Implicit Multidimensional Projection of Local Subspaces

Rongzheng Bian, Yumeng Xue, Liang Zhou, Jian Zhang, Baoquan Chen, Daniel Weiskopf, and Yunhai Wang
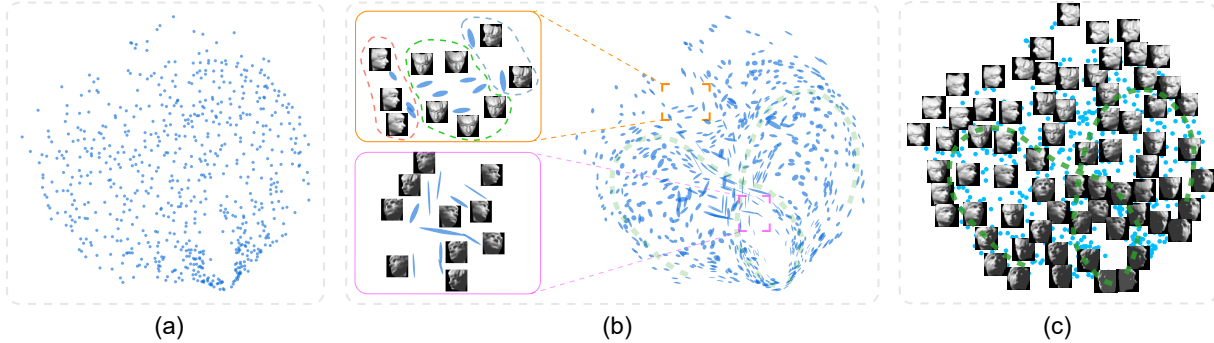
Fig. 1. The face dataset is projected to 2D using nonlinear MDS. (a) Traditional visualization just shows the projected data points; here, no obvious correlation can be seen. (b) With our new implicit local subspace projection method, global trends and local structures (i.e., local linear subspaces around data points in the original high-dimensional space) are visualized through the orientation and shape of glyphs that replace the dots in the 2D plot. Now, two groups of glyphs associated with right- and left-facing faces are clearly separated (near the bottom), and some more of the interesting local patterns are identified in the zoom-ins. (c) Two trends that cross at the center of the plot are highlighted in green in the image view (also in (b), transparent dashed lines in green); such crossing would not be visible without the orientation encoding in our new glyphs.

Abstract—We propose a visualization method to understand the effect of multidimensional projection on local subspaces, using implicit function differentiation. Here, we understand the local subspace as the multidimensional local neighborhood of data points. Existing methods focus on the projection of multidimensional data points, and the neighborhood information is ignored. Our method is able to analyze the shape and directional information of the local subspace to gain more insights into the global structure of the data through the perception of local structures. Local subspaces are fitted by multidimensional ellipses that are spanned by basis vectors. An accurate and efficient vector transformation method is proposed based on analytical differentiation of multidimensional projections formulated as implicit functions. The results are visualized as glyphs and analyzed using a full set of specifically-designed interactions supported in our efficient web-based visualization tool. The usefulness of our method is demonstrated using various multi- and high-dimensional benchmark datasets. Our implicit differentiation vector transformation is evaluated through numerical comparisons; the overall method is evaluated through exploration examples and use cases.

Index Terms—High-dimensional data visualization, dimensionality reduction, local linear subspaces, user interaction

✦

1 INTRODUCTION

Multidimensional data analysis has manifold applications in diverse domains such as finance, science, and engineering. It is often conducted by reducing data dimensionality with a dimensionality reduction (DR) technique and then visualizing the reduced data with scatterplots. By representing data points as visual marks in 2D space, 2D scatterplots are one of the most useful and common approaches [38] for visual exploration of multidimensional data. However, depicting the projected data with point-based information alone is oversimplified as only the density distribution can be perceived. In this paper, we propose a model to visually understand multidimensional local linear subspaces after dimensionality reduction—we refer to local linear subspace as the local linearized neighborhood in the original multidimensional space.

- R. Bian, Y. Xue and, Y. Wang are with Shandong University, Qingdao. Email: {bianrongzheng, xym6336, cloudseawang}@gmail.com.
- B. Chen is with Peking University. E-mail: baoquan.chen@gmail.com.
- J. Zhang is with CNIC, CAS. E-mail: zhangjian@sccas.cn.
- L. Zhou is with University of Utah. E-mail: lzhou@sci.utah.edu
- D. Weiskopf is with University of Stuttgart. E-mail: weiskopf@visus.uni-stuttgart.de
- Y. Wang and L. Zhou are joint corresponding authors.

There are many DR techniques that preserve certain structures of the data in low-dimensional space. Early techniques aim to faithfully represent the data's global structures, for example, principal component analysis (PCA) [18] and multidimensional scaling (MDS) [2, 45], however, they cannot effectively reveal the low-dimensional manifold embedded in high-dimensional data. In contrast, more recent DR methods, e.g., locally linear embedding (LLE) [36] and t-distributed stochastic neighbor embedding (t-SNE) [46], seek to map nearby points in the high-dimensional space to nearby points in the low-dimensional space. These methods can better preserve local linear structures, however, only encoding data points as positions of visual marks will not reveal such structures, except for the distance between them. In particular, the traditional point visualization misses any orientation information from the local linear structures.

Fig. 1 illustrates this issue for a typical DR example along with our strategy to resolve it. Fig. 1 (a) shows a traditional dot-based visualization in a scatterplot, where only the density distribution of data points can be perceived. With our new implicit local subspace projection method (Fig. 1 (b)), global trends and local structures are visualized through the orientation and shape of glyphs. With our approach, we can visually separate two different groups of glyphs near the bottom-right of the plot—they are generally associated with right- and left-facing faces (transparent dashed curves in green). Another example is a global trend highlighted with two green dashed curves in the image view in Fig. 1 (c); now, it is possible to identify two trends that cross at the center of the plot, each is associated with a different smoothly changing camera position. Fig. 1 (b) demonstrates that further interesting local

pattern can be identified with the glyphs, where three clusters of glyphs (bounded by dash lines in the orange zoom-in) correspond to the associated images with three distinct face orientations and the crossing pattern (see the purple zoom-in) well reflects the smooth transitions of face directions.

We propose a model to characterize multidimensional local subspaces with basis vectors anchored at each data point (all in the original space) and transform the basis vectors to a low-dimensional visualization space with an analytical vector transformation technique based on the implicit function theorem. The basis vectors are extracted using local PCA in the original multidimensional space. Our vector transformation method takes the DR technique as an implicit function and uses its analytical gradient to compute the accurate projected basis vectors. In this way, we guarantee that the transformation of the subspace is consistent with the projection of the points in the DR method. Our implicit function-based vector transformation has the advantage of being efficient and accurate at the same time.

To visualize the projected local subspace, we construct a glyph in the form of a closed B-spline curve that captures the deformations introduced by the transformation of the basis vectors. We reduce the overlap between glyphs by interactively changing their opacities and size, and order them according to glyph area. We measure the projection quality with two metrics: the loss function of the DR method and trustworthiness [13, 32], and compute the anisotropy of the local subspace in high dimensional data. Combining these measures, we design a set of linked views for interactively examining projection errors and the relationship between them and data attributes. Based on a web-based implementation, we demonstrate the effectiveness of our framework with case studies.

Our main contributions are as follows:

- A model that visualizes local subspaces for multidimensional projections such that global trends in the data can be inferred by the perception of local subspaces;

- An analytical vector transformation method based on implicit function differentiation; and

- The glyph representation that shows the geometry of transformed basis vectors of local structures in the same 2D domain as a traditional scatterplot.

Since we do not change the basic intuition associated with traditional multidimensional visualization, our approach readily fits in any existing DR visualization pipeline, regardless of whether it uses linear or nonlinear multidimensional projections. The source code is available for download on GitHub[1].

2 RELATED WORK

Dimensionality reduction is an important research topic in statistics, data science, and visualization. A comprehensive survey of DR methods can be found elsewhere [22,47]. One issue of DR techniques is their interpretation. This is a difficult problem [48] addressed in extensive previous work. A large body of research focuses on providing empirical guidance and design principles for interpreting DR results and assessing their quality. Some examples include a systematic literature review of the topic [37], metric-based quality assessments of synthetic data [21], and a study of different visual encoding schemes [38].

Another branch of research improves the interpretability of DR visualizations by including quality information [13, 32]. Aupetit [1] encodes Voronoi cells with luminance in the 2D visualization with distortion and uncertainty measurements to show the DR quality. Similarly, CheckViz [23] visualizes distortions in the local mappings with a 2D perceptually uniform color map applied to Voronoi-cell partitionings of the 2D scatterplot after dimensionality reduction. Seifert et al. [39] visualize the local stress metric [6] as a combination of a 2D heat map and a height map, simultaneously showing the stress levels of each datum and its neighboring area.

