# *nSimplex Zen*: A Novel Dimensionality Reduction for Euclidean and Hilbert Spaces

RICHARD CONNOR, University of St Andrews
LUCIA VADICAMO, CNR-ISTI

Dimensionality reduction techniques map values from a high dimensional space to one with a lower dimension. The result is a space which requires less physical memory and has a faster distance calculation. These techniques are widely used where required properties of the reduced-dimension space give an acceptable accuracy with respect to the original space.

Many such transforms have been described. They have been classified in two main groups: *linear* and *topological*. Linear methods such as Principal Component Analysis (PCA) and Random Projection (RP) define matrix-based transforms into a lower dimension of Euclidean space. Topological methods such as Multidimensional Scaling (MDS) attempt to preserve higher-level aspects such as the nearest-neighbour relation, and some may be applied to non-Euclidean spaces.

Here, we introduce *nSimplex Zen*, a novel topological method of reducing dimensionality. Like MDS, it relies only upon pairwise distances measured in the original space. The use of distances, rather than coordinates, allows the technique to be applied to both Euclidean and other Hilbert spaces, including those governed by Cosine, Jensen-Shannon and Quadratic Form distances.

We show that in almost all cases, due to geometric properties of high-dimensional spaces, our new technique gives better properties than others, especially with reduction to very low dimensions.

## 1 INTRODUCTION

The requirement to work with large, high-dimensional metric spaces is a long-standing and increasingly important requirement across many domains of computation. Typically, each element of such a space represents some real-world artefact, and the distance between elements gives a proxy (dis-)similarity function over the real-world domain.

As technology progresses, both the dimension of spaces and the size of collections tend to increase. These factors increasingly imply that apparently simple calculations, for example to find a few of the most similar elements within a large set, may become intractable. While sub-linear search times may in some cases be achieved using metric indexing techniques [11, 55], these also become

ineffective in high dimensions and it has been shown that even approximate search complexity often degrades to linear as dimensionality increases [43].

In the domain of image search for example, a collection of a million images would no longer be considered large, and using modern techniques the dimension of the space used to represent them is likely to be several thousand. In this case an exhaustive search to find the image in the collection which is most similar to another will require upward of $10^9$ numeric comparisons, with 10GBytes of data passing through main memory. If we consider that the YFCC benchmark image set [47] contains $10^{10}$ images, and the size of a representation derived from GoogleNet [45] has around 200, 000 dimensions, clearly any such computation is well beyond the scope of most computational contexts.

The purpose of dimensionality reduction is to reduce the dimension of the space, maintaining as far as possible the relative distances. This not only implies that less space is required in memory for the dataset, but also gives a faster distance computation.

There are a number of well-known approaches to dimensionality reduction. The Johnson-Lindenstrauss lemma shows a surprisingly small lower bound on the degree of distortion which is necessarily introduced when reducing a high-dimensional Euclidean space to a lower dimension. This bound may be achievable by using a Random Projection (RP) into the lower dimension, and if the domain is perfectly uniformly distributed then this may be the best technique that is achievable.

However much data is implicitly non-uniform, for example data deriving from a Convolutional Neural Network (CNN) typically lies on some complex manifold within the representational space [8]. In these cases non-random transforms, which take advantage of non-uniformities within the data, can achieve better results.

Dimensionality reduction mechanisms are often classified in two main groups. Linear transforms, such as Principal Component Analysis (PCA) [27, 41], apply only to Euclidean spaces and perform a matrix-based transform derived from properties of the whole space, with the primary goal of preserving pairwise distances as far as possible. Topological transforms, such as Multidimensional Scaling (MDS) [21], use the individual pairwise distances of the original space in an attempt to preserve higher-level relationships, such as the pairwise ordering of distances or the nearest-neighbour relation. Topological methods may be applicable to non-vector spaces, or even non-metric spaces with an appropriate dissimilarity function.

In this article we introduce a new mechanism, which has interestingly different properties, and which can be applied to Euclidean and, more generally, to any space that is isometrically embeddable in Hilbert space. It is primarily a distance-preservation mechanism, and uses properties of high-dimensional Euclidean geometry which have not been previously applied in this domain to preserve distances in the reduced-dimension space.

### 1.1 Outline of some Dimensionality Reduction mechanisms

We refer to the novel Dimensionality Reduction (DR) transform introduced here as *nSimplex Zen*. In this section, by way of motivation, we briefly contrast it with the three other best-known DR transform techniques. More detail of the other transforms is given in Section 3, and a full definition of *nSimplex Zen* in Section 4.

In all cases we consider transforming a given space of $n$ elements, with a dimensionality[1] of $m$, into a Euclidean space with $n$ elements and $k$ dimensions, where $k < m$.

    **Random Projection (RP)** is applicable only to an $m$-dimensional Euclidean space, where $m > k$. A randomised, orthonormal matrix of $m \times k$ dimensions is generated, and the $n \times m$

---

[1]the term *dimensionality* can be usefully applied to Hilbert spaces, see note in Section 3

matrix representing the data is transformed by matrix multiplication to an $n \times k$ matrix representing the reduced dimensional space.

**Principal Component Analysis (PCA)** is also applicable only to an $m$-dimensional Euclidean space. A matrix of $m \times m$ dimensions representing the principal components is generated from the original space, and the $n \times m$ matrix representing the data is transformed by matrix multiplication with the most significant $k$ columns of this matrix to yield the reduced dimensional space.

**Multidimensional Scaling (MDS)** is applicable to any metric or semi-metric space, and considers only the distances among the $n$ elements of the input space. An $n \times n$ (upper triangular) matrix of pairwise distances is taken as input, and the output is an $n \times k$ matrix, representing a Euclidean space which preserves pairwise distances as far as possible. MDS does not scale well, but an adaptation can be applied to large Euclidean spaces. A variant of MDS, Landmark MDS, can be applied to general metric spaces.

***nSimplex Zen*** is applicable to any metric space which is isometrically embeddable in Hilbert space[2]. It takes as input a reference set $\mathcal{R}$ comprising $k$ elements of the input space. A simplex in $(k-1)$ Euclidean dimensions is constructed using the pairwise distances measured within this set. With reference to this structure, each further element of the input space is then mapped to a $k$-dimensional Cartesian coordinate according to its distances from each element of $\mathcal{R}$.

*nSimplex* in its simplest form is a mapping from $(\mathcal{U}, d)$ to $(\mathbb{R}^k, \ell_2)$, where $k$ is the size of the reference set $\mathcal{R}$. This can be used in its own right as a DR technique. There are however two further functions which can be applied to the $\mathbb{R}^k$ space resulting from the mapping: *Zen* and *Upb*, which are defined in Section 4. There are thus three different spaces which may be formed as a result of the projection. *nSimplex Zen* refers to the mapping from $(\mathcal{U}, d)$ to $(\mathbb{R}^k, Zen)$ and is the main object of our attention.

In contrast with the other techniques, calculation of the *nSimplex* transform does not require any operations over matrices. Instead, it relies upon higher-level geometric properties of the original metric space, which are reflected within the generated $k$-dimensional space.

## 1.2 Relation to previous work

The ideas underlying *nSimplex Zen* have emerged after several years of research in the intersection between similarity search and distance geometry, in particular with respect to work done in the early 20th Century by, among others, Blumenthal, Hilbert, Menger, and Wilson, which we summarise in Section 2.1.

In [12] we showed how the four-point property possessed by some metric spaces[3] could be used to improve many metric search techniques, and in [13, 14] it was shown that this observation could be applied in practice to a number of state-of-the-art metric indexing techniques. We extended this work in [15] after making the observation that the major four-point spaces we had identified (Jensen-Shannon, Quadratic Form, Cosine, Triangular) also possessed the $n$-point property, and that the distance lower-bound we had identified for four-point spaces in the 2D projection applied more generally in higher dimensions. During this work we also identified the algorithm, referred to as *nSimplex Projection*, for constructing the simplex as given here in Appendix B, and the proof

---

[2]This includes any Euclidean space, and also metric spaces governed by appropriate variants of Cosine, Jensen-Shannon, Quadratic Form, and Triangular distances (see [12] and Appendix A for details.)

[3]see Section 2.1.2 for a description of this and the $n$-point property

given in Appendix C that the *Lwb* function given in Section 4 is a lower bound of the distance in the domain of the *nSimplex* transform[4].

We exploited the *nSimplex* Projection also in the context of similarity search as an intermediate step to transform the original data into permutations and binary strings to be employed in efficient approximate metric indexes [48, 49, 51].

