EuroVis Workshop on Visual Analytics (2024)
M. El-Assady and H.-J. Schulz (Editors)

Inverting Multidimensional Scaling Projections Using Data Point Multilateration

Daniela Blumberg[1], Yu Wang[2], Alexandru Telea[2], Daniel A. Keim[1], and Frederik L. Dennig[1]

[1]University of Konstanz, Germany
[2]Utrecht University, Netherlands

Abstract

Current inverse projection methods are often complex, hard to predict, and may require extensive parametrization. We present a new technique to compute inverse projections of Multidimensional Scaling (MDS) projections with minimal parametrization. We use mutilateration, a method used for geopositioning, to find data values for unknown 2D points, i.e.*, locations where no data point is projected. 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 given a minimum of $|D|$ data points. We qualitatively and quantitatively compare our technique with existing inverse projection techniques on synthetic and real-world datasets using mean-squared errors (MSEs) and gradient maps. When MDS captures data distances well, our technique shows performance similar to existing approaches. While our method may show higher MSEs when inverting projected data samples, it produces smoother gradient maps, indicating higher predictability when inverting unseen points.*

CCS Concepts

• ***Human-centered computing*** → *Visualization techniques;*

1. Introduction

Dimensionality reduction techniques, also called projections, are a key approach to visualizing high-dimensional data. These methods work by reducing the data to lower dimensions while trying to preserve its inherent relationships and structure. One such relationship is the similarity of data samples, usually expressed by their distance in high-dimensional space [SVP14, CG15, NA19]. A particular class of such methods based on Multidimensional Scaling (MDS) aims to *preserve* distances between the projected samples as close as possible to distances between the respective samples in the data space [Kru64]. Such methods create scatterplots that are easier to interpret when tasks require the assessment of sample similarities [Yin07, vdMPvdH09, HFA17]. While projections P map the data space to the visual space, *inverse projections* P^{-1} attempt the opposite. Given a point $\mathbf{q}$ in the visual (2D) space, its inverse projection is a data sample which, if projected by P, should yield a location close to $\mathbf{q}$. Inverse projections are most valuable for points in areas where no data is projected by P, which in the following we call *unknown* points. Inverse projections can be used as a generative model to create synthetic data and have multiple applications, such as user-driven data interpolation based on interaction in a 2D space for shape control [ABM*15] or electronic instrument synthesis [vWvO03]; visual exploration of trained Machine Learning (ML) models [REHT19, SHH20, OEJT22]; and assessment of distortions caused by direct projection methods [EAS*21]. Current inverse projection techniques suffer from artifacts and interpretability issues [WMT23, WT24]. We address these limitations for MDS-class methods. Specifically, we leverage their distance-preserving property in a *multilateration-based* inversion process, allowing us to invert MDS by algebraic expressions. Hence, our proposed method is simple to implement, simple to interpret, computationally efficient, and avoids the black-box behavior of deep learning-based inverse projections. To summarize, we contribute the following:

- An *interpretable* technique for the multilateration-based inversion of MDS projections.
- A *qualitative* and *quantitative* evaluation using gradient maps and mean-squared errors for comparison with existing approaches.
- We publish our source code, datasets, and results on OSF.

2. Related Work on Inverse Projections

We start by introducing some notations. Let $D=\{x_i\}_{1\leq i\leq|D|}$ be a dataset of high-dimensional samples $x_i \in \mathbb{R}^n$. A *projection* technique P maps D to $P(D)=\{P(x_i)|x_i \in D\}$, where $P(D) \subset \mathbb{R}^q$ with $q \ll n$. Usually $q=2$, so $P(D)$ can be directly visualized in a 2-dimensional scatterplot. Projection techniques have been extensively discussed and evaluated in several surveys [vdMPvdH09, SVP14, CG15, EMK*19, NA19].