Interactive tools with brushing-and-linking and carefully designed visual encodings are also useful for understanding DR results. Embedding Projector [42] allows the user to explore DR data as a 3D scatterplot and shows DR processes by animation; however, it is a general tool for overview and more in-depth analysis is not supported. A set of interactive visual analysis methods using various visual encoding are proposed to study the quality of dimensionality reduction methods on large-scale datasets [28]. Stahnke et al. [43] probe multidimensional projections with a set of integrated interaction techniques that allow for the investigation of each data point and a neighborhood as well as additional information, e.g., classes, clusterings, and original dimensions. Comparative visualization [8] provides results of multiple DR techniques, where the user can assess different behaviors of these techniques through interactive exploration of multidimensional data with linked views.

Relating DR results with original data dimensions is yet another way of interpreting those results. Axes of original data dimensions can be drawn as 3D biplot [16] axes to understand DR results in 3D plots [7]. An interactive framework [3] allows the user to visually explore forward (high-to-low) and backward (low-to-high) projections, where original data axes are shown in 2D biplots. Alternatively, a perturbation-analysis-based method [15] aims to understand nonlinear DR methods. Here, the goal is to recognize how generalized axis lines (visualized as contours) change according to user-specified infinitesimal perturbations; small changes of the data are modeled in the original multidimensional space and their effect on the projected axes in the DR display are computed by automatic differentiation. Our work has a different goal in mind: we want to understand the shape and orientation of the local structure, and, furthermore, global correlations of data points; and we use implicit differentiation to accurately transform basis vectors in the multidimensional local neighborhood.

Features of interest in multidimensional space may live in low-dimensional subspaces [49]. Therefore, subspace analysis that models multidimensional data by the union of multiple subspaces is a powerful tool for multidimensional data analysis. For example, subspace clustering localizes the search for features in relevant dimensions [33, 49]. Recently, sparse and low-rank subspace clustering methods have been widely used in machine learning, computer vision, and pattern recognition [11, 24]. Related concepts are useful for visualization as well. For example, automatic subspace searching, grouping, and filtering, followed by interactive analysis allow users to visually explore multidimensional data [44]. Alternatively, subspace clustering and animation of dynamic projections can be used for interactive visual exploration of subspaces [25]. These methods analyze lower-dimensional subspaces and point-based projections in multidimensional datasets. In contrast, we focus on shape and direction-based analysis of local neighborhoods of the same dimensionality of the data.

Flow-based scatterplots [4] and generalized sensitivity scatterplots [5] enhance 2D scatterplots with glyphs encoding local trends. These methods are effective in identifying local relationships between two variables. More recently, clusters in DR data have been visualized on scatterplots with winglets [27] that enhance data points with arcs encoding cluster information and uncertainty. However, the local patterns visualized by both approaches are not defined by the whole original data. In contrast, improved parallel coordinates plots [31, 51] can directly visualize local multivariate correlations of the original data. For example, indexed-points parallel coordinates [51] represent multidimensional locally fitted planar structures by using p-flat indexed points, allowing for effective pattern recognition through visual clustering. However, such parallel coordinates are effective only for data with a small number of dimensions [41]. In contrast, our method is designed for multidimensional projections, can be applied to high-dimensional data, considers multiple vectors to faithfully describe local subspaces, and—most importantly—works consistently with any DR method.

Recently, user-centered dimensionality reduction methods that allow for fine tuning of the projections have attracted increasing attention. Most methods [14, 20, 34] based on neighborhood reconstruction are designed to support full interactivity and controllability but require user-provided control points. In contrast, our method focuses on the
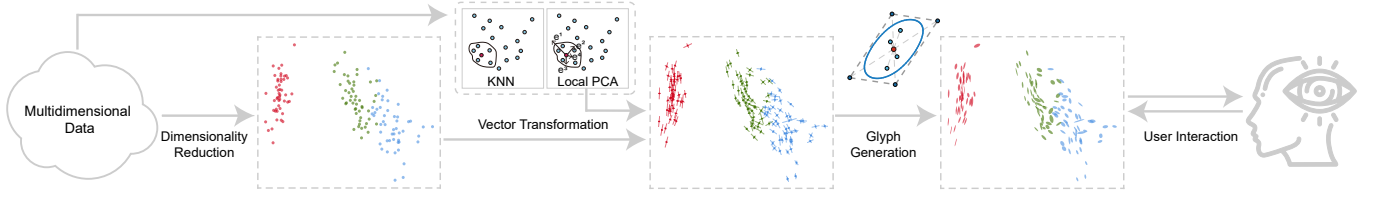
---

[1] https://github.com/VisLabWang/DRImplicitVecXform

Fig. 2. Workflow of our method: Dimensionalty reduction (left) projects points from the original space to the low-dimensional visualization space. We extend this conventional approach by first computing a linear subspace around each data point using local PCA and then projecting the basis vectors of the subspace to the low-dimensional visualization space. These transformed vectors are used to encode the deformation of the projected subspace in glyphs.

understanding of projection results with glyph perception and user interaction, and reveals the structure of local subspace around each projection point.

3 METHOD OVERVIEW

Our method augments point-based multidimensional projection with the shape and orientation information depicting the multidimensional local subspace around data points. The global structure of multidimensional datasets can be better understood by including local subspaces. Fig. 3 motivates our approach for a simple example of a synthetic data with two perpendicular planes in 3D space.

The projection is chosen in a way so that the projection direction is not perpendicular to the planes. Therefore, it is not possible to separate the two planes in the traditional point-based visualization (Fig. 3 (c)) unless we use color to distinguish them (Fig. 3 (f)). With our method, the two planes can be easily identified—even without color coding—by the shapes and directions of glyphs: thin, elongated, vertical-going glyphs associated with the plane that covers a smaller area after projection; and round glyphs that represent the plane that is better preserved in the projection (Fig. 3 (b)).

Our approach extends traditional multidimensional projection in three ways: (1) extract the local linear subspace in the original space, (2) project the subspace in a way that is consistent with the DR technique applied to the data points, and (3) visualize the information from the projection of the subspace. The respective workflow of our method is illustrated in Fig. 2.

For (1), we identify the local linear structure around each data point in the multidimensional space using the k-nearest neighbors (kNN) method. The local neighborhood of a point is fitted by a multidimensional ellipsoid as we perform PCA of the neighborhood and obtain its eigenvectors and the corresponding eigenvalues. In step (2), the extracted subspaces are transformed to the 2D visualization space. To this end, we transform the eigenvectors spanning local subspaces using
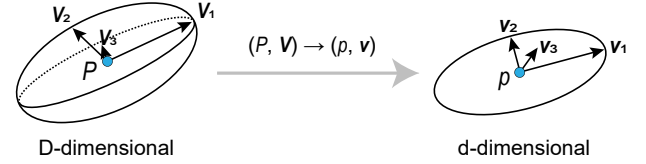
Fig. 4. The local subspace of a point P in the original space is spanned by basis vectors V_i, and these are transformed to the d-dimensional visualization space, resulting in p and v_i, respectively.

implicit differentiation as explained in Section 4. In practice, not all eigenvectors are needed as many of them are associated with small eigenvalues and contribute little to the subsequent computations, i.e., we use local linear subspaces that do not necessarily have the full dimensionality of the original dataset. Finally (3), data with local subspace information is visualized using 2D ellipse-like glyphs generated from the data points and the transformed vectors as discussed in Section 5.

4 MATHEMATICAL MODEL

This section first clarifies the terminology and mathematical assumptions that we make. Then, we discuss our way of representing local linear subspaces and the corresponding transformation of vectors. This leads to our approach to using implicit functions to compute the transformation for general nonlinear DR methods.

4.1 Multidimensional Projection of Points

We present a general definition of multidimensional projections that will be used as a basis for our extended projection approach. In the following, we use uppercase letters for variables that are related to the original multi-dimensional space, and lowercase letters for quantities in the projection space (i.e., typically the 2D visualization space).

Given a multi-dimensional dataset with the dimensionality D, a point in this space is denoted $P \in \mathbb{R}^D$. Multidimensional projection takes points at P to corresponding locations $p \in \mathbb{R}^d$ ($d \leq D$) in the lower-dimensional visualization space:

$$\pi : \mathbb{R}^D \to \mathbb{R}^d, \quad P \mapsto p = \pi(P) . \tag{1}$$

Our discussion and derivation is independent of the choice of π, i.e., we formulate our approach to work with any linear or nonlinear multi-dimensional projection.