In [18] we noted the application of the *Zen* function introduced in Section 4 is a better estimator of true distance than the *Lwb* function described in [15] in the domain of string similarity functions. Moreover, in [50] the *Zen* function has been exploited on other domains (image features and word embeddings) to effectively refine candidate results obtained using a permutation-based k-NN search without accessing the original data. We subsequently observed the very non-uniform pattern of angles measured within the simplexes thus formed from high-dimensional spaces. This angular distribution was separately examined and published in [19].

The main contribution of this article is the full exposition and analysis of the *Zen* function of the *nSimplex* construction as a general dimensionality reduction technique for Euclidean and other Hilbert spaces, all of which content is entirely novel.

### 1.3 Paper Outline

The rest of the article is structured as follows. Section 2 gives an overview of the necessary mathematical background required to understand the mechanism and motivation of the *nSimplex Zen* transform. Section 3 gives background on dimensionality reduction, and outlines the four mechanisms with which we compare *nSimplex Zen*. Section 4 gives a detailed account of the *nSimplex* transform, and its three related functions: *Zen*, *Lwb* and *Upb*. Section 5 gives experimental results comparing the quality of the reductions given by *nSimplex Zen* with the transforms introduced in Section 3, and Section 6 compares the run-time performance of the various mechanisms. Finally, Section 7 gives a discussion of the *nSimplex Zen* mechanism and its results, and concludes with some possible future work. Table 1 summarises notations used throughout this work.

## 2 BACKGROUND: METRIC SPACES AND THEIR PROPERTIES

The novel *nSimplex* mechanism is described fully in Section 4. Before it can be understood in detail a significant amount of mathematical background is required, and provided in this section. Section 2.1 gives some mathematical preliminaries, and Section 2.2 gives some more specific background in high-dimensional geometry. Section 3 gives a general background to dimensionality reduction.

### 2.1 Metric spaces, embeddings and simplexes

Our work relies on the ability to construct a *simplex* in a $k$-dimensional Euclidean space, whose edge lengths correspond to the distances among any $(k + 1)$ objects selected from any metric space which is $(k + 1)$-isometrically embeddable in a Hilbert space. These underlying concepts are briefly explained in this subsection.

*2.1.1 Metric and Semimetric Spaces.* Let $(\mathcal{U}, d)$ be a pair comprising a domain of objects $\mathcal{U}$ and a numeric dissimilarity function $d : \mathcal{U} \times \mathcal{U} \to \mathbb{R}$. In general the more similar the objects $x, y \in \mathcal{U}$, the smaller the value of $d(x, y)$. For a space to be semimetric, it requires $d$ to be positive or zero, with $d(x, y) = 0$ if and only if $x = y$, and symmetric, i.e. $d(x, y) = d(y, x)$. A (proper) metric space, governed by a (proper) distance function, also possesses the triangle inequality property, i.e. $d(x, z) \le d(x, y) + d(y, z)$. For the rest of this article, we use the terms *distance* and *metric space* to refer to functions and spaces with these properties.

---

[4]In fact these were not included in the published version due to space limitations, but were given in an adjunct *arXiv* publication [20]

Table 1. Table of notations used

| Notation | Meaning |
|---|---|
| $n$ | The cardinality of both domain and range of a DR transform |
| $m$ | The dimension of a Euclidean space which is the domain of a DR transform |
| $k$ | The dimension of a Euclidean space which is the co-domain of a DR transform |
| $(\mathcal{U}, d)$ | A metric space with domain $\mathcal{U}$ and distance function $d$ |
| $(\mathcal{S}, d)$ | A (typically large) finite subspace of $\mathcal{U}$, $\mathcal{S} \subset \mathcal{U}$ |
| $\mathcal{R}$ | A (typically small) set of reference objects drawn from $\mathcal{S}$ |
| $u, u_i$ | Individual objects drawn from $\mathcal{U}$ |
| $s, s_i$ | Individual objects drawn from $\mathcal{S}$ |
| $r, r_i$ | Individual reference objects drawn from $\mathcal{R}$ |
| $\tau$ | A DR transform mapping some $(\mathcal{U}, d)$ to $(\mathcal{U}', \zeta)$ |
| $\sigma$ | An *nSimplex* transform based on $k$ reference objects, $\sigma : \mathcal{U} \rightarrow \mathbb{R}^k$ |
| $\ell_2$ | The Euclidean distance function |
| $(\mathbb{R}^m, \ell_2)$ | A Euclidean space of $m$ dimensions |
| $\delta_{ij}$ | The distance $d(i, j)$ where $i, j \in (\mathcal{S}, d)$ |
| $\zeta_{ij}$ | The corresponding distance $\zeta(i, j)$ for corresponding $i, j \in (\mathcal{S}', \zeta)$, where $\mathcal{S}' = \tau(\mathcal{S})$ |

*2.1.2   Metric Spaces and Isometric Embeddings.* For metric spaces $(\mathcal{U}, d)$ and $(\mathcal{U}', \zeta)$ we say that $\mathcal{U}$ is *isometrically embeddable* in $\mathcal{U}'$ if there exists a function $f : \mathcal{U} \rightarrow \mathcal{U}'$ such that $\zeta(f(x), f(y)) = d(x, y)$ for all $x, y \in \mathcal{U}$. A *finite isometric embedding* is defined when such a function exists for a finite subset of $\mathcal{U}$. A finite isometric embedding may be generalised to any fixed size of subset; for example, we state that $(\mathcal{U}, d)$ is finitely $n$-embeddable in $(\mathcal{U}', \zeta)$ where such a function exists for any subset of $n$ values selected from $\mathcal{U}$.

These concepts give rise to an alternative definition of a metric space. Normally, a metric space is defined as a semimetric space which has the triangle inequality property. Alternatively, a metric space may be defined as a semimetric space which is finitely 3-embeddable in 2D Euclidean space, this being an equivalent property.

Finite isometric embeddings are summarised by Blumenthal [7]. He defines the *four-point property* to refer to any space that is finitely 4-embeddable in 3-dimensional Euclidean space. Wilson [54] shows various properties of such spaces, and Blumenthal points out that results given by Wilson, when combined with work by Menger [38], generalise to show that some spaces with the four-point property also have the $n$-point property: that is, for any $n$, they are finitely $n$-embeddable in $(n-1)$-dimensional Euclidean space. In a later work, Blumenthal [6] shows that any space which is isometrically embeddable in a Hilbert space has the $n$-point property. This is a generalisation of the better-known result that any $n$ points from a Euclidean space of any dimension may be isometrically embedded in $(n-1)$-dimensional Euclidean space, and is the main abstract result we rely upon here.

*2.1.3   Hilbert Spaces.* A Hilbert space is a real or complex inner product space that is also a complete metric space with respect to the distance function induced by the inner product. Hilbert spaces possess inner product and distance functions with properties analogous to, but not necessarily the same as, the dot product and distance functions of a Euclidean space.

A Hilbert space is a generalisation of a Euclidean space, allowing the study of vector spaces which do not necessarily have finite coordinate systems. Hilbert spaces have always been important in the study of abstract geometry. All Euclidean vector spaces are Hilbert spaces; in more recent years however, many further useful spaces have been identified as being isometrically embeddable in Hilbert space. These include spaces governed by the Jensen-Shannon, Quadratic Form, Triangular, and Cosine[5] distances. These spaces do not have Euclidean coordinates, and so cannot be manipulated via matrix arithmetic, but can be used as the domain of the transform we propose here as they inherit the $n$-point property of a Hilbert space.

*2.1.4 Construction of a simplex.* A simplex is the generalisation of a triangle or a tetrahedron in arbitrary dimensions of Euclidean space. In one dimension, a simplex is a line segment. In two dimensions it is a triangle, while in three dimensions it is a tetrahedron. In general, a point $\mathbf{v}_1$ forms a 0-simplex, and the $n$-simplex of vertices $\mathbf{v}_1, \ldots, \mathbf{v}_{n+1}$ is given by the union of the simplex formed from $\mathbf{v}_1, \ldots, \mathbf{v}_n$ with the line segments joining $\mathbf{v}_{n+1}$ to all vertices of that simplex.

The property that a Hilbert space $(\mathcal{U}, d)$ is finitely $(n+1)$-embeddable in $n$-dimensional Euclidean space directly implies that, for any $(n + 1)$ objects in $\mathcal{U}$, it is possible to construct a simplex with $(n + 1)$ vertices in $\mathbb{R}^n$, where each vertex corresponds to an object in $\mathcal{U}$, and the edges joining all pairs of adjacent vertices correspond with the distances between the corresponding objects in $\mathcal{U}$.

