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

DimenFix: a Novel Meta-Dimensionality Reduction Strategy for Feature Preservation

Q. Luo[1] and L. Christino[2] and E. Milios[1] and F. V. Paulovich[2]

[1]Dalhousie University, Canada
[2]Eindhoven University of Technology (TU/e), Netherlands

Abstract

Dimensionality Reduction (DR) methods have become essential tools for the data analysis toolbox. Typically, DR methods combine features of a multi-variate dataset to produce dimensions in a reduced space, preserving some data properties, usually pairwise distances or local neighborhoods. Preserving such properties makes DR methods attractive, but it is also one of their weaknesses. When calculating the embedded dimensions, through usually non-linear strategies, the original feature values are lost and not explicitly represented in the spatialization of the produced layouts, making it challenging to verify the features' contribution to the attained representations. Some strategies have been proposed to tackle this issue, such as coloring the DR layout or generating explanations. Still, they are post-processes, so specific features (values) are not guaranteed to be preserved or represented. This paper proposes DimenFix, a novel meta-DR strategy that explicitly preserves the values of a particular feature or external data (e.g., class, time, or ranking) in one of the embedded dimensions. DimenFix works with virtually any gradient-descent DR method and, in our results, has shown to be capable of representing features without heavily impacting distance or neighborhood preservation, allowing for creating hybrid layouts joining characteristics of scatter plots and DR methods.

CCS Concepts

• ***Mathematics of computing*** → ***Dimensionality reduction;*** • ***Computing methodologies*** → *Visual analytics;*

1. Introduction

Demand for visualizing and interpreting high-dimensional datasets has rapidly increased in recent years. One of the most popular strategies to interpret such datasets is projecting them to a lower dimensional space (usually 2D or 3D) while reproducing the relationship between pairs of instances in the data. This process is usually called Dimensionality Reduction (DR) and can be categorized into local and global methods [EMK*21,NA19]. While global methods (such as Multidimensional Scaling (MDS) [Tor52]) seek to preserve pairwise distance relationships, local ones (such as t-Distributed Stochastic Neighbor Embedding (t-SNE) [VdMH08]) focus on neighborhood preservation.

Common to most DR techniques is that the embedded dimensions are defined as combinations of the input data features. Consequently, understanding how feature values contribute to the produced layouts can be challenging since the positions of the projected data instances are typically influenced by all input features, in some cases, through non-linear combinations. Some strategies have been devised to allow for such interpretation. Features can be mapped to axes (lines) in the produced layouts to represent the influence of each feature [CMN*16]. Color can represent the values of a feature or external data (e.g., a class) [SSJ*22]. Or more advanced approaches can also be employed, for instance, contrastive [MJEG21] or feature importance [MJE21,TZv*21,TTT23] analyses to identify the features contributions for groups of instances. Although effective methods to interpret a DR layout, they are post-processes strategies, so they cannot guarantee that specific features (values) are preserved and ordered in the final layout. Therefore, the properties of usual scatter plots are not presented, and the interpretation of the produced x and y embedded axes (for a 2D layout) have no clear connection with the input features.

This paper proposes *DimenFix*, a novel meta-dimensionality reduction strategy that addresses this limitation by explicitly preserving the values of a particular feature or external data (e.g., class, time, or ranking) in one of the embedded axes. In this process, *DimenFix* maps such feature or external data into one of the embedded axes and controls the degree of freedom the embedded points' positions can change in that axis. In this paper, we discuss how to adapt Force Scheme [TMN03] to employ the *DimenFix* strategy, but it can be used in combination with virtually any gradient-descent method, such as t-SNE. Our experiments show that *DimenFix* can represent features in the final layout without heavily impacting distance or neighborhood preservation quality, even when external data, such as class, is used as the fixed axis. The code for *DimenFix* can be found on GitHub [LCP22].

