Computers & Graphics 129 (2025) 104234

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Special Section on EuroVA 2024 SI

MultiInv: Inverting multidimensional scaling projections and computing decision maps by multilateration

Daniela Blumberg [a,*], Yu Wang [b], Alexandru Telea [b], Daniel A. Keim [a], Frederik L. Dennig [a]

[a] Department of Computer and Information Science, University of Konstanz, Konstanz, 78464, Germany
[b] Department of Information and Computing Sciences, Utrecht University, Utrecht, 3584 CS, The Netherlands

ARTICLE INFO

Keywords:
Inverse projections
Dimensionality reduction
Data visualization

ABSTRACT

Inverse projections enable a variety of tasks such as the exploration of classifier decision boundaries, creating counterfactual explanations, and generating synthetic data. Yet, many existing inverse projection methods are difficult to implement, challenging to predict, and sensitive to parameter settings. To address these, we propose to invert distance-preserving projections like Multidimensional Scaling (MDS) projections by using multilateration – a method used for geopositioning. Our approach finds data values for locations where no data point is projected under the key assumption that a given projection technique preserves pairwise distances among data samples in the low-dimensional space. Being based on a geometrical relationship, our technique is more interpretable than comparable machine learning-based approaches and can invert 2-dimensional projections up to $|D|-1$ dimensional spaces if given at least $|D|$ data points. We compare several strategies for multilateration point selection, show the application of our technique on three additional projection techniques apart from MDS, and use established quality metrics to evaluate its accuracy in comparison to existing inverse projections. We also show its application to computing decision maps for exploring the behavior of trained classification models. When the projection to invert captures data distances well, our inverse performs similarly to existing approaches while being interpretable and considerably simpler to compute.

1. Introduction

Dimensionality reduction techniques, also known as multidimensional projections (MPs), are methods of choice for the analysis and visual exploration of high-dimensional data. Given such a dataset, projections create a (much) lower-dimensional version thereof in which relations such as inter-sample relationships are preserved as much as possible [1–3]. To achieve the above, a subset of projections aims to preserve *distance* relations between data samples, *e.g.*, Principal Component Analysis (PCA) [4], Multidimensional Scaling (MDS) [5], or Sammon's mapping [6]. Preserving distances, as compared to other relationships, allows a simpler interpretation of dense point clusters and neighborhoods in the resulting 2D scatterplots in terms of their shared data similarities [7–9].

Inverse projections P^{-1} aim to perform the inverse of a given projection mapping P, that is, map from the 2D space to the data space. They enable multiple applications such as user-driven data interpolation based on interaction in a 2D space for shape control [10] or electronic instrument synthesis [11]; visual exploration of trained Machine Learning (ML) models [12–14]; user-driven pseudo-labeling of training datasets [15]; assessment of distortions caused by direct projection methods [16]; and creation of counterfactual explanations, *i.e.,* help analyze why a classifier made a certain decision by generating close alternative inputs that would flip the prediction outcome [17].

While tens of direct projection techniques exist, only a handful of inverse projections have been proposed. All of these, however, suffer from various challenges such as non-smooth variations, a black-box, hard to interpret, behavior, and sometimes high computing times [18,19]. We addressed such limitations for MDS-class projections by leveraging their distance-preserving property via an algebraic *multilateration-based* inversion process [20]. We evaluate *MultiInv* – our multilateration-based inverse projection approach – on six datasets and in combination with four direct projections, namely MDS, CCA, Sammon's mapping, and PCA. We also explore six different strategies for the selection of points to compute the inversion and evaluate projection quality by considering four established quality metrics. Finally, we show how MultiInv, which is simple to implement, simple to interpret, computationally efficient, and avoids the black-box behavior of existing inverse projections, can

* Corresponding author.
E-mail addresses: daniela.blumberg@uni-konstanz.de (D. Blumberg), y.wang6@uu.nl (Y. Wang), a.telea@uu.nl (A. Telea), keim@uni-konstanz.de (D.A. Keim), frederik.dennig@uni-konstanz.de (F.L. Dennig).

https://doi.org/10.1016/j.cag.2025.104234
Received 15 January 2025; Received in revised form 22 April 2025; Accepted 23 April 2025
Available online 17 May 2025

be applied to compute gradient maps and decision maps for exploring trained ML models. To summarize, our contributions are as follows:

- We introduce MultiInv - an *interpretable* technique for the multi lateration-based inversion of distance-preserving projections.
- We give an extended *qualitative* and *quantitative* evaluation of our technique using gradient maps, decision maps, direct projection errors, and mean-squared inverse-projection errors to provide: (1) a comprehensive validation on different datasets and parameterizations, and (2) a comparison with existing inverse projection approaches.
- We publish our source code, datasets, and results on OSF.

2. Related work

2.1. Notations

We start by introducing the needed notations to describe direct and inverse projections. Let $D = \{x_i\}_{1 \le i \le |D|}$ be a dataset of high-dimensional samples $x_i \in \mathbb{R}^n$, also called observations or data points. A *projection* technique is a function $P : D \to \mathbb{R}^m$ $m \ll n$, where typically $m \in \{2, 3\}$ for visualization purposes. Without loss of generality, we confine our work to $m = 2$. For convenience, we next denote $P(D) = \{P(x_i) \mid x_i \in D\}$ to be the projection of the entire dataset D. An *inverse projection* P^{-1} is a function that maps m-dimensional points back to the high-dimensional space, defined implicitly by minimizing a cost of the form $P^{-1}(P(D)) \approx D$. A P^{-1} becomes valuable when applying it to points *outside* the set $P(D)$, which allows creating synthetic high-dimensional samples from any point $q \in \mathbb{R}^m$.

2.2. Projections

The main goal of a projection P is to preserve the underlying structure of D as well as possible in the low-dimensional space such that conclusions about D can be drawn directly from $P(D)$ [3]. Several projection techniques exist that differ in how they capture such structures and how they are technically implemented. Data structures, *i.e.,* the mapping goals of P, are usually defined by a loss function that is being minimized during the computation of P. Two main structure types can be found here: *Distance-preserving* projections aim to capture distances between data samples. For example, the objective function of Multidimensional Scaling (MDS) [5] directly optimizes for the preservation of pairwise distances by minimizing the stress $\sum_{i,j}(d_{ij} - \tilde{d}_{ij})^2$, where d_{ij} denotes the Euclidean distance in $\mathbb{R}^n$ and $\tilde{d}_{ij}$ the distance in $\mathbb{R}^m$. *Neighborhood-preserving* projections aim to map samples so their k-nearest neighbors are similar in both $\mathbb{R}^n$ and $\mathbb{R}^m$. Examples of such projections are t-SNE [21] and UMAP [22]. Projection techniques have been extensively discussed and evaluated in several surveys [1–3,8,23]. In our work, we focus on distance-preserving projections as these are simpler to interpret – distances in the projection scatterplot should directly reflect data similarity.

Projection methods can also be classified into linear or non-linear; and local or global. Linear methods map the data to $\mathbb{R}^m$ using a single linear transformation. While computationally efficient, they can, however, only capture linear structures in the data and fail for more complex datasets. Principal Component Analysis (PCA) [4] is a well-known linear technique that computes the m dimensions, called principal components, as linear combinations of the n data dimensions, by maximizing variance. This is equivalent to searching for a linear combination of dimensions that best preserves Euclidean distances [24]. In doing so, PCA focuses on global data structures. Metric Multidimensional Scaling (MDS) [5] is also a global method that is, however, non-linear. Non-linear methods allow to capture non-linear relationships in the data often existent in high-dimensional datasets with complex structures. Opposed to global methods, local approaches focus on the preservation of local data features. Projections that work similarly to MDS, *i.e.,* aim to preserve distances, but focus on local neighborhoods are Sammon's mapping [6] and Curvilinear Component Analysis (CCA) [25]. Sammon's mapping gives small distances in the data space a higher weight, *i.e.,* tends to better preserve distances of close samples in the projection. Its cost function is defined as $\sum_{i,j}(d_{ij} - \tilde{d}_{ij})^2 W(d_{ij})$, where W is a monotonically increasing weighting function that emphasizes small *data* distances. While Sammon's mapping penalizes missing neighbors, it is prone to false neighbors (see next Section 2.3). In contrast, CCA aims to preserve distances of points that are close in the projection by optimizing the cost function $\sum_{i,j}(d_{ij} - \tilde{d}_{ij})^2 W(\tilde{d}_{ij})$, where W is a monotonically increasing weighting function that emphasizes small *projected* distances.

2.3. Projection quality

Any projection function P will be unable to fully achieve its distance- or neighborhood-preserving objectives on sufficiently complex and/or high-dimensional datasets D. Practically, this leads to so-called distortions, *i.e.*, patterns in $P(D)$ which may not correspond to patterns in D; and/or patterns in D which are not captured by $P(D)$ [3,9,26]. Common distortions include *false* and *missing* neighbors. False neighbors refer to points close in $P(D)$ which are far apart in D. Missing neighbors refer to points close in D which are mapped far apart in $P(D)$. The loss functions of CCA and Sammon's mapping can be used respectively to identify false and missing neighbors [24].

Measuring and visualizing distortions is crucial to determine how well a $P(D)$ truly captures (all of) D. Several quality metrics and visualization approaches have been proposed to this end [9,24,26,27]. Quality metrics can be computed on different levels, *i.e.,* on point-pairs, neighborhoods, classes, or on an entire $P(D)$ [3,23]. We next focus on global metrics – the latter type – that are well-established and commonly used.

Trustworthiness [28] measures if the k nearest neighbors in $P(D)$ are neighbors in D, *i.e.,* is an indicator for false-neighbor distortions, and is defined as

$$\mathrm{TW}(k) = 1 - \frac{2}{|D|k(2|D| - 3k - 1)} \sum_{i=1}^{|D|} \sum_{x_j \in F_k(x_i)} (r(x_i, x_j) - k) \in [0, 1], \quad (1)$$

where $F_k(x_i)$ denotes the set of points that are among the k nearest neighbors of point x_i in $P(D)$ but not among the respective neighbors of the same point in D; and $r(x_i, x_j)$ refers to the rank of point x_j within the ordered nearest neighbors of x_i in $P(D)$. A TW value close to 1 indicates that one can trust the local patterns visible in $P(D)$.