Most nonlinear multidimensional projection techniques employ some complex optimization approaches to arrive at the projection π. Often, this map is not defined for all $\mathbb{R}^D$, but only the projection of the input points is computed, i.e., the target locations $p_i = \pi(P_i)$. In contrast, the linear projection π is directly accessible. A prominent case is the linear projection by PCA [18]:

$$\pi_{\mathrm{PCA}}(P) = \mathcal{M}P = p , \tag{2}$$

where $\mathcal{M}$ is a $d \times D$ matrix consisting of the first d eigenvectors (used as row vectors) that result from the diagonalization in PCA, i.e., in our typical use case of 2D visualization, these are the two eigenvectors corresponding to the two largest eigenvalues.
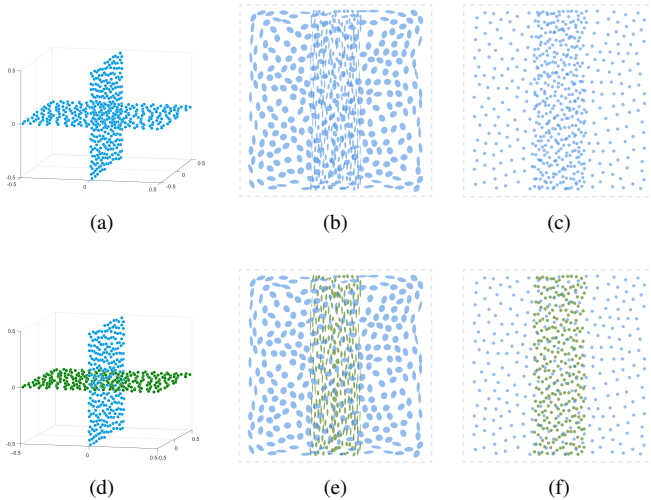

(a) (b) (c)

(d) (e) (f)

Fig. 3. Two perpendicular planes in 3D (a) and rendered with color-coding (d). The multidimensional projection is visualized using our method (b) and (e), and traditional point-based visualization (c) and (f).

4.2 Local Subspaces and Transformation of Vectors

Let a local subspace S within the high-dimensional space be anchored around a point at position $P \in \mathbb{R}^D$ and have dimensionality $L \leq D$. As shown in Fig. 4, the subspace is spanned by the set of basis vectors $\mathscr{V} = \{V_i\}_{i=1,\dots L}$:

$$S = \left\{ Q \in \mathbb{R}^D \;\middle|\; Q = \sum_{i=1}^{L} \alpha_i V_i + P \text{ with } \alpha_i \in \mathbb{R} \right\}, \qquad (3)$$

where L is the intrinsic dimensionality and the set of basis vectors $\mathscr{V}$ is obtained by computing local PCA [50]. Specifically, we first find the k nearest neighbors of P, then perform PCA with the point P and its k nearest neighbors, and finally construct $\mathscr{V}$ by picking the first L eigenvectors that can explain most of total variance.

To analyze the effect of π on the local subspace, we reduce this problem to understanding how vectors are mapped from the high-dimensional space to the low-dimensional visualization space. Let us consider a vector $V \in \mathbb{R}^D$ anchored at point P. This vector is transformed [17, page 290] according to

$$v = \frac{\partial \pi(P)}{\partial P} V, \qquad (4)$$

with the Jacobian matrix of size $d \times D$,

$$\frac{\partial \pi(P)}{\partial P} = \begin{pmatrix} \frac{\partial \pi(P)_1}{\partial P_1} & \cdots & \frac{\partial \pi(P)_1}{\partial P_D} \\ \vdots & \ddots & \vdots \\ \frac{\partial \pi(P)_d}{\partial P_1} & \cdots & \frac{\partial \pi(P)_d}{\partial P_D} \end{pmatrix}. \qquad (5)$$

The subscripts refer to the respective component of the Cartesian point.

It is important to point out that a vector V only "lives" together with an anchor point P, i.e., the two together form a fiber bundle [19]—here, a vector bundle or vector field. In summary, the combined multidimensional projection of points and attached vectors can be written as:

$$\boxed{(P, V) \mapsto (p, v) = \left(\pi(P), \frac{\partial \pi(P)}{\partial P} V \right).} \qquad (6)$$

Fig. 4 illustrates the geometry of points and vectors in both the original D-dimensional space and the d-dimensional visualization space.

Equation 6 can be applied to all basis vectors $V_i \in \mathscr{V}$ of the local subspace. However, it should be noted that the transformed basis vector do not necessarily form a basis in the target space $\mathbb{R}^d$. Often, the original subspace has dimensionality L larger than the available dimensionality d in the target space. Nevertheless, we can use the set of transformed vectors to learn about the characteristics of the multidimensional projection; see later in Sect. 5 and Sect. 7. This problem does not arise for $L \leq d$; here, the transformed basis vector(s) may indeed form a basis in the target space. A simple example is $L = 1$, i.e., when just a single vector is transformed to show the effect on local linear correlation.

4.3 Direct Computation of Vector Transformation

An open question is the actual computation of the vector transformation from Equation 6. For a few cases, this computation is straightforward. For the example of PCA (Equation 2), we have a simple linear map π and, therefore, the Jacobian matrix is identical to the matrix from the linear PCA map:

$$(P, V) \mapsto (p, v) = (\mathscr{M}P, \mathscr{M}V). \qquad (7)$$

The above computation is valid for any linear projection.

However, multidimensional projections, in general, are nonlinear and more complex. And they may not lend themselves to an analytic derivations of the Jacobian matrix. Here, we could resort to using a numerical computation of the partial derivatives for the Jacobian matrix.

The standard approach employs finite differences [35], for example, in the form of forward differences applied to each element of the Jacobian matrix:

$$\frac{\partial \pi(P)_i}{\partial P_j} \approx \frac{\pi(P + h e_j)_i - \pi(P)_i}{h}, \qquad (8)$$

with the unit vector e_j in direction of the j-th dimension and a small distance measure $h \in \mathbb{R}^+$. Forward differences provide a first-order approximation; second-order approximation is achieved by analogous central differences.

While finite differences are a valid means of approximating the vector transformation, they came with a number of shortcomings: They are just a numerical approximation and they need to compute the projection at a number of additional points $P + h e_j$, which are not in the original set of data points. The latter issue has two negative implications. First, we need more computations of projections, which incurs higher computational costs. Second, and more importantly, DR techniques are data-dependent and often work only on the given input data; therefore, it can be hard to feed in data points that are different from the original point set or these additional points will modify the projection itself, which would lead to systematic errors.

To resolve these issues we now introduce a new vector transformation approach based on the implicit function theorem.

4.4 Implicit Vector Transformation

Nonlinear projection methods typically find the relationship between P and $\pi(P)$ by optimizing an objective function $f(P, \pi(P))$. Here, the mapping between P and $\pi(P)$ is implicitly described by f. However, even the simpler linear projections can be formulated in this way. Therefore, we assume that any reasonable projection finds an optimal target location for each data point P by solving the following optimization problem to arrive at the projected data points $\pi(P)$:

$$\min_{\pi(P)} f(P, \pi(P)). \qquad (9)$$

In other words, any resulting point $\pi(P)$ "sits" in a local minimum with respect to the cost function. If this was not the case, the projection method could and should move the projected point $\pi(P)$ to reduce the cost.

Equation 9 leads to the necessary condition for the minimum if f is a smooth function, and the minimum is inside (not on boundaries of the domain):

$$\frac{\partial}{\partial \pi(P)} f(P, \pi(P)) = 0, \quad \text{where}$$

$$\frac{\partial}{\partial \pi(P)} = \left[\frac{\partial}{\partial \pi(P)_1}, \frac{\partial}{\partial \pi(P)_2}, \cdots, \frac{\partial}{\partial \pi(P)_d} \right]^T. \qquad (10)$$

Using the implicit-function differentiation theorem [26, Chapter 11], we take the partial derivative with respect to P on both sides of Equation 10, and apply the chain rule:

$$\frac{\partial^2 f(P, \pi(P))}{\partial P \partial \pi(P)} + \frac{\partial^2 f(P, \pi(P))}{\partial \pi(P)^2} \frac{\partial \pi(P)}{\partial P} = 0 \qquad (11)$$

$$\Rightarrow \boxed{\frac{\partial \pi(P)}{\partial P} = -\left(\frac{\partial^2 f(P, \pi(P))}{\partial \pi(P)^2} \right)^{-1} \left[\frac{\partial^2 f(P, \pi(P))}{\partial P \partial \pi(P)} \right].} \qquad (12)$$

Equation 12 is the key mathematical result of our paper. It describes the implicit vector transformation that we can now use to transform subspaces. The transformation is generally applicable and completely accurate as long as the reasonable assumption from Equation 9 holds for the underlying smooth multidimensional projection. Furthermore, the transformation has to be computed only for the original points P_i in the dataset, and not at any other locations or for any other points.

For a specific choice of nonlinear dimensionality projection method, we have to evaluate the two derivatives $\left(\frac{\partial^2 f(P, \pi(P))}{\partial \pi(P)^2} \right)^{-1}$ and $\frac{\partial^2 f(P, \pi(P))}{\partial P \partial \pi(P)}$ with the specific $f(P, \pi(P))$ used in the method.

4.5 Application to MDS

Here, we briefly explain the objective functions and the strategy to compute Equation 12 for the representative nonlinear method MDS. We have also derived solutions for t-SNE that can be found in the supplemental material.