DOI: 10.2312/eurova.20241110

2. Related Work

In the current literature, many different Dimensionality Reduction (DR) techniques are presented, each focusing on different aspects of the data to be preserved in the produced visual representation. A common taxonomy classifies the techniques into local and global [EMK*21, NA19]. While global methods, such as Multidimensional Scaling (MDS) [Tor52], Force Scheme [TMN03], or Part-Linear Multidimensional Projection (PLMP) [PSN10], look to preserve the overall pairwise distance between data instances, local techniques, such as t-Distributed Stochastic Neighbor Embedding (t-SNE) [VdMH08], Uniform Manifold Approximation and Projection (UMAP) [MHM18], and Least Square Projection (LSP) [PNML08], seek to preserve local neighborhoods.

Common to these techniques is the fact that they work by combining all the input m-dimensional data features to compose the final 2D layout. Different strategies have been suggested to allow for the interpretation of a DR layout, considering the contribution of input features. Mapping the input features to axes in the DR layout has been suggested [CMN*16], in which small axes represent small feature contributions, and the axis direction indicates how the related feature varies in the layout. Color is also used, where the data instances (or points in 2D) are colored using the values of a feature or class to map additional information [SSJ*22]. Recently, some more advanced strategies have been suggested to describe the importance of the features for DR layouts using Shapley values [MJE21], by contrastive analysis [MJEG21], or by detecting the most important features to describe groups [TZv*21, TTT23]. Although effective methods to interpret a DR layout, they are post-processes, so defining the input features' contribution to the layout's spatialization is still challenging. What is visible is bounded by the DR method, which can be near zero [CMN*16]. Neither are there guarantees that ordered values of a feature will follow an order in the final layout, making the interpretation of ordered features or external information, such as time or ranking (ordered categorical value), very hard or even possible to execute.

Our approach, *DimenFix*, focuses on solving this issue, allowing for the order of a feature or any external information to be represented in the final layout while still retaining the original distances and local neighborhoods of the m-dimensional space as much as possible in the produced layout.

3. Methodology

In general lines, *DimenFix* is a strategy that modifies any gradient-descent-like method, such as t-SNE [VdMH08], Force Scheme [TMN03] and UMAP [MHM18], to allow for the preservation of a given feature or external ordered information by mapping it to one of the 2D layout axes. In this section, we adapt the Force Scheme method and propose two modes for *DimenFix*: the *Strictly Fixed Mode* (Section 3.1), which does not allow a 2D point to move along the fixed axis (the feature values are completely preserved), and the *Moving-In-Range Mode* (Section 3.2) which allows a 2D point to move within a limited user-defined range.

3.1. Strictly Fixed Mode

Typically, any gradient-descent-like DR method is free to update the values of all n embedded dimensions in the optimization steps. *DimenFix* change this by allowing only $(n-1)$ dimensions to change. Before starting the gradient-descent-like process, one of the input features (or external information) is selected to be mapped to one of the embedding space coordinates. During each iteration of the gradient-descent process, the loss is calculated using all embedded dimensions, including the fixed feature. However, the movement allowed for the fixed dimension is bounded, so the amount of changes on one of the axes is explicitly controlled. Note that *DimenFix* does not add a feature as another dimension to a $(n-1)$ projection but instead takes the fixed feature into account during the gradient-descent process, which influences all free embedding coordinates.

As mentioned, *DimenFix* concept can be used with any gradient-descent-like technique. In this paper, we discuss how to adapt the Force Scheme [TMN03] technique and present results based on that. In more formal terms, let $x = (x^1, \ldots, x^m), x^i \in \mathbb{R}, 1 \le i \le m$ be a m-dimensional data instance, where $X = \{x_i\}, 1 \le i \le N$ denotes the input dataset and $x^j = (x^j_1, \ldots x^j_N)$ the j^{th} feature of X. Also, let $y = (y^1, \ldots, y^n), y^i \in \mathbb{R}, 1 \le i \le n$ be the mapping of x to the n-dimensional embedding space, with $Y = \{y_i\}, 1 \le i \le N$ the final embedding. In this paper, we adapt the original Force Scheme to optimize

$$\sum_i^N \sum_j^N \left(\delta(x_i, x_j) - d(y_i, y_j)\right)^2 \tag{1}$$

where $\delta(.,.)$ and $d(.,.)$ are distance functions on the original and embedded spaces, respectively.

