Info Vis

Article

Information Visualization
2022, Vol. 21(3) 197–219
© The Author(s) 2022
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/14738716221086589
journals.sagepub.com/home/ivi
SAGE

Visual cluster separation using high-dimensional sharpened dimensionality reduction

Youngjoo Kim[1], Alexandru C Telea[2], Scott C Trager[3] and Jos BTM Roerdink[1]

Abstract

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.

Keywords

High-dimensional data visualization, dimensionality reduction, clustering, astronomy

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 2Dand 3D, in terms of visually well-separated point clusters in a scatterplot. While *t*-distributed Stochastic Neighbor Embedding (*t*-SNE[1]) 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.[2] Besides *t*-SNE, many other nonlinear DR techniques have been proposed, for example, Random Projection (RP),[3,4] Landmark Multidimensional Scaling (LMDS),[5] ISOMAP,[6] Sammon Mapping,[7] and Uniform Manifold Approximation and Projection (UMAP).[8] While such methods typically achieve a poorer visual cluster

[1] Bernoulli Institute for Mathematics, Computer Science and Artificial Intelligence, University of Groningen, Groningen, The Netherlands
[2] Department of Information and Computing Sciences, Utrecht University, Utrecht, The Netherlands
[3] Kapteyn Astronomical Institute, University of Groningen, Groningen, The Netherlands

Corresponding author:
Youngjoo Kim, Bernoulli Institute, Computer Science and Artificial Intelligence, University of Groningen, Nijenborgh 9, Groningen 9747 AG, The Netherlands.
Email: lyoungjookiml@gmail.com

separation than *t*-SNE,[1,9] they are computationally more scalable and simpler to implement and use.[10]

Espadoto et al.[10] 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, for example, neighborhood and distance preservation, computational scalability, or simplicity.

In this paper, 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 Dimensionality Reduction and Cluster Separation. 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.[11]

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.[1,3,5,8] 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) catalogs.[2,12–14] 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.[15–19]

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 paper is structured as follows. Section Related Work outlines related work in dimensionality reduction. Section Proposed Method details our method. Section Results compares our method qualitatively and quantitatively with standard DR on several synthetic and real-world data sets. Section Application to astronomical data shows a practical use-case with unlabeled real-world astronomical data. Section Discussion discusses several aspects of our method, including its cluster segregation power, data distortion, scalability, and limitations. Section Conclusion concludes the paper.

Related work

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

Dimensionality reduction and cluster separation

While DR has multiple goals such as data compression,[20] feature extraction,[21] and exploratory analysis,[22] 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: for example, 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[23]; hierarchical clustering requires defining a similarity threshold[24]; and DBSCAN asks for the minimum number of neighborhoods required to form a dense region.[25,26] 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 $D = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ be a set of n-dimensional observations (samples, points), $\mathbf{x}_i = [x_i^1 \quad x_i^2 \quad \cdots \quad x_i^n] \in \mathbb{R}^n$. Here, x_i^j $(1 \leqslant j \leqslant n)$ 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.[27] A projection function P allows one to reason about a data set $D \subset \mathbb{R}^n$ by visually interpreting its projection (scatterplot), which we denote as $P(D) = \{P(\mathbf{x}_i) | \mathbf{x}_i \in D\}$. 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 $P(D)$.[28,29] Note that the function P from $\mathbb{R}^n$ to $\mathbb{R}^2$ is, in general, many-to-one in terms of point locations, that is, points that have different coordinates in $\mathbb{R}^n$ can be mapped to the same location in $\mathbb{R}^2$. Yet, every sample point $\mathbf{x}_i \in D$ is mapped to a unique point $P(\mathbf{x}_i) \in P(D)$ 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 nD data sets, as labeling a point $P(\mathbf{x}_i) \in P(D)$ in the projection directly labels its corresponding point $\mathbf{x}_i \in D$ in the data set, and conversely.

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.[10] Given the above, we say that a projection P has good VCS when $H(P(D))$ is very similar to $H(D)$, that is, 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 nD 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.[1,8]

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.[15–18] 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.[19] 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,[17,18] 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.[1] 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.[19] 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.[19,30]

However, *t*-SNE's complexity is quadratic in the number of points.[31] While accelerated variants exist,[32–34] 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.[35] A more recent competitor, UMAP, has been introduced and used in astronomical applications,[8,36] but to date has not been widely applied and assessed by domain experts in that field.

DR in astronomy

While applications of DR include biomedicine, computer security, and various other fields,[37,38] our main application domain in this paper 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 catalog[12–14] contains more than 1.6 billion objects with tens of dimensions. Sifting through these big data catalogs 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.[39] Since its first application in astronomy,[40] PCA continues to be widely used by astronomers, for example, to explore the space of stellar elemental abundances and describe how many controlling parameters exist,[41] and to find clusters in that space.[42]

More recent studies show the use of *t*-SNE in astronomy.[2,43] Anders et al.[2] 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 Anders et al.[2]). We explore the above challenges of using *t*-SNE for data labeling in Section Application to astronomical data.

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

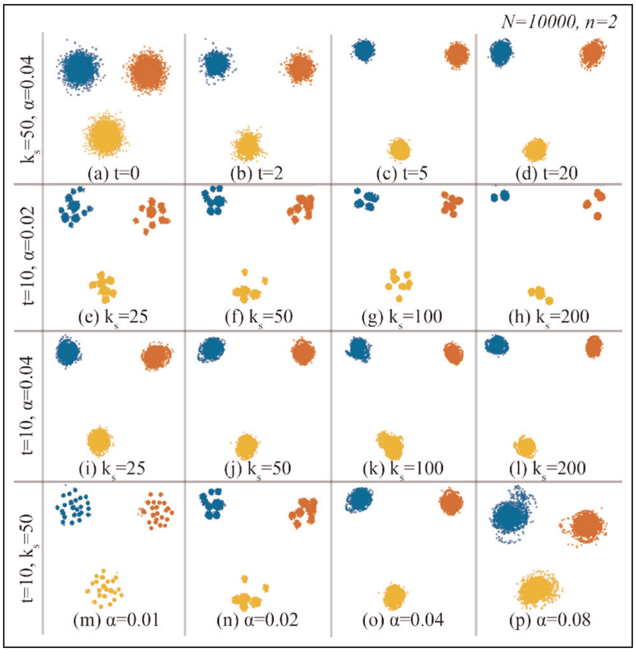

Figure 1. Effects of parameters used in LGC. 2D Gaussian data with 10K observations and three clusters (a) are used to show the effects of the number of iterations (T) as shown in (a–d), number of nearest neighbors (k_S) in (e–l), and learning rate (α) in (m–p). Points are color-coded based on their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

provides a clear visual separation of clusters for users to easily label these clusters.

Proposed method

We next present our method, which consists of two main steps: local gradient clustering (Section Local Gradient Clustering) and actual projections using RP, LMDS, *t*-SNE, and UMAP (Section Dimensionality Reduction Candidates for HD-SDR).

As outlined in Section Related Work, 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.[44–48] Such methods have previously been suggested to estimate modes of clusters, in combination with DR, to cluster and visualize high-dimensional data.[11] 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)[49] and iteratively shift samples upwards in the density gradient. For a data set $D = \{\mathbf{x}_i\}$, we define the multivariate kernel density estimator at location $\mathbf{x} \in \mathbb{R}^n$ by

$$\rho(\mathbf{x}) = \sum_{\mathbf{x}_i \in N(\mathbf{x})} K\left(\frac{\| \mathbf{x} - \mathbf{x}_i \|}{h}\right), \tag{1}$$

where $K(\cdot) : \mathbb{R}^+ \to \mathbb{R}^+$ is a radially-symmetric univariate kernel of bandwidth h. $N(\mathbf{x})$ denotes the set of samples $\mathbf{x}_i$ which affect the density ρ at location $\mathbf{x}$. In classical KDE,[44] $N(\mathbf{x}) = D$, that is, all samples affect all density locations. Another possibility is to use only samples closer to $\mathbf{x}$ than h, that is, $N(\mathbf{x}) = \{\mathbf{x}_i \in D : \| \mathbf{x} - \mathbf{x}_i \| < h\}$. This offers a better local control of the scale of patterns (clusters) formed by mean shift and also significantly accelerates the density estimation.[48] 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 $\mathbf{x}$ and its k_s-nearest neighbor in D. The free parameter k_s thus determines the simplification scale of the data set. More intuitively, all k_s-nearest neighbors of a point $\mathbf{x}$ are considered to be in the same cluster as $\mathbf{x}$.