In Appendix B we show an algorithm for determining Cartesian coordinates for the vertices of a simplex, given only the distances between all pairs of points. The algorithm is inductive, at each stage allowing the apex of an $n$-dimensional simplex to be determined given the coordinates of an $(n - 1)$-dimensional simplex, and the distances from the new apex to each vertex in the existing simplex.

The outcome of this algorithm represents a simplex in $n$-dimensional space as a lower triangular $n + 1$ by $n$ matrix representing the Cartesian coordinates of each vertex. For example, the rows of the following matrix represent the coordinates $v_{i,j}$ of four vertices $\mathbf{v}_1, \ldots, \mathbf{v}_4$ of a tetrahedron in 3D space:

$$
\begin{bmatrix}
0 & 0 & 0 \\
v_{2,1} & 0 & 0 \\
v_{3,1} & v_{3,2} & 0 \\
v_{4,1} & v_{4,2} & v_{4,3}
\end{bmatrix}
\tag{1}
$$

This matrix is derived from four objects $o_1, \ldots, o_4$ in the Hilbert space, and the distances $d(o_i, o_j)$ are the same as the distances $\ell_2(\mathbf{v}_i, \mathbf{v}_j)$ where $\mathbf{v}_i$ and $\mathbf{v}_j$ are vectors given by the $i$-th and $j$-th rows of the matrix, respectively.

For all such sets of objects, the invariant that $v_{i,j} = 0$ whenever $j \geq i$ can be maintained without loss of generality. For any simplex constructed, this can be achieved by rotation and translation within the Euclidean space while maintaining the distances among all the vertices. Furthermore, if we restrict $v_{i,j} \geq 0$ whenever $j = i - 1$ then in each row this component represents the *altitude* of the point $\mathbf{v}_i$ with respect to a base simplex formed by $\{\mathbf{v}_1, \ldots, \mathbf{v}_{i-1}\}$, which is represented by the matrix derived by selecting elements above and to the left of the entry $v_{i,j}$.

Finally, we note that as long as the entry $v_{i,j}$ is non-zero, i.e. represents a non-zero altitude above the base simplex defined by rows 1 to $i - 1$, then the set of vectors defined by the rows forms a basis for the $n$-dimensional space in which it is constructed. In this way, the process of forming the simplex gives an interesting comparison to the Gram-Schmidt method for forming a basis in a Euclidean space, but the simplex formation method does not require access to a coordinate

---

[5]in one particular form, see Appendix A for details of this and other metrics.
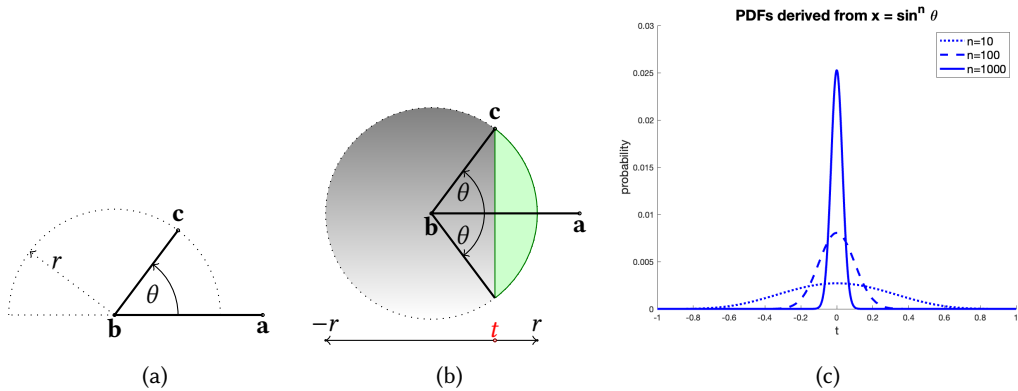
Fig. 1. In $n$ dimensions, for fixed **a** and **b** within a given plane, **c** is sampled from within the same plane at a fixed radius $r$ from **b**. As the dimensionality of the space increases, the probability of $\theta$ being close to $\pi/2$ increases rapidly: the right-hand plot shows probability density functions for various dimensions as $t = r\cos\theta$ varies between $-r$ and $r$.

space defining the original metric space, and can thus be applied to any metric space which is isometrically embeddable in a Hilbert space.

## 2.2 Angles in High-Dimensional Metric Spaces

In Section 4.3 we rely upon a property on the distribution of angles in high-dimensional Euclidean space that is described in this section. While the property itself is relatively straightforward, its derivation from high-dimensional Euclidean geometry is less so, and we therefore give a short justification.

In the context of a uniformly distributed Euclidean space of $n$ dimensions, we are interested in the distribution of angles formed by a hyperplane $H$ and a object **c** on a hypersphere[6] centred on a point $\mathbf{b} \in H$. Without loss of generality this distribution can be measured as the angle between three points **a**, **b**, and **c**, where objects **a** and **b** are fixed in $H$, and **c** is sampled within a fixed radius $r$ from **b**. The considered situation can be easily depicted in 2D (i.e., considering the plane through these three points) in Figure 1a. Given this arrangement, we wish to understand the distribution of the angle $\theta \in [0, \pi]$, that is the angle formed by the three points.

In a high-dimensional vector space it is generally known that two randomly selected vectors are very likely to be close to orthogonal [4, 5, Chapter 2], and it is therefore no surprise that the value of $\theta$ is likely to be close to $\pi/2$. In fact this is the only result required for the understanding of Section 4.3, however further explanation is reasonably required.

In [19] we quantified the distribution of this angle and we observed that as the dimension of the space $n$ increases so too does the probability of $\theta$ being close to $\pi/2$. In particular we observed that

- the total volume $V_n$ of a $n$-ball with radius $r$ is $V_n = r^n U_{n-1} \int_0^\pi \sin^n \psi \, d\psi$, where $U_{n-1}$ is the volume of a unit (n-1)-ball;

---

[6]A hyperspheres in $\mathbb{R}^n$ is a $n-1$-sphere. Note the possibly confusing conventions in the naming of $n$-spheres and $n$-balls: in general, the *surface* of an $n$-ball is denoted as an $(n-1)$-sphere. For example, the volume contained by a sphere in 3D space is a 3-ball, whereas its surface is a 2-sphere.

- the volume $V_n(\theta)$ of the portion of the $n$-ball delimited by the hyperplane through $\mathbf{c}$ and orthogonal to $H$ (i.e. the portion delimited by $\theta$ and denoted by the green-shaded area in Figure 1b) is proportional to $r^n \int_0^\theta \sin^n \psi \, d\psi$.

Actually, here we are interested in the "surface" of the portion of the $n$-ball delimited by $\theta$, rather than its volume. However such regions are strongly related geometrically; in fact in general the surface of an $n$-ball, i.e. an $(n-1)$-sphere, has a volume in $(n-1)$ dimensions. The angular distribution of volume in an $n$-sphere is identical to that in an $(n-1)$-ball [53], therefore the PDF for the distribution of points on an $(n-1)$-sphere is given by a normalisation of the same formula.

While the formula to computes the volume is difficult to quantify, requiring the integration of high powers of the sine function, it has also been observed that for high values of $n$ the function is numerically almost indistinguishable from the normal distribution function given by setting $\mu = \pi/2$ and $\sigma = 1/\sqrt{(n)}$ [1, 10], for which integral values are highly accessible. Figure 1c quantifies this volume as PDFs for various dimensions $n$. It can be observed that as $n$ increases the distribution of $t$ concentrates around the mean value 0, so the distribution of the angle $\theta$ concentrates around $\pi/2$.

This theoretical model can be verified by experiment in unbounded Euclidean spaces. As usual, there are various factors in real-world spaces (in particular boundedness and non-uniformity) which affect the observed distribution of angles, but the theoretical effect is still highly visible; a deeper analysis is given in [19]. In Section 5 we analyse Euclidean spaces drawn from examples in 100 to 4,096 dimensions, in which the effect is very evident.

## 3 RELATED WORK: DIMENSIONALITY REDUCTION

In simple terms, *dimensionality reduction* refers to the transformation of a set of values in dimension $m$ to a set of values in dimension $k$, where $k < m$. Associated with such a transformation is a controlled loss of information within the reduced-dimension space.