In the optimization process, the initial embedding configuration is set so that the fixed dimension receives the user-selected feature, and the others are randomly defined. Without loss of generality, let x_j be the original feature we aim to preserve and that we fix the first dimension of Y. In this initialization, we set $y^1_i = x^j_i, 1 \le i \le N$ and $y^k_i = \beta, 1 \le i \le N, 2 \le k \le n$ where β is a random number in $[0,1]$ – to ensure the projection's stability before starting the gradient-descent process we suggest normalizing each dataset feature between $[0,1]$. After that, the gradient-descent process updates the $(n-1)$ (last) free dimensions. Algorithm 1 describes this process.

3.2. Moving-In-Range Mode

Different points can sometimes share the same value on the fixed axis. For example, when the assigned feature is a categorical ordinal value. In those cases, the user may allow these points to move slightly in the fixed coordinate, avoiding them overlapping (too much) on the same embedded coordinates. To meet this need, we propose another mode to *DimenFix*, the *Moving-In-Range Mode*. Under this mode, the values on the fixed axis can be changed within a range, considering two different strategies.

3.2.1. Uniform Moving-In-Range mode

In the *Uniform Moving-In-Range* mode, we allow the fixed embedding dimension to move freely within a predefined range during the gradient-decent process. In simple terms, let $[-\alpha, +\alpha]$ be the moving range. On every optimization iteration, we calculate if the difference between the original value and the value after an iteration is within such a range. The update is executed in all dimensions if the coordinates of the fixed dimension are within the range; otherwise,

Algorithm 1: Force scheme adapted to apply *DimenFix*.

Data: X: original dataset
x^j: original feature to be preserved
Δ: learning rate
max: maximum number of iterations
Result: Y: final embedding

1 Set $y_i^1 = x_i^j, y_i^k = \beta, 1 \leq i \leq N, 2 \leq k \leq n$ where β is a random number in $[0,1]$
2 **while** $it < max$ **do**
3   **for** $x_r \in X$ **do**
4   **for** $x_s \neq x_r \in X$ **do**
5       $\vec{v} = y_s - y_r$
6       $y_s = \frac{(\delta(x_r,x_s)-d(y_r,y_s))\times|\delta(x_r,x_s)-d(y_r,y_s)|}{\Delta} \times \frac{\vec{v}}{||\vec{v}||}$
7       $y_s^1 = x_s^j$
8       $it = it + 1$
9   **end**
10   **end**
11 **end**

it is only executed on the free dimensions, and the fixed dimension is forced to be within the range. Without loss of generality, let x_j be the original feature we aim to preserve and that we fix the first dimension of Y. Also, let $\bar{y}^1$ be the new calculated values for the fixed embedded dimension. The values $y_s^1, 1 \leq s \leq N$ are updated to

$$y_s^1 = \begin{cases} x_s^j - \alpha & \bar{y}_s^1 < x_s^j - \alpha \\ x_s^j + \alpha & \bar{y}_s^1 > x_s^j + \alpha \\ \bar{y}_s^1 & \text{otherwise,} \end{cases} \tag{2}$$

at every iteration of the gradient-descent process, replacing the assignment executed on line 7 of Algorithm 1.

3.2.2. Gaussian Moving-In-Range mode

The *Uniform Moving-In-Range* mode allows moving a point on the fixed axis within a certain range. However, the hard threshold may cause the movement of the points in the fixed dimension to be inconsistent when compared to the other dimensions. If the user wants better results, the *Gaussian Moving-In-Range* mode allows points to go beyond the range threshold using a Gaussian function. Under this mode, we use a Gaussian to weigh the moving force. The farther the point is from its original position (the feature we are willing to preserve), the harder it is to move. As a result, some points may slightly move out of the pre-defined range, resulting in a more consistent movement.