Inverse projections are functions $P^{-1}:\mathbb{R}^q \to \mathbb{R}^n$ that (a) are smooth (needed by applications as interpolation); and (b) minimize a cost of the type $\sum_{x\in D}\|P^{-1}(P(x))-x\|$ for a given dataset D and

DOI: 10.2312/eurova.20241112

direct projection P, where $\|\cdot\|$ denotes the L_2 norm. While many projection techniques exist, only a few inverse projections have been proposed. A first such technique inverts a sample $x \in \mathbb{R}^2$ by Shepard interpolation of all samples $x_i \in D$ with weights $\|x - P(x_i)\|$ [vWvO03]. 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 [dSABI*12] to invert the piecewise-linear affine projection LAMP [JCC*11]. As iLAMP is not smooth, Amorim *et al.* [ABM*15] refined it using Radial Basis Functions (RBFs) to a local and piecewise continuous result. The UMAP technique [MHM18] was, 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 [EAS*21]. Deep learning has been used for direct and inverse projections, i.e., autoencoders [HS06]. Recently, supervised deep learning, using the training set $(D, P(D))$, was used for learning both P [EHT20, EHT21, AEC*22] and P^{-1} [ERH*19, EHT21]. Such methods are faster than earlier inverse projections (i.e., linear in the sample size) and work well for any data dimensionality n. Yet, it is hard for users to predict how deep-learned inverse projections actually work. To address this issue, Espadoto *et al.* [EAS*21] proposed several visualizations to examine distortions caused by P and P^{-1} methods.

Inverse projections have been used to compute so-called *decision maps* which depict the behavior of an ML 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) [REHT19, SHH20, OEJT22]. The hard-to-predict nature of inverse projections has led to decision maps showing fragmentation and jagged decision boundaries. Filtering poorly projected points has been used to alleviate this [REHT19]. 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. Fundamental issues concerning the interpretability of inverse projections remain. Recently, Wang *et al.* [WMT23, WT24] explored the limitation of such decision maps – and thus implicitly their underlying inverse projections – showing the limited interpretability and predictability of current inverse projection methods.

3. Method

As direct projection P, we consider 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 [dLM09]. 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 process, 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, this can be applied in any Euclidean space [WL24]. 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, which is true for most datasets.

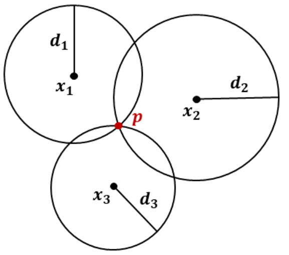

Figure 1: *Trilateration in* $\mathbb{R}^2$.

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 Figure 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 from the others to eliminate the squares. 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 $x_1, \ldots, x_{n+1}$ in $\mathbb{R}^n$, where $x_i = (x_{i,1}, \ldots, x_{i,n})^T \in \mathbb{R}^n$, $i = 1, \ldots, n+1$. Consider also a point $p = (p_1, \ldots, p_n)^T \in \mathbb{R}^n$ and its distances d_i to x_i. For each i, we have $\|x_i - p\| = d_i \Leftrightarrow \|x_i - p\|^2 = (d_i)^2 \Leftrightarrow \sum_{j=1}^n (x_{i,j} - p_j)^2 = (d_i)^2$. Assuming ideal distance preservation by MDS, 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 $P^{-1}(q) = p$), so we can compute $\|P(x_i) - P(p)\|$ directly as $\|P(x_i) - q\|$. The above gives us $n+1$ equations with n unknowns $p_1, \ldots, p_n$. As the unknowns are squared, we subtract the first equation (w.l.o.g.) from all others to get rid of the squared values. This yields a linear equation system of the form $Ap = b$ with

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

and

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

Choosing Reference Points for Multilateration: Choosing the samples x_i is a crucial aspect of our approach. Using all samples in D would yield a method similar to [vWvO03], which, as mentioned in Sec. 2, cannot capture local structures well. Moreover, 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, MDS usually does not preserve distances equally well across all samples in D, so certain subsets of $n+1$ samples in D may lead to more accurate inverse projections. We address this by evaluating several approaches for selecting a subset of $n+1$ samples from D.

4. Evaluation

We evaluate the quality of our inverse projection with two measures and four datasets, as follows.