Continuity [28] measures if the k nearest neighbors in D are also neighbors in $P(D)$, *i.e.,* indicates missing neighbors, and is computed as

$$\mathrm{CT}(k) = 1 - \frac{2}{|D|k(2|D| - 3k - 1)} \sum_{i=1}^{|D|} \sum_{x_j \in M_k(x_i)} (\hat{r}(x_i, x_j) - k) \in [0, 1], \quad (2)$$

where $M_k(x_i)$ refers to the set of points that are among the k nearest neighbors of x_i in D but not among the k nearest neighbors in $P(D)$; and $\hat{r}(x_i, x_j)$ is the rank of point x_j within the ordered nearest neighbors of x_i in D. Similar to trustworthiness, a CT value of 1 is best.

For labeled data that is ideally well separable in D, *Neighborhood Hit* [29] measures the proportion of the k nearest neighbors of a point in $P(D)$ that share the same class label as the point itself as

$$\mathrm{NH}(k) = \sum_{i=1}^{|D|} \frac{|\{x_j \in N_k(x_i) \mid l_j = l_i\}|}{|D|k} \in [0, 1], \quad (3)$$

where $N_k(x_i)$ denotes the set of k nearest neighbors of the point x_i in $P(D)$. A value NH = 1 tells that the projected data can be well separated into different classes.

Finally, *Normalized Stress* [30] measures how well a projection preserves pairwise distances as

$$\mathrm{NS} = \frac{\sum_{i,j}(d_{ij} - \tilde{d}_{ij})^2}{\sum_{i,j} d_{ij}^2} \in [0, 1], \quad (4)$$

where d_{ij} is the Euclidean distance of samples x_i and x_j in D and $\tilde{d_{ij}}$ is the Euclidean distance of the projected points $P(x_i)$ and $P(x_j)$. A distance-preserving projection should ideally produce NS = 0.

2.4. Inverse projections

An *inverse projection* is a function $P^{-1} : \mathbb{R}^m \to \mathbb{R}^n$ that aims to reverse the mapping of a given projection P for a dataset D. P^{-1} is typically constructed by minimizing a cost of the type $\sum_{x \in D} \|P^{-1}(P(x)) - x\|$, where $\|\cdot\|$ denotes the L_2 norm. Also, inverse projections should ideally be smooth so that one can use them to extrapolate synthetic data values away from the projected points $P(D)$.

While many projection techniques exist, only a few inverse projections have been proposed. One early such technique inverts a sample $x \in \mathbb{R}^2$ by Shepard interpolation of all samples $x_i \in D$ with weights $\|x - P(x_i)\|$ [11]. This inverse projection is smooth by design and can handle, in theory, any direct projection P. Yet, this method has a global nature, so it cannot handle local structures of high-dimensional datasets. Local Shepard interpolation was further refined by iLAMP [31] to invert the piecewise-linear affine projection LAMP [30]. As iLAMP uses only a small set of neighbor points, it is not smooth. Later, Amorim et al. [10] refined it using Radial Basis Functions (RBFs) to a local and piecewise continuous result. The UMAP technique [22] is, to our knowledge, the first technique (besides PCA-class methods) to jointly compute P and P^{-1}. However, UMAP's P^{-1} can be difficult to interpret as UMAP's P has a highly non-linear nature that tends to cluster similar samples, leaving large empty regions in the 2D space where P^{-1} must extrapolate far away from known points [16]. Deep learning, originally in the form of autoencoders [32], has also been used for direct and inverse projections. More recently, for a given dataset D and projection technique P, supervised deep learning, using the training set $(D, P(D))$, was used to learn both P [33–35] and P^{-1} [34,36]. Such methods are linear in the size of D, thus much faster than earlier inverse and/or direct projections, work well for any data dimensionality n, and are smooth by construction. Yet, it is hard for users to predict how deep-learned (inverse) projections actually work.

2.5. Inverse projection quality

The quality of inverse projections can be measured either *directly* by considering the inverse projection on points for which we have ground truth, *i.e.,* on projected data samples; or *indirectly* by assessing other desirable properties such as smoothness, as follows.

An inverse projection should ideally yield $P^{-1}(P(x)) = x$ for all $x \in D$. For a single $x_i \in D$, this can be evaluated by the *mean-squared error* (MSE) defined as

$$\begin{aligned} \mathrm{MSE}(x_i) &= \frac{1}{n} \sum_{j=1}^{n} (x_{ij} - P^{-1}(P(x_{ij})))^2 \\ &= \frac{1}{n} \|x_i - P^{-1}(P(x_i))\|^2. \end{aligned} \tag{5}$$

For an entire dataset D, the average MSE can be computed as

$$\begin{aligned} \mathrm{MSE}(D) &= \frac{1}{|D|} \sum_{i=1}^{|D|} \mathrm{MSE}(x_i) \\ &= \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{1}{n} \|x_i - P^{-1}(P(x_i))\|^2. \end{aligned} \tag{6}$$

Note that Eq. (6) can also be applied to subsets of D [16,36]. This is useful when splitting data into train and test sets.

For points $q \in (\mathbb{R}^2 - P(D))$, *i.e.,* points 'unseen' by P, we have no 'ground truth' in terms of samples in D that project there. These points are exactly the ones where we want to practically use an inverse projection later on to create synthetic data samples. The *gradient map* technique [16] can be used to evaluate the quality of an inverse projection for such unknown points by computing the total pseudo-derivative of P^{-1}. To this end, we need to quantize $q \in Z^2$ to a pixel location on a discrete 2D grid, such that we can estimate the gradient magnitude of P^{-1} at q via a symmetric finite difference filter in horizontal and vertical directions as

$$G(q) = \sqrt{\|P^{-1}(q_r) - P^{-1}(q_l)\|^2 + \|P^{-1}(q_u) - P^{-1}(q_d)\|^2},$$

where q_r, q_l, q_u, and q_d are the right, left, up, and down 4-neighboring pixels of q. When $G(q)$ has nearly constant (and, ideally, low) values over a 2D area, one-pixel 'moves' of q in that area cause only small changes to the inferred samples $P^{-1}(q) \in \mathbb{R}^n$, *i.e.*, P^{-1} is smooth at these locations, which is highly desirable. For instance, users interactively selecting points q in such areas will get new data points in a confined, controlled area in $\mathbb{R}^n$. Conversely, when $G(q)$ has high values, small changes in 2D, *e.g.*, caused by a user interactively moving the point q, can suddenly 'throw away' the generated samples $P^{-1}(q)$ into far-apart regions in $\mathbb{R}^n$, in a way that is perceived as confusing and uncontrollable by the user. Applications such as projection-assisted interpolation or data synthesis then become hard to support [10–12].

2.6. Decision maps

Projections are designed to preserve similarities in the data space D in the scatterplots $P(D)$ they create. Hence, such scatterplots can be used to judge the classification difficulty of a dataset: If a scatterplot, which has high projection quality metrics, fails to clearly separate points from different classes, this may indicate that a classifier will face challenges in achieving high accuracy for that dataset [37]. Decision maps extend the above idea of visually capturing the quality of a classification model to points outside a training or test set D in a dense fashion. They are created using inverse projections as follows: Each pixel of a 2D image is mapped back to the data space via P^{-1}, next classified by the model, and then colored to depict its corresponding class. Regions in the map with the same color then show the decision zones, and neighboring pixels of different colors depict the decision boundary of the classifier. Some algorithms, like DBM [12], are projection-agnostic, *i.e.*, they can be applied by combining any suitable direct and inverse projections P and P^{-1}. Other methods, like SDBM [14], jointly learn a projection and its inverse. Yet other methods leverage the inherent invertibility of some projection algorithms, as shown by Schulz et al. [13], who use a supervised version of UMAP to create decision maps.

The hard-to-predict nature of inverse projections – related to their limited smoothness discussed above in Section 2.5 – can lead to decision maps showing fragmentation and jagged decision boundaries. Filtering poorly projected points has been used to alleviate this [12]. Yet, artifacts still remain, so users need ways to know if these are caused by the inverse projection or the actual ML model under examination. As such, fundamental issues concerning the interpretability of inverse projections remain. Recently, Wang et al. [18,19] explored the limitations of decision maps – and thus implicitly their underlying inverse projections – showing the limited interpretability and predictability of current inverse projection methods.

3. Method

We next present *MultiInv*, our multilateration-based inverse projection method as proposed by Blumberg et al. [20]. The main idea of this approach builds upon the goal of metric MDS, which aims to map distances $\|x_i - x_j\|$, $x_i, x_j \in D$ to distances $\|P(x_i) - P(x_j)\|$ up to a scaling factor [38]. This property, expressed as $\|x_i - x_j\| \approx \|P(x_i) - P(x_j)\|$, is key to our approach. Further, we leverage geometrical relationships among the data samples by using multilateration. This principle, also known as trilateration in $\mathbb{R}^2$, aims to find a point's position via its distances to other known points. Typically used for geopositioning, multilateration can be applied in any Euclidean space [39].

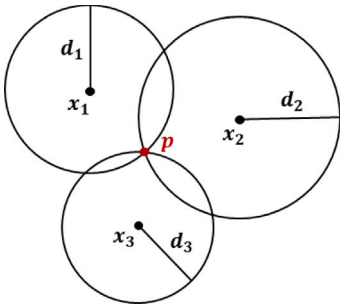

Fig. 1. Through *trilateration*, the position of point $p \in \mathbb{R}^2$ can be found given three reference points x_1, x_2, $x_3 \in \mathbb{R}^2$ and their distances (i.e., the scalar values d_1, d_2, d_3) to point p.

