DOI: 10.1111/cgf.13977
Eurographics Conference on Visualization (EuroVis) 2020
M. Gleicher, T. Landesberger von Antburg, and I. Viola
(Guest Editors)

Volume 39 (2020), Number 3

Quantitative Evaluation of Time-Dependent Multidimensional Projection Techniques

E. F. Vernier[1,2], R. Garcia[1], I. P. da Silva[1], J. L. D. Comba[1] and A. C. Telea[3]

[1]Federal University of Rio Grande do Sul, Brazil
[2]University of Groningen, the Netherlands
[3]University of Utrecht, the Netherlands

Abstract

Dimensionality reduction methods are an essential tool for multidimensional data analysis, and many interesting processes can be studied as time-dependent multivariate datasets. There are, however, few studies and proposals that leverage on the concise power of expression of projections in the context of dynamic/temporal data. In this paper, we aim at providing an approach to assess projection techniques for dynamic data and understand the relationship between visual quality and stability. Our approach relies on an experimental setup that consists of existing techniques designed for time-dependent data and new variations of static methods. To support the evaluation of these techniques, we provide a collection of datasets that has a wide variety of traits that encode dynamic patterns, as well as a set of spatial and temporal stability metrics that assess the quality of the layouts. We present an evaluation of 9 methods, 10 datasets, and 12 quality metrics, and elect the best-suited methods for projecting time-dependent multivariate data, exploring the design choices and characteristics of each method. Additional results can be found in the online benchmark repository. We designed our evaluation pipeline and benchmark specifically to be a live resource, open to all researchers who can further add their favorite datasets and techniques at any point in the future.

CCS Concepts

• ***Computing methodologies*** → ***Dimensionality reduction and manifold learning;***

1. Introduction

Dimensionality reduction (DR) methods, also called projections, are used in many applications in information visualization, machine learning, and statistics. Compared to other high-dimensional data visualization techniques, projections are especially effective for datasets with many observations (also called samples or points) and attributes (also called measurements, dimensions, or variables) [LMW*17]. Many projection techniques exist, with wide varieties of computational efficiency, ease of use, ability to preserve and/or enhance different data patterns. Surveys have also focused on assessing quantitative and qualitative aspects of projection techniques [NA19, VDMPVdH09, EMK*19], thereby helping practitioners in choosing a suitable one for a given context.

Most projection techniques have been designed and evaluated only for *static* data. Projecting dynamic (time-dependent) data is, however, equally important. Such data is found in most science and engineering areas, such as biology [TBZVC17], medicine [GF19], and finance [Kra19]. The body of research in time series visualization is rich [AMM*08], thereby underlining the importance of visualizing such data. Yet, there are only few examples of projecting time-dependent data [HWX*10, MDL07, WG11, BWS*12, NPTS17, JFSK16]. Even fewer works focus on designing projection techniques specifically for dynamic data [RFT16, FCS*19]. In particular, it is not clear how to measure *and* trade-off two key aspects of such projections: *visual quality* and *stability*. While visual quality was studied well for static projections, stability, seen as the ability to create a set of projections that allows users to maintain a cohesive mental map through time, is recognized as essential for dynamic data visualization [APP11, BLIC19], but has not been formally defined nor quantified for dynamic projections.

We work towards filling this gap in assessing projection techniques for dynamic data with the following main contributions:

- We propose novel variations of existing static projection *techniques* for the context of visualizing time-dependent data;
- We propose a set of *metrics* to quantify the stability of dynamic projections;
- We *benchmark* the visual quality and stability of dynamic projections on a dataset collection to get insights on which methods favor which of the measured quality aspects.

Our work can help researchers in targeting the identified challenges of current dynamic projection techniques, therefore potentially leading to improved ones. Separately, practitioners can use our findings into the process of determining which dynamic projection technique is best suited to their given user context. Finally,

our creation of an open benchmark for assessing dynamic projections (containing datasets, techniques, metrics, visualizations, and associated workflows) should benefit both user types by providing a basis via which such techniques can be transparently compared.

The structure of this paper is as follows. Section 2 outlines related work and evaluation techniques for projections for static and dynamic data. Section 3 details the proposed experiment we conducted to quantitatively assess the behavior of projection techniques for dynamic data, including techniques, datasets, and evaluated metrics. Section 4 presents the obtained results. Section 5 discusses the causes of the observed dynamic projection behavior. Finally, Section 6 concludes the paper. For replication purposes, all our datasets, code, workflow, and results are openly available [VGdS*19].

2. Related work

2.1. Preliminaries

We first introduce some notation. Let $\mathbf{x} \in \mathbb{R}^n$ be an n-dimensional sample. A revision $\mathbf{R}^t = \{\mathbf{x}_i^t\}$, or timestep, of our data consists of a set of N samples $\mathbf{x}_i^t$, $1 \leq i \leq N$ measured at the same time moment t. A dynamic dataset $\mathbf{D}$ is a list of T revisions $\mathbf{D} = \{\mathbf{R}^t\}, 1 \leq t \leq T$. For simplicity of exposition and implementation, but without loss of generality, we consider next that the sample count N is constant over time. In this case, $\mathbf{D}$ can be represented as a set of T N-by-n matrices, one for each timestep.

A projection technique is a function $P : \mathbb{R}^n \to \mathbb{R}^q$, where $q \ll n$. For visualization purposes, $q \in \{2,3\}$. Since 2D projections are by far the most commonly used, we next only consider the case $q = 2$. We denote the projection of observation $\mathbf{x}$ by $P(\mathbf{x})$. For each timestep t, let $P(\mathbf{R}^t)$ be the 2D scatterplot of all points in $\mathbf{R}^t$. Finally, let $P(\mathbf{D})$ be the set of T scatterplots for all timesteps of dataset $\mathbf{D}$. These can be rendered as animations, small multiples, trail sets, or other visualization encodings.

Visualization of high dimensional data [LMW*17] is a well studied topic populated with many techniques such as parallel coordinate plots [ID90], table lenses [RC94], scatterplot matrices [BCS96a], and dimensionality reduction (DR) methods [NA19, VDMPVdH09, EMK*19]. From all these we next focus only on DR techniques, both for static and dynamic data, and evaluation methods for both of these technique classes.

2.2. Techniques for *static* dimensionality reduction

The body of research that encompasses static DR is large and spans the fields of Information Visualization and Machine Learning. There are dozens of static techniques designed to optimize different objectives and to work well under different constraints. These can be classified and categorized using several taxonomies [VDMPVdH09] that guide users in choosing methods that meet their requirements. We do not further elaborate on such techniques, as several surveys extensively discuss static projections. Fodor *et al.* [Fod02] present, to our knowledge, the first survey of DR techniques covering non-linear, vector quantization, and deep learning methods. Yin [Yin07] surveys non-linear DR methods. Bunte *et al.* [BBH12] proposes a framework to quantitatively compare nine DR methods. Cunningham *et al.* [CG15] presents a theoretical comparison of 15 linear DR techniques. A similar survey, extended to 30 DR techniques, both linear and non-linear, is provided by Sorzano *et al.* [SVPM14]. Additional surveys look at DR methods in the larger context of high-dimensional data visualization, thus comparing and contrasting them with other visualization techniques [BCS96b, HG02, EHH12, KH13]. The most recent survey in this area [NA19] discusses technical aspects of DR methods, and also how such methods satisfy various user-level tasks.

2.3. Evaluations of *static* dimensionality reduction

Taxonomies as the ones listed above, compare DR methods mainly from technical (algorithmic) and task-suitability aspects. An increasingly visible alternative approach is to compare techniques by measuring various quality *metrics* on several techniques and datasets. A wealth of such quality metrics exist – for recent overviews, see [Pöl04, LV09, LGH13, NA19, EMK*19]. Different metrics gauge different desirable aspects of a projection, and usually, several metrics are jointly used to assess DR quality [GH15]. Just as for DR techniques, metrics can be organized using different taxonomies. Following [EMK*19], these are as follows. *Aggregate* metrics, such as trustworthiness, continuity, neighborhood hit, distance and class consistency [SNLH09, TBB*10], cluster visual separation metrics [AEM11, SMT13, SA15], and metrics that capture human perception based on machine learning [AS16] characterize an entire 2D scatterplot by a single scalar value. This is convenient when comparing (many) different scatterplots to choose a suitable one, such as in scagnostics applications. However, a scatterplot may exhibit different quality values in different areas, so a single aggregated value may not be suitable [JCC*11, NA19]. *Point pair* metrics address this by measuring how point pairs $(P(\mathbf{x}), P(\mathbf{y}))$ in a projection relate to their corresponding sample pairs $(\mathbf{x}, \mathbf{y})$. These include Shepard diagrams [JCC*11] and co-ranking matrices [LV09]. Finally, *local* metrics gauge separately every (small) neighborhood in a projection, thus providing the highest level of detail, and are typically visualized atop of the projection itself. These include the projection precision score [SvLB10], stretching and compression [Aup07, LA11], and false neighbors, missing neighbors, and average local errors [MCMT14, MMT15].

Since all the above metrics aim to capture spatial aspects of the projection, we refer to them next as spatial quality metrics. Recent surveys have proposed extensive evaluations of spatial quality metrics on benchmarks containing a variety of datasets and DR methods [EMK*19, VDMPVdH09]. However, time-dependent datasets were not considered.