Average Mean-Squared Error: An inverse projection should ideally yield $P^{-1}(P(x)) = x$ for all $x \in D$. Hence, the quality of P^{-1} for a single $x_i \in D$ can be measured by the mean-squared error

$$MSE(x_i) = \frac{1}{n}\sum_{j=1}^{n}(x_{i,j} - P^{-1}(P(x_{i,j})))^2 = \frac{1}{n}\|x_i - P^{-1}(P(x_i))\|^2.$$

Taking the average over the entire dataset D, leads to the quality measure

$$MSE(D) = \frac{1}{|D|}\sum_{i=1}^{|D|} MSE(x_i) = \frac{1}{|D|}\sum_{i=1}^{|D|}\frac{1}{n}\|x_i - P^{-1}(P(x_i))\|^2. \quad (1)$$

Note that $MSE(D)$ can also be applied to subsets of D [ERH*19]. This will become useful when dividing the data into train and test sets. For visual assessment, 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 $MSE(x_i)$, using a luminance colormap (dark=high MSE; bright=low MSE). Technically, this approximates $MSE(x_i)$ over the entire image space using piecewise-constant interpolation, and highlights regions of high and low distortions of the MDS projection and its inversion [Aup07, LA11].

Error for Unseen Points: For points $q \in \mathbb{R}^2 \setminus P(D)$, we have no 'ground truth' in terms of samples in D that are projected there by P. As such, we need different ways to evaluate the inverse projection at these locations – which are also the ones where we want to practically use the inverse projection. For this, we use the gradient map technique proposed in [EAS*21]. For each $q \in \mathbb{R}^2$, this computes the total pseudo-derivative of P^{-1} as

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

where q_l, q_r, q_u, and q_d are the left, right, 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 $P^{-1}(q) \in \mathbb{R}^n$, which is highly desirable. For instance, users interactively picking points q in such areas will get 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 $\mathbb{R}^n$. Applications such as interpolation or data synthesis then become hard to support [vWvO03, ABM*15, REHT19].

Datasets: We applied our method on synthetic data with a known topology as well as on real-world datasets: (1) *Blobs* is a synthetic dataset of 1000 points generated from a Gaussian distribution with 5 clusters and 100 additional randomly distributed noise samples ($|D| = 1100$, $n = 10$). (2) *Rings* is a synthetic dataset containing 3D ($n = 3$) points arranged in two interlacing rings having 100 and 80 samples respectively ($|D| = 180$). (3) *Iris* [Fis36] serves as a simple real-world application and represents 3 different flower species with $n = 4$ dimensions and $|D| = 150$ samples. (4) *Seismic* [SW10] shows a real-world scenario with more dimensions ($n = 24$) and includes $|D| = 646$ samples of seismic bumps in a coal mine.

4.1. Point Selection Strategies for Multilateration

We evaluated four different strategies for selecting the reference points x_i (see Sec. 3). First, we used the $n+1$ *furthest* points, *i.e.*, those having the largest distances d_i to the 2D point we want to invert. This approach could, in theory, work well, as MDS aims to optimize over global neighborhoods. Second, we used the $n+1$ *closest* points in 2D. The inverse projection should then behave more like the local iLAMP approach [ABM*15]. Yet, both of these strategies produce a few very high values for $MSE(x_i)$, most likely due to reference points being poorly chosen, *i.e.*, being collinear, coincident, or having distances not well preserved by MDS.

To tackle these issues, we used a *randomized* approach. For a given 2D point q to invert, we pick $n+1$ samples from D and compute the resulting $p = P^{-1}(q)$ using these samples as reference points (see Sec. 3). We repeat this s times, and finally set p to the medoid of all the generated values. Simply put, this strategy acts as a low-pass filter that limits the potentially undesirable effects due to a poor (random) selection of points from D.

Further, we tested a *stratified random* approach. 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.

We evaluate both randomized approaches using the average MSE (Eqn. 1). Figure 2 shows the average MSE for different numbers of selection steps s. We see that increasing s improves the accuracy of inversely projecting points from D. However, beyond a certain number of trials, the error more or less stabilizes. As our randomized approach (without clustering) outperforms the clustering one in terms of average MSE, we use this point selection strategy for the remaining evaluation with $s = 500$ selection steps.

