would like the user to label next. Visualization generally serves here as the key ingredient which lets humans and algorithms interact with each other to lead to efficient construction of effective machine learning models. For clarity, the term ‘labeling’ is used to indicate the process of assigning class labels to either the $n$D data or the visualization thereof. The correspondence of data points $D$ to points of a projection $P(D)$ is one-to-one – even when multiple points in $P(D)$ have the same 2D coordinates, as they have different identities. Therefore, labeling a point $P(x_i)$ in a projection (with suitable interaction techniques to disambiguate between points that exactly overlap at the same location) directly labels the corresponding point $x_i$ in the data set $D$ and vice versa.

Visual cluster separation is an important feature in the labeling process for several reasons. Having a projection in which users can easily distinguish separate clusters favors and guides the exploration and investigation of these clusters one by one, e.g., by interactive brushing, as opposed to having a projection in which no distinguishable visual structure exists. Moreover in the case that the user decides that all points in a cluster share sufficiently similar properties to be assigned the same label (by interactive brushing or similar mechanisms to expose the points’ attributes), the actual labeling process is simple. One can easily select all its points by e.g. lasso selection or rubberband selection and next assign them all the desired label, as the resulting cluster is visually well separated from other structures in the projection. Hence the goal of creating projections with a good visual cluster separation supports the more general task of data labeling.

1.2 OUTLINE OF THE SHARPENED DIMENSIONALITY REDUCTION METHOD

As is already apparent from the previous sections, various DR methods have been proposed with different advantages and disadvantages. Selecting the appropriate DR depends on what one wants to achieve from using DR and also on the type and traits of the data. Espadoto et al. have benchmarked DR methods by assessing their projection quality [62]. This benchmark has shown that there is no single appropriate method to guar-
antee the visual separation of clusters in the projections which also has other desirable characteristics: computational scalability, stability to data and/or hyperparameter changes, and OOS support.

For these reasons, the work in this thesis suggests a more generic approach to improve upon existing DR methods to gain the visual cluster separation for the purpose of exploratory analysis of high-dimensional data. The proposed approach first sharpens the potential clusters in \( n \)-D – meaning that clusters become more separated from each other – and then projects this \( n \)-D data into 2D or 3D so that the clusters are also visibly separated also in 2D or 3D. This approach is called Sharpened Dimensionality Reduction (Sharpened DR or SDR).

![Figure 1.5: Concept of SDR using 2D data. The 2D point cloud is sharpened by shifting points upstream in their density gradient to create separated clusters. The trajectories of the shifted points are shown using a black-to-red (start-to-end of shifting) colormap in the bottom figure.](image)
A conceptual illustration of SDR producing a clearer separation of clusters is shown in Fig. 1.5 using 2D data. The ‘sharpening’ step shifts the original high-dimensional data so that dense regions become more dense. This sharpening method is based on the work of Fukunaga and Hostetler who proposed Gradient Clustering (GC) as an application of mean-shift to cluster a set of points into separate groups based on their density [73]. In a different field, a very similar approach is used to create visually simplified drawings of large graphs or trail-sets [194]. This thesis adapts and exploits a GC approach to create projections with high visual cluster separation from high-dimensional data.

After the sharpening step, the sharpened data are projected to 2D using any existing DR method of choice. The key advantage is that one can choose methods which excel in certain aspects depending on the application or goal to optimize for, e.g., computational scalability, ease of use, stability, or OOS.

1.3 Applications in Astronomy

As mentioned at the beginning of this chapter, recent technological development has made it possible to acquire and store large volumes of big data in various fields of studies. This thesis concentrates particularly on applications in astronomy, a study of the ‘longest’ history of the Universe. Expanding the question of where we humans come from, astronomy aims to understand how our Universe has evolved to be the state it is in now and how it will look like in the future. Instead of a history book containing chronologically ordered events of mankind, astronomers look farther into the night sky to get a glimpse of how our Universe was formed, as the farther we look the further we travel back in time. Finding answers to these scientific, philosophical and fundamental questions of the Universe by understanding what is drawn (and still being drawn) on the black canvas above is what makes astronomy such a mysterious and fascinating subject.

While our ancestors studied this black canvas by eye, nowadays sophisticated instruments are available to capture visible light emitted from stars (stellar surveys) and also invisible signals in the form of different wavelengths, such as radio waves and x-rays. Astronomers have constructed these instruments on high mountain tops and even launched outer-space probes to step even higher above into the night sky to acquire high-quality signals.
1.3 applications in astronomy

Especially for stellar surveys, the recent accumulation of data has produced large quantities of high-dimensional data – from hundreds of thousands and up to $10^9$ stellar observations for hundreds of stellar attributes. For instance, *Gaia* includes astrometric, photometric, and (for bright stars) spectroscopic information of stars [43, 209]. Other recent surveys of stellar spectra include GALactic Archaeology with HERMES (GALAH) [23, 25, 137], Sloan Digital Sky Survey (SLOAN) [3], Radial Velocity Experiment (RAVE) [186, 187], Apache Point Observatory Galactic Evolution Experiment (APOGEE) [135], and Large Sky Area Multi-Object Fibre Spectroscopic Telescope (LAMOST) [46].

Given the large amount of samples and dimensions that such data sets contain, we argue that our earlier discussion for the added value of visualization – and in particular dimensionality reduction techniques having visual cluster separation – is directly applicable to the astronomical context. We further detail the way into which clustering, labeling, multidimensional visualization, and dimensionality reduction are currently used in astronomical studies.

1.3.1 Clustering in astronomy

A common goal concerning these astronomical stellar surveys is to separate physically meaningful clusters based on high-quality signals of unlabeled stars described by multiple attributes so that labels can be assigned to represent separate physical groupings of individual stars. In other words, this process creates (a) groups of related samples (stars) and (b) explains these groups at a higher level by means of labels which synthesize properties that the stars share in terms of their attribute values. This process allows scientists to reason about the large amount of available survey data at a higher level and in a less complex way.

Labels can be acquired either automatically by using algorithmic methods or manually by involving users in the labeling process, as follows. For automatic labeling of clusters, typical clustering methods can be used. There are various types of clustering methods applied to astronomical data such as partition-based methods which search for optimal cluster centers [129]; density-based methods which consider points with high density as the same cluster such as DBSCAN [125], which has recently gained popularity [92, 164]; hierarchy-based methods that use bottom-up agglomerative grouping of similar samples based on various types of distance computations (e.g., single linkage, average linkage, or full linkage) [107]; model-based methods that mainly use neural network models or probability-based models [118, 217]; and graph-based methods that op-
erate on graphs [232]. For a full review of clustering methods recently applied to astronomical spectra data, we refer to the study by Yang et al. [232]. Unlike automatic labeling using clustering methods, manual labeling of clusters implies that such clusters are presented to the expert user so that one can study them and can decide how certain labels can be assigned depending on their properties. Key to this process is presenting the clusters in context. Simply listing clusters, for instance, is arguably not intuitive, as one has to make sense of what the respective clusters mean. A simple solution is to use multidimensional visualization methods that we now discuss.

1.3.2 Multidimensional data visualization in astronomy

Multidimensional data visualization methods in astronomy have a clear advantage of providing the user with intuitive depictions of the underlying data. Visualization can directly depict the raw (unlabeled and/or unclustered) data in case that labeling and/or clustering is not yet done or desired. Alternatively, visualizations can help understand and refine the results of clustering. An example is projecting the clustered high-dimensional data to 2D and color-coding the points by their cluster labels [232]. Compared to other more abstract visualizations of clustering processes, dendrograms also help domain experts to better understand the size, spread, neighboring relations, and outliers of clustered data via the projection (e.g., see Fig. 1 in [130] and [175]). Table lens or PCP techniques can also be used to list the respective clusters and their aggregated properties (in terms of the available data dimensions) [167, 226]. However, as explained in Sec. 1.1, such techniques do not scale well to more than roughly a dozen dimensions and there are cases when one needs to see the actual observations and not only the clusters computed from them. Overall, multidimensional data visualization can be seen as an additional tool for domain experts instead of a competing tool to help cluster high-dimensional data.

Scatterplot matrices (SPMs) are also one of the basic visualization tools widely used in astronomical studies (see also Sec. 1.1.3). SPMs have been widely used in the literature for exploratory analysis and validation purposes in astronomy, due to their simplicity in depicting relations between different dimensions [31, 108, 191, 220]. Astronomers often call SPMs ‘Corner plots’, as reflected in the commonly used Python module (‘Corner.py’ [71]) for plotting SPMs [31, 220]. Other studies have used SPMs combined with other multidimensional data visualization methods such as PCPs [144, 218]. However, as mentioned earlier in this chapter, the main
limitation of these methods is their low-dimensional scalability (typically no more than a dozen dimensions at a time). Another problem of these attribute-centric methods is that they do not easily map concepts such as sample similarity, as they are attribute-centric methods aiming to show the values of each attribute separately. This is problematic if one wants to visually explore clusters of similar observations to reason about their assigned labels – or, alternatively, assign labels to them.

1.3.3 DR in astronomy

Unlike other multidimensional data visualizations, DR is a more observation-centric method that is able to visualize a large number of samples having many dimensions and can show the cluster-like structure of a data set in terms of existing groups of highly-similar samples as discussed in Sec. 1.1.4. This serves both cases where one has not actually executed a clustering operation on the data but wants to use the visual cluster structure to infer actual data clusters; and cases when such a clustering has been executed, and then the projection is used to show its results by e.g. color-coding the projected points by the computed cluster labels.

As discussed earlier, the effectiveness of DR for data exploration purposes highly depends on the visual cluster separation properties of the selected DR method (Sec. 1.1.6). Previous attempts at using standard DR methods for obtaining projections with a high visual cluster separation have not been completely successful, as outlined next.

High-dimensional data analysis using DR for clustering purposes – detecting compact groups of similar points in the projection and next reasoning about each of these – has widely been used in astronomy starting with one of the earliest DR methods, PCA [151]. Since its first application in astronomy [50], PCA continues to be widely used by astronomers, e.g., to explore the space of chemical compositions of stars, describe how many controlling parameters exist [199], and find clusters in that space [22].

More recent astronomy studies have used t-SNE [7, 204]. Anders et al. [7] show how domain experts use t-SNE to manually label interesting points and clusters in the 2D “abundance-space” (the term used in astronomy to denote projection space of stellar abundance data, which will be defined later in Chapter 5). However, this study attempts to manually label data clusters based on nearby, often-overlapping, and sometimes very small clusters in the 2D projection, as the cluster separation is not clear and this can lead to highly uncertain labels. Moreover, the labels generated did not arise solely from using the t-SNE projection but also from analysis of an
SPM of the original abundance-space data in an iterative process with the $t$-SNE projection (see Fig. 1 in [7]). Using a combination of two visual techniques is arguably more complex and time consuming than using a single technique that is sufficient for the labeling task. The aforementioned challenges of using $t$-SNE for data labeling is detailed in Chapters 2 and 5.

1.3.4 Application goal

The application component of this thesis aims to provide a useful tool for scientists in astronomy and also other domains to visually analyze high-dimensional data sets for data exploration purposes. In detail, this thesis demonstrates a use-case in astronomy where our proposed method of sharpened DR (introduced in Sec. 1.2) improves the limitation of low-quality visual cluster separation of previously applied DR methods on astronomical data sets and supports the exploration and reasoning about astronomical high-dimensional data. To achieve this goal, the work of this thesis reproduces the work from Anders et al. with GALAH, a larger and more-recent survey of stellar abundances. The GALAH data set is a large-scale spectroscopic stellar survey focusing on the chemical composition of several hundreds of thousands of stars. Separately, measurements from the nine-year-old space probe Gaia, launched by the European Space Agency (ESA), covers more than a billion stellar observations and is used to compensate for the missing or insufficient motion and kinematics data in GALAH. Using the GALAH and Gaia data sets, S. C. Trager – one of the co-authors of Chapters 2, 4, and 5 – aims to find the chemical composition of stars to ultimately unravel the formation and evolutionary history of the Milky Way. The obtained findings related to the observed relations between chemical compositions of stars and the formation of the Milky Way is explained in detail in Chapter 5.

1.4 This thesis

Summarizing the previous sections, the following aspects are the key desirable properties or requirements for a DR technique:

- **Quality** in terms of visual cluster separation;
- **Dimensional scalability** in terms of supporting tens to hundreds of dimensions;
- **Computational scalability** in terms of computing time for data with many dimensions and samples;
• **Genericity** in terms of whether the DR method is applicable to various data types (i.e., any labeled or unlabeled real-valued data)

• **Ease-of-use** in terms of parameters the user needs to set to produce the visualization;

• **Stability** when fitting new data to an existing projection (out-of-sample support).

As discussed so far, many DR methods exist which deal well with all of these aspects aside from the first – and conversely, the methods which handle well the first property fail in various degrees in handling the other subsequent properties. Therefore, the main challenge relates to how to add the first property to existing DR methods that can handle well the subsequent properties. Based on this challenge, the research question of this thesis can be formulated as follows:

*How can we improve the quality of DR methods in terms of visual cluster separation (VCS) for exploratory data analysis of unlabeled, large, high-dimensional data sets?*

To answer this question, the work of this thesis suggests a generic method to improve the visual cluster separation of existing DR methods by utilizing a simple pre-conditioning step called *sharpening* applied prior to DR. Given that the selected DR methods to be preconditioned in this way already have all the other properties except visual cluster separation, the proposed method adds this desirable property without removing the other key properties. Besides presenting the sharpened DR idea, the remainder of this thesis shows how this method can be extended in terms of computational scalability using neural networks and also be used to assist tasks such as labeling and clustering data.

The remainder of this thesis is structured as follows.

Chapter 2 first explains the proposed SDR method in detail. SDR is validated quantitatively and qualitatively based on quality metrics and perceived visual cluster separation on a number of labeled synthetic and real-world data sets. Finally, the proposed method is applied to a subset of the second data release of GALAH (GALAH DR2) to demonstrate the potential use-case in astronomical applications.

Chapter 3 briefly demonstrates how clustering can be used in line with SDR to ease the manual labeling of clusters in the resulting SDR projections, with the focus on (labeled) human-activity data applications. Sev-
eral clustering methods widely used are applied to the projections of synthetic data and real-world data sets using SDR and are quantitatively and qualitatively validated. Particularly for the labeled human-activity data, results show that the separated clusters are highly related to their physical meanings (e.g., dynamic and static human motions). However, the scalability of SDR and an additional clustering parameter atop the three parameters of SDR make it difficult for the user to easily benefit from applying clustering to SDR.

Chapter 4 presents how Neural Network Projections (NNP) can be used to improve the scalability of SDR while retaining the other key properties of DR – quality, genericity, and OOS support. Similarly to the previous chapter, GALAH DR2 is used to demonstrate the usage of this proposed method, SDR-NNP. Furthermore, a modified version of SDR-NNP, \( \alpha \)-SDR-NNP, uses pseudo-labels created by applying a clustering method (K-means) to the original \( n \times d \) data and using the pseudo-labels to modify the sharpening step. \( \alpha \)-SDR-NNP improves the ease-of-use by allowing users to control only a single parameter (\( K \) in K-means) instead of controlling the interdependent parameters of SDR. This single parameter is also shown to be stable in terms of OOS.

Chapter 5 focuses on analyzing the resulting clusters from a SDR projection of the third data release of GALAH (GALAH+ DR3) in depth to find physical meanings of the depicted clusters. This chapter includes a great deal of prerequisites in astronomy to understand the analysis steps of the resulting clusters present in the projection results; hence, this chapter starts by introducing the important terminologies used in literature. Next, the clusters from the SDR projection are manually labeled and subsequently analyzed in depth. The analysis concludes that the resulting clusters are associated with different subcomponents of the Milky Way, which demonstrates a practical use-case of SDR on an unlabeled astronomical data set.

We conclude in Chapter 6 by reflecting on how the research question from this chapter is addressed. Several aspects, limitations, and possible directions for future improvements of our work are discussed.
1.4 THIS THESIS
Applying dimensionality reduction (DR) to large, high-dimensional data sets can be challenging when distinguishing the underlying high-dimensional data clusters in a 2D projection for exploratory analysis. We address this problem by first sharpening the clusters in the original high-dimensional data prior to the DR step using Local Gradient Clustering (LGC). We then project the sharpened data from the high-dimensional space to 2D by a user-selected DR method. The sharpening step aids this method to preserve cluster separation in the resulting 2D projection. With our method, end-users can label each distinct cluster to further analyze an otherwise unlabeled data set. Our ‘High-Dimensional Sharpened DR’ (HD-SDR) method, tested on both synthetic and real-world data sets, is favorable to DR methods with poor cluster separation and yields a better visual cluster separation than these DR methods with no sharpening. Our method achieves good quality (measured by quality metrics) and scales computationally well with large high-dimensional data. To illustrate its concrete applications, we further apply HD-SDR on a recent astronomical catalog.

2.1 Introduction

Dimensionality reduction (DR) techniques depict high-dimensional data with low-dimensional scatter plots. DR is widely used because it preserves the structure of high-dimensional data. For example, when the data is distributed over several clusters, DR allows one to directly and visually examine such structures in 2D or 3D, in terms of visually well-separated point clusters in a scatterplot. While t-distributed Stochastic Neighbor Embedding (t-SNE [133]) is arguably one of the best DR techniques in creating visually well-separated clusters of similar-data points, the recent work of Anders et al. shows that even with t-SNE, when visual clusters overlap even slightly, manually labeling them can be challenging [7]. Besides
t-SNE, many other nonlinear DR techniques have been proposed, e.g., Random Projection (RP) [48, 230], Landmark Multidimensional Scaling (LMDS) [181], ISOMAP [195], Sammon Mapping [177], and Uniform Manifold Approximation and Projection (UMAP) [145]. While such methods typically achieve a poorer visual cluster separation than t-SNE [122, 133], they are computationally more scalable and simpler to implement and use [63].

Espadoto et al. [63] have benchmarked dozens of DR techniques using several quality metrics and showed that there is no ‘ideal’ DR technique that guarantees the visual separation of similar-data clusters for any kind of data. As such, we are interested to find a generic approach to improve upon existing DR methods in terms of visual cluster separation while keeping other attractive specific features these already have, e.g., neighborhood and distance preservation, computational scalability, or simplicity.

In this chapter, we show how sharpening the clusters in the original high-dimensional data can enhance Visual Cluster Separation (VCS) – loosely defined, for now, as the ability of a user to see separate clusters in a 2D projection. A more formal definition and explanation of the importance of VCS is introduced in Section 2.2.1. We sharpen the data clusters by Local Gradient Clustering (LGC) and then project the sharpened data to 2D using standard DR techniques. When the input high-dimensional data has cluster structures, our ‘High-Dimensional Sharpened DR’ (HD-SDR) method creates projections that show these clusters more clearly and better separated from each other than when using the baseline, original, DR method alone. As such, our approach is not a new DR technique, but a new way to enhance the VCS properties of any existing DR technique. To our knowledge, this the first time that such a sharpening approach is used to enhance VCS without any prior estimation of cluster modes [33].

We evaluate our HD-SDR method on synthetic and real-world labeled data using quality metrics that empirically and theoretically measure the preservation of neighbors and their corresponding labels, and use RP, LMDS, t-SNE, and UMAP as the baseline DR methods [48, 133, 145, 181]. By comparing the baseline DR with HD-SDR, our results show that sharpening assists those DR methods, which have difficulty in producing visually well-separated clusters, and create projections with clear VCS.

To demonstrate the practical usefulness of HD-SDR, we apply it to explore an unlabeled real-world astronomical data set drawn from the recent GALactic Archaeology with HERMES Data Release 2 (GALAH DR2) and Gaia Data Release 2 (Gaia DR2) catalogues [7, 25, 74, 75]. Astronomers are able to label and further analyze each distinct cluster using our method.
This use-case shows how our method can easily assist domain experts to manually and visually label data clusters by annotating their 2D projections, which leads to a better understanding of the large high-dimensional data at hand. Although currently out of our scope, HD-SDR could further be used to assist user-guided labeling in semi-supervised learning, where small portions of labeled data (given or manually assigned by end-users) are used to propagate labels to unlabeled data, which are then used to train a conventional classifier [14, 15, 17, 38, 223].

In summary, our contributions are as follows:

- We propose a novel method to improve Visual Cluster Separation (VCS) of DR methods by sharpening the original high-dimensional data prior to the projection. This is to our knowledge the first time that such a sharpening method is used to improve VCS in DR without any prior estimation of the cluster modes;

- We demonstrate both qualitatively and quantitatively that our method enhances VCS for DR methods that originally show weak cluster separation;

- We apply our method to unlabeled real-world astronomical data and show evidence that the resulting visual clusters have a physical meaning in our Milky Way Galaxy.

This chapter is structured as follows. Section 2.2 outlines related work in dimensionality reduction. Section 2.3 details our method. Section 2.4 compares our method qualitatively and quantitatively with standard DR on several synthetic and real-world data sets. Section 2.5 shows a practical use-case with unlabeled real-world astronomical data. Section 2.6 discusses several aspects of our method, including its cluster segregation power, data distortion, scalability, and limitations. Section 2.7 concludes the chapter.

2.2 RELATED WORK

We first briefly discuss the relation between cluster separation and DR used for exploratory analysis in Section 2.2.1. We next explain the importance of cluster separation in DR used for data labeling (Section 2.2.2), followed by specific use-cases of DR in astronomy, our main application area (Section 2.2.3).
2.2.1 Dimensionality Reduction and Cluster Separation

While DR has multiple goals such as data compression [10], feature extraction [20], and exploratory analysis [153], we focus here on the exploratory analysis using DR methods to visually support identifying clusters of similar-value data points. Finding separate clusters, defined as sets of unlabeled data points that have similarities but are different from other point sets, is challenging in data science and unsupervised learning. Data clustering serves multiple aims: e.g., finding natural modes or types of samples in distributions; classification; data aggregation and simplification; and data visualization. Although clustering algorithms do not explicitly use predefined labels as in supervised learning, they still need a priori knowledge of the data. For instance, \( k \)-means clustering explicitly requires the number of clusters [134]; hierarchical clustering requires defining a similarity threshold [103]; and DBSCAN asks for the minimum number of neighborhoods required to form a dense region [16, 125]. Given the above, there is no unique and/or ‘correct’ clustering of a given data set. Instead, the ‘cluster structure’ present in a data set is implied by a given clustering method and its hyperparameters.

Let \( D = \{x_1, \ldots, x_N\} \) be a set of \( n \)-dimensional observations (samples, points), \( x_i = [x_i^1, x_i^2, \ldots, x_i^n] \in \mathbb{R}^n \). Here, \( x_i^j \) \((1 \leq j \leq n)\) is the \( j \)th attribute value of the \( i \)th sample. A DR technique, or projection, can be modeled by a function \( P : \mathbb{R}^n \rightarrow \mathbb{R}^s \). In practice \( s = 2 \) is most used – for details we refer to Martins [140]. A projection function \( P \) allows one to reason about a data set \( D \subset \mathbb{R}^n \) by visually interpreting its projection (scatter-plot), which we denote as \( P(D) = \{P(x_i)|x_i \in D\} \). Hence, if data structure in terms of clusters exists in \( D \) (in the sense outlined in the previous paragraph), these should also be visible in \( P(D) \). A projection \( P \) reflects the data cluster separation present in \( D \) by the visual cluster separation present in \( P(D) \) [168, 169]. Note that the function \( P \) from \( \mathbb{R}^n \) to \( \mathbb{R}^2 \) is, in general, many-to-one in terms of point locations, i.e., points that have different coordinates in \( \mathbb{R}^n \) can be mapped to the same location in \( \mathbb{R}^2 \). Yet, every sample point \( x_i \in D \) is mapped to a unique point \( P(x_i) \in P(D) \) by using the index \( i \) as an identifier both in \( D \) and \( P(D) \). For these reasons, we use the term ‘labeling’ to refer to adding class labels to either the 2D or the \( n \)D data sets, as labeling a point \( P(x_i) \in P(D) \) in the projection directly labels its corresponding point \( x_i \in D \) in the data set, and conversely.

As mentioned earlier in Chapter 1, we can define (visual) cluster separation more formally as follows. Let \( H : S \rightarrow X \) be any metric or tool that is able to reason about the clusters present in a data set \( S \). Let \( H(S) \) denote the application of \( H \) to all points in \( S \). When \( S = D \), \( H \) captures the data
cluster separation. When $S = P(D)$, $H$ captures visual cluster separation (VCS). Examples of $H(D)$ are classifiers that assign a label $x \in X$ to data points so that points in the same cluster get the same label; clustering techniques that assign a cluster ID to similar data points (in this case $X \subset \mathbb{N}$) or count the number of clusters in a data set (in this case, $X = \mathbb{N}$). Several instances of $H(P(D))$ have been proposed to measure VCS in visualization research [63]. Given the above, we say that a projection $P$ has good VCS when $H(P(D))$ is very similar to $H(D)$, i.e., $P$ should ideally capture in the 2D visual space the same cluster structure that the metric $H$ finds in the data space. Note that ‘good VCS’ does not identically mean ‘high VCS’. Rather, good VCS implies two cases: (a) When $H(D)$ is high (data is well separated in the high-dimensional space), then $H(P(D))$ should also be high; and (b) when $H(D)$ is low (there is no clear cluster structure in the data), then $H(P(D))$ should also be low (the projection should not create artificial visual clusters that wrongly suggest that the data has this type of structure). Following this, there are two cases when $H(P(D))$ does not reflect well $H(D)$: we say that (1) $P$ undersegments the data $D$ if $H(D)$ contains more clusters than $H(P(D))$; this can be seen as $P(D)$ showing ‘false negatives’ in terms of missing visual clusters; and (2) $P$ oversegments $D$ if $H(D)$ contains fewer clusters than $H(P(D))$; this can be seen as $P(D)$ showing ‘false positives’ in terms of spurious visual clusters.

Visual cluster separation in distance-preserving projections of intrinsically low-dimensional data, where the $n$D distances are reflected well by their corresponding 2D counterparts, is easier to spot based on the distances between clusters. However, we argue that these cases of intrinsically low-dimensional data embedded into high-dimensional spaces are a minority. When exploring high-dimensional data by non-linear projections or projections that do not preserve distances (but neighborhoods), looking at visual clusters in $P(D)$ is the only way to reason about $D$ because the exact inter-point distances in $P(D)$ have little meaning. Hence, for such projections, VCS is also important. This is also reflected in methods such as t-SNE and UMAP [133, 145].

### 2.2.2 Dimensionality reduction for labeling

Semi-supervised learning methods propagate labels from a small set of predefined labeled data to the remaining unlabeled data points prior to training a conventional classifier [14, 15, 38, 223]. These methods take advantage of DR by letting users assign or propagate labels directly, and visually, in a projection. In visual analytics, user-centered Visual-Interactive Labeling (VIL) is combined with model-centered Active Learning (AL) to
achieve better labeling [17]. Such methods are highly effective when not enough labeled training data exist and/or when users need more control on label propagation. Yet, VIL requires strong VCS so users know when to stop visually propagating a label [14, 15], which not all DR methods deliver, as mentioned earlier.

$t$-SNE is a well-known nonlinear DR method which aims to preserve neighborhoods of a given point. Its popularity is arguably due to its good ability to separate similar data clusters present in high-dimensional spaces to create a strong visual separation of clusters in the 2D projection [133]. Recently, Bernard et al. showed that $t$-SNE is the preferred DR method for labeling, when compared with other methods such as non-metric MDS, Sammon mapping, and Principal Component Analysis (PCA), due to its clear cluster separation [17]. Lewis et al. also confirmed that users prefer visualization methods that clearly separate clusters (assuming, of course, such clusters exist in the data), as this is seen as a sign of quality of the method [17, 124].

However, $t$-SNE’s complexity is quadratic in the number of points [210]. While accelerated variants exist [160–162], these are quite complex to implement and not yet widespread. Additionally, due to its stochastic nature and underlying cost minimization process, it is hard for users to predict the results of $t$-SNE for a given data set and parameter settings [225]. A more recent competitor, UMAP, has been introduced and used in astronomical applications [145, 170], but to date has not been widely applied and assessed by domain experts in that field.

2.2.3 DR in astronomy

While applications of DR include biomedicine, computer security, and various other fields [76, 221], our main application domain in this chapter is astronomy. We cover next the important use-cases of DR in astronomy to explain the importance, recent works, and limitations of DR, and to elicit the needs of domain experts when using such methods.

Astronomical data sets have long been considered ‘big’ data, and are still growing larger due to the advancement of sensor technology and signal processing capacity. The recent Gaia DR2 catalogue [25, 74, 75] contains more than 1.6 billion objects with tens of dimensions. Sifting through these big data catalogues is an excellent test for DR methods.

High-dimensional data analysis using DR for clustering purposes has widely been used in astronomy, starting with one of the earliest DR methods, PCA [151]. Since its first application in astronomy [50], PCA contin-
ues to be widely used by astronomers, e.g., to explore the space of stellar elemental abundances and describe how many controlling parameters exist [199], and to find clusters in that space [22].

More recent studies show the use of $t$-SNE in astronomy [7, 204]. Anders et al. [7] show how domain experts use $t$-SNE to manually label interesting points and clusters in the 2D “abundance-space” (the term used in astronomy to denote projection space) using a number of stellar abundances as input. However, they attempt to manually label data clusters based on nearby, often-overlapping, and sometimes very small clusters in the 2D projection, which can lead to highly uncertain labels. Moreover, the labels generated did not arise solely from the $t$-SNE projection but also from analysis of a scatter plot matrix of the original abundance-space data in an iterative process with the $t$-SNE projection (see Figure 1 in [7]). We explore the above challenges of using $t$-SNE for data labeling in Section 2.5.

In summary, previous work has shown (1) the importance of DR in data labeling; (2) that a clear separation of clusters provide an intuitive labeling experience by end-users; (3) that $t$-SNE is the current state-of-the-art DR method used in manual labeling and user-guided label propagation; and (4) that $t$-SNE may not always give clear cluster separation, which explains the quest for an alternative DR method that provides a clear visual separation of clusters for users to easily label these clusters.

### 2.3 Proposed Method

We next present our method, which consists of two main steps: local gradient clustering (Section 2.3.1) and actual projections using RP, LMDS, $t$-SNE, and UMAP (Section 2.3.2).

#### 2.3.1 Local Gradient Clustering

As outlined in Section 2.2, we aim to obtain a high visual cluster separation in a projection by ‘preconditioning’ the DR method. Good candidates for this preconditioning are mean shift-based methods [34, 44, 73, 99, 229]. Such methods have previously been suggested to estimate modes of clusters, in combination with DR, to cluster and visualize high-dimensional data [33]. In contrast, our method does not need the mode information to create a high visual cluster separation.

Mean shift-based methods estimate the sample density using kernel density estimation (KDE) [183] and iteratively shift samples upwards in the
density gradient. For a data set $D = \{x_i\}$, we define the multivariate kernel density estimator at location $x \in \mathbb{R}^n$ by

$$\rho(x) = \sum_{x_i \in N(x)} K\left( \frac{\|x - x_i\|}{h} \right),$$

(2.1)

where $K(\cdot): \mathbb{R}^+ \rightarrow \mathbb{R}^+$ is a radially-symmetric univariate kernel of bandwidth $h$. $N(x)$ denotes the set of samples $x_i$ which affect the density $\rho$ at location $x$. In classical KDE [73], $N(x) = D$, i.e., all samples affect all density locations. Another possibility is to use only samples closer to $x$ than $h$, i.e., $N(x) = \{x_i \in D : \|x - x_i\| < h\}$. This offers a better local control of the scale of patterns (clusters) formed by mean shift and also significantly accelerates the density estimation [99]. However, this assumes that all data clusters have comparable scale in $D$, and that this scale ($h$) is known, which typically is not the case with high-dimensional data sets of varying density. We refine the above model by locally setting $h$ to the distance between $x$ and its $k_s$-nearest neighbor in $D$. The actual implementation can use different scaling factors for $h$ (e.g., $h^2$). The free parameter $k_s$ thus determines the simplification scale of the data set. More intuitively, all $k_s$-nearest neighbors of a point $x$ are considered to be in the same cluster as $x$.

After estimating $\rho$ over $D$, we shift all samples $x_i \in D$ for $t$ iterations along the density gradient by the following update rule

$$x_i^{next} = x_i + \alpha \frac{\nabla \rho(x_i)}{\text{max}(\|\nabla \rho(x_i)\|, \epsilon)},$$

(2.2)

where $\alpha$ is the ‘learning rate’, which determines the convergence speed of the process, and $\epsilon = 10^{-5}$ is a fixed regularization factor used to handle gradients near zero. For $K$, we use an Epanechnikov (parabolic) kernel, which is optimal for KDE in a mean-squared error sense [59]. This kernel yields smaller movements (shifts) as compared to a Gaussian kernel, thereby favoring the stability of the process. Note that we use implicit integration to consider the updated points when calculating the gradient and that Eqs. 2.1 and 2.2 are coupled, as we estimate the gradient $\nabla \rho$ (Eq. 2.1) after every iteration. This means that we perform the nearest neighbor search for every iteration as in Hurter et al. [99]. In contrast, classical Gradient Clustering (GC) [73] performs nearest neighbor search only for the first iteration, and uses those neighbors in subsequent iterations. Hurter et al. showed advantages of nearest neighbor search at every iteration in terms of robustness of the sample shift with respect to parameter tuning. Hence, we follow the same approach. Due to our usage of nearest neighbors, we call our sharpening approach Local Gradient Clustering (LGC),
2.3 PROPOSED METHOD

Figure 2.1: Effects of parameters used in LGC. 2D Gaussian data with 10K observations and three clusters (a) are used to show the effects of the number of iterations \((t)\) as shown in (a)–(d), number of nearest neighbors \((k_s)\) in (e)–(l), and learning rate \((\alpha)\) in (m)–(p). Points are color-coded based on their ground-truth labels. The cluster borders become fuzzy when using a too high \(t\), as shown in (d). \(k_s\) and \(\alpha\) both contribute to the degree of segmentation of the clusters; without choosing an appropriate \(\alpha\), \(k_s\) may not significantly affect the segmentation, as shown in rows (e)–(h) and (i)–(l). Note that \(\alpha\) uses a fixed range of \([0, 1]\).

by analogy with Gradient Clustering (GC). The key added value of LGC discussed in this chapter is its preconditioning of the data that leads to better results of DR techniques.