2.4. Techniques for *dynamic* dimensionality reduction

The literature is much less rich regarding DR methods that *explicitly* consider dynamic data. The dynamic t-SNE (dt-SNE) method of Rauber *et al.* [RFT16] extends the well-known t-SNE method [vdMH08] by adding a stability factor λ to the objective function. Such a factor jointly minimizes the Kullback-Leibler divergence proposed by t-SNE to preserve high-dimensional point neighborhoods and also restricts the amount of motion $\|P(\mathbf{x}^{t+1}) - P(\mathbf{x}^t)\|$ that points can have between consecutive timesteps. More recently,

Fujiwara *et al.* [FCS*19] proposed a PCA-based method to deal with streaming data. Note that this is a harder (and different) problem from the one we aim to study since one cannot anticipate changes occurring upstream in the data when optimizing for placement of points in 2D. As such, analyzing this (and similar) methods is out of our scope. Separately, several authors use DR methods to create static maps that describe multivariate time series. Hu *et al.* [HWX*10] use Self-Organizing Maps [KSH01] to create 2D trails that capture the dynamics of human motion data. Rauber *et al.* [RFT17] use similar trails, created by dt-SNE, to visualize the learning process of a neural network. Mao *et al.* [MDL07] use PCA to project text feature evolution in text sequences. Ward and Guo [WG11], Bernard *et al.* [BWS*12] and, more recently, Ali *et al.* [AJXW19] use similar approaches to find cyclic behavior, outliers, and trends in temporal data from medical, financial, and earth sciences domains. In contrast to the previous methods, m-TSNE [NPTS17] describes multivariate time series at a higher level of aggregation as single points instead of trails or polylines. Temporal MDS [JFSK16] projects $\mathbf{D}$ as a series of 1D projections, creating a map where the x-axis is time, and the y-axis shows the similarity of observations.

2.5. Evaluation of *dynamic* dimensionality reduction

Evaluating dynamic DR methods can be split into two aspects. First, just like for static DR methods, one is interested to see how well techniques capture the *spatial* aspects of the underlying data. For this, one typically uses the same types of spatial quality metrics as for static projections (Sec. 2.3). A separate important aspect for dynamic DR methods is *stability*. Loosely put, stability describes how a dynamic DR technique encodes *changes* in the data into *changes* in the 2D metaphor used to visualize the data. Such metaphors can be grouped into spatial ones, where different timesteps map to different plots, such as in small multiples; and animation-based ones, where different timesteps are encoded into frames of a 2D animation. Stability metrics were proposed and evaluated to assess the quality of other visualizations of dynamic data such as time-dependent treemaps [SSV18, VCT19, VTC18].

Stability is related to the capacity of a DR technique to deal with so-called out-of-core data. Simply put, this means the ability for a projection, created from a given dataset $\mathbf{D}$, to add extra points $\mathbf{X} \notin \mathbf{D}$ to the resulting 2D depiction $P(\mathbf{D})$, without distorting this depiction too much so that its understanding becomes hard. While recent works consider out-of-core and stability as key properties for DR projections [NA19, BFHL17, EHT19, GfVLD13, BSL*08], we are not aware of specific quality metrics that quantify these.

3. Experimental setup

To evaluate how dynamic DR techniques perform, we follow a methodology similar to the one proposed in [EMK*19] for evaluating static DR techniques, as follows. We first select a set of dynamic DR *techniques* to evaluate. Next, we select a collection of *datasets* that cover various aspects, or *traits*, that characterize high-dimensional dynamic data. Thirdly, we evaluate both spatial quality and stability *metrics* on all combinations of techniques and datasets; in this step, we also propose novel metrics to gauge stability. We describe all these steps next. The analysis of the discovered correlations between techniques, dataset traits, and quality metrics obtained from our experiments is discussed afterwards in Sec. 4.

3.1. Techniques

We selected the dynamic DR techniques to evaluate based on the following considerations. First, we only consider techniques P, which, given a dataset consisting of several timeframes $\mathbf{R}^t$, produce corresponding 2D scatterplots $P(\mathbf{R}^t)$. We argue that this is the most generic definition of a dynamic projection – from such scatterplots, other types of visualizations can be constructed next as desired (animation, small multiples, trails). This is analogous to expecting a generic static projection technique to deliver a 2D scatterplot. Hence, techniques that deliver different output types, such as m-TSNE [NPTS17] and temporal MDS [JFSK16], are excluded from our evaluation. Secondly, we only consider techniques that (1) are generic with respect to the input data (size, dimensionality, provenance) they can handle; (2) well-known and often used in practice, so their evaluation arguably serves a sizeable user group; and (3) easy to set up, control, and have publicly available implementations, for reproducibility. We next describe the selected techniques.

t-SNE and variants: Probably the simplest way to project dynamic data is to compute a single, global, projection $P(\mathbf{D})$ for the entire dataset $\mathbf{D}$ and next visualize the timeframes by using the desired method, be it animation, trails, or small multiples. We next call this the *global* (G) approach. While this arguably favors stability (since P sees all data $\mathbf{D}$ at once), it likely yields limited spatial quality, since P has the challenging task of placing well *all* points from *all* revisions in $\mathbf{D}$. An equally simple approach is to compute independent projections $P(\mathbf{R}^t)$ for each revision $\mathbf{R}^t$. We call this next the *per-timeframe* (TF) approach. This arguably favors spatial quality, since P must only optimize positions for each revision $\mathbf{R}^t$ separately, rather than the entire $\mathbf{D}$. However, this approach can yield poor stability, since timeframes are projected without knowledge of each other. Both the global and timeframe approaches were suggested, but not quantitatively evaluated, in the dt-SNE paper [RFT16]. Given this, and also the fact that t-SNE is a very well-known static technique, we next consider G-t-SNE, TF-t-SNE, and dt-SNE in our evaluation.

UMAP: This recent DR technique [MHSG18] has a mathematical foundation on Riemannian geometry and algebraic topology. According to recent studies [EMK*19, BMH*19], UMAP offers high-quality projections with lower computational cost and better global structure preservation than t-SNE, being thus an interesting competitor in the DR arena. We consider in our evaluation both the global (G-UMAP) and per-timeframe (TF-UMAP) variants of this technique.

PCA: Following [FCS*19, MDL07, WG11], we also consider Principal Component Analysis [Jol86], implementing the global and timeframe strategies. In detail, PCA performs a linear mapping of the data $\mathbf{D}$ to, in our case, 2D by maximizing the data variance in the 2D representation. The global strategy implies computing PCA once for the entire $\mathbf{D}$. In contrast, timeframe PCA means computing PCA separately for each revision $\mathbf{R}^t$. Given the widespread use of PCA in many fields of science, and also its out-of-core ability

(which, as outlined in Sec. 2.5, is related to stability), we consider both G-PCA and TF-PCA next in our evaluation.

Autoencoders: Often used in dimensionality reduction and representation learning, autoencoders [HS06, Bal87] are hourglass-shaped neural networks. They are composed of an encoder that takes the original data $\mathbf{D}$ and compresses it into a compact (latent) representation $P(\mathbf{D})$ of lower dimensionality (two in our case), and a decoder, which takes $P(\mathbf{D})$ and aims to reconstruct a good approximation of the original data $\mathbf{D}$. While autoencoders have been often used to create static projections of high-dimensional data, they have not, to our knowledge, been quantitatively evaluated for their ability to handle dynamic data. We evaluated four types of autoencoders, as follows. *Dense autoencoders* (AE) are comprised of only fully-connected (dense) layers and are the standard variant. *Convolutional autoencoders* (CAE) [MMCS11] have both fully-connected and convolutional layers. The convolutional layers apply a non-linear transformation to the data that takes into account the spatial correlation between attributes, for instance, the proximity of pixels in an image. *Variational autoencoders* may have both fully-connected layers (VAE) [KW13] and convolutional layers (CVAE). The main difference between dense and variational autoencoders is the addition of stochastic behavior in the intermediate layer of the latter. The encoder produces two vectors – an intermediate representation (IR) and an uncertainty degree σ for each IR value. The decoder tries to reconstruct the input through a sample from the latent space distribution with mean IR and variance σ, thus forcing the network to learn similar representation for similar inputs. Convolutional based architectures are not generic regarding input and a meaningful spatial relationship between attributes is expected (such as found on image data). We, therefore, restrain the analysis on this document to AE and VAE. The results of CAE and CVAE runs for the image based datasets (fashion and quickdraw) can be found on the online material [VGdS*19].

Implementation: We implemented the chosen dynamic DR techniques (G-t-SNE, TF-t-SNE, dt-SNE, G-UMAP, TF-UMAP, G-PCA, TF-PCA, AE, CAE, VAE, CVAE) as follows. For t-SNE and PCA, we used scikit-learn [PVG*11] with default parameters. For dt-SNE and UMAP, we used the implementation provided online by the authors [RFT16, MHSG18]. Finally, we implemented the four autoencoder models using Keras [C*15], with different numbers of layers, nodes per layer, optimizers, and training routines. Tab. 1 shows the values, for each autoencoder and dataset, that delivered the best results, and which we used next. The code, notebooks, and instructions to recreate our results are available online [VGdS*19].