The notion of dimensionality in general metric spaces is itself complex, and indeed the establishment of a generally agreed definition of *intrinsic dimensionality* is an ongoing research issue. In Euclidean spaces dimensionality can be understood as the minimum value of $m$ for which an isometric embedding in an $(\mathbb{R}^m, \ell_2)$ space can be defined. For example, a set of points on a plane can be defined in a 3-dimensional space, but the intrinsic dimensionality of this set is 2. Mechanisms such as PCA and MDS can be used to detect the value of intrinsic dimensionality in a Euclidean space. In general, metric spaces cannot be isometrically embedded in Euclidean space and so this definition cannot be used. The concept of dimensionality is important in non-vector spaces, and can be defined upon properties such as the distribution of distances among objects of the space, as shown for example in [11]. In this article we refer to "dimensionality reduction" transforms which map from metric spaces to Euclidean spaces of a lower dimensionality, in these cases we implicitly rely upon such definitions of dimensionality.

Dimensionality reduction techniques are usually classified into two groups: linear and topological. Linear mechanisms are concerned with preserving the accuracy of simple distances, while topological mechanisms are concerned with the preservation of higher-level properties, for example relative rather than absolute distances. In all cases, the main purpose is to reduce the cost of distance comparisons, by reducing both the size of the data and the cost of the distance measurement. In most contexts, we are interested in applying such transforms to a finite metric space $(\mathcal{S}, d)$ which is a (typically very large) subset of an infinite space $(\mathcal{U}, d)$. For example in metric search, the task is to find, from the subset $\mathcal{S}$, values which are similar to a query value $q \in \mathcal{U}$, where typically $q \notin \mathcal{S}$. This causes us to reconsider the general notion of dimensionality reduction. Rather than a transform which maps some finite space $(\mathcal{S}, d)$ to another space $(\mathcal{S}', \zeta)$, where $(\mathcal{S}, d)$ is a finite

Table 2. Dimensionality reduction algorithms used in this research.

| Name | Goals | Input | Output |
|---|---|---|---|
| PCA | Project data onto a lower-dimensional subspace while maximising the retained variance | Euclidean space: original data (or its covariance matrix) | Projection matrix (principal components) and projected data |
| RP | Project data onto a lower-dimensional subspace with small distortion in pairwise distances | Euclidean space: original data | Random projection matrix and projected data |
| MDS | Represent data in a lower-dimensional space, preserving the rank ordering of pairwise distances | Any semi-metric space: distance matrix (all pairwise distances) | Transformed data (coordinates) |
| LMDS | As MDS, but using a subset of landmarks to reduce computational cost | Any semi-metric space: pairwise distances between data objects and landmark points, and pairwise distances among all landmark points | MDS on landmarks and transformed data (coordinates) |
| nSimplex Zen | Represent data in a lower-dimensional space with small distortion in pairwise distances | Any Hilbert-embeddable space: pairwise distances between data objects and reference objects, and pairwise distances among all reference objects | Matrix representing a base simplex and the transformed data (coordinates) |

subset of some infinite space $(\mathcal{U}, d)$, we require a more general transform $\mathcal{T} : (\mathcal{U}, d) \to (\mathcal{U}', \zeta)$ which can map individual elements of $\mathcal{U}$ to $\mathcal{U}'$. This is important for two reasons. First, any analytic technique that analyses the entire finite domain $\mathcal{S}$ in order to map to a new one will be intractable for very large finite domains. Second, such a technique would not allow the mapping of a query value $q$ where $q \in \mathcal{U}$ but $q \notin \mathcal{S}$. We make this distinction as classical definitions of techniques such as MDS assume analysis of the entire finite domain.

In this section, as well as giving outline descriptions of these techniques, we also show how they can be applied in this more general context. There are many dimensionality reduction techniques, many of which have been developed for specific contexts. The Principal Component Analysis (PCA) [27, 41] stands out as the most widely used dimensionality reduction algorithm, finding applications in data compression, data processing, feature extraction, and data visualisation, among others. Random Projection (RP) [22], Multidimensional Scaling (MDS) [21, 32, 44], and their variants aim at representing the data in a lower-dimensional space in such a way that the distances between points in that space approximate the pairwise dissimilarities in the original space. Isomap [46] focuses on preserving geodesic distances and local structures, often used in data analysis and machine learning. Additionally, other techniques such as t-Distributed Stochastic Neighbour Embedding (t-SNE) [52] and Uniform Manifold Approximation and Projection (UMAP) [37], are commonly employed for exploratory data analysis and visualisation. Given our focus on distance-preserving mechanisms and not on data visualisation, here we introduce three general mechanisms most relevant to our context, namely PCA, RP, MDS, which we believe are the mechanisms in most common use. Table 2 provides an overview of these mechanisms, alongside our proposed nSimplex Zen.

## 3.1 Random Projection

According to the Johnson-Lindenstrauss Flattening Lemma (see e.g. [36, page 358]), a random projection can be used to transform a finite set of $n$ Euclidean vectors into a $k$-dimensional Euclidean

space ($k < n$) with a "small" distortion. Specifically the Lemma asserts that for any $n$ points of the space ($\mathbb{R}^m, \ell_2$), and for every $0 < \epsilon < 1$, there is a mapping into ($\mathbb{R}^k, \ell_2$) that preserves all the pairwise distances within a factor of $1 + \epsilon$, where $k = O(\epsilon^{-2} \log n)$. Note that this lemma depends on the size, and not the dimensionality, of the domain.

The low dimensional projection anticipated by the Johnson-Lindenstrauss lemma is particularly simple to implement. Specifically, for a Euclidean space represented by an $n \times m$ matrix, a suitable transform into $k$ dimensions can be achieved through a randomly generated $m \times k$ orthonormal matrix. Practically, there are even better ways of achieving the projection. Achlioptas [2] shows that equally good results can usually be achieved with a much cheaper transform, by creating a $m \times k$ pseudo-orthogonal matrix with, for example, the randomised strategy:

$$R_{i,j} = \sqrt{3} \times \begin{cases} +1 & \text{with probability } 1/6 \\ 0 & \text{with probability } 2/3 \\ -1 & \text{with probability } 1/6 \end{cases} \tag{2}$$

This greatly improves the efficiency of the projection, as it introduces many zeros into the projection matrix, and allows integer arithmetic to be used instead of floating point. Further, the strategy has been shown to give better outcomes in some circumstances than a truly orthonormal matrix. We use this strategy in our comparative experiments in Section 5.

It is self-evident that, for Euclidean data which is uniformly distributed, a random projection is no worse than any other linear technique. However, much real-world data is not uniformly distributed, often in ways that are difficult to predict or analyse, in which case other techniques typically perform better.

## 3.2 Principal Component Analysis

PCA [27, 41] is probably the best known and most widely used unsupervised dimensionality reduction technique, and has been used also for feature extraction and data visualisation. The main idea is to find a linear transformation of $m$-dimensional vectors to $k$-dimensional vectors ($k < m$) that best preserves the variance of the input data. Specifically, PCA determines the principal components of the data, which are those directions within the vector space showing maximum variance. The first such direction is found, and represented by a unit vector; then, the second direction is found within the ($m - 1$)-dimensional subspace orthogonal to this unit vector, and so on until a set of $m$ orthonormal vectors is established. These vectors are represented in an $m$-dimensional square matrix whose columns correspond to the unit vectors established by this process (i.e., the so-called principal components).

If the intrinsic dimensionality of the data is less than $m$, then the last steps of the process will discover a variance of 0 in all directions and the unit vectors derived become arbitrary.

The principal components can be computed by solving a maximisation problem. However, it has been shown that the principal components are the eigenvectors of the covariance matrix of the centred input data. Thus typically they are computed by using spectral analysis via Singular Value Decomposition of the data rather then solving the optimisation problem, which is more expensive.

The eigenvalues $\lambda_1, \ldots, \lambda_m$ give the variances of their respective principal components. Moreover, the ratio

$$\frac{\sum_{i=1}^{k} \lambda_i}{\sum_{j=1}^{m} \lambda_j} \tag{3}$$

represents the proportion of the total variance in the original data set accounted for by the first $k$ principal components.

The dimensionality reduction transform itself is achieved by multiplication of the matrix representing the domain by the first $k$ columns of the principal component matrix. If the value given by Equation 3 is large, then the loss of accuracy in distances measured in the projected space will be correspondingly small.

While PCA is defined as the orthogonal projection of the data onto a lower dimensional linear subspace, such that the variance of the projected data is maximised, there also exists an equivalent definition of PCA that gives rise to the same algorithm. In the latter, the PCA is defined as the linear projection that minimises the average projection cost, defined as the mean squared distance between the data points and their projections [24]. This property implies that PCA is the best strategy for dimensionality reduction in a Euclidean space, where the goal is to minimise the introduced inaccuracy of arbitrary distance measurements.

