Supplemental Material
ShaRP: Shape-Regularized Multidimensional Projections

Submission ID: 1011

Abstract

Projections, or dimensionality reduction methods, are techniques of choice for the visual exploration of high-dimensional data. Many such techniques exist, each one of them having a distinct visual signature — i.e. a recognizable way to arrange points in the resulting scatterplot. Such signatures are implicit consequences of algorithm design, such as whether the method focuses on local vs. global data pattern preservation; optimization techniques; and hyperparameter settings. In this work, we present a novel projection technique — ShaRP — that instead provides users explicit control over the visual signature of the created scatterplot, which can cater better to interactive visualization scenarios. ShaRP scales well with dimensionality and dataset size, generically handles any quantitative dataset, and provides this extended functionality of controlling projection shapes at a small, user-controllable cost in terms of quality metrics.

CONTENTS

A Comparison across all studied projection techniques

Fig. 1 shows a comprehensive comparison of results obtained by ShaRP (rightmost part of the image) and other techniques against which we compare. The AG, KM, and GT variants correspond to (pseudo)labels obtained by Agglomerative Clustering, K-Means, and ground truth labels, respectively.

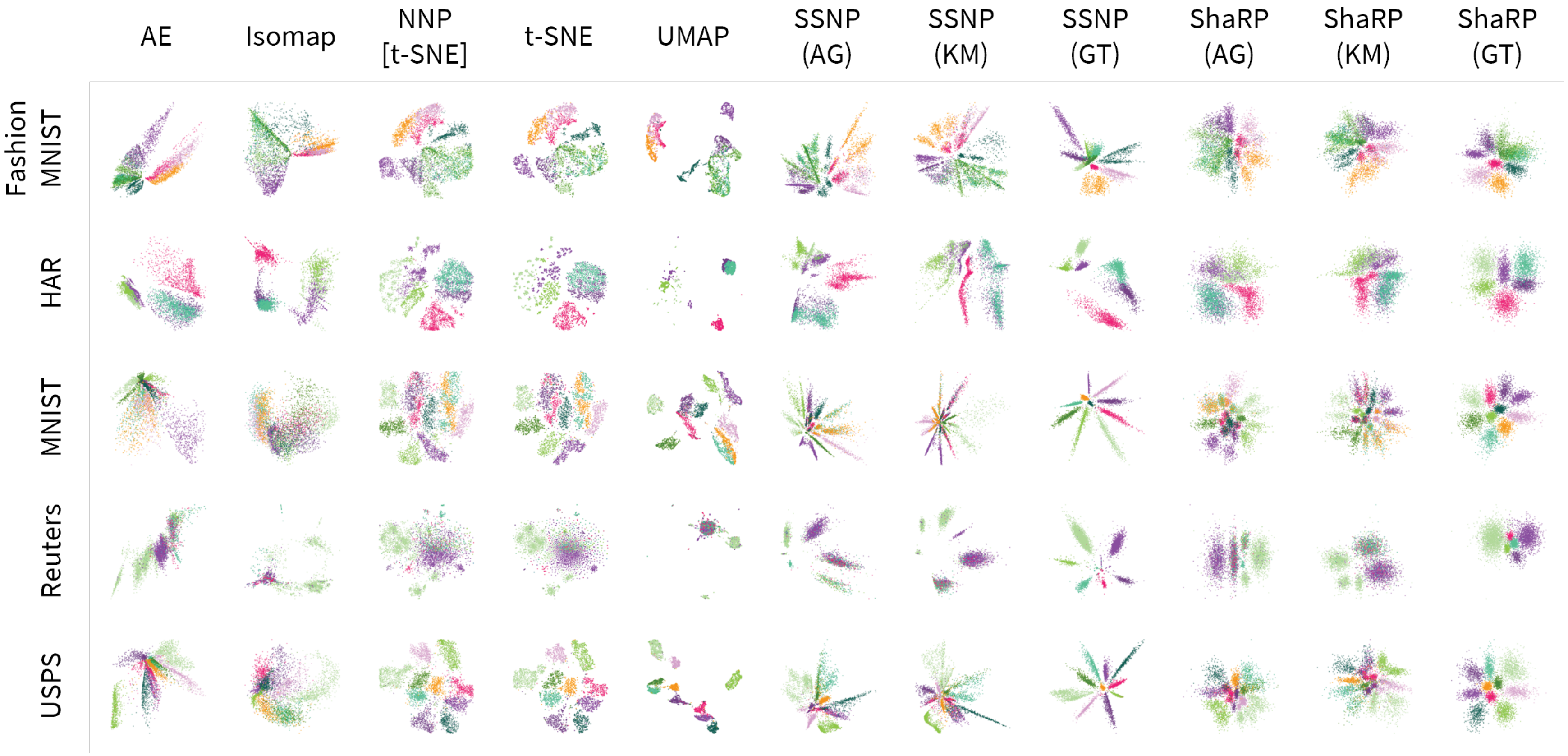

Figure 1: Comparison between all possible pairs of algorithm and dataset among those studied in this work.

B Further cluster shaping

In our work, we show ShaRP's ability to reshape clusters into rectangles/squares and triangles. Here, we show this feature over more datasets and with varying hyperaparameters.

Exploring squarified cluster generation

We are able to produce shapes that conform to squarified/rectangular shapes by employing a generalized Normal distribution in our sampling scheme. We show in Fig. 2 the results of varying the shape hyperparameter ω over different datasets.