3.2. Datasets

There is, to our knowledge, no standardized benchmark for evaluating DR techniques. Espadoto *et al.* [EMK*19] took a first step towards providing such a benchmark containing 19 datasets. However, all these are time-independent, thus not suitable for us. We followed here a similar approach, *i.e.* collecting a set of 10 high-dimensional and dynamic datasets that exhibit significant variations in terms of provenance, number of samples N, number of timesteps T, dimensionality n, intrinsic dimensionality ρ_n (percentage of n dimensions that explain 95% of the data variance), and sparsity ratio σ_n (percentage of zeros in the data). All datasets are labeled into 3 to 10 classes. We only use labels for visualization and quality assessment and not the projection itself. Table 2 shows the characteristics, or traits, for these datasets. Further details on them are listed below.

Table 1: *Hyperparameters of the autoencoder-based DR methods*

dataset	technique	# hidden layers	# nodes/layer	# epochs
cartolastd	AE	2	10, 10	50
cartolastd	VAE	2	10, 10	100
cifar10cnn	AE	2	10, 10	20
cifar10cnn	VAE	2	100, 10	20
esc50	AE	2	10, 10	40
esc50	VAE	2	100, 10	20
fashion	AE	3	500, 500, 2000	40
fashion	VAE	3	2048, 1024, 512	20
gaussians	AE	2	10, 10	20
gaussians	VAE	2	100, 10	20
nnset	AE	2	10, 10	20
nnset	VAE	2	100, 10	20
qtables	AE	2	10, 10	20
qtables	VAE	2	100, 10	20
quickdraw	AE	3	500, 500, 2000	40
quickdraw	VAE	3	2048, 1024, 512	20
sorts	AE	2	10, 10	20
sorts	VAE	2	100, 10	20
walk	AE	2	10, 10	20
walk	VAE	2	100, 10	20

- **cartolastd:** Player statistics for the second turn of the 2017 Brazilian football championship. Data was extracted from an open-source project [GG19] that scrapes the Cartola FC football platform. Each timestep corresponds to a tournament round. Variables relate to per-match performance of a given player (number of goals, assistances, fouls, defenses, etc.). Players are labeled by their playing position (goalkeeper, right or left-back, defender, midfielder, forward).
- **cifar10cnn:** Last hidden layer activations after each training epoch for a convolutional network trained to classify the CIFAR10 [Kri09] dataset.
- **esc50:** Sound samples of 8 classes (brushing teeth, chainsaw, crying baby, engine, laughing, rain, siren, wind) compressed to 128 frequencies and smoothed over time. Extracted from Piczak's ESC50 dataset [Pic15].
- **fashion:** 100 images from each of the 10 classes (T-shirt/top, trouser, pullover, dress, coat, sandal, shirt, sneaker, bag, ankle boot) of the FashionMNIST [XRV17] dataset with decreasing amounts of noise over time.
- **gaussians:** Synthetic dataset used to evaluate dt-SNE [RFT16]. Isotropic gaussian blobs in *nD* with diminishing spread over time.
- **nnset:** Internal states (weights and biases) of several neural networks during training with the MNIST dataset [LC10]. The networks have the same architecture but use different optimizers, batch sizes, and training-set sizes.
- **qtables:** Internal state of agents learning to move a car up a hill using the reinforcement learning algorithm Q-learning. The classes represent variations of learning rates and discounts.
- **quickdraw:** Drawing sequences for 600 objects of 6 different classes drawn by random people. Extracted from the "Quick, Draw!" Google AI experiment [JRK*16].
- **sorts:** This dataset was designed to compare the behavior of eight sorting algorithms. The algorithms sort different arrays of 100 random values in $[0, 1]$. As they do so, we take snapshots of

Table 2: *Datasets and their traits used in the evaluation.*

dataset	samples N	timesteps T	dimensions n	classes	intrinsic dim. ρ_n	sparsity ratio σ_n
cartolastd	696	19	17	5	0.6470	0.0000
cifar10cnn	1000	30	10	10	0.6599	0.0000
esc50	320	108	128	8	0.0345	0.0000
fashion	1000	10	784	10	0.4762	0.2971
gaussians	2000	10	100	10	0.3680	0.0000
nnset	80	30	8070	8	0.0057	0.0001
qtables	180	40	1200	9	0.0077	0.0007
quickdraw	600	89	784	6	0.4309	0.9013
sorts	80	100	100	8	0.3505	0.0100
walk	300	50	100	3	0.4783	0.0001

the intermediate states, until sorting is over. Each observed point is an (algorithm, array) run, and its feature vector is the partially sorted array at a given time.

- **walk:** Synthetic dataset with similar structure to *gaussians*. It contains 3 high-dimensional clusters oscillate (approach, intermingle and cross, and then drift apart) in $\mathbf{R}^{100}$ over time. We designed this dataset to see how well the studied DR techniques can capture the approaching, mingling, and drifting-away dynamics mentioned above.

Covering all variations of high-dimensional datasets with a benchmark is already daunting for static data [EMK*19], thus even more for dynamic data, as there are many types of dynamic patterns possible. Hence, we cannot claim that our benchmark is *exhaustive* in terms of the space it samples. However, we believe that the included datasets exhibit a rich variety of different traits (Tab. 2). Also, no two datasets are redundant, *i.e.*, have all traits similar. Given that, to date, no other benchmark exists for this task, we believe ours is a good start in supporting the intended evaluation.

3.3. Metrics

We measure the quality of all projection techniques (Sec. 3.1) on all datasets (Sec. 3.2) using both spatial quality and stability metrics, similarly to other evaluations of multivariate dynamic data visualizations such as treemaps [SSV18, VCT19, VTC18]. In our evaluation, we use the same metrics as the survey [EMK*19] (and a few extra ones) over all revisions $\mathbf{R}^t$, as follows.

3.3.1. Spatial metrics

Neighborhood preservation (S_{NP}): With values in $[0,1]$, with 1 being the best, this is the percentage of the k-nearest neighbors of $\mathbf{x} \in \mathbf{D}$ that project in the k-nearest neighborhood of $P(\mathbf{x})$.

Neighborhood hit (S_{NH}): With values in $[0,1]$, with 1 being the best, this is the fraction of the k-nearest neighbors of a projected point $P(\mathbf{x})$ that have the same class label as $P(\mathbf{x})$. Since we know that our datasets exhibit reasonably well-separated classes in $\mathbb{R}^n$, a proper DR technique (from the perspective of class separation tasks) should yield a high neighborhood hit.

Trustworthiness (S_{Trust}): With values in $[0,1]$, with 1 being the best, this measures how well the k nearest neighbors $NN^k(P(\mathbf{x}))$ of a projected point $P(\mathbf{x})$ match the k nearest neighbors $NN^k(\mathbf{x})$ of a data point $\mathbf{x}$. Simply put, trustworthiness measures how few missing neighbors [MCMT14] a projected point has. Formally, if $U^k(\mathbf{x})$ is the set of points that project in $NN^k(P(\mathbf{x}))$ but are not in $NN^k(\mathbf{x})$, and $r(\mathbf{x},\mathbf{y})$ is the rank of $\mathbf{y}$ in the ordered set of nearest neighbors $NN^k(P(\mathbf{x}))$, trustworthiness is then defined as $1 - \frac{2}{Nk(2N-3k-1)} \sum_{x=1}^{N} \sum_{y \in U^k(\mathbf{x})} (r(x,y) - k)$.

Continuity (S_{Cont}): With values in $[0,1]$, with 1 being the best, this measures how many missing neighbors [MCMT14] a projected point has. Following the above notations, let $V^k(\mathbf{x})$ be the points that are in $NN^k(\mathbf{x})$ but do not project in $NN^k(P(\mathbf{x}))$. Let also $\hat{r}(\mathbf{x},\mathbf{y})$ be the rank of $\mathbf{y}$ in the ordered set of neighbors $NN^k(\mathbf{x})$. Continuity is then defined as $1 - \frac{2}{Nk(2N-3k-1)} \sum_{x=1}^{N} \sum_{y \in V^k(\mathbf{x})} (\hat{r}(x,y) - k)$.

In contrast to [EMK*19], we compute neighborhood preservation, trustworthiness, and continuity for multiple (20) neighborhood sizes equally spread between $k = 1\%$ and $k = 20\%$ of the point count N. Similarly, for the neighborhood hit, we use 20 values for k, ranging from 0.25% to 5%. This allows us next to study the spatial quality of projections at different scales [MMT15].

Normalized stress (S_{Stress}): With values in $\mathbb{R}^+$, lower meaning better distance preservation, stress measures the pairwise difference of distances of points in nD and qD. We define S_{Stress} as $\sum_{ij} \left(d_{ij}^t - \overline{d_{ij}^t} \right)^2 / \sum_{ij} (d_{ij}^t)^2$, where d_{ij}^t and $\overline{d_{ij}^t}$ are the Euclidean distances between data points $\mathbf{x}_i^t$ and $\mathbf{x}_j^t$, and between their projections $P(\mathbf{x}_i^t)$ and $P(\mathbf{x}_j^t)$, respectively, for $1 \leq t \leq T$, for every point pair (i,j). To ease analysis, we scale distances using standardization.