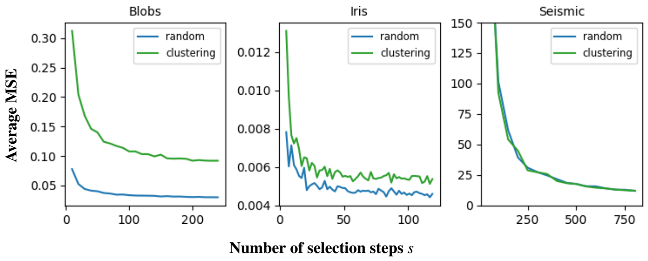

Figure 2: *Average MSE across datasets against number of selection steps for randomized and clustering-based selection strategies.*

4.2. Comparison to Existing Inverse Projection Techniques

We compare our approach to the inverse projection methods iLAMP [dSABI*12], iNN [ERH*19, EAS*21], and RBF [ABM*15], as these techniques are also applicable to MDS projections. For each method, we use the same parameters as proposed in the respective paper. We left SSNP [EHT21] and autoencoders [HS06] out from the comparison since these compute P and P^{-1} *jointly* (see Sec. 2).

As such, we cannot use these to invert a MDS projection or any other user-chosen projection for that matter.

We split the data into training and testing to assess the MSE of unseen test samples. Yet, our approach being a *lazy learner* has no training phase. Hence, to evaluate our P^{-1}, we restrict the multilateration to only use training samples as reference points.

	Ours	**iLAMP**	**iNN**	**RBF**
Blobs	0.0277	0.0057	0.0053	0.0059
Rings	0.0123	0.0012	0.0012	0.0008
Iris	0.0035	0.0034	0.0038	0.0036
Seismic	15.4107	0.0090	0.1601	0.0086

Table 1: *Average MSE on test data.*

Table 1 shows the average MSE of the test sample for each inverse projection technique when utilizing 80% of the data for training and the remaining 20% for testing. While achieving a similar average MSE for the *Iris* data, our method encounters limitations when MDS induces significant information loss. For the *Seismic* dataset, our average MSE on the test data is much higher than for the other inverse projection techniques tested. Yet, we see a notable difference in the MSE visualization in Figure 3 (*Blobs* dataset). 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. Separately, for our method, we see that areas in the middle of the projection scatterplot (being surrounded by many data points) have a lower MSE than points lying at the projection borders.

Gradient map visualizations match the above findings (see Figure 4). Despite having comparably high gradient ranges, these maps are quite smooth for our method, showing uniformly low values in the projection center. iLAMP yields high gradients visible as reticulated lines in gaps between projected point clusters and in areas with no projected samples. Our method shows low gradient values even in areas between clusters, see *e.g.* the *Iris* dataset. Both our method and iLAMP show uniformly low gradient values near and around projected points. Conversely, RBF's gradient maps show high values between projected samples and relatively low values in regions with no projected samples. 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 space. When the dataset dimensionality and complexity increases (see *Seismic*), our method reaches its limits, yielding high gradients especially in the corners of the map. Yet, the transition pattern is still relatively smooth in comparison to *all* other inverse projections. As Secs. 2 and 3 mention, low and constant gradients are essential for the practical *usage* of an inverse projection method.

5. Discussion and Future Work

Our approach, like all inverse projections P^{-1}, depends on the quality of the direct projection P. Our evaluation of inversely projecting known samples shows that (Sec. 4). This is unavoidable for *any* inverse projection since all such functions *must* be constructed from a given direct P. Neural network (NN) based techniques can accurately inversely project known samples, but are prone to overfit. Our approach, being purely based on the assumptions made by MDS, avoids this issue. 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. For us, $N = n+1$ for an n-dimensional dataset. However, our method does not require a training step, such as iNN [ERH*19], SSNP [EHT21], or autoencoders [HS06]. Thus, our computing effort is only in the inference step. Our technique can be used with progressive visualizations [Fek15] since our randomized strategy improves in accuracy with each iteration – a feature that, to our knowledge, none of the existing inverse projection techniques has. Finally, 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 MDS projection, and (4) the curse of dimensionality [VF05].