For a very large data set, the cost of calculating the principal components using the entire set is likely to be intractable. However this cost may be avoided by using a representative sample of the data to generate the principal components. As the projection to construct the reduced-dimension set comprises multiplication of the $n \times m$ data matrix by the $m \times k$ principal component matrix, principal components derived from a representative subset can be used to transform the remainder of the data.

## 3.3 Multidimensional Scaling

MDS [21] is a technique which analyses the pairwise distances within a finite semimetric space $(\mathcal{S}, d)$ and, given a target dimension $k$, generates a $k$-dimensional Euclidean space which preserves topological features of these distances as far as possible. There are two main variants of MDS, so-called "classic" (metric) and non-metric. Here we consider the "non-metric" version as this may be applied to spaces not governed by the Euclidean distance and can thus be compared with the *nSimplex* technique.

MDS iteratively constructs Euclidean vectors, using a gradient descent technique, in order to minimise a stress formula. In the non-metric variant, this is typically Kruskal's (see Section E.1) *stress1* definition:

$$S_K = \sqrt{\frac{\sum_{i<j}(\zeta_{ij} - d_{ij}^*)^2}{\sum_{i<j}\zeta_{ij}^2}} \tag{4}$$

In this formula, $i$ and $j$ are indices over the data objects in $\mathcal{S}$, $\zeta_{ij} = \zeta(\tau(s_i), \tau(s_j))$ is the Euclidean distance measured in the reduced space, and $d_{ij}^* = d^*(s_i, s_j)$ is a function generated by an isotonic regression over the true distances $d(s_i, s_j)$ as a function of the reduced distances $\zeta_{ij}$. Stress is therefore affected not by the absolute difference between distances in the two spaces, but instead according to the relative ordering of distances between them: if this is preserved, then the measured stress will be lower.

MDS is an expensive ($O(n^4)$) algorithm to compute, significantly limiting the size of data to which it can be applied. It has the further disadvantage that as the analysis is over a given finite set of distances among objects, it cannot therefore produce a transform which may be applied to other non-manifest elements of the same domain. This would imply that a representative sample cannot be used to construct a transform which can subsequently be applied to a very large domain, and also that a query from the same universal domain cannot be subsequently transformed into the generated $k$-dimensional space.

In fact it may be possible to generate such a transform when MDS is applied to an $m$-dimensional Euclidean space, using Procrustes analysis and a pseudo-inverse matrix operator, as follows. First, a representative set of $l$ objects is selected from the $n$ objects of the domain, and the $l \times l$ distance matrix is generated. MDS takes this as input and an $l \times k$ matrix is produced to represent the

$k$-dimensional space representing those objects. Procrustes analysis can then be used to produce a transform from this space back to a best fit within the original $m$-dimensional space. Although in general this transform is represented by a non-square matrix, and therefore is not guaranteed to have an inverse, a pseudo-inverse technique can be used to successfully construct the inverse transform. This can therefore be subsequently applied to other samples from the same original space. We use this technique in Section 5 to compare MDS as a dimensionality reduction technique for large Euclidean spaces.

### 3.4 Landmark Multidimensional Scaling

Landmark MDS (LMDS) [44] is a technique which allows MDS to be used for the generation of a general transform over metric spaces, using a triangulation technique. A representative set of landmark values $\mathcal{L}$ is selected from the domain $\mathcal{U}$, and classical MDS is applied in order to transform the (typically) non-Euclidean space $\mathcal{L}$ to a $k$-dimensional Euclidean space, minimising the stress as above.

As already noted, classical MDS does not generate a transform function which can be applied to data not included in the manifest space whose distances are used to construct this transform. Instead, LDMS allows the addition of further elements of the domain to the transformed space using only the distances calculated to each element of $\mathcal{L}$. A triangulation approach is then used to place each subsequent element into the reduced-dimension space with minimal stress on this set of distances.

In this manner, LMDS extends classical MDS in such a way that it can be extended for use over very large data sets and non-manifest queries in non-Euclidean spaces. We use this technique in Section 5.6 to compare LMDS with our *nSimplex Zen* transform over spaces not governed by the Euclidean distance.

## 4 THE *NSIMPLEX* PROJECTION

The *nSimplex* transform can be applied to Hilbert spaces in general, and relies on the Hilbert property that any $k + 1$ values can be isometrically embedded in an $k$-dimensional Euclidean space. In outline, the transform from a Hilbert space $\mathcal{U}$ to $\mathbb{R}^k$ is defined as follows:

(1) $k$ values $r_1, r_2, \ldots r_k$ are first selected from $\mathcal{U}$ to form a reference set $\mathcal{R}$. (Typically, $\mathcal{R}$ will be selected from a large finite subset $\mathcal{S}$ of $\mathcal{U}$.)

(2) All pairwise distances among the values in $\mathcal{R}$ are calculated, and used to construct a *base simplex* $\Sigma$ in a Euclidean space of $k - 1$ dimensions, where each vertex $\mathbf{v}_i$ in $\Sigma$ corresponds to one value $r_i$ in $\mathcal{R}$ with $\ell_2(\mathbf{v}_i, \mathbf{v}_j) = d(r_i, r_j)$ for all $i, j = 1, \ldots, k$.

(3) For any further value $u \in \mathcal{U}$, the distances between $u$ and all values in $\mathcal{R}$ are calculated.

(4) These distances are used to construct a point $\mathbf{v}_u$ in $k$-dimensional Euclidean space, where $\mathbf{v}_u$ is the apex of a simplex formed by its addition to the base simplex $\Sigma$, such that $\ell_2(\mathbf{v}_u, \mathbf{v}_i) = d(u, r_i)$ for all $i = 1, \ldots, k$

These apex points form the target of the transform. The process therefore gives a mapping $\sigma$ from the general Hilbert space $\mathcal{U}$ to a $k$-dimensional Euclidean space, where $\sigma(u) = \mathbf{v}_u, \forall u \in \mathcal{U}$. MATLAB and Java code for computing the nSimplex projection is available on GitHub at https://bitbucket.org/richardconnor/metric-space-framework[7]. For Python users, the corresponding code implementation can be accessed at https://github.com/vadicamo/nSimplex.

---

[7]    Matlab    implementation:    https://bitbucket.org/richardconnor/metric-space-framework/src/master/MATLAB/ NSimplexProjection/; Java implementation: https://bitbucket.org/richardconnor/metric-space-framework/src/master/src/n_ point_surrogate/

(a) Original data in 3D
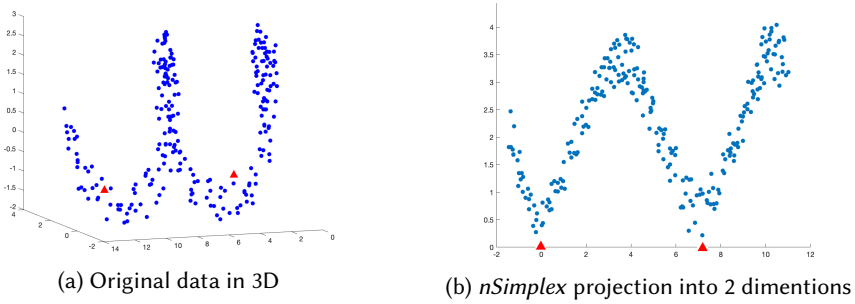
(b) *nSimplex* projection into 2 dimentions

Fig. 2. Example projection from 3D to 2D using *nSimplex*. The left figure shows some generated points roughly in a 3D spiral pattern. Two of these points (depicted with red triangles) have been randomly selected to form the reference set $\mathcal{R}$. The right figure shows the 2D projection, formed over a 1D simplex derived from the distance between these points, whose vertices are shown in red. Each other point from the 3D set has been plotted at the apex of the triangle formed from its distances to these two points.

As a simple example, Figure 2 shows a projection from a 3D Euclidean space $\mathcal{S}$ to a 2D space. In this case the reference set $\mathcal{R}$ comprises two values $r_1$ and $r_2$ selected randomly from $\mathcal{S}$, and the base simplex formed is a line segment. This is arbitrarily embedded in the 2D target space with vertex coordinates $\mathbf{v}_1 = [0, 0]$ and $\mathbf{v}_2 = [d(r_1, r_2), 0]$. Every other value $u$ from $\mathcal{S}$ is then placed into the 2D projection according to its distances from these two reference values, therefore forming for each point an apex of a triangle whose base is the line (1D simplex) formed from $\mathbf{v}_1$ and $\mathbf{v}_2$.