The MDS method has both linear and nonlinear versions. Here, we consider the SMACOF version [2], which is widely used as the default method in data analysis packages, for example, scikit-learn, due to its good performance. SMACOF minimizes the objective function:

$$F = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} (||x_i - x_j|| - ||y_i - y_j||)^2 , \quad (13)$$

where x_i, x_j are original high-dimensional data points, and y_i and y_j are projected low-dimensional data points. We can map this equation to our original formulation in Equation 12 by associating F with f, x_i, x_j with P, and y_i, y_j with $\pi(P)$. We first compute the partial derivative $\frac{\partial F}{\partial y_i}$ of Equation 13:

$$\frac{\partial F}{\partial y_i} = 2 \sum_{j \neq i} \left(1 - \frac{||x_i - x_j||}{||y_i - y_j||}\right) (y_i - y_j), \quad (14)$$

where the denominator requires all duplicate points to be removed.

From $\frac{\partial F}{\partial y_i}$, we then derive the second derivatives $\frac{\partial^2 f(P, \pi(P))}{\partial \pi(P)^2} = \frac{\partial^2 F}{\partial y_i \partial y_j}$ and $\frac{\partial^2 f(P, \pi(P))}{\partial P \partial \pi(P)} = \frac{\partial^2 F}{\partial y_i \partial x_j}$ for cases $i = j$ and $i \neq j$, respectively:

$$\frac{\partial^2 F}{\partial y_i \partial y_j} = \begin{cases} 2 \sum_{k \neq i} \left(\left(1 - \frac{||x_i - x_k||}{||y_i - y_k||}\right) I + \frac{||x_i - x_k||}{||y_i - y_k||^3} (y_i - y_k)(y_i - y_k)^T \right), & \text{if } i = j \\ 2 \left(\frac{||x_i - x_j||}{||y_i - y_j||} - 1 \right) I - \frac{2||x_i - x_j||}{||y_i - y_j||^3} (y_i - y_j)(y_i - y_j)^T, & \text{otherwise ,} \end{cases} \quad (15)$$

where I is the identity matrix, and

$$\frac{\partial^2 F}{\partial y_i \partial x_j} = \begin{cases} \sum_{k \neq i} -2 \frac{(y_i - y_k)}{||y_i - y_k||} \frac{(x_i - x_k)^T}{||x_i - x_k||}, & \text{if } i = j \\ 2 \frac{(y_i - y_j)}{||y_i - y_j||} \frac{(x_i - x_j)^T}{||x_i - x_j||}, & \text{otherwise .} \end{cases} \quad (16)$$

5 VISUALIZATION AND INTERACTION

In this section, we elaborate on how to visualize projected local subspaces with glyphs, explain specialized user interactions for visual exploration, and briefly report on the implementation of our method. Fig. 7 shows a screenshot of our interactive tool.

5.1 Glyph Generation

We use a glyph to visualize the transformation from the L-dimensional local linear subspace embedded in the original D-dimensional space to d-dimensional visualization space. The original basis vectors V_i are the eigenvectors that come from local PCA in high-dimensional space, and are assumed to be normalized to unit length. We incorporate a weight α_i that measures the importance of the eigenvalue λ_i associated with V_i, according to $\alpha_i = \lambda_i / \sum_{j=1}^{L} \lambda_j$. Therefore, our original basis is $\{\alpha_i V_i | i = 1, \ldots, L\}$, anchored at point P. According to Equation 6, it is transformed to $(\pi(P), \{\alpha_i v_i | i = 1, \ldots, L\})$. The latter one is what the glyph should visualize.

However, naive visualization is not possible because the $\alpha_i v_i$ usually do not form a basis in the d-dimensional visualization space. Typically $L > d$ and, thus direct visualization of all L vectors as lines (Fig. 5 (a)) might cause visual clutter that impairs perception. Ellipse based encoding can result in smooth visualizations (Fig. 5 (c)), but it supports only two transformed basis vectors. Although the convex hull (Fig. 5 (b)) of the transformed basis vectors can show all L basis vectors, it is not smooth and overlapping areas of neighboring glyphs are large and potentially misleading. Given the disadvantages of these options, we design a closed B-Spline convex hull glyph, as shown in Fig. 6 (d), which can represent the major transformed basis vectors and it is smooth and easy to distinguish.

Our solution to this problem is illustrated in Fig. 6: a glyph is generated by first computing the convex hull (Fig. 6 (b)) of the transformed vectors centering at the projected point $\pi(P)$ (Fig. 6 (a)); then, a smooth
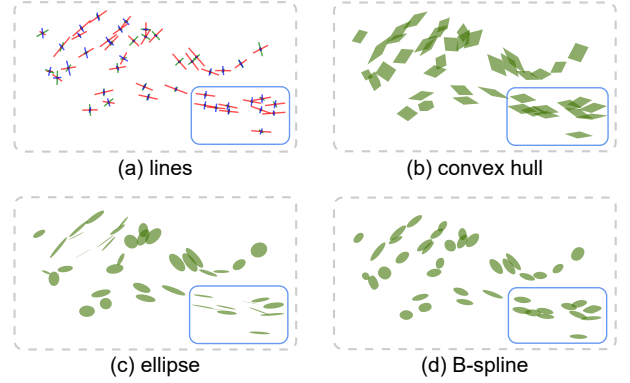

(a) lines   (b) convex hull

(c) ellipse   (d) B-spline

Fig. 5. Different options of glyph visualizations encoding transformed basis vectors: (a) lines, (b) convex hull, (c) ellipse, and (d) B-spline.

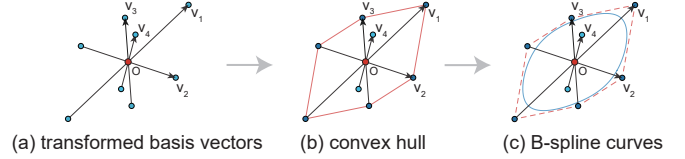(a) transformed basis vectors   (b) convex hull   (c) B-spline curves

Fig. 6. A glyph is generated for (a) the set of transformed basis vectors: the glyph (c) is computed using B-spline of (b) the convex hull of the transformed basis vectors.

shape is computed using a B-spline [9] within the convex hull whose vertices are used as control points (Fig. 6 (c)).

The B-spline is used because of its desired properties: it stays within the convex hull and follows a local control of its shape, which ensures that glyphs are comparable. One principal issue in glyph-based visualization is occlusion caused by the larger coverage on screen than by small dots. We alleviate the issue by using transparency and alpha blending: glyphs are assigned user-specified opacities, and they are drawn in layers such that larger glyphs are drawn below smaller ones.

Furthermore, scaling is supported in our visualization tool so that the user can change the size of glyphs globally. The visualization is generated by alpha-blending glyphs assigned with desired visual properties—color, opacity, and scale. These visual properties can be interactively modified by the user with interactions as shown on the left-hand side of Fig. 7.

5.2 Interaction

Our interactive exploration system is shown in Fig. 7. It comprises three modules: the control panel (Fig. 7 (a)), the local subspace projection viewer (Fig. 7 (b)) with glyph zoom-in (Fig. 7 (c)), and a linked point-based projection viewer with option to show image references (Fig. 7 (d)) for comparison. The local subspace projection viewer is synchronized with the point-based viewer for zooming-and-panning—it is particularly useful for comparing local trends of glyphs and changes in image data (projection viewer) in high-dimensional datasets in computer vision as shown in Section 7. Brushing-and-linking is supported for closer examinations: the user can select glyphs (in Fig. 7 (b)) or points (in Fig. 7 (d)) of interest with a lasso and corresponding points and glyphs are highlighted.

Furthermore, a set of specialized user interactions—including projection-quality filtering, flexible color mapping, and glyph structure zoom-in—are adopted in our method. The user changes opacity and size of glyphs with two sliders to reduce the occlusion for further explorations of local subspaces. Color mapping is useful when visualizing various metrics of data points and their local subspaces—for example, class, and the projection quality (see Section 5.2.1). The user may be also interested in the actual projected vectors from the local subspace. This information is especially useful for identifying anomalies. A zoom-in visualization of the glyph (Fig. 7 (c)) shows transformed basis vectors (red, green, blue, and cyan are for the first through fourth basis vector, respectively) inside the outline of the B-
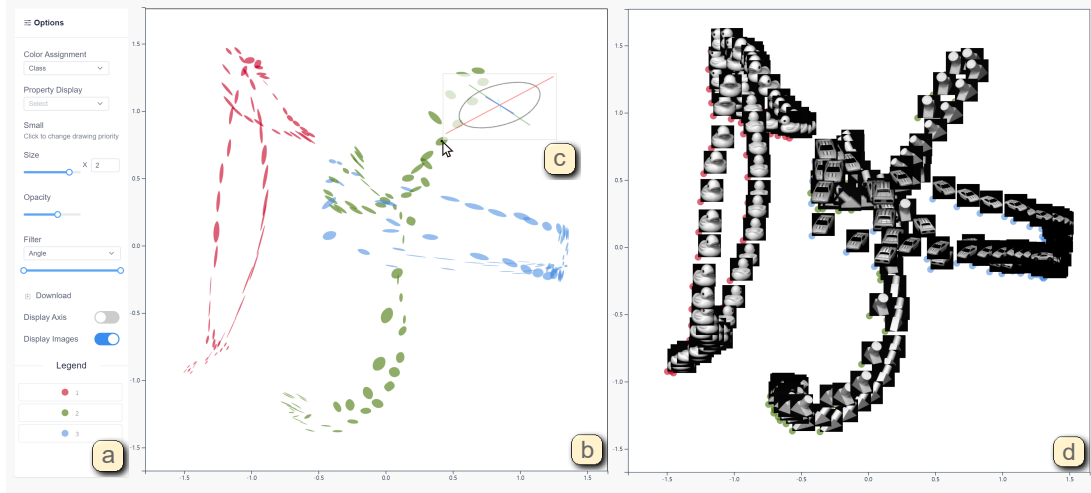
Fig. 7. Main modules in our web-based interactive visualization system: (a) control panel, (b) local subspace projection view with (c) zoomable glyphs of interest, and (d) point-based dimensionality reduction view with option to show image data for reference. This example shows the COIL dataset with PCA.

spline shape. The zoom-in is activated whenever the mouse pointer hovers over a glyph. Glyph filtering allows the user to focus on glyphs based on a user-selected metric among desired class, anisotropy, or projection quality.