After estimating ρ over D, we shift all samples $\mathbf{x}_i \in D$ for T iterations along the density gradient by the following update rule

$$\mathbf{x}_i^{next} = \mathbf{x}_i + \alpha \frac{\nabla \rho(\mathbf{x}_i)}{\max(\| \nabla \rho(\mathbf{x}_i) \|, \varepsilon)}, \tag{2}$$

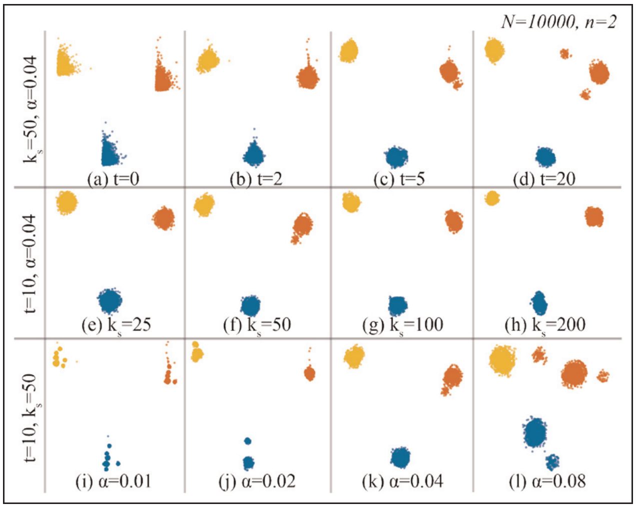

Figure 2. Effects of different parameters using 2D non-Gaussian (log-normal, $\mu = 0$, and $\sigma = 1$) data with 10 K observations. The effects of the parameters are similar to those in Figure 1. However, LGC with too large values of T and α is prone to outliers (long tails), as shown in (d) and (l). This problem can be solved by setting a larger value of k_s..

where α is the "learning rate," which determines the convergence speed of the process, and $\varepsilon = 10^{-5}$ 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.[50] This kernel yields smaller movements (shifts) as compared to a Gaussian kernel, thereby favoring the stability of the process. Note that equations (1) and (2) are coupled, as we estimate the gradient $\nabla \rho$ (equation (1)) after every iteration. This means that we perform the nearest neighbor search for every iteration as in Hurter et al.[48] In contrast, classical Gradient Clustering (GC)[44] 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), by analogy with Gradient Clustering (GC). The key added value of LGC discussed in this paper is its preconditioning of the data that leads to better results of DR techniques.

Figures 1 and 2 show the effect of the free parameters T (number of iterations), k_s (number of nearest neighbors), and α (learning rate) for LGC. Color encodes ground-truth labels, which are known for these data sets. Each row of Figures 1 and 2 shows the results of varying a single parameter, with the other two parameters fixed. The data set D contains synthetic Gaussian random data ($N = 10\,\mathrm{K}$ and $n = 2$) for Figure 1 and non-Gaussian (log-normal, $\mu = 0$, and $\sigma = 1$) random data ($N = 10\,\mathrm{K}$, $n = 2$) for Figure 2. We use $n = 2$ to demonstrate the effect of each parameter. 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 α: Controls the *speed* of shifts and affects the degree of segmentation, see the bottom rows of Figures 1 and 2. If α is too large, points move too far and can overshoot the mode of a cluster during LGC as shown in Figure 1(p) (see also Section IV-A in Fukunaga and Hostetler[44]). Conversely, too small α-values yield too small shifts (Figure 1(m)) and thus can result in an oversegmentation of the data (too many small clusters). The interconnection between α and k_s is discussed further in Section Parameter setting.

Nearest neighbors **k_s:** Controls how *localized* a shift is. Both k_s and α affect the degree of segmentation; yet, without choosing an appropriate α, k_s may not significantly affect the segmentation, as shown in the second and third rows of Figure 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 Noise-free data); a too large value of k_s increases the number of nearest neighbor searches resulting in slower computation (see Section Scalability).

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 Scalability). Setting $T = 10$ for all experiments in this paper 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 Figures 1 and 2(d) become fuzzier compared to those in Figures 1 and 2(c). This can be solved by using a smaller value of α. A similar issue is solved by decreasing the advection step in time.[48] 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 α to control the sharpening step after fixing the values of k_s and T. The effects of α on speed are discussed separately in Section Scalability. We implemented LGC in C++ for higher-speed performance, using Nanoflann[51–53] for the nearest neighbor search in $\mathbb{R}^n$. We have also evaluated other nearest-neighbor search algorithms (see Section Scalability). Our code is publicly available.[54]

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 Local Gradient Clustering). 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,[55] 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 nD 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.[1,3,5,8] The quantitative survey of DR methods of Espadoto et al.[10] 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,[36] 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 Espadoto et al.[10] (see Supplemental Material and Section Limitations and future work).

We briefly introduce the selected DR techniques used in this paper. Note that *t*-SNE was already explained in Section Related Work.

Random Projection (RP): Although nonlinear projections achieve better distance preservation for high-dimensional data,[6,56] 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.[3,4,57] 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.[3,4]

Landmark Multidimensional Scaling (LMDS) is a nonlinear variant of Multidimensional Scaling.[58] 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.[5] For completeness, we mention that we also used LMDS with increasingly more landmarks and obtained visually similar results (not included here for brevity).

Uniform Manifold Approximation and Projection (UMAP) is a recent competitor to *t*-SNE due to its strong separability of clusters. UMAP's model assumes that the data is close to being uniformly distributed on a Riemannian manifold; the Riemannian metric is locally constant; and the manifold is locally connected.[8] UMAP aims to find the lower, that is, two-dimensional embedding with a topological structure that best represents the fuzzy topological structure of the original data.[8]

Results

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

Synthetic data: qualitative evaluation

We generated Gaussian random data consisting of five clusters ($N = 5\,\text{K}$, $n = 20$) 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 3. The five trials are later used for a quantitative evaluation in Section Quantitative Evaluation. For synthetic data type (5), we add Gaussian noise using the standard deviation (σ) calculated by the definition $\text{SNR} = 10log_{10}P_s/\sigma^2(dB)$, where P_s is the power of the signal.

In Figure 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," that is, SRP, SLMDS, S*t*-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 α is found empirically by searching the fixed range $[0, 1]$ following the explanations in Section Local Gradient Clustering. This led us to using $\alpha = 0.04$ for SRP and SLMDS and $\alpha = 0.01$ for S*t*-SNE (perplexity $= 50$) and SUMAP.

Figure 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 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 3(g)), SUMAP performs worse than UMAP in separating subclusters using even a small $\alpha = 0.01$. Finally, in the fifth row of Figure 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.

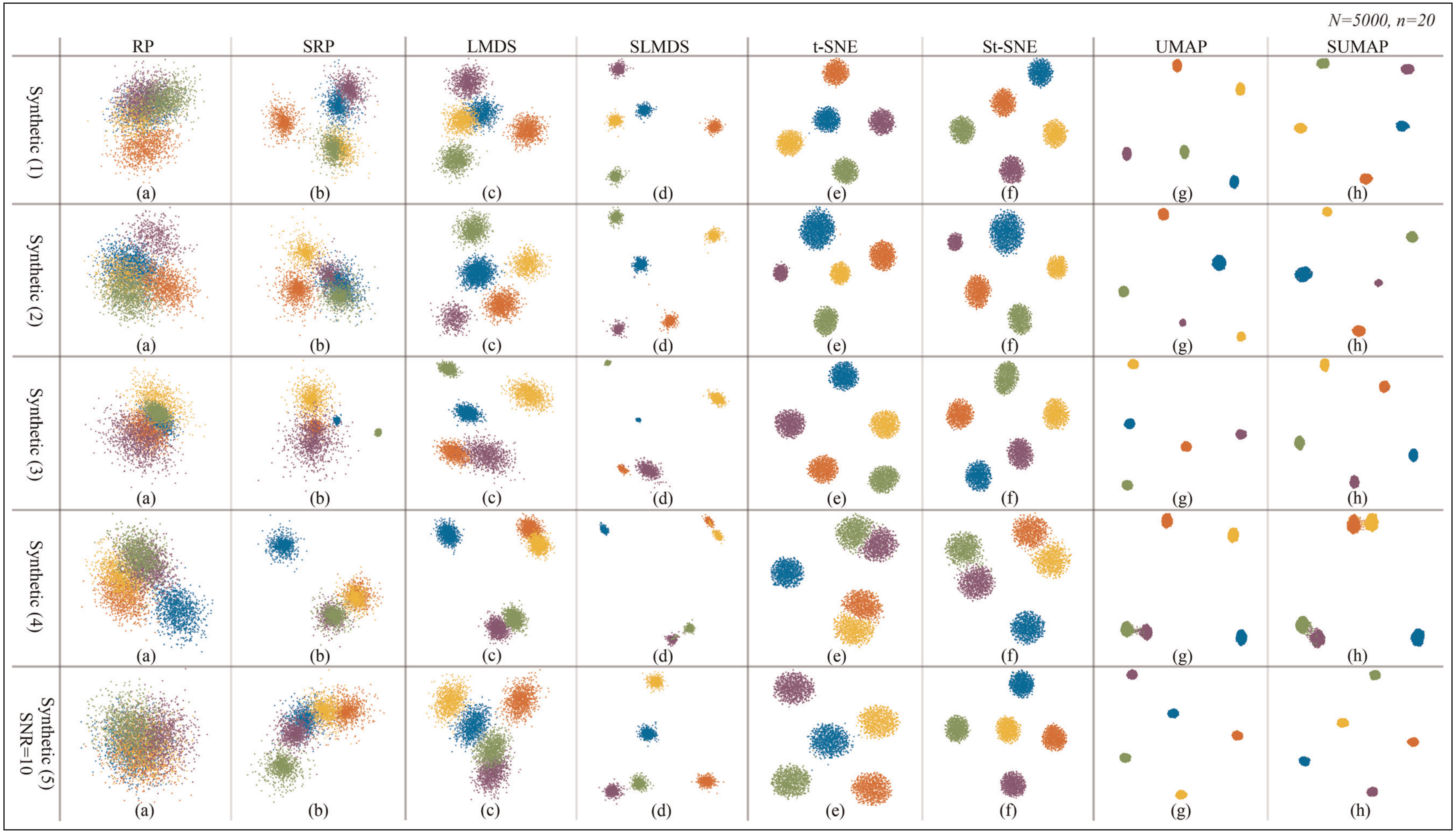

Figure 3. Comparison of DR and HD-SDR using five different types of synthetic data. DR methods are ordered from left to right based on how well clusters are separated. The samples are colored by the ground-truth labels of five different clusters for visual examination purposes. The results for SRP and SLMDS have been obtained with $\alpha = 0.04$ and for S*t*-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. k_s..