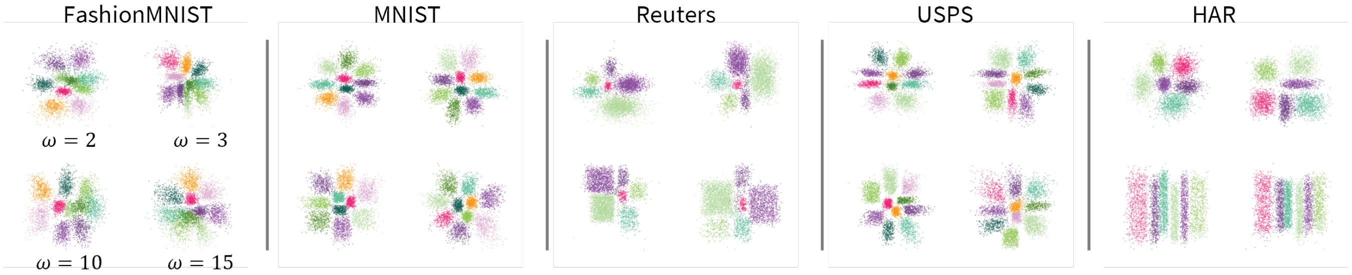

Figure 2: Demonstration of shape regularization towards rectangles for different values of ω across all datasets.

Exploring triangular cluster generation

We first define an equilateral triangle in $\mathbb{R}^2$ by arranging its vertices $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ in a matrix as

$$\mathbf{T} = [\mathbf{v}_1\ \mathbf{v}_2\ \mathbf{v}_3] = \begin{bmatrix} 0 & 1/2 & 1 \\ 0 & \sqrt{3}/2 & 0 \end{bmatrix}$$

The base shape is a source of bias, so we choose a base triangle that is symmetric around its center, i.e. equilateral. This initial bias can be overcome through the training process via a necessary extension we add to the sampling scheme.

Any convex combination of $\mathbf{v}_1, \mathbf{v}_2$, and $\mathbf{v}_3$ gives a point in the interior of this triangle – this is what using barycentric coordinates means. If we have a vector $\mathbf{w} = [w_1\, w_2\, w_3]^T$ where $w_i \in [0,1],\ i \in \{1,2,3\}$ and $\sum_{i=1}^{3} w_i = 1$, we obtain an interior point $\mathbf{p} \in \mathbb{R}^2$ by $\mathbf{p} = \mathbf{Tw}$.

Thus, we need to use a sampling distribution in ShaRP that generates vectors with the same properties as $\mathbf{w}$ above. The Dirichlet probability distribution

$$\mathbf{w} \sim \mathrm{Dir}(\alpha_1, \alpha_2, \alpha_3) \Rightarrow \mathbf{w} \in [0,1]^3, \quad \sum_{i=1}^{3} w_i = 1 \quad (\alpha_i > 0,\ \forall i)$$

does exactly that. We choose as prior the "uniform" distribution on the triangle, which corresponds to $\mathrm{Dir}(1,1,1)$.

If we stopped here, our algorithm would fail to learn a useful embedding since every data point will draw samples from a single triangle, *i.e.*, the encoding layer will map all points to the same region in 2D space. Hence, we augment our sampling scheme to allow triangles to be rotated, scaled, and translated.