These interactions allow us to adopt the "overview first, zoom and filter, then details-on-demand" [40] process for visual analysis. Specifically, we first observe the visualization of the whole dataset to find interesting general trends and local patterns: e.g., anomaly and crossings. Then, we focus on these patterns and examine them with interactive filtering and brushing-and-linking, and verify our findings with the original data if possible (for example, images); when necessary, we zoom in to further explore the details.

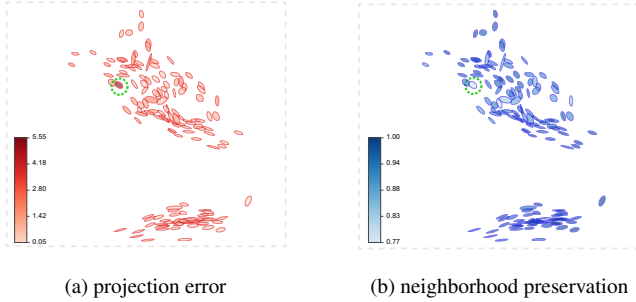(a) projection error   (b) neighborhood preservation

Fig. 8. Color mapping of two projection quality metrics for MDS-projected Iris dataset: (a) stress error where a lower score indicates better projection quality, and (b) trustworthiness where a higher score indicates better projection quality. The point highlighted by the green circle has the largest stress error and least trustworthiness.

5.2.1 Projection Quality

The quality of the DR method used for data points is also important for visual analysis. We introduce three metrics to describe the projection quality: projection error, neighbor preservation, and linearity.

The projection error metric is the loss function of the particular DR method (Fig. 8 (a)). For MDS, the metric is the stress error. The neighborhood preservation degree is independent of the DR method, measured by the trustworthiness [13, 32] (Fig. 8 (b)), where k is set to be the same as the one used for computing local PCA.

The linearity metric describes the anisotropy of the local subspace in the original dimensionality—the ratio of the magnitude between largest eigenvalue and second-largest eigenvalue. Unlike the above two metrics, coloring the linearity with each glyph only indicates how similar the glyph-encoded local subspaces in the original high-dimensional space are, but the change-of-linearity can be revealed by looking at the aspect ratio of the glpyh itself. For example, the glyphs selected by the blue

box in Fig. 10 has linearity close to one, but its result shape is elongated in Fig. 10 (b).

The conjunction of these metrics help us better understand behaviors of the DR method. For example, it seems that the projection error Fig. 8 (c)) is negatively correlated with the neighborhood correlation (Fig. 8 (d)), especially for the highlights with orange circles, in the MDS projection.

5.3 Implementation

The computational steps in our method—vector transformation by implicit differentiation and multidimensional projections—were implemented using Python and NumPy. These steps are calculated only once and the results are used as input to our web-based user interface. The user interface was implemented using JavaScript and Vue.js. The local subspace projection viewer is based on WebGL so that interactive visualization is achieved even for a large number of glyphs; the point-based projection viewer was realized using D3. Thanks to our efficient visualization tool, full interactivity is achieved for all datasets used in our paper.

6 NUMERICAL EVALUATION

Table 1. Statistics of transformed basis vectors of non-border data points of the planar data.

	mean len1	mean len2	mean angle	std of angles
Ours	0.10068096	0.10068095	89.92020374	0.00502165
Rand [15]	0.10005878	0.09957406	89.03798816	0.50298048

We numerically compare our accurate implicit vector transformation method to the approximated random approach used in DimReader [15]. In contrast to our analytical technique, DimReader randomly chooses half the points and calculates one projected vector of each data point using auto-differentiation, and the process is repeated for the number of L basis vectors. We use the aforementioned implementation for our method and our own implementation of the random approach with publicly-available code snippets of DimReader on the Internet (both in Python without acceleration) for all evaluations.

Note that our implicit function method is the analytical, i.e., accurate, transformation of basis vectors, whereas the random approach is an approximation. To verify the correctness of our method, a synthetic planar data sampled on a 3D regular grid is generated and projected to 2D using MDS. With a neighborhood of $k = 8$, the transformed basis vectors of all non-border data points should be, theoretically, exactly orthogonal, i.e., 90 degrees, and having the same magnitudes. It can be seen in Fig. 9 that glyphs of transformed basis vectors with our method appear identical (Fig. 9 (a)), whereas different shapes of glyph are

visible with the random approach (Fig. 9 (b)). Quantitatively, Table 1 summarizes basic statistics of transformed basis vectors of non-border data points of both methods: our method has a smaller difference of mean lengths of basis vectors (mean length 1 vs 2), a mean angle closer to 90 degrees (mean angle) with lower standard deviation (std of angles) than the random method. Furthermore, distributions of these measures are shown as histograms in the supplemental material. This verifies that our method is more accurate than the random method.

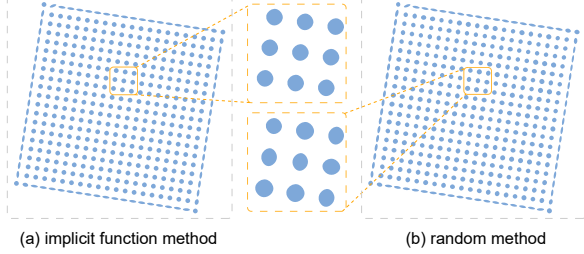(a) implicit function method   (b) random method

Fig. 9. Transformed basis vectors visualized as glyph by (a) our implicit function method and (b) the random method [15]. The data points are on a plane in 3D sampled using a regular grid, and projected to 2D with MDS.

Next, we perform evaluations on real-world multidimensional datasets using nonlinear projections via MDS and t-SNE. We employ typical benchmark datasets: Iris (147 4D points—three duplicated points are removed), Wine (178 13D points), and Digits40 (606 40D points). Fig. 10 compares the glyph visualizations of transformed basis vectors generated by our method and the random method on the Iris dataset color-coded with the linearity measure. It can be seen that the random method generates more abnormal glyphs—with excessively long and thin shapes or with distinct directions inside a neighborhood of uniform directions. We also measure computation times for the two methods for transforming the top five (four for the Iris data) basis vectors: the results show that our method is faster than the random method for all test datasets with both MDS and t-SNE projections (Table 2 of the supplemental material). All figures of glyph visualizations generated during the evaluation can be found in the supplemental material. This is evidence that our implicit differentiation method is a fast and accurate vector transformation method from high-dimensional to low-dimensional space.

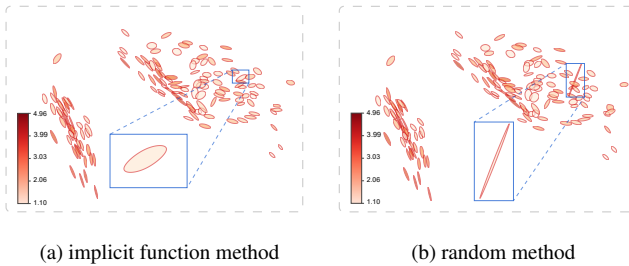(a) implicit function method   (b) random method

Fig. 10. Glyphs with basis vectors transformed by our implicit function method (a) and the random method [15] (b) for the MDS-projected Iris dataset. The color indicates the linearity of each glyph, while the shape indicates the change of the linearity in the 2D projection. We highlight the long thin glyph (selected by the box) that resulted from the random method.

7   CASE STUDIES

In this section, we show examples of visual analysis of various representative multidimensional datasets—ranging from benchmark datasets in machine learning to downsampled image datasets in computer vision—to demonstrate the usefulness of our method. We experiment with typical DR methods—PCA, MDS [2], and t-SNE [46]—on these datasets, and summarize and discuss interesting results. Further examples can be found in the supplemental material.