Table 1. Trait values for real-world data sets.

Data sets	Size (N)	Dimensionality (n)	IDR (v)	Classes (g)	Subclasses (g_{sub})
WiFi	Medium (2000)	Low (7)	High (0.6667)	Medium (4)	–
Banknote	Medium (1327)	Low (4)	High (0.5)	Small (2)	–
Olive Oil	Small (572)	Low (8)	Medium (0.1250)	Medium (3)	Large (9)
HAD	Large (24,075)	Medium (60)	Low (0.0167)	Medium (5)	–

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.

Data sets and their traits. We characterize data sets using the traits discussed in Espadoto et al.[10] 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* ($N \leqslant 1000$); *medium* ($1000 < N \leqslant 3000$); and *large* ($N > 3000$).

Dimensionality n: The number of dimensions, having three ranges: *low* ($n < 10$); *medium* ($10 \leqslant n < 100$); and *high* ($n \geqslant 100$).

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* ($0.1 \leqslant v < 0.5$); and *high* ($0.5 \leqslant v \leqslant 1$).

Classes g: The number of classes (ground-truth labels), having three ranges: *small* ($g \leqslant 2$); *medium* ($2 < g \leqslant 5$); and *large* ($g > 5$). We separately measure if the data has sub-classes and count these in g_{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 paper.

Table 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.[59] Each sample (banknote) is labeled as genuine or forged, and the data set is used to train classifiers to predict this label.[60] Projections are used to assess classification: If they show clearly separated different-label clusters, then the features can very likely discriminate between the labels.[28] We know this data set is easy to classify with accuracy close to 95%,[60,61] 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.[59,62,63] The data set has $N = 2000$ samples with $n = 7$ dimensions and is known to have four well-separated clusters.[64]

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.[65–67] The data is used to classify five motion-related human activities (sit, stand, walk, run, and dance).

HD-SDR applied to real-world data sets. Figure 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 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 4, last row). For DR methods that exhibit a strong cluster separation, that is, t-SNE and UMAP, the sharpening method improves the visual separation of clusters only by a small degree. Furthermore, S*t*-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 several subclusters, which are also found in recent work (compare Figure 4 second row with Figure 12(a) in).[11] For the Olive Oil data set, the subclusters are already known (see Section Data sets and their traits), which is why we color code its projections using both class and sub-class labels (Figure 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 Synthetic data: qualitative evaluation, synthetic data type (4)).

We also see an oversegmentation in HAD data projections. Oversegmentation is worse for S*t*-SNE and SUMAP than SRP and SLMDS, as shown in (e)–(h). This can be solved by using a larger α or by changing k_s (explained in Section Local Gradient Clustering). Further discussion of over- and under-segmentation is given in Section Discussion.

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 4, first column). On the other hand, S*t*-SNE and SUMAP show more oversegmentation than SRP and SLMDS, which suggests that LGC amplifies oversegmentation existing in a base projection.

Quantitative evaluation

While there are perception-based evaluations with extensive user studies on projection methods,[68] we evaluate here the projection methods quantitatively using quality metrics. As explained in Section Dimensionality Reduction and Cluster Separation, 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 $Q : (D, P(D)) \to \mathbb{R}^+$. 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 $Q(LGC(D), P(LGC(D))) \geqslant Q(D, P(D))$ 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.[10,69] 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.[70] Formally put:

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

$$Q_c(k) = 1 - \frac{2}{Nk(2N - 3k - 1)} \sum_{i=1}^{N} \sum_{j \in V_k(i)} (\hat{r}(i,j) - k), \tag{4}$$

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.[70] Q_t and Q_c lie in the range of 0 (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.[69,71] We average Q_j over all points, leading to

$$Q_j(k) = \frac{1}{N} \sum_{i=1}^{N} \frac{|W_k^2(i) \cap W_k^n(i)|}{|W_k^2(i) \cup W_k^n(i)|}, \tag{5}$$

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.[72,73] It ranges between $[0, 1]$ and is defined as

$$Q_h(k) = \frac{1}{N} \sum_{i=1}^{N} \frac{|G_k^n(i)|}{k}, \tag{6}$$

$$G_k^n(i) = \{j | g_j = g_i, j \in W_k^2(i)\}, \tag{7}$$

with $W_k^2(i)$ defined as earlier and g_i being the ground-truth labels (classes) of points i. Q_h is often used in classifier evaluation.[10] A discussion on the interpretation of Q_h, Q_j, Q_t and Q_c is given next in Section Evaluation of HD-SDR when analyzing the values of these metrics for both synthetic and real-world data.

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

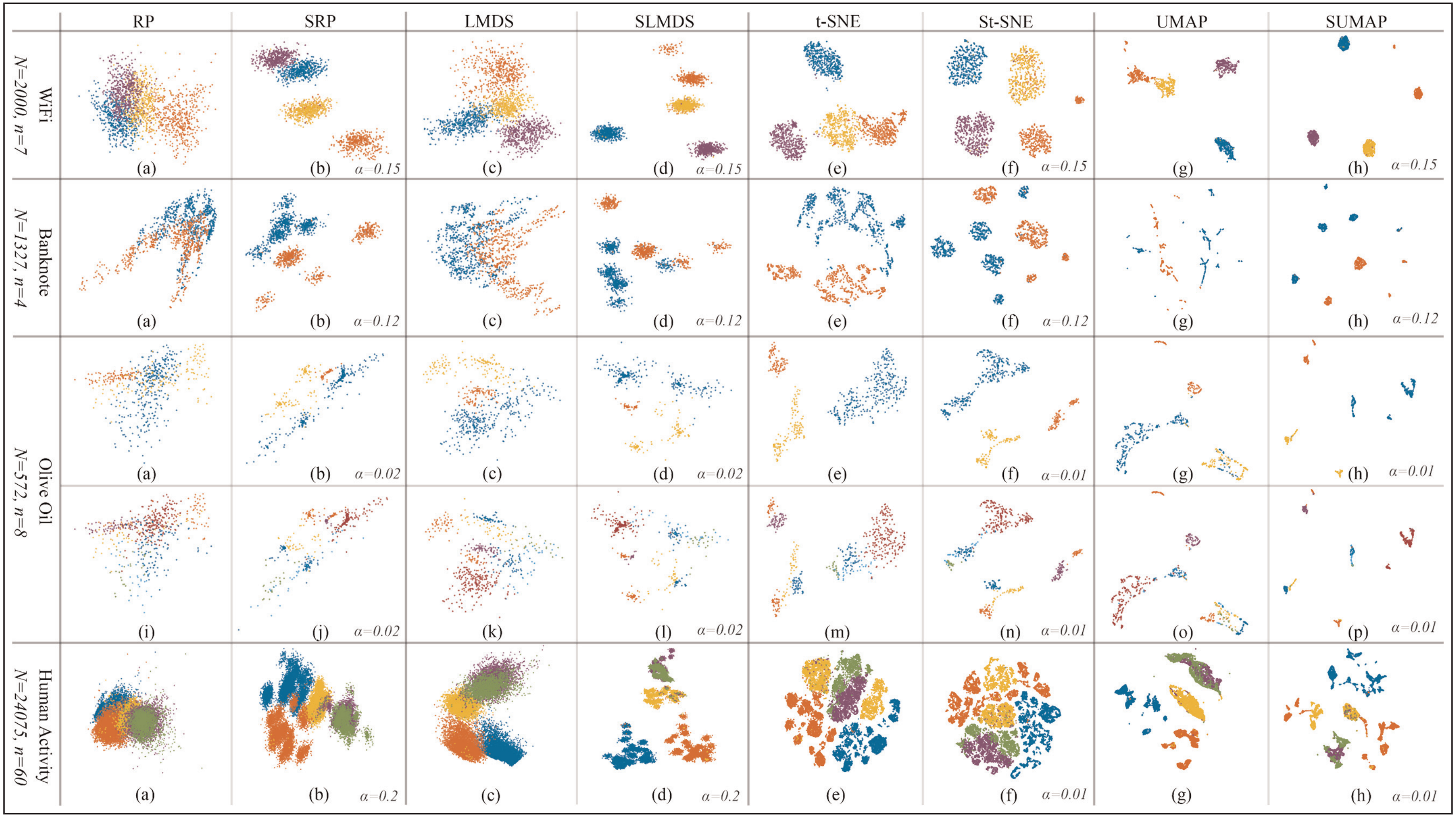

Figure 4. Comparison of DR and HD-SDR using four different real-world data. DR methods are ordered from left to right based on how well clusters are separated. The samples are colored by their ground-truth labels for visual examination purposes. Note that the sharpening method significantly enhances 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. k_s..