The set of learned parameters used to force shapes into triangles is then $\theta = (\phi \in [-\pi, \pi], s_x \in \mathbb{R}_+, s_y \in \mathbb{R}_+, t_x \in \mathbb{R}, t_y \in \mathbb{R}, \alpha_1, \alpha_2, \alpha_3)$. Here, ϕ is a rotation angle; s_x and s_y are scaling factors in the x and y directions; t_x and t_y are translation amounts in the x and y directions; and α_i are the sampling distribution parameters. A forward pass through this layer is then given by

$$\mathbf{w} \sim \mathrm{Dir}(\alpha_1, \alpha_2, \alpha_3)$$
$$\mathbf{p} = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \mathbf{Tw} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

As a result, we get clusters that are shaped like triangles. This parameterization is capable of generating every possible triangle in $\mathbb{R}^2$ and is more convenient than learning triangle vertices directly. We can even add individual regularization losses depending on the parameter's semantics. For example, we choose *not* to add regularization to ϕ, to allow it to freely range over its domain; we regularize s_x, s_y towards 1 and t_x, t_y towards 0.

We show in Fig. 3 the effect this has on different datasets, as well as what happens when we remove a degree of freedom from the sampling scheme, namely freezing the t_x, t_y translation amounts at 0.

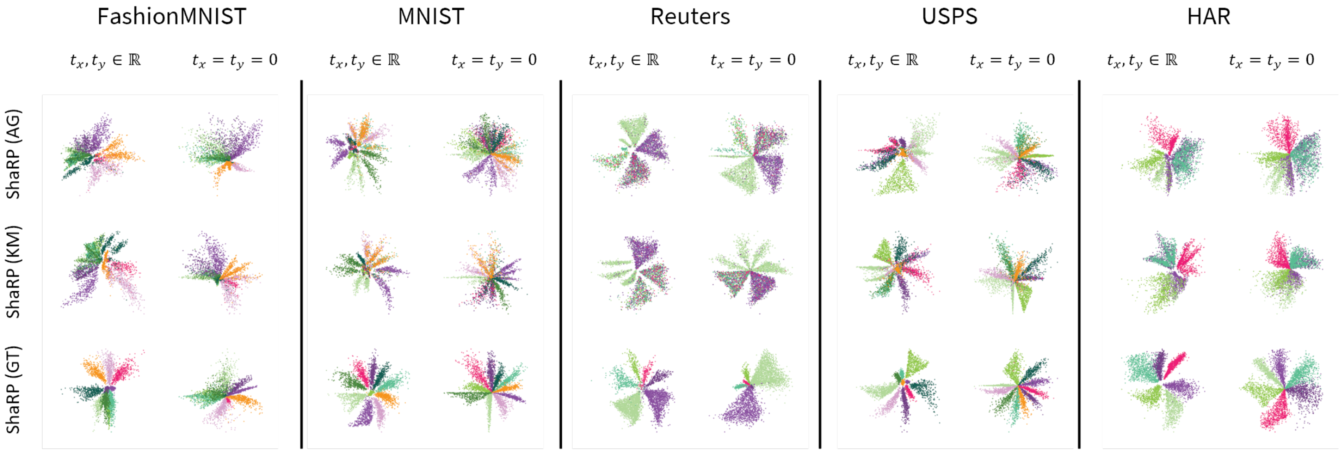

Figure 3: Triangle-oriented shape regularization demonstrated for all datasets studied. We present all three variants of ShaRP and also explore the impact of (dis)allowing the trainslation of triangles in space.

C Detailed performance measurements

ShaRP is one of the fastest techniques among the ones we compare it to. We show here the performance of learning a projection function (when the algorithm requires it) and then projecting the data points. These two steps are performed over an increasing number of samples, and we can see how the run time of different algorithms increases. t-SNE notably grows faster than all other techniques studied, while ShaRP presents run time growth linear in the number of training examples.