Shepard diagram metrics: The Shepard diagram is a scatterplot of d_{ij} by $\overline{d_{ij}}$, for every pair (i,j) in $\mathbf{D}$ (see Fig. 3b). It visually tells how different ranges of distances between points are affected by a projection. Plots close to a diagonal indicates good distance preservation. Deviations from this highlight patterns such as poor preservation of long/short distances, creation of false neighborhoods, or stretching and compression of the manifold on which the data is defined [JCC*11]. We summarize and quantify Shepard diagrams by measuring the relationship between the two distances. Following [EMK*19], we use Pearson correlation to measure the linearity of the relationship, and we add Spearman and Kendall correlation to measure the monotonicity of the relationship. The three resulting correlation metrics $S_{Pearson}, S_{Spearman}, S_{Kendall}$ range from -1 to 1, where 1 means perfect positive correlation.

3.3.2. Temporal stability metrics

As previously stated, there are no metrics in the literature specially designed to measure the temporal stability of DR methods. We next propose two such metrics, as follows. The two variables whose relationship we want to measure are the *change of the attributes* of a sample $\mathbf{x}$ from time t to $t+1$, measured as the nD Euclidean distance $\delta^t = \|\mathbf{x}^t - \mathbf{x}^{t+1}\|$, and *movement of the projection point* $P(\mathbf{x})$ from time t to $t+1$, measured as the 2D Euclidean distance $\overline{\delta^t} = \|P(\mathbf{x}^t) - P(\mathbf{x}^{t+1})\|$. Ideally, for a temporally stable P, we want $\overline{\delta^t}$ to be proportional to δ^t. However, this may be a too hard constraint for P to satisfy, just as perfect nD to 2D distance preservation is hard to achieve for static projections. A more relaxed requirement for a temporally stable P is to have $\overline{\delta^t}$ a monotonic increasing function of δ^t. Indeed, if this constraint were not obeyed by P, then

if an observation $\mathbf{x}^t$ changes only slightly over time, its projection $P(\mathbf{x}^t)$ could move a lot. That is, if $\delta^t \ll \overline{\delta^t}$, the projection P is unstable, and would convey the user the wrong impression that data is changing a lot. Conversely, if $\mathbf{x}^t$ strongly changes over time, but $P(\mathbf{x}^t)$ remains roughly static, *i.e.* if $\delta_i^t \gg \overline{\delta_i^t}$, then the user gets the wrong impression that the data is not changing. Hence, for a temporally stable P, the two changes $\overline{\delta^t}$ and δ^t should be positively correlated.

To measure the relationship of δ^t and $\overline{\delta^t}$, we adapt the static spatial quality metrics introduced in Sec. 3.3.1 as follows:

Normalized temporal stress (T_{Stress}): We define temporal stress as $\sum_{it}(\delta_i^t - \overline{\delta}_i^t)^2/(\delta_i^t)^2$, where the subscript i indicates sample point $\mathbf{x}_i$. As for the spatial normalized stress, we normalize distances using standardization. Low stress values indicate that the 2D changes $\overline{\delta^t}$ reflect closely their nD counterparts δ^t, which is desirable.

Temporal Shepard diagram metrics: Akin to the spatial metrics defined on Shepard diagrams, we measure the Pearson, Spearman, and Kendall correlations $T_{Pearson}, T_{Spearman}, T_{Kendall}$ between δ and $\overline{\delta}$ for every observation and consecutive timesteps. High correlation values indicate that the 2D changes $\overline{\delta^t}$ are strongly correlated with their nD counterparts δ^t, which is desirable.

4. Evaluation and Results

We evaluate the 12 quality metrics introduced in Sec. 3.3 on all (dataset, method) pairs formed by the selected 9 DR methods and 10 datasets, and analyze next the results. We do this by proposing several metric visualizations, from highly aggregated (to help forming first insights) to detailed (to examine more subtle points). For a direct impression, see also the videos showing the actual dynamic projections in action, available online at [VGdS*19].

4.1. Aggregated results

Figure 1 shows average metric values computed over all datasets and techniques. Light colors represent high metric values (preferred). The colormap in Fig. 1 was normalized independently by the min and max of each column (metric), and it was inverted for the stress-based metrics, as low values mean preferred results for these. At the bottom of each cell, a 1D scatterplot with density mapped to luminance shows the distribution of the values of the (metric, method) pair corresponding to that cell over all datasets. The red line shows the distribution mean. The table in Fig. 1 is divided into three blocks: The two left blocks show spatial metrics for distance and neighborhood preservation, respectively. The right block shows stability metrics.

Figure 1 helps us to find methods that strike a balance between spatial quality and stability. In this sense, (variational) autoencoders and G-PCA score, overall, the best. The other methods are good in one aspect but not the other: Timeframe t-SNE has high neighborhood metric values but poor distance preservation and the poorest stability from all assessed methods. Timeframe PCA has high distance preservation but relatively low stability. dt-SNE appears to be as good spatially as G-t-SNE, but slightly less stable. This is an interesting finding since dt-SNE was explicitly designed (but not quantitatively assessed) to aid stability.

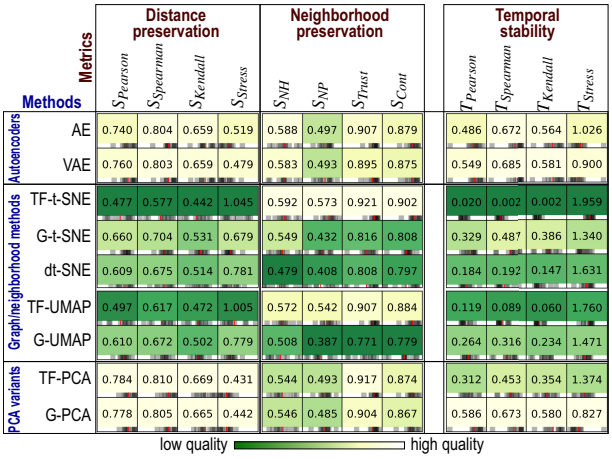

Figure 1: *Aggregated metric results over all datasets.*

4.2. Dataset-wise results

Figure 1 is simple to read but heavily aggregated, so it does not show how the quality of specific methods depends on specific *datasets*. To see this, Fig. 2 shows all metric results for all datasets without aggregation. As in Fig. 1, light colors mean good results. Columns are now not normalized. Column groups (a-f) represent spatial metrics, and columns (g-h) represent stability metrics. We use different quantitative colormaps to indicate different types of measured data. By examining Fig. 2, we obtain the following insights:

Unstable methods: TF-t-SNE is always unstable regardless of the dataset. This refines the instability finding over TF-t-SNE (Sec. 4.1) by showing that this occurs irrespective of the dataset. Also, it confirms the same observation in [RFT16], which, however, was not quantitatively confirmed there. The reason for this instability is the stochastic nature of t-SNE, which strongly manifests itself if we run the method from scratch on every new revision (timeframe). We could attribute the instability of TF-UMAP to the same reason.

Poor spatial quality: G-t-SNE and G-UMAP score poorly on distance and neighborhood preservation on most datasets. This is the aforementioned difficulty (Sec. 3.1) of constructing a *single* projection covering many samples in many timeframes. This is much harder than constructing a projection that preserves only neighborhoods formed by points in a *single* timeframe. We see here again the trade-off between spatial quality and stability.

Neighborhood preservation: Here we see dataset-specific behavior: For *gaussians*, S_{NP}, S_{Trust}, and S_{Cont} peak at a neighborhood size of roughly 10% of the dataset size. This makes sense since this is the size of the clusters present in this dataset – when k exceeds this value, the metrics will start considering points in other clusters, thus decrease. More interestingly, we see some outliers (dark bands in the heat-colormapped plots). These are techniques that score poorly for any k value. Among these, we find G-t-SNE, dt-SNE, and G-UMAP. At the other extreme, TF-t-SNE and TF-UMAP score the best results at neighborhood preservation, followed by AE, VAE, G-PCA, and TF-PCA.