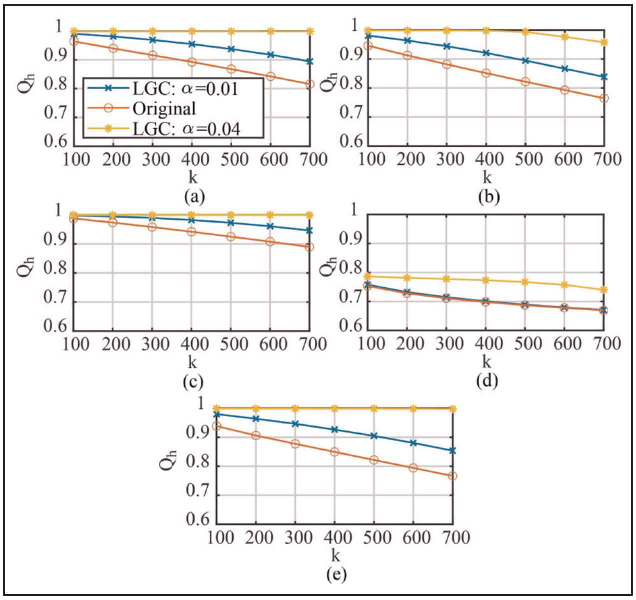

Figure 5. Comparison of neighborhood-hit (Q_h) for sharpened data and original data of the five different types of synthetic data used in Figure 3. For all synthetic data sets, Q_h is always higher for the sharpened data as compared with the original data. We also note that Q_h for sharpened data is higher when clusters are more separated ($\alpha = 0.04$ compared with $\alpha = 0.01$).k_s..

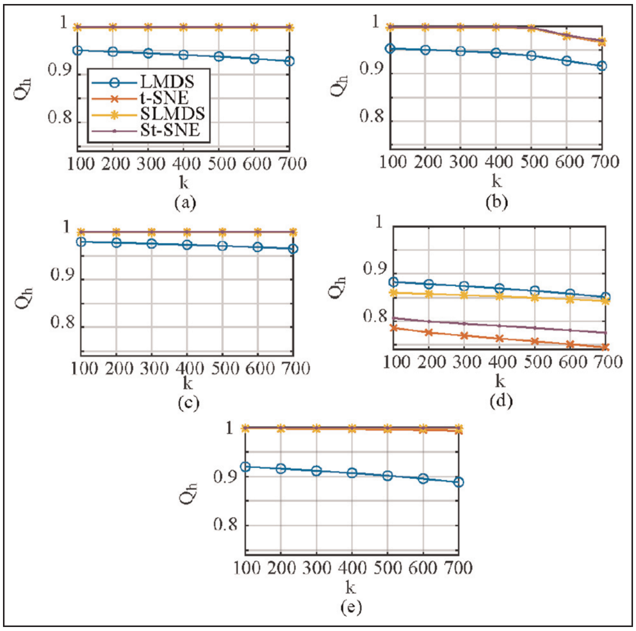

Figure 6. Comparison of neighborhood-hit (Q_h) for DR and HD-SDR of the five different types of synthetic data used for Figure 3. Note that S*t*-SNE, *t*-SNE, and SLMDS yield high Q_h-values near one for (a–c) and (e), which suggests that the corresponding labels of the *k*-size neighborhoods are well-preserved 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 3(a) to (d). More results including Q_t, Q_c, and Q_j for the five synthetic data sets can be found in the Supplemental Materials.

VCS, we expect $Q_h(LGC(D), P(LGC(D)) \approx 1$ and being larger than $Q_h(D, P(D))$.

Figure 5 shows the average Q_h over our five data set types for different *k*-values (see Section Synthetic data: qualitative evaluation). 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 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.

Evaluation of HD-SDR. We next evaluate Q_h for LMDS, SLMDS, *t*-SNE, and S*t*-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 for the HD-SDR methods (SLMDS and S*t*-SNE) indicates that our proposed sharpening yields better VCS than the original DR methods.

Synthetic data: Figure 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), S*t*-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 S*t*-SNE, but both values are close to one. This is in line with the projections in Figure 3 (third row) which show well-separated clusters for both *t*-SNE and S*t*-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 3, which is why Q_h is lower for SLMDS than LMDS. On the other hand, S*t*-SNE creates a slightly better separation of subclusters than *t*-SNE in Figure 3, which is why Q_h is higher for S*t*-SNE than for *t*-SNE in Figure 6(d). Furthermore, Figure 6(e) shows that our method is noise-resistant up to SNR = 10. Supplemental Figure 1 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 Supplemental Materials. All in all, Figures 3 and 6 show that our sharpening yields

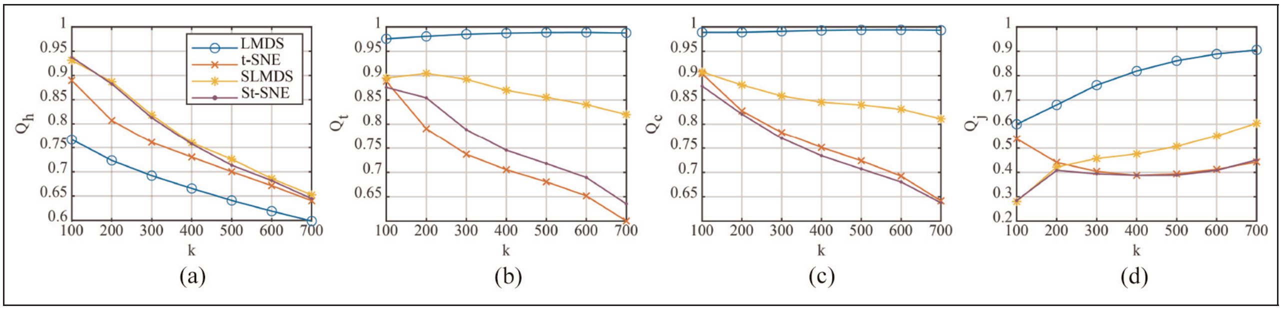

Figure 7. Results of four neighborhood-based quality metrics for Banknote data: (a) Neighborhood-hit (Q_h), (b) Trustworthiness (Q_t), (c) Continuity (Q_c), and (d) Jaccard set distance (Q_j). Note that Q_h is consistent with results from Figure 3 and best represents the visual cluster separation, whereas 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 Supplemental Materials.

well-separated visual clusters but is less effective for data with sub-cluster structure. Improvements aimed at sub-cluster data are discussed in Section Discussion.

Real-world data: Figure 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 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 4, second row).

Figure 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 4 shows that Q_h decreases with k for all studied methods. This is expected and in line with Figure 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 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 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 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 4(e) and (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 $k \geqslant 400$. This can be explained by the significant oversegmentation exhibited in the projections (last row of Figure 4(e) and (f)).

Supplemental Figure 8 complements Figure 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.[69,70] This is also visible in Supplemental Figure 8: 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 dimensionality curse. As

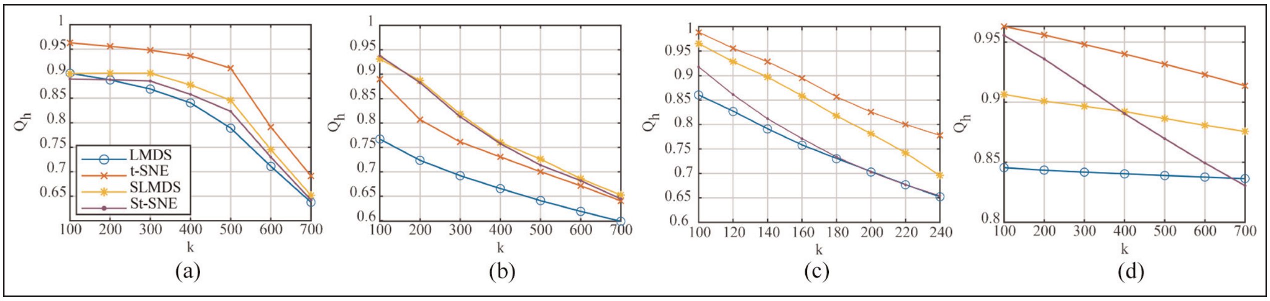

Figure 8. The neighborhood-hit (Q_h) metric for DR and HD-SDR using labeled real-world data sets with different values of k. (a) WiFi, (b) Banknote, (c), Olive oil, and (d) HAD. We note that Q_h is lower for St-SNE than for t-SNE for most values of k in (b–d) However for LMDS, which produces a weaker separation of clusters than t-SNE, SLMDS produces a higher value of Q_h compared with LMDS for all data sets.