Figures 2.1–2.2 show the effect of the free parameters \(t\) (number of iterations), \(k_s\) (number of nearest neighbors), and \(\alpha\) (learning rate) for LGC. Color encodes ground-truth labels, which are known for these data sets. Each row of Figures 2.1–2.2 shows the results of varying a single parameter, with the other two parameters fixed. The data set \(D\) contains synthetic Gaussian random data \((N = 10K\) and \(n = 2)\) for Figure 2.1 and non-Gaussian (log-normal, \(\mu = 0\), and \(\sigma = 1\)) random data \((N = 10K\), \(n = 2)\)
Figure 2.2: Effects of different parameters using 2D non-Gaussian (log-normal, $\mu = 0$, and $\sigma = 1$) data with 10K observations. The effects of the parameters are similar to those in Figure 2.1. However, LGC with too large values of $t$ and $\alpha$ is prone to outliers (long tails), as shown in (d) and (l). This problem can be solved by setting a larger value of $k_s$.

Indeed, for $n = 2$, we can directly look at $D$ to assess LGC without DR. Note that similar behaviors are shown for higher $n$-values. The effect of the three parameters is as follows.

**Learning rate $\alpha$:** Controls the speed of shifts and affects the degree of segmentation, see the bottom rows of Figure 2.1 and 2.2. If $\alpha$ is too large, points move too far and can overshoot the mode of a cluster during LGC as shown in Figure 2.1(p) (see also Section IV-A in [73]). Conversely, too small $\alpha$-values yield too small shifts (Figure 2.1(m)) and thus can result in an oversegmentation of the data (too many small clusters). The interconnection between $\alpha$ and $k_s$ is discussed further in Section 2.6.5.3.

**Nearest neighbors $k_s$:** Controls how localized a shift is. Both $k_s$ and $\alpha$ affect the degree of segmentation; yet, without choosing an appropriate $\alpha$, $k_s$ may not significantly affect the segmentation, as shown in the second and third rows of Figure 2.1. Here, we empirically fix $k_s = 50$ based on the stability and speed of our method. Too small $k_s$-values can create oversegmentation (many small clusters) and can sharpen dense areas of noise making our method unstable (see detailed discussion in Section 2.6.5.2); a
too large value of $k_s$ increases the number of nearest neighbor searches resulting in slower computation (see Section 2.6.1).

**Number of iterations $t$:** This parameter controls the amount of cluster separation. If $t$ is too small, points will shift only a few steps along the density gradient, resulting in little difference from the original data. We have observed that intra-cluster points are close enough for clusters to be visually well separated using $t = 5$ for Gaussian synthetic data and $t = 10$ for non-Gaussian synthetic data. Varying $t = 10$ by a factor of two may not significantly change the obtained result, but too many iterations also add to the computing time (discussed next in Section 2.6.1). Setting $t = 10$ for all experiments in this chapter allows us to obtain a data separation that is sufficient to yield a clear visual separation in the DR projection of the preprocessed data.

Points can overshoot the local mode given their $k_s$-nearest neighbors when using a too-large $t$-value. This is why the borders of clusters in Figure 2.1–2.2(d) become fuzzier compared to those in Figure 2.1–2.2(c). This can be solved by using a smaller value of $\alpha$. A similar issue is solved by decreasing the advection step in time [99]. However, in that context, the aim was to collapse close data points to a single point. This is not the aim of VCS, so we cannot use that approach in our context.

Summarizing, we can use a single free parameter $\alpha$ to control the sharpening step after fixing the values of $k_s$ and $t$. The effects of $\alpha$ on speed are discussed separately in Section 2.6.1. We implemented LGC in C++ for higher-speed performance, using Nanoflann [19, 148, 149] for the nearest neighbor search in $\mathbb{R}^n$. We have also evaluated other nearest-neighbor search algorithms (see Section 2.6.1). Our code is publicly available [110].

### 2.3.2 Dimensionality Reduction Candidates for HD-SDR

As explained in Section 1, the aim of our method is to improve the visual cluster separation for existing DR methods which are lacking in this respect; and do this in a computationally efficient way and with minimal parameter-setting effort. We have achieved the first concern (cluster separation) in the data space by using LGC (Section 2.3.1). Now we test our method on several DR techniques that take the LGC-sharpened data as input and project it. We use three different DR methods from the publicly-available C++ Tapkee toolkit [126], as well as UMAP available in Python. These are selected based on the following requirements:

- no prior knowledge (labels) of the data;
• computational scalability to large data sets (tens of thousands of samples, hundreds of dimensions);
• ease of use in terms of free parameters with documented presets;
• showing (weak or strong) visual cluster separation.

Adding to the last requirement, our aim is to sharpen clusters in \( n \)D so that the clusters are also visually separable after DR, rather than creating clusters via DR methods that show no clustering ability. This is why we select DR methods that exhibit different degrees of cluster separation. Random Projection (RP), Landmark Multidimensional Scaling (LMDS), \( t \)-SNE, and UMAP are the methods that best meet the above criteria [48, 133, 145, 181].

The quantitative survey of DR methods of Espadato et al. [63] found \( t \)-SNE, UMAP, Projection By Clustering (PBC), and Interactive Document Maps (IDMAP) to have the highest global quality. Here, we use UMAP because it is a strong competitor of \( t \)-SNE and has also been recently applied to astronomy [170], our main application domain. Empirically, UMAP, \( t \)-SNE, LMDS, and RP, in descending order, show the strongest cluster separation in our study. Apart from the above, note that any DR method can be used in our proposed approach. To show this, we apply our sharpening method on a labeled real-world WiFi data set and feed it to the DR implementations from [63] (see supplemental material and Section 2.6.5).

We briefly introduce the selected DR techniques used in this chapter. Note that \( t \)-SNE was already explained in Section 2.2.

**Random Projection (RP):** Although nonlinear projections achieve better distance preservation for high-dimensional data [185, 195], we use the linear DR technique Random Projection (RP) to demonstrate the sharpening effect on a DR with relatively poor cluster separation. RP projects a random matrix consisting of orthogonal unit vectors to lower dimensions, aiming to preserve pairwise distances. RP needs less memory and is faster to compute than PCA [48, 227, 230]. RP is of order \( O(N \times n \times s) \), where the \( N \) samples in \( \mathbb{R}^n \) are projected to an \( s \)-dimensional subspace [48, 230].

**Landmark Multidimensional Scaling (LMDS)** is a nonlinear variant of Multidimensional Scaling [203]. Computational scalability (linear in the sample count \( N \)) is achieved by projecting a small subset of the so-called landmark samples (5% of \( N \) in most of our experiments) by classical MDS, around which remaining samples are projected using a fast triangulation procedure [181]. For completeness, we mention that we also used LMDS with increasingly more landmarks and obtained visually similar results (not included here for brevity).
2.4 results

We compare HD-SDR with DR on both synthetic data (Section 2.4.1) and real-world data (Section 2.4.2). We run all our experiments on a PC having a Core i7-8650U (2.11GHz) processor with 16G RAM.

2.4.1 Synthetic data: qualitative evaluation

We generated Gaussian random data consisting of five clusters \(N = 5K, n = 20\) to cover five types of inter-sample distance distributions:

1. even spread of inter-cluster distances with equal intra-cluster densities with equal Gaussian variance;
2. even spread of inter-cluster distances with different intra-cluster densities;
3. uneven spread of inter-cluster distances (skewed distribution);
4. two pairs of subclusters and a single cluster;
5. noise added to (1) with a signal-to-noise ratio (SNR) of 10.

This way, we can explicitly control the clusters and their separation in the data, and thus assess how well the 2D projections capture this separation. We randomly generate five trials per data set type above and show the results of a single trial in Figure 2.3. The five trials are later used for a quantitative evaluation in Section 2.4.3. For synthetic data type (5), we add Gaussian noise using the standard deviation \((\sigma)\) calculated by the definition \(\text{SNR} = 10\log_{10}\frac{P_s}{\sigma^2}\text{(dB)}\), where \(P_s\) is the power of the signal.

In Figure 2.3, the five synthetic data set types (one per row) are projected using both the sharpened and unsharpened versions of RP, LMDS, \(t\)-SNE, and UMAP. The sharpened versions are denoted by the prefix ‘S’, i.e., SRP, SLMDS, \(St\)-SNE, and SUMAP. DR methods are ordered from left to right based on how well they separate clusters. Samples are colored by the cluster labels for visual examination. Here, the LGC parameter \(\alpha\) is found em-
Figure 2.3: Comparison of DR and HD-SDR using five different types of synthetic data. DR methods are ordered from left to right based on how well clusters are separated. The samples are colored by the ground-truth labels of five different clusters for visual examination purposes. The results for SRP and SLMDS have been obtained with $\alpha = 0.04$ and for 5-SNE (perplexity = 50) and SUMAP with $\alpha = 0.01$. Note that our sharpening method enhances the separation of clusters for DR methods with less ability to separate clusters (i.e., RP and LMDS). The HD-SDR is however less effective for separating subclusters, as shown in the fourth row.
Figure 2.4: Comparison of DR and HD-SDR using four different real-world data. DR methods are ordered from left to right based on how well clusters are separated. The samples are colored by their ground-truth labels for visual examination purposes. Note that the sharpening method significantly enhances the visual separability of clusters for the first three data sets, except for SUMAP as shown in the eighth column. Additionally, St-SNE and SUMAP exhibit more oversegmentation of clusters than SRP and SLMDS.
pirically by searching the fixed range $[0, 1]$ following the explanations in Section 2.3.1. This led us to using $\alpha = 0.04$ for SRP and SLMDS and $\alpha = 0.01$ for St-SNE (perplexity $= 50$) and SUMAP.

Figure 2.3 (first four columns) shows that SRP and SLMDS significantly reduce the amount of overlap between clusters in RP and LMDS for all data sets. For t-SNE and UMAP, LGC does not improve cluster separation (Figure 2.3, last four columns). This is expected, since t-SNE and UMAP already have a good cluster separation, while RP and LMDS do not. We also see that LGC performs worst for synthetic data which consists of subclusters (4), although SRP and SLMDS show some small improvements in visual cluster separation. In the worst case (Figure 2.3(g)), SUMAP performs worse than UMAP in separating subclusters using even a small $\alpha = 0.01$.

Finally, in the fifth row of Figure 2.3, SRP and SLMDS show slightly better cluster separations of noise-added data compared with RP and LMDS, respectively. For the same data set, St-SNE and SUMAP show a similar cluster separation as their counterparts, t-SNE and UMAP.

2.4.2 Real-world data: qualitative evaluation

Our method can be applied to any type of tabular data. We next compare HD-SDR and DR using a collection of real-world data of different kinds of data traits.

Table 2.1: Trait values for real-world data sets

<table>
<thead>
<tr>
<th>Data sets</th>
<th>Size ($N$)</th>
<th>Dimensionality ($n$)</th>
<th>IDR ($\alpha$)</th>
<th>Classes ($\Gamma$)</th>
<th>Subclasses ($\Gamma_{sub}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>WiFi</td>
<td>medium (2000)</td>
<td>low (7)</td>
<td>high (0.6667)</td>
<td>medium (4)</td>
<td>-</td>
</tr>
<tr>
<td>Banknote</td>
<td>medium (1327)</td>
<td>low (4)</td>
<td>high (0.5)</td>
<td>small (2)</td>
<td>-</td>
</tr>
<tr>
<td>Olive Oil</td>
<td>small (572)</td>
<td>low (8)</td>
<td>medium (0.1250)</td>
<td>medium (3)</td>
<td>large (9)</td>
</tr>
<tr>
<td>HAD</td>
<td>large (24075)</td>
<td>medium (60)</td>
<td>low (0.0167)</td>
<td>medium (5)</td>
<td>-</td>
</tr>
</tbody>
</table>

2.4.2.1 Data sets and their traits

We characterize data sets using the traits discussed in Espadoto et al. [63]. We exclude the Type and Sparsity ratio traits since we focus here only on dense tabular data. We add the Classes trait that describes the number of clusters the data consists of.

Size $N$: The number of samples, having three ranges: small ($N \leq 1000$); medium ($1000 < N \leq 3000$); and large ($N > 3000$).

Dimensionality $n$: The number of dimensions, having three ranges: low ($n < 10$); medium ($10 \leq n < 100$); and high ($n \geq 100$).
Intrinsic dimensionality ratio (IDR) $v$: The fraction of the $n$ principal components needed to explain 95% of the data variance. We use three ranges: low ($v < 0.1$); medium ($0.1 \leq v < 0.5$); and high ($0.5 \leq v \leq 1$).

**Classes** $\Gamma$: The number of classes (ground-truth labels), having three ranges: small ($\Gamma \leq 2$); medium ($2 < \Gamma \leq 5$); and large ($\Gamma > 5$). We separately measure if the data has sub-classes and count these in $\Gamma_{sub}$. Note that we use labels as the ground-truth because there is no other ground-truth to define meaningful clusters for the concrete data sets in our chapter.

Table 2.1 shows the data sets used for evaluation and their traits.

**Banknote**: This data set has $n = 4$ features extracted using the Fast Wavelet Transform from $N = 1327$ gray scale images of banknotes [54]. Each sample (banknote) is labeled as genuine or forged, and the data set is used to train classifiers to predict this label [154]. Projections are used to assess classification: If they show clearly separated different-label clusters, then the features can very likely discriminate between the labels [169]. We know this data set is easy to classify with accuracy close to 95% [154, 166], so our projections should show well-separated clusters.

**WiFi**: This data set consists of WiFi signal strengths from various routers measured at four indoor locations [18, 54, 172]. The data set has $N = 2000$ samples with $n = 7$ dimensions and is known to have four well-separated clusters [131].

**Olive oil**: The data has $N = 572$ samples of olive oil with $n = 8$ dimensions (fatty acid concentrations), with ground-truth label denoting one of the locations in Italy from where the oil was collected. The location consists of three super-classes (North, South, and the island of Sardinia) and sub-classes (three from the North, four from the South, and two from Sardinia).

**Human Activity Data (HAD)**: This data set consists of $N = 24075$ samples of accelerometer data of a smartphone, each with $n = 60$ dimensions [55–57]. The data is used to classify five motion-related human activities (sit, stand, walk, run, and dance).

### 2.4.2.2 HD-SDR applied to real-world data sets

Figure 2.4 shows the results of HD-SDR applied to our real-world data. DR methods are ordered from left to right based on how well clusters are separated. The parameter settings for HD-SDR and $t$-SNE are displayed together with the plots in Figure 2.4. Overall, HD-SDR yield clearer visual...
cluster separations than those of the corresponding DR methods without LGC.

The effect of LGC is more prominent when the underlying DR shows poor cluster separation, such as RP (a) and LMDS (c). Compared to these, SRP (b) and SLMDS (d) show a clear improvement. For the Banknote data set, sharpening significantly reduces overlaps between the two classes (genuine and forged) for RP and LMDS, which are DR methods that show poor cluster separation. A similar difference is visible for the HAD data set (Figure 2.4, last row). For DR methods that exhibit a strong cluster separation, i.e. t-SNE and UMAP, the sharpening method improves the visual separation of clusters only by a small degree. Furthermore, St-SNE in (f) for the Banknote data set shows more visual subclusters than t-SNE (c). Note that all the projections for the Banknote data set exhibit serveral subclusters, which are also found in recent work (compare Figure 2.4 second row with Figure 12(a) in [33]). For the Olive Oil data set, the subclusters are already known (see Section 2.4.2.1), which is why we color code its projections using both class and sub-class labels (Figure 2.4, third and fourth rows, respectively). We see that sub-classes are revealed by our projections, but not as well as the classes, similar to our experiments on synthetic data (Section 2.4.1, synthetic data type (4)).

We also see an oversegmentation in HAD data projections. Oversegmentation is worse for St-SNE and SUMAP than SRP and SLMDS, as shown in (e)–(h). This can be solved by using a larger $\alpha$ or by changing $k_s$ (explained in Section 2.3.1). Further discussion of over- and under-segmentation is given in Section 2.6.

In summary, LGC significantly enhances the visual separation of clusters for the WiFi, Banknote, and Olive Oil data, except for SUMAP, where it does not greatly improve upon UMAP (Figure 2.4, 1st column). On the other hand, St-SNE and SUMAP show more oversegmentation than SRP and SLMDS, which suggests that LGC amplifies oversegmentation existing in a base projection.

### 2.4.3 Quantitative Evaluation

While there are perception-based evaluations with extensive user studies on projection methods [66], we evaluate here the projection methods quantitatively using quality metrics. As explained in Section 2.2.1, visual cluster separation is an important property of projection methods which we aim to evaluate for our proposed HD-SDR method. To do this, we need the function $H$ to quantify clusters both in the data space $D$ and projection...
space \( P(D) \). There is, however, no unique way to measure the presence, extent, or even count of the clusters in such spaces. Hence, we next use ‘weak forms’ of \( H \) given by projection quality metrics. These are functions \( Q : (D, P(D)) \rightarrow \mathbb{R}^+ \). High \( Q \)-values for a projection \( P(D) \) indicate that \( P(D) \) preserves the data structure of \( D \) – in which case \( H(P(D)) \) should be close to \( H(D) \). In particular, if LGC brings added value, we should see that \( Q(LGC(D), P(LGC(D))) \geq Q(D, P(D)) \) for various data sets \( D \) and projections \( P \).

We focus specifically on neighborhood-based metrics, which are better than distance-based metrics when assessing tasks related to finding clusters in the data [63, 141]. From these, we consider the following four metrics.

**Trustworthiness** \((Q_t)\) and **continuity** \((Q_c)\) relate to errors produced by false neighbors (points that are neighbors in \( P(D) \) but not in \( D \)) and missing neighbors (points that are neighbors in \( D \) but not in \( P(D) \)), respectively [216]. Formally put:

\[
Q_t(k) = 1 - \frac{2}{N k (2N - 3k - 1)} \sum_{i=1}^{N} \sum_{j \in U_k(i)} (r(i, j) - k),
\]

\[
Q_c(k) = 1 - \frac{2}{N k (2N - 3k - 1)} \sum_{i=1}^{N} \sum_{j \in V_k(i)} (\hat{r}(i, j) - k),
\]

where \( U_k \) and \( V_k \) are the set of false neighbors and missing \( k \)-nearest neighbors of point \( i \), respectively; \( r(i, j) \) is the rank of point \( j \) in the ordered set of neighbors of point \( i \) in \( D \); and \( \hat{r}(i, j) \) refers to the rank of point \( j \) in the ordered set of neighbors of point \( i \) in \( P(D) \). While \( Q_t \) measures the credibility of neighborhood relationships in the projection, \( Q_c \) captures the discontinuities of the projection caused by missing neighbors [216]. \( Q_t \) and \( Q_c \) lie in the range of \([0, 1]\) (worst) to 1 (best).

**Jaccard set distance** \((Q_j)\) measures the fraction of the \( k \)-nearest neighbors of a point in \( P(D) \) that are also among the \( k \)-nearest neighbors of that point in \( D \) [123, 141]. We average \( Q_j \) over all points, leading to

\[
Q_j(k) = \frac{1}{N} \sum_{i=1}^{N} \frac{|W_k^2(i) \cap W_k^n(i)|}{|W_k^2(i) \cup W_k^n(i)|},
\]

where \( W_k^2(i) \) and \( W_k^n(i) \) are the sets of the \( k \)-nearest neighbors of point \( i \) in \( P(D) \) and \( D \), respectively. This metric also lies in the range \([0, 1]\). Low
values indicate that neighbors are poorly preserved and conversely for high values.

**Neighborhood-hit** ($Q_h$) measures the proportion of $k$-nearest neighbors of a given point that fall into the same class (have the same ground-truth labels), averaged over all data points [40, 158]. It ranges between $[0, 1]$ and is defined as

$$Q_h(k) = \frac{1}{N} \sum_{i=1}^{N} \frac{|G^n_k(i)|}{k}, \quad (2.6)$$

$$G^n_k(i) = \{ j | \Gamma_j = \Gamma_i, j \in W^2_k(i) \}, \quad (2.7)$$

with $W^2_k(i)$ defined as earlier and $\Gamma_i$ being the ground-truth labels (classes) of points $i$. $Q_h$ is often used in classifier evaluation [63]. A discussion on the interpretation of $Q_h$, $Q_j$, $Q_r$ and $Q_c$ is given next in Section 2.4.3.2 when analyzing the values of these metrics for both synthetic and real-world data.

### 2.4.3.1 Evaluation of LGC

To capture whether the neighbors and their corresponding labels are preserved well by LGC, we measure $Q_h$ on the sharpened data ($Q_h(LGC(D), P(LGC(D)))$) and compare it with $Q_h$ measured on the original data ($Q_h(D, P(D))$). For clear VCS, we expect $Q_h(LGC(D), P(LGC(D))) \approx 1$ and being larger than $Q_h(D, P(D))$.

Figure 2.5 shows the average $Q_h$ over our five data set types for different $k$-values (see Section 2.4.1). We see high $Q_h$-values for (a)–(c) and (e), suggesting that LGC has achieved the desired sharpening effect. Although the data set (d) shows lower $Q_h$-values than (a)–(c) and (e), the values are still higher than the $Q_h$-value of the original, unsharpened, data. We also see that $Q_h$ increases for $\alpha = 0.04$ (yellow curves) as compared to $\alpha = 0.01$ (blue curves). This is in line with Figure 2.3 which also uses $\alpha = 0.04$. Lastly, we see how $Q_h$ decreases with $k$. The $Q_h$ decreases is significant for $k > 1000$ (not shown in the figure). This is expected, since our synthetic data clusters have 1000 points per cluster.

### 2.4.3.2 Evaluation of HD-SDR

We next evaluate $Q_h$ for LMDS, SLMDS, $t$-SNE, and $St$-SNE. We evaluate LMDS and $t$-SNE and their SDR results to show the difference between the two methods that have different degrees of cluster separation. A higher $Q_h$
2.4 Results

Figure 2.5: Comparison of neighborhood-hit ($Q_h$) for sharpened data and original data of the five different types of synthetic data used in Figure 2.3. For all synthetic data sets, $Q_h$ is always higher for the sharpened data as compared with the original data. We also note that $Q_h$ for sharpened data is higher when clusters are more separated ($\alpha = 0.04$ compared with $\alpha = 0.01$).

For the HD-SDR methods (SLMDS and St-SNE) indicates that our proposed sharpening yields better VCS than the original DR methods.

**Synthetic data:** Figure 2.6 shows $Q_h$ for our five synthetic data types for different $k$-values. For each $k$, we show the average $Q_h$ over all data sets of that type. For cases (a), (b), and (e), St-SNE, t-SNE, and SLMDS have the highest $Q_h$-values in order, while LMDS scores lowest. For data set (c), t-SNE yields a slightly higher $Q_h$ than St-SNE, but both values are close to one. This is in line with the projections in Figure 2.3 (third row) which show well-separated clusters for both t-SNE and St-SNE. For case (d) we can see that, although the visual separation is clearer for SLMDS than for LMDS, the subclusters are mixed using synthetic data type (4) in Figure 2.3, which is why $Q_h$ is lower for SLMDS than LMDS. On the other hand, St-
SNE creates a slightly better separation of subclusters than t-SNE in Figure 2.3, which is why $Q_h$ is higher for St-SNE than for t-SNE in Figure 2.6d. Furthermore, Figure 2.6(e) shows that our method is noise-resistant up to SNR = 10. Figure 2.16 (supplemental) shows the corresponding $Q_t$, $Q_c$, and $Q_j$ metrics, which have roughly the same tendency as $Q_h$ discussed above. We also show the neighborhood-hit values of the SDR results using data with varying SNR values ranging from 10 to 40 in the appendices. All in all, Figures 2.6 and 2.3 show that our sharpening yields well-separated visual clusters but is less effective for data with sub-cluster structure. Improvements aimed at sub-cluster data are discussed in Section 2.6.

**Real-world data:** Figure 2.7 shows $Q_h$, $Q_t$, $Q_c$, and $Q_j$ for different $k$-values measured on the real-world Banknote data set. Although $Q_t$, $Q_c$, and $Q_j$ yield higher values for LMDS than SLMDS, the projection results (Figure 2.4) show that LMDS achieves a worse cluster separation compared with SLMDS. Even for t-SNE, $Q_c$ and $Q_j$ yield higher values compared with St-SNE, but St-SNE exhibits a better cluster separation than t-SNE (Figure 2.4, second row).

Figure 2.8 shows $Q_h$ measured for our four real-world data sets for different $k$-values. For the Olive oil data set, we use its super-class labels to compute $Q_h$. For this data set, it is important to limit $k$ since its classes are quite unbalanced: 323 (blue), 98 (orange), and 151 (yellow) points, respectively. Hence, we limit $k < 300$ for this data set. Overall, Figure 2.4 shows that $Q_h$ decreases with $k$ for all studied methods. This is expected and in line with Figure 2.4: As $k$ increases, $Q_h$ considers larger neighborhoods including points outside any visible (sub)cluster. The values of $Q_h$ for SLMDS are higher than for LMDS for all four data sets. These results reflect that the clusters are separated better in the sharpened projections than the original projections shown in Figure 2.4. For t-SNE and St-SNE, the results vary among different data sets. For the Banknote data, $Q_h$ for St-SNE is larger than for t-SNE, which is also reflected in the projections shown in Figure 2.4. However, for the other three data sets, t-SNE shows higher $Q_h$-values than St-SNE. For the WiFi data, the results can be explained by the corresponding projections in Figure 2.4, where t-SNE mixes points from different clusters. For the Olive oil data, $Q_h$ considers neighbors outside a cluster with the same class labels for each divided cluster shown in Figure 2.4(c)–(f). For the HAD data set, the $Q_h$-value for t-SNE is slightly higher than that for St-SNE when $k < 400$, but the situation reverses when $k \geq 400$. This can be explained by the significant oversegmentation exhibited in the projections (last row of Figure 2.4(e)–(f)).
Figure 2.23 (supplementary material) complements Figure 2.4 and the discussion above by showing the $Q_c$, $Q_t$, $Q_j$, and $Q_h$ for the WiFi, Olive Oil, and HAD real-world data sets, and confirms that HD-SDR, while yielding better visual cluster separation than the DR baseline, scores slightly lower quality metrics.

Previous work using $Q_t$, $Q_c$, and $Q_j$ showed inconsistent results for different values of $k$, which resulted in interpretation difficulties [141, 216]. This is also visible in Figure 8 (appendices): $Q_j$ increases with $k$, which is logical – in the limit, when $k$ equals the sample count $N$, the neighborhood becomes the entire data set, so $Q_j = 1$. $Q_t$ and $Q_c$ exhibit even more complex and non-monotonic behavior, often exhibiting local maxima for certain $k$-values.

In contrast to the above, $Q_h$ decreases monotonically with $k$: for small $k$-values, $Q_h$ has quite high values. This is expected for data sets that we know that are well separated into clusters having different labels (like ours). For such data sets, as long as $k$ is under the size of a cluster, $Q_h$ will be very high and nearly constant, since a neighborhood will tend to ‘pick’ same-label points from a single cluster. When $k$ exceeds the average number of samples having the same label (the average cluster size for data sets that are well separated into clusters), a neighborhood will inevitably contain more labels, resulting in $Q_h < 1$. In the limit, for a balanced data set of $C$ classes, $Q_h = 1/C$ when $k = N$. For all its limitations, $Q_h$ also has some advantages: $Q_h$ removes the dependency on the distance in the original data space. This is important as distances in that space are subject to the well-known curse of dimensionality. As such, whenever Euclidean distances are used and the dimensionality increases, neighborhoods become meaningless or very unstable (the ratio of the closest and farthest points tends to one) [6]. As $Q_h$ does not explicitly check where neighborhoods in nD and 2D are the same, but only the homogeneity of labels in a 2D neighborhood – assuming again that these are homogeneous in the data space for a data set well-separated into clusters having different labels – $Q_h$ is less sensitive to the above dimensionality issues.

Summarizing the above, we argue that although HD-SDR produces, in general, lower quality metrics than some of the baseline DR methods, (a) these quality metrics do not directly capture the visual cluster separation we aim to optimize for; and (b) this separation actually is shown to increase in the actual HD-SDR projections as compared to the baseline projections (see Figure 2.4). However, visual cluster separation does not increase when oversegmentation occurs. The oversegmentation issues are discussed next in Section 2.6. Based on the qualitative and quantitative studies above, we
note that conducting user studies on SDR and comparing them with the quantitative results can be interesting for future work.

2.5 Application to Astronomical Data

As a specific use case to show that VCS improves by our HD-SDR method, we aim to separate 10K previously unclassified stars into clusters that may represent distinct physical groups within our own Milky Way galaxy. This is a common goal in astronomy: a large data set of unlabeled objects – up to a few $10^9$ stars in current catalogues – needs to be classified into separate (physically meaningful) clusters, so that labels representing physical groupings can be applied to individual objects. Importantly, this process has to involve the user in deciding which similar objects (in the same cluster) can be assigned to the same label. As such, the goal here is to perform manual labeling, with labels having user-assigned semantics, and not automatic labeling of the type that clustering algorithms would support. Doing this manual labeling object-by-object is clearly impossible with such large data sets. Previous attempts using standard DR methods have not been completely successful [7]. We show here that the VCS of HD-SDR meets this goal.

We first aim to reproduce the results shown in a recent study of dissecting stellar abundance space with t-SNE [7] but using two more-recent data sets. First, we consider the second release of data from the Gaia satellite (known as Gaia DR2, publicly available since 2018) which contains observations of roughly 1.69 billion objects (stars, galaxies, quasars, and Solar System objects) [74, 75]. Secondly, we consider the second data release of the GALactic Archaeology with HERMES survey (GALAH DR2), also from 2018, a large-scale spectroscopic stellar survey including the properties of 342,682 stars in that release [25].
Figure 2.6: Comparison of neighborhood-hit ($Q_h$) for DR and HD-SDR of the five different types of synthetic data used for Figure 2.3. Note that St-SNE, t-SNE, and SLMDS yield high $Q_h$-values near one for (a)–(c) and (e), which suggests that the corresponding labels of the $k$-size neighborhoods are well-preserved for HD-SDR. However, HD-SDR for sub-clustered data produces lower $Q_h$ compared with DR, as shown in (d), and this can be seen visually in Figure 2.3(a)–(d). More results including $Q_r$, $Q_c$, and $Q_j$ for the five synthetic data sets can be found in the appendices.
Figure 2.7: Results of four neighborhood-based quality metrics for Banknote data: Neighborhood-hit ($Q_h$), Trustworthiness ($Q_t$), Continuity ($Q_c$), and Jaccard set distance ($Q_j$). Note that $Q_h$ is consistent with results from Figure 2.3 and best represents the visual cluster separation, whereas $Q_t$, $Q_c$, and $Q_j$ suggest the opposite. Note that $Q_t$, $Q_c$, and $Q_j$ do not consider class label information. More results including $Q_t$, $Q_c$, and $Q_j$ for the five synthetic data sets can be found in the appendices.
Figure 2.8: The neighborhood-hit ($Q_h$) metric for DR and HD-SDR using labeled real-world data sets with different values of $k$. We note that $Q_h$ is lower for St-SNE than for t-SNE for most values of $k$ in (b)–(d). However for LMDS, which produces a weaker separation of clusters than t-SNE, SLMDS produces a higher value of $Q_h$ compared with LMDS for all data sets.
Figure 2.9: We compare DR and HD-SDR (RP, LMDS, t-SNE, and UMAP) for an unlabeled astronomical data set with no ground-truth labels: GALAH DR2 (n = 10K with N = 10K). The projections are color-coded by one of the input values, [Fe/H], so that astronomers can further analyze the data [7]. The learning rate parameter is set to 0.18. Note that in all cases, HD-SDR shows a clearer separation of clusters compared with DR, and SLMDS, St-SNE, and SUMAP exhibit four major clusters with similar distributions of colors within each cluster, while SRP shows three major clusters with one of them having subclusters.
Figure 2.10: (a) SLMDS projection of the GALAH DR2 sample with clusters visually labeled by one of us. The labeled clusters in SLMDS help domains experts to further analyze the data as follows: (b) Tinsley diagram [200] shows the abundance of magnesium as a function of the iron abundance, used to interpret the origin and location of Milky Way stars. This diagram suggests to our domain-expert that stars in class 2 belong to the Milky Way’s "thin disk", while those in classes 1 and 3 appear to belong to the Milky Way’s "metal-rich thick disk" and "metal-poor thick disk", respectively; stars in class 4 appear to belong to the Milky Way’s "stellar halo". (c) This plot shows the barium abundance of the stars as a function of their iron abundance, a tracer of a different nucleosynthetic process (the slow-neutron-capture, "s-", process). Stars in class 4 have strongly different barium abundances for their low iron content. These may be "metal-poor barium stars", which arise from binary star interactions. The same analysis for St-SNE and SUMAP are shown in the supplemental results; the clusters in SRP Figure 2.9(b) are not easily separated, thus excluded.
The data set we use cross-matches GALAH DR2 with Gaia DR2 using the Gaia DR2 ID of each star as the matching key. This cross-match yields 6D phase-space coordinates (3D stellar positions and 3D velocities). To obtain credible data, samples that meet the following criteria are excluded: one or more of positions $x$, $y$, and $z$ exceeding 25K parsec (where distance information becomes seriously unreliable), samples with missing values of any attribute (stellar abundance measurements and errors and 6D phase-space coordinates), and any stellar abundance measurements that are deemed by the GALAH team as unreliable [25]. From the 76270 credible samples, we randomly select $N = 10K$ samples to project using RP, LMDS, $t$-SNE, and UMAP, and their sharpened versions. We use the same $n = 10$ attributes, i.e., the stellar abundances $[\text{Fe/H}]$, $[\text{Mg/Fe}]$, $[\text{Al/Fe}]$, $[\text{Si/Fe}]$, $[\text{Ca/Fe}]$, $[\text{Ti/Fe}]$, $[\text{Cu/Fe}]$, $[\text{Zn/Fe}]$, $[\text{Y/Fe}]$, and $[\text{Ba/Fe}]$, as in a similar data set visualized by $t$-SNE [7], for comparison purposes.

