Towards a Quantitative Survey of Dimension Reduction Techniques

Mateus Espadoto, Rafael M. Martins, Andreas Kerren, Nina S. T. Hirata, and Alexandru C. Telea

Abstract—Dimensionality reduction methods, also known as projections, are frequently used in multidimensional data exploration in machine learning, data science, and information visualization. Tens of such techniques have been proposed, aiming to address a wide set of requirements, such as ability to show the high-dimensional data structure, distance or neighborhood preservation, computational scalability, stability to data noise and/or outliers, and practical ease of use. However, it is far from clear for practitioners how to choose the best technique for a given use context. We present a survey of a wide body of projection techniques that helps answering this question. For this, we characterize the input data space, projection techniques, and the quality of projections, by several quantitative metrics. We sample these three spaces according to these metrics, aiming at good coverage with bounded effort. We describe our measurements and outline observed dependencies of the measured variables. Based on these results, we draw several conclusions that help comparing projection techniques, explain their results for different types of data, and ultimately help practitioners when choosing a projection for a given context. Our methodology, datasets, projection implementations, metrics, visualizations, and results are publicly open, so interested stakeholders can examine and/or extend this benchmark.

Index Terms—Dimensionality reduction, quality metrics, benchmarking, quantitative analysis, design space

1 Introduction

Exploring high-dimensional data is central to many application domains such as statistics, data science, machine learning, and information visualization. The main difficulty encountered in this task is the large size of such datasets, both in the number of observations (also called samples) and measurements recorded per observation (also called dimensions, features, variables, or attributes). As such, high-dimensional visualization has become an important subfield of Information Visualization (InfoVis) [1], [2], [3], [4].

Several techniques exist for high-dimensional data visualization, including glyphs [5], parallel coordinate plots [6], table lenses [7], [8], scatterplot matrices [9], dimensionality reduction methods [10], and multiple views linking the above visualization types [11]. In this family, *dimensionality reduction* (DR) methods, also called projections, have a particular place: compared to other techniques, they scale much better in terms of both the number of samples and the number of dimensions they can show on a given screen space area. As such, projections have become the tool of choice for exploring data which has a high number of dimensions (tens up to hundreds) and/or in applications where the individual identity of dimensions is less important, as in *e.g.* machine learning applications. In the last decade, many projection techniques have been proposed [10], [12], [13], of which t-SNE [14] is arguably one of the best known and most adopted by applications.

This explosion of the number and variety of projection techniques and their widespread use in many applications makes it hard for end users to understand how to *choose* a good technique for a given use context. Several functional and non-functional requirements must be considered, such as the ability of the projection to preserve certain patterns (*e.g.*, neighbors, distances, or clusters); doing this for a given number of dimensions (which can be low or very high); computational scalability in both observation and dimension counts; robustness to small changes in both data and algorithm parameters, *i.e.*, yielding similar results for small changes of these inputs; ease of use in terms of number and complexity of settings asked from the end user; and available implementations. Current literature addresses such questions by comparative studies (*e.g.*, in papers that propose new projection techniques), best-practice studies, or survey papers. Yet, such approaches have limitations: Technique papers typically cover only a few techniques; survey papers consider tens of techniques, but typically focus on high-level and/or more theoretical aspects, and less on benchmarking many projection techniques on combinations of datasets, technique parameter settings, and evaluating quality metrics. Best-practice studies fall somewhere in the middle.

This survey aims to address the above-mentioned limitations, as follows (see also Fig. 1). First, we overview related surveys in evaluation and comparison of DR methods (Sec. 3). Based on these, we propose taxonomies covering the types of multidimensional datasets, projection techniques, and quality metrics used to assess these. This way, we explicitly show which parts of the data, projection, and quality spaces we next cover, and how. We model these taxonomies based on a number of so-called *traits* of datasets, techniques, and metrics respectively (we use the term traits to avoid confusion with dimensions). Next, we sample these spaces by 18 datasets, 44 techniques, and 7 quality metrics respectively, to create a projection assessment benchmark. The respective taxonomies, their traits, and how these are sampled to yield our benchmark are discussed in Secs. 4, 5, and 6 respectively. We run this benchmark, using an optimization strategy to find the best projection-technique parameter values for the considered quality metrics (Sec. 7). Finally, we present and discuss the obtained measurements (Sec. 8). We outline several observations on the correlations

 DOI: 10.1109/TVCG.2019.2944182

between dataset types, projection techniques, and quality aspects. Next, we select a few special cases (points of interest in our data, projections, and quality space) and examine these in more detail (Sec. 8.4). Section 9 discusses the main findings and limitations of our survey. We conclude by outlining directions of future work (Sec. 10).

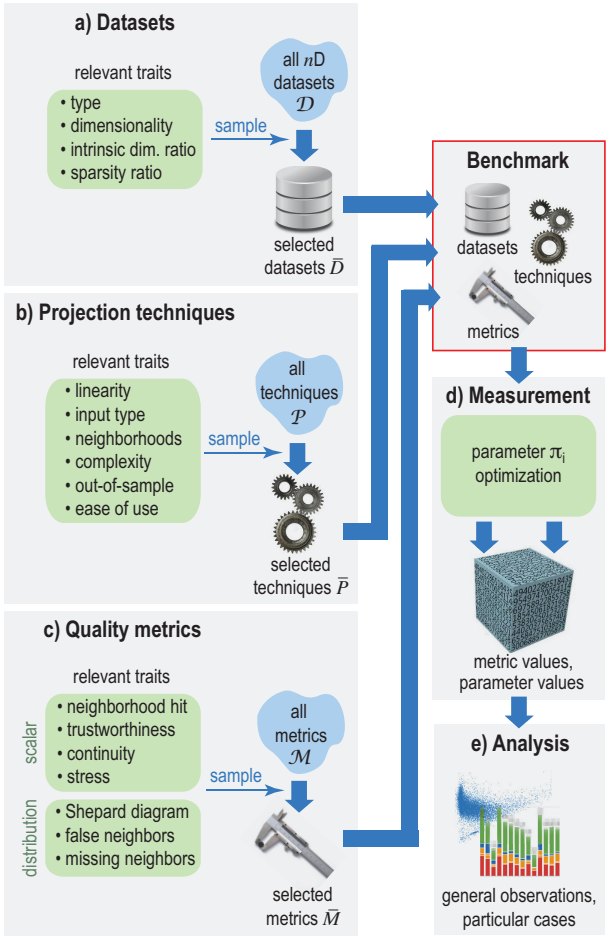

Figure 1. Workflow of survey outlining its main stages: Construction of benchmark of datasets (a), projection techniques (b), and quality metrics (c) based on taxonomies of these spaces; measurement of metric values using parameter optimization (d); and analysis of results (e). See Sec. 1.

2 Preliminaries

To discuss our (and related) work, we introduce a few notations. Let $\mathbf{x} = (x^1, \ldots, x^n)$, $x^i \in \mathbb{R}, 1 \leq i \leq n$ be a n-dimensional (nD) real-valued observation or sample, and let $D = \{\mathbf{x}_i\}$, $1 \leq i \leq N$ be a nD dataset of N samples. Let $\mathbf{x}^j = (x_1^j, \ldots, x_N^j)$, $1 \leq j \leq n$ be the j^{th} feature vector of D. Thus, D can be seen as a table with N rows (samples) and n columns (features or dimensions). A projection technique, or algorithm, is then a function

$$P : \mathbb{R}^n \rightarrow \mathbb{R}^q, \tag{1}$$

where $q \ll n$, and typically $q = 2$. P can also have p so-called free parameters, or hyperparameters, π_i, $1 \leq i \leq p$, which can be tuned by the end user to obtain different trade-offs of P. The projection $P(\mathbf{x})$ of a sample $\mathbf{x} \in D$ is a 2D point. Projecting an entire dataset D yields a 2D scatterplot, denoted as $P(D)$. We denote by cursive letters the power set (set of all sets) of a given type, *e.g.*, $\mathcal{D}$ is the set of all nD datasets D, and $\mathcal{P}$ is the set of all projection techniques P. Table 1 lists the projection techniques considered in this survey as well as the abbreviations we use for them.

To capture the quality of a projection technique P, let

$$M : \{(D \in \mathcal{D}, P(D))\} \rightarrow \mathbb{R}^k \tag{2}$$

be a metric that assigns to the pair formed by dataset D and its projection $P(D)$ a scalar ($k = 1$) or vector ($k > 1$) value. Let $\mathcal{M}$ be the set of all such metrics. Different metrics M capture different desirable aspects of a projection P. The key one, that all techniques consider, is preserving *similarity* of points when projecting from nD to qD. This is usually defined as Euclidean, geodesic, Procrustes, or cosine distance, or the probabilities of a point to have the same neighbors in $\mathbb{R}^n$ and $\mathbb{R}^q$ [14], [15]. Besides similarity, other quality aspects include computational scalability, ease of use (*vs* parameter setting), and robustness *vs* small input-data changes or hyperparameter changes. We discuss quality metrics in detail in Sec. 6.

Formally, to compare several projection techniques, we need to understand the distribution of *all* values of $\mathcal{M}$ over *all* values of $\mathcal{D}$ and $\mathcal{P}$, *i.e.*, how all quality metrics M vary over all combinations of datasets and projection techniques. Fully computing this distribution is practically impossible, since the spaces $\mathcal{P}$ and $\mathcal{M}$ have a very high cardinality, while $\mathcal{D}$ is infinite. All projection evaluation papers handle this by *sampling* $\mathcal{D}$, and $\mathcal{P}$, and $\mathcal{M}$ to select a small subset of datasets $\overline{D} \subset \mathcal{D}$, techniques $\overline{P} \subset \mathcal{P}$, and metrics $\overline{M} \subset \mathcal{M}$ to evaluate over. We call such a subset $B = \overline{D} \times \overline{P} \times \overline{M}$ a *benchmark*. An *evaluation* of a benchmark is thus the multidimensional set of values

$$E = \{M(D, P(D)) | (D, P, M) \in B\}. \tag{3}$$

We next discuss how existing surveys design evaluations E, *i.e.*, which decisions they take to sample the continuous spaces $\mathcal{D}$, $\mathcal{P}$, and $\mathcal{M}$ to evaluate B. We next propose ways to extend this state of the art in Secs. 4-6.

3 Related Work

In this section, we discuss related work on comparisons and evaluations of DR techniques. We do not detail here all projection techniques and quality metrics (and related papers) — this is done in context in Secs. 5 and 6, respectively. Several surveys that compare DR techniques for visualization (and sometimes beyond) have been published. We discuss these in chronological order. Since DR is at the crossroads of infovis and ML, we group surveys accordingly (Tab. 1).

3.1 Surveys from Machine Learning

Projection techniques are known and used since decades in ML [16], [17]. Fodor [18] presents the earliest survey on projection techniques that we are aware of, which includes what are now considered modern methods, *i.e.*, nonlinear methods, vector quantization, and neural networks. This survey discusses 12 projection methods, including linear

(PCA, ICA, FA) and some nonlinear (SOM, VQ, and NN) methods. While the survey (briefly) outlines the techniques underlying these methods, no actual side-by-side evaluation or quality metrics are discussed.

Yin [19] proposes a survey for nonlinear DR, focusing on visualization, covering seven projection methods. It also discusses variants of the stress metric for measuring projection quality. Yet, only two simple datasets are evaluated, using only four of the seven DR methods.

Maaten *et al.* [13] present, to our knowledge, the first systematic theoretical and *practical* comparison of PCA (linear) and other 13 (nonlinear) DR techniques. The theoretical side discusses the number of parameters, computational and memory complexities, and out-of-sample ability (whether a projection can handle new samples based on existing projected ones). Practical comparison includes measuring three scalar metrics (trustworthiness, continuity, and preservation of closest neighbors [20]) on 5 artificial and 5 real-world datasets. Optimal parameters were found using grid search. However, how the three quality metrics listed above were merged into a single quality (cost) function to optimize by grid search is not detailed. Also, the survey does not cover many well-known projection techniques (Tab. 1).

Bunte *et al.* [15] propose a theoretical framework to unify nine existing projection techniques. These are compared in terms of how *similarity* between points in nD and 2D is defined; which error metric the projection P minimizes; and which additional constraints the methods have. The techniques are evaluated on three datasets ranging between a few thousand and 20K points having between 16 and 36 dimensions. However, many existing projection techniques are not covered by this survey.

Sorzano *et al.* [12] present one of the most complete surveys from the viewpoint of number of discussed DR techniques – around 30, including variants of some main techniques. Yet, this survey has mainly a theoretical focus. Heuristics and cost functions underlying the DR techniques are discussed, but practical evaluation involves only a *single* image showing how LLE, HLLE, and ISO perform on a small synthetic dataset of 1K points in 3 dimensions. Measurements of quality metrics are not given.

Gisbrecht *et al.* [21] evaluate the suitability for data visualization of 10 DR techniques on 3 synthetic datasets of 1K three-dimensional points each. Projection quality is defined as a single scalar value using the rank-based criteria in [22]. Compared to earlier surveys, this one includes assessing computational scalability; and focuses on "popular" projection techniques, as these are more likely to be used in practice, so understanding how they perform is of increased added value. Yet, the evaluation confidence is limited by the very small number of tested datasets.

Cunningham *et al.* [23] present an excellent survey of *linear* DR techniques. This work is very similar in goals and structure to Sorzano *et al.* [12], *i.e.*, it aims to compare 15 DR techniques and a few sub-variants thereof from theoretical and mathematical viewpoints. No practical evaluations of quality metrics of existing techniques on datasets are given. Also, nonlinear projections are not considered.

Finally, Xie *et al.* [24] survey DR techniques based on the Random Projection (RP) method [25]. Such methods are arguably better at keeping data structure and/or reducing computational effort when dealing with a high dimension count. About 25 RP variants and a few sub-variants are discussed from a general perspective. This survey aims to provide a "reading map" for the RP literature. Yet, no side-by-side evaluation of existing methods on a benchmark, using specific quality metrics, is given.

3.2 Surveys from InfoVis

The infovis literature is rich in papers that evaluate projections. We next focus on key papers that share the aim of our work (comparing projections from a quantitative perspective). Additional papers related to assessing projection quality are discussed later in context.

Buja *et al.* [11] present one of the earliest surveys on projection usage to visualize multidimensional data. They propose a task-based taxonomy of interaction techniques for analyzing high-dimensional data. Projections, implemented in the XGobi tool [26], are just one of the considered techniques, in a linked-view set-up, to validate the proposed interaction taxonomy. In a related work, Hoffman *et al.* [1] compare 15 visualization techniques on two small datasets (hundreds of observations, 4 to 6 dimensions). Among these, three are projection techniques (Sammon's mapping (SAM), MDS, and Kohonen's self-organizing maps (SOM)). This survey does not contain any quantitative comparison of the discussed techniques.