Let x be the difference between the original and current position of the fixed dimension, that is, $x = x^j - y^1$, where x_j is the original feature we aim to preserve, and we fix the first dimension of Y. Since when $x = 0$, the moving force should be maximum (equal to 1), the mean in the Gaussian should be $\mu = 0$, and the Gaussian weighing function can be defined as

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{x^2}{2\sigma^2}\right), \tag{3}$$

so that $f(x)$ defines a fraction of the actual moving distance along the fixed axis.

To calculate σ, consider a user-defined confidence interval $0 < CI < 1$ and moving range $[-\alpha, +\alpha]$. First, we use the z-score table to find the z-score (z) where $^{(1-CI)}/_2$ is the area on the left side of the significant interval. This z-score refers to how far a value is from the mean in a standard normal distribution, and we use it as a reference to change the shape of the Gaussian function. With z, the interval spread α, and the condition $\mu = 0$, we use the z-score equation ($z = {}^{(x-\mu)}/_{\sigma}$) to calculate σ

$$\sigma = \frac{\alpha}{z} \tag{4}$$

Knowing that when $x = 0$, $f(x) = 1$, we can adapt $f(x)$ to weight how much the distance calculated by the base method (see Algorithm 1) should be used to move a point, resulting in

$$MR = \frac{1}{\frac{1}{\sigma\sqrt{2\pi}}\exp(0)} = \sigma\sqrt{2\pi} \tag{5}$$

Using Eq. 5 and Eq. 3, and considering that $\bar{y}^1$ is the newly calculated values for the fixed dimension, we can finally define the actual moving distance d_s for the point s in each iteration as

$$d_s = \sigma\sqrt{2\pi}\frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{(x_s^j - \bar{y}_s^1)^2}{2\sigma^2}\right) = \exp\left(-\frac{(x_s^j - \bar{y}_s^1)^2}{2\sigma^2}\right) \tag{6}$$

where x^j is the feature to be preserved, and the sign of $(x_s^j - \bar{y}_s^1)$ defines the direction of the movement so that

$$y_s^1 = x_s^j + (\text{sign}(x_s^j - \bar{y}_s^1) \times d_s) \tag{7}$$

replaces the assignment executed on line 7 of Algorithm 1.

4. Results

In this section, we discuss *DimenFix* results using the Force Scheme as the base technique (as discussed in the paper). We analyze *DimenFix* qualitatively and quantitatively, comparing it to other techniques. We use four benchmark datasets from [MK24]: **Iris** (4 dimensions, 150 instances), **Wine** (13 dimensions, 178 instances), **Breast Cancer** (30 dimensions, 569 instances), and **Segmentation** (19 dimensions, 2,310 instances).

In our first test, we compare the original Force Scheme and *DimenFix* layouts in Figure 1. The first two columns present Force Scheme layouts, and the third, fourth, and fifth columns *DimenFix* layouts. To generate such layouts, we randomly pick up one of the features of the original data to execute *DimenFix* (the feature values are mapped to the y-axis) – *sepal width* for **Iris**, *non-flavanoid_phenols* for **Wine**, *worst texture* for **Breast Cancer**, and *region-centroid-row* for **Segmentation** – and to color the second and fifth columns (the brighter the color, the larger the value) of projections. By analyzing the Force Scheme projections on the second column, it is possible to notice that in most, the selected feature does not smoothly vary from small to large values on the x or y axis. The values are mixed without much of a clear tendency. On the **Iris** projection, there is some tendency inside the two visible big groups, but interestingly, the tendencies are inverted. On the left-side group, small values are on the top, while on the right-side group, they are on the bottom. So, no global overall tendency can be observed. The **Wine** projection is the only globally preserving some ordering observed on the original feature. But even in this case, there is no smooth tendency, and the values are mixed in the