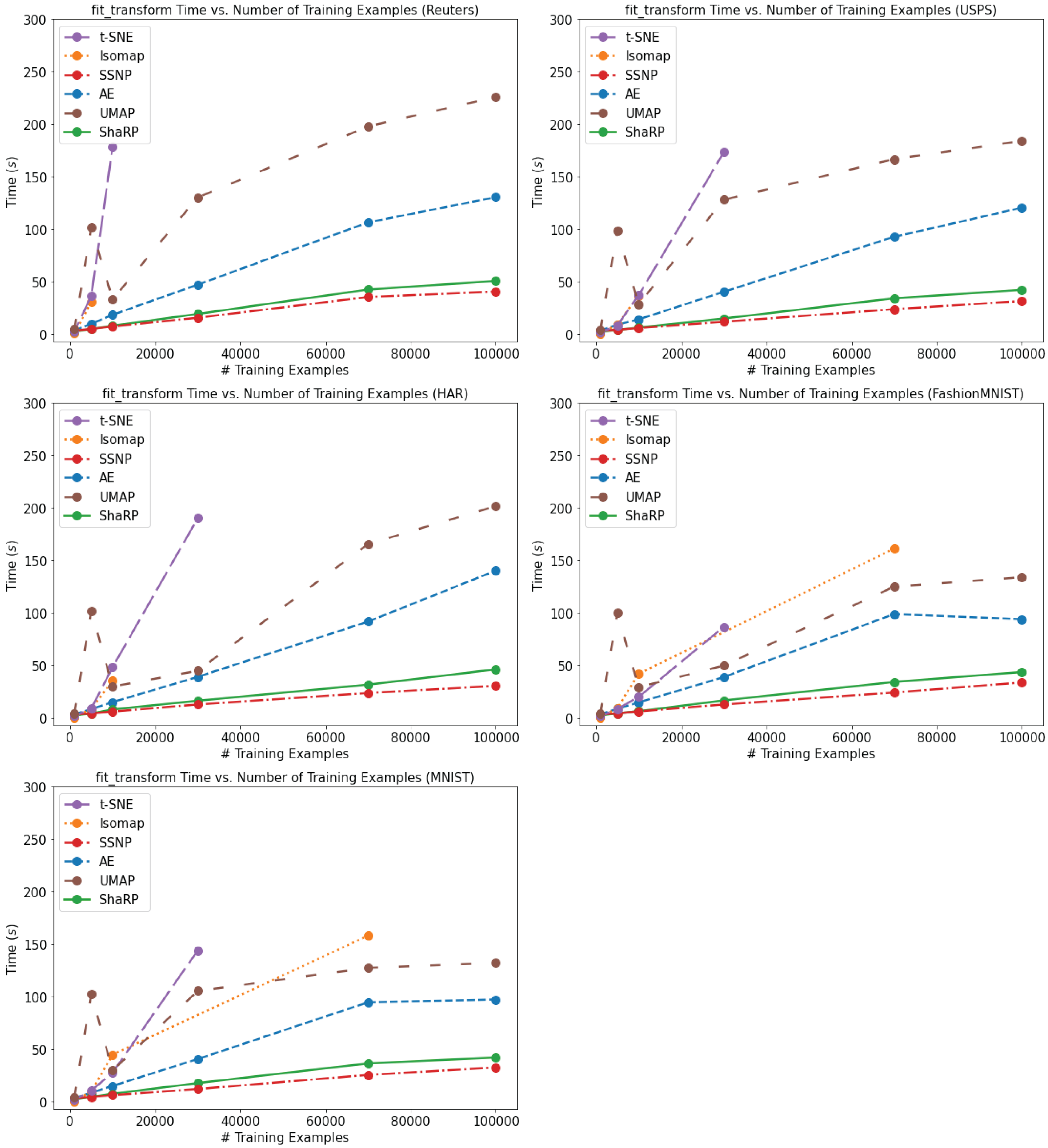

Figure 4: Run time measurements of `fit_transform` calls for different algorithms across all studied datasets.

D Detailed projection quality metrics

For completeness and ease of scrutiny, we provide the non-aggregated quality metrics the main paper's Tables 2 and 3. These are shown respectively in Table 1 and Table 2. ShaRP is also able to perform inverse projection since it uses an Auto-Encoder. We report here the Mean Squared Error between an input and its reconstruction as a quality measure of the inverse projection both over data (Train MSE) used for training and unseen data (Test MSE).

Table 1: Non-aggregated quality metrics for all combinations of technique and dataset used in our work.