Engel *et al.* [27] propose "an introduction to dimension reduction from a visualization point of view". They propose a taxonomy that compares nine DR methods from the viewpoint of their online behavior (out-of-sample ability) and computational complexity. Yet, no actual evaluation of quality metrics on datasets is given. The survey strongly makes the point that optimal parameter setting is an important but not well explored aspect that influences the quality, and finally usability, of projection techniques. We address this aspect in our work (Sec. 7).

Kehrer *et al.* [3] present a survey of methods for visualization of so-called "multi-faceted" scientific data. The survey overviews the context in which projection techniques are used in the (much) broader scope of visual analysis of multidimensional, multi-source, and multi-type datasets. Given this broad scope, details concerning the evaluation of projection techniques are not given. Liu *et al.* [2] present a related survey focused more specifically on visualizing high-dimensional data. They propose a 14-element taxonomy of techniques for high-dimensional visualization, of which dimensionality reduction is one. They also briefly introduce several projection quality metrics, such as global stress, local stress [28], ranking discrepancy [22], [29], and the projection precision score [30]. While seven concrete projection techniques are named, evaluating these and/or the aforementioned metrics is not covered.

Close to our goals, Nonato and Aupetit [10] survey the use of projections in visual analytics (VA) tasks. Their work, which is arguably together with [13] one of the most extensive and detailed surveys in DR literature, propose a taxonomy where 28 projection techniques are classified along their input data types, linearity, flexibility for supervision (label data), handling multilevel structures, locality, steerability, stability, and ability to handle out-of-code (large) data. They also discuss 14 projection quality

metrics. Yet, actual *measurements* of how techniques perform, with respect to metrics, on a representative benchmark of datasets, is not in the scope of this survey. Our work aims to fill in this gap. On the other side, [10] covers several other directions, most notably the relation between DR techniques, caused distortions, VA tasks affected by distortions, and visual enrichments that can alleviate such problems. All these aspects are not in the scope of our work.

3.3 Summary of Current Surveys

Given related work in ML and infovis, we can state that current surveys do not cover several aspects of our goal (Sec. 1). Table 2 overviews the number of evaluated DR techniques, number of datasets used for evaluation, and number of evaluated metrics. We see that some surveys include many techniques, but discuss these from a technical/mathematical viewpoint rather than a practical one [23], or have a more educational, rather than evaluational, purpose [12]. Visualization surveys cover much more than projections and thus cannot include in-depth evaluations [2], [27]. Van der Maaten *et al.* [13] is the closest survey to our aims. Following Fig. 1, we next extend this survey's workflow, by considering more DR techniques (44 in total), more datasets (18 in total), an explicit choice of datasets to cover better the variability present in high-dimensional data spaces $\mathcal{D}$, more quality metrics (5 scalar metrics and 2 visual ones), and a study of how quality depends on the projection algorithms' parameters.

4 DATASETS

Our first step (Fig. 1a) is to sample the space $\mathcal{D}$ of existing multidimensional *datasets* to get a representative collection $\overline{D}$ on which we evaluate projection techniques. For this, we propose a set of *traits* to characterize datasets (Sec. 4.1). Binning these traits enables us to construct $\overline{D}$ (Sec. 4.2).

4.1 Dataset Traits

The dataset traits we propose to describe $\mathcal{D}$ aim to capture aspects outlined as important for the behavior of projection algorithms in earlier surveys [12], [13], [23]. We also choose traits that are easy to understand and measure by non-specialist end-users (the audience of our work), so they can easily use them when evaluating existing techniques *vs* their own datasets. We propose the following five traits, along with sampling strategies that create classes of elements in $\mathcal{D}$ along each trait:

Type τ_D**:** This trait has three categorical values, *tabular*, *image*, and *text*, in line with the most frequent dataset types for which projections are used [10]. We define three classes: *tables*, *images*, and *text*, one per value of τ_D.

Size N**:** Number of samples in a dataset. We define three classes: *small* ($N \leq 1000$); *medium* ($1000 < N \leq 3000$); and *large* ($N > 3000$). These values are in line with typical dataset sizes used in projection evaluation papers.

Dimensionality n**:** Number of dimensions of a dataset. We define three classes: *low* ($n < 100$); *medium* ($100 \leq n < 500$); *high* ($n \geq 500$). Typically, the lower the dimensionality n is, the easier is the job of a projection technique. While it can be argued that this difficulty is chiefly a function of the intrinsic dimensionality (discussed next), typical end users first get exposed to, and can easily evaluate, n; in contrast, evaluating the intrinsic dimensionality is more involved, as one can define it in different ways, and also this metric can take different values in different neighborhoods of the data. Hence, we include n as a separate trait.

Intrinsic dimensionality ratio ρ_n**:** The percentage of principal components (of the total n), computed by PCA, needed to explain 95% of the data variance. Higher ρ_n values (in $[0, 1]$) typically tell that a projection P has difficulties in mapping the data to $\mathbb{R}^q$. We define three classes: *low* ($\rho_n \leq 0.1$); *medium* ($0.1 < \rho_n \leq 0.5$); *high* ($0.5 < \rho_n \leq 1$).

Sparsity ratio γ_n**:** We define $\gamma_n = 1 - \frac{u}{nN}$, $\gamma_n \in [0, 1]$, where u is the number of non-zero data values, and nN is the total number of data values in a dataset (including zero). Datasets have widely different γ_n values: Text word vectors are very sparse; tabular data with a few variables are very dense. Typically, the sparser the data, the closer are the datapoints in high-dimensional space [31], [32], so a projection P has difficulties in separating clusters in $\mathbb{R}^q$. We define three classes: *dense* ($\gamma_n \leq 0.2$); *medium* ($0.2 < \gamma_n \leq 0.8$); and *sparse* ($0.8 < \gamma_n \leq 1$).

Other traits are envisageable, such as considering data with (or without) missing values. We do not consider this specific trait, as it is hard to decide how to sample the 'lack of values' of $\mathcal{D}$ in a good, exhaustive, manner. Similarly, other trait classes are possible, *e.g.*, transaction data, time-series data, or network data for the 'type' trait. We subsume these to the 'table' class, as using too many classes would increase the (already large) evaluation effort by several factors.

Defining the above traits and their classes (bin values) is, of course, not a theoretically ideal way to reflect the distribution of all datasets in the space $\mathcal{D}$. Ideally, we would know which are the independent generative axes (traits) of this space, and how all datasets in the real world distribute along these, and derive the trait-bins by following characteristics of these distributions. However, since this information is not known, nor, we argue, can be inferred (even with significant effort), we take a different path: We choose *traits* based on data characteristics which are known, from previous surveys and DR papers, to be relevant for the behavior of DR methods; and choose trait *classes* (bins) based on the characteristics of datasets that end users will arguably meet when applying DR in practice.

Sampling $\overline{D}$ from $\mathcal{D}$ along these five traits allows us to evaluate projection techniques P on different types of datasets, aiming to answer questions such as:

- How does P work for datasets of different *types*? Is the type of a dataset important when choosing P?
- How does P scale with the number of *samples* and/or *dimensions* of a dataset?
- How does P handle data with low/medium/high intrinsic *dimensionality*?
- How does P behave for *sparse vs* non-sparse data?

Table 1
Projection techniques discussed in surveys on dimensionality reduction from machine learning and Infovis.
The last column corresponds to this paper, and the last row shows total number of techniques discussed in each survey. See Sec. 2.

Projection Acronym	**Projection Full Name**	Fodor *et al.* [18]	Hoffman *et al.* [1]	Yin *et al.* [19]	Maaten *et al.* [13]	Bunte *et al.* [15]	Engel *et al.* [27]	Sorzano *et al.* [12]	Cunningham *et al.* [23]	Gisbrecht *et al.* [21]	Liu *et al.* [2]	Xie *et al.* [24]	Nonato *et al.* [10]	**Ours**
AE	Autoencoder				●									●
CCA	CCA (Canonical Correlations Analysis)								●					
CHL	Chalmers												●	
CLM	ClassiMap												●	
CuCA	CCA (Curvilinear Component Analysis)												●	
DM	Diffusion Maps				●									●
DML	Distance Metric Learning								●					
EM	Elastic Maps							●						
FA	Factor Analysis	●						●	●					●
FD	Force-Directed												●	
FMAP	FastMap												●	●
FS	Feature Selection											●		
GDA	Generalized Discriminant Analysis													●
GPLVM	Gaussian Process Latent Variable Model													●
GTM	Generative Topographic Mapping							●					●	
ICA	Independent Component Analysis	●						●	●					
F-ICA	FastICA													●
NL-ICA	Nonlinear ICA	●												
IDMAP	IDMAP													●
ISO	Isomap		●	●	●	●	●			●			●	●
L-ISO	Landmark Isomap													●
KECA	Kernel Entropy Component Analysis							●						
KLP	Kelp												●	
LAMP	LAMP												●	●
LDA	Linear Discriminant Analysis								●		●	●	●	
LE	Laplacian Eigenmaps				●	●				●	●		●	●
LLC	Locally Linear Coordination				●									●
LLE	Locally Linear Embedding		●	●	●	●	●			●	●		●	●
H-LLE	Hessian LLE				●									●
M-LLE	Modified LLE													●
LMNN	Large-Margin Nearest Neighbor Metric													●
LoCH	Local Convex Hull												●	
LPP	Locality Preserving Projection								●					●
LR	Linear Regression								●					
LSP	Least Square Projection												●	●
LTSA	Local Tangent Space Alignment				●								●	●
L-LTSA	Linear Local Tangent Space Alignment													●
MAF	Maximum Autocorrelation Factors								●					
MC	Manifold Charting				●					●				●
MCA	Multiple Correspondence Analysis												●	
MCML	Maximally Collapsing Metric Learning													●
MDS	Metric Multidimensional Scaling	●	●	●	●	●	●	●	●		●		●	●
L-MDS	Landmark MDS													●
MG-MDS	Multi-Grid MDS						●							
N-MDS	Nonmetric MDS (Kruskal)		●				●						●	●
ML	Manifold Learning						●							
MVU	Maximum Variance Unfolding				●	●				●			●	
FMVU	Fast MVU													
L-MVU	Landmark MVU													●
NeRV	Neighborhood Retrieval Visualizer					●								
t-NeRV	t-NeRV					●								
NMF	Nonnegative Matrix Factorization							●	●					●
NLM	Nonlinear Mapping												●	
NN	Neural Networks	●												
PBC	Projection By Clustering													●
PC	Principal Curves	●						●						
PCA	Principal Component Analysis	●	●		●		●	●	●	●	●	●	●	●
I-PCA	Incremental PCA							●						●
K-PCA-P	Kernel PCA (Polynomial)													●
K-PCA-R	Kernel PCA (RBF)		●		●		●	●		●				●
K-PCA-S	Kernel PCA (Sigmoid)													●
L-PCA	Localized PCA							●						
NL-PCA	Nonlinear PCA	●		●				●						
P-PCA	Probabilistic PCA								●					●
R-PCA	Robust PCA							●						
S-PCA	Sparse PCA							●						●
PLMP	Part-Linear Multidimensional Projection												●	
PLP	Piecewise Laplacian-based Projection						●						●	
PLSP	Piecewise Least Square Projection													●
PM	Principal Manifolds			●										
PP	Projection Pursuit	●												
RBF-MP	RBF Multidimensional Projection												●	
RP	Random Projections	●										●		
G-RP	Gaussian Random Projection													●
S-RP	Sparse Random Projection													●
SAM	Sammon Mapping				●									
R-SAM	Rapid Sammon (Pekalska)												●	●
SDR	Sufficient Dimensionality Reduction								●					
SFA	Slow Feature Analysis								●					
SMA	Smacof												●	
SNE	Stochastic Neighborhood Embedding					●							●	
T-SNE	t-Dist. Stochastic Neighborhood Embedding					●				●	●			●
SOM	Self-Organizing Maps	●		●				●					●	
ViSOM	ViSOM (Visualization-induced SOM)			●										
SPE	Stochastic Proximity Embedding													●
G-SVD	Generalized SVD							●						
T-SVD	Truncated SVD													●
TF	Tensor Factorization							●						
UMAP	Uniform Manifold Approximation and Proj.													●
VQ	Vector Quantization	●						●						
Total		12	6	7	14	9	9	19	14	8	6	4	28	44

Table 2
Summary of surveys on dimensionality reduction from both machine learning (ML) and InfoVis (IV), with the respective number of DR techniques (discussed), datasets (used in evaluation), and metrics (computed on the datasets). The last column corresponds to this paper. See Sec. 3.3.

	Fodor *et al.* [18]	Hoffman *et al.* [1]	Yin *et al.* [19]	Maaten *et al.* [13]	Bunte *et al.* [15]	Engel *et al.* [27]	Sorzano *et al.* [12]	Cunningham *et al.* [23]	Gisbrecht *et al.* [21]	Liu *et al.* [2]	Xie *et al.* [24]	Nonato *et al.* [10]	**Ours**
Number of techniques	12	6	7	14	9	9	19	14	8	6	4	28	**44**
Number of datasets	-	2	2	10	3	-	3	-	3	-	3	-	**18**
Number of metrics	-	-	-	3	3	-	-	-	1	-	1	-	**7**
Field of survey	ML	IV	ML	ML	ML	IV	ML	ML	ML, IV	IV	ML	IV	**IV**

4.2 Chosing Datasets

Sampling $\overline{D}$ from $\mathcal{D}$ along the trait bins introduced in Sec. 4.1 is challenging. Taking one sample per combination of intervals would yield $3^5 = 243$ different datasets, which would make the evaluation impractical, given that we next want to evaluate several tens of techniques per sample. Also, finding *real world* datasets for all these trait values is hard. Separately, we need datasets having *labeled* data, given that some quality metrics depend on this (Sec. 6). Hence, we chose to manually collect a smaller set of 18 datasets which cover well (though not fully) the aforementioned space of trait values. The datasets are introduced below. Table 3 lists their trait values.

Bank Marketing (bank) [33]: Direct marketing campaign data of a Portuguese bank used to predict whether a client will subscribe to a banking product or not;
CIFAR10 (cifar10) [34]: Standard Computer Vision research dataset consisting of images of animals and vehicles, used for training image classfiers;
CNAE-9 (cnae9) [35]: Free text descriptions of Brazilian companies in the National Classification of Economic Activities, split in 9 classes based on economic activity;
COIL20 (coil20) [36]: Columbia University Image Library, consisting of images of 20 types of common objects;
Epileptic Seizure Recognition (epileptic) [37]: Data from brain activity used to detect epileptic seizures;
Fashion-MNIST (fashion_mnist) [38]: Similar to MNIST, this dataset consists of images of 10 types of clothing;
Flickr Material Database (fmd) [39]: Images of common materials used for training material recognition systems;
HAR (har) [40]: Data from 30 subjects performing activities of daily living, used for human activity recognition;
Hate Speech (hatespeech) [41]: Tweets labeled according to the type of offensive language they contain, used for training hate speech detectors;
HIVA (hiva) [42] : Dataset used to predict which chemical compounds are active against HIV infection;
IMDB (imdb) [43]: Movie ratings data used for sentiment analysis;
ORL (orl) [44]: Face images from 40 different subjects;
SECOM (secom) [45]: Data from a semiconductor manufacturing process, used for training failure detectors;
Seismic Bumps (seismic) [46]: Data used to forecast seismic bumps in a coal mine;
Sentiment Labeled Sentences (sentiment) [47]: Text dataset created for sentiment analysis;
SMS Spam Collection (sms) [48]: Data from SMS labeled messages collected for mobile phone spam research, used for training SMS spam detectors;
Spambase (spambase) [49]: Data used to train email spam classifiers;
Street View House Numbers (svhn) [50]: Computer Vision dataset of images of digits 0 to 9 from Google Street View.