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).[74] 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 4). However, visual cluster separation does not increase when oversegmentation occurs. The oversegmentation issues are discussed next in Section Discussion. 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.

Application to astronomical data

As a specific use case to show that VCS improves by our HD-SDR method, we aim to separate 10 K 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 catalogs – 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.[2] 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[2] 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).[12,13] 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.[14]

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 25 K 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.[14] From the 76,270 credible samples, we randomly select N = 10 K samples to project using RP, LMDS, t-SNE, and UMAP, and their sharpened versions. We use the same n = 10 attributes, that is, the stellar abundances [Fe/H], [Mg/Fe],[Al/Fe],[Si/Fe], [Ca/Fe], [Ti/Fe], [Cu/Fe], [Zn/Fe], [Y/Fe], and

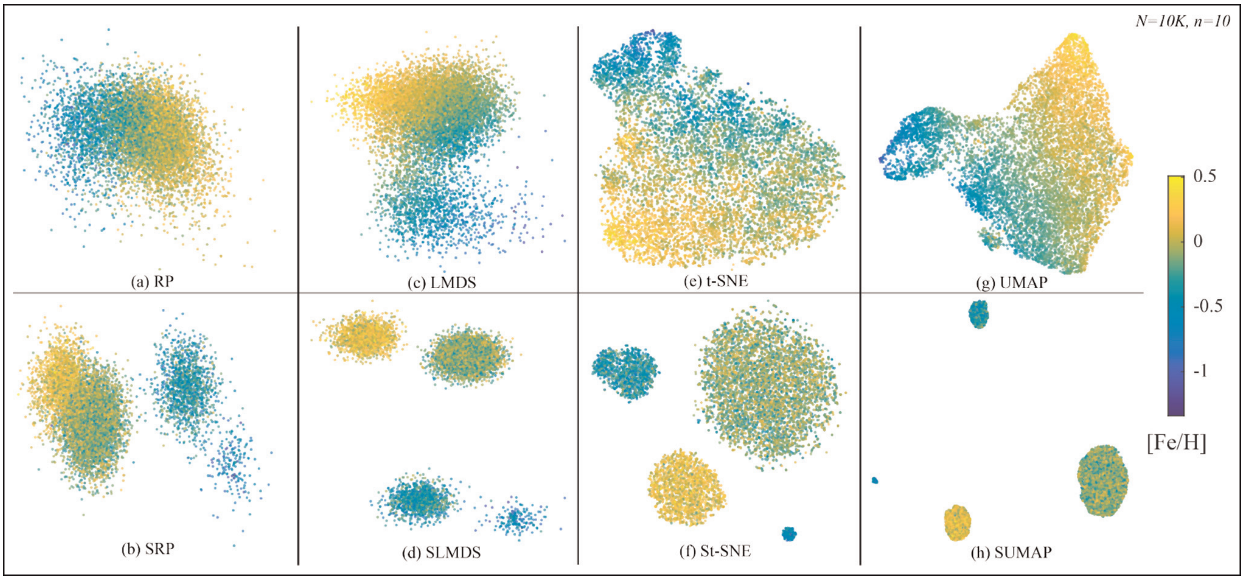

Figure 9. We compare DR and HD-SDR (RP, LMDS, *t*-SNE, and UMAP) for an unlabeled astronomical data set with no ground-truth labels: GALAH DR2 (*N* = 10 K with *n* = 10). The projections are color-coded by one of the input values, [Fe/H], so that astronomers can further analyze the data.[2] The learning 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, S*t*-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.

[Ba/Fe], as in a similar data set visualized by *t*-SNE,[2] for comparison purposes.

We present the resulting projections using HD-SDR and DR in Figure 9(a) to (h). The projections are color-coded by one of the input values, [Fe/H], so that astronomers can further analyze the data as in Anders et al.[2]; 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 S*t*-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 9(e) with figure 2 in Bleha and Obaidat[21]). 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, that is, [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 9, we further analyze the projection using SLMDS of this data set and 2D scatter plots of three abundances (Tinsley diagram[75]) with [Fe/H], [Mg/Fe], and [Ba/Fe]) in Figure 10. Note that the same analyses have been shown for S*t*-SNE and SUMAP in the Supplemental Materials (SRP has been excluded because the subclusters are inseparable). Due to the clear separation of four clusters using SLMDS, a domain-expert is able to manually assign four different class labels to the clusters, as shown in Figure 10(a). Next, we color-code the Tinsley diagram by the newly acquired labels (Figure 10(b) and (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.[76]

Discussion

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

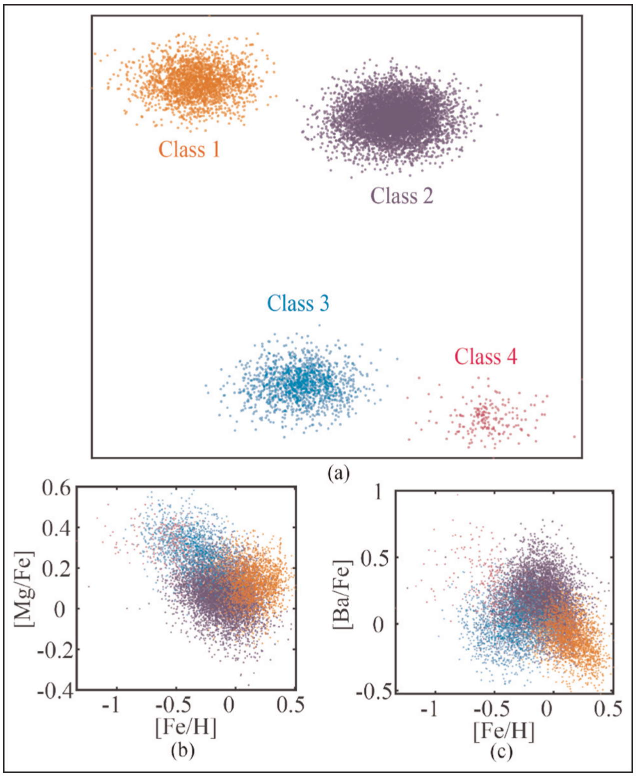

Figure 10. (a) SLMDS projection of the GALAH DR2 sample with clusters visually labeled by one of us. The labeled clusters in SLMDS help domains experts to further analyze the data as follows: (b) Tinsley diagram[75] shows the abundance of magnesium as a function of the iron abundance, used to interpret 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 S*t*-SNE and SUMAP are shown in the Supplemental Results; the clusters in SRP Figure 9(b) are not easily separated, thus excluded.

Scalability

Speed. Figure 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 $N \in \{10\,\text{K}, 20\,\text{K}, 30\,\text{K}, 40\,\text{K}, 50\,\text{K}\}$. For this plot, we used $\alpha = 0.1$. LGC is mainly affected by n, due to the nearest neighbor search, which is of order $O(N n \log n)$.

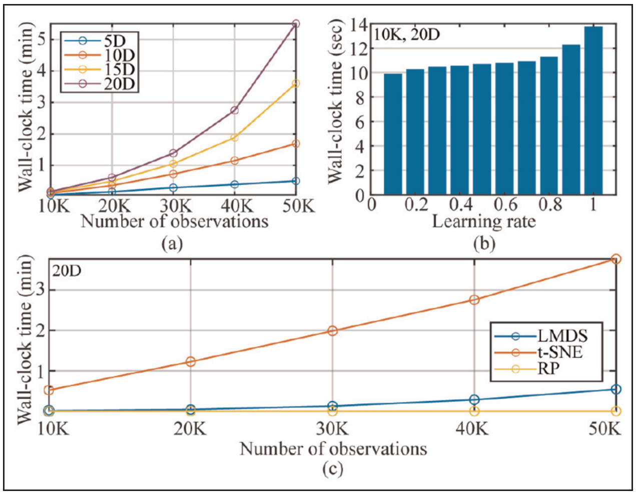

Figure 11. (a) Wall-clock timing of LGC, for different number of observations along the *x*-axis and varying dimensions up to 20D. (b) Wall-clock time measurement of LGC on a 10 K (20D) data set, using different learning rates α. (c) Wall-clock time measurements of RP, LMDS, and *t*-SNE applied to *n*D LGC data. We note that the wall-clock time of LGC increases with increasing α 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 Supplemental Materials.

The overall time complexity of LGC is $O(n\,T\,N \log N)$. LGC takes over 5 min to compute for data with 20D and 50 K samples (Figure 11(a)). Applying LGC to data with hundreds of dimensions may be impractical for end-users. A possible solution is outlined in Section High-dimensional data.