We present the resulting projections using HD-SDR and DR in Figure 2.9(a)–(h). The projections are color-coded by one of the input values, $[\text{Fe/H}]$, so that astronomers can further analyze the data as in [7]; as a first pass, one of us examined the projections to evaluate the impact of the sharpening on understanding the astrophysical importance of the resulting distributions. Several insights follow. First, (without considering the color-coding) we can see that all HD-SDR results have better cluster separation compared with the DR results, and that SLMDS $St$-SNE, and SUMAP exhibit four major clusters with similar distributions of colors within each cluster. We note that our $t$-SNE projection is very similar to that of Anders et al. (compare Figure 2.9(e) with Figure 2 in [7]1). We also see that SRP exhibits three major clusters, with one of them having subclusters, as shown in panel (b). Overall, HD-SDR offers a much better cluster separation, even more so than $t$-SNE, which is used to analyze a similar data set by Anders et al. We use one of the attributes, i.e., $[\text{Fe/H}]$ to color-code the projection. By doing so, the clusters in HD-SDR projections are more easily explained by this attribute than the structures apparent in the DR projections.

Out of the four HD-SDR projections shown in Figure 2.9, we further analyze the projection using SLMDS of this data set and 2D scatter plots of three abundances (Tinsley diagram [200]) with $[\text{Fe/H}]$, $[\text{Mg/Fe}]$, and $[\text{Ba/Fe}]$ in Figure 2.10. Note that the same analyses have been shown for $St$-SNE and SUMAP in the appendices (SRP has been excluded because the subclusters are inseparable). Due to the clear separation of four clusters us-

---

1 The figures are similar but not identical because Anders et al. used a different set of stellar abundances, the HARPS-GTO sample, based on stars taken for exoplanet identification, with ten times fewer stars but much higher quality; see [7] for more details.
ing SLMDS, a domain-expert is able to manually assign four different class labels to the clusters, as shown in Figure 2.10(a). Next, we color-code the Tinsley diagram by the newly acquired labels (Figure 2.10(b)–(c)). Without the labels, domain-experts would have to manually visit each point to further analyze each star. Using the color-coded points, domain experts are able to quickly infer the location and origin of each group of stars in the Milky Way.

Upon seeing these results, one of us and two other domain experts in astronomy\(^2\) noted that HD-SDR has a clear and higher potential in helping them to infer new results about the data at hand, compared with DR, in which clusters are less separable and are not strongly correlated with specific attributes. For example, in this case, the four classes could be identified in other tracers of the Milky Way’s history, like its dynamical structure [85].

2.6 DISCUSSION

In this section, we discuss several aspects of HD-SDR.

2.6.1 Scalability

2.6.1.1 Speed

Figure 2.11(a) shows the average wall-clock timings of LGC over 10 trials of randomly generated Gaussian data with five clusters for dimensionality \(n \in \{5, 10, 15, 20\}\) and sample counts \(N \in \{10K, 20K, 30K, 40K, 50K\}\). For this plot, we used \(\alpha = 0.1\). LGC is mainly affected by \(n\), due to the nearest neighbor search, which is of order \(O(N n \log n)\). The overall time complexity of LGC is \(O(n t N \log N)\). LGC takes over five minutes to compute for data with 20D and 50K samples (Figure 2.11(a)). Applying LGC to data with hundreds of dimensions may be impractical for end-users. A possible solution is outlined in Section 2.6.1.2.

Figure 2.11(b) shows how speed depends on the \(\alpha\) parameter using a data set with \(N = 10K\) samples and \(n = 20\). As \(\alpha\) increases, the wall-clock time gradually increases. Profiling shows that this is due to the time needed to build \(kd\)-trees for kNN. About 95% of the wall-clock time of LGC is due to the nearest neighbor (NN) search needed to compute \(\rho\) (Eq. 2.1). Performing the kNN search for every iteration (instead of just once as

---

\(^2\) Dr. Sarah Martell, Project Scientist of the GALAH Survey and a co-author of [204]; and Dr. Sara Lucatello, an expert on tracing the formation of the Milky Way through the abundances of its stars.

---
in the original gradient clustering algorithm [73]) increases the time complexity proportional to the number of iterations \(t = 10\) because \(kd\)-trees are constructed for every iteration. Easy speed-ups include replacing the current NN method [19] by approximated and parallelized versions such as FLANN [148, 149]. Multiple random projection trees (MRPT) [101, 102] reduces expensive distance evaluations, thereby achieving higher speed than ANN and FLANN. However, MRPT has several issues: an insufficient number of requested nearest neighbors are returned; single-precision floating point is used; retrieving distances between neighbor points is not easily supported; and inaccurate search – in the worst case, points that are far away from the query point are returned as nearest neighbors. Hence, MRPT is currently unfit for an accurate computation of nearest neighbors.

Figure 2.11: (a) Wall-clock timing of LGC, for different number of observations along the \(x\)-axis and varying dimensions up to 20D. (b) Wall-clock time measurement of LGC on a 10K (20D) data set, using different learning rates \(\alpha\). (c) Wall-clock time measurements of RP, LMDS, and \(t\)-SNE applied to \(nD\) LGC data. We note that the wall-clock time of LGC increases with increasing \(\alpha\) and depends heavily on the number of dimensions. Moreover, RP and LMDS (landmark ratio=0.05) take less than a minute to run, while \(t\)-SNE takes longer and the speed heavily depends on the number of samples. More results of speed experiments using different numbers of dimensions in RP, LMDS, and \(t\)-SNE and landmark ratios in LMDS are found in the appendices.
Separately, the sample shift (Eq. 2.2) can be trivially parallelized on the CPU or GPU for further acceleration, leading to speed-ups of two orders of magnitude, as shown by related work [212].

Figure 2.11(c) shows the wall-clock timings for LMDS, t-SNE, and RP on nD LGC data. When comparing the time measurements of LGC against standalone DR methods, all DR methods take less time to run compared with LGC for 50K observations, where t-SNE takes the longest. Note that LMDS, t-SNE, and RP are all from the same Tapkee library (UMAP is not and therefore has been excluded from the experiments). More timing results using different numbers of dimensions in RP, LMDS, and t-SNE, and landmark ratios in LMDS, can be found in the appendices.

Figure 2.12: Proportion of total variance explained by each component when using PCA on Human Activity Recognition (HAR) data. The HAR data set (561 dimensions) can be reduced to 10 dimensions while keeping 80% of the variance.

2.6.1.2 High-dimensional data

Section 2.6.1.1 states that applying LGC to data with hundreds of dimensions may be impractical due to speed issues. A solution is to first reduce the dimensionality with a simple and fast DR (i.e., PCA) and then apply HD-SDR. Figures 2.12–2.13 illustrate this. Here, Human Activity Recognition (HAR) data [54, 171] with \( n = 561 \) and \( N = 7352 \) for six basic activities are used: three static postures (standing, sitting, and lying); and three dynamic activities (walking, walking downstairs, and walking upstairs). First, we reduce \( n \) to 10 using PCA (keeping 80% of total variance, see Figure 2.12). Then, we use HD-SDR on this 10-dimensional data, with \( \alpha = 0.2 \). The obtained projections using SRP, SLMDS, and St-SNE (Figure 2.13 bottom row) all exhibit improvements over their original counterparts (Figure 2.13 top row). In particular, we can easily see that cluster (1) of SLMDS
Figure 2.13: Comparison of DR and HD-SDR on HAR data with dimensions pre-selected using PCA as in Figure 2.12. Note that the separation of clusters is slightly improved for HD-SDR as compared with DR (although subclusters are visually less separable). We note that nearly all samples in cluster (1) are from the *lying* movement, cluster (2) mostly includes samples from static postures (standing and sitting), and cluster (3) mostly includes samples from the three dynamic activities. (Figure 2.13(d)) is clearly separated from the others. Nearly all samples in this cluster are from the *lying* movement class. Although points with different class labels are mixed in clusters (2) and (3), most points in cluster (2) are from other static postures (standing and sitting), while most points in cluster (3) are from the three dynamic activities. Further note that separating sub-clusters still remains an issue (see Section 2.6.5 next).

2.6.2 Data distortion

Our method addresses the cluster separation problem by shifting points in the original space, which may cause data distortions. Section 2.4.3.1 shows that the LGC step actually *improves* neighborhood preservation with respect to the ground-truth labels in the original data, which is the main aspect we aim to capture. Any DR method performs, by definition, non-trivial amounts of data distortion when mapping from the high-dimensional space to 2D if the data is not originally already located on a smooth 2D manifold. Hence, no DR method can faithfully capture *all* aspects of any data set [63]. Users will be always exposed to certain types of data distortions and/or data aspects that are not captured in the 2D projection. This is especially true for local and nonlinear projection...
techniques, e.g., t-SNE and UMAP. Whether such distortions occur in the preprocessing step like our LGC, or in the projection, as for all other DR methods, does not remove the fact that such unavoidable changes occur. Hence, the fact that LGC changes the data does not imply that our technique is less trustworthy than any other DR technique, which change the data during the projection itself.

2.6.3 Relation to clustering

Our technique is aimed at supporting the exploratory analysis using DR methods, and thus has a close relation to data clustering. For example, Chen et al. [33] use mean shift, which is closely related to LGC, to create DR projections. They construct an explicit clustering of a data set $D$, $\bigcup_i C_i = D$, by mean shift, after which they project the cluster centers $c(C_i)$ to 2D and use these landmarks $P(c(C_i))$ to perform local MDS projections $P(C_i)$. Many other local DR methods work similarly [105, 153]. In contrast, we do not require an explicit clustering of the data to partition $D$ to project it piece-wise; rather, we use LGC as a preconditioning technique to improve a subsequent global projection of $D$. Moreover, our LGC updates the KDE gradient $\nabla \rho$ at every advection iteration (Eq. 2.2). This is different from classical mean shift [73] as used in [33], where $\nabla \rho$ is computed from the initial density estimate and then used unchanged during the update (Eq. 2.1). Updating $\nabla \rho$ leads to faster cluster separation, especially for noisy data [99, 212].

Separately, as mentioned already in Section 2.2.1, HD-SDR cannot, and should not, create projections with high VCS for all data sets. This would be misleading as it would suggest to the user that such structures exist in otherwise unstructured data. Hence, for data sets that lack such structure, one should expect HD-SDR to create projections with low VCS.

2.6.4 Preservation of outliers

Figure 2.14 compares DR and HD-SDR using data sets with outliers marked as crosses. To investigate the effect of SDR on outliers, we create five 20-dimensional hyperspheres, with 5K points randomly generated and uniformly distributed within radii $R \in \{1, 2, 3, 4, 5\}$. We then add 10 outliers distributed on the largest hypersphere surface ($R = 5$) to each data set. Unlike RP and LMDS, t-SNE and UMAP and their sharpened versions do not preserve outliers well. This is expected because t-SNE and UMAP are neighborhood-preservation DR methods, whereas RP and LMDS are distance-preservation methods (see Section 2.2.2). As Figure 2.14 (rows 1
Figure 2.14: Comparison of DR and HD-SDR using 10 outliers (red crosses) and 5K random points uniformly-distributed in a 20-dimensional hypersphere centered at the origin and with different radii $\mathcal{R} \in \{1, 2, 3, 4, 5\}$. The distance between an outlier and the origin is always five. Note that t-SNE and UMAP and their sharpened versions fail to preserve the outliers even when the distance between the outliers and the cluster is the largest (row 1). Further note that SRP and SLMDS preserve similar or larger numbers of outliers compared with RP and LMDS, respectively.
and 2) show, the farther the outliers are from the hypersphere surface, the more outliers are preserved by SRP and SLMDS than by RP and LMDS. As the hypersphere radius increases, both DR and SDR do not preserve outliers well, which is expected because the distance between the outliers and the data in the sphere decreases (rows 4 and 5). Overall, SDR preserves a similar number of outliers compared with DR, but a careful selection of parameters is needed for real-world data sets where the number of clusters and outliers vary. Section 2.6.5.3 further discusses parameter setting.

Figure 2.15: Results of DR and SDR with varying $k_s$-values for sharpening. The synthetic data set consists of 10K randomly generated samples in 20D. Note that St-SNE shows oversegmented clusters using $k_s = 10$, while (b) shows a single cluster as in (c) when using $k_s = 50$.

2.6.5 Limitations and future work

2.6.5.1 Undersegmentation and oversegmentation

In some fields, including subfields of astronomy, domain experts are interested in finding substructures [52], which relates to the undersegmentation issue. As Figure 2.3 shows, our method is less effective in capturing tightly connected subclusters. This is closely related to how far apart two clusters should be to be separated by the projection rather than rendered as a single cluster. This issue should be further explored both theoretically and empirically.

While the subcluster issue can also be seen as undersegmentation, HD-SDR further augments oversegmentation when using an oversegmentation-prone DR or a DR with strong cluster separation like UMAP or $t$-SNE (see Figure 2.4, fourth row). Oversegmentation is a known aspect of mean shift methods [99]. While there are many data-specific heuristics to set the scale of clusters, there is no generic way to avoid under- or oversegmentation. Hence, oversegmentation is not particular
to our method, as other projections with strong clustering also exhibit this problem.

Besides controlling the parameters of our method, one way of solving under- and oversegmentation would be to use an adaptive learning rate $\alpha$ to capture different detail levels in a cluster. Adaptive learning rates that consider the distribution of distances between neighboring points may allow sharpening to be more adaptive to different distributions of clusters. However, this approach is also risky because of errors in the density estimator propagating due to inconsistent point-shifts after each iteration. Studying adaptive learning rates is hence left to future work.

2.6.5.2 Noise-free data

When using SDR, it is assumed that there are no errors in the measurements or data processing stage. In real-world data sets, the data may intrinsically have uncertainties such as measurement errors or errors due to data processing. For example, with astronomical data, there can be measurement errors or missing values. Even though we show that our method is noise-resistant up to SNR = 10 (synthetic data type (5)), real-world data may contain non-Gaussian noise, which is why it is crucial for the user to consider these uncertainties for a more accurate analysis of the data at hand.

Furthermore, random noise can be sharpened producing oversegmentation when using our method. However, it is possible to negate the effects of sharpening small dense areas of noise by using a large enough value of $k_s$ in SDR. We demonstrate an extreme case in Figure 2.15 by comparing two different $k_s$-values in SDR using a single cluster with 10K randomly generated samples in 20D. We see that t-SNE and St-SNE using $k_s = 50$ show a single cluster, whereas St-SNE using $k_s = 10$ shows highly oversegmented clusters. The same observation can also be made using RP, LMDS, and UMAP (see appendices). Hence, it is important for the user to select a large enough $k_s$-value (i.e., $k_s \geq 50$) to prevent sharpening noise that may cause oversegmentation.

2.6.5.3 Parameter setting

Our parameters $k_s$ (how localized a shift is) and $\alpha$ (shift speed) are interconnected, see Section 2.3.1. This also holds for other kernel density estimation (KDE) methods [99]. While both $k_s$ and $\alpha$ affect the segmentation degree, if $k_s$ is large enough, then larger $k_s$-values may not significantly affect segmentation without choosing an suitable $\alpha$-value, see Figure 2.1
Setting a large enough \( k_s \)-value is also crucial to avoid sharpening noise as explained in Section 2.6.5.2, which is why we use \( k_s \geq 50 \).

Figure 2.24 (supplemental material) completes the insights from Figures 2.1–2.2 by showing the results of our method for the WiFi data set for multiple values of \( \alpha \) and \( k_s \), using \( t = 10 \) iterations, as discussed in Section 2.3.1. As explained in Section 2.4.2.1 and also visible in Figure 2.4, we know that this data set consists of four clusters. We see that HD-SDR produces four compact and well-separated clusters for the parameter combination \( \alpha = 0.15 \) and \( k_s = 100 \), which are in line with our recommended presets \((\alpha = 0.15, k_s \geq 50)\). The over- and undersegmentation produced by other parameter values follow the same trend for this real-world data set as for the synthetic data in Figure 2.1.

A too-large number of LGC iterations \((t\text{-value})\) can lead to over-shooting the local cluster centers during the gradient ascent and also longer computation. While Hurter et al. [99] decrease the advection speed \( \alpha \) over iterations to solve the overshooting problem, they aim to have all points in a visual cluster converge to a single location. This is clearly undesired for projections, so we use a constant advection speed. Finally, note that we stop LGC based on a fixed \( t \)-value. A better stop criterion would be to use a quality metric, e.g., neighborhood-hit \((Q_h)\). Exploring this (and how to do it efficiently) would be interesting for future work.

2.6.5.4 Post-processing in dimensionality reduction

Technically, our sharpening approach can also be applied to 2D instead of \( n \)-D data. However, this is problematic: We know in advance that the data distances are ‘uniform’ in \( n \)-D, so sharpening with a certain distance or speed will work uniformly for all data points [44, 73]. In contrast, in a 2D projection, distances are generally non-uniform – one 2D pixel may correspond to small or large data distances depending on where it is in the projection. Hence, we cannot sharpen 2D points with the same speed, and determining the speed to use per point is a major difficulty, as this would require knowing the inverse projection \( P^{-1} \). Another problem with sharpening after DR is that a poor projection (in terms of VCS) cannot possibly be ‘fixed’ by sharpening; sharpening will make it worse, as it will amplify its poor VCS. In contrast, sharpening before DR can produce clear VCS even for DR methods that originally exhibit poor VCS as shown in Figures 2.4 and 2.9.

Other post-processing methods for 2D projections exist, e.g., applying ‘clustering’ after DR or SDR to automatically label individual clusters in
a projection. This approach is currently out of our scope and will be explored in future works.

### 2.6.5.5 LGC for t-SNE and UMAP

Figures 2.3–2.4 show that some DR methods produce a clearer cluster separation even without LGC, in particular methods that already exhibit strong cluster separation and/or show oversegmentation, e.g., t-SNE and UMAP. UMAP yields a better VCS than other SDR results including SUMAP in most of the examples shown in this chapter except for Figure 2.9. However, due to the limitations of UMAP mentioned in Section 2.2 (too dense clusters and difficult parameter setting due to its stochastic nature), other DR methods with lower VCS may be preferred over UMAP, which is why we explore the sharpening effect on additional DR methods. This is also why we explicitly compare SDR with the original DR methods rather than with specific DR methods like UMAP or t-SNE.

### 2.6.5.6 Selection of baseline DR method

Figures 2.3–2.4 show that some DR methods yield a clearer cluster separation when aided by LGC than other methods. Besides these examples, other DR methods benefit from being combined with LGC. To study this, we applied LGC to all 44 DR methods in the benchmark of Espadoto et al. [63] for the WiFi data set using $\alpha = 0.15$ (see appendices). These results show that our method works with any DR method that we are aware of. While we use the same $\alpha = 0.15$ for all experiments, some combinations of LGC with certain DR methods produce better results with different $\alpha$-values. In particular, ISO and LMVU produce some separation of clusters, while the sharpened versions of them do not. Using a smaller $\alpha$ for S-ISO and S-LMVU results in clearer cluster separation (see appendices). This suggests that using different $\alpha$-values can solve issues with poorer cluster separation in HD-SDR than in DR. While out of scope of this chapter, exploring why certain DR methods are more suitable for HD-SDR based on a user-centric approach [67], and which SDR is effective for different data types, are important topics to study next.

### 2.7 Conclusion

We have presented a new method for dimensionality reduction (DR) that creates visually separated sample clusters targeted for user-guided labeling to explore and analyze the data using DR. Key to our method is a preconditioning step that “sharpens” the sample density in the data space.
prior to using any DR technique. We tested our method using both synthetic and real-world data from five different application domains, using RP, LMDS, t-SNE, and UMAP as DR methods. HD-SDR yields better visual cluster separation in the projection than the original DR methods that exhibit weak cluster separation. In terms of practical usefulness, astronomy experts see clear added-values in the results produced by HD-SDR on their data as compared with t-SNE, which was used in previous studies. This is the first time to our knowledge that a mean shift-based sharpening method is used without any prior knowledge of cluster modes to enhance the separability of clusters. We suggest that the LGC preconditioning step shows how LGC can lead to techniques that bridge the gap between DR methods with different abilities to separate clusters.

ACKNOWLEDGMENTS

This work is supported by the DSSC Doctoral Training Programme co-funded by the Marie Sklodowska-Curie COFUND project (DSSC 754315). The GALAH survey is based on observations made at the Australian Astronomical Observatory, under programmes A/2013B/13, A/2014A/25, A/2015A/19, A/2017A/18. We acknowledge the traditional owners of the land on which the AAT stands, the Gamilaraay people, and pay our respects to elders past and present. This work has made use of data from the European Space Agency (ESA) mission Gaia (https://www.cosmos.esa.int/gaia), processed by the Gaia Data Processing and Analysis Consortium (DPAC, https://www.cosmos.esa.int/web/gaia/dpac/consortium). Funding for the DPAC has been provided by national institutions, in particular the institutions participating in the Gaia Multilateral Agreement.

APPENDICES

We present figures to show the full results of using all parameter combinations, applying various DR methods as the baseline method of SDR, and quality metrics as follows.
Figure 2.16: Results of the Neighborhood hit ($Q_h$), Trustworthiness ($Q_t$), Continuity ($Q_c$), and Jaccard set distance ($Q_j$) quality metrics for five synthetic data sets introduced in Section 4.1.
Figure 2.17: Comparison of DR and SDR (α=0.15) applied to WiFi data using different DR techniques. The DR methods are provided by the implementations in the benchmark by Espadoto et al. [10]. These results show that any DR method can be used for SDR. Note that we use the same α = 0.15 for all experiments, but some combinations of LGC with certain DR methods produce better results with different α values as shown in supplemental material 2.18.
Figure 2.18: SDR and DR applied to the WiFi data set for ISO and LMVU DR methods from the benchmark of Espadoto et al. [10] using different values of $\alpha$. Note that ISO and LMVU produce some separation of clusters, while the sharpened versions of them do not. Using a smaller $\alpha$ for S-ISO and S-LMVU results in clearer cluster separation. These results suggest that exploring different values of $\alpha$ can solve issues with poorer cluster separation in SDR than in DR.
Figure 2.19: Analysis of GALAH DR2 sample using St-SNE and SUMAP projections (top row), where clusters are visually labeled by one of us (SCT) and the Tinsley diagrams [77] for the GALAH DR2 sample (bottom row), showing the abundance of magnesium and Barium as a function of the iron abundance, used to interpret the origin and location of Milky Way stars. Note that the clusters in SRP are not easily separated and are therefore excluded.
Figure 2.20: Extended results of the Neighborhood hit ($Q_h$) for data sets with different $k$-values and signal-to-noise ratio (SNR) values ranging from 10 to 40, where Gaussian noise has been added. The graphs in the right column show the cluttered area ($Q_h \approx 1$) for the graphs in the left column. Overall, note that the $Q_h$-values are higher for higher SNR values. Further note that the $Q_h$-values of SDR are always higher than those of the corresponding DR.
Figure 2.21: Results of DR and SDR with varying $k_s$-values for sharpening. The synthetic data set consists of 10K randomly generated observations in 20D. Note that (a), (d), and (g) show oversegmented clusters using $k_s = 10$, while (b), (e), and (h) show a single cluster using $k_s = 50$ for LGC.
Figure 2.22: Extended results of speed experiments in logarithmic scale using LMDS, t-SNE, and RP in Figure 11(c). (a) Wall-clock time of LMDS for different numbers of landmark ratios along the x-axis and varying dimensions up to 20D. (b)–(d) Wall-clock time of LMDS, t-SNE, and RP for data sets with different numbers of dimensions (5D, 10D, 15D, and 20D) and observations ranging from 10K to 50K. The wall-clock timings are an average of 10 trials of randomly generated Gaussian data. We note that a ninth-order polynomial can be fitted to LMDS in (a), fourth-order polynomials to both LMDS (b) and t-SNE (c), and a quadratic for RP (d) (norm of residuals < \( e^{-3} \)). Furthermore, the wall-clock time for LMDS heavily depends on the landmark ratio, and LMDS and t-SNE have little difference in wall-clock timings between varying dimensions.
Figure 2.23: Extended results of the quality metrics calculated on real-world data sets. The quality metrics include Neighborhood hit ($Q_h$), Trustworthiness ($Q_t$), Continuity ($Q_c$), and Jaccard set distance ($Q_j$).
Figure 2.24: SLMDS results on the WiFi data set using varying parameters of LGC. Each row shows the SLMDS results using different learning rate ($\alpha$) values and the columns with varying $k_s$-values, respectively.
2.7 CONCLUSION
Sharpened dimensionality reduction (SDR), which belongs to the class of multidimensional projection techniques, has recently been introduced to tackle the challenges in the exploratory and visual analysis of high-dimensional data. SDR has been applied to various real-world data sets, such as human activity sensory data and astronomical data sets. However, manually labeling the samples from the generated projection are expensive. To address this problem, we propose here to use clustering methods such as k-means, Hierarchical Clustering, Density-Based Spatial Clustering of Applications with Noise (DBSCAN), and Spectral Clustering to easily label the 2D projections of high-dimensional data. We test our pipeline of SDR and the clustering methods on a range of synthetic and real-world data sets, including two different public human activity data sets extracted from smartphone accelerometer or gyroscope recordings of various movements. We apply clustering to assess the visual cluster separation of SDR, both qualitatively and quantitatively. We conclude that clustering SDR results yields better labeling results than clustering plain DR, and that k-means is the recommended clustering method for SDR in terms of clustering accuracy, ease-of-use, and computational scalability.

3.1 Introduction

Dimensionality reduction (DR) is a commonly used approach to visualize and explore multidimensional data. Recently, Sharpened DR (SDR), a preprocessing method that enhances the cluster separation of DR, was proposed [112]. The preconditioning step sharpens the sample density using Gradient Clustering (GC) in \( n \)-dimensional space, where the sharpening effect is visible after DR. This separability of clusters allows end-users to explore the multidimensional data more easily.
Although SDR is capable of preserving the underlying cluster structures in $n$-dimensional space and representing them in 2D, labeling these structures in the resulting projections is time consuming. Moreover, identifying the visual clusters can be dependent on subjective perception. To address this problem, we focus here on suggesting a set of clustering methods and validation metrics that will automatically label the clusters from SDR and assess their quality both qualitatively and quantitatively. In this pipeline consisting of SDR and clustering, the labeling process remains as a black box to end-users, providing a convenient integration with further steps of data exploration or analysis using SDR.

The paper is structured as follows. Sec. 3.2 explains the proposed pipeline. Sec. 3.3 shows the experimental results and Sec. 3.4 discusses the results. Sec. 3.5 concludes the paper.

3.2 Method

Let $D = \{x_1, \ldots, x_N\}$ be a set of $N$ $n$-dimensional observations, where $x_i = [x_i^1 \ x_i^2 \ \cdots \ x_i^n] \in \mathbb{R}^n$ and $x_i^j$ is the $i^{th}$ observation of the $j^{th}$ ($1 \leq j \leq n$) dimension. The sharpening method can be seen as a function $S : \mathbb{R}^n \rightarrow \mathbb{R}^n$, whereas DR is a function $P : \mathbb{R}^n \rightarrow \mathbb{R}^{n'}$, where $n' << n$, where commonly $n' = 2$. We set $P$ to Landmark Multidimensional Scaling (LMDS) because it shows a clear separation of clusters for human activity and motion data [112, 195]. We aim here to replicate the results from Kim et al. to compare the clustering results between DR and SDR, which has not been shown before [112]. We also define $L$ as the list of labels for all points acquired from clustering and $G$ as the prior, which is the list of ground-truth labels of a data set. Note that we use the terms ‘points’, ‘observations’, and ‘samples’ interchangeably.

3.2.1 Clustering methods

Clustering methods are chosen to cover different types of methods such as partitional, hierarchical, density-based, and graph-based. A popular partitional algorithm, $k$-means clustering is a Euclidean distance-based algorithm, which makes it isotropic. Due to this property, it produces spherical clusters even when the actual clusters are non-spherical [5]. To address this problem, density-based clustering methods, which do not make assumptions on distribution or shapes of data, have been proposed [5], thus used in this paper. Refer to Sec. 3.4 for more discussion on the selection of clustering methods.
**k-means** assign points to clusters based on their minimum distance to cluster centroids, where the number of cluster centroids is equal to \( k \) [5, 129]. The parameter Replicates is used to limit the maximum number of iterations during expectation-maximization and is set to 10. Distance is measured using squared Euclidean distance.

**Hierarchical clustering (HC)** varies based on the proximity measures used. We choose here two of the most used measures—**complete** and **ward** linkages [113, 224]. HC first constructs a dissimilarity matrix using one of these linkages and merges the closest clusters until all points are in a single maximal cluster [5, 113, 224]. We build an agglomerative hierarchy using Euclidean distance.

**Density-based spatial clustering of applications with noise (DBSCAN)** groups closely packed points together and marks points in low-density regions as outliers or noise. Clusters are created based on core points that are selected based on the neighborhood with radius \( \epsilon \) containing at least MinPts points, and other points get assigned to one of these clusters [5, 65, 125]. Parameter MinPts is set to \( \log(N) \) and \( \epsilon \) is set to the value of a point that is farthest away from a line created by connecting the first and the last points in the \( k \)-nearest neighbors distance plot, where \( k = \text{MinPts} \). Distance is measured using squared Euclidean distance.

**Spectral clustering (SC)** uses eigenvalues of graph Laplacian matrices based on the edges in a graph to cluster similar nodes. A similarity graph is constructed using \( k \)-nearest neighbors where \( k \) is set to \( \log(N) \).

### 3.2.2 Evaluation metrics

For validation, data sets with ground-truth values are used in this paper. Here, we use accuracy, purity, and Normalized Mutual Information (NMI) to evaluate and compare the performances of different clustering methods. All metrics are in the range [0, 1], where values close to zero indicate poor clustering and vice versa.

**Accuracy** \((a)\): We find a permutation \((perm)\) of a set of unique values from the resulting labels and find the set that best matches the ground-truth labels. Formally, accuracy \((a)\) is defined as

\[
a = \max_{perm \in P} \frac{1}{N} \sum_{i=0}^{N} V(c_i = d_i),
\]

where \( P \) is the set of all possible permutations of the set of labels acquired from clustering, \( V \) is the binary function that yields 1 when two values are the same and zero for other cases, \( c_i \in perm \) is the label for the \( i \)th data point, and \( d_i \in G \) is the ground truth label for the \( i \)th data point [173].
Figure 3.1: Accuracy, purity, and NMI calculated for \( k \)-means, HC (complete and ward), DBSCAN, and SC results applied to different SLMDS-processed data: equally distributed (\( D_1 \)), varying density (\( D_2 \)), skewed (\( D_3 \)), sub-clustered (\( D_4 \)), and noise (\( D_5 \)) data.

**Purity** \((p)\): The points in each cluster are all assigned to the ground-truth label, which is most frequent in the cluster. The purity is computed by counting the number of correctly labeled points divided by \( N \), the total number of points. Formally, it is defined as \( p = \frac{1}{N} \sum_{i=1}^{k} \max_{j} |C_i \cap D_j| \), where \( k \) is the number of clusters, \( C_i \) is the \( i \)th cluster, and \( D_j \) is the \( j \)th ground-truth class [237]. \( p \) increases with the number of clusters [136].

**Normalized Mutual Information** \((NMI)\): This metric is defined as \( NMI = \frac{2I(L,G)}{H(L)+H(G)} \), where \( I \) is the mutual information between \( L \) and \( G \) and \( H(\cdot) \) denotes the entropy [190].

SDR is performed using [112] and all clustering methods and metrics (except NMI) from this paper are implemented in MATLAB. For NMI, we use the implementation by Chen [32]. We run the experiments on a PC having a Dual-Core Intel Core i5 (2.9 GHz) processor with 8G RAM.