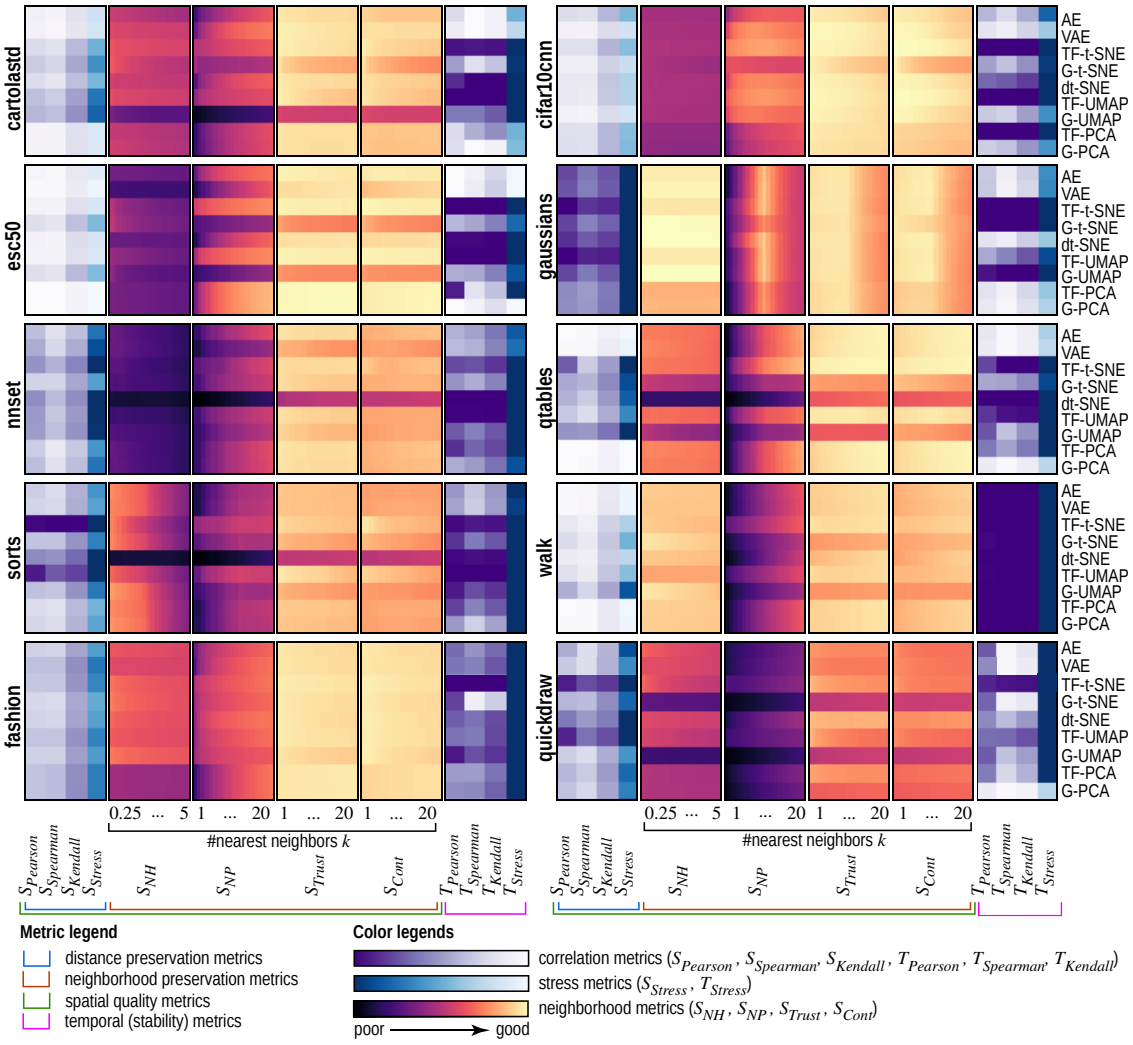

Figure 2: *Twelve spatial quality and temporal stability metrics evaluated for 9 DR methods run on ten datasets.*

Dynamic t-SNE: In contrast to the good results qualitatively observed on the single *gaussians* dataset showed in [RFT16], dt-SNE performs less well in both spatial quality and stability for several other of the considered datasets, being quality-wise somewhere between TF-t-SNE and G-t-SNE for all considered metrics.

Dataset difficulty: Some datasets are considerably harder to project with good quality than others, no matter which technique we use. For example, *walk* has poor stability for all techniques. In contrast, *gaussians* has good stability for all techniques (except the t-SNE and UMAP variants) and good neighborhood preservation for all techniques. To study how dataset characteristics influence quality, we compute the correlation of the distance-preservation, neighborhood, and temporal stability metrics (measured over all techniques) with the six traits that we used to characterize our datasets (Tab. 2). Table 3 shows the results. A few things stand out: As the number of samples N increases, the difficulty to preserve distances also increases, but neighborhoods are preserved better. Conversely, as sparsity σ_n increases, it becomes harder to preserve neighborhoods. Separately, we do not find any strong (positive or negative) correlation of temporal stability with any of the traits. Overall, this suggests that the traits are useful in predicting *spatial* quality of projections. However, we need additional traits that capture the data dynamics to reason about the projections' temporal stability.

Table 3: *Correlation between metric types and dataset traits.*

	samples N	timesteps T	dimensions n	classes	intrinsic dim. ρ_n	sparsity ratio σ_n
distance preservation	-0.429566	0.145921	-0.076177	-0.285476	-0.007806	-0.211705
neighborhood preservation	0.385248	-0.380503	-0.298868	0.243835	0.172121	-0.404517
temporal stability	0.150231	0.012017	-0.009754	0.275271	-0.085292	0.160295

4.3. Fine-grained analysis

While Fig. 2 shows all computed metrics for each (dataset, method) combination, metric values are still aggregated to a single scalar per combination. This does not show how metrics vary over the *extent*

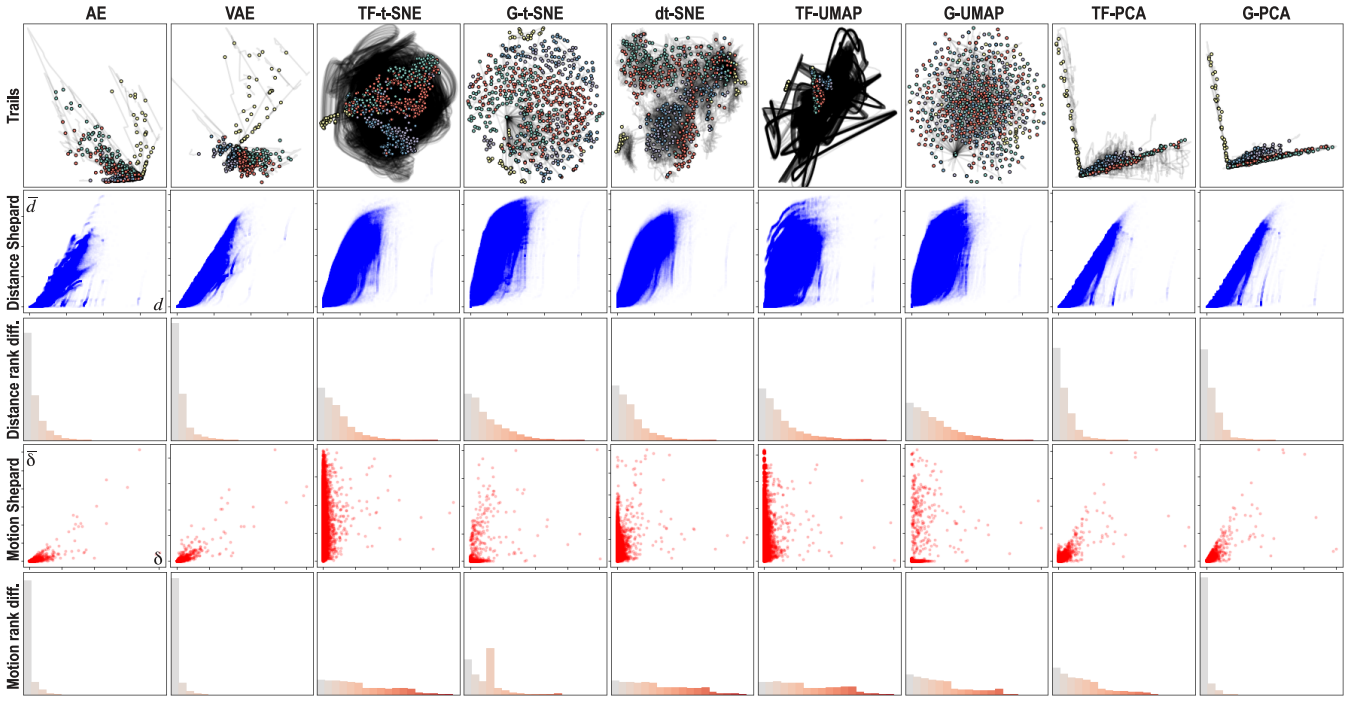

Figure 3: *Detailed analysis of distances and movements produced by all DR techniques on the* cartolastd *dataset.*

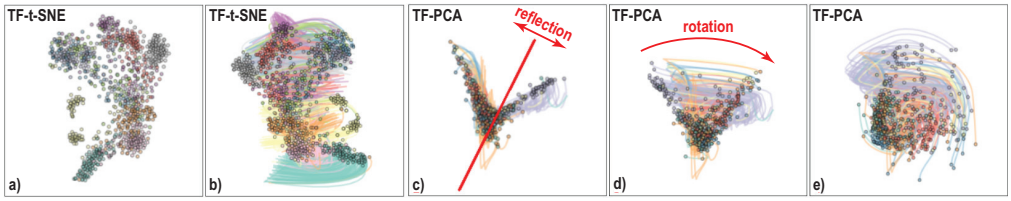

Figure 4: *Examples of instability in TF-t-SNE (a,b) and TF-PCA (c,d,e).*

of a projection and/or over *time*. There are more patterns in dynamic projections than we can capture by a set of metrics, no matter how good these are. To get such insights, we next present a fine-grained analysis that aggregates the metrics even less (see Figure 3) for a single dataset (*cartolastd*, chosen as it is alphabetically the first in our benchmark). Similar visualizations for all other datasets in the benchmark are available online [VGdS*19]. We next analyze these methods for this dataset from several perspectives, as follows.