Figure 11(b) shows how speed depends on the α parameter using a data set with $N = 10\,\text{K}$ samples and $n = 20$. As α 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 ρ (equation (1)). Performing the kNN search for every iteration (instead of just once as in the original gradient clustering algorithm[44]) 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[53] by approximated and parallelized versions such as FLANN.[51,52] Multiple random projection trees (MRPT)[77,78] 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. Separately, the sample shift (equation (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.[79]

Figure 11(c) shows the wall-clock timings for LMDS, *t*-SNE, and RP on *n*D 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 50 K 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 Supplemental Materials.

High-dimensional data. Section Speed 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 12 and 13 illustrate this. Here, Human Activity Recognition (HAR) data[59,80] 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 12). Then, we use HD-SDR on this 10-dimensional data, with $\alpha = 0.2$. The obtained projections using SRP, SLMDS, and S*t*-SNE (Figure 13 bottom row) all exhibit improvements over their original counterparts (Figure 13 top row). In particular, we can easily see that cluster (1) of SLMDS (Figure 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 Limitations and future work next).

Data distortion

Our method addresses the cluster separation problem by shifting points in the original space, which may cause data distortions. Section Evaluation of LGC 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.[10] 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, for example, 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.

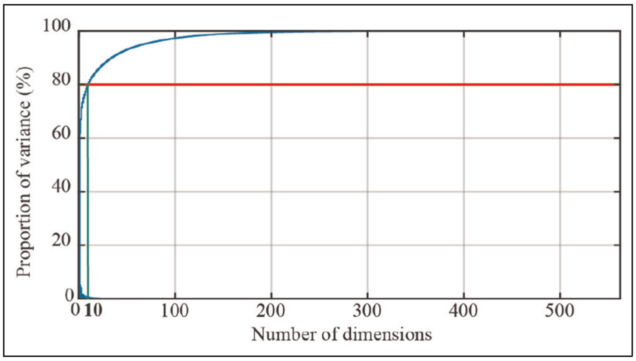

Figure 12. Proportion of total variance explained by each component when using PCA on Human Activity Recognition (HAR) data. The HAR data set (561 dimensions) can be reduced to 10 dimensions while keeping 80% of the variance.

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.[11] use mean shift, which is closely related to LGC, to create DR projections. They construct an *explicit* clustering of a data set D, $\cup_i C_i = D$, 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.[22,81] In contrast, we do not require an explicit clustering of the data to partition D to project it piecewise; 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 (equation (2)). This is different from classical mean shift[44] as used in,[11] where $\nabla\rho$ is computed from the initial density estimate and then used unchanged during the update (equation (2)). Updating $\nabla\rho$ leads to faster cluster separation, especially for noisy data.[48,79]

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

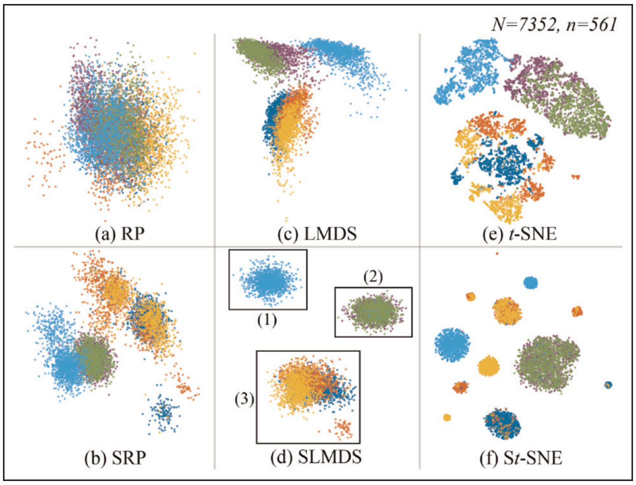

Figure 13. Comparison of DR and HD-SDR on HAR data with dimensions pre-selected using PCA as in Figure 12. Note that the separation of clusters is slightly improved for HD-SDR as compared with DR (although subclusters are visually less separable). We note that nearly all samples in cluster (1) are from 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. .

Preservation of outliers

Figure 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 5 K points randomly generated and uniformly distributed within radii $R \in \{1, 2, 3, 4, 5\}$.

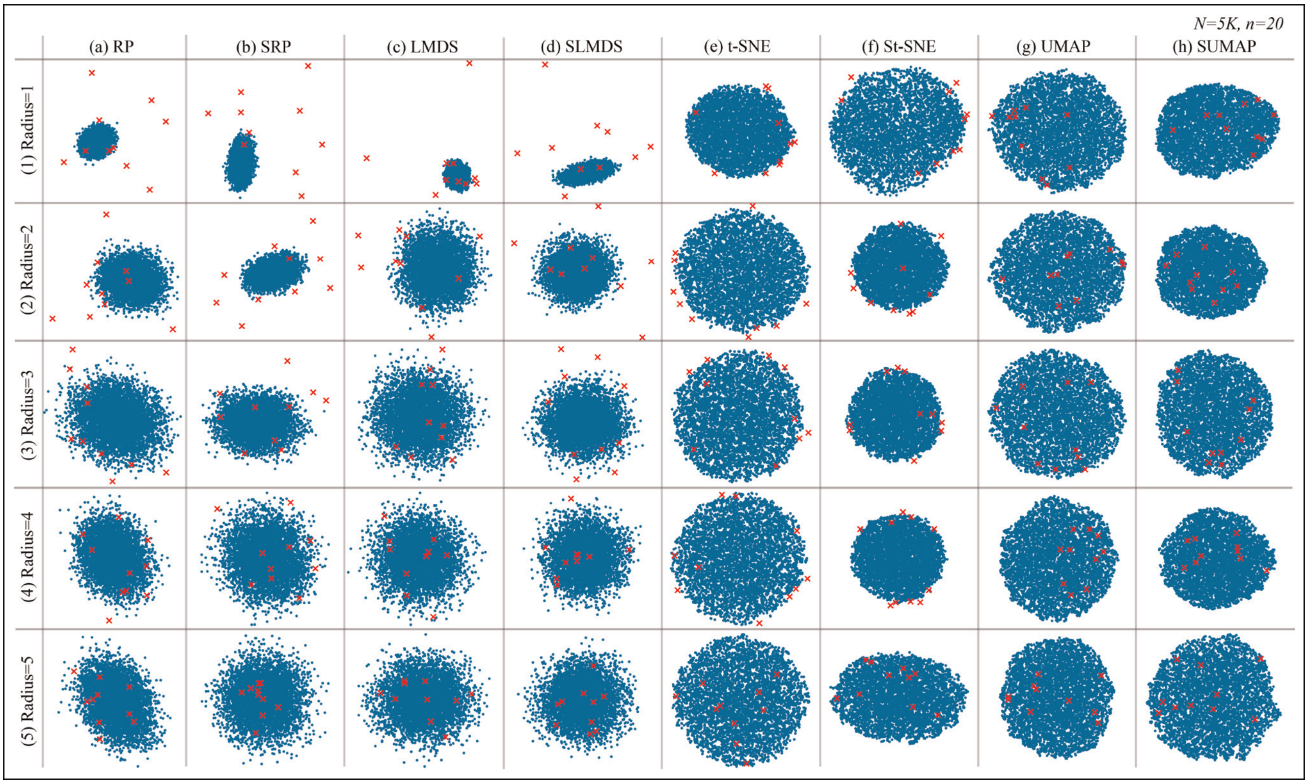

Figure 14. Comparison of DR and HD-SDR using 10 outliers (red crosses) and 5 K random points uniformly-distributed in a 20-dimensional hypersphere centered at the origin and with different radii $R \in \{1, 2, 3, 4, 5\}$. The distance between an outlier and the origin is always five. Note that *t*-SNE 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. .

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 Dimensionality reduction for labeling). As Figure 14 (rows 1 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 Parameter setting further discusses parameter setting.

Limitations and future work

Undersegmentation and oversegmentation. In some fields, including subfields of astronomy, domain experts are interested in finding *substructures*,[82] which relates to the *undersegmentation* issue. As Figure 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 4, fourth row). Oversegmentation is a known aspect of mean shift methods.[48] 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 α 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.

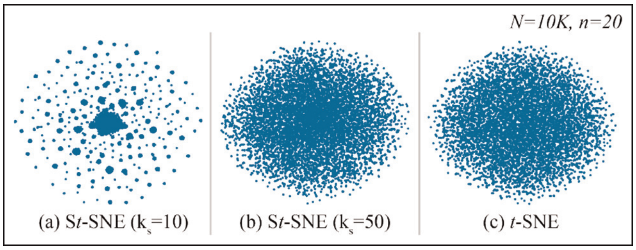

Figure 15. Results of DR and SDR with varying k_s-values for sharpening. The synthetic data set consists of 10 K randomly generated samples in 20D. Note that (a) S*t*-SNE shows oversegmented clusters using $k_s = 10$, while (b) shows a single cluster as in (c) when using $k_s = 50$.