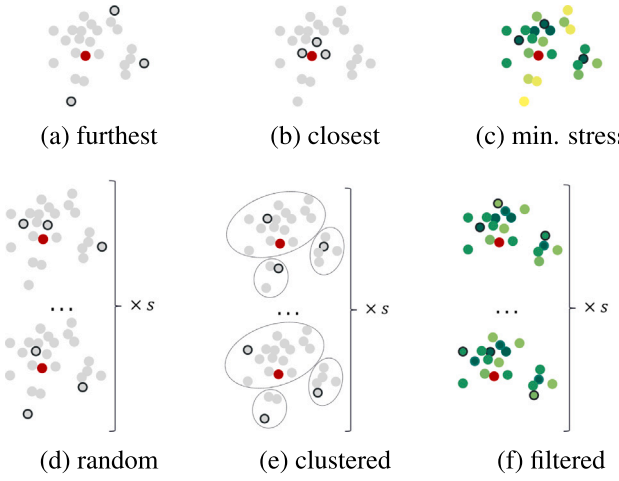

Fig. 2. Reference point selection strategies for multilateration (a)–(c) using a fixed set of reference points and (d)–(f) using multiple randomized iterations. The target point to be inversely projected is marked in red. Reference points are indicated by a black border. (c), (f) Green hues encode local stress values; darker green shows low values, and yellow higher values, respectively.

2D Example: We want to compute point $p = (p_1, p_2)^T$ given three known points $x_1 = (x_{1,1}, x_{1,2})^T$, $x_2 = (x_{2,1}, x_{2,2})^T$, and $x_3 = (x_{3,1}, x_{3,2})^T$ in $\mathbb{R}^2$ and their distances d_1, d_2, and d_3 to p (see Fig. 1). We get that

$$(x_{1,1} - p_1)^2 + (x_{1,2} - p_2)^2 = d_1^2$$
$$(x_{2,1} - p_1)^2 + (x_{2,2} - p_2)^2 = d_2^2$$
$$(x_{3,1} - p_1)^2 + (x_{3,2} - p_2)^2 = d_3^2.$$

We subtract one equation (here the first) from the other two:

$$(x_{2,1}-p_1)^2 + (x_{2,2}-p_2)^2 - (x_{1,1}-p_1)^2 - (x_{1,2}-p_2)^2 = d_2^2 - d_1^2$$
$$(x_{3,1}-p_1)^2 + (x_{3,2}-p_2)^2 - (x_{1,1}-p_1)^2 - (x_{1,2}-p_2)^2 = d_3^2 - d_1^2.$$

Rewriting and simplifying these equations to

$$\begin{aligned} &-2(x_{2,1} - x_{1,1})p_1 - 2(x_{2,2} - x_{1,2})p_2 \\ &= d_2^2 - d_1^2 - (x_{2,1}^2 - x_{1,1}^2) - (x_{2,2}^2 - x_{1,2}^2) \\ &-2(x_{3,1} - x_{1,1})p_1 - 2(x_{3,2} - x_{1,2})p_2 \\ &= d_3^2 - d_1^2 - (x_{3,1}^2 - x_{1,1}^2) - (x_{3,2}^2 - x_{1,2}^2) \end{aligned}$$

eliminates the squares of our target variables p_1 and p_2. Solving the resulting linear equation system then gives us p.

Generalization to n Dimensions: For any n-dimensional space with $n < |D|$, take $n+1$ known samples $x_1, \dots, x_{n+1}$ in $\mathbb{R}^n$, where $x_i = (x_{i,1}, \dots, x_{i,n})^T \in \mathbb{R}^n$, $i = 1, \dots, n+1$. Consider also a point $p = (p_1, \dots, p_n)^T \in \mathbb{R}^n$ and its distances d_i to each sample x_i. For each sample x_i, we have $\|x_i - p\| = d_i \Leftrightarrow \|x_i - p\|^2 = (d_i)^2 \Leftrightarrow \sum_{j=1}^{n}(x_{ij} - p_j)^2 = (d_i)^2$. Assuming ideal distance preservation by the direct projection P, the distances d_i are equal to $\|P(x_i) - P(p)\|$. Next, if we have a *given* point $q \in \mathbb{R}^2$ – the one we want to inversely project to p – then we have that $P(p) = q$ (since we want that $P^{-1}(q) = p$). Thus, we can compute $\|P(x_i) - P(p)\|$ directly as $\|P(x_i) - q\|$. The above gives us $n+1$ equations with n unknowns $p_1, \dots, p_n$. As the unknowns are squared, we subtract the first equation (w.l.o.g.) from all others. This yields a linear equation system of the form $Ap = b$ with

$$A = \begin{bmatrix} -2(x_{2,1} - x_{1,1}) & \dots & -2(x_{2,n} - x_{1,n}) \\ \vdots & \ddots & \vdots \\ -2(x_{n+1,1} - x_{1,1}) & \dots & -2(x_{n+1,n} - x_{1,n}) \end{bmatrix}$$

and

$$b = \begin{bmatrix} (d_2)^2 - \sum_{j=1}^{n}(x_{2,j})^2 - ((d_1)^2 - \sum_{j=1}^{n}(x_{1,j})^2) \\ \vdots \\ (d_{n+1})^2 - \sum_{j=1}^{n}(x_{n+1,j})^2 - ((d_1)^2 - \sum_{j=1}^{n}(x_{1,j})^2) \end{bmatrix}.$$

To determine the position of a sample $p \in \mathbb{R}^n$, we need to know the positions of $n+1$ other samples and their distances to p. Hence, we have the constraint that the dataset size $|D|$ must exceed the data dimensionality n, an assumption which holds for most datasets.

3.1. Choosing reference points for multilateration

Choosing the samples x_i that are used as reference points in the multilateration computation is a crucial aspect of our approach. Using all samples in D would yield a method similar to [11], which, as mentioned in Section 2.4, cannot capture local structures well – at a higher level, one can say that such an approach overfits the data. Also, our approach requires using precisely $n+1$ points for the equation system not to be over- or underdetermined. Ideally, we expect $Ap = b$ to provide exactly one solution for the position of p. However, if some of the samples x_i are collinear or coincident, the matrix A is singular (its determinant is zero), *i.e.,* the equation system becomes degenerate and does not yield a unique solution. Separately, a projection usually does not preserve distances equally well across all samples in D, so certain subsets of $n+1$ samples in D taken as reference points may lead to more accurate inverse projections. We address this by evaluating several selection strategies for choosing the reference points that can be categorized into (1) strategies using a fixed subset of $n+1$ samples from D as reference points for a specific target point, or (2) strategies using multiple randomized iterations considering multiple potential subsets as reference points for the target point.

Fixed Set of Reference Points: Our first approaches use a fixed subset of D as reference points for a specific target point $q \in \mathbb{R}^2$. This means that, to inversely project a single q, we deterministically select exactly $n+1$ samples. First, we considered using the samples corresponding to the $n+1$ *furthest* projected points, *i.e.*, those having the largest distances d_i to q (Fig. 2(a)). This approach could, in theory, work well for projections that aim to optimize global neighborhoods. Alternatively, we used the samples corresponding to the $n+1$ *closest* points to q in 2D (Fig. 2(b)). The inverse projection should then behave more like the local iLAMP approach [10] or, more generally, work for projections that focus on local neighborhoods.

While taking the furthest or closest points is simple to implement, we might consider poorly projected samples as reference points. If we think, for instance, of outliers being projected far from other points, the selection gets biased towards these points or vice versa for points being projected into the center of the 2D plot. When the distances of the corresponding data samples to the other samples in D are not well

preserved by the projection, our inverse projection will not be accurate. To counteract choosing poorly projected points as references, we next propose to select a set of $n+1$ samples with minimal distortions (Fig. 2(c)). For this, we calculate the average squared deviation of pairwise distances of a data sample to all other samples in D compared to the distances of their projections, *i.e.,* a local stress value as

$$\text{stress}_i = \frac{1}{|D|-1} \sum_{j \neq i} (d_{ij} - \tilde{d_{ij}})^2, \tag{7}$$

where d_{ij} is the Euclidean distance between samples x_i and x_j in D; and $\tilde{d_{ij}}$ is the Euclidean distance between $P(x_i)$ and $P(x_j)$. We then select the $n+1$ best-projected data points, *i.e.,* those having minimal local stress values. The downside of this approach is that, apart from inversely projecting points that are among the best-projected ones and cannot be reference points for themselves, we now use exactly the *same set* of points as references independent of target point q. Moreover, the selected points may lie in close proximity or in an unfortunate arrangement, possibly restricting the information to a concealed area of the data, with reference points possibly being collinear or coincident.

Multiple Randomized Iterations: To address the limitations stemming from a fixed set of reference points, we propose several strategies based on the idea of majority voting. Specifically, for a given target point, we consider several subsets of D as references, as follows. First, for a given 2D point q to invert, we select $n+1$ samples from D randomly and compute the resulting $p = P^{-1}(q)$ using these samples as reference points (Fig. 2(d)). We repeat this s times and finally set p to the medoid of all the generated values. Simply put, taking the medoid acts as a low-pass filter that limits the potentially undesirable effects due to a poor (random) selection of samples from D. Further, we tested a stratified random approach incorporating *clustering* (Fig. 2(e)). We first cluster D using k-means with $k = n+1$ (our target number of samples to select) and then randomly select one point from each cluster. We repeat the selection s times and take the medoid of the generated positions. Clustering is expected to reduce cases where selected points are collinear or coincident. However, due to the restriction to select exactly one point from each cluster, we might always select some poorly projected points if they are assigned to the same cluster, *e.g.,* outliers. Hence, we propose a third randomized approach that applies *filtering* to the data (Fig. 2(f)). Specifically, we filter out poorly projected data samples from the list of reference point candidates by computing the local stress values as in Eq. (7) and removing the 20% with the highest distortions. We then apply the simple randomized strategy with the remaining 80% as reference point candidates s times and take the medoid of the generated values. Note that, when using filtering, s is expected to be smaller compared to the other randomized strategies, as we already filtered out poorly projected samples. We evaluate and compare the performance of these different reference point selection strategies in Section 4.1.