Algorithm	Dataset	Trustworthiness	Continuity	Shepard Correlation	Stress	Neighborhood Hit	Distance Consistency	Train MSE	Test MSE
Isomap	FashionMNIST	0.920	0.976	0.749	3.740	0.685	0.544	—	—
	HAR	0.924	0.971	0.899	12.020	0.863	0.729	—	—
	MNIST	0.760	0.959	0.526	2.467	0.621	0.490	—	—
	Reuters	0.625	0.756	-0.153	1.834	0.753	0.540	—	—
	USPS	0.855	0.973	0.679	1.716	0.762	0.633	—	—
t-SNE	FashionMNIST	0.990	0.987	0.664	5.351	0.843	0.656	—	—
	HAR	0.992	0.985	0.578	48.560	0.969	0.813	—	—
	MNIST	0.985	0.972	0.410	8.961	0.945	0.781	—	—
	Reuters	0.742	0.912	0.021	30.816	0.846	0.580	—	—
	USPS	0.989	0.983	0.508	36.068	0.969	0.941	—	—
UMAP	FashionMNIST	0.981	0.988	0.642	0.175	0.804	0.665	—	—
	HAR	0.979	0.989	0.770	3.609	0.935	0.733	—	—
	MNIST	0.957	0.974	0.385	0.236	0.918	0.841	—	—
	Reuters	0.662	0.862	-0.107	2.045	0.759	0.517	—	—
	USPS	0.972	0.985	0.430	0.883	0.958	0.950	—	—
NNP[t-SNE]	FashionMNIST	0.966	0.987	0.679	0.925	0.772	0.656	—	—
	HAR	0.967	0.985	0.619	0.846	0.906	0.812	—	—
	MNIST	0.941	0.969	0.407	0.913	0.879	0.728	—	—
	Reuters	0.687	0.910	0.016	0.820	0.797	0.567	—	—
	USPS	0.961	0.982	0.476	0.858	0.955	0.938	—	—
AE	FashionMNIST	0.964	0.977	0.501	0.344	0.738	0.628	0.027	0.029
	HAR	0.945	0.974	0.825	0.301	0.822	0.653	0.006	0.006
	MNIST	0.912	0.914	-0.048	3.091	0.741	0.492	0.039	0.041
	Reuters	0.637	0.751	0.186	1.623	0.809	0.705	0.001	0.001
	USPS	0.932	0.958	0.222	0.740	0.857	0.663	0.027	0.029
SSNP (AG)	FashionMNIST	0.950	0.975	0.720	0.242	0.750	0.539	0.034	0.035
	HAR	0.925	0.966	0.607	0.245	0.867	0.777	0.007	0.007
	MNIST	0.856	0.924	0.218	0.420	0.811	0.638	0.046	0.048
	Reuters	0.604	0.790	0.226	0.270	0.753	0.649	0.001	0.001
	USPS	0.897	0.963	0.424	0.275	0.878	0.772	0.034	0.035
SSNP (KM)	FashionMNIST	0.961	0.981	0.722	0.188	0.744	0.650	0.030	0.031
	HAR	0.931	0.959	0.546	0.210	0.819	0.588	0.007	0.007
	MNIST	0.881	0.920	0.392	0.328	0.757	0.464	0.045	0.046
	Reuters	0.616	0.816	0.266	0.601	0.750	0.634	0.001	0.001
	USPS	0.915	0.959	0.465	0.306	0.852	0.727	0.032	0.034
SSNP (GT)	FashionMNIST	0.865	0.931	0.389	0.539	0.928	0.866	0.044	0.044
	HAR	0.882	0.942	0.640	0.413	0.991	0.982	0.008	0.008
	MNIST	0.779	0.920	0.362	0.569	0.993	0.927	0.052	0.052
	Reuters	0.585	0.760	0.325	0.488	0.979	0.969	0.001	0.001
	USPS	0.870	0.954	0.508	0.555	0.991	0.918	0.038	0.039
ShaRP (AG)	FashionMNIST	0.917	0.956	0.698	0.864	0.711	0.620	0.037	0.038
	HAR	0.883	0.925	0.546	0.744	0.796	0.692	0.008	0.008
	MNIST	0.822	0.910	0.167	0.859	0.776	0.669	0.052	0.052
	Reuters	0.585	0.706	0.137	0.665	0.740	0.594	0.001	0.001
	USPS	0.870	0.946	0.402	0.779	0.843	0.735	0.039	0.039
ShaRP (KM)	FashionMNIST	0.935	0.961	0.731	0.869	0.707	0.618	0.036	0.037
	HAR	0.899	0.924	0.542	0.741	0.774	0.632	0.008	0.008
	MNIST	0.860	0.904	0.135	0.859	0.731	0.594	0.050	0.051
	Reuters	0.589	0.760	0.091	0.679	0.732	0.633	0.001	0.001
	USPS	0.890	0.935	0.415	0.756	0.811	0.640	0.037	0.038
ShaRP (GT)	FashionMNIST	0.822	0.915	0.482	0.875	0.855	0.823	0.049	0.049
	HAR	0.821	0.893	0.544	0.739	0.962	0.955	0.010	0.010
	MNIST	0.736	0.900	0.289	0.857	0.979	0.958	0.054	0.054
	Reuters	0.550	0.690	0.006	0.672	0.949	0.739	0.001	0.001
	USPS	0.795	0.906	0.408	0.765	0.972	0.931	0.044	0.045