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 15 by comparing two different k_s-values in SDR using a single cluster with 10 K randomly generated samples in 20D. We see that *t*-SNE and S*t*-SNE using $k_s = 50$ show a single cluster, whereas S*t*-SNE using $k_s = 10$ shows highly oversegmented clusters. The same observation can also be made using RP, LMDS, and UMAP (see Supplemental Materials). Hence, it is important for the user to select a large enough k_s-value (i.e. $k_s \geqslant 50$) to prevent sharpening noise that may cause oversegmentation.

Parameter setting. Our parameters k_s (how localized a shift is) and α (shift speed) are interconnected, see Section Local Gradient Clustering. This also holds for other kernel density estimation (KDE) methods.[48] While both k_s and α affect the segmentation degree, if k_s is large enough, then larger k_s-values may not significantly affect segmentation without choosing an suitable α-value, see Figure 1 (second and third rows). Setting a large enough k_s-value is also crucial to avoid sharpening noise as explained in Section Noise-free data, which is why we use $k_s \geqslant 50$.

Supplemental Figure 9 completes the insights from Figures 1 and 2 by showing the results of our method for the WiFi data set for multiple values of α and k_s, using $T = 10$ iterations, as discussed in Section Local Gradient Clustering. As explained in Section Data sets and their traits and also visible in Figure 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 ($\alpha = 0.15$, $k_s \geqslant 50$). 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 1.

A too-large number of LGC iterations (T-value) can lead to over-shooting the local cluster centers during the gradient ascent and also longer computation. While Hurter et al.[48] decrease the advection speed α 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, for example, neighborhood-hit (Q_h). Exploring this (and how to do it efficiently) would be interesting for future work.

Post-processing in dimensionality reduction. Technically, our sharpening approach can also be applied to 2D instead of nD data. However, this is problematic: We know in advance that the data distances are “uniform” in nD, so sharpening with a certain distance or speed will work uniformly for all data points.[44,46] 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 4 and 9.

Other post-processing methods for 2D projections exist, for example, 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.

LGC for t-SNE and UMAP. Figures 3 and 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, for example, *t*-SNE and UMAP. UMAP yields a better VCS than other SDR results including SUMAP in most of the examples shown in this paper except for Figure 9. However, due to the limitations of UMAP mentioned in Section Related Work (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.

Selection of baseline DR method. Figures 3 and 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.[10] for the WiFi data set using $\alpha = 0.15$ (see Supplemental Materials). 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 α-values. In particular, ISO and LMVU produce some separation of clusters, while the sharpened versions of them do not. Using a smaller α for S-ISO and S-LMVU results in clearer cluster separation (see Supplemental Materials). This suggests that using different α-values can solve issues with poorer cluster separation in HD-SDR than in DR. While out of scope of this paper, exploring why certain DR methods are more suitable for HD-SDR based on a user-centric approach,[83] and which SDR is effective for different data types, are important topics to study next.

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.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: 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 programs 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.

ORCID iD

Youngjoo Kim https://orcid.org/0000-0002-9677-163X

Supplemental material

Supplemental material for this article is available online.

Notes

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 10 times fewer stars but much higher quality; see Anders et al.[2] for more details.
2. Dr. Sarah Martell, Project Scientist of the GALAH Survey and a co-author of Traven et al.[43]; and Dr. Sara Lucatello, an expert on tracing the formation of the Milky Way through the abundances of its stars.

References

1. van der Maaten L and Hinton G. Visualizing data using t-SNE. *J Mach Learn Res* 2009; 9:2579–2605.
2. Anders F, Chiappini C, Santiago BX, et al. Dissecting stellar chemical abundance space with t-SNE. *Astron Astrophys* 2018; 619:A125.
3. Dasgupta S. Experiments with random projection. In: *Proceedings of the sixteenth conference on uncertainty in artificial intelligence*, Stanford CA, June 2000, pp.143–151.
4. Xie H, Li J and Xue H. A survey of dimensionality reduction techniques based on random projection. *arXiv preprint*, arXiv:1706.04371, 2017.
5. Silva VD and Tenenbaum JB. Global versus local methods in nonlinear dimensionality reduction. *Adv Neural Inf Process Syst* 2003; 15:705–712.
6. Tenenbaum JB, de Silva V and Langford JC. A global geometric framework for nonlinear dimensionality reduction. *Science* 2000; 290:2319–2323.
7. Sammon JW. A nonlinear mapping for data structure analysis. *IEEE Trans Comput* 1969; C18:401–409.
8. McInnes L, Healy J and Melville J. UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint*, arXiv:1802.03426, 2018.
9. Lee JA and Verleysen M. *Nonlinear dimensionality reduction*. New York, NY: Springer, 2007.
10. Espadoto M, Martins RM, Kerren A, et al. Toward a quantitative survey of dimension reduction techniques. *IEEE Trans Vis Comput Graph* 2021; 27(3): 2153–2173.
11. Chen YC, Genovese CR and Wasserman L. A comprehensive approach to mode clustering. *Electron J Stat* 2016; 10(1): 210–241.
12. Gaia Collaboration. The Gaia mission. *Astron Astrophys* 2016; 595:A1.
13. Forveille T, Kotak R, Shore S, et al. Gaia data release 2. *Astron Astrophys* 2018; 616:E1.
14. Buder S, et al. The GALAH Survey: Second data release. *Mon Not R Astron Soc* 2018; 478. 4513.
15. Wang F and Zhang C. Label propagation through linear neighborhoods. *IEEE Trans Knowl Data Eng* 2008; 20(1): 55–67.
16. Cohn D, Caruana R and McCallum A. Semi-supervised clustering with user feedback. In: Basu S, Davidson I and Wagstaff K (eds) *Constrained clustering: advances in algorithms, theory, and applications*. London: Chapman and Hall/CRC, 2003, pp.17–32.
17. Benato BC, Telea AC and Falcão AX. Semi-supervised learning with interactive label propagation guided by feature space projections. In: *Proceedings of the 2018 31st SIBGRAPI conference on graphics, patterns and images (SIBGRAPI)*, Parana, Brazil, 29 October–1 November 2018, pp.392–399. New York, NY: IEEE.
18. Benato BC, Gomes JF, Telea AC, et al. Semi-automatic data annotation guided by feature space projection. *Pattern Recognit* 2021; 109:107–612.
19. Bernard J, Hutter M, Zeppelzauer M, et al. Comparing visual-interactive labeling with active learning: an experimental study. *IEEE Trans Vis Comput Graph* 2018; 24(1): 298–308.
20. Bank D, Koenigstein N and Giryes R. Autoencoders. *arXiv preprint*, arXiv:2003.05991v2 [cs.LG], 2021.
21. Bleha SA and Obaidat MS. Dimensionality reduction and feature extraction applications in identifying

computer users. *IEEE Trans Syst Man Cybern* 1991; 21(2): 452–456.