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

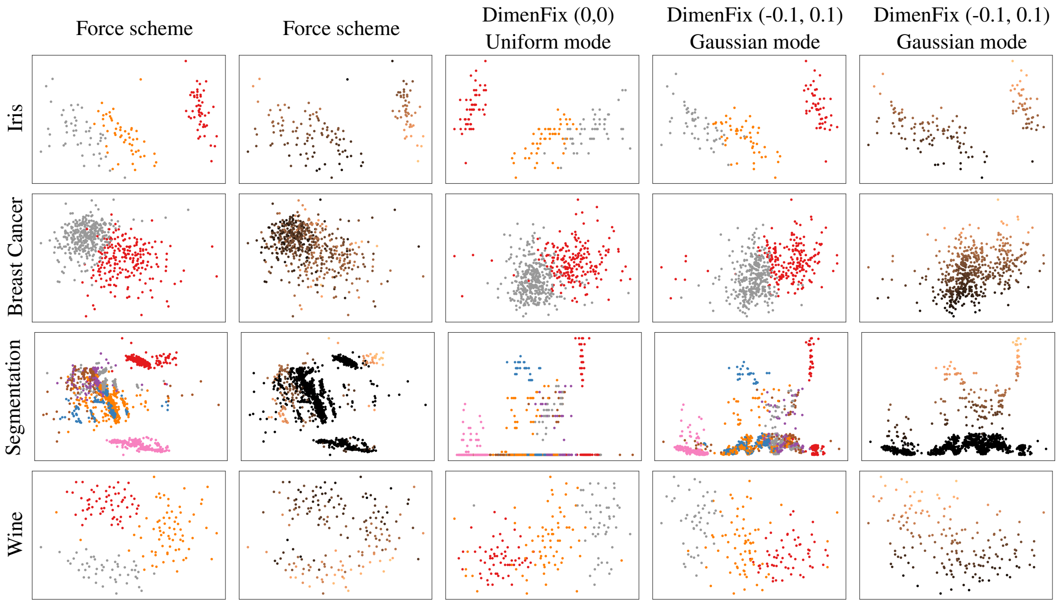

Figure 1: *Force Scheme and DimenFix projections. When particular features are mapped to color, it is possible to see that they do not smoothly vary on the Force Scheme layouts (from small to large values). Hence, reasoning about feature values and point positions is challenging. On DimenFix projections, this can be attained by fixing such features in one of the projection axes.*

middle of the layout. For the **Breast Cancer** and the **Segmentation** datasets, no tendency can be observed; small and large values are mixed, so when analyzing these layouts, not much could be said about the importance of such features to the final layout or infer the values of such features based on points positions.

The third column (*DimenFix (0,0) Uniform mode*) of Figure 1 presents *DimenFix* results using the *Uniform Moving-in-Range mode* (see Section 3.1). For the **Wine** and **Breast Cancer**, the results are consistent regarding class separation, similar to the original Force Scheme (first column). Only a few points of the orange class on the Wine dataset are out of place (regarding the class) – from the analytical point of view, this is indeed not bad since it is possible to verify what are the instances for which the selected feature has an impact in making them not separable from the instances of other classes. We can observe an interesting pattern for the **Iris** and **Segmentation** datasets. Different from the **Wine** and **Breast Cancer**, where the selected features cover a range of different values, in the **Iris** and **Segmentation**, many instances have the same values and they are defined in regular intervals (almost discrete), resulting in a "line" pattern. *DimenFix* can be executed using the *Gaussian Moving-in-Range* mode to address this issue, allowing the points to move inside a small interval. The fourth column presents the results (DimenFix (-0.1,0.1)). The "line" pattern disappears, and the overlap among instances with the same feature value is substantially reduced. Although the exact feature values are lost, the y-axis does not reflect the original value anymore, the feature tendency is maintained from small to large values (fifth column), allowing for the interpretation of the resulting layouts considering the fixed feature.