4. Evaluation and results

We evaluate the quality of MultiInv using established quality measures for inverse projections (see Section 2). For known samples, we compute the per-sample mean-squared error (MSE) and provide the average MSE over all samples (Eq. (6)) as a global quality metric for each projected dataset. To evaluate the inverse projection quality for unknown points, we show gradient maps [16] as a visual means of evaluation. As explained earlier, gradients should be overall constant and low-valued, so that neighboring points in 2D are mapped closely in the high-dimensional space by P^{-1}.

We further used MultiInv to compute decision maps. As explained in Section 2.6, such maps show the behavior of a trained classification model as decision zones (*i.e.*, data space areas where the same label is inferred) separated by decision boundaries (*i.e.*, places where the model changes the inferred label) [12–14]. As classification models for the decision maps, we used a k nearest neighbor classifier with $k = 5$, a random forest classifier, and logistic regression. Such classifiers should produce relatively compact decision zones with smooth boundaries [19]. As such, we can evaluate the quality of our inverse projection by checking for these attributes in the created decision maps. For the generation of gradient and decision maps, we use a grid size of 250×250 pixels.

Table 1
Datasets used in our evaluation with dimensionality n, intrinsic dimensionality ratio ρ_n, and number of samples $|D|$.

Dataset	n	ρ_n	$\lvert D\rvert$
Plane	3	0.67	200
Rings	3	1	180
Blobs	10	0.6	1100
Iris [40]	4	0.5	150
Seismic [41]	24	0.29	646
Bank [42]	63	0.46	2059

Datasets: We applied our method on synthetic data with a known topology as well as on real-world datasets: (1) *Plane* is a simple synthetic dataset of two same-sized clusters of points in 3D that lie approximately on a 2D plane, *i.e.,* the data structure can be approximated well by only 2 dimensions. (2) *Rings* is a synthetic dataset containing 3D points arranged in two interlacing rings having 100 and 80 samples, respectively. Given the interlacing of rings, this represents a scenario where any projection to 2D induces a loss of information. (3) *Blobs* is a synthetic dataset of 1000 points generated from a Gaussian distribution with 5 clusters and 100 additional randomly distributed noise samples with a dimensionality of $n = 10$. (4) *Iris* [40] serves as a simple real-world application and represents three different flower species with four dimensions and 150 samples. (5) *Seismic* [41] shows a real-world scenario with more dimensions ($n = 24$) and includes data samples of seismic bumps in a coal mine. (6) *Bank* [42] involves data of a direct marketing campaign from a Portuguese bank used to predict whether a customer will subscribe to a banking product or not and represents the most high-dimensional included dataset with $n = 63$. An overview of the datasets is given in Table 1. The intrinsic dimensionality ratio ρ_n [23] measures the percentage of principal components (computed by PCA) that are needed to explain 95% of the variance in the data. Thus, it indicates whether the projection may encounter difficulties when mapping the data to 2D.

Projections: As direct projection methods, we considered metric MDS, CCA, Sammon's mapping, and PCA, as they all have a distance-preserving objective (see Section 2.2). While MDS and PCA are global methods, CCA and Sammon's mapping focus on local neighborhoods, enabling us to evaluate our P^{-1} for diverse types of projections. The quality of an inverse projection depends, by construction, on the quality of the direct projection P one aims to invert. As such, we want to make sure that our direct projections we start with are of high quality in the first place. We measure this by the trustworthiness, continuity, neighborhood hit, and normalized stress metrics (see Section 2). For the first three, we set a value of $k = 7$, following [8,23,43]. Table 2 shows these measurements for all included datasets and projections. As visible, we get quite high quality, meaning our inputs are a viable basis for inversion through multilateration or, more generally, any other inverse projection method.

4.1. Comparison of reference point selection strategies for multilateration

We evaluated six different strategies for selecting the reference points (see Section 3). Table 3 provides the average MSE as quality metric for inverting known points (*i.e.,* projected data samples) for these different strategies on all included datasets and projection methods. Among the three strategies with a fixed set of reference points, the selection of points with *minimum local stress* values clearly outperforms the others, having consistently lower MSE values. The *furthest* and the

Table 2
Projection quality measurements with established metrics (see Section 2) for different datasets and projection techniques. Arrows in brackets indicate good values for the respective metric. All values are rounded to the second decimal place.

Dataset	TW (↑)				CT (↑)				NH (↑)				NS (↓)			
	MDS	CCA	Sammon	PCA	MDS	CCA	Sammon	PCA	MDS	CCA	Sammon	PCA	MDS	CCA	Sammon	PCA
Plane	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
Rings	.99	.98	.98	.98	1	1	1	1	.96	.92	.92	.95	.01	.07	.06	.02
Blobs	.93	.92	.92	.92	.94	.95	.97	.96	.99	.98	.97	.97	.02	.1	.09	.06
Iris	.98	.97	.97	.97	.99	.99	.99	.99	.95	.94	.94	.94	0	0	0	0
Seismic	.96	.96	.96	.97	.98	.98	.98	.99	.9	.9	.9	.9	.04	.23	.23	.1
Bank	.88	.8	.8	.85	.9	.9	.9	.9	.85	.85	.85	.85	.13	.47	.47	.38

Table 3
Average Mean-Squared-Error on whole dataset D of MultiInv with different reference point selection strategies using six datasets and four projection methods. The values are rounded to the fourth decimal place. The best-performing strategy is highlighted for each dataset and direct projection.

Dataset	furthest	closest	min. stress	random	clustered	filtered	furthest	closest	min. stress	random	clustered	filtered
	MDS						**CCA**					
Plane	2.2033	0.0842	**0.0003**	0.0005	0.0017	**0.0003**	0.0011	0.0087	0.0003	**0.0002**	0.0003	**0.0002**
Rings	6.1×10^{24}	1.4×10^{24}	**0.014**	0.0154	0.0145	0.0145	4.04×10^{22}	9.67×10^{22}	0.0398	0.0268	0.0228	**0.024**
Blobs	26.8795	39.9975	0.0541	**0.0244**	0.1012	0.028	3.6068	4994.72	0.0263	0.0252	0.0412	**0.0213**
Iris	828	7.3×10^{24}	0.0179	0.0049	0.0064	**0.0041**	176.62	2.55×10^{55}	**0.0032**	0.0033	0.0033	0.0033
Seismic	1.3×10^{14}	1.9×10^{20}	**0.1889**	7.2821	2.8×10^{14}	8.3341	9.2×10^{12}	1.2×10^{20}	**0.1814**	4.6731	3.6×10^{12}	4.6177
Bank	23 873	340.51	58.2284	10.2324	50.0446	**7.3672**	1.2×10^{5}	77.6901	**0.4611**	0.8906	5.8215	0.8809
	Sammon						**PCA**					
Plane	0.0011	0.0087	0.0003	**0.0002**	**0.0002**	**0.0002**	**0.0**	0.0013	**0.0**	**0.0**	**0.0**	**0.0**
Rings	6.0×10^{26}	4.7×10^{25}	0.6165	0.0275	**0.0232**	0.025	9.8×10^{21}	1.2×10^{22}	0.3434	0.0131	0.0137	**0.0126**
Blobs	421.61	3314.46	6.3059	0.0274	0.1299	**0.0234**	0.2985	29.2552	**0.0143**	0.0206	0.0425	0.0163
Iris	120.28	3.1×10^{23}	**0.0032**	0.0034	0.0035	**0.0032**	259.96	6.9×10^{24}	0.003	**0.0029**	0.003	**0.0029**
Seismic	9.2×10^{12}	1.2×10^{20}	**0.1814**	4.3715	2.7×10^{12}	4.7279	2×10^{14}	2.3×10^{20}	**0.1852**	7.3388	9.8×10^{12}	4.5916
Bank	1.2×10^{5}	77.7166	**0.4692**	0.8896	5.8236	0.8721	87.9265	9.8×10^{5}	1.0846	1.0864	3.8416	**0.8708**

closest point selection strategies produce a few very high values for MSE(x_i), resulting in a high average MSE for multiple datasets and projection techniques. This is most likely due to reference points being poorly chosen, *i.e.*, being collinear, coincident, or having distances not well preserved by the projection.

Fig. 4 visualizes the local stress values for all tested datasets and projection methods, showing that pairwise distances are indeed not equally well preserved by the projections. For MDS, high-stress values tend to appear especially for samples along the outer edge of projected data closer to the projection borders and are most prominent among outliers. The only exception is the *Rings* dataset, where high-stress values appear on the projected intersections of the rings. Since these intersections are not representative of the original 3D data, distances of these data points towards the others are most distorted. Similar observations of local stress values can be made for the other projections as well, with the difference that high-stress values are not only prominent at the borders of the projection but tend to appear in the center as well. In general, however, points inside the same cluster tend to be equally well projected with a *smooth* change of local stress values over close points. As the *minimum stress* strategy takes the points whose distances are best preserved as references, it does not as easily encounter high MSE(x_i) values as the furthest and closest point selection strategies. However, as reference points are chosen independently of the target point, yielding an almost completely fixed set of reference points for a given dataset, this method works like a linear transformation of the space with respect to these points. This becomes evident when looking at the resulting gradient and decision maps as exemplified by the *Blobs* dataset for the PCA projection in Fig. 3. The technique produces perfectly smooth and similarly low gradients for the whole plot. Most points in the decision

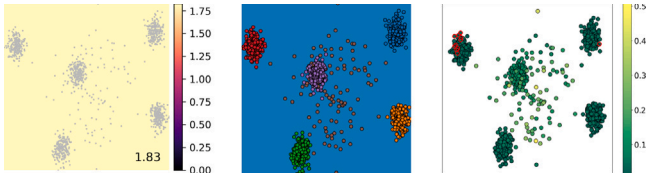

Fig. 3. Gradient map (left) and decision map (middle) for the *Blobs* dataset when using PCA as projection and MultiInv as inverse projection. Reference points (marked red on the right) for multilateration were selected based on the minimum local stress value (right). The decision map was generated by training a k nearest neighbor classifier with $k = 5$ and coloring each pixel according to its assigned class when inversely projecting it to the data space.

map are misclassified as belonging to the blue-colored cluster in the upper right corner, although they are closer to projected points of other classes. This is due to the restricted fixed set of reference points, where one is taken from the blue cluster and all remaining from the red cluster.