5 Projection Techniques

Just as we sampled the space of multidimensional datasets $\mathcal{D}$ (Sec. 4), we must now sample the space of projection techniques $\mathcal{P}$ (Fig. 1b). For this, we could use one of the projection taxonomies in the literature. Yet, this poses problems: There is so far no agreed 'universal' taxonomy. Different taxonomies serve different goals. For instance, Van der Maaten *et al.* [13] organize techniques on the type of *optimization* method they use; Cunningham *et al.* [23] follow a similar approach, but cover only *linear* techniques; Sorzano *et al.* [12] classify methods on *implementation* aspects (statistics-based, dictionary-based, and projection-based); Engel *et al.* [27] also classify methods on *implementation* aspects (projection-based, graph-based, and manifold learning). Finally, Nonato *et al.* [10] classify methods along eight traits (Sec. 3.2). We follow a similar approach, but use different traits, as explained next.

5.1 Projection Traits

We base our sampling $\overline{P}$ of the space of projection techniques $\mathcal{P}$ on eight traits that reflect what non-specialist users consider when choosing a technique, as follows.

Linearity: A projection is linear or nonlinear. Both types are well-covered in the literature and equally important in practice. Linear projections are easy to understand and use, but cannot capture well sample distributions spread on complex manifolds in nD. Nonlinear projections are better for such datasets, but are harder to control parameter-wise;

Input type: A projection P reads either a *distance* matrix $A = (d(\mathbf{x}_i, \mathbf{x}_j))$, $1 \leq i \leq N$, $1 \leq j \leq N$, where d is a dissimilarity function over $\mathcal{D}$, or the set $D = \{\mathbf{x}_i\}$ of high-dimensional *samples* themselves. When samples are available, one can always derive a distance matrix from them, but not conversely;

Neighborhood: A projection P claims to preserve local or global neighborhoods. Local-neighborhood methods try to preserve distances between a point and its (close) neighbors in D, which may yield better cluster separation, but lose the meaning of distances between clusters in the projected space [14]. Global methods try to preserve all-point-pair distances, which may result in more faithful projections of the high-dimensional space, but show cluster separation less well [16];

Ease of use: Number of free parameters (hyperparameters) that P exposes to the end user. More parameters give more flexibility, but finding optimal settings is harder;

Computational complexity: Algorithmic complexity of P, in big-O notation, as a function of N and n. Low-complexity methods are best for interactive visual exploration, but may have trouble in creating accurate results;

Out-of-sample: Ability to project new data based on earlier training. Useful when one wants to study dynamic datasets which add new samples over time [10], [51], [52];

Inverse transform: Ability to map low-dimensional $\mathbb{R}^q$ data to the original $\mathbb{R}^n$ space. Useful for explaining patterns in the projection [10], [53], [54], [55];

Determinism: Ability to reproduce its results regardless of random seed initialization. Useful when reproducible results are expected.

Table 3
Selected datasets $\overline{D}$ and their trait values. See Sec. 4.2.

Dataset	Type (τ_D)	Size (N)	Size class	Dimensionality (n)	Dimensionality class	Intrinsic dim. (ρ_n)	Intrinsic dim. class	Sparsity (γ_n)	Sparsity class
bank	*tables*	2059	*medium*	63	*low*	0.0317	*low*	0.6963	*medium*
cifar10	*images*	3250	*large*	1024	*high*	0.0706	*low*	0.0024	*dense*
cnae9	*text*	1080	*medium*	856	*high*	0.3201	*medium*	0.9922	*sparse*
coil20	*images*	1440	*medium*	400	*medium*	0.0105	*low*	0.3858	*medium*
epileptic	*tables*	5750	*large*	178	*medium*	0.2191	*medium*	0.0067	*dense*
fashion_mnist	*images*	3000	*medium*	784	*high*	0.2385	*medium*	0.5021	*medium*
fmd	*images*	997	*small*	1536	*high*	0.3073	*medium*	0.0095	*dense*
har	*tables*	735	*small*	561	*high*	0.1194	*medium*	0.0001	*dense*
hatespeech	*text*	3222	*large*	100	*medium*	0.6130	*high*	0.9993	*sparse*
hiva	*tables*	3076	*large*	1617	*high*	0.2498	*medium*	0.9091	*sparse*
imdb	*text*	3250	*large*	700	*high*	0.5790	*high*	0.9945	*sparse*
orl	*images*	400	*small*	396	*medium*	0.0006	*low*	0.9000	*sparse*
secom	*tables*	1567	*medium*	590	*high*	0.0102	*low*	0.2617	*medium*
seismic	*tables*	646	*small*	24	*low*	0.0417	*low*	0.5883	*medium*
sentiment	*text*	2748	*medium*	200	*medium*	0.8080	*high*	0.9936	*sparse*
sms	*text*	836	*small*	500	*medium*	0.7240	*high*	0.9947	*sparse*
spambase	*text*	4601	*large*	57	*low*	0.0351	*low*	0.7741	*medium*
svhn	*images*	733	*small*	1024	*high*	0.8734	*high*	0.0001	*dense*

5.2 Selected Projections

Following the above, we select a set $\overline{P}$ of 44 DR techniques which cover a wide set of the end-user-relevant trait values (Sec. 5.1). As when selecting datasets to create $\overline{D}$ (Sec. 4.2), our sample $\overline{P}$ cannot cover *all* possible DR techniques. To make $\overline{P}$ as relevant as possible, we selected DR techniques that are well known, often met in literature or practice, have a readily available implementation, and can be applied to generic multidimensional datasets (as opposed to handling very specific kinds of data). This way, we maximize the likelihood that $\overline{P}$ includes most techniques of interest that a typical user will consider and want to ask questions about.

We next describe the selected techniques. Citations indicate the specific variant of a technique we used. Table 4 summarizes their trait values, including the publicly available implementations we used in our evaluation. Except the number of free parameters of each algorithm, and the implementation we used, which are self-explaining, all other traits are described in [10]. We do not detail the theoretical or algorithmic aspects of these techniques, as these are covered in earlier surveys or original papers cited below, and since we aim to evaluate these techniques from an *end user* perspective rather than from a designer's or mathematician's one. We group these techniques along two attributes, linearity and type of neighborhood, each having two values, yielding four groups. This simple taxonomy helps non-specialist users to first select an appropriate group of techniques for their problem, after which they can refine selection based *e.g.* on quality metrics (Sec. 6).

Linear and Global: Techniques that use only linear transformations and consider all samples at a time. This group includes **PCA** [16] and its variations, Incremental PCA (**I-PCA**) [56], Probabilistic PCA (**P-PCA**) [57], and Sparse PCA (**S-PCA**) [58], all of which use orthogonal transformations to derive a set of uncorrelated variables. Factor Analysis (**FA**) [16] and Fast ICA (**F-ICA**) [59] are related to PCA, but aim at uncovering latent variables not captured by existing data dimensions. Nonnegative Matrix Factorization (**NMF**) [60] and Truncated SVD (**T-SVD**) [61] use matrix factorization to find representations in lower dimensions. Locality Preserving Projection (**LPP**) [62] is an algorithm based on linear projective maps.

Nonlinear and Local: Techniques that use nonlinear functions and seek to preserve the local neighborhood for each sample. This group contains manifold learning techniques such as Isomap (**ISO**) [63] and its faster variant Landmark Isomap (**L-ISO**) [64], both of which use geodesic distances to estimate neighborhoods; Locally Linear Embedding (**LLE**) [65] and its variants Hessian LLE (**H-LLE**) [66], Modified LLE (**M-LLE**) [67] and Local Tangent Space Alignment (**LTSA**) [68], Laplacian Eigenmaps (**LE**) [69], Diffusion Maps (**DM**) [70], Manifold Charting (**MC**) [71], and Local Linear Coordination (**LLC**) [72]. Other techniques in this group are Local Affine Multidimensional Projections (**LAMP**) [73], which uses orthogonal mapping theory to build accurate local transformations; Projection by Clustering (**PBC**) [74], a fast method that represents sample similarity by proximity; Interactive Document Maps (**IDMAP**) [75], which maps data by a fast projection, then refine the result using a force scheme; and Maximally Collapsing Metric Learning (**MCML**) [76], that use convex optimization to learn a quadratic Gaussian metric. Last but not least, we have t-Stochastic Neighborhood Embedding (**T-SNE**) [14], a method that aims to maximize the probability that similar samples are placed close to each other, and which is considered a gold-standard for 2D projection; and Uniform Manifold Approximation and Projection (**UMAP**) [52], which aims to find a $\mathbb{R}^q$ fuzzy topological structure closest possible to the $\mathbb{R}^n$ topological data structure. Compared to t-SNE, UMAP produces in general more clustered results, and is significantly faster.

Nonlinear and Global: Techniques that use nonlinear functions and consider all samples at a time. Techniques in this group are Metric Multidimensional Scaling (**MDS**) [17], Nonmetric Multidimensional Scaling (**N-MDS**) [77] and Landmark MDS (**L-MDS**) [78]. Kernel PCA (**K-PCA**) [79] and Gaussian Process Latent Variable Model (**GPLVM**) [80] are nonlinear extensions of PCA that use kernel methods and probabilistic models, respectively. Gaussian (**G-RP**) and Sparse Random Projections (**S-RP**) [25] project the original input space on randomly generated matrices. Maximum Variance Unfolding (MVU) [81] and its variations Fast MVU (F-MVU) [82] and Landmark MVU (**L-MVU**) [83] aim to unfold the data manifold by maximizing

Euclidean distances between points while preserving pairwise distances in a neighborhood graph. Generalized Discriminant Analysis (**GDA**) [84], also known as Kernel LDA, is a nonlinear generalization of LDA, a supervised linear DR technique [85]. Least Square Projection (**LSP**) [86] and its faster version Piecewise Least Square Projection (**PLSP**) [87] use least squares approximations. Other methods in this class are Rapid Sammon (**R-SAM**) [88] and Fastmap (**FMAP**) [89]. Autoencoders (**AE**) [90] use neural networks to generate low-dimensional data representations that can be used as projections. Stochastic Proximity Embedding (**SPE**) [91] aim to preserve similarities between a set of related points.

Linear and Local: Techniques that use only linear transformations to reduce dimensionality on separate small neighborhoods. This includes Large-Margin Nearest Neighbor Metric Learning (**LMNN**) [92], which learns a Mahalanobis distance metric by using semidefinite programming; and Linear LTSA (**L-LTSA**) [93], a variation of LTSA [68] that uses linear mappings.

Other techniques: Besides the above projection techniques, and technique traits, several others exist. A particular one is the use of *labeled* data when computing the projection [94], [95]. Using such information helps better separating classes present in the data, which, in turn, yields better values for several of the projection quality techniques discussed next in Sec. 6. We did not include these techniques in the survey as they would be hard to compare against techniques that do not use label information (which are in the majority).

6 QUALITY METRICS

The third and last component of our benchmark (Fig. 1c) covers the projection quality metrics used to assess the selected methods $\overline{P}$ on the selected datasets $\overline{D}$. Using metrics to gauge the quality of DR methods is an established field for which separate surveys exist [10], [22], [29], [96]. Since DR is essentially ill-posed, several such metrics must be *jointly* used to assess the quality of a DR technique [21]. We next describe several of these metrics, along with the reasons for choosing to include them in our benchmark (for s summary, see Tab. 5). Also, for completeness, we point to other metrics which we did not include, and outline the reasons for that.

Following Eqn. 2, we classify metrics based on their output dimensionality (k value).

6.1 Scalar metrics

The simplest and most used quality metrics yield a scalar value ($k = 1$, Eqn. 2) for a projection $P(D)$. We choose the first four from the following scalar metrics, given that they are well known, easily interpretable, and used in most DR papers. Table 5 lists their definitions.

Trustworthiness M_t**:** With values in $[0, 1]$, with 1 being the best, this measures the proportion of points in D that are also close in $P(D)$ [20]. A related metric measures the false neighbors of a projected point [97]. M_t tells how much one can trust that local patterns in a projection, *e.g.*, clusters, represent actual patterns in the data. In the definition (Tab. 5), $U_i^{(K)}$ is the set of points that are among the K nearest neighbors of point i in the 2D space but not among the K nearest neighbors of point i in $\mathbb{R}^n$; and $r(i, j)$ is the rank of the 2D point j in the ordered set of nearest neighbors of i in 2D. We chose $K = 7$ for this study, in line with [13], [98];

Continuity M_c**:** With values in $[0, 1]$, with 1 being the best, this measures the proportion of points in $P(D)$ that are also close together in D [20]. It is closely related to the missing neighbors of a projected point [97]. In the definition (Tab. 5), $V_i^{(K)}$ is the set of points that are among the K nearest neighbors of point i in $\mathbb{R}^n$ but not among the K nearest neighbors in 2D; and $\hat{r}(i, j)$ is the rank of the $\mathbb{R}^n$ point j in the ordered set of nearest neighbors of i in $\mathbb{R}^n$. As for M_t, we chose $K = 7$.

Normalized stress M_σ**:** With values in $[0, 1]$, with 0 being the best, this measures the preservation of point-pairwise distances from D to $P(D)$ [73]. Different distance metrics Δ^n for D, and Δ^q for $P(D)$ respectively can be used, the most typical being the Euclidean one. Good projections have low stress values. By weighting distances differently depending on points having the same label or not, stress can be adapted to account for labeled data [99]. However, we do not use this variant as plain stress is far more often used in the literature.

Neighborhood hit M_{NH}**:** With values in $[0, 1]$, with 1 being the best, this is the proportion of the K neighbors $N_i^{(K)}$ of a point i in $P(D)$ that have the same label l as point i itself, averaged over all points in $P(D)$ [86]. M_{NH} measures how well separable labeled data is in a projection $P(D)$, which helps gauging if the technique P is good to explore such data for, *e.g.*, classifier design purposes [100]. As before, we set here $K = 7$. This metric is applicable, by definition, only to labeled datasets, and is similar in purpose to existing techniques [101]. Note also that using this metric makes most practical sense only if the data is well separable into classes in the original $\mathbb{R}^n$ space. The datasets used in our survey come all from well-known benchmarks for ML, in particular classifier design, so they should have this property.

We also considered other metrics for our benchmark, *e.g.*, Kullback-Leibler divergence [102], Local Continuity meta-criterion [103], Topographic Product [104], and Procrustes Measure [105]. Interpreting such metrics is harder [10] and thus provides arguably less (clear) feedback for our typical target users, so we refrained from using them.