Stability visual assessment: Figure 3a shows the actual dynamic projections with point trails $(P(\mathbf{x}_i^1),\ldots,P(\mathbf{x}_i^T))$, one per player i. Colors map the players' labels. This visualization already says a lot about the behavior and similarities of the studied DR methods (see also the submitted videos). The instability of TF-t-SNE and TF-UMAP becomes apparent, as their trails cover a very large area in the projection space. However, these methods achieve a quite good separation of same-label clusters. In contrast, dt-SNE shows trails that depict much local movement. Both PCA variants show relatively little movement, with points oscillating along two main axes, which are the main eigenvectors computed by the methods. At the other extreme, AE, VAE, and G-t-SNE show the least motion. However, this does not imply by itself a high quality: G-t-SNE, for instance, achieves indeed a better visual spreading of samples in the available projection space, but it has very poor neighborhood preservation (see G-t-SNE results in Fig. 2) and, as already discussed above, it also has very poor stability.

Distance preservation: Figure 3b shows the Shepard diagram of distances, which is a scatterplot of d_{ij} by $\overline{d_{ij}}$, for every pair (i,j) in $\mathbf{D}$, that helps us understand the distance preservation aspect of each technique. We see that the AE and PCA variants have overall better distance preservation (plots closer to the diagonal) than the t-SNE/UMAP variants. Also, we see that AE and PCA typically *compress* nD distances to 2D (points mainly under the main diagonal), whereas the t-SNE/UMAP variants both compress and stretch these (points are located both under and above this diagonal).

Inspired by the Spearman and Kendall correlations, we consider next the agreement of *ranks* instead of aggregating it to a single value. Figure 3c shows this, for distance preservation, by a histogram of the *absolute* rank differences of nD and 2D distances between point pairs. In a projection with $S_{Spearman} = S_{Kendall} = 1$,

such differences would be minimized, *i.e.*, the k^{th} largest 2D distance $\overline{d_{ij}}$ should correspond to the k^{th} largest nD distance d_{ij} for every point pair (i, j). In this case, all rank differences are zero, which would yield a histogram showing a single high bar at zero (left of the histogram). Significant rank differences spread the histogram to the right, showing poor monotonicity between the two variable ranks. From these plots, we see, again, that AE and VAE score the best, followed by G-PCA, TF-PCA, and then the t-SNE and UMAP variants.

Stability metrics: Figure 3d shows Shepard diagrams for the point movements, *i.e.*, scatterplots of δ by $\overline{\delta}$ for every sample compared to itself in the next timestep, for all timesteps. Note that, in these scatterplots, every point is a *sample*, whereas in the classical Shepard diagrams (Fig. 3b), every point is a *pair* of samples. Ideally, we want δ to be positively correlated to $\overline{\delta}$, which means a plot close to the main diagonal. The AE and PCA variants show the closest plots to the main diagonal, thus, best stability. At the other extreme, TF-t-SNE shows widely varying 2D change for similar nD change, thus, high instability. Finally, Figure 3e shows the absolute rank difference histograms for change. Their interpretation follows the one for the distance-preservation histograms (Fig. 3c): Left peaked histograms indicate high stability, whereas flatter ones indicate a discrepancy in 2D *vs* nD changes. These histograms strengthen the insights obtained so far, making it even clearer that the AE and G-PCA methods are far stabler than the t-SNE, UMAP and TF-PCA.

5. Understanding dynamic projection behavior

The coarse-grained and fine-grained analyses presented so far highlighted that there are significant differences in the behavior of dynamic DR methods that depend on both the method and the dataset. In this process, we also saw that visual quality and stability seem to be, in general, mutually competing for concerns – methods that are good in one are not the best in the other. We further explore these observations as follows. First, we analyze the causes of the observed (lack of) stability and link these to the way the studied DR techniques operate (Sec. 5.1). Next, we summarize all our findings and propose a workflow to assist the practitioner in selecting a suitable DR technique for projecting dynamic data (Sec. 5.2).

5.1. Analysis of (un)stable behavior

Beside empirically measuring and observing that different DR techniques have widely different stabilities, it is useful to analyze the *causes* of these differences, which we do next.

t-SNE and UMAP: Our results tell that TF-t-SNE and TF-UMAP, that is, projections computed independently for each timestep, are the most unstable of the assessed techniques. This is so since these are stochastic methods that optimize non-convex objective functions using randomly seeded gradient descent. Hence, different runs with the same data can create projections where different clusters might be formed and/or placed at different 2D positions. Figure 4a,b shows the last scenario. From timesteps 1 to 2 of the TF-t-SNE run of the *fashion* dataset, even though the local structure remains the same, the absolute position of the points and clusters changes drastically. In conclusion, using t-SNE/UMAP independently per timeframe is definitely not a good option for dynamic data.

dt-SNE: We encountered several cases where dt-SNE seems to have trouble optimizing its objective function – for details, see the videos for *qtables* and *sorts*. In both these cases, dt-SNE did not capture any of the spatial structures present in the data, nor produced any sensible movement. These visual findings can be confirmed by the dark lines (low-quality values) in Fig. 2. We also noticed that dt-SNE is very sensitive to the choice of hyperparameters. Concluding, whereas the initial findings in [RFT16], obtained on a single dataset (*gaussians*) position this technique as a good option for projecting dynamic data, our additional findings raise questions about the practical value of this technique.

PCA: We also see instability in TF-PCA, but for different reasons than the ones discussed above. Specifically, if there is a change in rank of the top two eigenvectors from timestep t to the next one, *i.e.*, one of the associated eigenvalues becomes larger than the other, the projection exhibits an artifact that resembles a *reflection* – see the *quickdraw* dataset in the two timesteps in Fig. 4b,c. Alternatively, if the data changes sufficiently for the eigenvectors to change considerably, the projection shows a *rotation*-like artifact – see the two timesteps in Fig. 4d,e. In contrast to t-SNE and UMAP, these artifacts are not due to stochastic seeding, but due to the way PCA works. Given the above, it is now clear why G-PCA is very stable – it chooses the two largest-variation axes for the *entire* dataset (all timesteps). The price to pay for this stability is that G-PCA may not yield the axes that best describe the data variation at each timestep, thus not the best spatial quality.

Autoencoders: Similarly to G-PCA, these techniques are stable since they train with the entire dataset (all timesteps) to learn a latent representation that encodes the global data distribution. Once trained, the encoder is a deterministic function that maps *nD* data to 2D. The main disadvantage of autoencoders over G-PCA is usability: PCA is simple to implement and use. Autoencoders, in contrast, have the 'usual' deep learning challenges, most notably finding the optimal network architecture and hyperparameter values.

5.2. Finding similarly behaving techniques

Figure 1 showed a high-level aggregated view of the quality metrics of the studied DR techniques, outlining that the autoencoders and PCA variants score better, in general, on both spatial quality and stability, than graph neighborhood techniques (t-SNE, dt-SNE, and UMAP). However, that image (and related analysis) was too aggregated. At the other extreme, Fig. 2 and related discussion showed a fine-grained analysis of all metrics measured for all techniques run on all datasets. From both these analyses, it is quite hard to understand how (and when) different techniques behave similarly. This is arguably important for practitioners interested in choosing a technique in a given context (dataset type and metrics to maximize).

Figure 5 supports this similarity analysis, as follows. Each point is here a technique run on a dataset, attributed by the computed 12 quality metrics. We project these points to 2D using UMAP, thus, creating a 'projection of projections' map. The four images in Fig. 5 use different visual codings to reveal several insights, as follows. Image (a) shows the techniques and datasets, coded by glyph, respectively categorical colors. Points in this plot are clustered more due to *datasets* than *techniques* – that is, quality is more

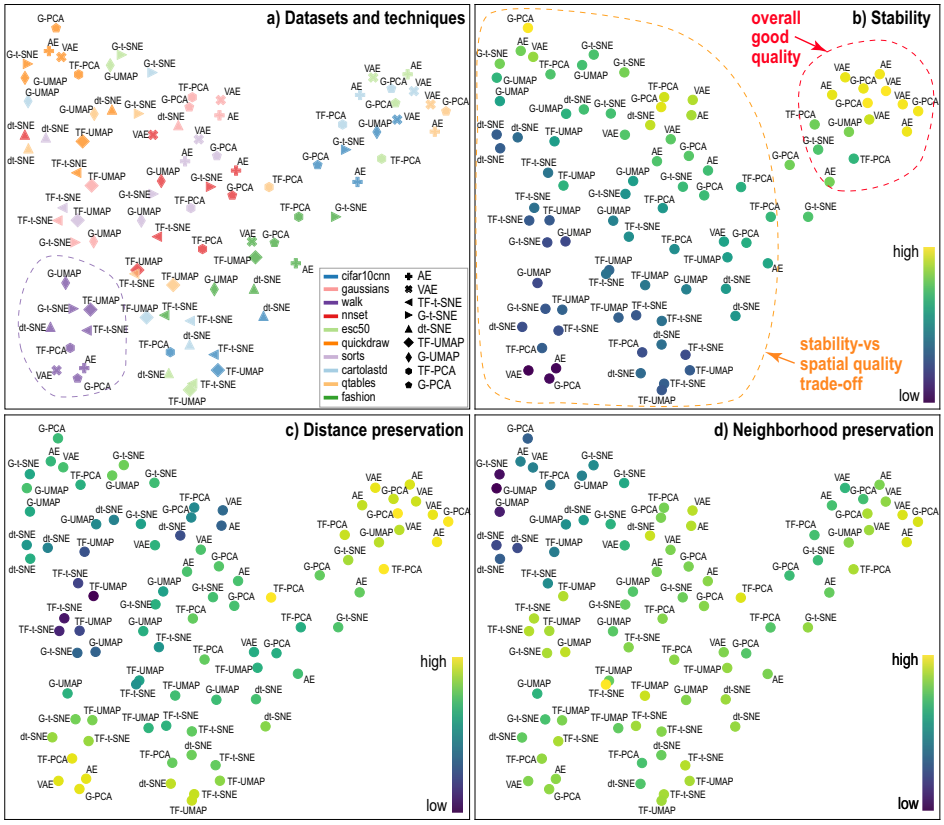

Figure 5: *Projection of projections map showing the similarity of all evaluated techniques on all datasets (Sec. 5.2).*