### 3.3 Results

#### 3.3.1 Synthetic data

We first test our pipeline on the same type of synthetic data sets used in Kim *et al.* [112] and further aim to replicate the results of [112] to compare the clustering performance for DR and SDR. We generate several Gaussian random data sets consisting of \( N = 5K \) and \( n = 20 \) to cover five types of...
Figure 3.2: Clusters obtained from \( k \)-means, hierarchical clustering (complete and ward), DBSCAN, and spectral clustering on the LMDS- and SLMDS-processed versions of two human activity data: regrouped human activity data \((n = 50, N = \text{24075})\) and regrouped human activity recognition data \((n = 10, N = \text{7352})\).

inter-sample distance distributions: \((D_1)\) an even spread of equal Gaussian variance (equal distribution); \((D_2)\) an even spread of clusters with different densities; \((D_3)\) an uneven spread of clusters (skewed); \((D_4)\) sub-clustered data with two pairs of sub-clusters and a one single cluster; \((D_5)\) noise (signal-to-noise ratio, \(\text{SNR} = 10\)) added to \(D_1\) \[112\].

We successfully replicated the results from Kim \textit{et al.} and next calculate the evaluation metrics. In Fig. 3.1, all five methods show high metric values for all types of synthetic data sets except for \(D_4\). This is expected because of the low performance of SDR in separating sub-clusters. We also see that \(k\)-means has the best score of accuracy, purity, and NMI for \(D_2\) and \(D_3\). We can also observe that \(k\)-means maintains scores above 0.90 for \(D_1-D_5\), while the other four methods display a drop in \(D_2\). DBSCAN and SC result in the lowest metric scores and SC performs especially poorly compared to the others for \(D_1\) and \(D_4\).
Table 3.1: Average accuracy for $k$-means, HC (complete and ward), DBSCAN, and SC on LMDS- and SLMDS-processed real-world data.

<table>
<thead>
<tr>
<th></th>
<th>$k$-means</th>
<th>HC (complete)</th>
<th>HC (ward)</th>
<th>DBSCAN</th>
<th>SC</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HAD</strong></td>
<td>LMDS</td>
<td>0.8149</td>
<td>0.5745</td>
<td>0.7796</td>
<td>0.5730</td>
</tr>
<tr>
<td></td>
<td>SLMDS</td>
<td>0.9713</td>
<td>0.9388</td>
<td>0.9637</td>
<td>0.7063</td>
</tr>
<tr>
<td><strong>HAR</strong></td>
<td>LMDS</td>
<td>0.9829</td>
<td>0.4971</td>
<td>0.9845</td>
<td>0.8107</td>
</tr>
<tr>
<td></td>
<td>SLMDS</td>
<td>0.9961</td>
<td>0.9961</td>
<td>0.9961</td>
<td>0.9942</td>
</tr>
<tr>
<td><strong>WiFi</strong></td>
<td>LMDS</td>
<td>0.9160</td>
<td>0.8915</td>
<td>0.9040</td>
<td>0.2714</td>
</tr>
<tr>
<td></td>
<td>SLMDS</td>
<td>0.9395</td>
<td>0.9395</td>
<td>0.9395</td>
<td>0.9273</td>
</tr>
<tr>
<td><strong>Banknote</strong></td>
<td>LMDS</td>
<td>0.5758</td>
<td>0.6793</td>
<td>0.5328</td>
<td>0.4932</td>
</tr>
<tr>
<td></td>
<td>SLMDS</td>
<td>0.6844</td>
<td>0.7828</td>
<td>0.6844</td>
<td>0.3134</td>
</tr>
</tbody>
</table>

3.3.2 Real-world data

We use four different SLMDS-processed real-world data sets also tested by Kim *et al.* [112].

**Human Activity Data (HAD)** consists of human activities recorded using an accelerometer from a smartphone ($N = 24075, n = 60$) [54–57]. The HAD data set has five classes (sitting, standing, walking, running, and dancing) and the HAD* data set has four super-classes–sitting, standing, normal (walking), and dynamic (running and dancing) movements–defined later.

**Human Activity Recognition (HAR)** is from the UCI Machine Learning Repository and contains records of six activities captured using a smartphone [54, 171]. We reduce $n$ from 561 to 10 using principal component analysis (PCA), keeping 80% of total variance ($N = 7352, n = 10$) [112]. Activities include lying, standing, sitting, and three walking motions (walking, walking downstairs, and walking upstairs), and HAR* is defined with three super-classes–lying, sitting or standing, and walking (walking, walking up or downstairs) movements–defined later.

**WiFi** is from the UCI Machine Learning Repository and consists of WiFi signal intensities from various routers measured by a smartphone at four different indoor locations ($N = 2K, n = 6$) [18, 54, 172].

**Banknote** is from the UCI Machine Learning Repository [54] and consists of four features extracted using the Wavelet Transform from $N = 1327$
Table 3.2: Average purity for $k$-means, HC (complete and ward), DBSCAN, and SC on LMDS- and SLMDS-processed real-world data.

<table>
<thead>
<tr>
<th></th>
<th>$k$-means</th>
<th>HC (complete)</th>
<th>HC (ward)</th>
<th>DBSCAN</th>
<th>SC</th>
</tr>
</thead>
<tbody>
<tr>
<td>HAD$^*$</td>
<td>LMDS 0.8149</td>
<td>0.6190</td>
<td>0.7796</td>
<td>0.5785</td>
<td>0.8278</td>
</tr>
<tr>
<td></td>
<td>SLMDS 0.9713</td>
<td>0.9388</td>
<td>0.9637</td>
<td>0.7646</td>
<td>0.7672</td>
</tr>
<tr>
<td>HAR$^*$</td>
<td>LMDS 0.9829</td>
<td>0.5257</td>
<td>0.9845</td>
<td>0.8139</td>
<td>0.9763</td>
</tr>
<tr>
<td></td>
<td>SLMDS 0.9961</td>
<td>0.9961</td>
<td>0.9961</td>
<td>0.9959</td>
<td>0.9961</td>
</tr>
<tr>
<td>WiFi</td>
<td>LMDS 0.9160</td>
<td>0.8915</td>
<td>0.9040</td>
<td>0.2781</td>
<td>0.9055</td>
</tr>
<tr>
<td></td>
<td>SLMDS 0.9395</td>
<td>0.9395</td>
<td>0.9395</td>
<td>0.9373</td>
<td>0.7825</td>
</tr>
<tr>
<td>Banknote</td>
<td>LMDS 0.5758</td>
<td>0.6793</td>
<td>0.5554</td>
<td>0.7252</td>
<td>0.5554</td>
</tr>
<tr>
<td></td>
<td>SLMDS 0.6844</td>
<td>0.7828</td>
<td>0.6844</td>
<td>0.9785</td>
<td>0.5906</td>
</tr>
</tbody>
</table>

Table 3.3: Average NMI for $k$-means, HC (complete and ward), DBSCAN, and SC on LMDS- and SLMDS-processed real-world data.

<table>
<thead>
<tr>
<th></th>
<th>$k$-means</th>
<th>HC (complete)</th>
<th>HC (ward)</th>
<th>DBSCAN</th>
<th>SC</th>
</tr>
</thead>
<tbody>
<tr>
<td>HAD$^*$</td>
<td>LMDS 0.7101</td>
<td>0.5930</td>
<td>0.7471</td>
<td>0.5880</td>
<td>0.7077</td>
</tr>
<tr>
<td></td>
<td>SLMDS 0.9090</td>
<td>0.8441</td>
<td>0.8969</td>
<td>0.7656</td>
<td>0.8106</td>
</tr>
<tr>
<td>HAR$^*$</td>
<td>LMDS 0.9306</td>
<td>0.2566</td>
<td>0.9313</td>
<td>0.7975</td>
<td>0.9110</td>
</tr>
<tr>
<td></td>
<td>SLMDS 0.9767</td>
<td>0.9767</td>
<td>0.9767</td>
<td>0.9705</td>
<td>0.9767</td>
</tr>
<tr>
<td>WiFi</td>
<td>LMDS 0.7887</td>
<td>0.7386</td>
<td>0.7611</td>
<td>0.0549</td>
<td>0.7759</td>
</tr>
<tr>
<td></td>
<td>SLMDS 0.8686</td>
<td>0.8686</td>
<td>0.8686</td>
<td>0.8536</td>
<td>0.7682</td>
</tr>
<tr>
<td>Banknote</td>
<td>LMDS 0.0184</td>
<td>0.2068</td>
<td>0.0045</td>
<td>0.2106</td>
<td>0.0123</td>
</tr>
<tr>
<td></td>
<td>SLMDS 0.2094</td>
<td>0.4044</td>
<td>0.2094</td>
<td>0.5311</td>
<td>0.0718</td>
</tr>
</tbody>
</table>

grayscale images of banknote specimens ($n = 4$). Samples are labeled as either genuine or forged.

3.3.3 Qualitative evaluation

We show in Fig. 3.2 the projections color-coded based on their clustering results of HAD$^*$ and HAR$^*$ data sets. Note that we here mainly show results with super-class labels (HAD$^*$ and HAR$^*$) and add the results of sub-class labels (HAD and HAR) to the supplemental materials ([90]) instead. We discuss these later in Sec. 3.4. For the HAD$^*$ data set, Fig. 3.2 shows
that the cluster related to walking is placed close to the cluster dynamic movements, and the other two clusters for sitting and standing have more distance between each other and from other activities. Overall, \( k \)-means and HC perform better at separating the four super-classes compared to other clustering methods. Further note that the clustering performances \((i.e.,\) accuracy, purity, and NMI\) of most clustering methods are higher for SLMDS-processed data than LMDS-processed data due to the sharpening effect.

For the HAR\(*\) data set, the LMDS projection displays three visually distinguishable clusters, which are placed very close to each other and we observe a lot of noise near the boundaries of the clusters. This makes the clustering more challenging near the boundaries. This phenomenon is most evident in HC (complete) and DBSCAN. In contrast, SLMDS produces three well-separated spherical groups. All five clustering methods seem to correctly separate the data into three distinct clusters. Due to space limitations, clustering results for WiFi and Banknote data sets are added to the supplemental materials ([90]).

### 3.3.4 Quantitative evaluation

We compare next the evaluation metrics introduced in Sec. 3.2 (Subsec. 3.2.2). Tables 3.1–3.3 each show different evaluation scores for real-world data sets. Overall, we observe from the clustering evaluation metrics that \( k \)-means performs the best and SC performs the worst (lower accuracy and purity scores and little difference in NMI). HC (complete and ward) shows similar scores, but HC (ward) produces slightly higher scores than HC (complete) for the HAD\(*\) and HAR\(*\) data sets.

Tables 3.1–3.3 show that accuracy, purity, and NMI scores for SLMDS are higher than those for LMDS using all real-world data sets, except the Banknote data set (see Sec. 3.4 for more discussion and limitations). We observe further that SLMDS improves all three evaluation metric scores for most clustering methods \((k\)-means and HC complete and ward\). Especially for the WiFi and HAR\(*\) data sets, the scores were near unity (highest score).

### 3.4 Discussion

**Selection of clustering methods** The selection is based on the method’s availability, ease-of-use, and performance. To show the overall performance of the selected clustering methods regardless of SLMDS and LMDS,
we computed the quantitative metrics for plain 2D synthetic data (varying the number of clusters and \( N \)). We used ten data sets for each combination to calculate the average performance of each clustering method. Results show that the three quantitative metrics for all five clustering methods are above 0.9 for all data sets when the number of clusters are set to five (see supplemental materials ([90])).

**Computational scalability** The time complexity of \( k \)-means is \( O(IkNn) \), where \( I \) is the number of iterations, \( k \) is the number of clusters, \( N \) is the number of observations, and \( n \) is the number of dimensions [136]. The time complexity is \( O(N^3) \) for HC [49], \( O(N \log N) \) for DBSCAN [65, 125], and \( O(N^3) \) for SC [205]. We computed the wall-clock time measurements using the same data set from above and conclude that \( k \)-means clustering is the fastest among the five clustering methods (see supplemental materials [90] for detailed results).

**Limitations** Applying clustering to SLMDS has some limitations. First, the parameter settings for both clustering and SLMDS are not entirely automatic, which requires time and effort from the end-users to fully explore the parameter space. Next, clustering sub-clusters is still challenging. SLMDS by nature captures the super-class structures better than subclasses (refer to clustering metric scores of synthetic data sets in the supplemental materials [90]) [112]. This means that the performance of clustering largely depends on SLMDS and that in some cases clustering will perform badly even though clusters are well-separated as in Fig. 3.2.

### 3.5 conclusion

We presented a pipeline consisting of SDR and clustering to ease the manual labeling process for end-users using SDR. We tested our pipeline using
both synthetic and real-world data sets and compared the clustering performances of SDR- and DR-processed data. Overall, the qualitative and quantitative results verified that the highly separated clusters produced by SDR yield higher clustering performance (than DR), which has not been shown before. Furthermore, we recommend $k$-means for clustering human activity data based on the ease-of-use, performance, and computational scalability. This proposed pipeline and set of validation metrics will ultimately provide a user-friendly environment for data exploration and semi-automatic classification.

**Acknowledgements**

This work is supported by DSSC Doctoral Training Program co-funded by the Marie Sklodowska-Curie COFUND project (DSSC 754315).

**Appendices**

We present next the additional results to support Section 3.3.

Table 3.4: Average accuracy, purity, NMI scores for $k$-means, hierarchical clustering (complete and ward linkage), DBSCAN, and spectral clustering on SLMDS-processed HAD and HAR data sets

<table>
<thead>
<tr>
<th>Measurement</th>
<th>Data</th>
<th>$k$-means</th>
<th>HC (complete)</th>
<th>HC (ward)</th>
<th>DBSCAN</th>
<th>SC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>HAD</td>
<td>0.8137</td>
<td>0.8128</td>
<td>0.8039</td>
<td>0.6624</td>
<td>0.8488</td>
</tr>
<tr>
<td></td>
<td>HAR</td>
<td>0.6249</td>
<td>0.6005</td>
<td>0.6340</td>
<td>0.5459</td>
<td>0.6301</td>
</tr>
<tr>
<td>Purity</td>
<td>HAD</td>
<td>0.8704</td>
<td>0.8396</td>
<td>0.8558</td>
<td>0.7205</td>
<td>0.8577</td>
</tr>
<tr>
<td></td>
<td>HAR</td>
<td>0.6384</td>
<td>0.6079</td>
<td>0.6413</td>
<td>0.5468</td>
<td>0.6339</td>
</tr>
<tr>
<td>NMI</td>
<td>HAD</td>
<td>0.8161</td>
<td>0.7705</td>
<td>0.8059</td>
<td>0.7189</td>
<td>0.8149</td>
</tr>
<tr>
<td></td>
<td>HAR</td>
<td>0.6451</td>
<td>0.6754</td>
<td>0.6722</td>
<td>0.7433</td>
<td>0.6159</td>
</tr>
</tbody>
</table>
Figure 3.4: Clusters obtained from $k$-means, hierarchical clustering (complete and ward linkage), DBSCAN, and spectral clustering on the LMDS- and SLMDS-processed versions of the HAD data set ($n = 50$, $N = 24075$). The HAD results show overlaps between two dynamic activities—running and dancing. The other three activities (sitting, standing, and walking) seem to have little overlap with different activities.
Figure 3.5: Clusters obtained from k-means, hierarchical clustering (complete and ward linkage), DBSCAN, and spectral clustering on the LMDS- and SLMDS-processed versions of the HAR data set ($n = 10$, $N = 7352$). The HAR results contain two overlapping regions. The first region contains two static human activities (sitting and standing) and the other consists of three dynamic walking motions (walking, walking downstairs and upstairs).
Figure 3.6: Accuracy of the clusters obtained from $k$-means, hierarchical clustering (complete and ward linkage), DBSCAN, and spectral clustering on 2D synthetic data. These results show that all five clustering algorithms yield nearly perfect accuracy for any number of clusters or observations. Hierarchical clustering (complete and ward linkage) and spectral clustering perform well especially without fluctuation of values. Although $k$-means displays a slight drop in accuracy for data with five clusters and 3000 observations, the values are still above 0.9.
Figure 3.7: Purity of the clusters obtained from k-means, hierarchical clustering (complete and ward linkage), DBSCAN, and spectral clustering on 2D synthetic data. These results show that all five clustering algorithms yield nearly perfect purity for any number of clusters or observations. Hierarchical clustering (complete and ward linkage) and spectral clustering perform well especially without fluctuation of values. Although k-means displays a slight drop in purity for data with five clusters and 3000 observations, the values are still above 0.9.
Figure 3.8: NMI of the clusters obtained from $k$-means, hierarchical clustering (complete and ward linkage), DBSCAN, and spectral clustering on 2D synthetic data. These results show that all five clustering algorithms yield nearly perfect NMI for any number of clusters or observations. Hierarchical clustering (complete and ward linkage) and spectral clustering perform well especially without fluctuation of values. Although $k$-means displays a slight drop in NMI for data with five clusters and 3000 observations, the values are still above 0.9.
Figure 3.9: Wall-clock timing of k-means, hierarchical clustering (complete and ward linkage), DBSCAN, and spectral clustering on 2D synthetic data with varying number of clusters and observations. We observe that k-means is the most fastest among other clustering algorithms.
Figure 3.10: Clusters obtained from $k$-means, hierarchical clustering (complete and ward linkage), DBSCAN, and spectral clustering on the LMDS- and SLMDS-processed versions of the WiFi data set ($n = 6, N = 2K$).
Figure 3.11: Clusters obtained from $k$-means, hierarchical clustering (complete and ward linkage), DBSCAN, and spectral clustering on the LMDS- and SLMDS-processed versions of the Banknote data set ($n = 1327$, $N = 4$).
3.5 CONCLUSION
Dimensionality reduction (DR) methods create 2D scatterplots of high-dimensional data for visual exploration. As such scatterplots are often used to reason about the cluster structure of the data, this requires DR methods with good cluster preservation abilities. Recently, Sharpened DR (SDR) was proposed to enhance the ability of existing DR methods to create scatterplots with good cluster structure. Following this, SDR-NNP was proposed to speed the computation of SDR by deep learning. However, both SDR and SDR-NNP require careful tuning of four parameters to control the final projection quality. In this work, we extend SDR-NNP to simplify its parameter settings. Our new method retains all the desirable properties of SDR and SDR-NNP. In addition, our method is stable vs setting all its parameters, making it practically a parameter-free method, and also increases the quality of the produced projections. We support our claims by extensive evaluations involving multiple data sets, parameter values, and quality metrics.

4.1 Introduction

The visual analysis of high-dimensional data is challenging due to its many observations (also known as points or samples) and values recorded per sample (also known as dimensions, features, or variables) [63, 127, 153]. Dimensionality reduction (DR), also known as projection, is particularly suited for such data since DR methods scale visually to thousands of dimensions and hundreds of thousands of samples. DR techniques such as the well-known t-SNE [133] and UMAP [145] methods, can segregate data clusters into well-separated visual clusters, which enables one to reason about the former by seeing the latter, a property also known as preservation of data structure [13].

A recent survey [63] noted that many DR techniques score below t-SNE or UMAP in cluster segregation but have other important assets – simple
usage and implementation, computational scalability, and out-of-sample behavior. Following this, [112] recently proposed Sharpened DR (SDR) to generically improve the cluster segregation ability of any DR technique by sharpening the input data by a variant of the Mean Shift (MS) algorithm [44]. However, SDR is impractical to use as MS is prohibitively expensive in high dimensions.

In a recent paper [111], we reduced the computational cost of SDR by using deep learning. Our proposed method, called SDR-NNP, leverages an earlier DR method, called Neural Network Projection (NNP) [60], to learn the combined steps of data sharpening and projection. SDR-NNP has the following features – to our knowledge, not yet jointly achieved by existing DR methods:

**Quality (C1):** Better cluster separation than existing DR methods, as measured by well-known metrics in the DR literature;

**Scalability (C2):** Linear in sample and dimension counts, allowing the projection of data sets of up to a million samples and hundreds of dimensions in a few seconds on commodity GPU hardware;

**Genericity (C3):** Handles any real-valued (unlabeled) high-dimensional data;

**Stability and out-of-sample (OOS) support (C4):** Projects new samples for a learned projection without recomputing it, in contrast to standard t-SNE and any other non-parametric methods.

However, SDR-NNP depends on four parameters – the number of nearest neighbors $k_s$ in the MS process, the number of MS data-sharpening iterations $t$, the so-called learning rate $\alpha$ (speed of MS), and the number of training epochs $e$. While SDR-NNP proposes a good default for $e$, it only suggests ranges from which users can pick $t$ and $\alpha$ and does not further explore how to set $k_s$. Tuning each single parameter can change the projection, and also the projection quality measured by established metrics, in subtle ways. In practice, users have to examine different combinations of $k_s$, $t$, and $\alpha$ by trial-and-error. This is slow since all these parameters affect the training data that SDR-NNP uses, i.e., one has to retrain the method after each parameter change. More importantly, if changing these parameters can lead to very different projections, then the entire goal of stability – that is, having a method that generates consistent results for a given input data set – would be compromised. Simply put: Sharpening the data, as SDR-NNP does it, is useful and desired, but only effective in practice if it can be done in a stable, ideally parameter-free, manner.
In this paper, we address these issues by reducing SDR-NNP’s four-parameter space to a single parameter. The new parameter $K_m$ controls a $k$-means clustering process done in the input high-dimensional data. We control SDR’s data sharpening process based on the local homogeneity of neighborhoods in terms of the cluster labels they get assigned by $k$-means. Our new method, which we call $\alpha$-SDR-NNP, keeps the quality (C1), scalability (C2), genericity (C3), and stability and OOS (C4) features of SDR-NNP listed above.

**Ease-of-use (C5):** Our new method is far stabler than SDR-NNP in terms of visual cluster separation and quality metrics of the computed projections for both changes in the four parameters $k_s$, $t$, $\alpha$, and $e$ it inherits from SDR-NNP and the new parameter $K_m$ it adds. Practically put, our new method can be seen as parameter-free, thus stable in its application. Our method also increases the quality (C1) of the produced projections vs SDR-NNP for the same parameter values.

We structure this paper as follows: Section 4.2 discusses related work on dimensionality reduction. Section 4.3 details SDR-NNP and $\alpha$-SDR-NNP. Section 4.4 presents the results that support our above claims, including a detailed quantitative and qualitative comparison of SDR-NNP and $\alpha$-SDR-NNP. Section 4.5 discusses our two methods. Finally, Section 4.6 concludes the paper.

### 4.2 Background

Let $x = (x^1, \ldots, x^n)$, $x^i \in \mathbb{R}$, $1 \leq i \leq n$ be an $n$-dimensional (nD) real-valued sample, and let $D = \{x_j\}$, $1 \leq j \leq N$ be a data set of $N$ samples. A DR technique is a function

$$P : \mathbb{R}^n \to \mathbb{R}^q,$$

where $q \ll n$, and typically $q = 2$. The projection $P(x)$ of a sample $x \in D$ is a point $p \in \mathbb{R}^q$. Projecting the whole set $D$ yields a $q$D scatterplot denoted next as $P(D)$.

DR methods aim to satisfy multiple requirements. Table 4.1 outlines prominent ones present in several DR surveys [47, 58, 63, 95, 127, 153, 184, 211, 230]. Besides these, DR techniques also require locality, steerability, and multilevel computation [153]. We do not focus on such additional requirements as these are less mainstream.

The quality (Q) and cluster separation (CS) requirements need additional explanations. Projection quality is assessed by *local* metrics that measure
how a small neighborhood of points in $D$ maps to a neighborhood in $P(D)$ and conversely. Local quality metrics include the following (see Tab. 4.2 for the formal definitions):

**Trustworthiness** $Q_t$ [215]: Measures the fraction of close points in $D$ that are also close in $P(D)$. $Q_t$ tells how much one can trust that local patterns in a projection represent actual data patterns. In the definition (Table 4.2), $U_i^{(k)}$ is the set of points that are among the $k$ nearest neighbors of point $i$ in the 2D space but not among the $k$ nearest neighbors of point $i$ in $\mathbb{R}^n$; and $r(i, j)$ is the rank of the 2D point $j$ in the ordered-set of nearest neighbors of $i$ in $P(D)$;

**Continuity** $Q_c$ [215]: Measures the fraction of close points in $P(D)$ that are also close in $D$. In the definition (Table 4.2), $V_i^{(k)}$ are the points in the $k$ nearest neighbors of point $i$ in $D$ but not among the $k$ nearest neighbors in 2D; and $\hat{r}(i, j)$ is the rank of the $\mathbb{R}^n$ point $j$ in the ordered set of nearest neighbors of $i$ in $D$;

**Neighborhood Hit** $Q_h$ [158]: Measures how well a projection $P(D)$ separates labeled data. $Q_h$ is the number $y^l_k$ of the $k$ nearest neighbors of a point $y \in P(D)$, denoted by $y_k$, that have the same label $l$ as $y$, averaged over $P(D)$. Put simply: consider a projection, i.e., a 2D scatterplot $P(D)$, where every point has a label equal to the label the corresponding high-dimensional point projected there. If we assume a well-separated data set in the high-dimensional space, i.e., a data set where close points in this space have similar labels, then a good projection should keep this structure – that is, close points in the 2D scatterplot should also have similar labels. The usage and practical intuition behind the $Q_h$ metric has been extensively explored in DR literature, see for example [169]. By construction, the $Q_h$ metric requires labeled data sets to be used.

**Shepard diagram correlation** $Q_r$ [105]: The Shepard diagram is a scatterplot of the pairwise distances between all points in $P(D)$ vs the corresponding distances in $D$. Points below, respectively above, the main diagonal show distance ranges for which false neighbors, respectively missing neighbors, occur. The closer the plot is to the main diagonal, the better the overall distance preservation is. The scatterplot’s Spearman rank correlation $Q_r$ measures this – a value $Q_r = 1$ indicates a perfect (positive) distance correlation.

All above metrics are *local*, i.e., capture preservation of data structure in $D$ at the scale given by the neighborhood size $k$. In practice, what a ‘good’ $k$ value is for a given data set $D$ is unknown. $k$ can also vary locally within $D$. 

102
as function of the point density. At a higher level, projections are used to reason about the overall data structure in $D$ by creating, ideally, visual clusters that are as well separated in $P(D)$ as data clusters are in $D$, a property called cluster separation (CS). High-CS projections show, e.g., how many point clusters exist and how these correlate (or not) with labels or specific attributes [153], or predict how easy it is to train a classifier for $D$ based on the CS in $P(D)$ [169]. In general, it is hard to design objective metrics for CS like one does for local quality, because a ‘well separated data cluster’ in $D$ is not evident. Hence, CS is typically assessed on (labeled) data sets $D$ for which the ground-truth data-separation is well known, e.g., MNIST [121].

Table 4.1: Summary of desirable requirements (characteristics) of DR methods.

<table>
<thead>
<tr>
<th>Requirement name</th>
<th>Description of the requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quality (Q)</td>
<td>Captures local data structures well, as measured by the projection local-quality metrics in Table 4.2.</td>
</tr>
<tr>
<td>Cluster separation (CS)</td>
<td>Captures data structures present at larger scales than local structures, e.g., clusters, as visual clusters in the 2D scatterplot.</td>
</tr>
<tr>
<td>Scalability (S)</td>
<td>Can project data sets of hundreds of dimensions and millions of samples in a few seconds on commodity hardware.</td>
</tr>
<tr>
<td>Ease-of-use (EoU)</td>
<td>Has few (ideally: no) free parameters, which are intuitive and easy to tune to get the desired results.</td>
</tr>
<tr>
<td>Genericity (G)</td>
<td>Can project any (real-valued) data set, with or without labels.</td>
</tr>
<tr>
<td>Out-of-sample (OOS)</td>
<td>Can fit new data in an existing projection. OOS projections are also stable – small input-data changes cause only small projection changes.</td>
</tr>
</tbody>
</table>

Table 4.2: Local quality metrics for projections. All metrics range in $[0, 1]$ with 0 being lowest, and 1 being highest, quality.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trustworthiness ($Q_t$)</td>
<td>$1 - \frac{2}{NK(N-1)} \sum_{i=1}^{K} \sum_{j=1}^{K} (r(i, j) - k)$</td>
</tr>
<tr>
<td>Continuity ($Q_c$)</td>
<td>$1 - \frac{2}{NK(N-1)} \sum_{i=1}^{K} \sum_{j=1}^{K} (\hat{r}(i, j) - k)$</td>
</tr>
<tr>
<td>Neighborhood hit ($Q_h$)</td>
<td>$\frac{1}{N} \sum_{y \in P(D)} \hat{y}_i/y_k$</td>
</tr>
<tr>
<td>Shepard diagram correlation ($Q_r$)</td>
<td>Spearman’s rank of $(</td>
</tr>
</tbody>
</table>

We next discuss existing DR methods in the light of the requirements in Tab. 4.1. We group these into unsupervised and supervised methods, as follows.

**Unsupervised methods:** Principal Component Analysis [106] (PCA) is simple, fast, out-of-sample (OOS), and easy-to-interpret, also used as preprocessing for other DR techniques that require a moderate data dimen-
sionality $n$ [153]. Being linear and global, PCA has low quality and CS, especially for data of high intrinsic dimensionality.

MDS [202], Landmark MDS [182], Isomap [195], and LLE [174] with its variations [51, 235, 236] detect and project the (neighborhood of the) high-dimensional manifold on which data is embedded, and can capture non-linear data structure. Such methods yield higher quality than PCA, but can be hard to tune, do not all support OOS, and do not work well for high-intrinsic-dimensional data.

Force-directed methods such as LAMP [105] and LSP [158] yield good quality, good scalability, and are simple to use. Yet, not all force-directed methods have OOS ability. Clustering-based methods, such as PBC [156], share many features with force-directed methods, such as good quality, but also lack OOS.

Stochastic Neighborhood Embedding (SNE) methods, like the well-known $t$-SNE [133], have high overall quality and CS. Yet, $t$-SNE has a (high) complexity of $O(N^2)$ in sample count, is very sensitive to small data changes, can be hard to tune [225], and has no OOS. Tree-accelerated $t$-SNE [210], hierarchical SNE [160], approximated $t$-SNE [161], and various GPU variants of $t$-SNE [29, 163] improve scalability, but are algorithmically quite complex, and still have sensitivity, tuning, and OOS issues. Uniform Manifold Approximation and Projection (UMAP) [145] has comparable quality to $t$-SNE, is much faster and has OOS. Still, UMAP is also sensitive to parameter tuning.

Autoencoders (AE) [93, 114] aim to generate a compressed, low-dimensional representation of the data in their bottleneck layers by training to reproduce the data input at the output. They have similar quality to PCA and are easy to set up, train, and use, are fast, and have OOS abilities. Self-organizing maps (SOM) [117] share with AE the ease-of-use, training, and speed. Yet, both AE and SOM lag behind $t$-SNE and UMAP in CS, which is, as explained, essential for interpreting projections.

**Supervised methods:** ReNDA [11] uses two neural networks to implement (1) a nonlinear generalization of Fisher’s Linear Discriminant Analysis [70] and (2) an autoencoder, used for regularization. ReNDA scores well on predictability and has OOS, but needs pre-training of each individual network and has low scalability. Recently, Neural Network Projections (NNP)[60] proposed to select a subset $D_s \subset D$ to project by any DR method to yield a training projection $P(D_s) \subset \mathbb{R}^2$. The pair $(D_s, P(D_s))$ is then used to train a regression neural network. NNP is very fast, simple
to use, generic, and has OOS. NNP’s major limitation is a lower CS than its training projection $P(D_s)$.

SDR-NNP [111], our earlier method which we extend in this paper, effectively runs NNP on a training set of high-dimensional samples which is first sharpened by mean shift (described further below). SDR-NNP keeps all desirable features of NNP except ease-of-use: Sharpening requires carefully setting three parameters to get good final results. We describe SDR-NNP in full detail in Sec. 4.3 as it forms the basis of our new technique $\alpha$-SDR-NNP which solves the parameter setting problem.

**Semi-supervised methods:** The SSNP method [61] takes a mid path between supervised methods (e.g., NNP) and unsupervised ones (e.g., AE). Similar to NNP, SSNP has an encoder-decoder architecture. Besides the standard reconstruction loss in autoencoders (AEs), SSNP adds a classification loss. This loss uses either ground-truth labels from the data set $D$ or pseudolabels computed from $D$ by a clustering algorithm. That is, SSNP aims to jointly (a) reconstruct an input data set from a low-dimensional (more precisely, 2D) representation and (b) classify the input samples based on their (pseudo)labels. The combination of both losses creates a projection which both preserves the original dimensions of the data (a) and also the coarse-scale similarity of the data points (b). SNP produces 2D projections which look quite similar to those created by our methods described in Sec. 4.3. However, important differences exist:

- Our methods consists of two distinct operations: high-dimensional data sharpening, followed by projection. SSNP only performs the projection step;
- SSNP is a semi-supervised method that uses only label information to learn how to project. Our methods, like NNP, learn from a user-selected projection technique;
- SSNP and our methods use fundamentally different network architectures. SSNP uses two different networks for training and inference. We use a single architecture for training and inference;
- A key goal for SDR-NNP is to enhance separation between unla-

leled data clusters so that these can next be labeled by users (see next Sec. 4.4.4). This is out of scope for SSNP.