Visual separation metrics: A special class of scalar quality metrics aims to capture perceived *visual separation* of clusters in scatterplots [106], [107], [108]. Closer to our context, Tatu *et al.* [109], [110] study four such metrics on 2D scatterplots containing labeled samples. Sedlmair and Aupetit survey 14 additional metrics for the same goal [111]. Both above papers conclude that Distance Consistency (DSC) [112] (called Class Consistency Measure (CCM) in [110]), defined as the percentage of points $\mathbf{x}$ whose nearest class-center-of-mass belongs to the same class as $\mathbf{x}$, best approximates the way humans rank visual separation. More recently, ML techniques were proposed to search a large space of 2002 synthesied metrics to capture even more accurately human

Table 4
Selected projection techniques for evaluation and their trait values (Sec. 5.1). In the *Complexity* column, n is the number of dimensions; N is the number of samples; i is the number of iterations (for neural network training), and w is the number of weights (for a neural network).

Projection	linearity	Input	Neighborhood	Free parameters	Complexity	Out-of-sample	Inverse transform	Deterministic	Implementation
AE	nonlinear	samples	global	network size	$O(iNw)$	yes	no	no	Keras
DM	nonlinear	samples	local	2	$O(N^3)$	no	no	yes	Tapkee
FA	linear	samples	global	1	$O(n^3)$	yes	no	yes	scikit-learn
FMAP	nonlinear	distances	global	0	$O(N)$	no	no	yes	Vispipeline
GDA	nonlinear	distances	global	1	$O(n^3)$	no	no	yes	DR Toolbox
GPLVM	nonlinear	distances	global	1	$O(n^3)$	no	no	no	DR Toolbox
F-ICA	linear	samples	global	2	$O(n^3)$	yes	yes	yes	scikit-learn
IDMAP	nonlinear	samples	local	3	$O(N^2)$	no	no	yes	Vispipeline
ISO	nonlinear	samples	local	1	$O(N^3)$	yes	no	yes	scikit-learn
L-ISO	nonlinear	samples	local	1	$O(N^3)$	no	no	no	Vispipeline
LAMP	nonlinear	samples	local	3	$O(Nn)$	yes	yes	no	Vispipeline
LE	nonlinear	distances	local	0	$O(N^3)$	no	no	no	scikit-learn
LLC	nonlinear	samples	local	3	$O(in^3)$	no	no	yes	DR Toolbox
LLE	nonlinear	samples	local	3	$O(N^3)$	yes	no	no	scikit-learn
H-LLE	nonlinear	samples	local	3	$O(N^3)$	yes	no	no	scikit-learn
M-LLE	nonlinear	samples	local	3	$O(N^3)$	yes	no	no	scikit-learn
LMNN	linear	samples	local	3	$O(n^2)$	no	no	yes	DR Toolbox
LPP	linear	samples	global	1	$O(N^3)$	yes	no	yes	Tapkee
LSP	nonlinear	samples	local	4	$O(N^3)$	no	no	yes	Vispipeline
LTSA	nonlinear	samples	local	3	$O(N^3)$	yes	no	no	scikit-learn
L-LTSA	linear	samples	local	1	$O(N^3)$	no	no	no	Tapkee
MC	nonlinear	samples	local	2	$O(in^3)$	no	no	yes	DR Toolbox
MCML	nonlinear	samples	local	0	$O(n^2)$	no	no	no	DR Toolbox
MDS	nonlinear	distances	global	2	$O(N^3)$	no	no	no	scikit-learn
L-MDS	nonlinear	distances	global	1	$O(N^3)$	no	no	no	Tapkee
N-MDS	nonlinear	samples	global	2	$O(iN^2)$	no	no	no	scikit-learn
L-MVU	nonlinear	samples	global	2	$O(N^3)$	no	no	no	DR Toolbox
NMF	linear	samples	global	4	$O(n^2)$	yes	yes	no	scikit-learn
PBC	nonlinear	samples	local	4	$O(N\sqrt{N})$	no	no	yes	Vispipeline
PCA	linear	samples	global	0	$O(n^3)$	yes	yes	yes	scikit-learn
I-PCA	linear	samples	global	0	$O(n^3)$	yes	yes	no	scikit-learn
K-PCA-P	nonlinear	samples	global	1	$O(N^3)$	yes	yes	no	scikit-learn
K-PCA-R	nonlinear	samples	global	1	$O(N^3)$	yes	yes	no	scikit-learn
K-PCA-S	nonlinear	samples	global	1	$O(N^3)$	yes	yes	no	scikit-learn
P-PCA	linear	samples	global	1	$O(N^3)$	yes	no	yes	DR Toolbox
S-PCA	linear	samples	global	3	$O(N^3)$	yes	no	yes	scikit-learn
PLSP	nonlinear	samples	global	0	$O(N^3)$	no	no	yes	Vispipeline
G-RP	nonlinear	samples	global	0	$O(Nn^3)$	yes	no	no	scikit-learn
S-RP	nonlinear	samples	global	0	$O(Nn^3)$	yes	no	no	scikit-learn
R-SAM	nonlinear	samples	global	0	$O(iN^2)$	yes	no	no	Vispipeline
T-SNE	nonlinear	distances	local	3	$O(iN^2)$	no	no	no	Multicore TSNE
SPE	nonlinear	samples	global	2	$O(N^2)$	no	no	no	Tapkee
T-SVD	linear	samples	global	1	$O(N^2)$	yes	yes	no	scikit-learn
UMAP	nonlinear	distances	local	3	$O(iN^2)$	yes	yes	no	umap-learn

perception [113]. While such measures can very effectively model human perception of class separation, they cannot be *directly* used in our context: Our aim is to model how well a DR scatterplot captures aspects of the nD data, and not how the plot is actually perceived by users for a given task. By analogy, while the ground-truth in the comparison in [111], [113] and related work is human perception, such ground truth is the nD data in our case. In terms of [10], we are interested in the DR errors at the so-called 'model stage', not at the visualization stage. Hence, we cannot directly use such metrics in our case. The only exception here is the DSC metric, which relates the 2D scatterplot to the underlying nD data structure. However, DSC's formulation assumes that clusters are *well separated* in nD (for details see [110], [112]). Our benchmark, and the space $\mathcal{D}$, contain more general datasets which do not necessarily comply with this constraint.

Other scalar metrics exist and, as with any work, we had to make choices of which aspects we leave out of our study. For instance, *stability* [10] is a relevant metric which quantifies how much a projection changes upon changes of either its parameters or changes in the input data. Ideally, projections should be smooth functions of such changes, so small data and/or parameter changes imply only small changes in the visual result. However, formally defining stability in both these senses is hard, and quantifying it over the entire spectrum of data- and parameter-changes is even harder. Hence, our choice to leave stability out of the evaluation.

6.2 Point-pair metrics

While simple to compute and interpret, scalar metrics average a projection's quality over all its points. Comparing DR techniques using only averages is either misleading or not insightful enough. This was recognized by Joia *et al.* [73] when comparing the distortions of (Euclidean) distances caused by several projection techniques, and further elaborated by Nonato *et al.* [10]. For example, two projections may have similar average distortions, but one may preserve small distances better than the other, which makes it more suitable for, *e.g.*, cluster analysis. To capture such aspects, point-pair metrics measure properties of every point pair in the data (and projection result), as follows.

Shepard diagram [73]: The Shepard diagram is a scatterplot of the pairwise (Euclidean) distances between all points in

Table 5
Projection quality metrics computed in this survey. Right column gives the metric ranges, with optimal values marked bold. See Sec. 6.

Metric	Definition	Type	Range
Trustworthiness (M_t)	$1 - \frac{2}{NK(2n-3K-1)} \sum_{i=1}^{N} \sum_{j \in U_i^{(K)}} (r(i,j) - K)$	scalar	$[0, \mathbf{1}]$
Continuity (M_c)	$1 - \frac{2}{NK(2n-3K-1)} \sum_{i=1}^{N} \sum_{j \in V_i^{(K)}} (\hat{r}(i,j) - K)$	scalar	$[0, \mathbf{1}]$
Normalized stress (M_σ)	$\frac{\sum_{ij} (\Delta^n(\mathbf{x}_i, \mathbf{x}_j) - \Delta^q(P(\mathbf{x}_i), P(\mathbf{x}_j)))^2}{\sum_{ij} \Delta^n(\mathbf{x}_i, \mathbf{x}_j)^2}$	scalar	$[\mathbf{0}, 1]$
Neighborhood hit (M_{NH})	$\sum_{i=1}^{N} \frac{\lvert j \in N_i^{(K)} : l_j = l_i \rvert}{KN}$	scalar	$[0, \mathbf{1}]$
Shepard diagram (S)	Scatterplot $(\lVert \mathbf{x}_i - \mathbf{x}_j \rVert, \lVert P(\mathbf{x}_i) - P(\mathbf{x}_j) \rVert), 1 \leq i \leq N, i \neq j$	point-pair	-
Shepard goodness (M_S)	Spearman rank correlation of Shepard diagram	scalar	$[0, \mathbf{1}]$
Average local error ($M_a(i)$)	$\frac{1}{N-1} \sum_{j \neq i} \left\lvert \frac{\Delta^n(\mathbf{x}_i, \mathbf{x}_j)}{\max_{i,j} \Delta^n(\mathbf{x}_i, \mathbf{x}_j)} - \frac{\Delta^q(P(\mathbf{x}_i), P(\mathbf{x}_j))}{\max_{i,j} \Delta^q(P(\mathbf{x}_i), P(\mathbf{x}_j))} \right\rvert$	local (per-point)	$[\mathbf{0}, 1]$

$P(D)$ *vs* the corresponding distances in D. The closer the plot is to the main diagonal, the better overall distance preservation is. Plot areas below, respectively above, the diagonal indicate distance *ranges* for which false neighbors, respectively missing neighbors, occur. We quantitatively assess the quality of a Shepard diagram by computing its Spearman rank correlation M_S. A value of $M_S = 1$ indicates a perfect (positive) correlation of distances.

Other point-pair metrics include the co-ranking matrix [22] of the pairwise (Euclidean) distances between all points in $P(D)$ *vs* corresponding distances in D. It is related to the Shepard diagram as both their main diagonals can be interpreted similarly. The co-ranking matrix allows analyzing false and missing neighbors. Yet, summarizing this matrix to a simple to interpret value, as we did for the Shepard diagram using the Spearman rank, is harder. Hence, we did not include this metric in our benchmark.

6.3 Local Metrics

Both scalar and point-pair metrics are *sample-agnostic*, *i.e.*, they do not tell how projection errors correlate with specific samples or sample groups. Knowing this is important to assess which *patterns* in a projection $P(D)$ one can trust and which not. Several so-called spatial distribution, visual, distortion, or local, metrics have been proposed for this. These take different values for each point in $P(D)$, *i.e.* have $k = N$ in Eqn. 2, as follows.

Projection precision score [30]: This is the normalized distance between the two k-dimensional vectors having as components Euclidean distances between a point $\mathbf{y} \in P(D)$ and its K nearest neighbors in D, respectively $P(D)$, visualized by color-coding $P(D)$. Yet, this metric cannot differentiate false from missing neighbors;

Stretching and compression [114], [115]: These measure the increase (stretching), respectively decrease (compression) of distances of a point $\mathbf{y} \in P(D)$ *vs* all other points in $P(D)$ *vs* the corresponding distances in D. These metrics are visualized using a Voronoi-based partitioning of the 2D projection space which, as the authors note, may lead to bias due to how Voronoi cells depend on small perturbations of their underlying sites;

Average local error [97]: This metric assigns, for each point i, the averaged sum $M_a(i)$ of differences between its normalized distances in $\mathbb{R}^n$ and $\mathbb{R}^q$ to all other points j in the dataset (Tab. 5). $M_a(i)$ ranges in $[0, 1]$. Small values indicate good placement of point i *vs* all other points. This metric has been adapted to also show neighborhood preservation [98]. It is typically displayed using heat maps.

Local metrics show subtle differences between DR techniques. Yet, they need more presentation space in contrast to scalar and point-pair metrics, and also need to be visually (manually) assessed. Given space limits, we next consider only the average local projection error M_a, which we analyze for a subset of projections and datasets (Sec. 8.4.2).

7 Measurement Method

Following Eqn. 3, we need to evaluate our metrics $\overline{M}$ on all projection techniques in $\overline{P}$ applied to all considered datasets in $\overline{D}$ (Fig. 1d). In this process, we must also consider the *free parameters* that different projection techniques expose (Tab. 4). Obviously, the quality metric values will depend on these parameter choices. We next discuss the parameter search process used to handle this.

First, we need to define what is an *optimal* projection with respect to our quality metrics. For this, we aggregate the considered metrics to yield

$$\mu = \frac{1}{5}(M_{NH} + M_t + M_c + (1 - M_\sigma) + M_S) \qquad (4)$$

where M_{NH}, M_t, M_c, M_σ, and M_S are the scalar neighborhood hit, trustworthiness, continuity, stress, and Shepard goodness metrics in Sec. 6. As all these metrics range over $[0, 1]$, and we consider them equally important, they have identical weights in Eqn. 4. Eqn. 4 does not consider local metrics (Sec. 6.3) since these yield images meant to be qualitatively assessed, and also can not be always ranked (ordered) in terms of one image being globally better or worse than another one.

Let now $\pi_i \in \Pi_i$ denote the free parameters of a projection technique $P \in \overline{P}$, where Π_i indicate their allowable ranges. Let $\mu(P(D, \pi_i))$ be the aggregated quality of P run over dataset D with parameter values π_i. How can we then define what the "optimal" quality for a technique P is (which we need next to compare techniques)? We propose for this two solutions, as follows.

Dataset-wise view: For each $D \in \overline{D}$, we compute the optimal projection $P^{opt}(D)$ by doing a grid search that maximizes μ (Eqn. 4) over the ranges Π_i of all parameters π_i of P, *i.e.*,

$$P^{opt}(D) = \arg\max_{\pi_i \in \Pi_i} \mu(P(D, \pi_i)) \qquad (5)$$

The exact details of the optimization process, including the parameters π_i and their ranges Π_i for each technique P, are given in Appendix **??**. During this process, we were careful to reset the random seeds used by non-deterministic optimization algorithms, *e.g.*, t-SNE, at the beginning of each subsequent execution of the same algorithm. This way, the results we report in here can be replicated.

We next denote the quality of $P^{opt}(D)$ by $\mu^{opt}(P, D)$, and the parameter set realizing this quality by $\pi^{opt}(P, D)$ respectively. The values $\mu^{opt}(P, D)$ allow us to study how well a technique P performs over the different types of datasets in $\overline{D}$, and also how well different techniques perform for the same dataset. Key to this is that the techniques are *optimized independently per dataset*. This allows seeing the "best" that a given technique can yield for each dataset. Also, by studying the distribution of optimal parameters $\pi^{opt}(P, D)$ over $\overline{D}$, we can assess how much parameter tuning a technique P actually needs in practice. The results of using the dataset-wise view on our benchmark are discussed in Sec. 8.1).