In fact, not quite any set of values can be used for $\mathcal{R}$. The distances within the set must be able to form a set of linearly independent points in the projected space; in most spaces this is rarely an issue for a random selection. In fact the choice made for $\mathcal{R}$ affects various aspects of the projection, and will be discussed in detail later. It may also be noted that, in Step (4), there are two possible apex points that might be formed, one on either side of the hyperplane containing the base simplex. In our example, we have aligned the base (1-dimensional) simplex with the X axis, and selected apices with a positive Y coordinate. This choice can in fact be generalised over any number of dimensions, as the base simplex can always be formed with only zero values in the $k^{th}$ dimension, i.e., it is constructed so that it lies in the hyperplane $\{[x_1, \ldots, x_k] \in \mathbb{R}^k | x_k = 0\}$. We include in Appendix B an algorithm to construct a simplex with these properties in arbitrary dimensions.

In this example, the 2D projection could be formed for any metric space, as the triangle inequality property means that it is always possible to construct the apex points of the triangles, i.e. determine a point $\mathbf{v}_u$ such that $\ell_2(\mathbf{v}_u, \mathbf{v}_1) = d(u, r_1)$ and $\ell_2(\mathbf{v}_u, \mathbf{v}_2) = d(u, r_2)$[8]. However as we will show the properties of the derived space are stronger if the domain of the transform has the Hilbert properties.

In due course we will define three functions over this $k$-dimensional coordinate space. First, however, we will introduce its important properties.

### 4.1 Properties

For a Hilbert space $\mathcal{U}$ governed by a distance function $d$, we refer to $\sigma_{\mathcal{R}} : \mathcal{U} \to \mathbb{R}^k$ as an *nSimplex* transform defined by some appropriate set $\mathcal{R}$ of $k$ reference points selected from $\mathcal{U}$. For the sake of simplicity we henceforth use the notation $\sigma$ in place of $\sigma_{\mathcal{R}}$.

The most important properties of the *nSimplex* transform are the following:

---

[8]The apex is in the intersection of a hypersphere centred in $\mathbf{v}_1$ with radius $d(u, r_1)$ and a hypersphere centred in $\mathbf{v}_2$ with radius $d(u, r_2)$. The intersection exists because $d(r_1, r_2) \le d(u, r_1) + d(u, r_2)$.

- $\sigma$ is a contraction mapping, i.e.

$$\forall u_i, u_j \in \mathcal{U}, \quad \ell_2(\sigma(u_i), \sigma(u_j)) \quad \leq \quad d(u_i, u_j)$$

- Over the same coordinate space, there exists a function $Upb$ which is an expansion mapping, i.e.

$$\forall u_i, u_j \in \mathcal{U}, \quad d(u_i, u_j) \quad \leq \quad Upb(\sigma(u_i), \sigma(u_j))$$

- Finally, there exists a further function $Zen$ which gives a value between these two, i.e.

$$\forall u_i, u_j \in \mathcal{U}, \quad \ell_2(\sigma(u_i), \sigma(u_j)) \quad \leq \quad Zen(\sigma(u_i), \sigma(u_j)) \quad \leq \quad Upb(\sigma(u_i), \sigma(u_j))$$

It can be seen from these inequalities that the $Zen$ function is a better estimator of the true distance than either $\ell_2$ or $Upb$. In fact, as we will show, the $Zen$ function acts as an excellent estimator of true distance particularly when the original space is high dimensional, allowing good estimates to be made even when these are projected onto relatively low dimensions. To give better consistency of naming, we will henceforth refer to the $\ell_2$ function as $Lwb$ when it is used in this context.

We first give definitions of the three functions, and will give a geometric explanation in the following section. Let $\mathbb{R}^k$ be a space in the co-domain of some *nSimplex* transform $\sigma$. Let the Euclidean coordinates of $\mathbf{x}, \mathbf{y} \in \mathbb{R}^k$ be given by $[x_1, x_2, \ldots, x_k]$ and $[y_1, y_2, \ldots, y_k]$ respectively. Then

$$base\_dist(\mathbf{x}, \mathbf{y}) \quad = \quad \sum_{i=1}^{k-1} (x_i - y_i)^2 \tag{5}$$

$$Lwb(\mathbf{x}, \mathbf{y}) \quad = \quad \sqrt{base\_dist(\mathbf{x}, \mathbf{y}) + (x_k - y_k)^2} \tag{6}$$

$$Upb(\mathbf{x}, \mathbf{y}) \quad = \quad \sqrt{base\_dist(\mathbf{x}, \mathbf{y}) + (x_k + y_k)^2} \tag{7}$$

$$Zen(\mathbf{x}, \mathbf{y}) \quad = \quad \sqrt{base\_dist(\mathbf{x}, \mathbf{y}) + x_k^2 + y_k^2} \tag{8}$$

Of the three functions, only $Lwb$ is a proper metric. The others are not even semimetric, as for example they do not have the identity property: i.e. $Zen(\mathbf{x}, \mathbf{x}) \neq 0$ if the last vector component is non-zero. They do, however, all possess the triangle inequality property, and so are suitable for use with metric search techniques. In fact the lack of the identity property from the $Zen$ function is actually a requirement for it to produce very good estimates when used in low dimensions.

Furthermore, it can be seen that the three functions can, if required, be evaluated efficiently as a triple, by observing that

$$lwb^2(\mathbf{x}, \mathbf{y}) + 2x_k y_k = zen^2(\mathbf{x}, \mathbf{y}) = upb^2(\mathbf{x}, \mathbf{y}) - 2x_k y_k$$

## 4.2 Geometry of the Simplex

The easiest introduction to the intuition of the $Lwb$ and $Upb$ functions is to consider first a projection into two dimensions. Although this is not the primary intended use, it is useful to illustrate principles that apply also more generally in higher dimensions with the simpler case.

Figure 3a shows two objects $u_1$ and $u_2$ from a (potentially high-dimensional) Hilbert space $(\mathcal{U}, d)$ projected into two dimensions, using two reference objects $r_1$ and $r_2$. The reference objects are used to form the one-dimensional simplex comprising the line segment $[(0, 0), (d(r_1, r_2), 0)]$ and
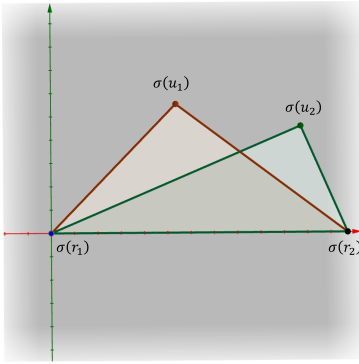
(a) A 2D *nSimplex* projection of a space $(\mathcal{U}, d)$. The projection $\sigma$ is constructed according to the reference objects $r_1, r_2 \in (\mathcal{U}, d)$. The distances $d(u, r_1)$ and $d(u, r_2)$ give a unique position in the 2D plane for any element $u \in \mathcal{U}$.
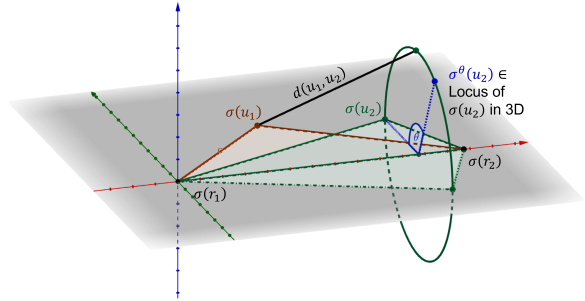
(b) From the Hilbert properties, any four objects from $(\mathcal{U}, d)$ can be isometrically embedded in a 3D space. It can be seen that the distance $\ell_2(\sigma(u_1), \sigma(u_2))$ is a lower bound of the true distance $d(u_1, u_2)$.

Fig. 3. Two-dimensional projection of two values based on two reference objects (3a), and the two possible planar tetrahedra formed by all four objects (3b).

the objects $u_1$ and $u_2$ are projected into the 2D space, each as a separate apex of the base simplex formed by this line, according to their respective distances from $r_1$ and $r_2$. The points in the 2D space are thus the projections $\sigma(u_1)$ and $\sigma(u_2)$ of the *nSimplex* transform, where $\sigma$ is a mapping $\sigma : \mathcal{U} \to \mathbb{R}^2$ defined by the two reference points $r_1$ and $r_2$.