Table 2: Non-aggregated quality metrics for different sampling schemes within ShaRP.

Dataset	Shape	Trustworthiness	Continuity	Shepard Correlation	Stress	Neighborhood Hit	Distance Consistency	Train MSE
FashionMNIST	◯	0.835	0.907	0.463	0.880	0.829	0.766	0.048
	□[$\omega = 15$]	0.829	0.916	0.495	0.891	0.813	0.771	0.049
	□[$\omega = 5$]	0.825	0.915	0.485	0.892	0.842	0.816	0.050
	△	0.826	0.898	0.482	0.875	0.895	0.781	0.050
	△†	0.807	0.808	0.208	0.866	0.834	0.641	0.053
HAR	◯	0.823	0.810	0.521	0.749	0.965	0.829	0.010
	□[$\omega = 15$]	0.829	0.886	0.513	0.768	0.928	0.909	0.010
	□[$\omega = 5$]	0.825	0.880	0.582	0.749	0.952	0.936	0.010
	△	0.828	0.854	0.563	0.662	0.978	0.921	0.010
	△†	0.830	0.824	0.426	0.717	0.980	0.928	0.010
MNIST	◯	0.732	0.897	0.251	0.859	0.977	0.962	0.055
	□[$\omega = 15$]	0.735	0.900	0.230	0.864	0.968	0.948	0.055
	□[$\omega = 5$]	0.735	0.896	0.271	0.878	0.979	0.961	0.055
	△	0.747	0.874	0.119	0.850	0.986	0.896	0.054
	△†	0.738	0.799	0.148	0.844	0.959	0.769	0.056
Reuters	◯	0.554	0.701	0.104	0.676	0.962	0.851	0.001
	□[$\omega = 15$]	0.558	0.715	0.256	0.760	0.977	0.953	0.001
	□[$\omega = 5$]	0.556	0.699	0.300	0.714	0.975	0.933	0.001
	△	0.560	0.696	0.265	0.581	0.981	0.953	0.001
	△†	0.561	0.637	0.002	0.665	0.977	0.793	0.001
USPS	◯	0.802	0.922	0.285	0.774	0.973	0.931	0.044
	□[$\omega = 15$]	0.798	0.919	0.329	0.792	0.962	0.934	0.045
	□[$\omega = 5$]	0.799	0.918	0.350	0.793	0.975	0.948	0.045
	△	0.823	0.898	0.363	0.775	0.992	0.887	0.043
	△†	0.804	0.826	0.185	0.763	0.972	0.837	0.045

Shape	Sampling Scheme
◯	ellipses, Gaussian sampling
□[$\omega = k$]	squares, generalized Normal sampling with $\omega = k$
△	triangles, Dirichlet sampling
△†	triangles, Dirichlet sampling with $t_x = t_y = 0$