Future Work: While inverse projections can enrich projection scatterplots [LA11, NA19, DMKEA23], 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, reference point selection can be optimized, *e.g.*, by selecting points based on their relative angle or filtering out poorly-projected samples (following [REHT19]). Inverting really high-dimensional data will require many samples to avoid problems stemming from sparsity. While we restrict our approach to MDS with the Euclidean distance, future work can explore other projections and distance measures. There is a further need to evaluate the limitations of our approach, including its runtime. With regard to other inverse projection approaches, we found that objective criteria for the parameterization are missing or rather vague. For example, NN-based approaches require an NN architecture tailored to the dataset and are difficult to interpret. Our method only requires the number of selection steps s as a parameter.

6. Conclusion

We proposed a new method for inverting MDS-class projections. Our approach is purely based on Euclidean geometry, which avoids the need for more complex machine learning and deep-learning based approaches, which are hard to fine-tune, understand, interpret, and predict. Our method is simple to implement, efficient to compute, and can be applied to any generic high-dimensional dataset. We evaluate our method by qualitatively and quantitatively comparing it to three key existing approaches for inverse projections. Quantitatively, our approach only yields similar errors in inversly projecting known points if the distances are well preserved by MDS. However, 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. Still, our approach reaches limitations when the dimensionality or complexity of the underlying data is high.

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).

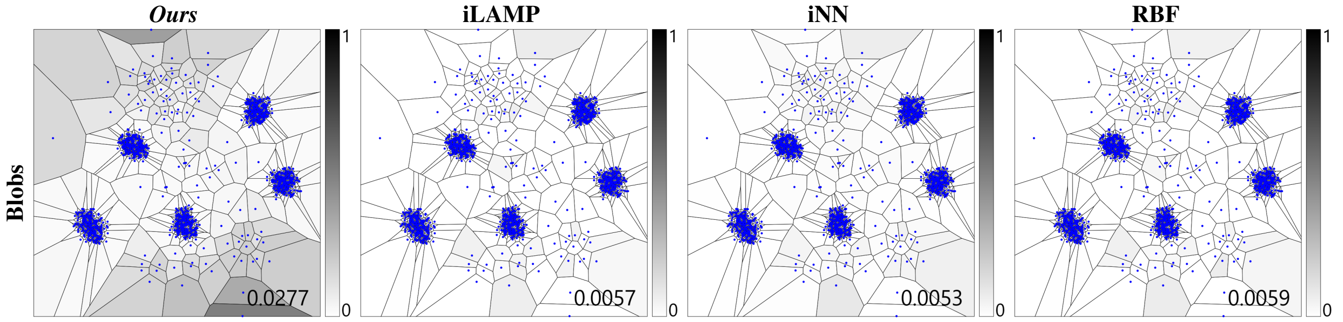

Figure 3: *Comparison of inverse projections showing the MSE for known points encoded in Voronoi cells associated with each projected sample of the Blobs dataset. The number at the bottom right corner indicates the average MSE for the test samples (20% of the dataset).*