Due to the Hilbert properties of the projection domain, any 4 values can be isometrically embedded in an 3-dimensional Euclidean space therefore there must exist a tetrahedron in 3-dimensional Euclidean space whose vertices correspond to the four objects $\{r_1, r_2, u_1, u_2\}$, and whose edge lengths correspond to the distances between each corresponding pair. Two of the faces of this tetrahedron are congruent with the two triangles illustrated in Figure 3a. Considering only the 2D projection, 5 of the 6 inter-vertex distances have been calculated and are directly available from the projection (i.e., $d(r_1, r_2)$ and $d(r_i, u_j)$, $i, j = 1, 2$). The distance $d(u_1, u_2)$ is not available. Without loss of generality the vertices of the tetrahedron are $\{\sigma(r_1), \sigma(r_2), \sigma(u_1), \mathbf{v}_{u_2}\}$, where the vertex $\mathbf{v}_{u_2} \in \mathbb{R}^3$ can be calculated only by explicitly computing $d(u_1, u_2)$. However, it is possible to put upper and lower bounds on this distance from the tetrahedral geometry which is guaranteed to exist in a 3D projection.

Figure 3b shows a third dimension added to this diagram, which can accommodate the fourth unknown vertex $\mathbf{v}_{u_2}$ of the tetrahedron. Note that this is a *hypothetical* space, in that it is not explicitly constructed by the 2D *nSimplex* projection, but only used to reason about properties of the 2D projection. We introduce the term $\sigma^\theta(u)$ to refer to the mapping of an object $u \in \mathcal{U}$ into this $(\mathbb{R}^3, \ell_2)$ space so that $\ell_2(\sigma^\theta(u), \sigma(r_i)) = d(u, r_i)$, $i = 1, 2$, while still considering the 2D projection $\sigma$. The angle $\theta$ is the angle formed by the point $\sigma^\theta(u)$ and the hyperplane in $\mathbb{R}^3$ containing the other three vertex of the tetrahedron.

The two adjacent faces of the tetrahedron share the line segment $[\sigma(r_1), \sigma(r_2)]$ as their common edge. Given that the tetrahedron must exist, it can be fully defined by the five available edge lengths in combination with the true angle $\theta^*$ between these faces, which must be somewhere in the interval $[0, \pi]$ radians. Without loss of generality, we fix the point $\sigma(u_1)$ in the $XY$ plane. The locus of the point $\sigma^\theta(u_2)$ in the higher dimension is thus restricted to the circle defined by the rotation of apex point $\sigma(u_2)$ around the $X$ axis, and its exact location within the 3D space could be determined with knowledge of the distance $d(u_1, u_2)$, or equivalently from the knowledge of the exact angle $\theta^*$.

It is clear that the lower and upper bounds of the distance $d(u_1, u_2)$ in the original space occur with the planar tetrahedra formed when the angle $\theta$ is 0 and $\pi$ radians respectively. These planar tetrahedra are contained within the 2D space of the original projection $\sigma$, and the 3-dimensional model does not need to be explicitly formed in order to establish their geometry. The tetrahedron defined by the angle $\theta = 0$ has vertices exactly as already projected. Due to the manner in which the projection is constructed, with the final coordinate of the projection representing the altitude of the apex point above the hyperplane containing the base simplex (see Appendix B for full details), the tetrahedron defined by the angle $\theta = \pi$ can be created simply by taking the negative value of the $Y$ coordinate of point $\sigma(u_2)$. These observations lead directly to the derivation of the *Lwb* and *Upb* functions as defined in Equations 6 and 7 respectively. It should be noted that these bounds apply only to projections made from general Hilbert spaces.

As noted above, any metric space can be projected into two dimensions, as the ability to perform this mapping is guaranteed by the triangle inequality property. However, the lower and upper bound properties do not hold for the 2D projection unless a stronger condition, the ability to isometrically embed any four objects into 3D Euclidean space[9], also holds in the domain of the projection.

While the intuitive argument given is valid only for the two-dimensional projection, it carries through a projection into any number of dimensions, as the Hilbert properties give the ability to isometrically embed any $k$ objects into $(k-1)$ Euclidean dimensions. It is possible, for example when $k = 3$, to rotate the apex of a tetrahedron through a fourth dimension, around the plane containing its triangular base, whilst preserving the edge lengths, but this is not so clear in terms of intuition. The general result as stated above, that the *Lwb* and *Upb* functions given in Section 4.1 are lower and upper bounds respectively of the true distance, is independent of the dimension of the projection when applied to any Hilbert space. We enclose a proof of correctness of this result in Appendix C.

### 4.3 The *Zen* function

Figure 4a is an illustration of the same 2D projection as in Figure 3, but shows the triangle $\Delta\sigma(r_1)\sigma(r_2)\sigma^\theta(u_2)$ with a different orientation in the hypothetical 3D space, while Figure 4b shows the case where this triangle is set at the angle $\pi/2$ with respect to the hyperplane $H$ containing $\sigma(r_1), \sigma(r_2), \sigma(u_1)$. The *Zen* (zenith) function is named after this last orientation, and gives the $\ell_2$ distance between the points $\sigma(u_1)$ and $\sigma^\theta(u_2)$ when $\sigma^\theta(u_2)$ is at the zenith of this circle, i.e. the point with the highest altitude above the hyperplane $H$. This distance can be simply calculated using only the information in the projection, as given in Equation 8. In this section, we explain why this function provides the best estimator for the true distance $d(u_1, u_2)$ in an original high-dimensional space.

---

[9]This is the so-called *four-point property*; it is slightly more general than Hilbert properties, and some useful non-Hilbert metric spaces possess this. We have previously defined such spaces as *supermetric* [14], and shown how general metric search techniques can be improved through its use.

(a) Four objects $r_1, r_2, u_1$ and $u_2$ are projected into two dimensions. $\sigma^\theta(u_2)$ is a hypothetical position in 3D space.

(b) Adding $\mathbf{b}$, the centre of the locus of $\sigma^\theta(u_2)$, and setting the angle $\theta = \angle\sigma(u_2)\mathbf{b}\sigma^\theta(u_2)$ to $\pi/2$.
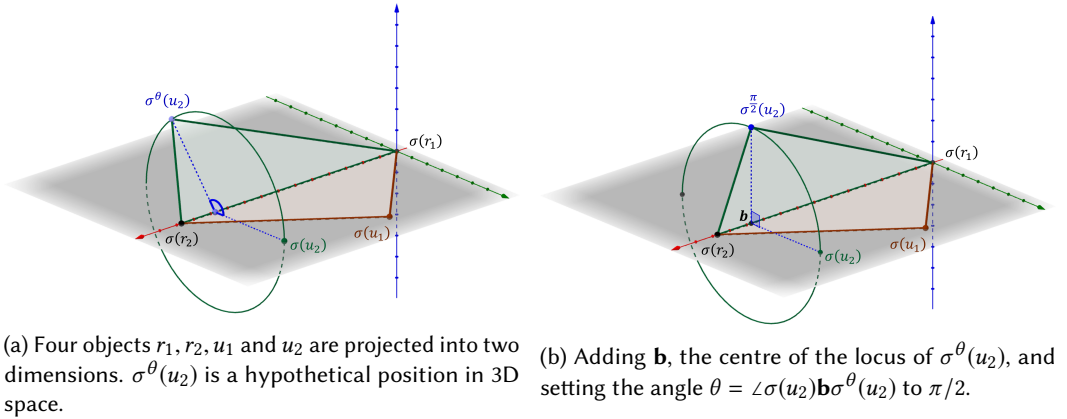
Fig. 4. The *Zen* function is defined when the angle between the two triangles is set at $\pi/2$ in the *hypothetical* further dimension. There is no requirement to calculate a projection in this dimension: $Zen(\sigma(u_1), \sigma(u_2)) = \ell_2(\sigma(u_1), \sigma^\theta(u_2))$.

Considering Figure 4 again, and all of the possible tetrahedra that could be formed from the five known and the one unknown distances, it might be supposed that there is no constraint on the particular position in which the point $\sigma^\theta(u_2)$ is most likely to lie on the circle depicted. This assumption however typifies the danger of basing intuition on low-dimensional spaces. In fact there is no absolute constraint, but there is a very significant probabilistic constraint, assuming the domain is evenly distributed, and this gets tighter as the *dimensionality* of the domain increases.