**Sharpening data:** Finding clusters of similar data points is a key task in data science, addressed by tens of clustering methods [16, 231]. Mean Shift (MS) [34, 44, 73] is particularly relevant to our work. MS computes the kernel density estimation of a data set $D$ and next shifts points in $D$ upstream
along the density gradient. This effectively clusters \( D \), with applications in image segmentation \([44]\) and graph drawing \([99]\). Recently, Sharpened DR (SDR) \([112]\) used MS for the first time to assist DR: A data set \( D \) is sharpened by a few MS iterations, not to be confused with the clustering goal of the original MS. The sharpened data set is next projected by a fast, easy-to-use, but potentially low-CS DR method. Sharpening ‘preconditions’ the used DR method to overcome its lack of CS. Yet, as MS is very slow for high-dimensional data, this makes SDR impractical.

Table 4.3: Summary of DR techniques in Sec. 4.2 and their features from Table 4.1.

<table>
<thead>
<tr>
<th>Method</th>
<th>Q</th>
<th>S</th>
<th>EoU</th>
<th>G</th>
<th>OOS</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCA</td>
<td>low</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>yes</td>
</tr>
<tr>
<td>MDS</td>
<td>mid</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>no</td>
</tr>
<tr>
<td>L-MDS</td>
<td>mid</td>
<td>mid</td>
<td>low</td>
<td>low</td>
<td>no</td>
</tr>
<tr>
<td>Isomap</td>
<td>mid</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>no</td>
</tr>
<tr>
<td>LLE</td>
<td>mid</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>no</td>
</tr>
<tr>
<td>LAMP</td>
<td>mid</td>
<td>mid</td>
<td>high</td>
<td>high</td>
<td>yes</td>
</tr>
<tr>
<td>LSP</td>
<td>mid</td>
<td>mid</td>
<td>high</td>
<td>high</td>
<td>no</td>
</tr>
<tr>
<td>PBC</td>
<td>mid</td>
<td>mid</td>
<td>high</td>
<td>high</td>
<td>no</td>
</tr>
<tr>
<td>UMAP</td>
<td>high</td>
<td>high</td>
<td>low</td>
<td>high</td>
<td>yes</td>
</tr>
<tr>
<td>( t )-SNE</td>
<td>high</td>
<td>low</td>
<td>low</td>
<td>high</td>
<td>no</td>
</tr>
<tr>
<td>Autoencoders</td>
<td>low</td>
<td>high</td>
<td>high</td>
<td>low</td>
<td>yes</td>
</tr>
<tr>
<td>SOM</td>
<td>low</td>
<td>high</td>
<td>high</td>
<td>low</td>
<td>no</td>
</tr>
<tr>
<td>ReNDA</td>
<td>mid</td>
<td>low</td>
<td>low</td>
<td>mid</td>
<td>yes</td>
</tr>
<tr>
<td>NNP</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>yes</td>
</tr>
<tr>
<td>SDR</td>
<td>high</td>
<td>low</td>
<td>mid</td>
<td>high</td>
<td>no</td>
</tr>
<tr>
<td>SDR-NNP</td>
<td>high</td>
<td>high</td>
<td>low-mid</td>
<td>high</td>
<td>yes</td>
</tr>
<tr>
<td>( \alpha )-SDR-NNP</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>yes</td>
</tr>
</tbody>
</table>

Table 4.3 summarizes the DR techniques discussed above showing how they fare with respect to the requirements discussed earlier in this section. No reviewed method satisfies all the requirements optimally. We next describe our earlier method SDR-NNP (Tab. 4.3 one but last row) and our extension to it, \( \alpha \)-SDR-NNP (Tab. 4.3 last row).

### 4.3 SDR-NNP AND \( \alpha \)-SDR-NNP METHODS

As Sec. 4.2 explained, SDR and NNP have complementary features: SDR yields good cluster separation (CS), while NNP is fast, easy to use, and has OOS ability. Our combined SDR-NNP technique joins these advantages and works as follows (see also Fig. 4.1, parts marked in blue). We use SDR to sharpen a small data subset to create an initial 2D projection (Sec. 4.3.1).
Next, we train NNP on the sharpened data and its 2D projection (Sec. 4.3.2) and use it to project the whole data set. Section 4.3.3 introduces \( \alpha \)-SDR-NNP and outlines how this method improves upon SDR-NNP.

4.3.1 **Sharpened Dimensionality Reduction**

SDR has two main components, as follows (for full details, see [112]):

**Data sharpening:** Given a data set \( D \in \mathbb{R}^n \), SDR computes its density using the kernel density estimator \( \rho(x) : \mathbb{R}^n \rightarrow \mathbb{R}^+ \) defined as

\[
\rho(x) = \sum_{y \in N(x)} L \left( \frac{\|x - y\|}{h} \right),
\]

(4.2)

where \( N(x) \) is the set of \( k_s \)-nearest neighbors of \( x \) in \( D \); \( L \) is a parabolic kernel [59]; and \( h \) is the distance of \( x \) to its \( k_s^{th} \) (farthest) neighbor in \( N(x) \). In other words, the ‘bandwidth’ of density estimation \( h \), which further determines how data sharpening finds clusters in the input data set, is locally controlled by the number of nearest neighbors \( k_s \). As we shall see next in Sec. 4.3.3, our new method simplifies this even further by removing the need to explicitly specify \( k_s \).

Next, SDR shifts points \( x \in D \) using the update rule

\[
x_{\text{next}} = x + \alpha \frac{\nabla \rho(x)}{\max(\|\nabla \rho(x)\|, \epsilon)},
\]

(4.3)
where \( \alpha \in [0, 1] \) is a ‘learning rate’ parameter that controls the shift speed (higher values yield higher speed) and \( \epsilon = 10^{-5} \) is a regularization parameter. After every update (Eqn. 4.3), the density \( \rho \) is computed again (Eqn. 4.2). This sharpening approach is called Local Gradient Clustering (LGC) by analogy with Gradient Clustering (GC) [73].

SDR has three parameters: \( t \) (number of iterations); \( k_s \) (number of nearest neighbors); and \( \alpha \) (learning rate), all marked in brown in Fig. 4.1. The SDR-NNP method uses \( k_s \geq 50 \) following [112], and setting \( \alpha \) and \( t \) is discussed in Sec. 4.4. The \( \alpha \)-SDR-NNP method replaces the need to fiddle with these parameters (see Secs. 4.3.3 and 4.4.2).

**Projection:** SDR takes the LGC-sharpened data set \( D_s \) produced from the input data set \( D \) and projects it by a projection method of choice \( P \) (typically fast but not necessarily OOS), called the baseline DR method next, to obtain a 2D projection \( P(D_s) \). The data in \( D_s \) are better separated than in \( D \) due to LGC, which helps \( P \) to yield better cluster separation in \( P(D_s) \) than in \( P(D) \).

### 4.3.2 SDR-NNP

SDR-NNP uses SDR (Sec. 4.3.1) on a small data subset to obtain \( P(D_s) \). To project the full data set \( D \), one next trains the NNP regressor [60] using \( D_s \) as input and \( P(D_s) \) as output. The NNP network has three fully-connected hidden layers with ReLU activation [4], initial weights set to He Uniform [82], and an initial bias value set to 0.0001. The output layer has 2 units, one per 2D coordinate, and uses sigmoid activation to constrain output values to \([0, 1]\). We used three different network sizes, namely, \( x\text{-small} \) (75, 30, 75 units per layer), \( \text{small} \) (150, 60, 150 units per layer) and \( \text{medium} \) (300, 120, 300 units per layer). We trained the network using the ADAM optimizer [115], as described in the NNP paper. After training, SDR-NNP has a regressor able to mimic the behavior of SDR for unseen data, thus adding OOS capability, and computational scalability to SDR.

### 4.3.3 \( \alpha \)-SDR-NNP

A key challenge for the original SDR method [112] which is also shared by SDR-NNP is the control of the sharpening process. As outlined there, the projection results can be quite sensitive to the exact combination of \( \alpha \), \( k_s \), and \( t \) parameters. To these, SDR-NNP introduces a fourth parameter, the number of training epochs \( e \). Earlier results from both the original SDR and SDR-NNP, and as we also discuss next in Sec. 4.4.1, show that
4.3 \textit{sdr-nnp and $\alpha$-sdr-nnp methods}

the hardest to control parameters are $\alpha$ and the number of nearest neighbors $k_s$. These two parameters influence most the visual cluster separation which is the main added value behind the SDR and SDR-NNP proposal. As Kim \textit{et al.} \cite{112} noted, these parameters are not fully independent – when changing $k_s$, one should also change the considered range for exploration of good $\alpha$ values. The interdependency of the same two parameters was observed when using LGC to bundle graph and trail drawings \cite{99, 212}.

To alleviate this problem, we must understand its causes. Consider a data set $D \in \mathbb{R}^n$ and its density estimation $\rho$ (Eqn. 4.2). For simplicity, we depict this for the 1D case in Fig. 4.2a. For the example in the figure, the original data set had two quite well separated clusters, shown by the red and cyan bars denoting high sample density $\rho$ (Fig. 4.2a). LGC sharpens this density, practically separating the two high-density clusters even further (Fig. 4.2c). This is the desired outcome since such well-separated data clusters will project to well separated visual clusters further by SDR or SDR-NNP.

However, consider now a data set $D$ with the density $\rho$ as in Fig. 4.2b. There is far less clear separation between data clusters here, shown by the fading-to-white bars below the density plot in image (b). From this input, LGC will create a sharpened data set $D_s$ looking as in Fig. 4.2b, \textit{i.e.}, very similar to the one where the density showed two very clearly separated peaks. This is due to the normalization of the gradient in the LGC update rule (Eqn. 4.3) which means that small density variations have similar sharpening effects as large ones. This is clearly not desirable since it separates the two density peaks in Fig. 4.2b too strongly. The result is \textit{ oversegmentation} of the projection $P(D_s)$, as observed in \cite{111, 112}. Rather,
we would like to obtain the data set $D_s$ in Fig. 4.2d which sharpens density only where it is clearly well separated from neighbor peaks, and leaves the between-peak region, where we cannot really identify two separated clusters, unchanged.

To achieve this, we need supplementary information, namely that indicated by the red and cyan bars in Fig. 4.2, i.e., where the high values of the data set’s density are located. This implies that we cannot use a single global parameter setting for LGC, but rather need to modulate LGC by such local information. We achieve this in our new $\alpha$-NNP-SDR method as follows (see also Fig. 4.1, parts marked in green). First, we use a simple clustering algorithm, $k$-means, to cluster NNP-SDR’s training data $D$, with a user-set cluster-count parameter $K_m$. Next, during each LGC iteration (Eqn. 4.3), we use the labels assigned by $k$-means to determine, for each point $x \in D$, how homogeneous its neighborhood is, i.e., how many points share the same clustering label as $x$. For this, we compute the $Q_h$ metric (Tab. 4.2) but now using the pseudolabels assigned by $k$-means rather than ground truth labels. Finally, we modify the LGC sharpening (Eqn. 4.3) to use a local learning rate ($\alpha Q_h(x)$) instead of the global learning rate $\alpha$. This decreases $\alpha$ in regions where $Q_h$ is small, i.e., $k$-means finds that two data clusters are close to each other, such as the middle part of the density plot in Fig. 4.2b. Points there move far less when applying Eqn. 4.3, yielding the desired LGC result in Fig. 4.2d. In areas where the density is very low (around the middle of Fig. 4.2a), $Q_h$ is higher, so points there move fast, like in the original LGC, yielding the desired result in Fig. 4.2b. We show next in Sec. 4.4.2 that this allows us to control only the cluster count $K_m$ of $k$-means to obtain more stable, and higher quality results, than when controlling $\alpha$, $k_s$, and $t$ in SDR-NNP.

4.4 results

We measured the performance of SDR-NNP and $\alpha$-SDR-NNP by the four metrics in Tab. 4.2 computed for $k = 7$, in line with [63, 141, 211]. Note that $k$, the number of nearest neighbors used to compute the metrics in Tab. 4.2, is smaller than $k_s$, the number of nearest neighbors used to evaluate $\rho$ (Eqn. 4.2). Indeed, $k_s$ needs to be relatively large to smooth out local noise in the computation of the gradient $\nabla \rho$; in contrast, $k$ is typically set small to capture more local quality aspects of a projection.

Evaluation used six publicly available real-world data sets (Tab. 4.4), all being reasonably high-dimensional and large (tens of dimensions, thousands of samples), and with a non-trivial data structure. All dimensions were rescaled to the $[0, 1]$ range, to match NNP’s sigmoid activation func-
4.4 RESULTS

<table>
<thead>
<tr>
<th>Data set name and provenance</th>
<th>N</th>
<th>n</th>
<th>Data set description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Air Quality [219]</td>
<td>9358</td>
<td>13</td>
<td>Measurements from air sensors used to study and predict air quality</td>
</tr>
<tr>
<td>Concrete [233]</td>
<td>1030</td>
<td>8</td>
<td>Measurements of chemico-physical properties of concrete used to study concrete strength</td>
</tr>
<tr>
<td>Reuters [196]</td>
<td>5000</td>
<td>100</td>
<td>Attributes extracted from news report documents using TF-IDF [176], a standard method in text processing. This is a subset of the full data set which contains data for the six most frequent classes only. Used to study how features can predict news’ types (classes)</td>
</tr>
<tr>
<td>Spambase [97]</td>
<td>4001</td>
<td>57</td>
<td>Text data set used to train email spam classifiers</td>
</tr>
<tr>
<td>Wisconsin [189]</td>
<td>569</td>
<td>32</td>
<td>Features extracted from images of breast masses used to detect malignant cells</td>
</tr>
<tr>
<td>Wine [45]</td>
<td>6497</td>
<td>11</td>
<td>Samples of white and red Portuguese <em>vinho verde</em> used to describe perceived wine quality</td>
</tr>
</tbody>
</table>

Section 4.4.1 details the quality of SDR-NNP. Section 4.4.2 presents the evaluation of $\alpha$-SDR-NNP as compared to SDR-NNP. Section 4.4.3 studies the computational scalability of both methods. Finally, Section 4.4.4 presents an application of SDR-NNP to the analysis of astronomical data.

4.4.1 Evaluation of SDR-NNP

We first studied SDR-NNP’s quality with respect to its parameters (number of iterations $t$, learning rate $\alpha$, training epochs $e$) using Landmark MDS (LMDS), $t$-SNE, and PCA as baseline DR methods. A discussion on the selection of DR methods for SDR can be found in Sec. 6 from [112]. All results here and in Sec. 4.4.2 use a medium size for the NNP network. Results computed for other network sizes look very similar and are provided in the supplementary material.

**Number of iterations $t$:** Figure 4.3 shows how $t$ affects the sharpening of clusters for LMDS and $t$-SNE (PCA results in supplementary material). For
Figure 4.3: Iteration parameter t effect: SDR-NNP learned from LMDS (a) and t-SNE (b) for varying t values (columns) and data sets (rows), fixed 𝛼 = 0.1, e = 1000 epochs.
Figure 4.4: Learning rate $\alpha$ effect: SDR-NNP learned from LMDS (a) and $t$-SNE (b) for varying $\alpha$ values (columns) and data sets (rows), fixed $t = 10$ iterations, $\epsilon = 1000$ epochs.
Figure 4.5: Training epochs effect SDR-NNP learned from LMDS (a) and t-SNE (b) for varying \( \epsilon \) values (columns) and data sets (rows), fixed \( r = 10, \alpha = 0.1 \). Red column shows the training projections \( P(D) \).
all data sets, 4 to 8 iterations suffice to have the clusters sharply defined in the projection. Table 4.5 shows quality metrics as functions of \( t \) for all three baseline projections. Increasing \( t \) can increase quality (Air Quality, Reuters with LMDS and PCA) but generally slightly decreases quality for LMDS and PCA. For \( t \)-SNE, this decrease is visible for all data sets, which is explainable by the fact that \( t \)-SNE already has a very high quality which is hard to be learned by NNP (see [60]). However, as already argued in [112], local quality metrics will likely decrease when using SDR to favor visual cluster separation.

**Learning rate** \( \alpha \): Figure 4.4 shows results for SDR-NNP when varying \( \alpha \) for LMDS and \( t \)-SNE (PCA results in supplementary material). Too small or too large \( \alpha \) values tend to affect the projection adversely. Values in the range \( \alpha \in [0.05, 0.1] \) show the best results, i.e., a good separation of the projection into distinct clusters. Table 4.6 shows quality metrics as function of \( \alpha \) for all three baseline projections. The effect of \( \alpha \) on quality is similar with that of \( t \) with some combinations (Reuters with LMDS and PCA) showing an overall slight decrease for small \( \alpha \) values.

**Training epochs** \( e \): Figure 4.5 shows how \( e \) affects projection quality. The early stopping strategy used by NNP [60] – stopping training on convergence, defined as the epoch where the validation loss stops decreasing (roughly \( e = 60 \) in practice) – does not give good results for SDR-NNP. The resulting projections (Fig. 4.5a,b leftmost columns) show a fuzzy version of the training projections (Fig. 4.5a,b rightmost columns). This is due to the fact that SDR-NNP needs to learn both the LGC data sharpening and the projection \( P \), which is more effort than learning just \( P \), as NNP did. For more training epochs, Fig. 4.5 shows that SDR-NNP reproduces the training projection very faithfully. SDR-NNP produces good results with as little as \( e = 300 \) epochs, except for the Air Quality data set, where \( e = 3000 \) epochs were needed for best results. On average, \( e = 1000 \) epochs led to good results for all data sets and other parameter settings, so we choose this as a **preset** value for \( e \). We keep this preset also for our new method \( \alpha \)-SDR-NNP in the latter’s evaluation (Sec. 4.4.2).

**Cluster separation:** The projections in Figs. 4.3–4.5 deserve some comments. As visible there, varying the \( t \) and \( \alpha \) parameters can create artificial *oversegmentation* – the appearance of many small clusters in the projection, which is an artificial cluster separation (CS), see e.g. Fig. 4.4b, Reuters, \( \alpha \geq 0.1 \). This effect is strongest, and undesirable, for baseline projections which already do have a good CS, such as \( t \)-SNE. In contrast, for projections with a low CS, such as LMDS, artificial oversegmentation is far less
### Table 4.5: Metrics for SDR-NNP learned from LMDS, PCA, and t-SNE, different numbers of iterations $t$, $\alpha = 0.1$, $e = 1000$. $Q_h$ values miss for the Air Quality and Concrete data sets since these are not labeled.

<table>
<thead>
<tr>
<th>Data set</th>
<th>LMDS</th>
<th>PCA</th>
<th>t-SNE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$t$</td>
<td>$Q_1$</td>
<td>$Q_2$</td>
</tr>
<tr>
<td>Air Quality</td>
<td>0</td>
<td>0.941</td>
<td>0.972</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>0.962</td>
<td>0.979</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>0.954</td>
<td>0.970</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>0.949</td>
<td>0.970</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>0.950</td>
<td>0.968</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0.954</td>
<td>0.968</td>
</tr>
<tr>
<td>Concrete</td>
<td>0</td>
<td>0.940</td>
<td>0.979</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>0.938</td>
<td>0.958</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>0.912</td>
<td>0.944</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>0.895</td>
<td>0.941</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>0.884</td>
<td>0.938</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0.876</td>
<td>0.935</td>
</tr>
<tr>
<td>Reuters</td>
<td>0</td>
<td>0.817</td>
<td>0.895</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>0.835</td>
<td>0.913</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>0.858</td>
<td>0.915</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>0.883</td>
<td>0.909</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>0.884</td>
<td>0.910</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0.882</td>
<td>0.907</td>
</tr>
<tr>
<td>Spambase</td>
<td>0</td>
<td>0.740</td>
<td>0.909</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>0.737</td>
<td>0.881</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>0.723</td>
<td>0.855</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>0.711</td>
<td>0.845</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>0.701</td>
<td>0.837</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0.710</td>
<td>0.840</td>
</tr>
<tr>
<td>Wisconsin</td>
<td>0</td>
<td>0.895</td>
<td>0.959</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>0.892</td>
<td>0.915</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>0.804</td>
<td>0.857</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>0.790</td>
<td>0.849</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>0.780</td>
<td>0.847</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0.775</td>
<td>0.842</td>
</tr>
<tr>
<td>Wine</td>
<td>0</td>
<td>0.864</td>
<td>0.973</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>0.867</td>
<td>0.932</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>0.843</td>
<td>0.916</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>0.840</td>
<td>0.904</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>0.845</td>
<td>0.901</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0.842</td>
<td>0.899</td>
</tr>
</tbody>
</table>
Table 4.6: Metrics for SDR-NNP learned from LMDS, PCA, and t-SNE, learning rates $\alpha$, $t = 10$ iterations, $e = 1000$ epochs. $Q_h$ values miss for the Air Quality and Concrete data sets since these are not labeled.

<table>
<thead>
<tr>
<th>Data set</th>
<th>$\alpha$</th>
<th>LMDS</th>
<th>PCA</th>
<th>t-SNE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$Q_1$</td>
<td>$Q_2$</td>
<td>$Q_3$</td>
<td>$Q_4$</td>
</tr>
<tr>
<td>Air Quality</td>
<td>0.01</td>
<td>0.971</td>
<td>0.990</td>
<td>0.969</td>
</tr>
<tr>
<td></td>
<td>0.05</td>
<td>0.976</td>
<td>0.983</td>
<td>0.963</td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td>0.951</td>
<td>0.969</td>
<td>0.948</td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td>0.866</td>
<td>0.941</td>
<td>0.911</td>
</tr>
<tr>
<td>Concrete</td>
<td>0.01</td>
<td>0.959</td>
<td>0.983</td>
<td>0.731</td>
</tr>
<tr>
<td></td>
<td>0.05</td>
<td>0.932</td>
<td>0.957</td>
<td>0.601</td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td>0.870</td>
<td>0.933</td>
<td>0.578</td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td>0.858</td>
<td>0.920</td>
<td>0.540</td>
</tr>
<tr>
<td>Reuters</td>
<td>0.01</td>
<td>0.822</td>
<td>0.900</td>
<td>0.758</td>
</tr>
<tr>
<td></td>
<td>0.05</td>
<td>0.839</td>
<td>0.913</td>
<td>0.737</td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td>0.870</td>
<td>0.910</td>
<td>0.698</td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td>0.866</td>
<td>0.902</td>
<td>0.700</td>
</tr>
<tr>
<td>Spambase</td>
<td>0.01</td>
<td>0.755</td>
<td>0.911</td>
<td>0.527</td>
</tr>
<tr>
<td></td>
<td>0.05</td>
<td>0.775</td>
<td>0.893</td>
<td>0.415</td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td>0.712</td>
<td>0.843</td>
<td>0.380</td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td>0.604</td>
<td>0.667</td>
<td>0.265</td>
</tr>
<tr>
<td>Wisconsin</td>
<td>0.01</td>
<td>0.900</td>
<td>0.960</td>
<td>0.932</td>
</tr>
<tr>
<td></td>
<td>0.05</td>
<td>0.868</td>
<td>0.885</td>
<td>0.869</td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td>0.803</td>
<td>0.856</td>
<td>0.757</td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td>0.717</td>
<td>0.764</td>
<td>0.693</td>
</tr>
<tr>
<td>Wine</td>
<td>0.01</td>
<td>0.895</td>
<td>0.972</td>
<td>0.811</td>
</tr>
<tr>
<td></td>
<td>0.05</td>
<td>0.914</td>
<td>0.944</td>
<td>0.734</td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td>0.837</td>
<td>0.913</td>
<td>0.672</td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td>0.739</td>
<td>0.821</td>
<td>0.479</td>
</tr>
</tbody>
</table>
present. Like SDR, SDR-NNP is best used when combined with baseline DR methods with a low CS capability. We show next in Sec. 4.4.2 how our new method, $\alpha$-SDR-NNP, largely removes all these parameter setting issues.

### 4.4.2 Evaluation of $\alpha$-SDR-NNP

Section 4.4.1 showed how SDR-NNP improves cluster separation as compared to the baseline DR projection it builds atop of. However, this evaluation also showed that, to get good results, we still have to fine tune the SDR parameters $t$ and $\alpha$. The effect of the parameter $k_s$ was not explored. The only parameter which showed to have a good preset is $e = 1000$ training epochs.

Our new method, $\alpha$-SDR-NNP, has all these parameters of SDR-NNP plus an additional one, the number of $k$-means clusters $K_m$ (Sec. 4.3.3). Still, we argue that $\alpha$-SDR-NNP is easier to use than SDR-NNP, and support this point by three evaluations on two labeled data sets (Reuters and Wine), as follows.

**Visual comparison:** We first run $\alpha$-SDR-NNP and SDR-NNP for various combinations of their free parameters $t$, $\alpha$, and $k_s$, setting $K_m$ to the true number of clusters in the evaluated data sets (Reuters: $K_m = 6$; Wine: $K_m = 7$). For this, we take the parameter ranges found to deliver good results from SDR-NNP’s evaluation (Sec. 4.4.1) and sample each range with five values, leading to the values $t \in \{1, 5, 8, 12, 20\}$ iterations, $k_s \in \{15, 30, 50, 80, 100\}$ neighbors, and $\alpha \in \{0.01, 0.05, 0.1, 0.2, 0.4\}$ learning rates. For each combination of parameter values, we compute the projections of SDR-NNP and $\alpha$-SDR-NNP for the Wine and Reuters data sets.

Visually comparing all these $5 \times 5 \times 5 \times 2 = 250$ projection-pairs is not practical. Hence, we next set each of the three parameters to its median value in its sample-set and visually compare the results for the $5 \times 5$ combinations of the other two free parameters, see Figs. 4.6 to 4.11. Since all these figures are structured similarly, we only explain how to interpret the first one (the others can be interpreted similarly): Figure 4.6 show the SDR-NNP and $\alpha$-SDR-NNP projections for varying $k_s$ and $\alpha$ for a fixed value of $t = 8$ iterations for the Reuters data set. We directly see that the SDR-NNP projections (top) change considerably more than the $\alpha$-SDR-NNP projections (bottom). In particular, SDR-NNP’s results become increasingly fuzzy with dropping visual cluster separation for higher $\alpha$ values. A very similar effect is visible when varying $t$ and $\alpha$ (Fig. 4.7) and $t$ and $k_s$, respectively (Fig. 4.8). In contrast, $\alpha$-SDR-NNP shows the same vi-
visual cluster separation for all parameter combinations. For the Wine data set (Figs. 4.9-4.11), this fuzzy effect is less visible. Yet, the visual cluster separation of the SDR-NNP projections varies quite a lot, while \( \alpha \)-SDR-NNP generates more stable results for the different parameter values. We conclude that, in practice, users can ignore fine-tuning \( k_s \) and \( \alpha \) for \( \alpha \)-SDR-NNP, and simply use the median values in their respective sample-sets as default settings.

**Quality metrics comparison:** Table 4.7 compares the four projection quality metrics (Tab. 4.2) of SDR-NNP vs \( \alpha \)-SDR-NNP for all the experiments shown earlier in Figs. 4.6-4.11. Each table shows the effect of varying one of the three parameters \( t, k_s \), and \( \alpha \). For such a parameter value, the shown metrics are aggregates of the 25 combinations of values of the other two parameters – for example, the first row of Tab. 4.7, column \( Q_t \), shows the average value of trustworthiness computed for all 25 value combinations of \( k_s \) and \( \alpha \) and for \( t = 1 \). From this table, we see that \( \alpha \)-SDR-NNP achieves similar but often higher values of quality metrics, the increase being as large as 9%. We also see that, in general, the standard deviation values are smaller for \( \alpha \)-SDR-NNP than for SDR-NNP. This means that our new method achieves its quality metrics more consistently – or, in other words, that these values are less susceptible to change when one varies the three parameters, which is desirable. We note that an increase of several percentage points in projection quality is significant. Recent surveys [63] showed that top-quality projection methods in the entire DR literature of the last decades differ by as few as 2 to 5 percentage points. NNP, the method that we use to drive our own \( \alpha \)-SDR-NNP technique, had a quality of a few percentage points lower than the state-of-the-art projection methods it tries to imitate, most notably t-SNE [60].
Figure 4.6: **SDR-NNP** (top) *vs* **α-SDR-NNP** (bottom), Reuters data set. Fixed $t = 8$ iterations, varying number of neighbors $k_s$ and learning rate $\alpha$. 

**Note:** The images show the visualization of data points for different values of $k_s$ and $\alpha$. The data points are color-coded to represent different classes, and the layout illustrates the clustering effect of the dimensionality reduction algorithms.
Figure 4.7: SDR-NNP (top) vs $\alpha$-SDR-NNP (bottom), Reuters data set. Fixed $k_s = 50$ neighbors, varying number of iterations $t$ and learning rate $\alpha$. 
Figure 4.8: SDR-NNP (top) vs $\alpha$-SDR-NNP (bottom), Reuters data set. Fixed $\alpha = 0.1$ learning rate, varying number of iterations $t$ and neighbors $k_s$. 
Figure 4.9: **SDR-NNP** (top) vs **α-SDR-NNP** (bottom), Wine data set. Fixed $t = 8$ iterations, varying number of neighbors $k_s$ and learning rate $\alpha$. 
Figure 4.10: SDR-NNP (top) vs $\alpha$-SDR-NNP (bottom), Wine data set. Fixed $k_s = 50$ neighbors, varying number of iterations $t$ and learning rate $\alpha$. 

Figure 4.10: SDR-NNP (top) vs $\alpha$-SDR-NNP (bottom), Wine data set. Fixed $k_s = 50$ neighbors, varying number of iterations $t$ and learning rate $\alpha$. 

Figure 4.10: SDR-NNP (top) vs $\alpha$-SDR-NNP (bottom), Wine data set. Fixed $k_s = 50$ neighbors, varying number of iterations $t$ and learning rate $\alpha$. 