To tackle the issues stemming from a fixed set of reference points, we used *randomized* approaches. Our previous evaluation [20] showed that, for a sufficiently high number of selection steps s, the errors of inversely projecting points from D more or less stabilize beyond a certain number of trials. Hence, we selected s for each dataset individually, so that the error does not fluctuate much, *i.e.,* is representative for the strategy. In a stratified random approach, we incorporated *clustering* with the restriction to select exactly one point from each cluster as reference point. While clustering is expected to reduce cases where selected points are collinear or coincident, it seems that for some

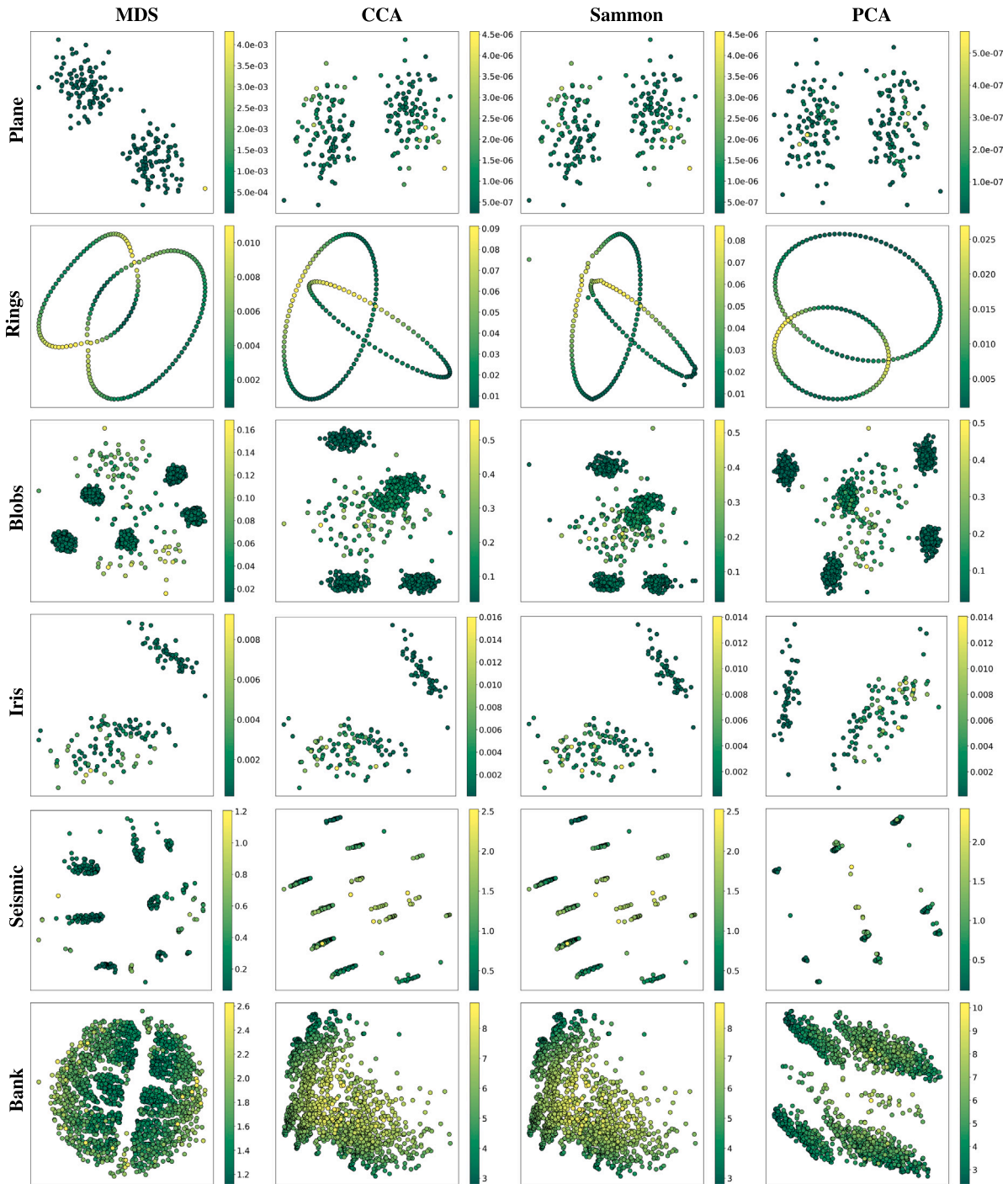

Fig. 4. Local stress values for different datasets and projections. Low stress values are colored in dark green and high stress values in yellow. Points are ordered according to the stress values such that points with high values are plotted on top of the others.

datasets, we always select some poorly projected points, resulting in high errors. This is shown by the simple randomized approach (without clustering) outperforming the clustering one in terms of average MSE (see Table 3).

Filtering out the worst 20% of data samples and applying the randomized strategy with the remaining 80% as reference point candidates is an alternative reference point selection strategy. When comparing the simple randomized with the filtering strategy, we can, however, see that no strategy strongly outperforms the other since the MSE values are more or less on the same level. When looking at the gradient ranges in Fig. 6, we can notice that they tend to be smaller for the filtered random approach than the simple random one. As the simple randomized and the randomized with filtering strategies performed overall best among the reference point selections, we use them both for the remaining evaluation.

4.2. Comparison to existing inverse projection techniques

We next compare our MultiInv approach to the inverse projection methods iLAMP [31], iNN [16,36], and RBF [10], as these techniques are also usable in conjunction with a user-chosen projection technique. For each method, we use the same parameters as proposed in the respective paper. We left SSNP [34] and autoencoders [32] out from the comparison since these compute projection P and inverse projection

Table 4
Average Mean-Squared-Error (MSE) on test data (20% of whole data) for the inverse projection techniques under evaluation using six datasets and four projection methods. Results are rounded to the fourth decimal place. The best-performing combination of direct and inverse projection is highlighted for each dataset.

Dataset	*Ours - random*	*Ours - filtered*	iLAMP	iNN	RBF	*Ours - random*	*Ours - filtered*	iLAMP	iNN	RBF
	MDS					**CCA**				
Plane	.0003	.0002	.0006	.0002	.0002	.0002	.0002	.0007	.0001	.0001
Rings	.0141	.0161	.0012	.0012	**.0008**	.0274	.0241	.0014	.0009	.001
Blobs	.0244	.0286	.0057	**.0053**	.0059	.0255	.0221	.0056	.0059	.006
Iris	.0039	.004	.0034	.0038	.0036	.0029	.0024	.004	.0031	.0029
Seismic	8.1972	7.6382	.009	.1601	.0086	3.9003	4.0219	.0071	.0074	.0069
Bank	10.1569	7.4881	**.0616**	.0641	.0651	.914	.8877	.078	.0735	.0809
	Sammon					**PCA**				
Plane	.0002	.0002	.0007	.0001	.0001	**.0**	**.0**	.0007	.0001	.0001
Rings	.0364	.0213	.0014	.0009	.0011	.0139	.0133	.0014	.0012	.0012
Blobs	.028	.0226	.0061	.0059	.0063	.021	.0164	.0062	.0059	.0064
Iris	.0025	.003	.0043	.0034	.0031	.0024	**.0022**	.0044	.0028	.0033
Seismic	3.8713	4.7056	.0071	.0081	.0069	6.8332	4.2455	.005	.1257	**.0049**
Bank	.8905	.8854	.078	.0749	.081	1.101	.8767	.0705	.0657	.0736

P^{-1} *jointly* (see Section 2). As such, we cannot use these to invert a given projection, *i.e.,* in our case MDS, CCA, Sammon's mapping, and PCA.

Evaluation on known points: To assess the average MSE of unseen test samples, we split the data into training and testing subsets. Technically, our approach is a *lazy learner* [44], *i.e.,* has no training phase and does not need a separate test set for evaluating its quality. Yet, to be able to fairly compare our P^{-1} with inverse projections that require training, we restrict multilateration to only use training samples as reference point candidates. Table 4 shows the average MSE of the test samples for each inverse projection technique when utilizing different projections, 80% of the data for training and the remaining 20% for testing. For the synthetic *Plane* and the real-world *Iris* data, our method manages to achieve a low average MSE for all projections under evaluation, which is competitive towards the results of iLamp, iNN, and RBF. What these datasets have in common is that both have an *intrinsic dimensionality* $n \cdot \rho_n$ of 2 (see Table 1), meaning that two dimensions are already sufficient to represent 95% of the data variance. This indicates that under good projection conditions, *i.e.,* almost no loss of information induced by P, multilateration is a valid and well-performing inverse projection technique. However, the average MSEs also show that our method encounters limitations when the projection induces significant information loss. For the higher-dimensional datasets *Seismic* and *Bank*, the average MSE on the test data is much higher for MultiInv than for the other evaluated inverse projection techniques. However, the difference varies strongly across the four projection techniques with MDS tending to produce higher errors specifically for our P^{-1}. This shows that different projections produce low-dimensional representations of the same dataset that vary strongly in quality, which in turn strongly influences the quality of the inverse projection techniques.