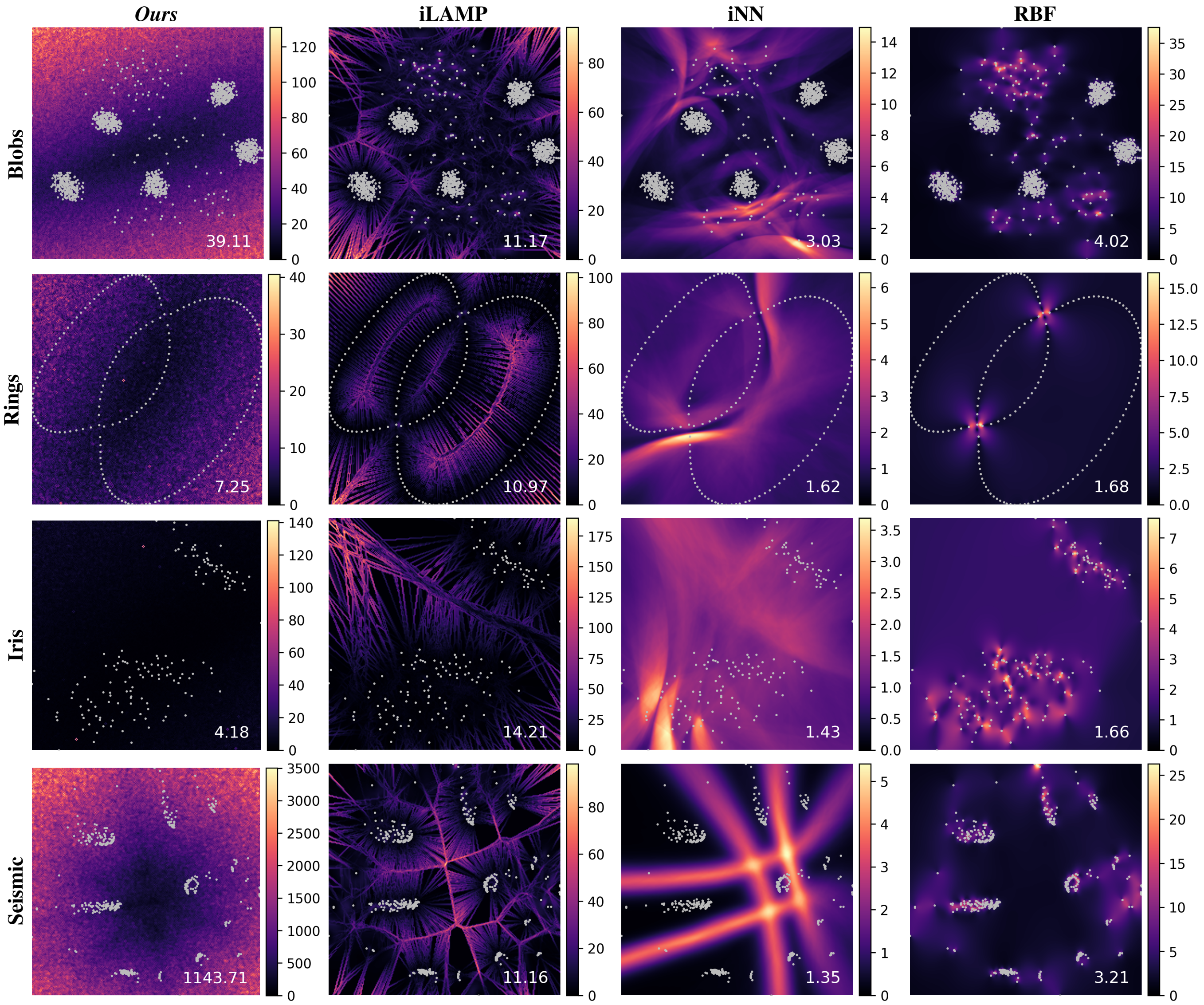

Figure 4: *Gradient maps of inverse projections for four datasets. Darker colors indicate a low rate of change, and lighter areas show a high rate of change. The number at the bottom right shows the average gradient.*

Proceedings published by Eurographics - The European Association for Computer Graphics.

References

[ABM*15] Amorim E., Brazil E. V., Mena-Chalco J., Velho L., Nonato L. G., Samavati F., Sousa M. C.: Facing the high-dimensions: Inverse projection with radial basis functions. *Comput. Graph. 48* (2015), 35–47. doi:10.1016/J.CAG.2015.02.009. 1, 2, 3

[AEC*22] Appleby G., Espadoto M., Chen R., Goree S., Telea A. C., Anderson E. W., Chang R.: HyperNP: Interactive visual exploration of multidimensional projection hyperparameters. *Comput. Graph. Forum 41*, 3 (2022), 169–181. doi:10.1111/CGF.14531. 2