Another potential application for *DimenFix* is when the feature to fix is not part of the dataset features, such as a class. *DimenFix* results varying the moving intervals for the benchmark datasets are shown in Figure 2. Starting from a zero interval in the second column (similar to the *Uniform Move-in-Range mode*), the intervals increase in the remaining projection columns. As expected, when the interval is zero, the instances of the same class are overlapped in a "line" pattern, equally spaced between the classes. This is similar to applying a uni-dimensional Force Scheme to set one of the projection axes, with the other being the class values. In this figure, the first column presents the results of this uni-dimensional plus class projection strategy for illustration. The real benefit of *DimenFix* emerges when the moving intervals increase. Even for a small interval of $[-0.2, 0.2]$, it is already possible to see the instances of the same class occupying the spaces between the classes in the layout, increasing its overall quality (this is also quantitatively true, as we will show later). However, the classes overlap more as this interval increases. For instance, for the **Iris** dataset in the interval $[-0.9, 0.9]$, the vertical separation between classes does not exist. The classes are separated because the data allows that, not because of *DimenFix*. This also happens up to an extent to the other datasets, especially for the **Segmentation** where most of the classes mix in the center of the layout. Still, they are better separated than the original layout of Figure 1.

Lastly, a quantitative analysis is performed measuring the pairwise distance preservation using Kruskal's stress [Kru64] and the neighborhood preservation using thrusthworthiness [VPN*10]. Boxplots summarizing multiple executions of the original Force Scheme (Original), *DimenFix* fixing the features (DF/feature) or classes (DF/class), and projections executing the Force scheme to 1D and using either the features (FS/feature) or classes (FS/class) to set the second dimension are shown in Figure 3. As expected, the

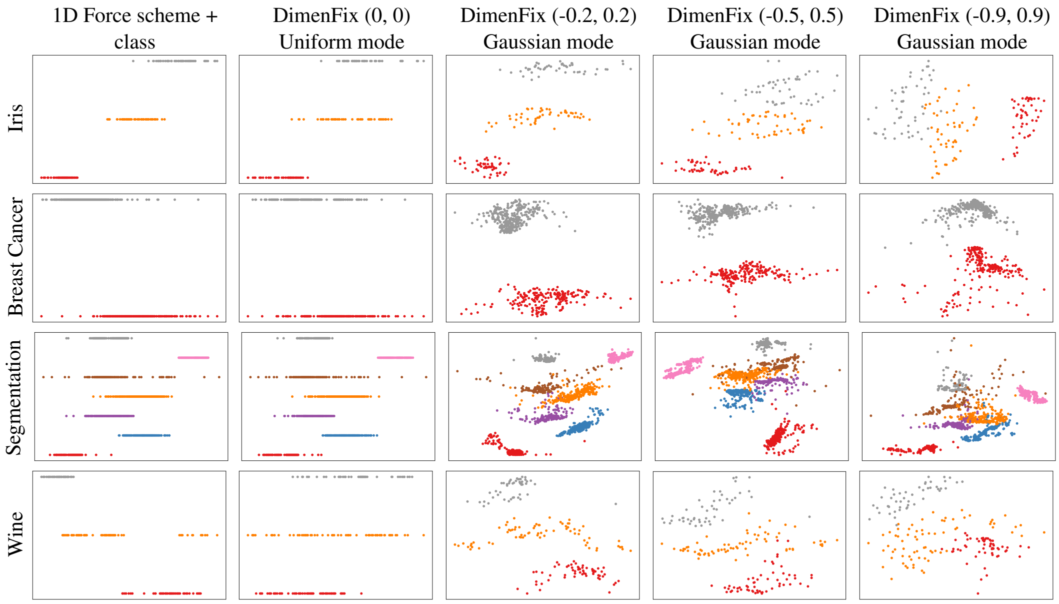

Figure 2: *DimenFix varying the moving interval and 1D Force Scheme projections fixing the class in the y-axis. DimenFix allows the points to occupy the space between the classes in the y-axis, resulting in better projections.*