Projection-wise view: The dataset-wise view chooses a separate optimal parameter set $\pi^{opt}(P, D)$ for each dataset D. Clearly, this expensive grid-search cannot be done in routine practice. Rather, users like to have so-called parameter *presets* when applying a technique P on *any* dataset. The projection-wise view measures quality in this way: For each technique $P \in \overline{P}$, we choose as preset the parameter-set $\pi^{preset}(P) \in \pi^{opt}(P, D)$ that yields the best quality μ most times (statistical mode) over all datasets in $\overline{D}$. The projection-wise view allows comparing projection techniques more globally, *i.e.*, seeing how they perform with respect to each other when no per-dataset parameter tuning takes place. The results of using the projection-wise view on our benchmark are discussed in Sec. 8.2).

8 Results

After evaluating our benchmark, we obtain a multidimensional dataset consisting of five quality metrics (plus the aggregated one, see Eqn. 4), measured for 44 projections, each run over 18 datasets, and additional values for optimal parameters – in total, over 5000 measured values. We next present and discuss ways to (visualy) explore this measurements dataset to gain insights on the tested projections, and answer different questions on various levels of detail.

8.1 How good are projections, and for which data?

To answer this, we use the dataset-wise view (Sec. 7). The table in Fig. 2 has rows for DR techniques P and columns for datasets D. Each cell shows the optimal quality value $\mu^{opt}(P, D)$, color coded by a sequential colormap. Grey cells show techniques which could not complete the projection of the respective datasets (crashing or hanging).

Scanning Fig. 2 along *rows* shows how much the optimal quality of a given projection varies over the studied datasets. For instance, we see that the projection-set starting with GLPVM and ending with LLE has quite similar (and high) optimal qualities over all datasets — that is, Fig. 2 shows a relatively dark-green compact block of cells starting with the GLPVM row and ending with the LLE row. In contrast, if we focus in Fig. 2 on the block spanned by the PCA-class projection rows (starting with PCA and ending with S-PCA), over all columns, we see little variation of colors along columns and more variation along rows, respectively. Hence, the PCA-class projections have quite similar optimal qualities for the given dataset, but the optimal qualities vary as a function of the dataset itself. Another salient pattern is given by N-MDS. While N-MDS only failed for a single dataset (*spambase*, grey cell on N-MDS row in Fig. 2), the respective row shows a higher error for N-MDS for basically all datasets and compared to most other projection techniques.

Scanning the figure along columns shows which are the best projections for a given dataset. For instance, we see that the earlier-mentioned set of projections starting with GPLVM and ending with LLE, and also all PCA variants, yield very good quality for the *orl*, *secom*, and *seismic* datasets. This can be explained by the fact that these datasets have a low intrinsic dimensionality (see Tab. 3) and these projection techniques handle very well such data. In contrast, *sentiment* has the second-highest intrinsic dimensionality of all datasets, and we see that it also yields relatively lower optimal qualities for all projections. Overall, t-SNE, UMAP, IDMAP and PBC perform well on average for most datasets. PLSP, LLTSA, LPP and GDA perform poorly. The PCA variants perform reasonably well for most datasets.

To better understand quality, we next explore how easy is to obtain optimal values for it (Sec. 8.1.1), and how quality depends on parameter values (Sec. 8.1.2) and dataset types (Sec. 8.1.3).

8.1.1 How easy is to obtain optimal quality?

The rightmost four columns of Fig. 2 show the standard deviations of the optimal parameters $\pi^{opt}(P, D)$, computed over all datasets D treated by every projection P, normalized to the interval $[0, 1]$, depicted by a heat colormap (darker = higher standard deviation). Empty cells correspond to techniques that have less than four parameters (see Appendix A). Good projections are those which yield high optimal quality values (green cells along their rows) *and* achieve this with little parameter tuning (light cells in the parameter columns, if they have any parameters). For instance, ISO is better than L-ISO as it achieves overall the same maximal quality, but requires less parameter tuning (0.27 *vs* 0.47 variance). PBC is better than NMF as it achieves slightly higher maximal quality and requires less parameter tuning for that. We also see that the four overall best performing techniques (t-SNE, UMAP, IDMAP and PBC) require some tuning effort over most parameters to yield optimal results. Of the techniques that do not require parameter tuning, PCA is the best performing, albeit with lower quality than that of the best projections.

8.1.2 How does quality depend on parameter settings?

To give more insight into this, inside each cell, a four-bin histogram shows how many runs, with different parameter values, done during the grid-search process over the parameters π_i, achieved a quality μ in the ranges $[0.0, 0.62)$, $[0.62, 0.75)$, $[0.75, 0.87)$, and $[0.87, 1.0]$ respectively. These seemingly arbitrary ranges were selected because most of

Table 6
Correlation between dataset traits and optimal quality values.

Intrinsic dim.	Sparsity ratio	Dimensionality
-0.630390	-0.289365	-0.090593

the data is located above the 0.5 threshold, so creating uniformly divided bins would not produce the desired outcome. Note that above 0.5, the bins are divided uniformly. Histograms with long *leftmost* bars indicate that, for the respective projection technique and dataset combination, most runs (parameter values) yield bad quality (undesirable situation). Histograms with long *rightmost* bars indicate that the respective technique-dataset combination achieves good quality for most parameter combinations (desirable situation). For instance, K-PCA-P has overall one long leftmost bar for all datasets, so it yields poor quality for most of the tested parameter combinations. There is no technique that shows long rightmost bars for all datasets. Hence, it is very hard to consistently obtain high quality for all datasets by parameter tuning. Histograms having several *non-zero* bars indicate methods where parameter tuning is crucial to obtain good performance, *e.g.*, LLE. Histogram shapes also depend on the *dataset*: For the *svhn* dataset column, most histograms are 'spread out', indicating that this dataset is hard – it requires more parameter tuning than other datasets to get good quality. For *imdb*, most histograms have one long leftmost bar, telling that this dataset is *very hard* to project regardless of parameter tuning.

8.1.3 How does quality depend on dataset type?

To answer this, Table 6 presents the correlation between dataset traits (Sec. 4.1) and the optimal quality values in Fig. 2. Several findings follow:

- *Intrinsic dimensionality* ρ_n is the trait that most influences quality μ, with average correlation -0.63. Hence, data with high intrinsic dimensionality is hard to project by all studied techniques;
- *Sparsity ratio* γ_n follows, with a correlation of -0.29 to quality, indicating that sparser datasets are harder to project well;
- *Dimensionality* n has a very low correlation of -0.09, barely affecting quality. The same holds for dataset type τ_D and size N. Hence, one should not worry in practice about these traits when choosing a projection technique that should yield high quality. Of course, such traits influence other aspects, such as computational speed (discussed later in Sec. 8.4.1).

8.2 How good are parameter-preset projections?

Using parameter presets is desirable for typical end users. We examine how well projections perform using presets with the projection-wise view (Sec. 7). Figure 3 shows a table with the same layout as the dataset-wise view (Fig. 2). However, we now compute the quality μ^{preset} using the *same* preset parameters $\pi^{preset}(P)$. The four rightmost colums show the preset values. Comparing this image with Fig 2, we see how quality drops overall. For more insight, we show the quality loss $\mu^{opt} - \mu^{preset}$ separately in Fig. 4. Figure 3 answers several practical questions:

- The four right columns shows which parameter *presets* one can use for each projection technique to get overall good quality, regardless of the dataset;
- Comparing *rows* allows seeing how two projections fare, quality-wise, when using presets. Overall, t-SNE, UMAP, IDMAP and PBC are the best-performing techniques in this sense;
- Comparing *columns* shows datasets which are 'easy' (*e.g.*, *orl*, *secom*, and *seismic*) or 'hard' (*e.g.*, *cnae9*, *imdb*, and *sentiment*) to project well when using presets. When one has a concrete dataset, one can find the benchmark dataset sharing similar traits (see table at the bottom of Fig. 3) and infer how a given projection would perform on it, or which is a good projection for this kind of dataset, using presets. This allows a first rough selection of good projection candidates.

8.3 Which projections perform similarly well?

The dataset-wise and projection-wise views convey many details on the specific behavior of a projection as function of the datasets and parameter tuning. However, this amount of detail can be overwhelming for the end user interested in comparing projections on a high level. Moreover, we do not have insights into the behavior of projections *vs* their raw, non-aggregated, quality metrics. For this, we consider each projection technique P attributed by the values of its five quality metrics (Eqn. 4), averaged over all datasets D. We next project this set using MDS and color the resulting scatterplot by the average quality μ (Fig. 5a). Similar, but more elaborated designs, have been used to compare projections, [116], [117]. This 'projection of projections' map shows how similar all techniques are from the perspective of all raw quality metrics over all datasets. We see a clear gradient of the average quality μ ranging from N-MDS and GDA (poorest) to UMAP, t-SNE, PBC and IDMAP (best). We also see that methods in the same family perform relatively similar, *e.g.*, the PCA variants. To explain the direction orthogonal to the color gradient, we color points (projection techniques) in turn by each metric and look for patterns. We find that this direction maps well the stress M_σ. These insights depend, of course, on the quality of the MDS projection used. To choose a good projection for this dataset, we could find DR methods that score well on datasets having similar trait values ($n = 5$, $N = 44, \gamma_n = 1, \tau_D = tabular$) following our analysis in Sec. 8.1. We do not take this path here since this dataset is very small and simple, and thus arguably projectable well by established methods such as MDS. To gain more confidence, we redo the plot using t-SNE (Fig. 5b). The orientation of the average quality (color gradient) and stress axes differs, but the overall pattern is very similar. Using these plots, users can compare projection techniques from the perspective of overall quality (to choose optimal ones), but also can choose techniques which behave similarly to a given technique of interest.

8.4 Detailed study of selected good projections

Several of our analyses so far point out that the top-four quality projections are UMAP, t-SNE, PBC and IDMAP. We now analyze these in more detail, from the perspective of speed (Sec. 8.4.1) and the way they distribute their errors

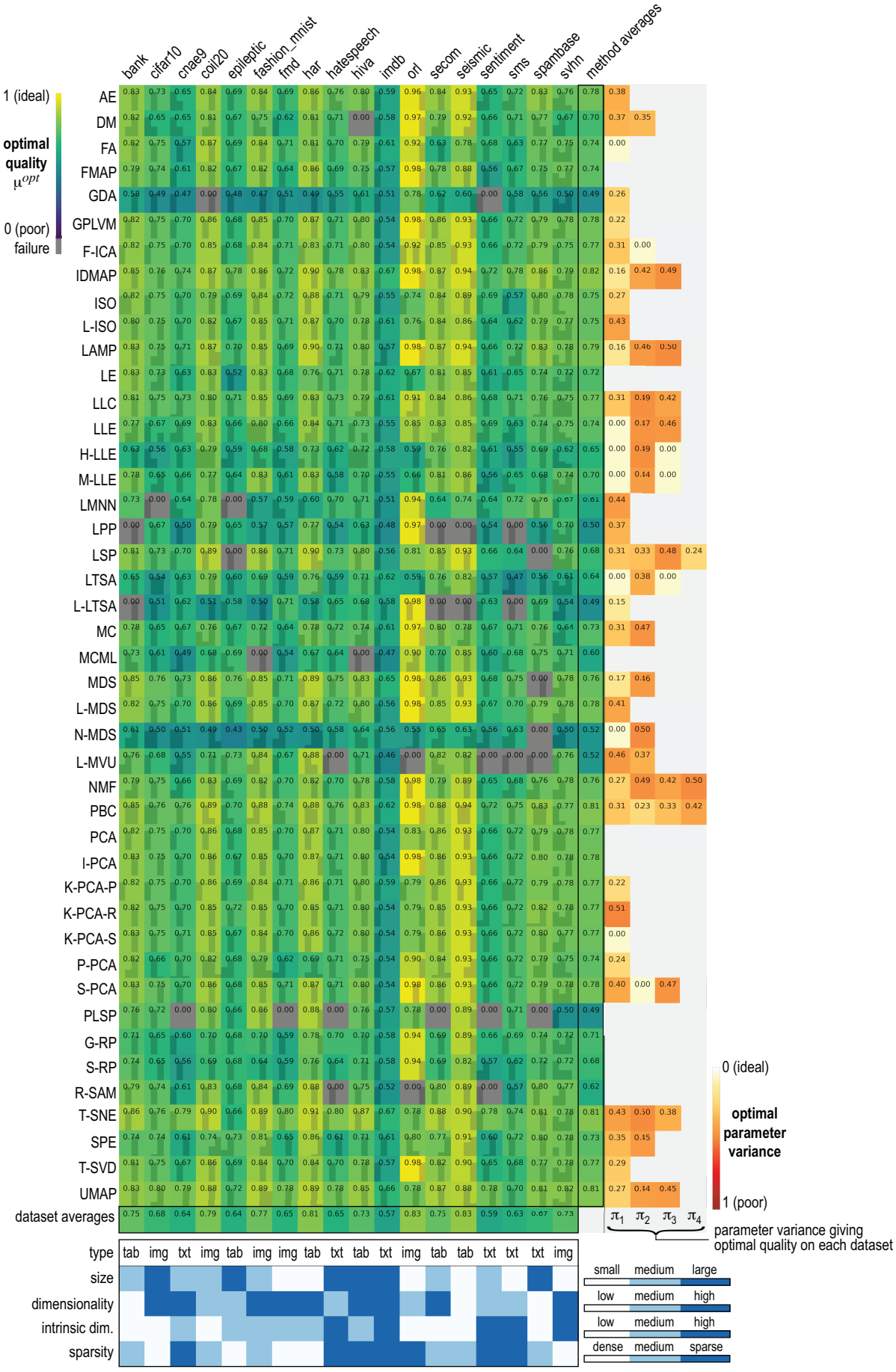

Figure 2. Dataset-wise view showing optimal quality per dataset (columns), all projections (rows). Histograms in each cell indicate number of runs divided into four quality bins, the bottom table shows dataset trait values. See Sec. 8.1. Parameters π_i are discussed in Appendix 1.