[Aup07] Aupetit M.: Visualizing distortions and recovering topology in continuous projection techniques. *Neurocomputing 70*, 7-9 (2007), 1304–1330. doi:10.1016/J.NEUCOM.2006.11.018. 3

[CG15] Cunningham J. P., Ghahramani Z.: Linear dimensionality reduction: survey, insights, and generalizations. *J. Mach. Learn. Res. 16* (2015), 2859–2900. doi:10.5555/2789272.2912091. 1

[dLM09] de Leeuw J., Mair P.: Multidimensional Scaling Using Majorization: SMACOF in R. *J. Stat. Softw. 31*, 3 (2009), 1–30. doi:10.18637/jss.v031.i03. 2

[DMKEA23] Dennig F. L., Miller M., Keim D. A., El-Assady M.: FS/DS: A theoretical framework for the dual analysis of feature space and data space. *IEEE Trans. Vis. Comput. Graph.* (2023). Early Access. doi:10.1109/TVCG.2023.3288356. 4

[dSABI*12] dos Santos Amorim E. P., Brazil E. V., II J. D., Joia P., Nonato L. G., Sousa M. C.: iLAMP: Exploring high-dimensional spacing through backward multidimensional projection. In *7th IEEE Conf. Vis. Anal. Sci. Technol.* (2012), IEEE Computer Society, pp. 53–62. doi:10.1109/VAST.2012.6400489. 2, 3

[EAS*21] Espadoto M., Appleby G., Suh A., Cashman D., Li M., Scheidegger C., Anderson E. W., Chang R., Telea A. C.: UnProjection: Leveraging inverse-projections for visual analytics of high-dimensional data. *IEEE Trans. Vis. Comput. Graph. 29*, 2 (2021), 1559–1572. doi:10.1109/TVCG.2021.3125576. 1, 2, 3

[EHT20] Espadoto M., Hirata N. S. T., Telea A. C.: Deep learning multidimensional projections. *Inf. Vis. 19*, 3 (2020), 247–269. doi:10.1177/1473871620909485. 2

[EHT21] Espadoto M., Hirata N. S. T., Telea A. C.: Self-supervised dimensionality reduction with neural networks and pseudo-labeling. In *16th Int. Joint Conf. Comput. Vision Imaging Comput. Graph. Theor. Appl.* (2021), SCITEPRESS, pp. 27–37. doi:10.5220/0010184800270037. 2, 3, 4

[EMK*19] Espadoto M., Martins R. M., Kerren A., Hirata N. S. T., Telea A. C.: Toward a quantitative survey of dimension reduction techniques. *IEEE Trans. Vis. Comput. Graph. 27*, 3 (2019), 2153–2173. doi:10.1109/TVCG.2019.2944182. 1

[ERH*19] Espadoto M., Rodrigues F. C. M., Hirata N. S. T., Jr. R. H., Telea A. C.: Deep learning inverse multidimensional projections. In *10th Int. EuroVis Works. Vis. Anal.* (2019), Eurographics Association, pp. 13–17. doi:10.2312/EUROVA.20191118. 2, 3, 4

[Fek15] Fekete J.-D.: ProgressiVis: a Toolkit for Steerable Progressive Analytics and Visualization. In *1st Works. Data Syst. Interact. Anal.* (Chicago, United States, 10 2015). 4

[Fis36] Fisher R. A.: The use of multiple measurements in taxonomic problems. *Annals of Eugenics 7*, 2 (1936), 179–188. doi:10.1111/j.1469-1809.1936.tb02137.x. 3

[HFA17] Heulot N., Fekete J., Aupetit M.: Visualizing dimensionality reduction artifacts: An evaluation. *CoRR abs/1705.05283* (2017). arXiv:1705.05283. 1

[HS06] Hinton G. E., Salakhutdinov R. R.: Reducing the dimensionality of data with neural networks. *Science 313*, 5786 (2006), 504–507. doi:10.1126/science.1127647. 2, 3, 4