original Force Scheme attained the best results for stress and trustworthiness since two dimensions are fully used to approximate the original distances (and neighborhoods). Nevertheless, the results of fixing features are not much worse for both *DimenFix* and 1D Force Scheme, indicating that the resulting layouts are somewhat comparable to the ones generated by the original technique regarding distance and neighborhood preservations. However, when fixing the class, the preservation decreases, especially the pairwise distance (stress). This is expected given the classes' discrete nature and the non-ordinal relationship between the classes in any of these datasets (class 1 is not larger than class 0). Nevertheless, notice that *DimenFix* results are always better than the 1D Force Scheme strategy, and especially when fixing the class, they are much better, indicating that taking the fixed feature/class into consideration in the optimization process (varying in a range or not) result in more precise projections that can convey relationships between features or classes and the point positions. Something not usually possible using the original Force Scheme.

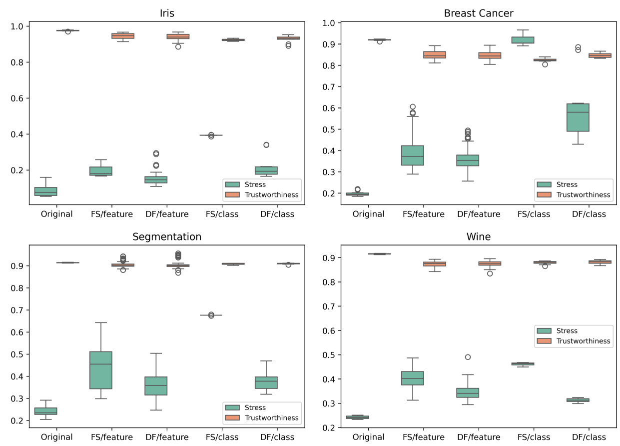

Figure 3: *Comparing DimenFix different modes with the Original Force Scheme and 1D Force Scheme strategy DimenFix allows for feature values preservation without heavily degrading the quality of the attained layouts.*

5. Conclusion and Future Work

This paper presented a novel meta-Dimensionality Reduction (DR) strategy, *DimenFix*, built upon any gradient-descent-based DR method. Unlike normal DR methods, *DimenFix* allows users to fix a dataset feature/class (or any external data, e.g., time) to an axis in the projection layout without affecting (too much) its quality so that the projection spatialization reflects the fixed values. With this hybrid strategy joining DR and scatter plot capabilities, different goals can be achieved, for instance, (a) better preserving a particular feature of the dataset while simultaneously reducing the dimensionality and (b) understanding the data distribution with respect to a specific feature specified by the user. Despite the promising results, an in-depth analysis of *DimenFix* results, a better approach to defining its hyperparameters, such as the moving range ($[-\alpha, +\alpha]$), a math-grounded strategy to compute the coordinates of the instances on the fixed dimension when unordered external data (e.g., class) is used (in our results, the order is arbitrary), and its application to other DR methods, such as t-SNE and UMAP, are necessary and left as future work.

References

[CMN*16] Coimbra D. B., Martins R. M., Neves T. T., Telea A. C., Paulovich F. V.: Explaining three-dimensional dimensionality reduction plots. *Information Visualization 15*, 2 (2016), 154–172. doi:10.1177/1473871615600010. 1, 2

[EMK*21] Espadoto M., Martins R. M., Kerren A., Hirata N. S. T., Telea A. C.: Toward a quantitative survey of dimension reduction techniques. *IEEE Transactions on Visualization and Computer Graphics 27*, 3 (2021), 2153–2173. doi:10.1109/TVCG.2019.2944182. 1, 2

[Kru64] Kruskal J. B.: Nonmetric multidimensional scaling: a numerical method. *Psychometrika 29*, 2 (1964), 115–129. 4