driven by the dataset nature than by which projection technique is used. For instance, we see the *sorts* dataset well-separated as the purple cluster bottom-left in Fig. 5a. Images (b-d) show the same projection, colored by stability, distance preservation, and neighborhood preservation, respectively. The left part of the projection (orange dashed line, Fig. 5b) shows cases where stability and distance (and/or neighborhood) preservation are mutually complementary, *i.e.*, when we obtain high stability, we get low distance/neighborhood preservation and conversely. The top-right part of the projection (red dashed line, Fig. 5b) shows cases where both stability and spatial quality are quite high. All these cases use the AE, VAE, and G-PCA techniques. The central area of the projection is covered mainly by t-SNE, dt-SNE and UMAP, telling that these projections have average behavior (as compared to autoencoders and PCA variants). Looking at the color-coded plots (images b-d), we see that these projections do not score highest on any of the considered metrics.

The plots in Fig. 5 can guide choosing a DR technique to project dynamic data: Given a dataset $\mathbf{D}$ to project, (1) find the most similar dataset $\mathbf{D}'$ in the benchmark, *i.e.*, that contains data of similar nature (*e.g.*, natural images, sounds) and is obtained via a similar acquisition process; (2) decide what is important for the dynamic projection of $\mathbf{D}$ – stability, distance preservation, neighborhood preservation, or a mix of them; (3) find the projection techniques P in the respective quality plots that have the desired qualities on $\mathbf{D}'$, and possibly also consider other projection techniques that behave similarly (close points in the plots). These techniques P are then good candidates to project $\mathbf{D}$ with.

6. Conclusion

This paper is an initial step towards understanding the behavior of dimensionality reduction techniques in the context of dynamic/temporal data. We hope that the information and results presented here help practitioners who want to understand their complex data and that this work can be used by authors interested in developing DR techniques as a tool for evaluation and comparison. We proposed a publicly available benchmark with 9 methods, 10 datasets, and 12 quality metrics. To evaluate the viability of different techniques for the task, we computed spatial and temporal stability metrics for all possible combinations, thus providing an extensive collection of results. Based on the results, we presented a discussion that elaborates on the causes for understanding the dynamic behavior. All our experiments are documented and detailed online [VGdS*19] to allow further analysis and reproducibility.

There are many ways this work can be extended in the future. The benchmark can be extended with new methods, a better way to choose hyperparameters, new datasets, and new metrics. With a larger number of datasets, we can perform a robust test of the impact of dataset traits on the metrics. We can also integrate streaming data techniques, datasets, and tests.

7. Acknowledgements

This study was financed in part by CAPES (Finance Code 001) and CNPq (Process 304336/2019-0).

References

[AEM11] ALBUQUERQUE G., EISEMANN M., MAGNOR M.: Perception-based visual quality measures. In *Proc. IEEE VAST* (2011), pp. 11–18. 2

[AJXW19] ALI M., JONES M. W., XIE X., WILLIAMS M.: TimeCluster: dimension reduction applied to temporal data for visual analytics. *Visual Computer 35*, 6-8 (2019), 1013–1026. 3

[AMM*08] AIGNER W., MIKSCH S., MÜLLER W., SCHUMANN H., TOMINSKI C.: Visual methods for analyzing time-oriented data. *IEEE TVCG 14*, 1 (2008), 47–60. 1

[APP11] ARCHAMBAULT D., PURCHASE H., PINAUD B.: Animation, small multiples, and the effect of mental map preservation in dynamic graphs. *IEEE TVCG 17*, 4 (2011), 539–552. 1

[AS16] AUPETIT M., SEDLMAIR M.: SepMe: 2002 new visual separation measures. In *Proc. IEEE PacificVis* (2016). 2

[Aup07] AUPETIT M.: Visualizing distortions and recovering topology in continuous projection techniques. *Neurocomputing 10*, 7–9 (2007), 1304–1330. 2

[Bal87] BALLARD D. H.: Modular learning in neural networks. *AAAI* (1987), 279–284. 3

[BBH12] BUNTE K., BIEHL M., HAMMER B.: A general framework for dimensionality reducing data visualization mapping. *Neural Computation 24*, 3 (2012), 771–804. 2

[BCS96a] BECKER R. A., CLEVELAND W. S., SHYU M.-J.: The visual design and control of trellis display. *JCGS 5*, 2 (1996), 123–155. 2

[BCS96b] BUJA A., COOK D., SWAYNE D. F.: Interactive high-dimensional data visualization. *JCGS 5*, 1 (1996), 78–99. 2

[BFHL17] BOYTSOV A., FOUQUET F., HARTMANN T., LETRAON Y.: Visualizing and exploring dynamic high-dimensional datasets with LION-tSNE. 46. URL: http://arxiv.org/abs/1708.04983, arXiv:1708.04983. 3

[BLIC19] BREHMER M., LEE B., ISENBERG P., CHOE E. K.: A comparative evaluation of animation and small multiples for trend visualization on mobile phones. *IEEE TVCG* (2019). 1

[BMH*19] BECHT E., MCINNES L., HEALY J., DUTERTRE C.-A., KWOK I. W. H., NG L. G., GINHOUX F., NEWELL E. W.: Dimensionality reduction for visualizing single-cell data using umap. *Nature Biotechnology 37*, 1 (2019), 38–44. 3

[BSL*08] BUJA A., SWAYNE D. F., LITTMAN M. L., DEAN N., HOFMANN H., CHEN L.: Data visualization with multidimensional scaling. *JCGS 17*, 2 (2008), 444–472. 3

[BWS*12] BERNARD J., WILHELM N., SCHERER M., MAY T., SCHRECK T.: Timeseriespaths: Projection-based explorative analysis of multivarate time series data. *WSCG 20*, 2 (2012), 97–106. 1, 3

[C*15] CHOLLET F., ET AL.: Keras. https://keras.io, 2015. 4

[CG15] CUNNINGHAM J., GHAHRAMANI Z.: Linear dimensionality reduction: Survey, insights, and generalizations. *JMLR 16* (2015), 2859–2900. 2

[EHH12] ENGEL D., HÜTTENBERGER L., HAMANN B.: A survey of dimension reduction methods for high-dimensional data analysis and visualization. In *Proc. IRTG Workshop* (2012), vol. 27, pp. 135–149. 2

[EHT19] ESPADOTO M., HIRATA N., TELEA A.: Deep Learning Multidimensional Projections. URL: http://arxiv.org/abs/1902.07958v1. 3

[EMK*19] ESPADOTO M., MARTINS R., KERREN A., HIRATA N., TELEA A.: Towards a quantitative survey of dimension reduction techniques. *IEEE TVCG* (2019), 1–1. 1, 2, 3, 4, 5

[FCS*19] FUJIWARA T., CHOU J., SHILPIKA S., XU P., REN L., MA K.: An incremental dimensionality reduction method for visualizing streaming multidimensional data. *IEEE TVCG* (2019), 1–1. 1, 2, 3

[Fod02] FODOR I. K.: A survey of dimension reduction techniques. *US Dept. of Energy, Lawrence Livermore National Labs* (2002). Tech. report UCRL-ID-148494. 2

[GF19] GRILLENZONI C., FORNACIARI M.: On-line peak detection in medical time series with adaptive regression methods. *Econometrics and Statistics 10* (2019), 134 – 150. 1

[GfVLD13] GARCÍA-FERNÁNDEZ F. J., VERLEYSEN M., LEE J. A., DÍAZ I.: Stability comparison of dimensionality reduction techniques attending to data and parameter variations. *EuroVis* (2013), 2–6. 3

[GG19] GOMIDE H., GUALBERTO A.: caRtola, 2019. https://github.com/henriquepgomide/caRtola. 4

[GH15] GISBRECHT A., HAMMER B.: Data visualization by nonlinear dimensionality reduction. *WIREs Data Mining Knowledge Discovery 5* (2015), 51–73. 2