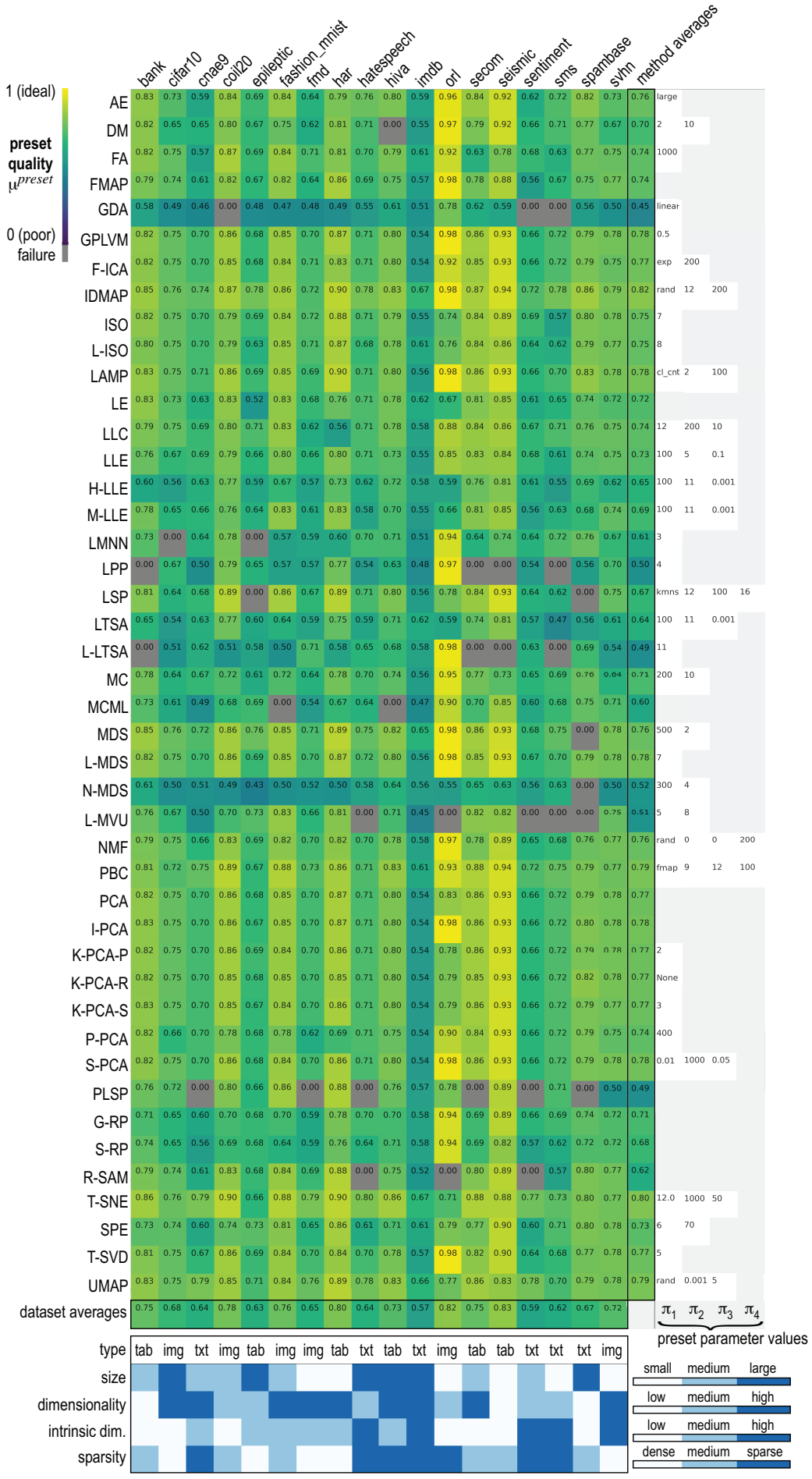

Figure 3. Projection-wise view (Sec. 8.2) of quality per dataset (columns), all projections (rows) for preset parameters. Appendix 1 discusses parameters π_i.

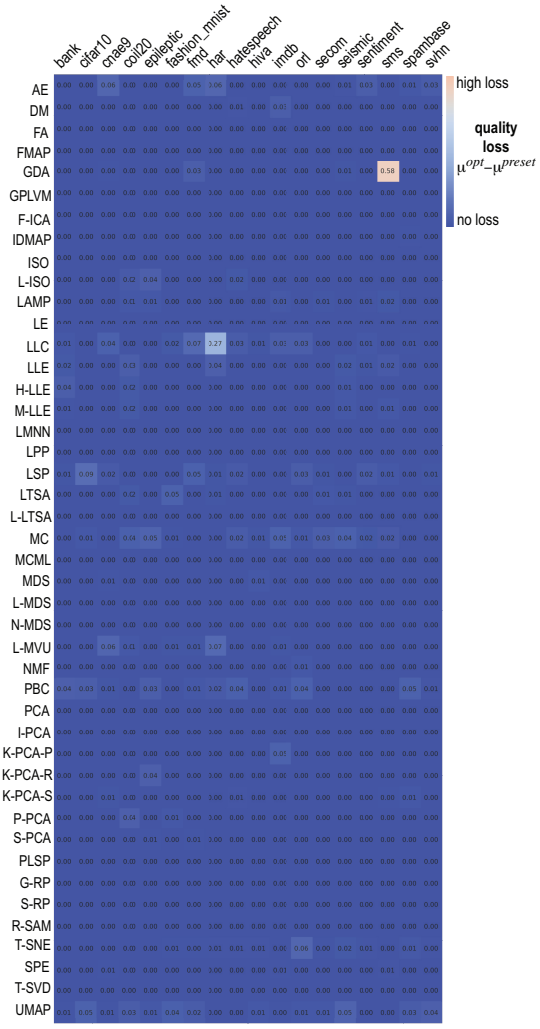

Figure 4. Quality loss $\mu^{opt} - \mu^{preset}$. See Sec. 8.2.

over the 2D space (Sec. 8.4.2). These extra insights can help users selecting a best technique from this top-four set.

8.4.1 How fast are the best projections?

We measure the speed of the four selected projections on synthetic Gaussian datasets for varying number of dimensions n and observations N. We sample n ranging from 50 to 1,000, and N ranging from 500 to 50,000, with 30 samples each, yielding 900 datasets that we next project and time. Figure 6 shows the results. Note that the four color scales correspond to different time ranges, as the four techniques have very different speeds. Normalizing all colors within the same range would suppress seeing interesting variations of the speed *vs* the parameters n and N. Hence, we chose to normalize colors per projection, and rely to annotations to convey the different time scales. We see that UMAP and PBC are almost two orders of magnitude faster than t-SNE and IDMAP. Color gradients tell us that the dimensionality n affects speed more for PBC and t-SNE than for UMAP and IDMAP. For the latter two, the sample count N affects speed more. Also, we see that t-SNE's color gradient is less smooth, being 'punctured' at a few places by outliers such as the bright yellow one (Fig. 6, red surrounding marker). These indicate combinations of n and N for which t-SNE took significantly longer than for similar input values, and are due to the stochastic nature of the algorithm itself. In contrast, the patterns exhibited by UMAP, PBC, and IDMAP show a smoother variation of speed with n and N.

8.4.2 How do projections spread their errors?

We now analyze in detail how the four best techniques spread their errors over the projection $P(D)$ using Shepard diagrams and local metrics (Sec. 6). Since we cannot present the analysis of all 18 datasets in $\overline{D}$ for space reasons, we select a subset of four datasets, and run the four selected techniques on them using their parameter presets (Fig. 3). The selected datasets represent each type of data considered in this study, namely *text* (**cnae9**), *tables* (**har**), and *images* (**coil20**, **fashion_mnist**).

First, we use Shepard diagrams (Sec. 6.2) to see how well the four techniques preserve high-dimensional distances (Fig. 7 left). Overall, we see that IDMAP preserves distances better than the other three techniques. At the other end, UMAP creates the most complex pattern, including both compressing and stretching distances from $\mathbb{R}^n$ to 2D. PBC and t-SNE create quite similar patterns. This is quite interesting, as it tells that we can use PBC to get very similar results to t-SNE, and PBC is about two orders of magnitude faster (Fig. 6).

Next, we show the actual projection scatterplots (Fig. 7 right), colored by the average local error M_a (Sec. 6.3). For each scatterplot, we color code M_a using a low (yellow) to high (purple) colormap. Per-scatterplot minimal and maximal M_a values are shown under the plots. We obtain several insights:

Emerging patterns: We see that the visual patterns formed by t-SNE and PBC are quite similar, in line with the earlier-detected similarity of their distance patterns in Shepard diagrams (Fig. 7 left). In contrast, IDMAP creates less well-separated visual clusters than all other three techniques, while UMAP creates more separated visual clusters. However, we should note that, without additional information on the ground-truth (nD data), the presence or absence or well-separated clusters in the projection is not an indication of the projection's accuracy.

Errors correlate with datasets: Looking at the M_a extrema, we see that all techniques find *fashion_mnist* to be the hardest to project, followed by *coil20*, *cnae9*, and *har*.

Error correlation with techniques: Overall, IDMAP produces the lowest errors. The other three techniques however cannot be decisively ranked, as they sometimes produce higher, and sometimes lower, errors than each other depending on the dataset. Moreover, achieving a higher pattern segregation typically implies higher M_a errors, compare,*e.g.*, t-SNE *vs* IDMAP (**har** or **fashion_mnist** datasets). Hence, M_a should not be used as a discriminative tool for comparing projections: when studying a projection computed by a given technique, M_a is most useful for finding which scatterplot points are best (worst) projected. For a similar use-case of projection-error color-coding, see [118]. This task is discussed further below.

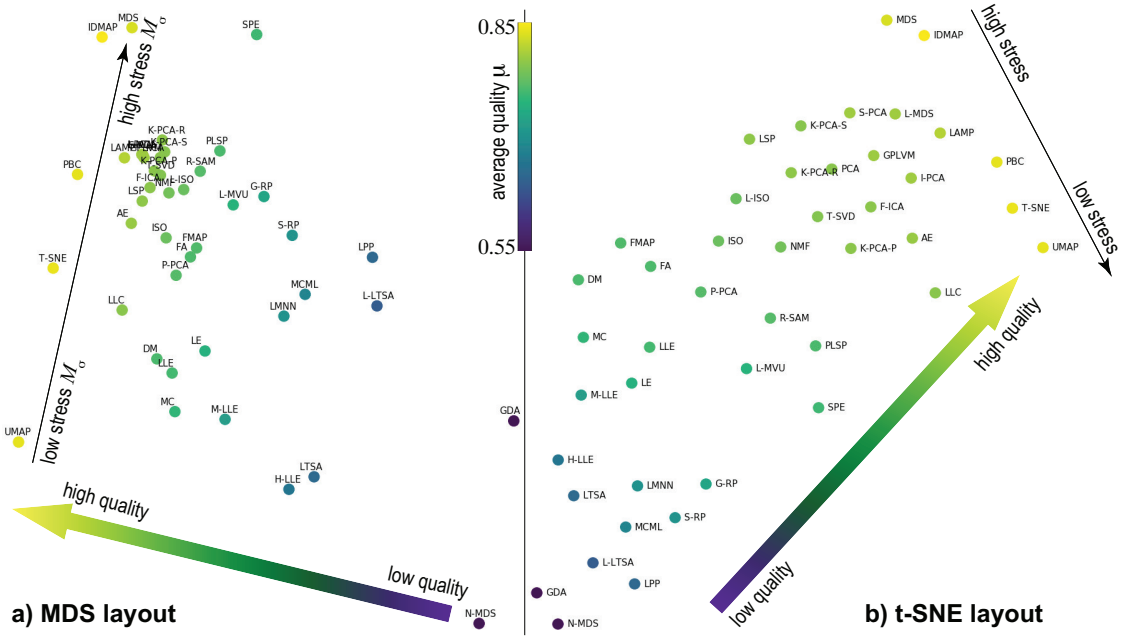

Figure 5. Projection of projections based on five quality metrics. Color shows average quality μ. See Sec. 8.3.

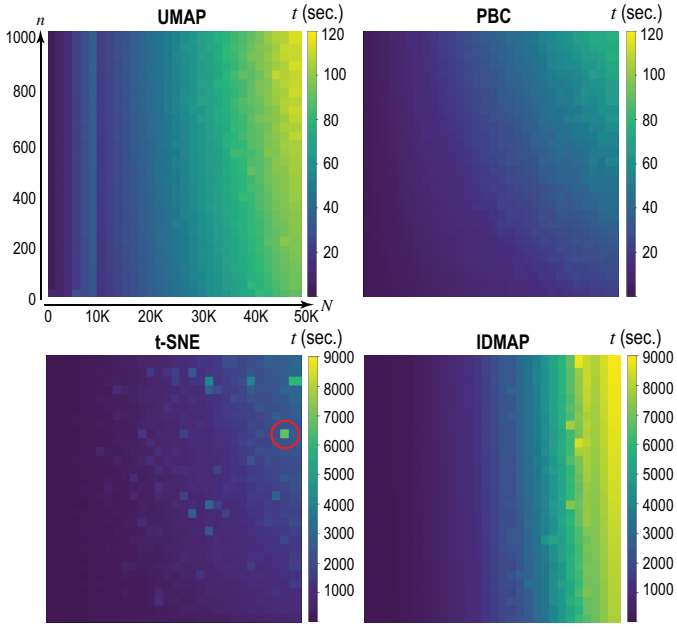

Figure 6. Running times for UMAP, PBC, t-SNE and IDMAP for synthetic Gaussian datasets with n dimensions, N samples. See Sec. 8.4.1.

Error distribution: All techniques generate quite similar distributions of error values (over the error range) for all datasets, with typically few high-error points. Lowest-error points (yellow) occur most often close to the scatterplot boundaries, which has also been observed for different datasets and projections earlier [97]. In contrast, high error points (purple) appear at very different locations as a function of the technique and dataset. Hence, to actually trust a given projection, one should always (be able to) inspect such errors.

Summarizing the above, we see that t-SNE and PBC offer the best overall quality in terms of producing low errors on average, good segregation of similar point-groups (clusters), and few high-error points.

9 Discussion

Typical surveys of projection methods propose taxonomies that cover such methods and help readers understand their underlying algorithmics and finding technically-related methods. Typical papers introducing new projection techniques present these, and offer (usually quite limited) qualitative and, sometimes, quantitative comparison with a few other techniques. Our survey covers quite different material addressing different, more practical, goals. We next discuss several observations we made during this work.

Benchmark: We present, to our knowledge, the first workflow for quantitatively evaluating projection techniques 'in the large'. For this, we describe high-dimensional data along five traits, and propose a representative sampling thereof using 18 real-world datasets of widely different dimensionality, size, type, intrinsic dimensionality, and sparsity. We next select 44 projection techniques which include, arguably, all well-known algorithms in the literature. We evaluate these techniques on these datasets along five quality metrics. In contrast to all similar evaluations so far, we study quality variation as a function of (a) the dataset traits, and (b) algorithm parameters. The entire benchmark (datasets, measurements, source code for techniques and measurement tools) is public [119], being the first such benchmark in the dimensionality reduction field. The entire workflow is implemented in Python. Specific projection implementation details are given in Tabs. 4 and **??**.