22. Nonato LG and Aupetit M. Multidimensional projection for visual analytics: linking techniques with distortions, tasks, and layout enrichment. *IEEE Trans Vis Comput Graph* 2019; 25(8): 2650–2673.
23. MacQueen J. Some methods for classification and analysis of multivariate observations. In: *Proceedings of the 5th Berkeley symposium on mathematical statistics and probability*, Oakland, CA, 1965, vol. 1, pp.281–297.
24. Johnson SC. Hierarchical clustering schemes. *Psychometrika* 1967; 32(3): 241–254.
25. Ling RF. On the theory and construction of k-clusters. *Comput J* 1972; 15(4): 326–332.
26. Berkhin P. A survey of clustering data mining techniques. In: Kogan J, Nicholas C and Teboulle M (eds) *Grouping multidimensional data*. Berlin, Heidelberg: Springer, 2006, pp.25–71.
27. Martins RM. *Explanatory visualization of multidimensional prejections*. PhD Thesis, Universidade de São Paulo, Brazil, 2016. https://www.teses.usp.br/teses/disponiveis/55/55134/tde-30092016-133421/publico/RafaelMessiasMartins_revisada.pdf (accessed 1 December 2021).
28. Rauber PE, Falcão AX and Telea AC. Projections as visual aids for classification system design. *Inf Vis* 2018; 17(4): 282–305.
29. Rauber PE, Fadel SG, Falcao AX, et al. Visualizing the hidden activity of artificial neural networks. *IEEE Trans Vis Comput Graph* 2017; 23(1): 101–110.
30. Lewis J, van der Maaten L and de Sa V. A behavioral investigation of dimensionality reduction. *Proc Ann Meet Cogn Sci Soc* 2012; 34:671–676.
31. van der Maaten L. Accelerating t-SNE using tree-based algorithms. *J Mach Learn Res* 2014; 15(1): 3221–3245.
32. Pezzotti N, Lelieveldt BPF, Maaten LVD, et al. Approximated and user steerable t-SNE for progressive visual analytics. *IEEE Trans Vis Comput Graph* 2017; 23(7): 1739–1752.
33. Pezzotti N, Höllt T, Lelieveldt B, et al. Hierarchical stochastic neighbor embedding. *Comput Graph Forum* 2016; 35(3): 21–30.
34. Pezzotti N, Thijssen J, Mordvintsev A, et al. GPGPU linear complexity t-SNE optimization. *IEEE Trans Vis Comput Graph* 2020; 26(1): 1172–1181.
35. Wattenberg M, Viégas F and Johnson I. How to use t-SNE effectively, https://distill.pub/2016/misread-tsne (2019, accessed 1 December 2021).
36. Reis I, Rotman M, Poznanski D, et al. Effectively using unsupervised machine learning in next generation astronomical surveys. *arXiv preprint*, arXiv:1911.06823, 2019.
37. Wallach I and Lilien R. The protein-small-molecule database, a non-redundant structural resource for the analysis of protein-ligand binding. *Bioinformatics* 2009; 25(5): 615–620.
38. Gashi I, Stankovic V, Leita C, et al. An experimental study of diversity with off-the-shelf antivirus engines. In: *IEEE International symposium on network computing and applications*, Cambridge, MA, 9–11 July 2009, pp.83–91. New York, NY: IEEE.
39. Murtagh F and Heck A. *Multivariate data analysis*. New York, NY: Springer-Verlag, 1987, vol. 131.
40. Deeming TJ. Stellar spectral classification: I. Application of component analysis. *Mon Not R Astron Soc* 1964; 127(6): 493–516.
41. Ting YS, Freeman KC, Kobayashi C, et al. Principal component analysis on chemical abundances spaces. *Mon Not R Astron Soc* 2012; 421(2): 1231–1255.
42. Boesso R and Rocha-Pinto HJ. Clustering in the stellar abundance space. *Mon Not R Astron Soc* 2018; 474(3): 4010–4023.
43. Traven G, Matijevič G, Zwitter T, et al. The GALAH survey: classification and diagnostics with t-SNE reduction of spectral information. *Astrophys J Suppl Ser* 2017; 228(2): 24–37.
44. Fukunaga K and Hostetler L. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Trans Inf Theory* 1975; 21(1): 32–40.
45. Cheng Y. Mean shift, mode seeking, and clustering. *IEEE Trans Pattern Anal Mach Intell* 1995; 17(8): 790–799.
46. Comaniciu D and Meer P. Mean shift: A robust approach toward feature space analysis. *IEEE Trans Pattern Anal Mach Intell* 2002; 24(5): 603–619.
47. Wu KL and Yang MS. Mean shift-based clustering. *Pattern Recognit* 2007; 40(11): 3035–3052.
48. Hurter C, Ersoy O and Telea A. Graph bundling by kernel density estimation. *Comput Graph Forum* 2012; 31:865–874.
49. Silverman BW. *Density estimation for statistics and data analysis. Monographs on statistics and applied probability.* London: Chapman and Hall, 1986. Vol. 26.
50. Epanechnikov VA. Non-parametric estimation of a multivariate probability density. *Theory Probab Appl* 1969; 14(1): 153–158.
51. Muja M and Lowe DG. Fast approximate nearest neighbors with automatic algorithm configuration. In: *Proceedings of the fourth international conference on computer vision theory and applications*, Lisboa, Portugal, 5–8 February 2009, pp.331–340.
52. Muja M and Lowe DG. Scalable nearest neighbor algorithms for high dimensional data. *IEEE Trans Pattern Anal Mach Intell* 2014; 36(11): 2227–2240.
53. Blanco JL and Rai PK. Nanoflann: A C++ header-only fork of FLANN, https://github.com/jlblancoc/nanoflann (2014, accessed 1 December 2021).
54. Kim Y, Telea AC, Trager SC, et al. High-dimensional sharpened dimensionality reduction implementation, https://youngjookim.github.io/sdr/ (2021, accessed 1 December 2021).
55. Lisitsyn S, Widmer C and Garcia FJI. Tapkee: an efficient dimension reduction library. *J Mach Learn Res* 2013; 14:2355–2359.
56. Sorzano COS, Vargas J and Pascual Montano A. A survey of dimensionality reduction techniques. *arXiv preprint*, arXiv:1403.2877[stat.ML], 2014.

57. Wold S, Esbensen K and Geladi P. Principal component analysis. *Chemometr Intell Lab Syst* 1987; 2(1–3): 37–52.
58. Torgerson WS. Multidimensional scaling: I. Theory and method. *Psychometrika* 1952; 17(4): 401–419.
59. Dua D and Graff C. UCI machine learning repository, http://archive.ics.uci.edu/ml (2017, accessed 1 December 2021).
60. OpenML. Banknote authentication classification benchmark, https://www.openml.org/d/1462 (2019).
61. Rak M, König T, Steffen J, et al. Density difference detection with application to exploratory visualization. In: *International conference on pattern recognition applications and methods*, Lisbon, Portugal, 10–12 January 2015, pp.193–207.
62. Rohra JG, Perumal B, Narayanan SJ, et al. User localization in an indoor environment using fuzzy hybrid of particle swarm optimization & gravitational search algorithm with neural networks. In: *Proceedings of sixth international conference on soft computing for problem solving*, Patiala, India, 23–24 December 2016, pp.286–295. Singapore: Springer.
63. Bhatt RB. *Fuzzy-rough approaches for pattern classification*. Independently Published, 2017.
64. Lu S. Clustering by the way of atomic fission. *arXiv preprint*, arXiv:190611416, 2019.
65. El Helou A. Sensor HAR recognition app, https://www.mathworks.com/matlabcentral/fileexchange/54138-sensor-har-recognition-app (2020, accessed 1 December 2021).
66. El Helou A. *Parameters and calibration of a low-g 3-axis accelerometer AN4508 application note*. STMicroelectronics, https://www.st.com/resource/en/application_note/an4508-parameters-and-calibration-of-a-lowg-3axis-accelerometer-stmicroelectronics.pdf (2014, accessed 1 December 2021).
67. El Helou A. *Sensor data analytics* (French Webinar Code). MATLAB Central File Exchange. https://www.mathworks.com/matlabcentral/fileexchange/54139-sensor-data-analytics-french-webinar-code (2017, accessed 01 December 2021).
68. Etemadpour R, Motta R, de Souza Paiva JG, et al. Perception-based evaluation of projection methods for multidimensional data visualization. *IEEE Trans Vis Comput Graph* 2015; 21(1): 81–94.
69. Martins RM, Minghim R and Telea AC. Explaining neighborhood preservation for multidimensional projections. In: *Proceedings of the CGVC*, 2015, pp.7–14.
70. Venna J and Kaski S. Neighborhood preservation in nonlinear projection methods: An experimental study. In: *Proceedings of the ICANN 2001*, Vienna, Austria, 21–25 August 2001, pp.485–491. Berlin, Heidelberg: Springer.
71. Levandowsky M and Winter D. Distance between sets. *Nature* 1971; 234(5323): 34–35.
72. Paulovich FV, Nonato LG, Minghim R, et al. Least square projection: A fast high-precision multidimensional projection technique and its application to document mapping. *IEEE Trans Vis Comput Graph* 2008; 14(3): 564–575.
73. Coimbra DB, Martins RM, Mota E, et al. Analyzing the quality of local and global multidimensional projections using performance evaluation planning. *Theor Comput Sci* 2021; 872(8): 41–54.
74. Aggarwal C, Hinneburg A and Keim D. On the surprising behavior of distance metrics in high dimensional space. In: *Proceedings of the ICDT 2001*, London, 4 January 2001. Heidelberg: Springer, pp.420–434.
75. Tinsley BM. Evolution of the stars and gas in galaxies. *Fundament Cosmic Phys* 1980; 5:287–388.
76. Helmi A. Streams, substructures and the early history of the Milky Way. *arXiv preprint*, arXiv:2002.04340, 2020.
77. Hyvönen V, Pitkänen T, Tasoulis S, et al. Fast nearest neighbor search through sparse random projections and voting. In: *Proceedings of the 2016 IEEE conference on big data*, Washington, DC, 5–8 December 2016, pp.881–888.
78. Jääsaari E, Hyvönen V and Roos T. Efficient autotuning of hyperparameters in approximate nearest neighbor search. In: *Pacific-Asia conference on knowledge discovery and data mining*, Macau, China, 14–17 April 2019, pp.590–602. Cham: Springer.
79. van der Zwan M, Codreanu V and Telea A. CUBu: Universal real-time bundling for large graphs. *IEEE Trans Vis Comput Graph* 2016; 22(12): 2550–2563.
80. Reyes-Ortiz JL, Oneto L, Samà A, et al. Transition-aware human activity recognition using smartphones. *Neurocomputing* 2016; 171:754–767.
81. Joia P, Paulovich FV, Coimbra D, et al. Local affine multidimensional projection. *IEEE Trans Vis Comput Graph* 2011; 17(12): 2563–2571.
82. Dressler A and Shectman SA. Evidence for substructure in rich clusters of galaxies from radial-velocity measurements. *Astron J* 1988; 95:985–995.
83. Etemadpour R, Linsen L, Paiva JG, et al. Choosing visualization techniques for multidimensional data projection tasks: a guideline with examples. In: *Proceedings of the VISIGRAPP*, Berlin, Germany, 11–14 March 2015, pp.166–186. Cham: Springer.