[LCP22] Luo Q., Christino L., Paulovich F.: visml/neo_force_scheme: Eurovis submission, Nov. 2022. URL: https://doi.org/10.5281/zenodo.7379477, doi:10.5281/zenodo.7379477. 1

[MHM18] McInnes L., Healy J., Melville J.: Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426* (2018). 2

[MJE21] Marcílio-Jr W. E., Eler D. M.: Explaining dimensionality reduction results using shapley values. *Expert Systems with Applications 178* (2021), 115020. URL: https://www.sciencedirect.com/science/article/pii/S0957417421004619, doi:https://doi.org/10.1016/j.eswa.2021.115020. 1, 2

[MJEG21] Marcílio-Jr W. E., Eler D. M., Garcia R. E.: Contrastive analysis for scatterplot-based representations of dimensionality reduction. *Computers & Graphics 101* (2021), 46–58. URL: https://www.sciencedirect.com/science/article/pii/S0097849321001783, doi:https://doi.org/10.1016/j.cag.2021.08.014. 1, 2

[MK24] Markelle Kelly Rachel Longjohn K. N.: The UCI Machine Learning Repository, 2024. URL: https://archive.ics.uci.edu. 3

[NA19] Nonato L. G., Aupetit M.: Multidimensional projection for visual analytics: Linking techniques with distortions, tasks, and layout enrichment. *IEEE Transactions on Visualization and Computer Graphics 25*, 8 (2019), 2650–2673. doi:10.1109/TVCG.2018.2846735. 1, 2

[PNML08] Paulovich F. V., Nonato L. G., Minghim R., Levkowitz H.: Least square projection: A fast high-precision multidimensional projection technique and its application to document mapping. *IEEE Transactions on Visualization and Computer Graphics 14*, 3 (2008), 564–575. doi:10.1109/TVCG.2007.70443. 2

[PSN10] Paulovich F. V., Silva C. T., Nonato L. G.: Two-phase mapping for projecting massive data sets. *IEEE Transactions on Visualization and Computer Graphics 16*, 6 (2010), 1281–1290. doi:10.1109/TVCG.2010.207. 2

[SSJ*22] Sohns J.-T., Schmitt M., Jirasek F., Hasse H., Leitte H.: Attribute-based explanation of non-linear embeddings of high-dimensional data. *IEEE Transactions on Visualization and Computer Graphics 28*, 1 (2022), 540–550. doi:10.1109/TVCG.2021.3114870. 1, 2

[TMN03] Tejada E., Minghim R., Nonato L. G.: On improved projection techniques to support visual exploration of multi-dimensional data sets. *Information Visualization 2*, 4 (2003), 218–231. 1, 2

[Tor52] Torgerson W. S.: Multidimensional scaling: I. theory and method. *Psychometrika 17* (1952), 401–419. 1, 2

[TTT23] Thijssen J., Tian Z., Telea A.: Scaling Up the Explanation of Multidimensional Projections. In *EuroVis Workshop on Visual Analytics (EuroVA)* (2023), Angelini M., El-Assady M., (Eds.), The Eurographics Association. doi:10.2312/eurova.20231098. 1, 2

[TZv*21] Tian Z., Zhai X., van Driel D., van Steenpaal G., Espadoto M., Telea A.: Using multiple attribute-based explanations of multidimensional projections to explore high-dimensional data. *Computers & Graphics 98* (2021), 93–104. URL: https://www.sciencedirect.com/science/article/pii/S0097849321000765, doi:https://doi.org/10.1016/j.cag.2021.04.034. 1, 2

[VdMH08] Van der Maaten L., Hinton G.: Visualizing data using t-sne. *Journal of machine learning research 9*, 11 (2008). 1, 2

[VPN*10] Venna J., Peltonen J., Nybo K., Aidos H., Kaski S.: Information retrieval perspective to nonlinear dimensionality reduction for data visualization. *Journal of Machine Learning Research 11*, 2 (2010). 4