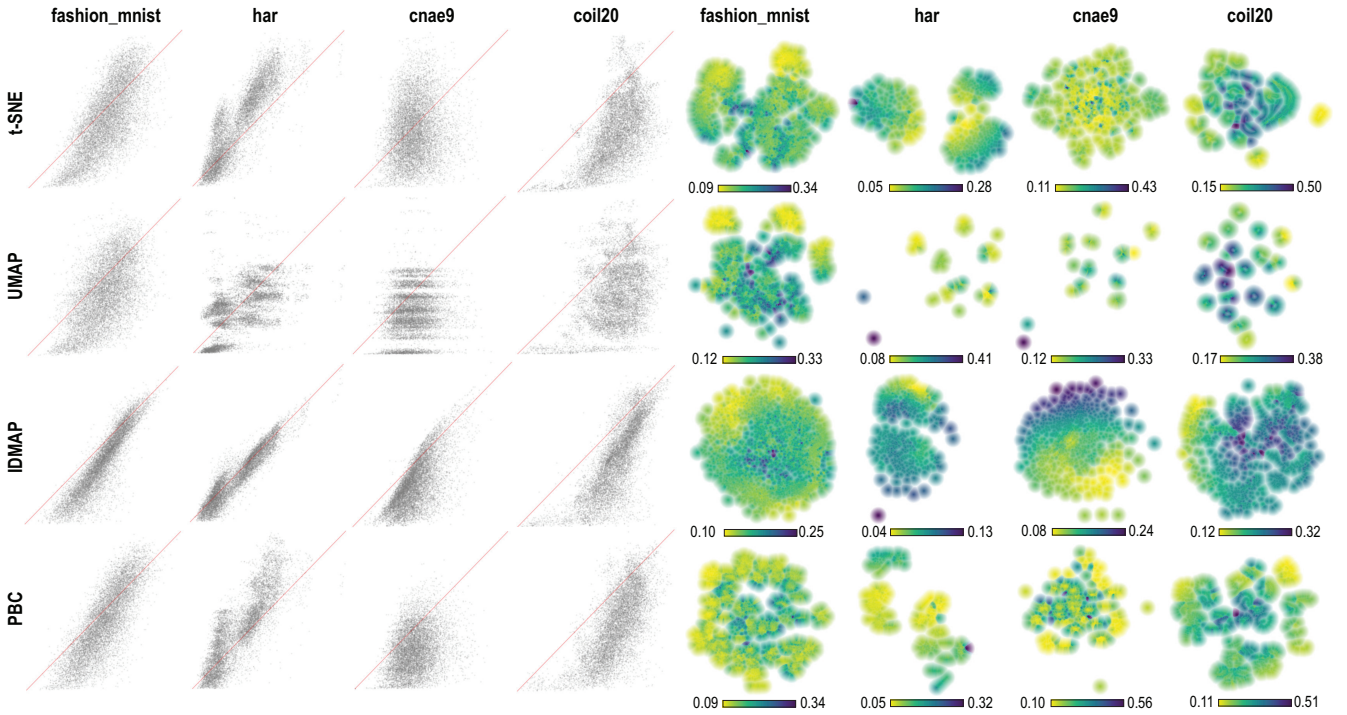

Figure 7. Left: Shepard diagrams for the analyzed four projections and four datasets. The x and y axes map inter-point distances in $\mathbb{R}^n$, respectively $\mathbb{R}^2$. Right: Average local errors M_a for the same datasets and projections. See Sec. 8.4.2.

Best-quality projections: Our studies showed that t-SNE, UMAP, PBC, and IDMAP yield the best quality *vs* the considered metrics and over the considered datasets, when using preset parameters. Our parameter analysis also shows that these techniques yield high quality quite consistently when their parameters are tuned. We also provide parameter presets and show that using these decrease the optimal quality of the studied projections only slightly. All in all, this tells end users that choosing one of these four techniques, with its respective parameter presets, can consistently deliver good quality.

Similar-quality projections: We compare all 44 studied projections from the perspective of all 5 quality parameters. Our results show that the "space" of projection techniques can be easily ordered, from low to high quality ones, and that the notion of average quality (using the 5 proposed quality metrics) does make sense—see smooth color-coded average quality gradient in Fig. 5. This helps end users to see which projections behave similarly quality-wise, supporting trade-off scenarios, when one wants to swap a technique for a similar-quality one that has, *e.g.*, a more robust, or faster, implementation.

Refining decisions: We analyze the top-four best quality techniques from the additional viewpoints of speed, distance preservation, and error spread over the 2D space. Our results show that the four techniques are quite different, even if their scalar (aggregated) quality metrics are quite similar. We discover that UMAP and PBC are about two orders of magnitude faster than t-SNE and IDMAP. However, UMAP has the worst distance-preservation pattern of the four. This offers directly actionable ways for end users to select a suitable projection from this set of four depending on their desires regarding speed and/or distance preservation.

Limitations: Densely covering the huge space of dataset types, projection techniques, algorithm parameters, and quality metrics is definitely very hard. Our work so far represents only a limited *sample* B of this space (Sec. 2). For instance, one could consider more datasets, traits, trait classes, quality metrics, or consider more runs of the considered projections, to account for those which have a non-deterministic behavior. An interesting avenue is to generate synthetic datasets that sample the desired dataset traits in a user-controlled manner. Doing this would allow a richer, and more automated, evaluation. However, how to suitably construct such a controlled dataset generator, able also to generate labels for well-separated point groups (needed for the visual assessment of projection results and computing the neighborhood hit), is not a trivial question, hence one that we consider for a significant future-work iteration.

However, our sample is considerably *denser* than other similar samples (evaluations) present in the literature, in all the considered aspects (datasets, parameter values, quality metrics, and number of studied projection techniques). Hence, we argue that our work is a necessary (but definitely not final) next step from current state-of-the-art in the quest of quantitatively evaluating the projection landscape.

We make all our results (methodology, data, code, measurements) open and public [119], so B is a 'live benchmark' that will grow as us, or others, will add datasets, techniques, and metrics to it. This way, coverage can increase over time with incremental efforts, sparing professionals from the very large effort required to set up such work from scratch.

Concrete directions in which we plan to extend this work include (a) considering more dataset traits (Tab. 3), such as amount and type of noise; and (b) adding visual quality metrics to quantify the *perceived* quality of projections for given tasks, *e.g.*, class separation [106], [109], [111], and metrics for the robustness of projection to noise.

10 Conclusion

This paper presents a survey of multidimensional projection techniques from the perspective of end users interested in understanding how specific algorithms, and their parameter settings, perform on specific types of high-dimensional datasets. For this, we proposed a methodology for constructing a benchmark that includes 44 techniques (including various combinations of their parameter values), 18 datasets, and 7 quality metrics. We propose an automatic way to evaluate this benchmark, and also several visualizations to analyze the gathered data. Our main contribution is making the methodology, benchmark, and related artifacts (datasets, techniques, metrics, visualizations, related code) publicly open, so interested researchers can study these results but also contribute to enrich the benchmark. Additionally, our current evaluation of the benchmark can be used to choose projection that score best on any of the evaluated criteria, similar to each other, or on global average quality, with t-SNE, UMAP, PBC, and IDMAP being the top-ranked ones in the latter respect.

Many extensions are possible based on the current foundation. First, given its open source nature, our benchmark can be easily enhanced by adding more techniques, metrics, and datasets. In particular, adding the many metrics proposed by recent approaches using machine learning [113] is a low-ganging fruit. In this process, the size and dimensionality of the collected evaluation datasets will also grow. Hence, we will consider new visualization methods to explore the gathered data to better answer concrete end-user questions, such as why do certain techniques behave similarly; which parameters of a given technique most strongly affect a given quality metric; and which techniques are best suited to project datasets having certain traits. Last but not least, coupling preojection metrics measured on real-world datasets to gauge which technique is actually better for which VA task, following up on [10] is an important potential extension. With these extensions, we hope that ours, and others' contributions, will make the benchmark grow to be a useful 'live' resource for the infovis and Machine Learning communities at large.

Acknowledgments

To be inserted in the final version of the paper.

References

[1] P. Hoffman and G. Grinstein, "A survey of visualizations for high-dimensional data mining," *Information Visualization in Data Mining and Knowledge Discovery*, vol. 104, pp. 47–82, 2002.
[2] S. Liu, D. Maljovec, B. Wang, P.-T. Bremer, and V. Pascucci, "Visualizing high-dimensional data: Advances in the past decade," *IEEE TVCG*, vol. 23, no. 3, pp. 1249–1268, 2015.
[3] J. Kehrer and H. Hauser, "Visualization and visual analysis of multifaceted scientific data: A survey," *IEEE TVCG*, vol. 19, no. 3, pp. 495–513, 2013.
[4] J. Tang, J. Liu, M. Zhang, and Q. Mei, "Visualizing large-scale and high-dimensional data," in *Proc. WWW*, pp. 287–297, 2016.
[5] A. Yates, A. Webb, M. Sharpnack, H. Chamberlin, K. Huang, and R. Machiraju, "Visualizing multidimensional data with glyph SPLOMs," *Computer Graphics Forum*, vol. 33, no. 3, pp. 301–310, 2014.
[6] A. Inselberg and B. Dimsdale, "Parallel coordinates: A tool for visualizing multi-dimensional geometry," in *Proc. IEEE VIS*, pp. 361–378, 1990.
[7] R. Rao and S. K. Card, "The table lens: Merging graphical and symbolic representations in an interactive focus+context visualization for tabular information," in *Proc. ACM SIGCHI*, pp. 318–322, 1994.
[8] A. C. Telea, "Combining extended table lens and treemap techniques for visualizing tabular data," in *Proc. EuroVis*, pp. 120–127, 2006.
[9] R. Becker, W. Cleveland, and M. Shyu, "The visual design and control of trellis display," *Journal of Computational and Graphical Statistics*, vol. 5, no. 2, pp. 123–155, 1996.
[10] L. Nonato and M. Aupetit, "Multidimensional projection for visual analytics: Linking techniques with distortions, tasks, and layout enrichment," *IEEE TVCG*, 2018.
[11] A. Buja, D. Cook, and D. F. Swayne, "Interactive high-dimensional data visualization," *Journal of Computational and Graphical Statistics*, vol. 5, no. 1, pp. 78–99, 1996.
[12] C. Sorzano, J. Vargas, and A. Pascual-Montano, "A survey of dimensionality reduction techniques," 2014. arXiv:1403.2877 [stat.ML].
[13] L. van der Maaten and E. Postma, "Dimensionality reduction: A comparative review," tech. rep., Tilburg University, Netherlands, 2009. Tech. report TiCC TR 2009-005.
[14] L. van der Maaten and G. E. Hinton, "Visualizing data using t-sne," *JMLR*, vol. 9, pp. 2579–2605, 2008.
[15] K. Bunte, M. Biehl, and B. Hammer, "A general framework for dimensionality reducing data visualization mapping," *Neural Computation*, vol. 24, no. 3, pp. 771–804, 2012.
[16] I. T. Jolliffe, "Principal component analysis and factor analysis," in *Principal Component Analysis*, pp. 115–128, Springer, 1986.
[17] W. S. Torgerson, *Theory and Methods of Scaling*. Wiley, 1958.
[18] I. K. Fodor, "A survey of dimension reduction techniques," tech. rep., US Dept. of Energy, Lawrence Livermore National Labs, 2002. Tech. report UCRL-ID-148494.
[19] H. Yin, "Nonlinear dimensionality reduction and data visualization: A review," *Intl. Journal of Automation and Computing*, vol. 4, no. 3, pp. 294–303, 2007.
[20] J. Venna and S. Kaski, "Visualizing gene interaction graphs with local multidimensional scaling," in *Proc. ESANN*, pp. 557–562, 2006.
[21] A. Gisbrecht and B. Hammer, "Data visualization by nonlinear dimensionality reduction," *WIREs Data Mining Knowledge Discovery*, vol. 5, pp. 51–73, 2015.
[22] J. A. Lee and M. Verleysen, "Quality assessment of dimensionality reduction: Rank-based criteria," *Neurocomputing*, vol. 72, no. 7, pp. 1431–1443, 2009.
[23] J. Cunningham and Z. Ghahramani, "Linear dimensionality reduction: Survey, insights, and generalizations," *JMLR*, vol. 16, pp. 2859–2900, 2015.
[24] H. Xie, J. Li, and H. Xue, "A survey of dimensionality reduction techniques based on random projection," 2017. arXiv:1706.04371 [cs.LG].
[25] S. Dasgupta, "Experiments with random projection," in *Proc. UAI*, pp. 143–151, Morgan Kaufmann, 2000.
[26] D. F. Swayne, D. Cook, and A. Buja, "XGobi: Interactive dynamic data visualization in the X window system," *J Computational and Graphical Statistics*, vol. 7, no. 1, pp. 113–130, 1998.
[27] D. Engel, L. Hüttenberger, and B. Hamann, "A survey of dimension reduction methods for high-dimensional data analysis and visualization," in *Proc. IRTG Workshop*, vol. 27, pp. 135–149, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2012.
[28] C. Seifert, V. Sabol, and W. Kienreich, "Stress maps: analysing local phenomena in dimensionality reduction based visualisations," in *Proc. IEEE VAST*, 2010.
[29] W. Lueks, A. Gisbrecht, and B. Hammer, "Visualizing the quality of dimensionality reduction," *Neurocomputing*, vol. 112, pp. 109–123, 2013.
[30] T. Schreck, T. von Landesberger, and S. Bremm, "Techniques for precision-based visual analysis of projected data," *Information Visualization*, vol. 9, no. 3, pp. 181–193, 2010.
[31] R. Bellman, *Dynamic Programming*. Princeton University Press, 1957.
[32] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When is "nearest neighbor" meaningful?," in *International conference on database theory*, pp. 217–235, Springer, 1999.

[33] S. Moro, P. Cortez, and P. Rita, "A data-driven approach to predict the success of bank telemarketing," *Decision Support Systems*, vol. 62, pp. 22–31, 2014.

[34] A. Krizhevsky and G. E. Hinton, "Learning multiple layers of features from tiny images," 2009. Technical report TR-04-09.

[35] P. M. Ciarelli and E. Oliveira, "Agglomeration and elimination of terms for dimensionality reduction," in *Proc. IEEE ISDA*, pp. 547–552, 2009.

[36] S. A. Nene, S. K. Nayar, H. Murase, *et al.*, "Columbia object image library (coil-20)," tech. rep., Columbia University, 1996. Technical report CUCS-005-96.

[37] R. G. Andrzejak, K. Lehnertz, F. Mormann, C. Rieke, P. David, and C. E. Elger, "Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state," *Physical Review E*, vol. 64, no. 6, pp. 061907–1–061907–8, 2001.

[38] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms," 2017. arXiv 1708.07747 [cs.LG].

[39] L. Sharan, R. Rosenholtz, and E. Adelson, "Material perception: What can you see in a brief glance?," *Journal of Vision*, vol. 9, no. 8, pp. 784–784, 2009.

[40] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine," in *Proc. Intl. Workshop on Ambient Assisted Living*, pp. 216–223, Springer, 2012.

[41] T. Davidson, D. Warmsley, M. Macy, and I. Weber, "Automated hate speech detection and the problem of offensive language," in *Proc. AAAI ICWSM*, pp. 512–515, 2017.

[42] I. Guyon, A. Saffari, G. Dror, and G. Cawley, "Agnostic learning vs. prior knowledge challenge," in *Proc. IEEE IJCNN*, pp. 829–834, 2007.

[43] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proc. NAACL-HLT*, pp. 142–150, Association for Computational Linguistics, 2011.