[JCC*11] Joia P., Coimbra D. B., Cuminato J. A., Paulovich F. V., Nonato L. G.: Local affine multidimensional projection. *IEEE Trans. Vis. Comput. Graph. 17*, 12 (2011), 2563–2571. doi:10.1109/TVCG.2011.220. 2

[Kru64] Kruskal J. B.: Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika 29*, 1 (1964), 1–27. doi:10.1007/BF02289565. 1

[LA11] Lespinats S., Aupetit M.: CheckViz: Sanity check and topological clues for linear and non-linear mappings. *Comput. Graph. Forum 30*, 1 (2011), 113–125. doi:10.1111/J.1467-8659.2010.01835.X. 3, 4

[MHM18] McInnes L., Healy J., Melville J.: UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *CoRR abs/1802.03426* (2018). arXiv:1802.03426, doi:10.48550/arXiv.1802.03426. 2

[NA19] Nonato L. G., Aupetit M.: Multidimensional projection for visual analytics: Linking techniques with distortions, tasks, and layout enrichment. *IEEE Trans. Vis. Comput. Graph. 25*, 8 (2019), 2650–2673. doi:10.1109/TVCG.2018.2846735. 1, 4

[OEJT22] Oliveira A. A. M., Espadoto M., Jr. R. H., Telea A. C.: SDBM: Supervised Decision Boundary Maps for Machine Learning Classifiers. In *17th Int. Joint Conf. Comput. Vision Imaging Comput. Graph. Theor. Appl.* (2022), SCITEPRESS, pp. 77–87. doi:10.5220/0010896200003124. 1, 2

[REHT19] Rodrigues F. C. M., Espadoto M., Hirata R., Telea A. C.: Constructing and visualizing high-quality classifier decision boundary maps. *Information 10*, 9 (2019), 280. doi:10.3390/INFO10090280. 1, 2, 3, 4

[SHH20] Schulz A., Hinder F., Hammer B.: DeepView: Visualizing classification boundaries of deep neural networks as scatter plots using discriminative dimensionality reduction. In *29th Int. Joint Conf. Artif. Intell.* (2020), ijcai.org, pp. 2305–2311. doi:10.24963/IJCAI.2020/319. 1, 2

[SVP14] Sorzano C. O. S., Vargas J., Pascual-Montano A. D.: A survey of dimensionality reduction techniques. *CoRR abs/1403.2877* (2014). arXiv:1403.2877. 1

[SW10] 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. 55*, 1 (2010), 91–114. 3

[vdMPvdH09] van der Maaten L., Postma E., van den Herik J.: *Dimensionality Reduction: A Comparative Review*. Tech. Rep. TiCC TR 2009-005, Tilburg University, Netherlands, 10 2009. 1

[VF05] Verleysen M., François D.: The curse of dimensionality in data mining and time series prediction. In *Comput. Intell. Bioinspir. Syst.* (2005), vol. 3512 of *Lecture Notes in Computer Science*, Springer, pp. 758–770. doi:10.1007/11494669_93. 4

[vWvO03] van Wijk J. J., van Overveld C. W. A. M.: Preset based interaction with high dimensional parameter spaces. In *Data Visualization: The State of the Art* (2003), Kluwer, pp. 391–406. 1, 2, 3

[WL24] Widdison E., Long D. G.: A review of linear multilateration techniques and applications. *IEEE Access 12* (2024), 26251–26266. doi:10.1109/ACCESS.2024.3361835. 2

[WMT23] Wang Y., Machado A., Telea A. C.: Quantitative and qualitative comparison of decision-map techniques for explaining classification models. *Algorithms 16*, 9 (2023), 438. doi:10.3390/A16090438. 1, 2

[WT24] Wang Y., Telea. A.: Fundamental limitations of inverse projections and decision maps. In *19th Int. Joint Conf. Comput. Vis. Imaging Comput. Graph. Theory and Applications* (2024), SciTePress, pp. 571–582. 1, 2

[Yin07] Yin H.: Nonlinear dimensionality reduction and data visualization: A review. *Int. J. Autom. Comput. 4*, 3 (2007), 294–303. doi:10.1007/S11633-007-0294-Y. 1