While Table 4 shows global indicators of the quality of inverse projections, it does not explain the location or sources of errors. Fig. 5 covers this in greater detail for the *Blobs* dataset projected by MDS. For visual assessment of the MSEs, we show a 2D scatterplot of $P(D)$ and color-code its Voronoi diagram, where $P(x_i)$ are the centroids of the cells, with values $\mathrm{MSE}(x_i)$. We use a luminance colormap with dark mapping a high MSE and bright a low MSE, respectively. Technically, this approximates $\mathrm{MSE}(x_i)$ over the entire image space using piecewise-constant interpolation and highlights regions of high and low distortions of the MDS projection and its inversion [24,26]. We see a notable difference in the MSE visualization of our method as compared to the other inverse projections in the distribution of errors. Our method spreads darker colors, *i.e.,* higher MSE values, across *multiple* Voronoi cells, rather than having them concentrated in cells around the projected test samples. This shows the relative *independence* of our approach from the choice of training and test data, unlike ML-based methods. For our method, we also see that areas in the middle of the projection scatterplot (being surrounded by many data points) have a lower MSE than points at the projection borders. When comparing these patterns to the local stress values (Fig. 4), we can retrieve similar distributions. This shows (1) the strong dependency of our method on projection quality and (2) the applicability of multilateration as an indicator for projection distortions.

Evaluation on unknown points: Gradient map visualizations (see Fig. 6 for MDS), which show the inverse projection's behavior away from the known points in $P(D)$, match the above findings. Despite having comparably high gradient ranges, these maps are quite smooth for our method, showing uniformly low values in the projection center surrounded by data samples in $P(D)$. iLAMP yields high gradients visible as reticulated lines in gaps between projected point-clusters and in areas with no projected points. In contrast, our method shows low gradient values even in areas between clusters, where no data points exist, see *e.g.* the *Iris* and *Plane* datasets. Both our method and iLAMP show uniformly low gradient values near and around projected points. Conversely, RBF's gradient maps show relatively high values between projected points and low values in regions with no projected points. The gradient map for iNN shows a mix of the above. Notably, for *Rings*, RBF and iNN produce high gradients at the intersection of the rings in the projection, *i.e.*, regions where the projection is misleading as the rings do not intersect or touch in the original 3D data space. Overall, RBF and iNN perform best among the inverse projections in terms of low gradient ranges. When the dataset dimensionality and complexity increase (see *e.g. Seismic* and *Bank*), our method reaches its limits, yielding high gradients, especially in the map corners. Yet, the gradients are relatively smooth in comparison to *all* other inverse projections. As mentioned in Section 2.5, low and constant gradients are essential for the practical usage of an inverse projection method. These general findings for MDS are mostly true for all projections under investigation. The respective gradient maps can be found in the Appendix.

Fig. 7 shows the decision maps computed by our method for a $k = 5$-nearest neighbor classifier and our six datasets projected by MDS. While these maps slightly differ from those produced by a random forest or logistic regression classifier (see Appendix) for the same projection technique, some patterns appear independently of the used classifier, indicating that they are caused by the inverse projection and not the classifier. The iLamp patterns for the decision map match the gradient

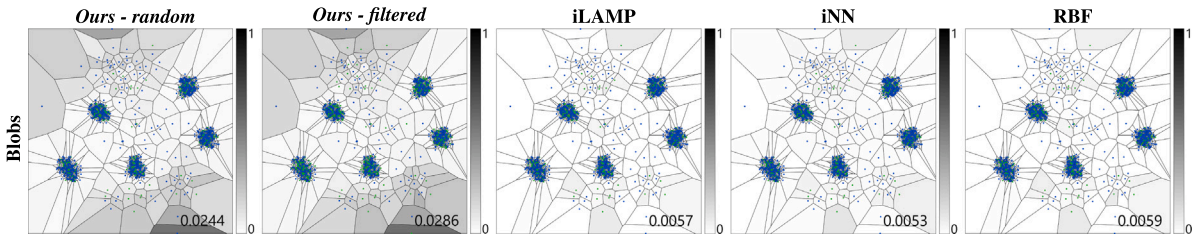

Fig. 5. Comparison of inverse projections showing the MSE for known points encoded in Voronoi cells associated with each projected sample of the *Blobs* datasets when applying MDS. The number at the bottom right corner indicates the average MSE for the test samples (20% of the dataset; colored in green).

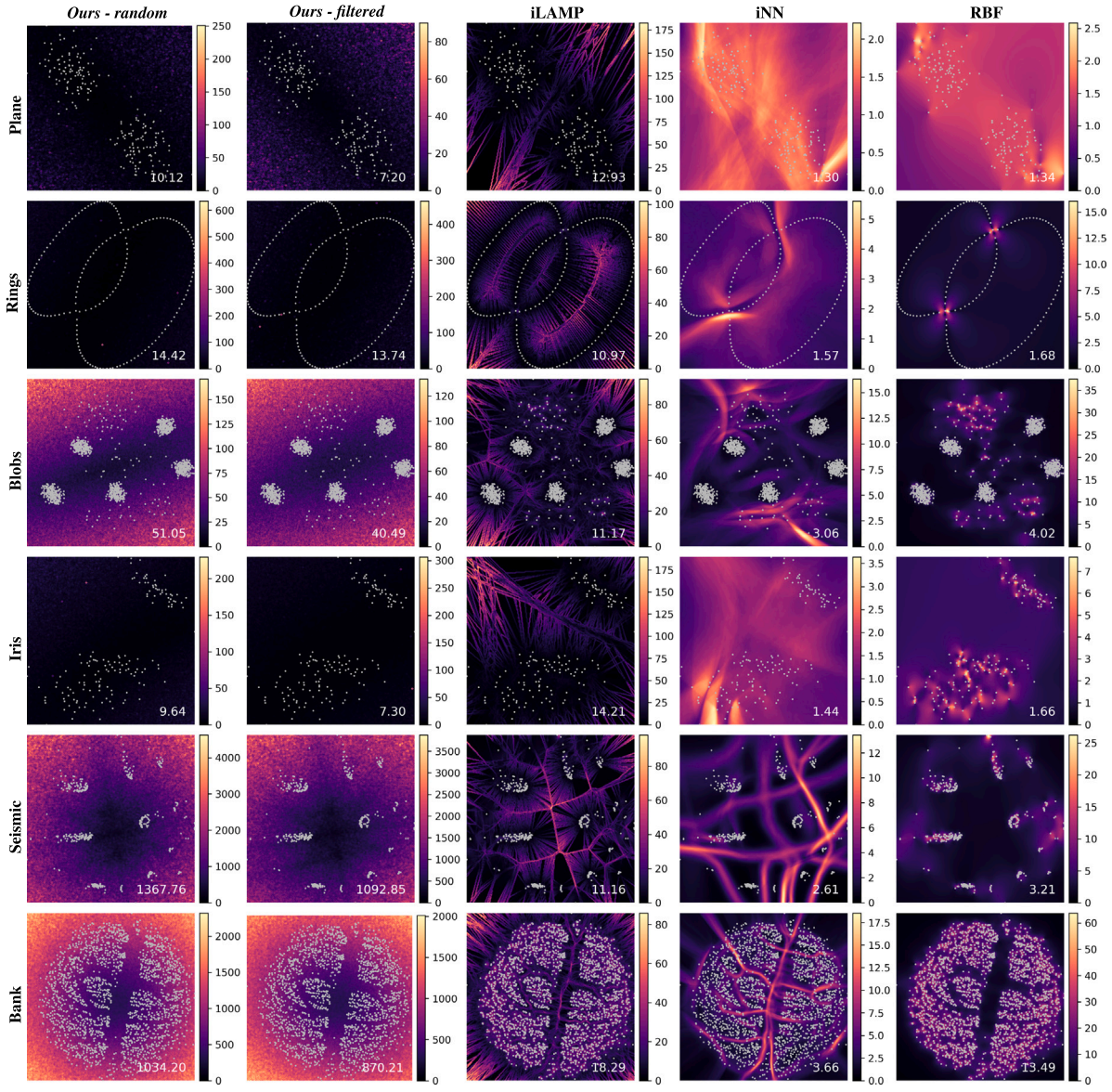

Fig. 6. Gradient maps of inverse projections for six datasets projected using MDS. Darker colors indicate a low rate of change, and lighter areas show a high rate of change. The number at the bottom right shows the average gradient.

map results, *i.e.,* decision boundaries are fragmented and jagged in areas where no data points are projected. For the *Plane* dataset, we argue that our inverse projection method produces the best map – with an almost linear boundary that separates the scatterplot into almost equally sized classes. For the other datasets, the boundaries appear more jagged with our approach than the others. This demonstrates uncertainties regarding the class in these regions. Otherwise, we can see a good separation of classes. For *Rings* and *Blobs*, we further see clear differences between the decision maps when using MultiInv: For *Rings*, the map is incorrect close to the deceptive intersection of the two rings, which is a result of the loss of information induced by the projection. This shows again that our approach depends more on projection quality than other inverse projections. For *Blobs*, the main difference is that the decision zones of brown noise samples are far smaller in the maps created by our method. Yet, this is not the case when using the random forest classifier instead of the k nearest neighbor classifier. The more