[44] F. S. Samaria and A. C. Harter, "Parameterisation of a stochastic model for human face identification," in *Proc. IEEE WACV*, pp. 138–142, 1994.

[45] M. McCann and A. Johnston, "SECOM dataset," 2008. UCI Machine Learning Repository.

[46] M. Sikora and L. Wróbel, "Application of rule induction algorithms for analysis of data collected by seismic hazard monitoring systems in coal mines," *Archives of Mining Sciences*, vol. 55, no. 1, pp. 91–114, 2010.

[47] D. Kotzias, M. Denil, N. de Freitas, and P. Smyth, "From group to individual labels using deep features," in *Proc. ACM SIGKDD*, pp. 597–606, 2015.

[48] T. A. Almeida, J. M. G. Hidalgo, and A. Yamakami, "Contributions to the study of SMS spam filtering: new collection and results," in *Proc. ACM Symposium on Document Engineering*, pp. 259–262, 2011.

[49] M. Hopkins, E. Reeber, G. Forman, and J. Suermondt, "Spambase dataset," 1999. Hewlett-Packard Labs.

[50] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," in *Proc. NIPS workshop on deep learning and unsupervised feature learning*, 2011.

[51] P. Rauber, A. Falcao, and A. Telea, "Visualizing time-dependent data using dynamic t-SNE," in *Proc. EuroVis – short papers*, pp. 43–49, 2016.

[52] L. McInnes, J. Healy, and J. Melville, "UMAP: Uniform manifold approximation and projection for dimension reduction," 2018. arXiv:1802.03426v2 [stat.ML].

[53] E. Amorim, E. Brazil, J. Daniels, P. Joia, L. Nonato, and M. Sousa, "iLAMP: Exploring high-dimensional spacing through backward multidimensional projection," in *Proc. IEEE VAST*, 2012.

[54] R. da Silva, P. Rauber, R. Martins, R. Minghim, and A. C. Telea, "Attribute-based visual explanation of multidimensional projections," in *Proc. EuroVA*, 2015.

[55] M. Espadoto, F. C. M. Rodrigues, and A. Telea, "Visual analytics of multidimensional projections for constructing classifier decision boundary maps," in *Proc. IVAPP*, SCITEPRESS, 2019.

[56] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 125–141, 2008.

[57] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *Journal of the Royal Statistical Society*, vol. 61, no. 3, pp. 611–622, 1999.

[58] H. Zou, T. Hastie, and R. Tibshirani, "Sparse principal component analysis," *Journal of Computational and Graphical Statistics*, vol. 15, no. 2, pp. 265–286, 2006.

[59] A. Hyvarinen, "Fast ICA for noisy data using Gaussian moments," in *Proc. IEEE ISCAS*, vol. 5, pp. 57–61, 1999.

[60] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proc. NIPS*, pp. 556–562, 2001.

[61] N. Halko, P.-G. Martinsson, and J. A. Tropp, "Finding structure with randomness: Stochastic algorithms for constructing approximate matrix decompositions," 2009. arXiv:0909.4061 [math.NA].

[62] X. He and P. Niyogi, "Locality preserving projections," in *Proc. NIPS*, pp. 153–160, 2004.

[63] J. B. Tenenbaum, V. De Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[64] Y. Chen, M. Crawford, and J. Ghosh, "Improved nonlinear manifold learning for land cover classification via intelligent landmark selection," in *Proc. IEEE IGARSS*, pp. 545–548, 2006.

[65] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[66] D. L. Donoho and C. Grimes, "Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data," *Proc. of the National Academy of Sciences*, vol. 100, no. 10, pp. 5591–5596, 2003.

[67] Z. Zhang and J. Wang, "MLLE: Modified locally linear embedding using multiple weights," in *Proc. NIPS*, pp. 1593–1600, 2007.

[68] Z. Zhang and H. Zha, "Principal manifolds and nonlinear dimensionality reduction via tangent space alignment," *SIAM journal on scientific computing*, vol. 26, no. 1, pp. 313–338, 2004.

[69] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Proc. NIPS*, pp. 585–591, 2002.

[70] R. R. Coifman and S. Lafon, "Diffusion maps," *Applied and computational harmonic analysis*, vol. 21, no. 1, pp. 5–30, 2006.

[71] M. Brand, "Charting a manifold," in *Proc. NIPS*, pp. 985–992, 2002.

[72] Y. W. Teh and S. T. Roweis, "Automatic alignment of hidden representations," in *Proc. NIPS*, pp. 841–848, 2002.

[73] P. Joia, D. Coimbra, J. A. Cuminato, F. V. Paulovich, and L. G. Nonato, "Local affine multidimensional projection," *IEEE TVCG*, vol. 17, no. 12, pp. 2563–2571, 2011.

[74] F. V. Paulovich and R. Minghim, "Text map explorer: a tool to create and explore document maps," in *Proc. IEEE IV*, pp. 245–251, 2006.

[75] R. Minghim, F. V. Paulovich, and A. A. Lopes, "Content-based text mapping using multi-dimensional projections for exploration of document collections," in *Proc. SPIE*, Intl. Society for Optics and Photonics, 2006.

[76] A. Globerson and S. T. Roweis, "Metric learning by collapsing classes," in *Proc. NIPS*, pp. 451–458, 2006.

[77] J. B. Kruskal, "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis," *Psychometrika*, vol. 29, no. 1, pp. 1–27, 1964.

[78] V. De Silva and J. B. Tenenbaum, "Sparse multidimensional scaling using landmark points," tech. rep., Stanford University, 2004.

[79] B. Schölkopf, A. Smola, and K.-R. Müller, "Kernel principal component analysis," in *Proc. ICANN*, pp. 583–588, Springer, 1997.

[80] N. D. Lawrence, "Gaussian process latent variable models for visualisation of high dimensional data," in *Proc. NIPS*, pp. 329–336, 2004.

[81] K. Q. Weinberger, F. Sha, and L. K. Saul, "Learning a kernel matrix for nonlinear dimensionality reduction," in *Proc. ICML*, 2004.

[82] L. van der Maaten, "An introduction to dimensionality reduction using Matlab," tech. rep., Maastricht University, 2007. Technical Report MICC 07-07.

[83] K. Q. Weinberger, B. Packer, and L. K. Saul, "Nonlinear dimensionality reduction by semidefinite programming and kernel matrix factorization.," in *AISTATS*, Citeseer, 2005.

[84] G. Baudat and F. Anouar, "Generalized discriminant analysis using a kernel approach," *Neural computation*, vol. 12, no. 10, pp. 2385–2404, 2000.

[85] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[86] F. V. Paulovich, L. G. Nonato, R. Minghim, and H. Levkowitz, "Least square projection: A fast high-precision multidimensional projection technique and its application to document mapping," *IEEE TVCG*, vol. 14, no. 3, pp. 564–575, 2008.

[87] F. V. Paulovich, D. M. Eler, J. Poco, , C. P. Botha, R. Minghim, and L. G. Nonato, "Piecewise laplacian-based projection for interactive data exploration and organization," *Computer Graphics Forum*, vol. 30, no. 3, pp. 1091–1100, 2011.

[88] E. Pekalska, D. de Ridder, R. P. W. Duin, and M. A. Kraaijveld, "A new method of generalizing Sammon mapping with application to algorithm speed-up," in *Proc. ASCI*, vol. 99, pp. 221–228, 1999.

[89] C. Faloutsos and K. Lin, "FastMap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets," *ACM SIGMOD*, vol. 24, no. 2, pp. 163–174, 1995.

[90] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[91] D. K. Agrafiotis, "Stochastic proximity embedding," *Journal of Computational Chemistry*, vol. 24, no. 10, pp. 1215–1221, 2003.

[92] K. Weinberger, J. Blitzer, and L. Saul, "Distance metric learning for large margin nearest neighbor classification," in *Proc. NIPS*, pp. 1473–1480, 2006.

[93] T. Zhang, J. Yang, D. Zhao, and X. Ge, "Linear local tangent space alignment and application to face recognition," *Neurocomputing*, vol. 70, no. 7-9, pp. 1547–1553, 2007.

[94] S. Lespinats, M. Aupetit, and A. Meyer-Baese, "ClassiMap: A new dimension reduction technique for exploratory data analysis of labeled data," *IJPRAI*, vol. 29, no. 6, 2015.

[95] Y. Wang, K. Feng, X. Chu, J. Zhang, C.-W. Fu, M. Sedlmair, X. Yu, and B. Chen, "A perception-driven approach to supervised dimensionality reduction for visualization," *IEEE TVCG*, vol. 24, no. 5, pp. 1828–1840, 2018.

[96] G. Pölzlbauer, "Survey and comparison of quality measures for self-organizing maps," in *Proc. Workshop on Data Analysis (WDA)*, pp. 67–82, 2004.

[97] R. Martins, D. Coimbra, R. Minghim, and A. C. Telea, "Visual analysis of dimensionality reduction quality for parameterized projections," *Computers & Graphics*, vol. 41, pp. 26–42, 2014.

[98] R. Martins, R. Minghim, and A. C. Telea, "Explaining neighborhood preservation for multidimensional projections," in *Proc. CGVC*, pp. 121–128, Eurographics, 2015.

[99] Y. Koren and L. Carmel, "Robust linear dimensionality reduction," *IEEE TVCG*, vol. 10, no. 4, pp. 459–470, 2004.

[100] P. E. Rauber, A. X. Falcão, and A. C. Telea, "Projections as visual aids for classification system design," *Information Visualization*, vol. 17, no. 4, pp. 282–305, 2017.

[101] M. Aupetit, "Sanity check for class-coloring-based evaluation of dimension reduction techniques," in *Proceedings of the Fifth Workshop on Beyond Time and Errors: Novel Evaluation Methods for Visualization*, pp. 134–141, ACM, 2014.

[102] G. E. Hinton and S. T. Roweis, "Stochastic neighbor embedding," in *Proc. NIPS*, pp. 857–864, 2003.

[103] L. Chen and A. Buja, "Local multidimensional scaling for nonlinear dimension reduction, graph drawing, and proximity analysis," *Journal of the American Statistical Association*, vol. 104, no. 485, pp. 209–219, 2009.

[104] H.-U. Bauer and K. R. Pawelzik, "Quantifying the neighborhood preservation of self-organizing feature maps," *IEEE Transactions on neural networks*, vol. 3, no. 4, pp. 570–579, 1992.

[105] Y. Goldberg and Y. Ritov, "Local procrustes for manifold embedding: a measure of embedding quality and embedding algorithms," *Machine learning*, vol. 77, no. 1, pp. 1–25, 2009.

[106] G. Albuquerque, M. Eisemann, and M. Magnor, "Perception-based visual quality measures," in *Proc. IEEE VAST*, pp. 11–18, 2011.

[107] M. Sedlmair, T. Munzner, and M. Tory, "Empirical guidance on scatterplot and dimension reduction technique choices," *IEEE TVCG*, pp. 2634–2643, 2013.

[108] R. Motta, R. Minghim, A. Lopes, and M. Oliveira, "Graph-based measures to assist user assessment of multidimensional projections," *Neurocomputing*, vol. 150, pp. 583–598, 2015.

[109] A. Tatu, G. Albuquerque, M. Eisemann, J. Schneidewind, H. Theisel, M. Magnor, , and D. Keim, "Combining automated analysis and visualization techniques for effective exploration of high dimensional data," in *Proc. IEEE VAST*, pp. 59–66, 2009.

[110] A. Tatu, P. Bak, E. Bertini, D. Keim, and J. Schneidewind, "Visual quality metrics and human perception: An initial study on 2D projections of large multidimensional data," in *Proc. AVI*, pp. 49–56, ACM, 2010.

[111] M. Sedlmair and M. Aupetit, "Data-driven evaluation of visual quality measures," *Comp Graph Forum*, vol. 34, no. 3, pp. 545–559, 2015.

[112] M. Sips, B. Neubert, J. Lewis, and P. Hanrahan, "Selecting good views of high-dimensional data using class consistency," *Comp Graph Forum*, vol. 28, no. 3, pp. 831–838, 2009.

[113] M. Aupetit and M. Sedlmair, "SepMe: 2002 new visual separation measures," in *Proc. IEEE PacificVis*, 2016.

[114] M. Aupetit, "Visualizing distortions and recovering topology in continuous projection techniques," *Neurocomputing*, vol. 10, no. 7–9, pp. 1304–1330, 2007.

[115] S. Lespinats and M. Aupetit, "CheckViz: Sanity check and topological clues for linear and nonlinear mappings," *Computer Graphics Forum*, vol. 30, no. 1, pp. 113–125, 2011.

[116] R. Cutura, S. Holzer, M. Aupetit, and M. Sedlmair, "VisCoDeR: A tool for visually comparing dimensionality reduction algorithms," in *Proc. ESANN*, Univ. Catholique Louvain, 2018.

[117] J. Peltonen and Z. Lin, "Information retrieval approach to meta-visualization," *Machine Learning*, vol. 99, no. 2, pp. 189–229, 2015.

[118] N. Heulot, J.-D. Fekete, and M. Aupetit, "Visualizing dimensionality reduction artifacts: An evaluation," 2017. arXiv:1705.05283v1 [cs.HC].

[119] The Authors, "Dimensionality reduction online benchmark," 2019. https://mespadoto.github.io/proj-quant-eval.

Mateus Espadoto is currently a PhD student at the Institute of Mathematics and Statistics, University of São Paulo and at the Bernoulli Institute, University of Groningen. He has about 20 years of experience in data management and software development. His research interests include machine learning, high-dimensional data visualization and visual analytics.

Rafael M. Martins received his double PhD in Computer Science (2016) from the Universities of São Paulo and Groningen on explanatory visualizations of dimensionality reduction. He is currently Senior Lecturer in Computer Science and Media Technology at Linnaeus University, Sweden. He is interested in diverse aspects and applications of multidimensional visualization.

Andreas Kerren received his PhD in Computer Science from Saarland University, Germany and habilitation (2008) from Växjö University, Sweden. He is full professor at the Department of Computer Science and Media Technology, Linnaeus University (Sweden), where he heads the Information and Software Visualization (ISOVIS) group. His research interests cover Information Visualization, Visual Analytics, and Human-Computer Interaction. He is editorial board member of the Information Visualization and Computer Graphics Forum journals, was program chair at IEEE VISSOFT 2013/2018 and IVAPP 2013-15/2018-20, and has edited several human-centered visualization books.

Nina S. T. Hirata holds a PhD degree in Computer Science from the University of São Paulo, Brazil. Currently, she is an associate professor of the Computer Science Department, Institute of Mathematics and Statistics, at the same university. Her main research interest is in the development and application of machine learning based techniques on image analysis and understanding.

Alexandru C. Telea received his PhD (2000) in Computer Science from the Eindhoven University of Technology, the Netherlands. He was assistant professor in visualization and computer graphics at the same university (until 2007) and then full professor of visualization at the University of Groningen, the Netherlands. Since 2019 he is full professor of visual data analytics at Utrecht University, the Netherlands. His interests include high-dimensional visualization, visual analytics, and image-based information visualization.