[HG02] HOFFMAN P., GRINSTEIN G.: A survey of visualizations for high-dimensional data mining. *Information Visualization in Data Mining and Knowledge Discovery 104* (2002), 47–82. 2

[HS06] HINTON G. E., SALAKHUTDINOV R. R.: Reducing the dimensionality of data with neural networks. *Science 313*, 5786 (2006), 504–507. 3

[HWX*10] HU Y., WU S., XIA S., FU J., CHEN W.: Motion track: Visualizing variations of human motion data. *Proc. IEEE PacificVis* (2010), 153–160. 1, 2

[ID90] INSELBERG A., DIMSDALE B.: Parallel coordinates: A tool for visualizing multi-dimensional geometry. In *Proc. VIS* (1990), pp. 361–378. 2

[JCC*11] JOIA P., COIMBRA D., CUMINATO J. A., PAULOVICH F. V., NONATO L. G.: Local affine multidimensional projection. *IEEE TVCG 17*, 12 (2011), 2563–2571. 2, 5

[JFSK16] JÄCKLE D., FISCHER F., SCHRECK T., KEIM D. A.: Temporal MDS plots for analysis of multivariate data. *IEEE TVCG 22*, 1 (2016), 141–150. 1, 3

[Jol86] JOLLIFFE I.: *Principal Component Analysis*. Springer Verlag, 1986. 3

[JRK*16] JONGEJAN J., ROWLEY H., KAWASHIMA T., KIM J., FOX-GIEG N.: The Quick, Draw! - A.I. Experiment. https://quickdraw.withgoogle.com/, 2016. 4

[KH13] KEHRER J., HAUSER H.: Visualization and visual analysis of multifaceted scientific data: A survey. *IEEE TVCG 19*, 3 (2013), 495–513. 2

[Kra19] KRAPL A. A.: The time-varying diversifiability of corporate foreign exchange exposure. *Journal of Corporate Finance* (2019), 101506. 1

[Kri09] KRIZHEVSKY A.: *Learning multiple layers of features from tiny images*. Tech. rep., 2009. 4

[KSH01] KOHONEN T., SCHROEDER M. R., HUANG T. S. (Eds.): *Self-Organizing Maps*, 3rd ed. Springer-Verlag, 2001. 2

[KW13] KINGMA D. P., WELLING M.: Auto-encoding variational bayes. 1–14. URL: http://arxiv.org/abs/1312.6114, arXiv:1312.6114. 4

[LA11] LESPINATS S., AUPETIT M.: CheckViz: Sanity check and topological clues for linear and nonlinear mappings. *CGF 30*, 1 (2011), 113–125. 2

[LC10] LECUN Y., CORTES C.: MNIST handwritten digit database. URL: http://yann.lecun.com/exdb/mnist/ [cited 2016-01-14 14:24:11]. 4

[LGH13] LUEKS W., GISBRECHT A., HAMMER B.: Visualizing the quality of dimensionality reduction. *Neurocomputing 112* (2013), 109–123. 2

[LMW*17] LIU S., MALJOVEC D., WANG B., BREMER P. T., PASCUCCI V.: Visualizing high-dimensional data: Advances in the past decade. *IEEE TVCG 23*, 3 (2017), 1249–1268. 1, 2

[LV09] LEE J. A., VERLEYSEN M.: Quality assessment of dimensionality reduction: Rank-based criteria. *Neurocomputing 72*, 7-9 (2009), 1431–1443. 2

[MCMT14] MARTINS R. M., COIMBRA D. B., MINGHIM R., TELEA A. C.: Visual analysis of dimensionality reduction quality for parameterized projections. *CG 41*, 1 (2014), 26–42. 2, 5

[MDL07] MAO Y., DILLON J. V., LEBANON G.: Sequential document visualization. *IEEE TVGC 13*, 6 (2007), 1208–1215. 1, 3

[MHSG18] MCINNES L., HEALY J., SAUL N., GROSSBERGER L.: UMAP: Uniform manifold approximation and projection. *JOSS 3*, 29 (2018), 861. 3, 4

[MMCS11] MASCI J., MEIER U., CIRESAN D., SCHMIDHUBER J.: Stacked convolutional auto-encoders for hierarchical feature extraction. *ICANN* (2011), 52–59. 4

[MMT15] MARTINS R. M., MINGHIM R., TELEA A. C.: Explaining neighborhood preservation for multidimensional projections. *CGVC* (2015). 2, 5

[NA19] NONATO L. G., AUPETIT M.: Multidimensional projection for visual analytics: Linking techniques with distortions, tasks, and layout enrichment. *IEEE TVCG 25*, 8 (2019), 2650–2673. 1, 2, 3

[NPTS17] NGUYEN M., PURUSHOTHAM S., TO H., SHAHABI C.: m-TSNE: A framework for visualizing high-dimensional multivariate time series. URL: http://arxiv.org/abs/1708.07942. 1, 3

[Pic15] PICZAK K. J.: ESC: Dataset for Environmental Sound Classification. In *Proc. ACM MM* (2015), pp. 1015–1018. 4

[Pöl04] PÖLZLBAUER G.: Survey and comparison of quality measures for self-organizing maps. In *Proc. Workshop on Data Analysis (WDA)* (2004), pp. 67–82. 2

[PVG*11] PEDREGOSA F., VAROQUAUX G., GRAMFORT A., MICHEL V., THIRION B., GRISEL O., BLONDEL M., PRETTENHOFER P., WEISS R., DUBOURG V., VANDERPLAS J., PASSOS A., COURNAPEAU D., BRUCHER M., PERROT M., DUCHESNAY E.: Scikit-learn: Machine learning in Python. *JMLR 12* (2011), 2825–2830. 4

[RC94] RAO R., CARD S. K.: The table lens. *Proc. CHI* (1994), 222. 2

[RFT16] RAUBER P. E., FALCÃO A. X., TELEA A. C.: Visualizing time-dependent data using dynamic t-SNE. *EuroVis* (2016). URL: https://github.com/paulorauber/thesne. 1, 2, 3, 4, 6, 9

[RFT17] RAUBER P., FALCÃO A., TELEA A.: Visualizing the hidden activity of artificial neural networks. *IEEE TVCG 23*, 1 (2017), 101–110. 3

[SA15] SEDLMAIR M., AUPETIT M.: Data-driven evaluation of visual quality measures. *CGF 34*, 3 (2015), 545–559. 2

[SMT13] SEDLMAIR M., MUNZNER T., TORY M.: Empirical guidance on scatterplot and dimension reduction technique choices. *IEEE TVCG* (2013), 2634–2643. 2

[SNLH09] SIPS M., NEUBERT B., LEWIS J., HANRAHAN P.: Selecting good views of high-dimensional data using class consistency. *CGF 28*, 3 (2009), 831–838. 2

[SSV18] SONDAG M., SPECKMANN B., VERBEEK K.: Stable treemaps via local moves. *IEEE TVCG 24*, 1 (2018), 729–738. 3, 5

[SvLB10] SCHRECK T., VON LANDESBERGER T., BREMM S.: Techniques for precision-based visual analysis of projected data. *Information Visualization 9*, 3 (2010), 181–193. 2

[SVPM14] SORZANO C., VARGAS J., PASCUAL-MONTANO A.: A survey of dimensionality reduction techniques. URL: http://arxiv.org/abs/1403.2877. 2

[TBB*10] TATU A., BAK P., BERTINI E., KEIM D., SCHNEIDEWIND J.: Visual quality metrics and human perception: An initial study on 2D projections of large multidimensional data. In *Proc. AVI* (2010), pp. 49–56. 2

[TBZVC17] TEO G., BIN ZHANG Y., VOGEL C., CHOI H.: Pecaplus: statistical analysis of time-dependent regulatory changes in dynamic single-omics and dual-omics experiments. *npj Systems Biology and Applications 4*, 1 (2017), 3. 1

[VCT19] VERNIER E., COMBA J. L., TELEA A. C.: A stable greedy insertion treemap algorithm for software evolution visualization. In *Proc. SIBGRAPI* (2019), pp. 158–165. 3, 5

[vdMH08] VAN DER MAATEN L., HINTON G.: Visualizing data using t-SNE. *JMLR 9* (2008), 2579–2605. 2

[VDMPVdH09] VAN DER MAATEN L., POSTMA E., VAN DEN HERIK J.: Dimensionality reduction: A comparative review. *JMLR 10* (2009), 66–71. 1, 2

[VGdS*19] VERNIER E., GARCIA R., DA SILVA I., COMBA J., TELEA A.: Additional resources repository, 2019. https://eduardovernier.github.io/dynamic-projections/. 2, 4, 6, 7, 10

[VTC18] VERNIER E., TELEA A. C., COMBA J.: Quantitative comparison of dynamic treemaps for software evolution visualization. In *VISSOFT 2018* (2018), pp. 96–106. 3, 5

[WG11] WARD M. O., GUO Z.: Visual exploration of time-series data with shape space projections. *CGF 30*, 3 (2011), 701–710. 1, 3

[XRV17] XIAO H., RASUL K., VOLLGRAF R.: Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. URL: https://arxiv.org/abs/1708.07747. 4

[Yin07] YIN H.: Nonlinear dimensionality reduction and data visualization: A review. *IJAC 4*, 3 (2007), 294–303. 2