E Quality metric definitions

We provide the formal definition of each quality metric used in our work in Table 3. For this, we use the following notation: $\mathrm{NN}_i^{(K)}$ is the set of K-nearest neighbors of $\mathbf{x}_i$ in the high-dimensional space; $\widehat{\mathrm{NN}}_i^{(K)}$ is the set of K-nearest neighbors of $\hat{\mathbf{x}}_i = P(\mathbf{x}_i)$ in the low-dimensional space. We also use these as functions, omitting the data point index: $\mathrm{NN}_i^{(K)} = \mathrm{NN}^{(K)}(\mathbf{x}_i)$. We denote by $r(i,j)$ (resp. $\hat{r}(i,j)$) the rank of the $\mathbb{R}^n$ (resp. $\mathbb{R}^q$) point $\mathbf{x}_j$ (resp. $P(\mathbf{x}_j)$) in the ordered set of nearest neighbors of $\mathbf{x}_i$ (resp. $P(\mathbf{x}_i)$) in $\mathbb{R}^n$ (resp. $\mathbb{R}^q$). Further, we use $Y : \mathbf{X} \to \{1, \ldots, K\}$ as the function that outputs a given data point's label (where here K means the number of classes) and $c : \{1, \ldots, K\} \to \mathbb{R}^n$ as the function that returns the centroid of a given class. We also employ indicator function notation $[\![\cdot]\!]$, that returns 1 when the condition inside it is true, and 0 otherwise.

Table 3: Quality metric definitions. The best value for each metric is **bold** under the Range column.

Metric	Definition	Range	Parameters
Trustworthiness [4]	$1 - \frac{2}{NK(2n-3K-1)} \sum_{i=1}^{N} \sum_{j \in \mathrm{NN}_i^{(K)} \setminus \widehat{\mathrm{NN}}_i^{(K)}} (\hat{r}(i,j) - K)$	$[0, \mathbf{1}]$	$K = 7$
Continuity [4]	$1 - \frac{2}{NK(2n-3K-1)} \sum_{i=1}^{N} \sum_{j \in \widehat{\mathrm{NN}}_i^{(K)} \setminus \mathrm{NN}_i^{(K)}} (r(i,j) - K)$	$[0, \mathbf{1}]$	$K = 7$
Shepard Diagram Correlation [1]	Spearman's ρ of $(\lVert \mathbf{x}_i - \mathbf{x}_j \rVert, \lVert P(\mathbf{x}_i) - P(\mathbf{x}_j) \rVert), 1 \leq i \leq N, i \neq j$	$[0, \mathbf{1}]$	—
Stress [1]	$\frac{\sum_{\mathbf{x} \in \mathbf{X}} \sum_{\mathbf{x}' \in \mathbf{X}} (\lVert \mathbf{x} - \mathbf{x}' \rVert - \lVert P(\mathbf{x}) - P(\mathbf{x}') \rVert)^2}{\sum_{\mathbf{x} \in \mathbf{X}} \sum_{\mathbf{x}' \in \mathbf{X}} \lVert \mathbf{x} - \mathbf{x}' \rVert^2}$	$[\mathbf{0}, \infty)$	—
Neighborhood Hit [2]	$\frac{1}{N} \sum_{\mathbf{x} \in \mathbf{X}} \frac{\lvert \{\mathbf{x}' \in \mathbf{X} \mid P(\mathbf{x}') \in \widehat{\mathrm{NN}}^{(K)}(P(\mathbf{x})) \wedge Y(\mathbf{x}') = Y(\mathbf{x})\} \rvert}{K}$	$[0, \mathbf{1}]$	$K = 7$
Distance Consistency [3]	$\frac{1}{N} \sum_{\mathbf{x} \in \mathbf{X}} [\![Y(\mathbf{x}) = \arg\min_{y \in \{1, \ldots, K\}} \lVert P(\mathbf{x}) - P(c(y)) \rVert]\!]$	$[0, \mathbf{1}]$	—
Mean Squared Error	$\frac{1}{N} \sum_{\mathbf{x} \in \mathbf{X}} \lVert \mathbf{x} - P^{-1}(P(\mathbf{x})) \rVert$	$[\mathbf{0}, \infty)$	—

References

[1] P. Joia, F. V. Paulovich, D. Coimbra, J. A. Cuminato, and L. G. Nonato. Local affine multidimensional projection. *IEEE transactions on visualization and computer graphics*, 17(12):2563–2571, Dec 2011.

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

[3] M. Sips, B. Neubert, J. P. Lewis, and P. Hanrahan. Selecting good views of high-dimensional data using class consistency. *Computer graphics forum*, 28(3):831–838, Jun 2009.

[4] J. Venna and S. Kaski. Local multidimensional scaling. *Neural networks*, 19(6):889–899, 2006.