Figure 4.10: SDR-NNP (top) vs $\alpha$-SDR-NNP (bottom), Wine data set. Fixed $k_s = 50$ neighbors, varying number of iterations $t$ and learning rate $\alpha$.
Figure 4.11: SDR-NNP (top) vs $\alpha$-SDR-NNP (bottom), Wine data set. Fixed $\alpha = 0.1$ learning rate, varying number of iterations $t$ and neighbors $k_s$. 
Table 4.7: Quality metrics comparing SDR-NNP with $\alpha$-SDR-NNP.

a) Number of iterations $t$ (average)

<table>
<thead>
<tr>
<th>Data set</th>
<th>SDR-NNP</th>
<th>$\alpha$-SDR-NNP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$t_Q$</td>
<td>$Q_c$</td>
</tr>
<tr>
<td>wine</td>
<td>1</td>
<td>0.981</td>
</tr>
<tr>
<td>wine</td>
<td>5</td>
<td>0.959</td>
</tr>
<tr>
<td>wine</td>
<td>8</td>
<td>0.952</td>
</tr>
<tr>
<td>wine</td>
<td>12</td>
<td>0.944</td>
</tr>
<tr>
<td>wine</td>
<td>20</td>
<td>0.936</td>
</tr>
<tr>
<td>reuters</td>
<td>1</td>
<td>0.967</td>
</tr>
<tr>
<td>reuters</td>
<td>5</td>
<td>0.946</td>
</tr>
<tr>
<td>reuters</td>
<td>8</td>
<td>0.927</td>
</tr>
<tr>
<td>reuters</td>
<td>12</td>
<td>0.908</td>
</tr>
<tr>
<td>reuters</td>
<td>20</td>
<td>0.881</td>
</tr>
</tbody>
</table>

b) Number of iterations $t$ (standard deviation)

<table>
<thead>
<tr>
<th>Data set</th>
<th>SDR-NNP</th>
<th>$\alpha$-SDR-NNP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$t_Q$</td>
<td>$Q_c$</td>
</tr>
<tr>
<td>wine</td>
<td>1</td>
<td>0.021</td>
</tr>
<tr>
<td>wine</td>
<td>5</td>
<td>0.035</td>
</tr>
<tr>
<td>wine</td>
<td>8</td>
<td>0.037</td>
</tr>
<tr>
<td>wine</td>
<td>12</td>
<td>0.038</td>
</tr>
<tr>
<td>wine</td>
<td>20</td>
<td>0.039</td>
</tr>
<tr>
<td>reuters</td>
<td>1</td>
<td>0.001</td>
</tr>
<tr>
<td>reuters</td>
<td>5</td>
<td>0.034</td>
</tr>
<tr>
<td>reuters</td>
<td>8</td>
<td>0.055</td>
</tr>
<tr>
<td>reuters</td>
<td>12</td>
<td>0.066</td>
</tr>
<tr>
<td>reuters</td>
<td>20</td>
<td>0.076</td>
</tr>
</tbody>
</table>

c) Number of neighbors $k_s$ (average)

<table>
<thead>
<tr>
<th>Data set</th>
<th>SDR-NNP</th>
<th>$\alpha$-SDR-NNP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$k_s$</td>
<td>$Q_t$</td>
</tr>
<tr>
<td>wine</td>
<td>15</td>
<td>0.948</td>
</tr>
<tr>
<td>wine</td>
<td>30</td>
<td>0.954</td>
</tr>
<tr>
<td>wine</td>
<td>50</td>
<td>0.956</td>
</tr>
<tr>
<td>wine</td>
<td>80</td>
<td>0.957</td>
</tr>
<tr>
<td>wine</td>
<td>100</td>
<td>0.957</td>
</tr>
<tr>
<td>reuters</td>
<td>15</td>
<td>0.923</td>
</tr>
<tr>
<td>reuters</td>
<td>30</td>
<td>0.925</td>
</tr>
<tr>
<td>reuters</td>
<td>50</td>
<td>0.926</td>
</tr>
<tr>
<td>reuters</td>
<td>80</td>
<td>0.927</td>
</tr>
<tr>
<td>reuters</td>
<td>100</td>
<td>0.928</td>
</tr>
</tbody>
</table>
### 4.4 Results

**d) Number of neighbors $k_s$ (standard deviation)**

<table>
<thead>
<tr>
<th>Data set</th>
<th>SDR-NNP</th>
<th>$\alpha$-SDR-NNP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$k_s$</td>
<td>$Q_l$</td>
</tr>
<tr>
<td>wine</td>
<td>15</td>
<td>0.044</td>
</tr>
<tr>
<td>wine</td>
<td>30</td>
<td>0.038</td>
</tr>
<tr>
<td>wine</td>
<td>50</td>
<td>0.036</td>
</tr>
<tr>
<td>wine</td>
<td>80</td>
<td>0.035</td>
</tr>
<tr>
<td>wine</td>
<td>100</td>
<td>0.035</td>
</tr>
<tr>
<td>reuters</td>
<td>15</td>
<td>0.065</td>
</tr>
<tr>
<td>reuters</td>
<td>30</td>
<td>0.062</td>
</tr>
<tr>
<td>reuters</td>
<td>50</td>
<td>0.061</td>
</tr>
<tr>
<td>reuters</td>
<td>80</td>
<td>0.060</td>
</tr>
<tr>
<td>reuters</td>
<td>100</td>
<td>0.058</td>
</tr>
</tbody>
</table>

**e) Learning rate $\alpha$ (average)**

<table>
<thead>
<tr>
<th>Data set</th>
<th>SDR-NNP</th>
<th>$\alpha$-SDR-NNP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\alpha$</td>
<td>$Q_l$</td>
</tr>
<tr>
<td>wine</td>
<td>1</td>
<td>0.981</td>
</tr>
<tr>
<td>wine</td>
<td>5</td>
<td>0.959</td>
</tr>
<tr>
<td>wine</td>
<td>8</td>
<td>0.952</td>
</tr>
<tr>
<td>wine</td>
<td>12</td>
<td>0.944</td>
</tr>
<tr>
<td>wine</td>
<td>20</td>
<td>0.936</td>
</tr>
<tr>
<td>reuters</td>
<td>1</td>
<td>0.967</td>
</tr>
<tr>
<td>reuters</td>
<td>5</td>
<td>0.946</td>
</tr>
<tr>
<td>reuters</td>
<td>8</td>
<td>0.927</td>
</tr>
<tr>
<td>reuters</td>
<td>12</td>
<td>0.908</td>
</tr>
<tr>
<td>reuters</td>
<td>20</td>
<td>0.881</td>
</tr>
</tbody>
</table>

**f) Learning rate $\alpha$ (standard deviation)**

<table>
<thead>
<tr>
<th>Data set</th>
<th>SDR-NNP</th>
<th>$\alpha$-SDR-NNP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\alpha$</td>
<td>$Q_l$</td>
</tr>
<tr>
<td>wine</td>
<td>0.01</td>
<td>0.001</td>
</tr>
<tr>
<td>wine</td>
<td>0.05</td>
<td>0.011</td>
</tr>
<tr>
<td>wine</td>
<td>0.1</td>
<td>0.020</td>
</tr>
<tr>
<td>wine</td>
<td>0.2</td>
<td>0.026</td>
</tr>
<tr>
<td>wine</td>
<td>0.4</td>
<td>0.023</td>
</tr>
<tr>
<td>reuters</td>
<td>0.01</td>
<td>0.001</td>
</tr>
<tr>
<td>reuters</td>
<td>0.05</td>
<td>0.006</td>
</tr>
<tr>
<td>reuters</td>
<td>0.1</td>
<td>0.027</td>
</tr>
<tr>
<td>reuters</td>
<td>0.2</td>
<td>0.054</td>
</tr>
<tr>
<td>reuters</td>
<td>0.4</td>
<td>0.070</td>
</tr>
</tbody>
</table>
Moreover, as also mentioned in Sec. 4.1, our key goal with this work was not to increase absolute quality of the obtained projection, but increase the stability and ease of computing a good-quality projection without having to fiddle with the four parameters of its predecessor method, SDR-NNP. The fact we also obtained higher quality, along with the desired ease-of-use, is an extra bonus point for our method.

Figure 4.12: $\alpha$-SDR-NNP for different numbers of $k$-means clusters $K_m$. Columns show results when setting $K_m$ to half, equal to, and double the true number of clusters (6 for Reuters and 7 for Wine).

Figure 4.13: MDS projection of projections computed for all experiments colored by method (SDR-NNP: green, $\alpha$-SDR-NNP: purple), Reuters and Wine data sets.
4.4 Results

Effect of $K_m$: The results so far show that $\alpha$-SDR-NNP is stable with respect to the original three parameters $t$, $\alpha$, and $k_s$ of SDR-NNP. However, $\alpha$-SDR-NNP introduces one new parameter, the number of clusters $K_m$ for $k$-means. To study how stable our method is for this new parameter, we ran it for varying values of $K_m$. Specifically, for the Reuters and Wine data sets, we set $K_m$ to be half, equal to, and double the true number of clusters known to exist in these data sets. Figure 4.12 shows that the resulting projections are very similar in terms of visual clusters being produced. This means that our method is not sensitive to setting $K_m$, unlike SDR-NNP’s sensitivity to setting $t$, $\alpha$, and $k_s$.

Putting it all together: Figure 4.13 shows a so-called ‘projection of projections’ [63] for the Reuters and Wine data set. Every point in such a scatterplot is a given projection technique. Green points are SDR-NNP and purple points are $\alpha$-SDR-NNP. The different same-color points represent instances of the respective technique for the different values of $t$, $k_s$, and $\alpha$ discussed in the above evaluations. Points are projected to 2D using MDS based on the values of their four quality metrics. Points in the projection which are close indicate methods which perform similarly quality-wise. For both data sets, we see a high concentration of purple points in a tail-like structure, while the green points are far more spread around. This indicates that $\alpha$-SDR-NNP generates more consistent (similar) quality values than SDR-NNP, thus, is less sensitive in this respect to parameter changes. This strengthens our claim that $\alpha$-SDR-NNP allows users to generate good projections with less parameter tweaking than SDR-NNP.

Table 4.8: Time measurements for SDR and ($\alpha$-)SDR-NNP in seconds, GALAH data set. See also Fig. 4.14.

<table>
<thead>
<tr>
<th>Samples</th>
<th>SDR</th>
<th>($\alpha$-)SDR-NNP inference</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>0.220</td>
<td>0.108</td>
</tr>
<tr>
<td>2000</td>
<td>0.957</td>
<td>0.059</td>
</tr>
<tr>
<td>5000</td>
<td>14.799</td>
<td>0.092</td>
</tr>
<tr>
<td>10000</td>
<td>157.420</td>
<td>0.149</td>
</tr>
<tr>
<td>20000</td>
<td>1302.268</td>
<td>0.414</td>
</tr>
<tr>
<td>30000</td>
<td>4736.995</td>
<td>0.355</td>
</tr>
<tr>
<td>40000</td>
<td>11058.267</td>
<td>0.727</td>
</tr>
</tbody>
</table>

4.4.3 Computational Scalability

We measured scalability by comparing the execution time of the original SDR method with SDR-NNP using samples from the GALAH data set (described next in Sec. 4.4.4) with increasing sizes, namely, 1K, 2K, 5K, 10K,
20K, 30K, and 40K samples. Using more samples was not needed since SDR already took over three hours at 40K samples. Figure 4.14 and Table 4.8 show these results. For $|D_s| = 10K$ training samples and $e = 1000$ epochs, SDR-NNP takes about 373 seconds to train (Fig. 4.14, orange line). Still, this is already faster than SDR for 15K samples. In inference mode (after training), SDR-NNP is orders of magnitude faster than SDR, taking less than one second to project 40K samples (Fig. 4.14, green curve). SDR takes over three hours for the same data size (Fig. 4.14, blue curve). For $\alpha$-SDR-NNP, training time is slightly higher than for SDR-NNP due to the cost of $k$-means clustering and $Q_h$ computation (Fig. 4.1, green steps), but inference time is identical to SDR-NNP.

4.4.4 Case Study: Astronomical Data sets

We used SDR-NNP for a use-case using real-world astronomical data – the same subset of 10K samples from the GALactic Archaeology with HERMES survey (GALAH DR2) [25] used by Kim et al. to show that SDR-NNP can create similar projections to SDR. The GALAH DR2 data set consists of various stellar abundance attributes of 342682 stars. Data cleaning fol-
4.4 RESULTS

allowed [112]: (1) cross-match the star ID of GALAH DR2 with Gaia data release 2 (Gaia DR2) to gain extra information on stellar kinematics (i.e., 6D phase-space coordinates—x, y, z, u, v, and w) [25, 74, 75]; (2) exclude stars with implausible values (exceeding 25K parsec in x, y, and z attributes), having unreliable stellar abundances, or with missing values in any dimension. Preprocessing delivered 76270 stars (samples) from which we took the same randomly selected subset D of 10K stars as in [112] to run SDR with the same \( \alpha = 0.18 \) (Sec. 4.3.1). We trained SDR-NNP on these 10K stars and used the trained SDR-NNP network to project the remaining 66270 stars.

Figure 4.15 shows SDR-NNP applied to the 66K test data with LMDS and t-SNE as baseline DR methods. Points are colored based on the value of the attribute [Fe/H], which is of interest to domain experts to explain possible data clusters. The first four columns show SDR-NNP for varying training epoch counts \( e \). The red column shows the training projection \( P(D_s) \) of 10K samples. We see that the structure of the training projection (four clusters) is well reflected by SDR-NNP from \( e = 300 \) epochs onwards. The test projections are more fuzzy. This is expected, as these contain 66K unseen samples which, albeit drawn from the same data set, cannot perfectly match the four clusters determined by the 10K training samples. The rightmost column in Fig. 4.15 shows the result of the ‘raw’ NNP method, i.e., trained to imitate LMDS, and t-SNE, without the sharpening step of SDR, respectively. These results show clearly far less cluster separation (CS) than either the SDR-NNP training projection (red column) or the inferred SDR-NNP projections (leftmost four columns). This shows the added value of the sharpening step: Without it, NNP, albeit fast and OOS-capable, cannot produce useful projections. Table 4.9 shows quality metrics corresponding to the images in Fig. 4.15 which support the above observations.

SDR-NNP’s good cluster separation allows astronomers to easily label clusters for further analysis to infer the physical meaning of stars. To show this, we manually labeled clusters from SDR-NNP learned from LMDS to reproduce the same analyses made by Kim et al. (Fig. 10 in [112]) to understand the origin and location of stars in each cluster. Figure 4.16a shows the manually labeled clusters by one of the authors (astronomy expert). Stars from class 5 are separately labeled as outliers. Figures 4.16b,c are the Tinsley diagram [200] and the copper abundance of the stars— a tracer of supernovae type 1a— as a function of their iron abundance, respectively. From these plots, astronomers can identify class-1 stars as thin-disk stars, class-2 stars as metal-rich thick disk stars, class-3 and class-5 (outlier) stars
Figure 4.15: SDR-NNP of 66K samples learned from LMDS and t-SNE instead of SDR applied to the same test data. Rightmost column: NNP trained with LMDS and t-SNE applied to the same test data (10K samples). Red column: training projection (10K samples). Leftmost column: NNP trained with LMDS and t-SNE instead of SDR applied to the same test data (10K samples). Parameters are $t = 10$ iterations, and $\alpha = 0.18$. Red column: training projection (10K samples) for different numbers of training epochs.

SDR-NNP parameters are $t = 10$ iterations, and $\alpha = 0.18$. Red column: training projection (10K samples). For different numbers of training epochs.
Figure 4.16: Analysis of GALAH DR2 with SDR-NNP learned from LMDS. (a) Labeling of clusters (classes 1–4) and outliers (class 5). (b) Tinsley diagram and (c) copper abundance of stars vs their iron abundance. Astronomers can infer from (b,c) that class 1 is mostly thin disk stars, class 2 is mostly metal-rich thick disk stars, classes 5 and 3 are normal thick disk stars, and class 4 is the Gaia Enceladus (GES) in the Milky Way.
Table 4.9: Metrics for SDR-NNP learned from LMDS, PCA, and t-SNE on the GALAH data set for train and test samples, varying number of training epochs \( e \) (‘early’ indicates the early-stopping heuristic).

<table>
<thead>
<tr>
<th>Mode</th>
<th>LMDS</th>
<th>PCA</th>
<th>t-SNE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( Q_0 )</td>
<td>( Q_r )</td>
<td>( Q_t )</td>
</tr>
<tr>
<td>Train</td>
<td>early</td>
<td>0.802</td>
<td>0.877</td>
</tr>
<tr>
<td></td>
<td>300</td>
<td>0.703</td>
<td>0.790</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>0.695</td>
<td>0.772</td>
</tr>
<tr>
<td></td>
<td>3000</td>
<td>0.692</td>
<td>0.756</td>
</tr>
<tr>
<td>Test</td>
<td>early</td>
<td>0.775</td>
<td>0.862</td>
</tr>
<tr>
<td></td>
<td>300</td>
<td>0.667</td>
<td>0.768</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>0.667</td>
<td>0.757</td>
</tr>
<tr>
<td></td>
<td>3000</td>
<td>0.657</td>
<td>0.737</td>
</tr>
</tbody>
</table>

as the normal thick disk stars, and class-4 stars as Gaia Enceladus (GES) – a group of stars that originated from a galaxy that merged with the Milky Way several billions years ago. The original SDR method could not do this analysis and find class-4 stars since it could run on the entire data set due to its prohibitively low speed.

4.5 Discussion

We discuss how SDR-NNP and our new extension, \( \alpha \)-SDR-NNP, perform with respect to the criteria laid out in Sec. 4.1.

**Quality (C1):** SDR-NNP can create projections which are very similar visually, but also in terms of quality metrics, to those created by SDR. Importantly, the strong separation of similar-valued samples, the key property that SDR promoted, is retained by SDR-NNP. Combined with properties C2–C4 (which SDR does not have), this makes SDR-NNP superior to SDR. Compared to NNP used on the unsharpened data (Fig. 4.15), SDR-NNP shows significantly better cluster separation, which makes it superior to NNP. Atop all these, \( \alpha \)-SDR-NNP shows, for most parameter values, higher quality metrics – thus better projections – than SDR-NNP.

**Scalability (C2):** \( \alpha \)-SDR-NNP is faster than SDR alone from roughly 15K samples onwards, even when considering training time. In inference mode (after training), \( \alpha \)-SDR-NNP is several orders of magnitude faster than SDR, being able to project tens of thousands of observations in under a second on a high-end PC. Importantly, \( \alpha \)-SDR-NNP’s speed is linear in the number of dimensions and samples (a property inherited from the NNP architecture), and can handle samples in a streaming fashion, one at
a time, i.e., does not need to hold the entire high-dimensional data set in memory. This makes (α-)SDR-NNP scalable to large data sets of millions of samples. Compared to SDR-NNP, our new method α-SDR-NNP is slightly slower for training but has identical inference speed.

We evaluated our method on relatively small data sets – up to 10K samples (see Tab. 4.4). Of course, larger high-dimensional data sets exist. However, we preferred to use these data sets since they are well-known in the visualization and machine learning communities, as part of multiple benchmarks and many papers. As such, readers can directly compare our projection results for these data sets with other results in the literature. Moreover, we note that using larger data sets will only help our method. Computation speed of our projections is linear in the sample count, as stated above, which is the optimal result one can get. Using larger data sets implies having higher sample densities, so estimating this density only increases in accuracy (Eqn. 4.2). As such, using smaller data sets is actually a bigger challenge for our method.

**Genercity (C3):** (α-)SDR-NNP can project any data set having quantitative variables and any dimension count. Tables 4.5, 4.6, 4.9, and 4.7 show that (α-)SDR-NNP achieves high quality on data sets of different nature and coming from a wide range of application domains (air sensors, civil engineering, text mining, imaging, and chemistry).

**Stability and out-of-sample support (C4):** (α-)SDR-NNP inherits the stability and OOS support of NNP, making it possible to train on a small subset of a given data set and then stably project additional data drawn from the same distribution. Moreover, our new method, α-SDR-NNP, is stable with respect to its single free parameter $K_m$, which makes it by construction a good method to sharpen-and-project high-dimensional data. Simply put, α-SDR-NNP delivers similar visual results for a given data set and any settings of $K_m$, which means in practice that users can benefit from a sharpened projection without worrying about how they set the parameter controlling the computation of this projection.

**Ease-of-use (C5):** Once trained, (α-)SDR-NNP is parameter-free. SDR-NNP has three relevant parameters affecting its preprocessing LGC step – number of sharpening iterations $t$, learning rate $\alpha$, neighbors $k_s$ used in estimating the data density; and one affecting training – number of training epochs $e$. We showed that $e = 1000$ is a good preset for SDR-NNP, thus also for α-SDR-NNP. The other three parameters affect the resulting learned projection in several ways, such as changing the visual cluster separation (undersegmentation, oversegmentation). Hence, SDR-NNP is not
easy to use, as it requires some amount of parameter tuning experimentation. Moreover, every parameter change implies retraining which takes minutes (see Sec. 4.4.3).

\(\alpha\)-SDR-NNP largely solves this problem by adapting the learning rate to the local cluster structure of the data, which is estimated by \(k\)-means. This requires introducing a new parameter, the number \(K_m\) of \(k\)-means clusters used during training (Sec. 4.3.3). We showed that \(\alpha\)-SDR-NNP is far less sensitive to changes of \(t\), \(\alpha\), and \(k_s\), so these parameters can be simply set to default values; and is also insensitive to setting \(K_m\), as shown in Sec. 4.4.2. Intuitively put, our new method allows the user to control the desired final projection outcome at a higher and more global level (that is, in terms of expected clusters \(K_m\)) than the local controls that SDR-NNP required in terms of number of iterations \(t\), learning rate \(\alpha\), and density-estimation bandwidth \(k_s\). This is especially important since such local parameters can vary a lot over a given data set, so there is no way to determine good global defaults for them for the entire data set. In contrast, the expected number of clusters \(K_m\) is a much higher-level parameter which does not depend strongly on the local data structure, so, for a given data set, is a much easier-to-control setting. Given the above, our new method can be seen as virtually parameter-free, thus easy and effective to use.

**Limitations:** While inheriting the abovementioned desirable properties from NNP, \((\alpha\text{-})SDR\text{-NNP}\) also inherit some of its limitations. Its OOS support cannot extend to data sets of a completely different nature than those it was trained on – arguably, a limitation that most machine learning methods have. Also, \((\alpha\text{-})SDR\text{-NNP}\) is only as good as the baseline projection \(P\) that was used during training. Using a low quality projection leads to \((\alpha\text{-})SDR\text{-NNP}\) learning, and reproducing, that behavior.

Separately, SDR-NNP is prone to instability in the generated projection, which manifests itself as over- or undersegmentation of the data into too many, respectively too few, visual clusters, as a function of its parameter values. As discussed above, our new method \(\alpha\text{-SDR\text{-NNP}}\) largely removes this problem. Still, \(\alpha\text{-SDR\text{-NNP}}\) has its own limitations. Its only parameter \(K_m\), the number of \(k\)-means clusters (Sec. 4.3.3), can be set quite freely to values differing as much as twice from the true number of clusters in the data (see Fig. 4.12 and related text). Still, there can be data sets for which the user has no idea, even within this error margin, to what a good \(K_m\) setting is. Exploring how \(\alpha\text{-SDR\text{-NNP}}\) behaves in those cases and, if necessary, refining it to be even less sensitive on the \(K_m\) setting, is for future work.
Applications: As our method is generic (C3), it can handle high-dimensional data sets coming from any application domain. A particular application domain where such projections are very useful is in engineering classification models. More particularly, engineering classifiers for image data is an attractive application area of $\alpha$-SDR-NNP, since the method can be easily enhanced to display the actual images corresponding to the projected points. Thereby, users can examine a projection, e.g., labeled by ground-truth or inferred information, and get insights in why and where mis-classifications occur. Such scenarios involving using projections have been presented in recent research on medical image classification [30, 128, 165] and cell imaging [128, 234]. $\alpha$-SDR-NNP is especially attractive for such use-cases since these provide a known number of clusters $K_m$ in the data to be detected, equal to the number of classes to be inferred. As such, setting the single free parameter $K_m$ of our method is simple. We are considering exploring how our method can address such use-cases in future work.

4.6 conclusion

We have presented SDR-NNP and $\alpha$-SDR-NNP, two new methods for computing projections of high-dimensional data sets for visual exploration. Our methods have several desirable and complementary characteristics of two earlier projection methods, namely NNP (speed, out-of-sample support, ability to accurately imitate a wide range of existing projection techniques) and SDR (projecting complex data sets into visually well-separated clusters of similar samples). SDR-NNP remove the main obstacle for practical usage of SDR – its high computational time. $\alpha$-SDR-NNP further enhances SDR-NNP’s ease-of-use by removing the latter method’s sensitivity to parameter setting – $\alpha$-SDR-NNP is essentially a parameter-free method. $\alpha$-SDR-NNP also increases the quality metrics of its resulting projections as compared to SDR-NNP. We have demonstrated both methods on a range of data sets coming from different application domains. In particular, we showed how SDR-NNP can bring added value in the exploration of a large and recent astronomical data set leading to findings which were not achievable by SDR or NNP alone.

Future work can target several directions. Our work showed that it is possible to learn sharpening methods for high-dimensional data, so it is interesting to apply our techniques to other domains data sharpening is used, e.g., image segmentation, graph bundling, and data clustering and simplification. For the projection use-case, refining $\alpha$-SDR-NNP’s network architecture to speed its training is of high practical interest. Finally, deploy-
ing $\alpha$-SDR-NNP as a main tool for astronomers to analyze their million-sample data sets is a goal we aim to pursue in the short term.

ACKNOWLEDGEMENT

This work is supported by the DSSC Doctoral Training Programme co-funded by the Marie Sklodowska-Curie COFUND project (DSSC 754315), and FAPESP grant 2020/13275-1, Brazil. The GALAH survey is based on observations made at the Australian Astronomical Observatory, under programmes A/2013B/13, A/2014A/25, A/2015A/19, A/2017A/18. We acknowledge the traditional owners of the land on which the AAT stands, the Gamilaraay people, and pay our respects to elders past and present. This work has made use of data from the European Space Agency (ESA) mission Gaia (https://www.cosmos.esa.int/gaia), processed by the Gaia Data Processing and Analysis Consortium (DPAC, https://www.cosmos.esa.int/web/gaia/dpac/consortium). Funding for the DPAC has been provided by national institutions, in particular the institutions participating in the Gaia Multilateral Agreement.
Recent astronomical surveys have produced high-dimensional data sets, typically tens of parameters for hundreds of thousand to billions of objects. Dimensionality reduction (DR) can be used to find interesting structures in the original $n$-dimensional ($nD$) space in terms of clusters in the 2D projections. However, the resulting clusters are often challenging to explore, as they can overlap, due to inherent limitations of DR. Sharpened DR (SDR) addresses this problem by sharpening the $nD$ data prior to DR, so that the resulting 2D clusters are well-separated. In this chapter, we apply SDR on the ‘GALactic Archaeology with HERMES survey’ (GALAH+ DR3), which contains chemical abundance measurements for hundreds and thousands of stars. We create a 2D projection from the original $nD$ abundance space and then explore the astronomical implications of the clearly separated clusters visible in the projection. From our analysis, we find that the resulting clusters can be associated with different subcomponents of the Milky Way, such as metal-rich or metal-poor thin- and thick-disk stars and various groups of halo stars. As the various stars in the SDR clusters likely formed over different time scales, our visualization technique could potentially help astronomers to better understand the formation and evolutionary history of the Milky Way.

5.1 INTRODUCTION

The formation of galaxies like the Milky Way, a typical galaxy in the local Universe, remains a major puzzle. Astronomers now believe that nearly all big galaxies are made up of a combination of stars formed within the galaxy itself and stars formed in other, smaller, galaxies that have been accreted over the lifetime of the galaxy. Our Milky Way galaxy appears to consist of four major components as shown in Fig. 5.1, defined largely by their motions and to a lesser extent by their spatial distributions: the
Figure 5.1: Depiction of our Milky Way, viewed edge-on as if outside our Galaxy. The Milky Way is composed of bulge, thin-disk, thick-disk, and halo stars (see text for more description). There are different groups of halo stars that are formed within or outside the Milky Way such as in-situ stars and other streams of stars (e.g., Helmi streams, GES, Sequoia, and Thamnos).

fast-moving thin-disk stars that rotate around the center of the Galaxy in a thin, disk-like plane; the slower-moving thick-disk stars that also orbit around the center of the Galaxy, but more slowly and in a thicker, less-dense disk; the bulge, stars moving quickly in more-or-less randomly oriented, mostly radial orbits, tracing out a somewhat flattened spheroidal shape around the center of the Galaxy; and the halo, consisting of stars moving on relatively slow, radially plunging orbits describing a roughly spherical shape that extends, at lower and lower density, far beyond even the extent of the thin disk. By number and mass, the thin disk contains the most stars, with the halo and thick disk containing the least. Halo and bulge stars tend to be the oldest, with thick-disk stars being on average slightly younger, and thin-disk stars being on average younger than the other components. For halo stars, in particular, there appear to be many groups of halo stars containing prominent subcomponents such as in-situ stars (formed within the Milky Way) and streams of stars (e.g., the Helmi streams [87]) that are thought to be remnants of both disrupted globular clusters and stars from other galaxies that have been accreted onto the Milky Way, such as Gaia-Enceladus-Sausage (GES), Sequoia, and Thamnos [42].

However, separating out the various generations of star formation events is very challenging (for a recent review of the state of the art of this subject,
see [86]), because there are no clear boundaries at the location of these accreted stars as they are scattered throughout the halo. One way to tackle this challenge is by using the chemical composition of stars. In the late 1950s, Burbidge et al. [26] realized that different chemical elements were formed at different times in the history of the Universe through different nucleosynthetic processes [155]. In other words, the distribution of the elements will be similar among stars that were formed at the same time and place, which allows astronomers to use the composition of stars in the Milky Way to unravel its formation and evolutionary history. Unfortunately, determining the precise relation between the composition of stars, their ages, and the environments in which they formed is difficult [201], due largely to theoretical uncertainties in how stars form new elements through explosive and non-explosive processes [9, 155].

It is therefore of great interest to astronomers (e.g., one of the co-authors of this study, Prof. S.C. Trager) to understand which stars contain which elements and how much of each element. However, instead of using the amount of a single element, astronomers use the ratio of two different elements, because the relative amount of elements varies among stars and gives hints to understand the underlying element formation processes. To this end, astronomers use ‘stellar abundances’ that measure the composition ratio of two different chemical elements in the spectra of stars relative to that of the Sun. An abundance ratio (or ‘abundance’ in this chapter) \([X/Y]\) of an element X with respect to another element Y for some star is defined as

\[
[X/Y] = \log_{10} \left( \frac{N_X}{N_Y} \right)_* - \log_{10} \left( \frac{N_X}{N_Y} \right)_{\odot},
\]

(5.1)

where \(N_X\) and \(N_Y\) are the number densities of element X and Y (i.e., the number of atoms of element X and Y per unit volume), respectively. The subscripts \(*\) and \(\odot\) refer to the star in question and our Sun, respectively – that is, Eqn. 5.1 measures the relative amount of elements of the respective stars is in comparison to the those of the Sun. One can next trace back what processes form the Milky Way by understanding the different element formation mechanisms associated with these stellar abundances.

We introduce in Fig. 5.2 four major element-formation mechanisms – the alpha-process, the rapid neutron capture process (\(r\)-process), the slow neutron capture process (\(s\)-process), and the process that produces iron-peak elements – to explain how the major chemical elements of the stars from our Milky Way are formed. Note that we show below and in Fig. 5.2 the
dominating process(es) for each element, but there are other processes\(^1\) that are responsible for the production of each element.

- **The alpha-process** converts helium into heavier alpha-elements by building up from \(\alpha\) particles (helium nuclei). Alpha-elements, which are lighter than \(^{56}\text{Fe}\) in the periodic table, are abundant in the Universe. Roughly all alpha elements, such as magnesium, silicon, titanium, and calcium are produced by Type II supernovae, the explosions of massive stars.

- The **\(r\)-process** occurs for rapid neutron captures in iron-creating elements such as uranium and europium. The \(r\)-process elements are typically produced also by Type II supernovae of high-mass stars. The \(r\)-process produces heavier elements than in the alpha-process.

- The **\(s\)-process** occurs through slow neutron captures and produces elements such as barium and strontium. The \(s\)-process mainly occurs in asymptotic giant branch (AGB) stars\(^2\). The \(s\)-process produces heavier elements than in the alpha-process.

---

1 For a full view, refer to the following link that shows the cosmic origin of each element in the periodic table: [https://apod.nasa.gov/apod/ap230108.html](https://apod.nasa.gov/apod/ap230108.html)

2 AGB stars are highly evolved intermediate-mass (~ 3–8 \(M_\odot\), where \(M_\odot\) is the solar mass) stars that burn H and He in shells, leading to a slow build-up of neutrons able to produce heavier elements like Y, Ba, Sr, Ce, and others.
5.1 Introduction

Figure 5.3: An example of the Tinsley diagram [200]. (a), (b), and (c) relate to halo, thick-disk, and thin-disk stars, respectively. The point where [Mg/Fe] decreases is related the ‘delay time of Type Ia supernovae (SNe Ia)’, as magnesium is no longer produced by SNe II while iron builds up from production by SNe Ia.

- The iron-peak elements, like copper, zinc, manganese, and iron itself are produced by both type Ia and II supernovae. Type Ia supernovae result from the explosion of white-dwarf stars, the embers (remains) of low-mass stars (the Sun will eventually become a white dwarf), interacting with a companion star.

It is also important to note that the aforementioned stars that undergo Type Ia or Type II supernovae have different life spans – Type II supernovae occur for newly formed stellar populations (populations younger than 100–500 million years), while type Ia supernovae typically occur more than 300 million years after a population is formed. The timescale of AGB stars is typically 500 Myr (million years)–2 Gyr (billion years). Therefore, astronomers can analyze the time scale of star formation events in a stellar system by looking at the stellar abundances.

Specific element abundances are indicative of specific evolutionary events. The most commonly and conventionally used elements for abundances are [Fe/H] and [X/Fe], where X is any other element. [Fe/H] is particularly important for astronomers because the hydrogen content is known to change little (although it is slowly decreasing as stellar fusion converts hydrogen to helium), whereas the iron content is increasing (unsteadily) in time [91], allowing astronomers to infer the time scale of star formation.
events. Following the discussion above, different abundance ratios [X/Fe] are indicative of different evolutionary timescales.

With these abundances one way to infer such evolutionary events is by looking at 2D scatterplots of specific abundance pairs such as [Fe/H]–[Mg/Fe] (for example). Out of these abundance pairs, the 2D scatterplot of [Fe/H] versus [Mg/Fe] is of special interest to astronomers, as the location or the distribution of stars in this plot is directly associated to different star formation events, which can be used to ‘reverse engineer’ the abundance data into information about the star-formation events in the stellar system.

The aforementioned 2D scatterplot ([Fe/H] vs. [Mg/Fe]) is called the Tinsley diagram, where it plots the magnesium to iron ratio ([Mg/Fe]) as a function of the metallicity ([Fe/H]) [200], following the work of Wallerstein [222]. An example of the Tinsley diagram is shown in Fig. 5.3. Astronomers generally look at three different parts of Fig. 5.3, in the order (a)–(c). Stars that fall into area (a) are associated with Type II supernovae (SNe II), as it is assumed that Type II supernovae produce constant [Mg/Fe]3. Stars that have [Fe/H] < −0.5 are often called ‘metal-poor’ and stars with higher [Fe/H] are called ‘metal-rich’. As [Fe/H] increases along the x-axis, the point where [Mg/Fe] decreases in (b) – the ‘knee’ – is directly related to the delay time of Type Ia supernovae (SNe Ia), as this inflection point represents the epoch at which SNe Ia start to produce iron in sufficient quantities to affect [Mg/Fe]. The [Mg/Fe] decrease in (c) is associated with continuous star formation because magnesium is no longer produced while iron builds up. Note that the ‘knee’ depends on the star-formation efficiency (SFE) in a given star-forming environment, which controls how many SNe II are formed before the SNe Ia start to explode and change the abundance pattern. As the current cosmological model seems to indicate that larger objects – like the Milky Way – are built from smaller objects, which have their own star-formation histories, stars that were formed originally in smaller systems can have different ‘knees’ if their original hosts had different SFEs than the bulk of the stars that formed the Milky Way halo. By tracing stars with different ‘knees’ or by finding stars with different kinematics that have the same ‘knee’, one can learn about the objects that formed the Milky Way’s halo.

Related to the aforementioned nucleosynthetic processes, astronomers can link stars that fall in categories Fig. 5.3(a)–(c) to different components

---

3 Note that this is not strictly true, as the Mg production of SNe II depends on the progenitor mass, but it appears to be roughly the case on average in typical stellar systems; see, e.g., Pagel [155]
of the Milky Way. This reasoning comes from the fact that thick-disk stars consist of older stars created in earlier stages of galaxy formation, thus containing fewer heavy elements compared with thin-disk stars. The same applies to halo stars compared with thick-disk stars. Stars in Fig. 5.3(a), (b), and (c) have therefore been assumed to be halo, thick-disk, and thin-disk stars, respectively, in previous studies. Astronomers can then infer the component and origin of the stars in the Milky Way by analyzing the Tinsley diagram.