Examples of multidimensional datasets from the UCI machine learning repository [10] are shown in Fig. 11. The Wine dataset is a 13D data of 178 instances with 3 classes. The PCA and MDS results as shown in Fig. 11 (Wine), rows one and two. With our method (left boxes), we can see global trends from the variations of local subspace glyphs. In the PCA result, sizes of most glyphs are rather uniform, and the orientations of glyphs vary smoothly, forming a global arc structure; glyphs in the MDS projection suggest a swirling global structure. Local subspace visualization of the t-SNE projection (Fig. 11, Wine, row three, left box) shows connections between the three clusters, and the glyphs on the exterior seem to form the boundary of the clusters—this may indicate that the clusters sit on the same manifold. Besides the global trends, we can easily spot glyphs of distinct shapes (e.g., the elongated green glyphs in MDS and t-SNE) that can be further investigated. With traditional point-based visualizations (right boxes), none of the global trends can be observed nor local points-of-interest can be found as indicated by abnormal glyphs.

Visualizations of the Seeds dataset (210 data points of 7D) do not exhibit clear global trends compared to the Wine example. However, it can be observed that in the PCA (Fig. 11, Seeds, row one) and the MDS (Fig. 11, Seeds, row two) projections the glyphs form a twisting structure with crossings next to the central class. The t-SNE result (Fig. 11, Seeds, row three) provides a similar look as of the Wine example; glyphs around the boundary of clusters seem to comprise the shape of a bow tie.

The COIL dataset [30] is a collection of 20 objects each of them photographed 72 times at different rotation angles. We pick three objects: a duck, a car, and a block. Scatterplots color-coded by object class visualizing projections of PCA, MDS, and t-SNE are shown in Fig. 12 (a), (c), and (e), respectively, where no correlation can be seen between points. With our implicit local subspace projection method (Fig. 12 (b)), global trends and local structures can be visualized with the orientation and shape of glyphs. In the orange zoom-in on the red class, we see that the transition of orientations of glyphs matches that of the "duck" images. Four larger anisotropic glyphs are displayed in the purple zoom-in on the blue class—they are associated with backward facing "cars." Furthermore, the anisotropy of glyphs is correlated with the rotation angle (with respect to front-facing images) of images. In the MDS case (Fig. 12 (row 2)), glyphs generated by our method (Fig. 12 (d)) form twisting structures that are similar to the PCA case. For example, by examining the images of the red glyphs in the orange zoom-in, it is confirmed that the crossings are actually associated to the change of orientation of the "duck"; similarly, the green glyphs in the zoom-in on the right show two distinct directions of the "block". Our visualization of the t-SNE result (Fig. 12 (f)) shows that the orientation of glyphs changes smoothly within each cluster. The orange zoom-in on the green cluster along with the actual images confirms that our method successfully captures the smoothly changing image orientations in the local subspaces. The purple zoom-in confirms that there exists a crossing of orientations of the car.

The Face data contains 698 face images of different camera positions and lighting conditions. Local subspace visualizations of the Face data with MDS is already shown in Fig. 1, and PCA and t-SNE results are shown in Fig. 13, respectively. With PCA projection colorized by trustworthiness (Fig. 13 (a)), dense glyphs are seen on the left, and two trends of glyphs—one going upward and another going downward—separate around the center of the visualization. The intersection region between these two trends has lower trustworthiness and contains three distinct types of glyphs (see the zoom-in), corresponding to the face images with three distinct orientations. Similarly, two distinct types of glyphs selected by the magenta box are associated with distinct face directions.

The clusters can be clearly seen in our t-SNE result (Fig. 13 (b)), but no prominent global trends can be recognized. However, we can identify local patterns that exhibit trends; for example, in the orange zoom-in, the trends of glyphs are associated with the smooth change of face orientation. By coloring each glpyh with the t-SNE loss function (KL divergence), we can see that the glyphs around the projection center has large size and loss values, indicating that the derivatives of
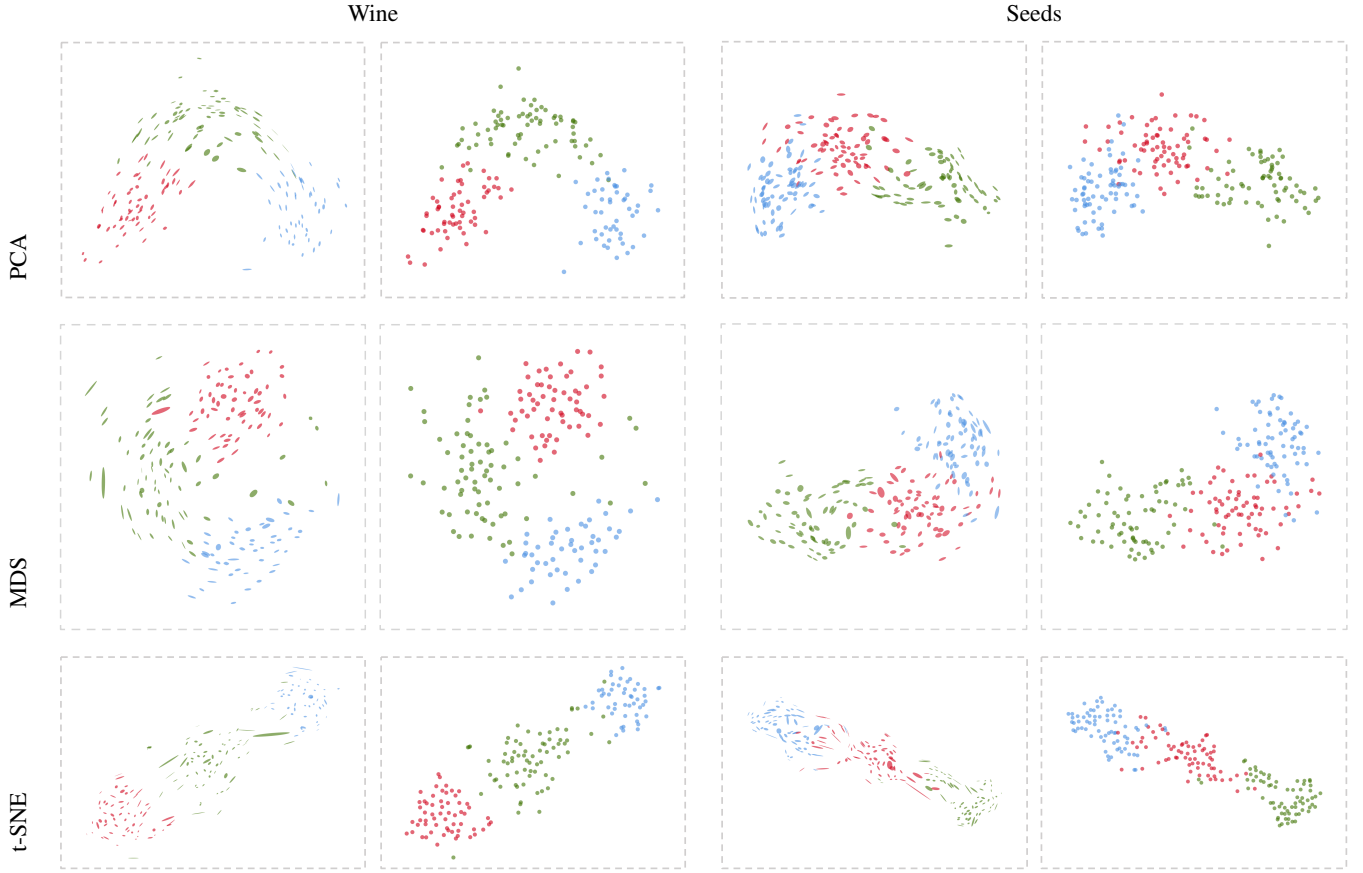
Fig. 11. Visualizations of multidimensional datasets of Wine (first column) and Seeds (second column). Each projection method is visualized using our method (left boxes) and the point-based visualizations (right boxes). We can see global trends and twisting structures from the variations of local subspace glyphs.

the corresponding data points have large magnitudes.

These experiments demonstrate the usefulness of our method in enhancing the anomaly identification ability and revealing global and local trends in data projections.

8 DISCUSSION, CONCLUSION, AND FUTURE WORK

We have introduced a method for visualizing local linear structures in multidimensional projections using implicit differentiation. For each data sample in the original space, a local neighborhood is extracted and its structure is fitted using PCA. Basis vectors of the multidimensional local neighborhood anchored at the respective data point are transformed to the low-dimensional visualization space by calculating differentiation of implicit functions. We have derived analytical solutions for typical multidimensional projections. Next, for each data point, a glyph centered at the projected data point is generated by constructing a B-spline shape within the convex hull spanned by the transformed basis vectors. To aid the analysis of our glyph representation, a linearity metric and a projection quality metric are used to depict the faithfulness of the transformation. We have built an interactive, web-based visual analysis system that supports full-fledged, comparative analysis of our results and the point-based scatterplots.

We would like to discuss a few aspects that could affect the results of our method. The number of neighbors k of the local subspace is a user-set parameter—a small k extracts local information of the data, whereas a large k reveals more global information. We have observed that datasets with rather high dimensionality (e.g., Face) are less sensitive to k than datasets with lower dimensionality (e.g., Iris). Our implicit differentiation transformation requires good convergence of the DR method—otherwise, numerical errors occur during the computation of the Jacobian matrix. Also, duplicate data points need to be removed as they would yield dividing by zero error when computing the Jacobian.