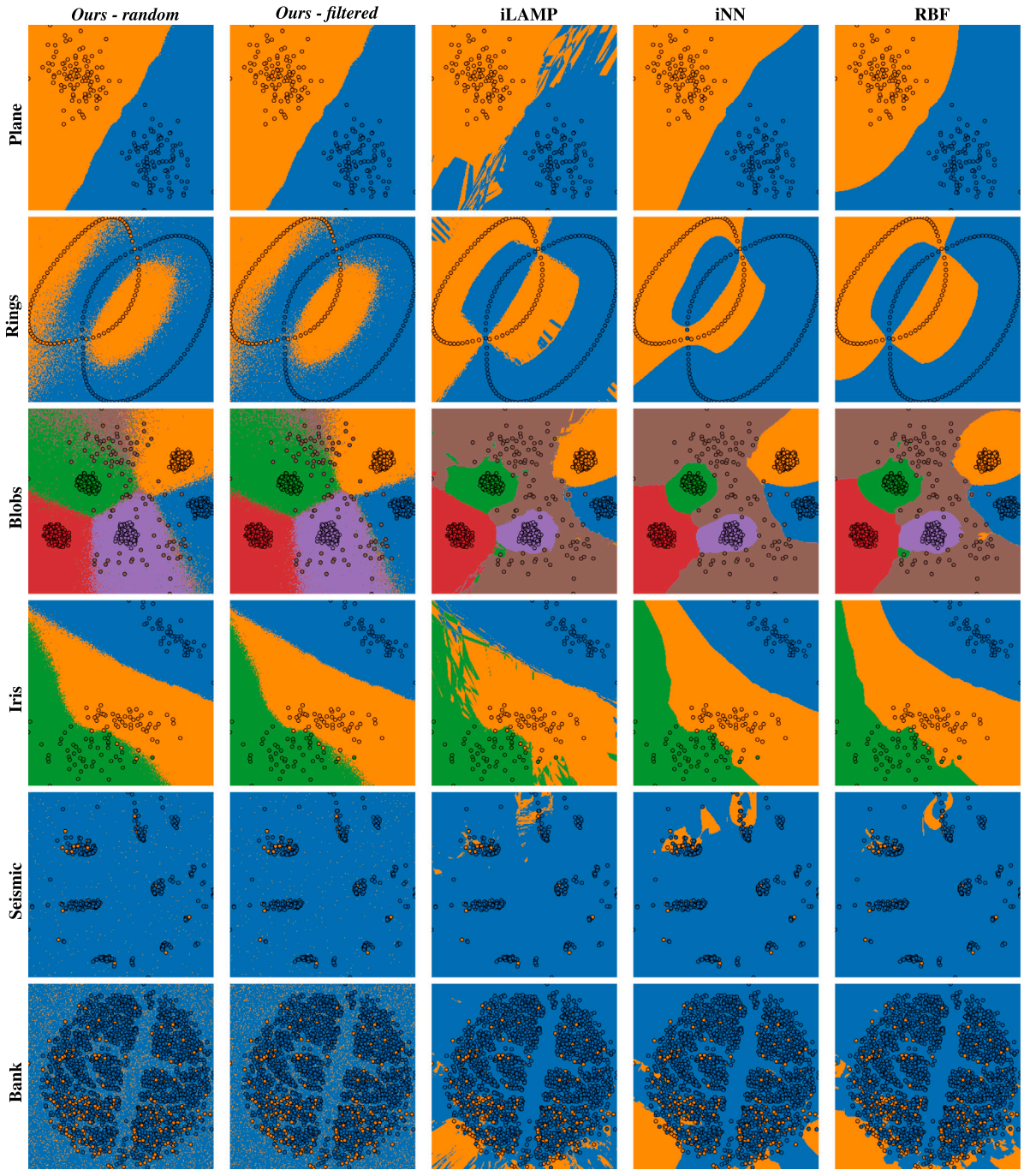

Fig. 7. Decision maps of inverse projections for six datasets projected using MDS and classified by a k-nearest neighbor classifier with $k = 5$.

high-dimensional datasets *Seismic* and *Bank* demonstrate again the limitations of our approach. Yet, the classification is for those datasets not as easy as for the others since classes are not well separated by the projection. Similar results for all other projections and classifiers are given in the Appendix.

5. Discussion

We next discuss several key points of our approach:

Quality: Our results indicate that the effectiveness of *MultiInv* strongly depends on the quality of the direct projection P we aim to invert and also on the dataset's (intrinsic) dimensionality and configuration of data samples. We found that the quality of our approach diminishes with increasing complexity and dimensionality of the dataset. There are several possible reasons for this: (1) sparse data, (2) high intrinsic dimensionality, (3) distortions in the direct projection, and (4) the curse of dimensionality [45].

The two points – distance-preserving nature of P and dataset dimensionality – are subtly interconnected. Our key assumption is that P is able to preserve distances well, something that can be measured by low stress values. When this is the case, our method can yield accurate inverse projections. However, this constraint does not hold for (a) methods P which were not designed with the goal of distance preservation, *e.g.*, neighborhood-preserving methods like t-SNE or UMAP; or (b) datasets which, due to their intrinsic dimensionality, cannot be projected well while keeping distances preserved. As such, we summarize the *application scope* of our method to be for situations

when one knows to have a dataset that can be well projected while preserving distances. For other cases, we deem more general inverse projection methods, such as iNN, as more suitable – with the incurred constraints of such methods, *i.e.,* the need for a training phase and the possibility of overfitting.

Simplicity: Our method is simple to implement, works for basically any n-dimensional unlabeled quantitative dataset, and requires only the number of selection steps s as a free parameter. This contrasts with other inverse projection methods like SSNP (which needs labeled data) or any deep-learning method (which needs an architecture tailored to the input dataset apart from carefully tuned hyperparameter values).

Scalability: Solving linear equation systems has a runtime complexity of $\mathcal{O}(N^3)$ (worst case) and $\mathcal{O}(N^2)$ (best case) for N equations, which corresponds to computing the inverse of a single target point. For us, $N = n + 1$ for an n-dimensional dataset. Such costs can be decreased if one looks for an approximate solution – which should be fine in practice for visualization purposes. When using a randomized reference point selection strategy for multilateration, the cost is multiplied by the number of selection steps s. However, when using the randomized strategy with filtering, the required number of selection steps is expected to be lower. Additionally, our method does not require a training step, such as iNN [36], SSNP [34], or autoencoders [32]. Compared to these methods, our computing effort is limited only to the inference step. Separately, our technique can be used with progressive visualizations [46] since our randomized strategies improve in accuracy with each iteration – a feature that, to our knowledge, none of the existing inverse projection methods has.

Future Work: While inverse projections can enrich projection scatterplots [3,24,47], they must be interpretable to be practically effective. Algorithm-wise, we claim our technique meets this goal more than other existing techniques. Yet, improvements are possible. Despite our negative results partitioning the data by k-means for stratified sampling and the limitations of a fixed set of reference points, we believe that the reference point selection can be further optimized. Future work can explore alternative reference point selection strategies that work on a non-random basis. When incorporating filtering of data samples to select reference points, split proportions other than 0% and 20% could be further investigated. There is also a need to evaluate the limitations of our approach, including its runtime. Especially for the novel randomized reference point selection strategy incorporating filtering based on local stress values, it can be interesting to investigate its influence on the required number of selection steps s in comparison to the simple randomized approach. As explained, our method's effectiveness is tied to distance-preserving projections. Future work could explore adaptations or modifications that enable generalization to other popular techniques like t-SNE or UMAP, which do not explicitly optimize for distance preservation. In general, inverting high-dimensional data by multilateration will require many samples to avoid problems stemming from sparsity. When comparing the quality of our method for reconstructing known data samples, we detected a match of MSE patterns to local stress values. Thus, the applicability of multilateration for evaluating projection methods could be exploited in greater detail.

6. Conclusion

We proposed *MultiInv*, an extended approach for the multilateration-based inversion of MDS-class projections. Our technique is purely based on Euclidean geometry and the assumption of distance preservation. This avoids the need for more complex machine learning and deep-learning based approaches, which are hard to fine-tune, understand, interpret, and predict. Algorithm-wise, our method is simple to implement, efficient to compute, and can be applied to any generic high-dimensional dataset where the dataset size exceeds the dimensionality. We evaluate our inverse projection method by qualitatively and quantitatively comparing it to three key existing approaches for inverse projections, using metric MDS, CCA, Sammon's mapping, and PCA as direct projections. Quantitatively, our approach only yields similar errors in inversely projecting known points if the distances are well preserved by the projection, indicated by a low intrinsic dimensionality of the dataset. Qualitatively, our method produces smoother gradients between projected points (and clusters thereof), meaning that our method may be better suited for applications where users (interactively) change the position of a 2D point to infer a smoothly changing data value. Decision maps computed by our method strengthen these findings. Still, our approach reaches limitations when the dimensionality or complexity of the underlying data is high and projections induce a high loss of information.

CRediT authorship contribution statement

Daniela Blumberg: Writing – original draft, Visualization, Software, Methodology, Data curation, Conceptualization. **Yu Wang:** Writing – review & editing, Visualization, Software, Investigation. **Alexandru Telea:** Writing – review & editing, Writing – original draft, Data curation. **Daniel A. Keim:** Writing – review & editing, Funding acquisition. **Frederik L. Dennig:** Writing – review & editing, Supervision, Methodology, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We acknowledge financial support by the Federal Ministry for Economic Affairs and Climate Action (BMWK, grant No. 03EI1048D) and the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project-ID 251654672 – TRR 161 (Project A03).

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.cag.2025.104234.

Data availability

I have shared the link to my data and code in the paper.

References

[1] Sorzano COS, Vargas J, Pascual-Montano AD. A survey of dimensionality reduction techniques. 2014, CoRR arXiv:1403.2877.
[2] Cunningham JP, Ghahramani Z. Linear dimensionality reduction: survey, insights, and generalizations. J Mach Learn Res 2015;16:2859–900. http://dx.doi.org/10.5555/2789272.2912091.
[3] Nonato LG, Aupetit M. Multidimensional projection for visual analytics: Linking techniques with distortions, tasks, and layout enrichment. IEEE Trans Vis Comput Graph 2019;25(8):2650–73. http://dx.doi.org/10.1109/TVCG.2018.2846735.
[4] Jolliffe IT. Principal component analysis. Springer series in statistics, Springer; 1986, http://dx.doi.org/10.1007/978-1-4757-1904-8.
[5] Kruskal JB. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. Psychometrika 1964;29(1):1–27. http://dx.doi.org/10.1007/BF02289565.
[6] Sammon Jr JW. A nonlinear mapping for data structure analysis. IEEE Trans Comput 1969;18(5):401–9. http://dx.doi.org/10.1109/T-C.1969.222678.
[7] Yin H. Nonlinear dimensionality reduction and data visualization: A review. Int J Autom Comput 2007;4(3):294–303. http://dx.doi.org/10.1007/S11633-007-0294-Y.
[8] van der Maaten L, Postma E, van den Herik J. Dimensionality reduction: A comparative review. Tech. Rep, (TiCC TR 2009-005). Tilburg University, Netherlands; 2009.
[9] Heulot N, Fekete J, Aupetit M. Visualizing dimensionality reduction artifacts: An evaluation. 2017, CoRR arXiv:1705.05283. arXiv:1705.05283.