In summary, the analysis of stellar abundances can ultimately be used to find stars that have similar abundances and abundance patterns, and furthermore to understand how many different nucleosynthetic processes are at play [199] and to understand the lifetime of these stars in the Milky Way [155]. The stellar abundances considered in the analysis can be mathematically modelled as an \( n \)-dimensional linear vector space, called the \( n \)-D abundance space \( (\mathbb{R}^n) \). By analyzing different clusters (or groups) of stars in this \( n \)-D abundance space and finding the number of clusters in that space, astronomers are able to achieve the aforementioned goal. However, the number of dimensions (i.e., abundances) is typically high \( (n \geq 10) \), which makes it difficult to intuitively and visually search for these clusters. While there exists a number of multidimensional visualizations, dimensionality reduction (DR) is the preferred method as DR plots similar-valued samples close to each other forming clusters and at the same time scales well for high-dimensional data. DR methods are defined in this chapter as a function \( P : \mathbb{R}^n \rightarrow \mathbb{R}^s \), where \( s = 2 \) is typically used [140]. Applying DR methods to the \( n \)-D abundance space results in a 2D projection of that space, which we will denote as ‘2D abundance space’.

Astronomers have previously applied various DR techniques for stellar abundance studies. Among these techniques, principal component analysis (PCA) [159] has been widely used to explore stellar abundance space [199] and to search for clusters in the abundance space [21]. Anders et al. used \( t \)-distributed Stochastic Neighbor Embedding (\( t \)-SNE) [133] to dissect the abundance space of the HARPS GTO sample [7, 146]. More recent studies suggest using sharpened DR (SDR), which uses a preconditioning step prior to DR to achieve better separated clusters that are otherwise less separable using conventional DR methods [111, 112]. Kim et al. [111, 112] used SDR on 10 different stellar abundances from a random subset of stars in the second data release of ‘GALactic Archaeology with HERMES survey (GALAH DR2)’ [25] to visually explore the clusters in 2D and associate these clusters with different components (and origin) in the Milky Way.
In this chapter, we aim to explore the 2D stellar abundance space of the most recent data release of GALAH (GALAH+ DR3) [23], which allows astronomers to determine abundances for hundreds of thousands of stars and explore the abundance space with statistically significant data sets for the first time. We use SDR here to project the nD abundance space to a 2D space, and ultimately understand how stars from different clusters were formed, whether these clusters have similar kinematic processes, and to ultimately confirm and better understand the current formation and evolutionary history of the Milky Way. Furthermore, we explore SDR with various parameter settings, which has not been fully explored for astronomical surveys.

The chapter is structured as follows. Sec. 5.2 explains the data set in use and SDR. Sec. 5.3 shows the SDR results, Sec. 5.4 analyzes each group of stars from the SDR results. Finally, Sec. 5.5 and Sec. 5.6 further discuss and concludes the chapter.

5.2 Method

5.2.1 Data set

In this chapter, we use the latest data release of GALAH (GALAH+ DR3) and cross-match it with the early release of the latest Gaia data (Gaia eDR3) [43, 209] data set to acquire additional information of the stars such as 6D phase-space coordinates.

5.2.2 Attributes

We use here $n = 10$ stellar abundances – [Fe/H], [Mg/Fe], [Al/Fe], [Si/Fe], [Ca/Fe], [Ti/Fe], [Cu/Fe], [Zn/Fe], [Y/Fe], and [Ba/Fe] – as input attributes for SDR. We use the same attributes used by Kim et al. [111, 112], but from GALAH+ DR3 instead of DR2. These attributes were originally selected based on the same abundances used by Anders et al. [7]. We use the same set of abundances, as we aim to reproduce the results and do a follow-up study of Anders et al. and Kim et al. [7, 112], but with a larger and more accurate data set. Note that [Sr/Fe] was excluded by Kim et al. [112] because [Sr/Fe] was not released in the GALAH DR2 catalogue [25]. In this chapter we exclude [Sr/Fe] because of data reliability issues (see Sec. 6.3 in [23]), even though it was used in [7]; the number of stars that do not have quality issues with the measurement of [Sr/Fe] is also significantly lower than for other abundances used as SDR input (see Fig. 22 in [23]).
For these reasons, we use 10 stellar abundances as input to SDR. However, SDR can in general be applied to higher number of dimensions.

5.2.3 Data cleaning

Next, we acquire a clean data set based on a few criteria from the GALAH+ DR3 catalogue associated with our data set. The GALAH+ DR3 catalogue contains data-quality flags that filter out data with issues, such as corrupted data or data with unreliable astrometry. Additional information on astrometry is required because the distance to a star is used to calculate its surface gravitational acceleration, which affects the spectrum and thus affects the abundance. We use the following inclusion criteria for stars:

1. Stars that have no known quality issues with the measurement of abundances (i.e., flag_fe_h=0 and flag_X_fe=0 from the GALAH+ DR3 catalogue parameter associated with [X/Fe], where X is Mg, Al, Si, Ca, Ti, Cu, Zn, Y, and Ba) [23].

2. Stars that have no identified issues with the [Fe/H] measurement (flag_fe_h=0).

3. Stars with no spectral flags (flag_sp=0). The first bit of flag_sp indicates unreliable astrometry in Gaia DR2, i.e., the star has a Renormalised Unit Weight Error (RUWE) value higher than 1.4 [23]. RUWE indicates the quality of the measurement of the position of stars. Hence, spectral flag values other than zero indicate less reliable astrometry in the cross-matched Gaia DR2 data.

4. Stars that are not too distant (distance_bstep<5 kiloparsec (kpc)). Objects (like stars) become apparently fainter with increasing distance, resulting in lower signal-to-noise. The visibility of thin-disk stars and part of the thick-disk stars is also restricted by dust. The distance is inferred (using a Bayesian inference code [178]) by comparing the observed magnitude and color of the star to models of stellar evolution of stars with the chemical composition, temperature and surface gravitational acceleration inferred from its spectrum.

---

4 Inaccurate spectra can also result in RUWE = 0.
5 Gaia DR2 was used here instead of Gaia eDR3, because Gaia eDR3 was not released during the main analysis of GALAH+ DR3 [23].
6 The apparent brightmesses of objects of the same intrinsic brightness measured from Earth are inversely proportional to the distance squared.
7 The dust layer exponentially decreases further away from the plane of the disk, so halo stars above or below the disk are less affected.
8 The color of a star indicates its surface temperature: stars with high surface temperatures (e.g., 10,000 K are blue, and those with low surface temperatures are red (e.g., 3,500 K).
Figure 5.4: (a) Toomre diagram for 312,018 ‘clean’ stars, of which 309,053 stars satisfy the condition $V_{cut} > |V - V_{LSR}|$ (disk stars colored in blue) and 2,815 stars (halo stars colored in green) do not. (b) Toomre diagram for the 10K subset of stars, of which 7,185 stars satisfy the $V_{cut} > |V - V_{LSR}|$ (disk stars colored in blue) and 2,815 stars (halo stars colored in green) do not.

The difference in magnitude between the observed and modeled star is a direct measurement of the distance, after correcting for dust extinction, which can be seen from the difference in the observed and modeled color. A limit of 5 kpc is set to include halo stars from various subcomponents in the Solar neighborhood, such as in-situ stars or stars from different merger events. The same limit is also used in [120], where new stars from the Helmi streams have been discovered.

(5) Stars that have a parallax divided by its parallax error higher than 10 ($\text{parallax\_over\_error}>10$), so that the selected stars have reliable distance and (space) velocity measurements.

(6) Stars that have no missing values in all 10 attributes, as SDR cannot deal with matrices with missing values [112]. In particular, [Sr/Fe] is excluded due to too many missing values (see Sec. 5.2.2 above).

The total number of stars that satisfy the aforementioned data cleaning criteria are 312,018 (approximately 53% of the entire data set of 588,571 stars) and we will call this set the ‘clean’ data set.

5.2.4 Random subset

We next select a 10K random subset from the clean data set instead of using all 312K stars, because running SDR on hundreds of thousands of observations is infeasible or requires the use of high-performance computing due to its expensive computation [112]. Therefore a smaller subset
of 10K observations is used. Other methods to solve this scalability issue are discussed further in Sec. 5.5.

When selecting a random subset, we prefer a subset that consists of both halo and disk stars, as our goal is to analyze different clusters of stars with varying subcomponents and origins in the Milky Way. To identify disk and halo stars, astronomers often use the Toomre diagram as shown in Fig. 5.4, which plots the quadratic addition of the velocities $V_R$ and $V_z$ as a function of the perpendicular y-axis velocity ($V_T$). Given the Local Standard of Rest (LSR) velocity ($V_{\text{LSR}}$), which is the velocity of the Sun’s orbit around the Galactic Center, stars with velocity $V$ that satisfy $|V - V_{\text{LSR}}| \geq V_{\text{cut}}$ (where $V_{\text{LSR}} = 232 \text{ km s}^{-1}$ and $V_{\text{cut}} = 210 \text{ km s}^{-1}$) are identified as halo stars (green colored points in Fig. 5.4) \[119\]. For this condition, the (absolute) velocity difference between a star and the Sun is used instead of positional coordinates ($x$, $y$, and $z$) of a star, because in the local Solar neighborhood, halo stars and disk stars are mixed together and can only be distinguished through their motions or their elemental abundances.

Fig. 5.4(b) shows the 10K subset of stars, of which 7,185 stars satisfy the $V_{\text{cut}} > |V - V_{\text{LSR}}|$ (disk stars colored in blue) and 2,815 stars (halo stars colored in green) do not. As halo stars only account for a small portion of the sample stars (less than 1%) in the clean data set\[9\], randomly selecting stars from the full clean data set will result in a subset consisting of almost entirely disk stars – a well-known problem also in many machine learning applications when so-called unbalanced data sets are used. To avoid this sampling issue and gain a more balanced subset of 10K stars, we select all 2,815 halo stars and randomly select the rest (7,185 stars) from the identified disk stars. The Toomre diagram for the newly acquired subset is shown in Fig. 5.4(b). Note that other sampling strategies can be readily used to obtain our desired small (10K samples) data set so that it captures well the distribution of both halo and disk stars.

5.2.5 *Sharpened dimensionality reduction (SDR)*

We apply SDR on the cleaned, randomly sampled subset which consists of 10 dimensions and 10K observations. Sharpening is first performed on the 10-dimensional data prior to DR. Sharpening shifts all data samples iteratively along the density gradient of a multivariate kernel density estimator

---

\[9\] Note that this bias is not confined to the clean data set, but also exists in the original GALAH data set due to observing constraints that limit the stellar distances to the very inner halo of the Galaxy [23, 25]
model. The detailed method of SDR is explained in [112] (See Sec. Proposed Method).

The SDR method has three parameters: $\alpha$, $t$, and $k$. The learning rate $\alpha$ is the most important parameter out of the three, as it controls the speed of shifts along the density gradient, which affects the degree of segmentation of the input data set [112]. This degree of segmentation relates to the number of sub-clusters formed in the resulting projection. Sub-clusters in the data may be destroyed (i.e., undersegmentation) if $\alpha$ is set too large. Conversely, too many sub-clusters may result (i.e., oversegmentation) if $\alpha$ is set too small. The number of iterations $t$ controls the number of shifts, and the number of nearest-neighbors $k$ controls how localized the shift is [112]. In this chapter, we use the recommended ranges and values from Kim et al. [112]. The detailed parameter values and the combination of these values for testing SDR are described later in Sec. 5.3.

For the baseline DR method – that is, the DR method applied to the sharpened high-dimensional data set to yield the final 2D scatterplot – we use Landmark Multidimensional Scaling (LMDS). This DR method shows a good separation of clusters in the 2D projection using GALAH DR2 and is relatively fast to compute [112]. The landmark ratio parameter for LMDS is set to 0.1 (i.e., 10% of the stars are selected as ‘landmarks’ in the initial DR before projecting the remaining stars).

5.3 SDR Results

We test SDR on our 10K data set with different SDR parameter settings. Our aim here is to find an SDR result (that is, 2D projection of the input data) that visually exhibits the following conditions:

1. Clusters are well-separated in terms of the color-coded [Fe/H] attribute, which is used by astronomers to further analyze the projection [7, 112];

2. The number of clusters should not be too high (e.g., more than 10 clusters), where oversegmentation can occur and further analysis would be complicated;

3. The number of clusters should not be too low (e.g., fewer than five clusters), as cluster structures in the abundance space have a risk of being undersegmented, leaving out details.

Let us explain this in more detail. Subcomponents of the Milky Way have different chemical compositions and this difference is also prominent in the GALAH+ DR3 data set. Therefore, choosing a small number of clus-
5.3 SDR Results

Figure 5.5: SDR results with different parameter settings, where the points are color-coded by [Fe/H]. Panels I(a)–(i) show the results with increasing \( \alpha \) in the rows and increasing \( k \) in the columns while \( t \) is set to 10, and panels II(a)–(b) show the results for \( t = 2, 6 \), respectively. Panels I(a)–(d) and II(a)–(b) have too many oversegmented clusters, while too few clusters are present in panels I(f)–(i). Panel (e) is the selected SDR plot used for further analysis in Sec. 5.4 and is highlighted in red.

Trees (e.g., three or four) would result in obvious results, such as different clusters representing the thin-disk, (metal-rich or metal-poor) thick-disk stars, and halo stars, as also suggested in Kim et al. [112]. However, we would also like to confirm that there are differences in the composition of different halo stars (e.g., in-situ, GES, Sequoia, or Thamnos stars) and to explore other clusters potentially representing new subsets of stars that are of interest to astronomers.

We test all combinations of \( \alpha = \{0.01, 0.03, 0.06, 0.07, 0.08, 0.09, 0.1, 0.12\} \), \( k = \{50, 100, 200\} \), and \( t = \{2, 4, 6, 8, 10\} \), and then color-code the projections with the first attribute (i.e., [Fe/H]). The aforementioned ranges of the
Figure 5.6: (a) SDR projection of the 10K GALAH+ DR3 subset with points color-coded by [Fe/H]. Clusters are well-separated with little overlap, which allow one to easily manually label the eight clusters (Class 1–8) and outliers (Class 9) as shown in (b).

Parameter values are based on the recommendations by Kim et al. [112]. We show a few results in Fig. 5.5 to show how the cluster separation is affected by different parameter settings. Fig. 5.5(I)(a)–(i) shows $\alpha$ increasing in the rows and $k$ increasing in the columns. The results for varying $t$ are shown in Fig. 5.5(II)(a)–(c). For the full results with all combinations of parameter settings, please refer to the Supplemental materials (Fig. 5.16–5.18). Several insights follow from our results. First, a higher number of iterations (i.e., $t = 10$) yield better separation of clusters satisfying condition (1). Second, most of the results with $k = 200$ (when $t = 10$) yield too few clusters (fewer than five). Third, there are several results that satisfy conditions (1)–(3) when the learning rate ($\alpha$) is set to either 0.07 and 0.08 (when $t = 10$, $k = 50, 100$). Fig. 5.5(I)(a)–(c) and (II)(a) suggest that there are largely two structures present\(^{10} \) — stars with very low metallicity vs. remaining stars (with higher metallicity). Although there are several plots that satisfy conditions (1)–(3), we select a single SDR projection that retains the overall structure and also allows us to gain insight in the individual clusters present in the 2D abundance space.

As mentioned earlier, a key issue in using visualization methods to explore high-dimensional data is that there are many potential views, or images, to depict the same data, e.g., created by different methods, or the

\(^{10}\) Note that the overall pattern of the structures in the resulting projections look different, but are actually alike when vertically or horizontally flipped or rotated. This is due to either how subsamples are randomly initialized and moved on the plot or how eigenvectors point in different directions, both during the DR step.
Figure 5.7: The abundance [X/Fe] as a function of [Fe/H] is plotted (X is one of the nine abundances used as SDR input – [Mg/Fe], [Al/Fe], [Si/Fe], [Ca/Fe], [Ti/Fe], [Cu/Fe], [Zn/Fe], [Y/Fe], and [Ba/Fe]). In particular, panels (I)(a) and (II)(a) are Tinsley diagrams. To avoid clutter, (I) shows the only Classes 1–4, and (II) shows Classes 5–9.

same method with different parameters. This is a practical problem – domain experts can be overwhelmed by such an abundance of views and, therefore, prefer to have a smaller set of views – ideally, a single view to study. Of course, this implies trade-offs as, in general, no single visualization (of complex data) can capture all data aspects. Still, if we aim to select a single SDR projection, we choose the one that satisfies conditions (2)–(3) and best satisfies condition (1). We focus on (1) in particular, because [Fe/H] (also called the metallicity) is an important tracer of different nucleosynthetic processes (see Sec. 5.1). The selected SDR result is shown in Fig. 5.5(I)(e). The possibility of selecting other SDR results are discussed later in Sec. 5.5. Fig. 5.6(b) shows the same SDR plot with manually assigned class labels to these well-separated clusters (Class 1–8) and outliers (Class 9). Note that the enumeration of these labels is arbitrary. The physical meaning of these classes is discussed in Sec. 5.4.
Figure 5.8: The Tinsley diagram – the abundance $\text{[Mg/Fe]}$ as a function of $\text{[Fe/H]}$ – is shown in the form of a contour plot of Fig. 5.7. From this diagram, we can assume that Class 1 consist of mostly scattered halo stars, Class 2 are mostly (metal-poor) thick-disk stars mixed with other stars, Classes 3 and 4 are mostly the GES stars, Classes 5 and 6 are mostly thin-disk stars, and Classes 7 and 8 are mostly thick-disk stars.

Figure 5.9: The abundance $\text{[Eu/Fe]}$ as a function of $\text{[Fe/H]}$ is plotted. To avoid clutter, (a) shows the only Classes 1–4, and (b) shows Classes 5–9.

5.4 Components and origin of stars in the Milky Way

We explore here the astronomical implications of the labeled clusters (Class 1–9) resulting from the SDR projection in Sec. 5.3.
5.4 Components and origin of stars in the Milky Way

5.4.1 Tinsley diagram

As stated earlier in this chapter, the key reason to visualize our data is to trace back abundance patterns to star formation patterns. We explore next the physical meaning of each identified cluster in Fig. 5.6(b) in terms of its subcomponents and origin in the Milky Way.
Figure 5.12: The left column shows the Toomre diagrams and the right column shows the Energy as a function of $L_Z$ for different Classes, both in the form of contours of the points in Fig. 5.11(a) and (d). A small peak shown in the red circled area of (b) suggests that Classes 1–2 contain stars possibly from the Thamnos merger event (and some Sequoia stars in Class 1) [42].

Figure 5.13: $[\text{Mg/Cu}]$ is plotted as a function of $[\text{Na/Fe}]$. To avoid clutter, (a) shows the only Classes 1–4, and (b) shows Classes 5–9.

We first plot the identified SDR clusters on the Tinsley diagram, as shown in Fig. 5.7(I)(a) and (II)(a). To avoid clutter, we plot stars in Classes 1–4 in
5.4 Components and Origin of Stars in the Milky Way

Figure 5.14: [Eu/Mg] is plotted as a function of [Fe/H]. To avoid clutter, (a) shows only Classes 1–4, and (b) shows Classes 5–9.

Figure 5.15: Hertzsprung–Russell diagram that plots the surface gravity (g) as a function of the surface temperature (log $T_{\text{eff}}$ K) in logarithmic scale. To avoid clutter, (a) shows only Classes 1–4, and (b) shows Classes 5–9.

(I)(a) and Classes 5–8 in (II)(a). We will ignore Class 9 for now, as Class 9 stars are outliers. For a clearer view of how points are distributed, we show the contour plot of Fig. 5.7(I)(a) and (II)(a) in Fig. 5.8. From Fig. 5.8, we can immediately observe that different Classes consist of stars from different components of the Milky Way. Class 1 seems to be mostly scattered halo stars and Class 2 looks like (metal-poor) thick-disk stars mixed with other stars. For Classes 3–4, a knee around [Fe/H] $\sim -0.7$ to $-0.8$ can be observed for Class 4, and at [Fe/H] $\sim -1.2$ for Class 3, which suggests that these Classes contain GES stars (see Fig. 5 in [143]). Classes 5–6 seem to be mostly thin-disk stars, where Class 6 seems to consist of very metal-rich thin-disk stars, and finally, Classes 7–8 are mostly thick-disk stars, where Class 7 has more metal-poor thin-disk stars compared with Class 8.
5.4.2 $[X/Fe]−[Fe/H]$ plots

Next we explore other plots that show the abundance $[X/Fe]$ as a function of $[Fe/H]$ ($X$ is one of the nine abundances used as SDR input – $[Mg/Fe]$, $[Al/Fe]$, $[Si/Fe]$, $[Ca/Fe]$, $[Ti/Fe]$, $[Cu/Fe]$, $[Zn/Fe]$, $[Y/Fe]$, and $[Ba/Fe]$) as shown in Fig. 5.7. Note that the $[Mg/Fe]−[Fe/H]$ plot is the Tinsley diagram. We also show an additional $[X/Fe]−[Fe/H]$ plot in Fig. 5.9, where $X$ is Eu. Other $[X/Fe]−[Fe/H]$ plots including $X=[Sr/Fe]$, $[Zr/Fe]$, $[Ce/Fe]$ are shown in Fig. 5.10. As explained earlier in Sec. 5.1, different element formation mechanisms are associated with different elements as shown in the list below.

(i) Fe: produced by both SNe Ia and SNe II
(ii) Mg, Si, Ca, Ti: alpha elements produced by SNe II
(iii) Cu, Zn: iron-peak elements in large amounts produced by Type Ia
(iv) Y, Ba, Sr, Ce: $s$-process elements produced by (most likely) AGB stars
(v) Eu: $r$-process element produced by SNe II

When exploring $[X/Fe]−[Fe/H]$ plots, several insights follow based on the list of elements above. For example, as shown in Fig. 5.7(II)(c)–(e), a similar decrease in $[X/Fe]$ from the $[X/Fe]−[Fe/H]$ plots ($X=Si,Ca,Ti$) is observed as also shown in the knee of the Tinsley diagram. This knee occurs because Mg, Si, Ca, and Ti originate in mechanism (ii), mostly produced by SNe II. It is often difficult to analyze the data solely based on a single diagram, which is why we explore the other $[X/Fe]−[Fe/H]$ plots along with the kinematics analyses in the next section.

5.4.3 Kinematics

We plot the Toomre diagram again as shown in Fig. 5.11(a) and (d), where here the points are color-coded by Class. See Sec. 5.2.4 for a detailed explanation of the Toomre diagram. We also plot the $L_Z−E$ in panels (b) and (e) ($L_Z$ is star’s angular momentum perpendicular to the Milky Way’s disk, and $E$ is the kinetic energy of that star) diagram. (e) and (f) shows $L_Z−\sqrt{J_R}$, where $J_R$ is the radial action$^{11}$. These diagrams help understand the motion of stars, which is used to identify stars from different subcomponents. Note that an outlier (star ID: 15564913-6814532) is excluded in Fig. 5.11 be-

---

$^{11}$ The radial action is a measure of a star’s energy in radial oscillations as it moves along its orbit.
cause it is too far away from the other stars (for the full results including this star, please refer to the Supplemental materials). For a clearer view of the patterns (e.g., range of values in each Class and decrease or increase in x- or y-axes values), we additionally show separate contours of the density field of the stars in each Class (from Fig. 5.11) in Fig. 5.12. From the kinematics plots in Figs. 5.11 and 5.12, we can confirm our assumption that different Classes contain different subcomponents of the Milky Way. We analyze next these kinematics plots with the other [X/Fe]–[Fe/H] plots to gain further insight of each Class.

5.4.4 Classes 1–4

We assumed that Classes 1–4 consisted of mostly halo stars and thick-disk stars by looking at the Tinsley diagram in Fig. 5.7. By analyzing the kinematics plots from Fig. 5.12(a)–(b), we can further observe that Class 1 stars are mostly halo stars mixed with some thick-disk stars and that Class 2 stars contain mostly metal-poor thick-disk stars mixed with halo stars. Looking closer at these plots, we observe a small peak as shown in the red circle of Fig. 5.12(b). A similar peak in the top left region of the $L_Z$ versus $E$ diagram has also been observed for the Thamnos and Sequoia merger events in [42] (see Fig. 27 in [42]), where both have low metallicity and Sequoia has a higher $E$-value ($L_Z \lesssim -0.80\, L_\odot$, $E \gtrsim -2.8\, V_\odot^2$, $\odot$ represents the solar neighborhood [152]) than Thamnos. This observation suggests that Class 1 contains stars from the Thamnos merger event, although Class 1 likely has Sequoia stars as well [42]. Class 2 also seems to contain stars from the Thamnos merger event [42].

Additionally, Buder et al. show that stars accreted from satellite galaxies can be best identified by plotting [Mg/Mn] as a function of [Na/Fe] (i.e., accreted stars are expected in (-0.3, 0.5)) [24], as [Na/Fe] shows a clear separation between accreted and the remaining stars and [Mg/Mn] is a tracer of SNe II and SNe Ia [24, 116]. However, we use here copper (one of our input SDR attributes) instead of manganese\(^1\) and plot [Na/Fe]–[Mg/Cu] in Fig. 5.13, as [Mg/Cu] is also a tracer of SNe II and SNe Ia [24, 116]. Based on the above, Fig. 5.13(a) suggests that most of the stars in Classes 1–4 are accreted halo stars originally not from the Milky Way [24]. We further observe from the high [Eu/Mg] (europium is an $r$-process element) in Fig. 5.14(a), that Classes 3–4 seem to be related to the accreted Gaia–Enceladus–Sausage (GES) stars [142] that originate from a dwarf galaxy that merged with the Milky Way approximately 10 billion years ago [88].

\(^1\)Both manganese and copper are iron-peak elements produced by SNe II and SNe Ia (see Sec. 5.1).
The Eu enhancement is also clearer in [Eu/Mg] (Fig. 5.14(a)) than in [Eu/Fe] (Fig. 5.9(a)) as also mentioned by Matsuno et al. [142].

For Classes 3–4, we observe a similar group of GES stars in the Tinsley diagram from Matsuno et al. (see Fig. 5 in [143]), where the turn-off for these stars are around [Fe/H] \(\approx -1\). We can see that Classes 3–4 contain mostly GES stars from Fig. 5.12(c)–(d). However, Classes 3–4 have different abundances as already shown in Fig. 5.8, suggesting that Class 3 stars are mostly halo stars, whereas Class 4 stars seems to be the ‘knee’ of GES.

Although Class 1 stars are more metal-poor than Class 4 stars based on Fig. 5.8 and Fig. 5.7(l)(a), it also seems that Classes 1–4 are continuations of each other, containing mostly accreted halo stars, based on the aforementioned analyses. Class 2 seems to include not only metal-poor thick-disk and Thamnos stars but also GES stars. For Classes 2 and 3 the differences are small for [Eu/Fe] and [Eu/Mg], and stars in Class 3 seem to be a subset of Class 4, as shown in the other abundances in Fig. 5.7(l). The difference seems to come from yttrium and barium, where Class 3 stars have lower [Y/Fe] and [Ba/Fe] than Class 2 stars (see also the histogram of these abundances in Fig. 5.20(k) and (m) in the Supplemental materials). Note that both abundances – [Y/Fe] and [Ba/Fe] – trace the s-process. The differences between Classes 1–4 may explain the reason why the Classes 1–4 are shown as separate clusters in the SDR projection.

5.4.5 Classes 5–8

There are also some distinct features for Classes 5–8 in Figs. 5.7–5.12. Both Classes 5 and 6 seem to contain almost only thin-disk stars when observing the kinematics plots in Fig. 5.11 and Fig. 5.12(c)–(d), but from Fig. 5.8, we can see that Class 5 consists of mostly normal thin-disk stars \((-0.5 < [\text{Fe/H}] < 0.2)\), while Class 6 is especially high in metallicity \([\text{Fe/H}]\), suggesting that Class 6 stars are very metal-rich thin-disk stars.

For Classes 7 and 8, we observe from Figs. 5.8 and 5.12(e)–(f) that Class 7 contains metal-poor thin-disk stars mixed with normal thick-disk stars, whereas Class 8 mostly contains metal-rich thick-disk stars and some metal-rich halo stars. Furthermore, we can see that Class 8 is distinct from the other Classes shown in the Hertzsprung–Russell diagram that plots the surface gravity \((g)\) as a function of the surface temperature \((T_{\text{eff}})\) in logarithmic scale as shown in Fig. 5.15(b). However, it is unclear what this group of stars represents.
5.4.6 Summary

If we group and summarize all our findings obtained so far using SDR, Tinsley diagrams, Toomre diagrams, and density plots, as explained in the previous sections, we reach the following conclusions:

- Class 1 are mostly in-situ halo stars mixed with some Thamnos and Sequoia stars and thick-disk stars
- Class 2 are mostly metal-poor thick-disk stars mixed with Thamnos and GES stars
- Class 3 are mostly GES stars (halo)
- Class 4 are mostly GES stars (‘knee’ of GES) mixed with some metal-poor thick-disk stars
- Class 5 are mostly normal thin-disk stars
- Class 6 are mostly very metal-rich thin-disk stars
- Class 7 are mostly metal-poor thin-disk stars mixed with some thick-disk stars
- Class 8 are mostly metal-rich thick-disk stars mixed with some thin-disk stars

5.5 Discussion

SDR summarization power: Although we selected a single SDR result for further analysis in this chapter, other plots may be of more interest to astronomers depending on the amount of detail they want to see in the 2D abundance space. In the case of oversegmentation, we observed that the resulting clusters from SDR are sometimes continuations of each other in terms of the subcomponents. The large number of clusters from oversegmentation and overlap between them made it challenging to further analyze the individual clusters. On the other hand, when the SDR parameters were carefully controlled, astronomers were able to see clusters that capture interesting details, while also retaining the overall structure.

SDR parameter control: Previously, [112] has shown the possibility of using SDR on abundance data to associate the resulting clusters with major subcomponents (i.e., thin- and thick-disk, and halo stars) using a small number of clusters. In this chapter, we controlled the SDR parameters to acquire more clusters while avoiding oversegmentation to further analyze the differences between the clearly separated clusters. Results show
that different clusters were associated with metal-poor and metal-rich disk stars, and also for stars from various merger events. For instance, a clear separation between some GES stars and other stars were shown in the resulting clusters. A separate cluster was also associated with in-situ and Thamnos stars mixed with some Sequoia and thick-disk stars, while another cluster was associated with metal-poor thick disk, Thamnos, and GES stars. Hence, selecting an SDR result with a clear separation of clusters allowed us to gain more detail and insight on the different chemical composition of stars from various subcomponents.

**SDR plot selection:** Although we manually selected a single SDR result, performing an automatic labeling of the numerous SDR plots that have different parameter settings is also possible using clustering algorithms and quality metrics to quantify the separation of clusters. However, selecting a quantitative measurement that best reflects the perceptual quality of a projection depends not only on the data, but also on what one aims to preserve from the $nD$ data. Therefore, in this chapter, we selected a single SDR plot based on the conditions introduced in Sec. 5.3 and asked an astronomy expert to explore the implications of the separated clusters in this plot and explored the results further.

In this chapter, we made use of a single attribute (i.e., $[\text{Fe/H}]$) to support the selection of the best SDR plot that visually shows the most separation of clusters when color-coded by $[\text{Fe/H}]$ and satisfies the conditions in Sec. 5.3. Although $[\text{Fe/H}]$ was selected based on prior knowledge – $[\text{Fe/H}]$ is an important tracer of nucleosynthetic processes (see Sec. 5.1) – other attributes can also be used. One possible way to visually look for attribute(s) that influence the projection most is by applying the work of da Silva et al. [179], where intuitive visualizations of dimensions that least vary or are ranked based on the variance of point-neighborhoods are shown.

**Data subsampling:** We used here SDR on a 10K subset, which represents only a small portion of the original GALAH+ DR3 data set due to the scalability of SDR. To run SDR on all 312K stars, a neural network-based approach of SDR (SDR-NNP) can be applied [111]. SDR-NNP can be used to run the remaining 311K stars based on a high-quality training projection (e.g., Fig. 5.6). By using SDR-NNP with a training projection of a portion of the full data set may even result in a better projection of the full GALAH+ DR3 data set instead of only using SDR, as the data set has a bias between the number of thin-disk stars and the remaining stars (see Sec. 5.2.4). This approach should be tested and explored further.
Baseline DR method: For the baseline DR method, we selected LMDS, which is a distance-preservation DR method [112]. However, other neighborhood-preservation DR methods with a strong cluster separation characteristic such as t-SNE (or UMAP) can also be used. Although t-SNE may result in a clearer separation of clusters for the GALAH+ DR3 data set compared with LMDS (our baseline DR method), we did not use t-SNE because of several reasons stated below, supported by Fig. 5.16–5.18 and Fig. 5.21–5.23 in the supplemental materials.

First, while the number of clusters are similar, outliers can easily be merged into another cluster in t-SNE compared with LMDS as shown in the supplemental materials (compare Fig. 5.16–5.18 with Fig. 5.21–5.23, respectively). We observe from our results that LMDS is generally better in preserving outliers compared with t-SNE (compare Fig. 5.16–5.18 with Fig. 5.21–5.23, respectively). Kim et al. also showed consistent results (merged outliers) with the GALAH DR2 data set, where both LMDS and t-SNE (as the baseline DR method for SDR) produced the same set of stars in each cluster, but with no existing outliers [112].