In terms of scalability, we argue that the visual scalability of our method is comparable to traditional point-based visualizations, be-

cause the glyphs' appearance can be interactively adjusted, and most information can be perceived with point-sized (as in the point-based visualizations) glyphs as shown in examples throughout the paper. For larger datasets, our current method also suffers from the overdraw problem, as point-based visualizations in general, and we could adopt clutter reduction techniques [12], e.g., sampling, for better visual scalability. For computational scalability, the memory size is the limiting factor of computing the Jacobian matrix. The largest dataset we have experimented on our machine contains 7494 data points of 16-D. In the future, we would like to optimize the calculation of the Jacobian matrix on GPUs.

In conclusion, our method is a promising tool for understanding and analyzing multidimensional data and DR methods. A benefit of our method is that main trends and anomalies in the projected data can be quickly identified with glyphs—large differences of orientation, shape, and size can be easily perceived. Once the glyphs of interest are identified, the user can further analyze the data with interactive tools to understand the cause of trends or anomalies. We have shown the usefulness of our method through a number of multidimensional datasets with popular DR methods, including PCA, MDS, and t-SNE. In this process, we have gained new insights into familiar datasets and widely used DR methods. Another benefit is that our implicit vector transformation is accurate and fast. Our vector transformation method has been assessed through numerical comparisons, and our overall method has been evaluated through case studies on benchmark datasets.

In the future, we would like to integrate our approach with more DR methods together. One prominent example is UMAP [29], which keeps a continuous transform function internally and thus can directly transform any point without our partial derivative based transformation. Second, more intelligent analysis approaches could be used to assist analyzing our results, for example, pattern recognition methods that automatically detect trends and local patterns of interest. Finally, we would like to study the effectiveness of our approach in real applications
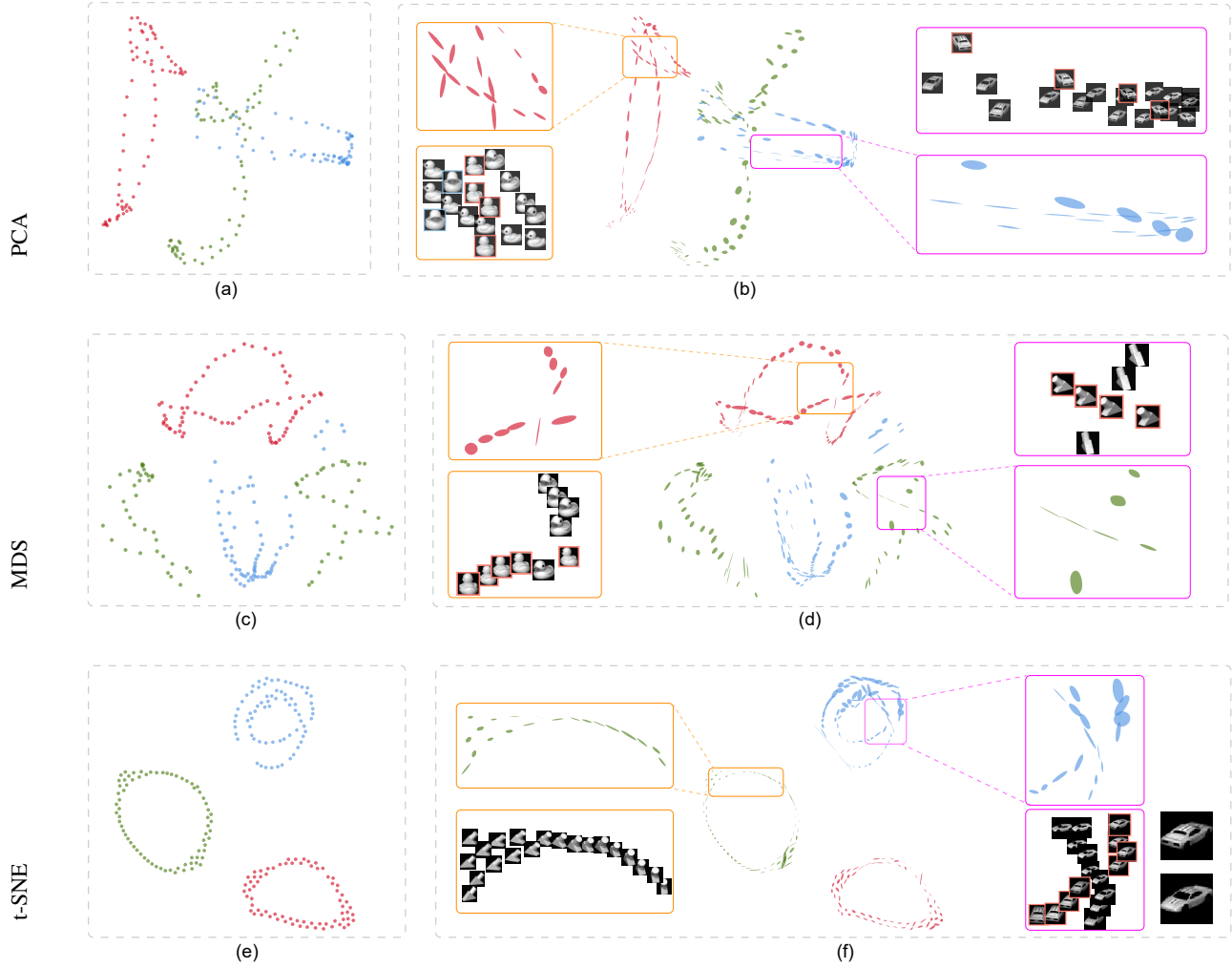
Fig. 12. Local subspace visualizations of the COIL dataset with PCA (first row), MDS (second row) shows that the different orientation and shape of the glyphs in adjacent areas correspond to different original data. And t-SNE (third row) shows that the orientation of glyphs changes smoothly within each cluster.
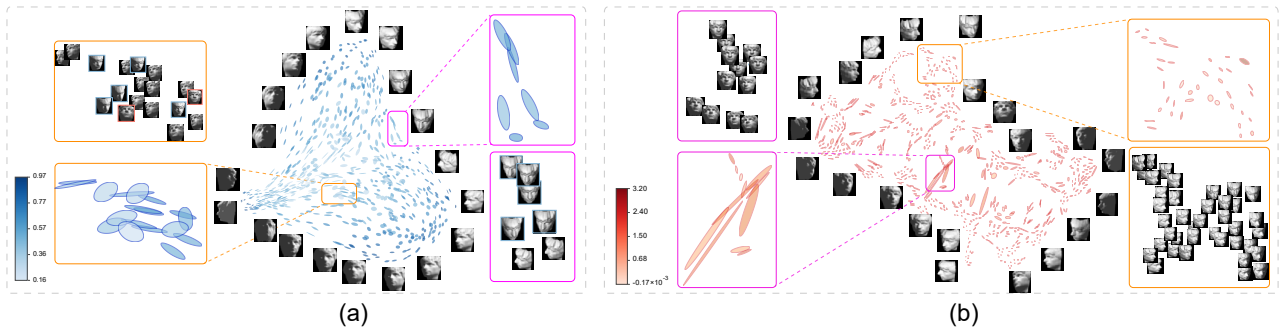

Fig. 13. Local subspace visualizations of the Face dataset with PCA (a) and t-SNE (b), which are colorzed by two different quality metrics: trustworthiness and KL divergences, respectively. From the PCA result, we can see that two trends of glyphs separate around the center, one going upward and another going downward, and in each zoom-in, the distinct types of glyphs are associated with distinct face directions. From the t-SNE result, we can identify local patterns that exhibit trends.

and compare with other explanation methods [15].

REFERENCES

[1] M. Aupetit. Visualizing distortions and recovering topology in continuous projection techniques. *Neurocomputing*, 70(7):1304–1330, 2007. Advances in Computational Intelligence and Learning. doi: 10.1016/j.neucom.2006.11.018

[2] I. Borg and P. J. Groenen. *Modern Multidimensional Scaling: Theory and Applications*. Springer, 2005. doi: 10.1111/j.1745-3984.2003.tb01108.x

[3] M. Cavallo and Ç. Demiralp. A visual interaction framework for dimensionality reduction based data exploration. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 635:1–635:13, 2018. doi: 10.1145/3173574.3174209

[4] Y. Chan, C. D. Correa, and K. Ma. Flow-based scatterplots for sensitivity analysis. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology*, pp. 43–50, 2010. doi: 10.1109/VAST.2010.5652460

[5] Y.-H. Chan, C. D. Correa, and K.-L. Ma. The generalized sensitivity scatterplot. *IEEE Transactions on Visualization and Computer Graphics*, 19(10):1768–1781, 2013. doi: 10.1109/TVCG.2013.20

[6] L. Chen and A. Buja. Local multidimensional scaling for nonlinear dimension reduction, graph drawing, and proximity analysis. *Journal of the American Statistical Association*, 104(485):209–219, 2009. doi: 10.1198/jasa.2009.0111

[7] D. B. Coimbra, R. M. Martins, T. T. Neves, A. C. Telea, and F. V. Paulovich. Explaining three-dimensional dimensionality reduction plots. *Information Visualization*, 15(2):154–172, 2016. doi: 10.1177/1473871615600010

[8] R. Cutura, S. Holzer, M. Aupetit, and M. Sedlmair. VisCoDeR: A tool for visually comparing dimensionality reduction algorithms. In *Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pp. 105–110, 2018. doi: 10.25365/thesis.51534

[9] C. De Boor. *A Practical Guide to Splines*. Springer-Verlag, 1978.