[10] Amorim E, Brazil EV, Mena-Chalco J, Velho L, Nonato LG, Samavati F, et al. Facing the high-dimensions: Inverse projection with radial basis functions. Comput Graph 2015.

[11] van Wijk JJ, van Overveld CWAM. Preset based interaction with high dimensional parameter spaces. In: Data visualization: the state of the art. Kluwer; 2003, p. 391–406.

[12] Rodrigues FCM, Espadoto M, Hirata R, Telea AC. Constructing and visualizing high-quality classifier decision boundary maps. Information 2019;10(9):280. http://dx.doi.org/10.3390/INFO10090280.

[13] Schulz A, Hinder F, Hammer B. DeepView: Visualizing classification boundaries of deep neural networks as scatter plots using discriminative dimensionality reduction. In: 29th international joint conference on artificial intelligence. ijcai.org; 2020, p. 2305–11. http://dx.doi.org/10.24963/IJCAI.2020/319.

[14] Oliveira AAM, Espadoto M, Hirata Jr. R, Telea AC. SDBM: Supervised Decision Boundary Maps for Machine Learning Classifiers. In: 17th international joint conference on computer vision, imaging and computer graphics theory and applications. SCITEPRESS; 2022, p. 77–87. http://dx.doi.org/10.5220/0010896200003124.

[15] Benato B, Grosu C, Falcao A, Telea A. Human-in-the-loop: Using classifier decision boundary maps to improve pseudo labels. Comput Graph 2024.

[16] Espadoto M, Appleby G, Suh A, Cashman D, Li M, Scheidegger C, et al. UnProjection: Leveraging inverse-projections for visual analytics of high-dimensional data. IEEE Trans Vis Comput Graph 2021;29(2):1559–72. http://dx.doi.org/10.1109/TVCG.2021.3125576.

[17] Schlegel U, Rauscher J, Keim DA. Interactive counterfactual generation for univariate time series. In: Workshop on eXplainable knowledge discovery in data mining (XKDD) @ ECML-pKDD. 2024.

[18] Wang Y, Machado A, Telea AC. Quantitative and qualitative comparison of decision-map techniques for explaining classification models. Algorithms 2023;16(9):438. http://dx.doi.org/10.3390/A16090438.

[19] Wang Y, Telea A. Fundamental limitations of inverse projections and decision maps. In: 19th international joint conference on computer vision, imaging and computer graphics theory and applications. SciTePress; 2024, p. 571–82. http://dx.doi.org/10.5220/0012352300003660.

[20] Blumberg D, Wang Y, Telea A, Keim DA, Dennig FL. Inverting Multidimensional Scaling Projections Using Data Point Multilateration. In: 16th int. euroVis workshop visual analytics. The Eurographics Association; 2024, http://dx.doi.org/10.2312/eurova.20241112.

[21] van der Maaten L, Hinton G. Visualizing data using t-SNE. J Mach Learn Res 2008;9(86):2579–605, URL http://jmlr.org/papers/v9/vandermaaten08a.html.

[22] McInnes L, Healy J, Melville J. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. 2018, http://dx.doi.org/10.48550/arXiv.1802.03426, CoRR arXiv:1802.03426 arXiv:1802.03426.

[23] Espadoto M, Martins RM, Kerren A, Hirata NST, Telea AC. Toward a quantitative survey of dimension reduction techniques. IEEE Trans Vis Comput Graph 2019;27(3):2153–73. http://dx.doi.org/10.1109/TVCG.2019.2944182.

[24] Lespinats S, Aupetit M. CheckViz: Sanity check and topological clues for linear and non-linear mappings. Comput Graph Forum 2011;30(1):113–25. http://dx.doi.org/10.1111/J.1467-8659.2010.01835.X.

[25] Demartines P, Hérault J. Curvilinear component analysis: a self-organizing neural network for nonlinear mapping of data sets. IEEE Trans Neural Networks 1997;8(1):148–54. http://dx.doi.org/10.1109/72.554199.

[26] Aupetit M. Visualizing distortions and recovering topology in continuous projection techniques. Neurocomputing 2007;70(7–9):1304–30. http://dx.doi.org/10.1016/J.NEUCOM.2006.11.018.

[27] Martins RM, Coimbra DB, Minghim R, Telea AC. Visual analysis of dimensionality reduction quality for parameterized projections. Comput Graph 2014.

[28] Venna J, Kaski S. Neighborhood preservation in nonlinear projection methods: An experimental study. In: International conference on artificial neural networks. Lecture notes in computer science, vol. 2130, Springer; 2001, p. 485–91. http://dx.doi.org/10.1007/3-540-44668-0_68.

[29] Paulovich FV, Nonato LG, Minghim R, Levkowitz H. Least square projection: A fast high-precision multidimensional projection technique and its application to document mapping. IEEE Trans Vis Comput Graph 2008;14(3):564–75. http://dx.doi.org/10.1109/TVCG.2007.70443.

[30] Joia P, Coimbra DB, Cuminato JA, Paulovich FV, Nonato LG. Local affine multidimensional projection. IEEE Trans Vis Comput Graph 2011;17(12):2563–71. http://dx.doi.org/10.1109/TVCG.2011.220.

[31] dos Santos Amorim EP, Brazil EV, II JD, Joia P, Nonato LG, Sousa MC. iLAMP: Exploring high-dimensional spacing through backward multidimensional projection. In: 7th IEEE conf. vis. anal. sci. technol.. IEEE Computer Society; 2012, p. 53–62. http://dx.doi.org/10.1109/VAST.2012.6400489.

[32] Hinton GE, Salakhutdinov RR. Reducing the dimensionality of data with neural networks. Science 2006;313(5786):504–7. http://dx.doi.org/10.1126/science.1127647.

[33] Espadoto M, Hirata NST, Telea AC. Deep learning multidimensional projections. Inf Vis 2020;19(3):247–69. http://dx.doi.org/10.1177/1473871620909485.

[34] Espadoto M, Hirata NST, Telea AC. Self-supervised dimensionality reduction with neural networks and pseudo-labeling. In: 16th international joint conference on computer vision, imaging and computer graphics theory and applications. SCITEPRESS; 2021, p. 27–37. http://dx.doi.org/10.5220/0010184800270037.

[35] Appleby G, Espadoto M, Chen R, Goree S, Telea AC, Anderson EW, et al. HyperNP: Interactive visual exploration of multidimensional projection hyperparameters. Comput Graph Forum 2022;41(3):169–81. http://dx.doi.org/10.1111/CGF.14531.

[36] Espadoto M, Rodrigues FCM, Hirata NST, Hirata Jr R, Telea AC. Deep learning inverse multidimensional projections. In: 10th international euroVis workshop on visual analytics. Eurographics Association; 2019, p. 13–7. http://dx.doi.org/10.2312/EUROVA.20191118.

[37] Rauber PE, Falcão AX, Telea AC. Projections as visual aids for classification system design. Inf Vis 2018;17(4):282–305. http://dx.doi.org/10.1177/1473871617713337.

[38] de Leeuw J, Mair P. Multidimensional Scaling Using Majorization: SMACOF in R. J Stat Softw 2009;31(3):1–30. http://dx.doi.org/10.18637/jss.v031.i03.

[39] Widdison E, Long DG. A review of linear multilateration techniques and applications. IEEE Access 2024;12:26251–66. http://dx.doi.org/10.1109/ACCESS.2024.3361835.

[40] Fisher RA. The use of multiple measurements in taxonomic problems. Ann Eugen 1936;7(2):179–88. http://dx.doi.org/10.1111/j.1469-1809.1936.tb02137.x.

[41] Sikora M, Wróbel L. Application of rule induction algorithms for analysis of data collected by seismic hazard monitoring systems in coal mines. Arch Min Sci 2010;55(1):91–114.

[42] Moro S, Cortez P, Rita P. A data-driven approach to predict the success of bank telemarketing. Decis Support Syst 2014;62:22–31. http://dx.doi.org/10.1016/J.DSS.2014.03.001.

[43] Martins RM, Minghim R, Telea AC. Explaining neighborhood preservation for multidimensional projections. In: Computer graphics and visual computing. Eurographics Association; 2015, p. 7–14. http://dx.doi.org/10.2312/CGVC.20151234.

[44] Han J, Kamber M, Pei J. Classification: Advanced Methods. In: Data mining: concepts and techniques. third ed.. Morgan Kaufmann; 2012, http://dx.doi.org/10.1016/B978-0-12-381479-1.00009-5.

[45] Verleysen M, François D. The curse of dimensionality in data mining and time series prediction. In: Computational intelligence and bioinspired systems. Lecture notes in computer science, vol. 3512, Springer; 2005, p. 758–70. http://dx.doi.org/10.1007/11494669_93.

[46] Ulmer A, Angelini M, Fekete J, Kohlhammer J, May T. A survey on progressive visualization. IEEE Trans Vis Comput Graph 2024;30(9):6447–67. http://dx.doi.org/10.1109/TVCG.2023.3346641.

[47] Dennig FL, Miller M, Keim DA, El-Assady M. FS/DS: a theoretical framework for the dual analysis of feature space and data space. IEEE Trans Vis Comput Graph 2024;30(8):5165–82. http://dx.doi.org/10.1109/TVCG.2023.3288356.