Second, we avoided t-SNE because the ‘two structures’ – stars with very low metallicity vs. remaining stars with higher metallicity – that were identified using LMDS (described in Sec. 5.3) were not present with t-SNE (see Fig. 5.21–5.23).

Third, t-SNE is slow, non-parametric, and difficult to control the parameters (despite a clearer separation of clusters) compared with LMDS, as already shown in [112]. The main advantage of using SDR was to use a fast and simple DR technique to obtain more cluster separation, and therefore we favor LMDS over t-SNE.

Outliers: Finally, in-depth analyses of Class 9 and especially star ID 15564913-6814532 (see Sec. 5.4) are required to gain insight on which components of the Milky Way each outlier might represent and why they are separated from the other clusters.

5.6 conclusion

Previous work mainly found different clusters based on the kinematics data and used both abundances and kinematics data to later identify different subcomponents of the Milky Way using these clusters [42, 96, 143].

13 Note that we add all possible parameter combinations due to having three parameters – $\alpha$, $k$, and $t$ – where the effects of these parameters are interconnected.

14 The reason there were no outliers also for LMDS in that study was mainly because a high $\alpha$-value was selected (like $\alpha = 0.1$ in Fig. 5.5).
While it has been possible to create these clusters using only abundances [72], and later identify different subcomponents using both abundances and kinematics data instead, this approach has not been widely explored among astronomers due to the limitations in chemical composition measurement and lack of data until recently. However, with the recent GALAH+ DR3 catalogue, we apply SDR and show that clusters existing in the chemical abundance space can be directly associated with different subcomponents of the Milky Way.

Summarizing, we have used a sharpened dimensionality reduction technique, SDR, to acquire 2D abundance spaces of the recent high-dimensional data set of GALAH+ DR3 with various parameter settings and selected one for further analysis. We have observed that the cluster separation power of SDR helps find potential clusters that may be present in the nD abundance space. We show that the manually labeled SDR clusters are related to different subcomponents of our Milky Way. These classes are an interesting start to probing the chemical diversity of the solar neighborhood and should be followed up in more detail.

ACKNOWLEDGMENTS

This work is supported by DSSC Doctoral Training Program co-funded by the Marie Sklodowska-Curie COFUND project (DSSC 754315). This work made use of the Third Data Release of the GALAH Survey (Buder et al. 2021). The GALAH Survey is based on data acquired through the Australian Astronomical Observatory, under programs: A/2013B/13 (The GALAH pilot survey); A/2014A/25, A/2015A/19, A2017A/18 (The GALAH survey phase 1); A2018A/18 (Open clusters with HERMES); A2019A/1 (Hierarchical star formation in Ori OB1); A2019A/15 (The GALAH survey phase 2); A/2015B/19, A/2016A/22, A/2016B/10, A/2017B/16, A/2018B/15 (The HERMES-TESS program); and A/2015A/3, A/2015B/1, A/2015B/19, A/2016A/22, A/2016B/12, A/2017A/14 (The HERMES K2-follow-up program). We acknowledge the traditional owners of the land on which the AAT stands, the Gamilaraay people, and pay our respects to elders past and present. This paper includes data that has been provided by AAO Data Central (datacentral.aao.gov.au). This work has made use of data from the European Space Agency (ESA) mission Gaia (https://www.cosmos.esa.int/gaia), processed by the Gaia Data Processing and Analysis Consortium (DPAC, https://www.cosmos.esa.int/web/gaia/dpac/consortium). Funding for the DPAC has been provided by national institutions, in particular the institutions participating in the Gaia Multilateral Agreement.
We present figures to support the selection of a single SDR plot in Sec. 5.3 and to show a full view of the kinematics and abundance histograms described in Sec. 5.4.

Figure 5.16: SDR results with different parameter settings using LMDS as the selected baseline DR method. $\alpha$ increases in the rows and $t$ increases in the columns, while $k = 50$. 

5.6 conclusion

APPENDICES
Figure 5.17: SDR results with different parameter settings using LMDS as the selected baseline DR method. $\alpha$ increases in the rows and $t$ increases in the columns, while $k = 100$. The red-colored box indicates our selected SDR plot in Sec. 5.3.
Figure 5.18: SDR results with different parameter settings using LMDS as the selected baseline DR method. $\alpha$ increases in the rows and $t$ increases in the columns, while $k = 200$. 
Figure 5.19: Plots to understand the kinematics of stars. The first column shows the Toomre diagram, the second plots the Energy as a function of $L_Z$, and the last shows $\sqrt{J_R}$ as a function of $L_Z$. To avoid clutter, the top row shows only Classes 1–4, and the second row shows Classes 5–9. Note that there is an outlier (star ID: 15564913-6814532) in the upper-right corner (red circle) in all three panels of the second row.

Figure 5.20: Histogram of various abundances ([Fe/H], [Na/Fe], [Mg/Fe], [Al/Fe], [Si/Fe], [Ca/Fe], [Ti/Fe], [Cu/Fe], [Zn/Fe], [Sr/Fe], [Y/Fe], [Zr/Fe], [Ba/Fe], [Ce/Fe], and [Eu/Fe]) labeled by different Classes introduced in Sec. 5.3.
Figure 5.21: SDR results with different parameter settings using $t$-SNE as the baseline DR method. $\alpha$ increases in the rows and $t$ increases in the columns, while $k = 50$. 
Figure 5.22: SDR results with different parameter settings using $t$-SNE as the baseline DR method. $\alpha$ increases in the rows and $t$ increases in the columns, while $k = 100$. 

**Table:**

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>$t$</th>
<th>$k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>2</td>
<td>100</td>
</tr>
<tr>
<td>0.01</td>
<td>4</td>
<td>100</td>
</tr>
<tr>
<td>0.01</td>
<td>6</td>
<td>100</td>
</tr>
<tr>
<td>0.01</td>
<td>8</td>
<td>100</td>
</tr>
<tr>
<td>0.01</td>
<td>10</td>
<td>100</td>
</tr>
</tbody>
</table>

...
Figure 5.23: SDR results with different parameter settings using $t$-SNE as the baseline DR method. $\alpha$ increases in the rows and $t$ increases in the columns, while $k = 200$. 

- $\alpha=0.01, t=1, k=200$  
- $\alpha=0.01, t=4, k=200$  
- $\alpha=0.01, t=6, k=200$  
- $\alpha=0.01, t=8, k=200$  
- $\alpha=0.01, t=10, k=200$  

- $\alpha=0.03, t=2, k=200$  
- $\alpha=0.03, t=4, k=200$  
- $\alpha=0.03, t=6, k=200$  
- $\alpha=0.03, t=8, k=200$  
- $\alpha=0.03, t=10, k=200$  

- $\alpha=0.06, t=2, k=200$  
- $\alpha=0.06, t=4, k=200$  
- $\alpha=0.06, t=6, k=200$  
- $\alpha=0.06, t=8, k=200$  
- $\alpha=0.06, t=10, k=200$  

- $\alpha=0.07, t=2, k=200$  
- $\alpha=0.07, t=4, k=200$  
- $\alpha=0.07, t=6, k=200$  
- $\alpha=0.07, t=8, k=200$  
- $\alpha=0.07, t=10, k=200$  

- $\alpha=0.08, t=2, k=200$  
- $\alpha=0.08, t=4, k=200$  
- $\alpha=0.08, t=6, k=200$  
- $\alpha=0.08, t=8, k=200$  
- $\alpha=0.08, t=10, k=200$  

- $\alpha=0.09, t=2, k=200$  
- $\alpha=0.09, t=4, k=200$  
- $\alpha=0.09, t=6, k=200$  
- $\alpha=0.09, t=8, k=200$  
- $\alpha=0.09, t=10, k=200$  

- $\alpha=0.1, t=2, k=200$  
- $\alpha=0.1, t=4, k=200$  
- $\alpha=0.1, t=6, k=200$  
- $\alpha=0.1, t=8, k=200$  
- $\alpha=0.1, t=10, k=200$  

- $\alpha=0.12, t=2, k=200$  
- $\alpha=0.12, t=4, k=200$  
- $\alpha=0.12, t=6, k=200$  
- $\alpha=0.12, t=8, k=200$  
- $\alpha=0.12, t=10, k=200$
In this thesis, we have proposed several ways to support the exploratory analysis of high-dimensional data using projection techniques. In Chapter 2, we have proposed a sharpening step prior to dimensionality reduction (DR) to enhance the visual cluster separation of projections while keeping the other desirable requirements of such methods, such as ease of use, simple implementation, computational scalability, and out-of-sample (OOS) support. In Chapter 3 we have shown how clustering can be used in line with our method – Sharpened DR (SDR) – to support the labeling of clusters in 2D projections. Next in Chapter 4, we have presented several ways to further improve the scalability, ease-of-use, and stability of SDR. Finally in Chapter 5, we have presented an in-depth analysis of manually labeled clusters from an SDR projection of a recent astronomical survey of stellar abundances, GALactic Archaeology with HERMES Data Release 3 (GALAH+ DR3).

We conclude the thesis by revisiting the main research question stated in Sec. 1.4 and reflecting on how the previous chapters addressed this question. In this chapter, several aspects and limitations of the proposed work and possible directions for future improvements are discussed.

6.1 Addressing the Research Question

The research question we introduced in Chapter 1 is repeated below.

*How can we improve the quality of DR methods in terms of visual cluster separation (VCS) for exploratory data analysis of unlabeled, large, high-dimensional data sets?*

We have addressed this question along three different aspects: (a) improving VCS, which is the primary goal of this thesis; (b) allowing exploratory data analysis of unlabeled, large, high-dimensional data; and (c) applying the proposed method to the target application domain of this thesis, namely astronomical surveys of stellar abundances. We now discuss these aspects in order as follows.
(a) Improving VCS

We proposed a generic method that can be applied to any existing DR method instead of introducing a new DR method that improves VCS. The key idea of this generic method was to pre-condition the high-dimensional data prior to the application of the chosen DR method to sharpen the sample density in the high-dimensional space so that clusters in the lower-embedded space become visually more separated. This method is based on Gradient Clustering (GC), which estimates the sample density using Kernel Density Estimation (KDE) and iteratively shifts samples along the density gradient. In contrast to classical GC, where the sample density is estimated globally (based on all samples), GC is applied locally by estimating a local density of a neighborhood of data points.

In Chapter 2 we showed how sharpening is applied prior to a number of well-known DR methods to enhance VCS in the obtained projections. A qualitative and quantitative evaluation of the method showed that DR methods that have weak cluster separation benefit the most from this sharpening approach. This leaves the final selection of applying (or not) our SDR approach can assist the user to select the baseline DR method – if one already avails of a DR method with sufficient VCS, then the proposed approach may not produce desired results. Conversely, if one avails of a DR method with desirable properties – ease of use, scalability, ease of implementation, and/or stability – the SDR approach can be transparently and generically added to the selected DR method. By this addition of VCS, the chosen DR method keeps all its desirable properties by construction, since the preprocessing approach does not modify the subsequent DR method in any way; and VCS is added also by construction, as the data that the chosen DR method receives is better separated in its high-dimensional space.

We first applied SDR to an astronomical survey. Specifically, we applied SDR to a subset of the second release of GALAH (GALAH DR2) in Chapters 2, 4 and briefly demonstrated how SDR can assist domain experts in manually labeling clusters in unlabeled high-dimensional data, which was otherwise challenging using vanilla DR methods. We present a more in-depth analysis of the resulting clusters for a more-recent data release of GALAH (GALAH+ DR3) in Chapter 5.

(b) Exploratory analysis of unlabeled, large high-dimensional data

We next addressed the research question by supporting the other important and desired characteristics of DR defined earlier in Chapter 1.4 –
namely, scalability, ease-of-use, genericity, and OOS support. In more detail, the SDR method proposed in Chapter 2 improved VCS while keeping the desirable properties of the pre-conditioned DR method. However, by construction, adding a preprocessing step to any method influences the end-to-end pipeline in all its aspects. Specifically, the SDR method presented in Chapter 2 decreased the overall speed and ease of use (in terms of parameter setting) of the end-to-end method, as SDR is slow and introduces extra parameters that users need to tune.

As scalability is an important factor in analyzing large high-dimensional data sets, we first addressed this issue in Chapter 4 by Neural Network Projection (NNP), a method which implements both the sharpening step and the preprocessed DR method (to be chosen by the user) using neural networks. The resulting method, called SDR-NNP, significantly enhanced the scalability of SDR (and indirectly also improved the scalability of the baseline DR method following SDR). SDR-NNP is linear in the number of samples and dimensions. A trained SDR-NNP network can be used to compute up to a million samples and hundreds of dimensions in a few seconds on commodity GPU hardware. Moreover, fitting unseen samples on a trained SDR-NNP network (so-called inference mode) provides OOS support, which is another desired property of the proposed global pipeline.

Concerning ease-of-use support, SDR-NNP has three parameters (i.e., number of iterations for shifting points (t); number of nearest neighbors for density estimation (k)); and the learning rate for the update-rule for shifting points (α) inherited by SDR, and one parameter from SDR-NNP, the number of epochs (E). Especially the k and α parameters were found to be difficult to set, as they were highly interdependent in SDR. For these reasons, we proposed α-SDR-NNP to support the ease-of-use of SDR-NNP. α-SDR-NNP first applies a simple clustering algorithm, K-means, to acquire pseudo-labels of the high-dimensional data with a single user-set cluster-count parameter (Km). The Neighborhood Hit (NH) metric is then computed using these pseudo-labels and next used for shifting the samples. A qualitative and quantitative evaluation of α-SDR-NNP showed that Km was insensitive to small errors and was easier to control than tuning the three main parameters (α, t, and k) of SDR. Hence we address the main issue of ease of use (i.e., parameter setting).

While visual cluster separation in SDR (or SDR-NNP, α-SDR-NNP) allowed users to easily perform manual labeling, in Chapter 3 we showed examples of using clustering methods along with SDR to ease this labeling process for users. Several clustering methods such as K-means, Hierarchical Clustering, Density-Based Spatial Clustering of Applications with
Noise (DBSCAN), and Spectral Clustering were used after applying SDR. Clustering were applied to SDR and DR on labeled synthetic and real-world data sets and were qualitatively and quantitatively assessed and compared. We concluded in this chapter that SDR showed higher clustering performance compared with DR, which demonstrates again the visual cluster separation power of SDR. We also applied SDR to labeled human-activity data and results showed that the separated clusters from SDR were associated with different human motions (e.g., static versus dynamic), which were otherwise difficult to separate with a plain DR method.

(c) Exploratory analysis of stellar abundances

For the target domain of this thesis – analysis of astronomical data – previous attempts using only DR methods (e.g., t-distributed Stochastic Neighbor Embedding (t-SNE)) to manually label sample clusters using (only) stellar abundances have not been fully successful as mentioned in Chapter 1. However, we showed in Chapters 2 and 4 that using SDR results in easier manual labeling for astronomers. We suggested in these studies that the resulting manually-labeled clusters were associated with different components of our galaxy, such as thin- and thick-disk stars and halo stars. As a follow-up study, Chapter 5 has taken a step further to analyze the resulting clusters from SDR in depth, as the aim of the application-side of this thesis was to find the chemical composition of stars using SDR to understand the formation and evolution of the Milky Way. We applied SDR to a more-recent data release of GALAH (GALAH+ DR3) and results showed that the visually separate clusters were associated with a range of different subcomponents of the Milky Way, such as accreted stars, in-situ stars, and metal-poor or metal-rich thin- and thick-disk stars. These results have demonstrated that SDR is applicable to unlabeled real-world astronomical data sets where one does not explicitly know what the resulting clusters in the projection physically mean.

Short summary

In this thesis we proposed SDR and its extended versions, SDR-NNP and $\alpha$-SDR-NNP for exploratory analysis of unlabeled, large high-dimensional data by labeling visually separated clusters either manually or automatically. The proposed approach of using sharpening prior to DR yielded enhanced VCS. Furthermore, SDR-NNP and $\alpha$-SDR-NNP assisted users to explore unlabeled, large high-dimensional data sets by supporting other important aspects of DR such as scalability, genericity, ease-of-use, and stability.
6.2 LIMITATIONS AND FUTURE WORK

Finally, we now discuss the limitations and potential directions for future work based on the previous chapters of this thesis.

Data distortion

The key approach to the work of this thesis involves changing the values of the original data samples based on the point density. Samples are shifted towards the local maxima of their density to enhance data separation and next, the visual cluster separation in the projection of the data. Hence one may argue that this approach may distort the data due to the artificial shifts with potentially undesirable consequences. While this argument is correct in the strictest sense, such data distortions are not different (nor worse) than other existing distortions in comparable data-processing and/or visualization pipelines. As noted in Chapter 2, we have argued in this thesis that any DR method creates, by definition, non-trivial amounts of distorted data during its execution if the data is not located on a smooth 2D manifold (and even if the data is located on such a manifold the results depend on the DR method). Hence changes in the original data do not necessarily mean that the proposed methods are less trustworthy than any other plain DR method. Instead, the essential aspect and added-value of any projection method – including the proposed SDR – is that the resulting visualization helps users in completing given tasks involving the data at hand. Visual cluster separation (VCS) is an essential component of such tasks, e.g., in detecting sample clusters and further explaining and/or labeling these. A method with high VCS therefore demonstrably supports the aforementioned tasks (albeit introducing data alteration), whereas classical DR methods also convey a picture of the data which cannot accurately capture all data aspects but provide less VCS and thus are worse at supporting the aforementioned tasks. At a higher level, we have argued in this thesis that a DR visualization of high-dimensional data will never be able to accurately convey all data aspects by construction. Therefore, we suggest in this thesis that favoring VCS, for the price of altering the input data, is better than having no VCS even though the projection does not convey accurately the input data.

Parameter and baseline DR methods

As shown in Chapter 2, SDR was less effective when used in combination with baseline DR methods such as t-SNE and Uniform Manifold Approximation and Projection (UMAP) that already possess strong visual cluster
separation abilities. Due to the stochastic nature and limited ease-of-use (due to parameter setting) of these methods, these methods may not be among the preferred DR methods. Instead, we suggest applying SDR with baseline DR methods that exhibit poor VCS but have other desirable properties, such as ease of use, computational scalability, and implementation simplicity was suggested. As there is not a single DR method applicable to all kinds of data and the relative importance of various desirable requirements depends on the application context at hand, we recommended that users to explore SDR and its extensions with various DR methods to gain optimal results.

Setting parameters for $\alpha$-SDR-NNP was easier than for its predecessors SDR and SDR-NNP, as changing the single SDR-NNP parameter $K_m$ produced stable and predictable behavior in terms of the obtained number of visual clusters in the projection. However, there can be data sets where the user has no expected value for the parameter $K_m$. Therefore, exploring further on how to set $K_m$ to plausible values depending on the data set at hand is left for future work.

Selecting the optimal baseline DR method and setting its parameters both leave room for future work in assessing results of SDR. A fundamental problem here is that projecting high-dimensional data sets has no clear, universally valid ground truth to test the projection. We discuss many projection quality metrics throughout this thesis. However, as also previously discussed, these quality metrics are of a local nature and often do not capturing the essence of the task that the user wants to accomplish next using the projection. Hence designing new metrics for assessing the quality of a projection – or rather its fit for a given task – is a large and important open topic for DR research in general.

**Other aspects of DR**

While ($\alpha$-)SDR-NNP inherits several advantages of NNP, it also inherits some of the limitations of NNP. For instance, OOS is not supported for a data set with a completely different nature than those used for training. However, this limitation arguably exists for most machine-learning methods: this is the well-known ‘generalization problem’. While we argue in the thesis that this problem cannot be solved in a general case (as one cannot train a model to predict any sort of unseen data), we also argue that one could devise techniques to measure the difference of the unseen data to the training data and evaluate the potential errors (in terms of size and type) that the trained model will generate on the unseen data. This approach would immediately help users in understanding how certain they
can be if using a given trained model is sufficiently safe (or not) for a given unseen data set.

Other than the desired traits of DR suggested throughout this thesis, there are other traits that users may prefer to retain in the projection. For instance, there are outliers and subclusters. For outlier detection, the outlier preservation ability from the baseline DR method is passed on to SDR and its extended methods. When using baseline DR methods such as t-SNE and UMAP, however, SDR may result in outliers being merged into other point clusters even if the parameters are carefully tuned. Moreover, such outliers can be merged even for other baseline DR methods if the SDR parameters are set to enhance the separation too much. A careful selection of parameters is required for real-world data sets that have varying number of clusters and outliers. There is clearly room for improvement in terms of automatically setting the parameters based on user-prescribed guidelines concerning what should (or should not) be preserved in the resulting projection.

Finally, all of the proposed methods in this thesis assume that the data has no errors in the data acquirement or processing steps. However, real-world data sets, such as those in astronomy in particular, may intrinsically have uncertainties such as measurement errors and errors due to pre-processing the data. In some cases, random noise may even be sharpened producing overly segmented clusters. Therefore, it is important for the user to consider these uncertainties for a more accurate analysis. An interesting direction concerning this is to allow users to specify noise or uncertainty models of the available data and consider these when executing SDR and/or when depicting the final projection.

Applications

All of the proposed methods in this thesis are generic – they can be applied to any real-world high-dimensional data with quantitative attributes. Extensions to categorical attributes are arguably relatively simple by using e.g. one-hot encoding, which is a method to convert categorical attributes into a form that can be understood by machine learning algorithms [81, 188]. Additional applications of the proposed methods include assisting classifiers for image data so that the actual images corresponding to a projected point are displayed by the SDR enhancement. SDR and the extended methods can also be used in classifying astronomical data\(^1\), medical image classification (e.g., anomalies in brain), and cell-imaging

---

\(^1\) E.g., recent studies on SDR-NNP used for classification of stars, galaxies, and quasars by Lourens and Trager [138]
(e.g., detecting cancer cells) [128, 234]. In summary, SDR can improve visual cluster separation and can be applied to a wide range of applications. Finally, even though SDR-NNP and $\alpha$-SDR-NNP enhance the scalability of their predecessor SDR, speeding up the training step of both methods is of great interest for practical use-cases. The enhancement in scalability and ease-of-use of $\alpha$-SDR-NNP has made it possible to run the full GALAH data set of billions of samples that was previously impractical with SDR(-NNP). Therefore, deploying this method as a main tool for astronomers is of great interest\(^2\). The findings from Chapter 5 are also an intriguing start towards investigating the chemical diversity within the solar vicinity and should be followed up in more detail to unravel the mysteries of the evolution and formation of our Universe. In conclusion, the work of this thesis aims to open the gate to the practical scalable deployment of DR with strong visual cluster separation to the current big data challenges.

\(^2\) E.g., SDR(-NNP) is configured by astronomers for easier access in [139]
# List of Symbols

Symbols used throughout the thesis are listed below.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N$</td>
<td>Size or the number of samples, observations, or points</td>
</tr>
<tr>
<td>$n$</td>
<td>Number of dimensions, attributes, or variables</td>
</tr>
<tr>
<td>$D$</td>
<td>$n$-dimensional data set $D {x_i} \subset \mathbb{R}^n$</td>
</tr>
<tr>
<td>$P$</td>
<td>Projection function $P : \mathbb{R}^n \rightarrow \mathbb{R}^s$</td>
</tr>
<tr>
<td>$N_k$</td>
<td>Set of $k$-nearest neighbors</td>
</tr>
<tr>
<td>$\rho$</td>
<td>Kernel density</td>
</tr>
<tr>
<td>$t$</td>
<td>Number of iterations in sharpening</td>
</tr>
<tr>
<td>$k_s$</td>
<td>Number of nearest neighbors in sharpening</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>Learning rate in sharpening</td>
</tr>
<tr>
<td>$\upsilon$</td>
<td>Intrinsic dimensionality ratio (IDR)</td>
</tr>
<tr>
<td>$\Gamma$</td>
<td>Number of classes (ground-truth labels)</td>
</tr>
<tr>
<td>$CS$</td>
<td>Cluster separation</td>
</tr>
<tr>
<td>$S$</td>
<td>Scalability</td>
</tr>
<tr>
<td>$EoU$</td>
<td>Ease-of-use</td>
</tr>
<tr>
<td>$G$</td>
<td>Genericity</td>
</tr>
<tr>
<td>$OOS$</td>
<td>Out-of-sample</td>
</tr>
<tr>
<td>$Q$</td>
<td>Quality</td>
</tr>
<tr>
<td>$Q_h$</td>
<td>Neighborhood-hit</td>
</tr>
<tr>
<td>$Q_t$</td>
<td>Trustworthiness</td>
</tr>
<tr>
<td>$Q_c$</td>
<td>Continuity</td>
</tr>
<tr>
<td>$Q_j$</td>
<td>Jaccard set distance</td>
</tr>
<tr>
<td>$Q_r$</td>
<td>Shepard diagram correlation</td>
</tr>
<tr>
<td>$S$</td>
<td>Sharpening</td>
</tr>
<tr>
<td>$a$</td>
<td>Accuracy</td>
</tr>
<tr>
<td>$p$</td>
<td>Purity</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>$NMI$</td>
<td>Normalized Mutual Information</td>
</tr>
<tr>
<td>$H$</td>
<td>Entropy</td>
</tr>
<tr>
<td>$e$</td>
<td>Number of training epochs in ($\alpha$)-SDR-NNP</td>
</tr>
<tr>
<td>$k_m$</td>
<td>Parameter controlling the $k$-means process in $\alpha$-SDR-NNP</td>
</tr>
<tr>
<td>$[X/Y]$</td>
<td>Abundance (ratio) of an element $X$ with respect to another element $Y$</td>
</tr>
<tr>
<td>$[Fe/H]$</td>
<td>Metallicity or abundance (ratio) of iron with respect to hydrogen</td>
</tr>
<tr>
<td>$V_R$</td>
<td>Velocity of a star in the direction of galactic rotation</td>
</tr>
<tr>
<td>$V_z$</td>
<td>z-axis velocity of a star</td>
</tr>
<tr>
<td>$V_T$</td>
<td>Perpendicular y-axis velocity of a star</td>
</tr>
<tr>
<td>$V_{LSR}$</td>
<td>Local Standard of Rest (LSR) velocity</td>
</tr>
<tr>
<td>$L_Z$</td>
<td>Star’s angular momentum perpendicular to the Milky Way’s disk</td>
</tr>
<tr>
<td>$E$</td>
<td>Kinetic energy of a star</td>
</tr>
<tr>
<td>$J_R$</td>
<td>Radial action of a star</td>
</tr>
<tr>
<td>$g$</td>
<td>Gravity</td>
</tr>
<tr>
<td>$T_{\text{eff}}$</td>
<td>Surface temperature</td>
</tr>
</tbody>
</table>


I am deeply grateful to my supervisors, Alex, Scott, and Jos, for their mentorship, guidance, and support throughout my PhD journey. I am so fortunate to have had the opportunity to work with such enthusiastic and dedicated researchers.

Alex, I am indebted to you for your detailed feedback and suggestions, which helped me to improve my work immeasurably. Your insights and expertise were invaluable, and I am so grateful for your willingness to share them with me.

Scott, I am grateful for your constant support and for always keeping me on track. You were always there to offer a listening ear and a helping hand and I really enjoyed our weekly meetings.

Jos, I thank you for your valuable advice and for your willingness to co-supervise my project. I am so grateful for your insights and guidance, and I am so glad that I had the opportunity to work with you.

I would also like to express my sincere gratitude to the members of my reading committee, Michael, Thijs, and Christophe, for their time and valuable feedback on my thesis. Michael, I admire your vast knowledge and insights, and I enjoyed our coffee corner conversations. Thijs, I appreciate your thoughtful comments and suggestions, which helped me to improve my work. Christophe, I thank you for your willingness to serve on my reading committee.

Next, I cannot stress enough how I could not have made it through this journey without the emotional support from my dearest paronymphs, Jiwoo and Georg. Jiwoo, we have a long history of studying in the same institutions. I cannot write everything down, but I think you know already. You have been my mentor in science and I have learned a great deal from you ever since I met you as my TA during my bachelor’s degree. I will probably still learn many things from you even after my PhD. Georg, you were the first new person I met in Groningen, but I did not imagine that that coincidence would turn out to be this kind of relationship! Georg (and of course dear Pia), I thank you two for being there for me when we were having a hard time during our PhDs. I thank you for being a great listener and for always being there for me. Jiwoo and Georg, I enjoyed our
conversations that jumped in topics starting from (of course it has to start with) games as we are The Crew, to science, history, politics, and music. I am so grateful for your friendship and support. I would like to thank you both for always being there for me, for your encouragement, and for your friendship. I am so lucky to have you in my life.

I would also like to express my sincere gratitude to the SVCG and Kapteyn groups, where I learned so much from my dearest colleagues. Starting with Pieter (and Rocio), I am deeply grateful for inviting me into your inner circle. I can proudly say that you are my first Dutch friend I made in Groningen! I enjoyed our board game nights, lego talk, and small talk so much, and I really miss you both. Mateus, thank you for collaborating with me and I am so grateful for the opportunity to work alongside such an excellent scientist and engineer. It was such a joy to work on SDR-NNP with you and I have learned so much. Jiri, thank you for your valuable comments during presentations. I am grateful for your insights and expertise, and I admire your commitment to making the group a vibrant and supportive community. Thank you for checking in with me from time to time. Gerben (and Cynthia), I did not expect that we would end up meeting together not as four of us, but also with the little ones. I enjoyed our talks and board game evenings, and I always seem to be laughing my head off when I am with you guys! I am so grateful for your friendship and support. Kristiina and Anqi, I learned a lot from our group meetings. I admire your strength of mind and your dedication to your research. Lingyun, I am so grateful for your friendship and support. You were there for me during the early stages of my PhD, when I was confused with literally everything. Your emotional support and insights helped me to stay on track. I am so lucky to have you in my life. Venus (and Iris), I enjoyed chatting with you and always appreciated your kind words. Dennis, Chengtao, and Lorenzo, thank you for being there for me when I was a newbie. I enjoyed our coffee corner talks so much. Lucas, thank you for your kind advice. Htet, I enjoyed our chats! Bohuan, I enjoyed our talk on science! Thank you for sharing your knowledge with me. Steffan and Cara, I enjoyed our small talk and I am always grateful for your kind words. Marten, thank you for using SDR and taking it to another level. Antonio, Filip, Jun, Kuanhao, Sjoerd, Zikai, Mian, and Yanrui, I hope we can get to know each other more in the future. I am so grateful for the opportunity to have worked with such talented and dedicated colleagues mentioned above. I have learned so much from you all, and I am a better researcher and person because of it.
I would also like to thank my DSSC fellows, who I learned a great deal from. Samira and Hyoyin, I enjoyed our coffee corner talks and I learned a great deal from you both.

I am also grateful to the bernoulli and Kapteyn secretaries and staff members, Ineke, Elina, Sarah, Renske, Jan, Magda, Heleen, Christa, Martine, and Tanja, for their help and support with literally everything. I would especially like to thank Ineke, Elina, and Heleen for their help during the most difficult times. I would also like to thank my scientific advisor Stefania for her support on the DSSC project and community.

I am also grateful to Frank, Jonas, Hung, and other colleagues from CIT for the opportunity to work with you on generating synthetic data as an internship. I learned a great deal from the experience and it was an honor to be able to work alongside you. I would also like to thank Prof. Cheolsoo Park, who made this journey possible.

I think now is a good time to express my gratitude towards my amazing friends and family who have also supported me throughout this journey.

I would like to add a heartfelt thank you to my dearest friends. Matthijs and Jiufang, I am grateful to have such an amazing Dutch teacher and friend. Just talking with you two makes me forget about stressful things. Matthijs, thank you once again for translating the samenvatting! Seunghee, Yoo-jin, and Ji-eun, I am so grateful for your friendship and support. You have always been there for me, celebrating with me at my highest moments and supporting me through my lowest. You are the best friends a person could ask for.

I would also like to thank the Korean church community for their prayers and emotional support. I have always felt welcome and supported by the church community, and I am grateful for your prayers and kind words. Sangpil, Donghyun, and Arthur, thank you for sharing your novel ideas with me during our small (and seemingly crazy, but brilliant) talks. Doohyun and Soohee, thank you for the musical inspiration. I have learned so much from you, and I am grateful for your friendship.

Finally, I would like to say a big thank you to my family. No words can express my gratitude towards my whole family. Heejun, I literally could not have made it through this journey without you. You have always been my biggest supporter, and I am so grateful for your love and encouragement. Mum and Dad, thank you so much for raising me and for always believing in me. I would not be the person I am today without your support. Keehwan, I thank you for always being there for me, no matter what. Grandma, Grandpa, and my parents-in-law, I always learn so much from
you, and I cherish the moments we are together. Dear my aunt’s and uncle’s family, Heekyung, Hayoung, Inhwa, and Suhwan, thank you all for your prayers and your emotional support. I am so grateful for your love and support. Little Yuna, thank you for bringing joy and laughter into my daily life and I love you so much.

I am so blessed to have such amazing supervisors, colleagues, friends, and family in my life. I am truly grateful for all of you.

향상 은 마음을 다해 응원해 주셔서 정말 감사드리고 제 무엇과도 바꿀 수 없는 제 소중한 가족, 언제나 사랑합니다.
COLOPHON

This document was typeset using the typographical look-and-feel classicthesis developed by André Miede. The style was inspired by Robert Bringhurst’s seminal book on typography “The Elements of Typographic Style”. classicthesis is available for both \TeX and \LaTeX:

http://code.google.com/p/classicthesis/

Final Version as of July 4, 2023 (classicthesis).