[10] D. Dua and C. Graff. UCI machine learning repository, 2017.

[11] E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2765–2781, 2013. doi: 10.1109/TPAMI.2013.57

[12] G. Ellis and A. Dix. A taxonomy of clutter reduction for information visualisation. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1216–1223, 2007. doi: 10.1109/TVCG.2007.70535

[13] M. Espadoto, R. M. Martins, A. Kerren, N. S. T. Hirata, and A. C. Telea. Towards a quantitative survey of dimension reduction techniques. *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–1, 2019. doi: 10.1109/TVCG.2019.2944182

[14] S. G. Fadel, F. M. Fatore, F. S. Duarte, and F. V. Paulovich. Loch: A neighborhood-based multidimensional projection technique for high-dimensional sparse spaces. *Neurocomputing*, 150:546 – 556, 2015. doi: 10.1016/j.neucom.2014.07.071

[15] R. Faust, D. Glickenstein, and C. Scheidegger. DimReader: axis lines that explain non-linear projections. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):481–490, 2019. doi: 10.1109/TVCG.2018.2865194

[16] K. R. Gabriel. The biplot graphic display of matrices with application to principal component analysis. *Biometrika*, 58(3):453–467, 1971. doi: 10.2307/2334381

[17] H. Goldstein. *Classical Mechanics*. Addison-Wesley Publishing Company, second ed., 1980. doi: 10.1063/1.3067728

[18] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417, 1933. doi: 10.1037/h0071325

[19] D. Husemoller. *Fibre Bundles*. Springer-Verlag, 3rd ed., 1994. doi: 10.1007/978-1-4757-2261-1

[20] P. Joia, D. Coimbra, J. A. Cuminato, F. V. Paulovich, and L. G. Nonato. Local affine multidimensional projection. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2563–2571, 2011. doi: 10.1109/TVCG.2011.220

[21] J. Lee. Quality assessment of nonlinear dimensionality reduction based on k-ary neighborhoods. *Proceedings of the Workshop on New Challenges for Feature Selection in Data Mining and Knowledge Discovery at ECML/PKDD*, 4:21–35, 2008. doi: 10.5555/3053814.3053816

[22] J. A. Lee and M. Verleysen. *Nonlinear Dimensionality Reduction*. Springer Publishing Company, 1st ed., 2007. doi: 10.1007/978-0-387-39351-3

[23] S. Lespinats and M. Aupetit. CheckViz: sanity check and topological clues for linear and non-linear mappings. *Computer Graphics Forum*, 30(1):113–125, 2011. doi: 10.1111/j.1467-8659.2010.01835.x

[24] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma. Robust recovery of subspace structures by low-rank representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):171–184, 2013. doi: 10.1109/TPAMI.2012.88

[25] S. Liu, B. Wang, J. J. Thiagarajan, P.-T. Bremer, and V. Pascucci. Visual exploration of high-dimensional data through subspace analysis and dynamic projections. *Computer Graphics Forum*, 34(3):271–280, 2015. doi: 10.1111/cgf.12639

[26] L. H. Loomis and S. Sternberg. *Advanced Calculus*. Jones and Bartlett, revised ed., 1990.

[27] M. Lu, S. Wang, J. Lanir, N. Fish, Y. Yue, D. Cohen-Or, and H. Huang. Winglets: Visualizing association with uncertainty in multi-class scatterplots. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):770–779, 2019. doi: 10.1109/TVCG.2019.2934811

[28] R. M. Martins, D. B. Coimbra, R. Minghim, and A. Telea. Visual analysis of dimensionality reduction quality for parameterized projections. *Computers & Graphics*, 41:26 – 42, 2014. doi: 10.1016/j.cag.2014.01.006

[29] L. McInnes, J. Healy, N. Saul, and L. Großberger. Umap: Uniform manifold approximation and projection. *Journal of Open Source Software*, 3(29):861, 2018. doi: 10.21105/joss.00861

[30] S. A. Nene, S. K. Nayar, and H. Murase. Columbia object image library (COIL-20). Technical Report Technical Report CUCS-005-96, 1996.

[31] H. Nguyen and P. Rosen. DSPCP: a data scalable approach for identifying relationships in parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics*, 24(3):1301–1315, 2017. doi: 10.1109/TVCG.2017.2661309

[32] L. G. Nonato and M. Aupetit. Multidimensional projection for visual analytics: Linking techniques with distortions, tasks, and layout enrichment. *IEEE Transactions on Visualization and Computer Graphics*, 25(8):2650–2673, 2019. doi: 10.1109/TVCG.2018.2846735

[33] L. Parsons, E. Haque, and H. Liu. Subspace clustering for high dimensional data: A review. *SIGKDD Explorations Newsletter*, 6(1):90–105, 2004. doi: 10.1145/1007730.1007731

[34] F. V. Paulovich, L. G. Nonato, R. Minghim, and H. Levkowitz. Least square projection: A fast high-precision multidimensional projection technique and its application to document mapping. *IEEE Transactions on Visualization and Computer Graphics*, 14(3):564–575, 2008. doi: 10.1109/TVCG.2007.70443

[35] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, second ed., 1992.

[36] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000. doi: 10.1126/science.290.5500.2323

[37] D. Sacha, L. Zhang, M. Sedlmair, J. A. Lee, J. Peltonen, D. Weiskopf, S. C. North, and D. A. Keim. Visual interaction with dimensionality reduction: A structured literature analysis. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):241–250, 2017. doi: 10.1109/TVCG.2016.2598495

[38] M. Sedlmair, T. Munzner, and M. Tory. Empirical guidance on scatterplot and dimension reduction technique choices. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2634–2643, 2013. doi: 10.1109/tvcg.2013.153

[39] C. Seifert, V. Sabol, and W. Kienreich. Stress maps: Analysing local phenomena in dimensionality reduction based visualisations. In J. Kohlhammer and D. Keim, eds., *EuroVAST 2010: International Symposium on Visual Analytics Science and Technology*, 2010. doi: 10.2312/PE/EuroVAST/EuroVAST10/013-018

[40] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proc. of 1996 IEEE Symposium on Visual Languages*, pp. 336–343. IEEE, 1996. doi: 10.1109/vl.1996.545307

[41] H. Siirtola, T. Laivo, T. Heimonen, and K.-J. Räihä. Visual perception of parallel coordinate visualizations. In *Proceedings of the International Conference Information Visualisation*, pp. 3–9, 2009. doi: 10.1109/IV.2009.25

[42] D. Smilkov, N. Thorat, C. Nicholson, E. Reif, F. B. Viégas, and M. Wattenberg. Embedding projector: Interactive visualization and interpretation of embeddings. *arXiv preprint arXiv:1611.05469*, 2016.

[43] J. Stahnke, M. Dörk, B. Müller, and A. Thom. Probing projections: Interaction techniques for interpreting arrangements and errors of dimensionality reductions. *IEEE Transactions on Visualization and Computer Graphics*,

22(1):629–638, 2016. doi: 10.1109/TVCG.2015.2467717

[44] A. Tatu, F. Maaß, I. Färber, E. Bertini, T. Schreck, T. Seidl, and D. Keim. Subspace search and visualization to make sense of alternative clusterings in high-dimensional data. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology*, pp. 63–72, 2012. doi: 10.1109/VAST.2012.6400488

[45] W. S. Torgerson. Multidimensional scaling of similarity. *Psychometrika*, 30(4):379–393, 1965. doi: 10.1007/BF02289530

[46] L. Van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(2579-2605):85, 2008.

[47] L. Van Der Maaten, E. Postma, and J. Van den Herik. Dimensionality reduction: a comparative review. *Journal of Machine Learning Research*, 10:66–71, 2009.

[48] A. Vellido, J. D. Martín-Guerrero, and P. J. Lisboa. Making machine learning models interpretable. In *Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pp. 163–172, 2012.

[49] R. Vidal. Subspace clustering. *IEEE Signal Processing Magazine*, 28(2):52–68, 2011. doi: 10.1109/MSP.2010.939739

[50] Y. Zhan, J. Yin, X. Liu, and G. Zhang. Adaptive neighborhood select based on local linearity for nonlinear dimensionality reduction. In *International Symposium on Intelligence Computation and Applications*, pp. 337–348. Springer, 2009. doi: 10.1007/978-3-642-04843-2_36

[51] L. Zhou and D. Weiskopf. Indexed-points parallel coordinates visualization of multivariate correlations. *IEEE Transactions on Visualization and Computer Graphics*, 24(6):1997–2010, 2017. doi: 10.1109/TVCG.2017.2698041