*4.3.1 Considering higher dimensions.* If the projection is onto a $k$-dimensional space, then the hypothetical space being considered is in $k + 1$ dimensions. In that case, $k$ reference objects and any two data points $u_1$ and $u_2$ are projected in $\mathbb{R}^k$ using the *nSimplex* projection $\sigma$. The vertex $\sigma^\theta(u_2) \in \mathbb{R}^{k+1}$ is obtained by rotating $\sigma(u_2)$ around the $k-2$ dimensional space containing the base simplex formed by $\{\sigma(r_1), \ldots, \sigma(r_k)\}$; the angle $\theta$ is the angle formed by $\sigma^\theta(u_2)$ and the hyperplane $H = \{[x_1, \ldots, x_{k+1}] \in \mathbb{R}^{k+1} \mid x_{k+1} = 0\}$ containing both $\sigma(u_1)$ and the simplex base. In other words, $\sigma^\theta(u_2)$ lies in the intersection of $k$ hyperspheres $B_i = \{\mathbf{v} \in \mathbb{R}^{k+1} \mid \ell_2(\mathbf{v}, \sigma(r_i)) = d(u_2, r_i)\}$ for $i = 1, \ldots, k$, that forms a circle on a plane orthogonal to the hyperplane containing the base simplex of vertices $\sigma(r_1), \ldots, \sigma(r_k)$ and the projected point $\sigma(u_1)$. The exact angle $\theta^*$ that would give $\ell_2(\sigma(u_1), \sigma^{\theta^*}(u_2)) = d(u_1, u_2)$ it is not known without explicitly calculating $d(u_1, u_2)$. However, as shown in Section 2.2, in high dimensional space the most likely value for this angle is $\pi/2$, and furthermore, as the dimension of the space increases, the variance rapidly decreases. This variance is a factor of the dimensionality of the domain, rather than the range, of the projection. With this angle set to $\pi/2$, the distance $\ell_2(\sigma(u_1), \sigma^\theta(u_2))$ in the hypothetical further dimension can be simply calculated given the projection values $\sigma(u_1)$ and $\sigma(u_2)$ in the $k$-dimensional space of the projection. This finally gives the explanation of the *Zen* formula (Equation 8) which gives this distance in the context of the projection from $m$ to $k$ dimensions.

In Appendix C we give a formal derivation of this intuitive argument in arbitrary Hilbert spaces. In particular, we show that if $\sigma : \mathcal{U} \to \mathbb{R}^k$ is the *nSimplex* transform defined by a set of $k$ reference

points then for any $u_1, u_2 \in \mathcal{U}$ given the transformed points $\mathbf{x} = \sigma(u_1)$ and $\mathbf{y} = \sigma(u_2)$ it holds

$$d(u_1, u_2) = \sqrt{\sum_{i=1}^{k-1}(x_i - y_i)^2 + x_k^2 + y_k^2 - 2x_k y_k \cos \theta} \qquad (9)$$

where $\theta$ corresponds to the angle $\angle\sigma(u_2)\mathbf{b}\sigma^\theta(u_2)$ in Figure 4b. It is clear from this form that as the probability of $\theta$ being close to $\pi/2$ increases (as happens when the dimensionality of domain $\mathcal{U}$ increases) the *Zen* function applied to the $\sigma$ projection gives an increasingly accurate estimate of the true distance $d(u_1, u_2)$.

There is one caveat here however. If $d(u_1, u_2)$ is very small, this can affect the probability of $\theta$ being close to $\pi/2$. Considering Figure 4 it can be seen that if $d(u_1, u_2)$ is close to the lower-bound, shown in the diagram by the distance $d(\sigma(u_1), \sigma(u_2))$, then $\theta$ will be significantly less than $\pi/2$. Equation 9 shows that, for small values of $\theta$, the *Zen* function will not be such a good estimator.

The implications of the general result however are quite extraordinary: it is possible to compress a space of perhaps thousands of dimensions into a very low-dimensional space of only a few dimensions, where the majority of pairwise distances are well-preserved. We demonstrate that this is in fact the case in Section 5.

## 5  EXPERIMENTAL ANALYSIS

Experimental analysis is presented in four main sections, each of which tests dimension reduction over a different class of metric space. Section 5.3 tests the different transforms against uniformly generated Euclidean spaces, and Section 5.4 uses two high-dimensional Euclidean spaces deriving from real-world applications. Section 5.5 tests two spaces governed by the Cosine metric, and Section 5.6 tests two spaces governed by the Jensen-Shannon metric.

For the first three of these sections, the mechanisms tested are: *nSimplex Zen*, PCA, MDS and RP. For Jensen-Shannon distance, where there is no coordinate space, the mechanisms tested are *nSimplex Zen* and LMDS.

First, in Sections 5.1 and 5.2 respectively the quality measures and data sets used are introduced.

All of the code used to generate our experimental results is available from https://github.com/richardconnor/dr-matlab-code.

### 5.1  Quality Measurement

Dimensionality reduction is a very generic concept, defining any mechanism whose purpose is to transform a set of values into a lower-dimensional space whilst maintaining, as far as possible, the most important aspects of the geometry of the original space. This rather general definition leaves much room for the interpretation of quality, depending on the context of use. A comprehensive survey of quality measurement techniques is given in [26]; based on this, we have picked the following measures as the most representative for the general context.

For a space $(S, d)$ which has been reduced to a lower-dimensional space $(S', \zeta)$ using a DR transform $\mathcal{T}$, we adopt the following notation and measures

$$\begin{aligned} \delta_{ij} &= d(s_i, s_j) & s_i, s_j \in (S, d) \\ \zeta_{ij} &= \zeta(s_i', s_j') & s_i', s_j' \in (S', \zeta) \quad \text{where } s_i' = \mathcal{T}(s_i) \, \forall i \end{aligned}$$

**Shepard Plots** A scatter plot of sampled distances $\delta_{ij}$ from the domain, plotted against distances $\zeta_{ij}$, which gives a simple visual impression of quality. Plots are typically overlaid with the monotonic function implied by the Kruskal Stress measurement.

Table 3. Data sets used in experiments and their outline properties. *Cosine* distance refers to the $\ell_2$ metric applied over $L_2$−normalised data, and for Jensen-Shannon distance the data is $L_1$-normalised as required.

| Data Set | Representational Dimension | Metric | Dimension of 80% variance |
|---|---|---|---|
| 100-dimensional generated | 100 | Euclidean | 80 |
| 500-dimensional generated | 500 | Euclidean | 400 |
| Twitter GloVe | 200 | Euclidean | 120 |
| MirFlickr fc6 | 4096 | Euclidean | 109 |
| ANN SIFT | 128 | Cosine | 28 |
| MirFlickr fc6 RELU | 4096 | Cosine | 1111 |
| 100-dimensional generated | 100 | Jensen-Shannon | n/a |
| MirFlickr GIST | 480 | Jensen-Shannon | n/a |

**Kruskal Stress** The Kruskal *stress1* criterion, which gives a measure of the monotonicity of the transform. This is a topological measure; stress will be zero if the DR transform is purely monotonic, independent of the actual values of $\delta_{ij}$ and $\zeta_{ij}$.

**Sammon Stress** Deriving from Sammon mapping, the Sammon stress formula is affected by the absolute differences between $\delta_{ij}$ and $\zeta_{ij}$, as well as their topological relationship.

**Quadratic Loss** A purely distance-based measure, which particularly punishes the existence of outliers in $\delta_{ij} - \zeta_{ij}$.

**Spearman Rho** A topological measure of order preservation of distances within sampled pairs of objects from the domain, essentially a measure of the likelihood that $\delta_{ij} < \delta_{ik}$ implies $\zeta_{ij} < \zeta_{ik}$.

**kNN Query Recall** Here the results of *k*NN searches in the reduced space are tested for quality against the same search performed in the original space. This aspect is not measured in [26], and we are not aware of any commonly accepted measure for testing it. Nonetheless it seems that nearest-neighbour search over the reduced space is an important use of these techniques. It is not captured by any of the quality metrics listed above, as behaviour over very small distances may differ from randomly sampled distances. We have therefore devised our own measure of recall, described in Appendix E, where discounted cumulative gain is measured over a relevance function based on rank.

Appendix E gives fuller background on all of these measurements.

## 5.2 Test data and methodology

In all cases we have used data sets that are widely available, or can be recreated using widely available software, and have at least one million elements to allow reasonable recall experiments. The data sets used and their main features are given in Table 3. Details are given in Appendix D.

In all experiments, a randomly selected subset of objects from the domain is used as a *witness*[10]. set with which to create the transforms. The witness set should be sufficiently large for the initial analysis of any manifold within which the actual data set is embedded, depending on the technique being considered, to allow the general ($\mathcal{U} \rightarrow \mathbb{R}^k$) transform to be created. The RP transform is

---

[10]or *training* set; we prefer the term *witness* in this context to denote a relatively small representative subset.
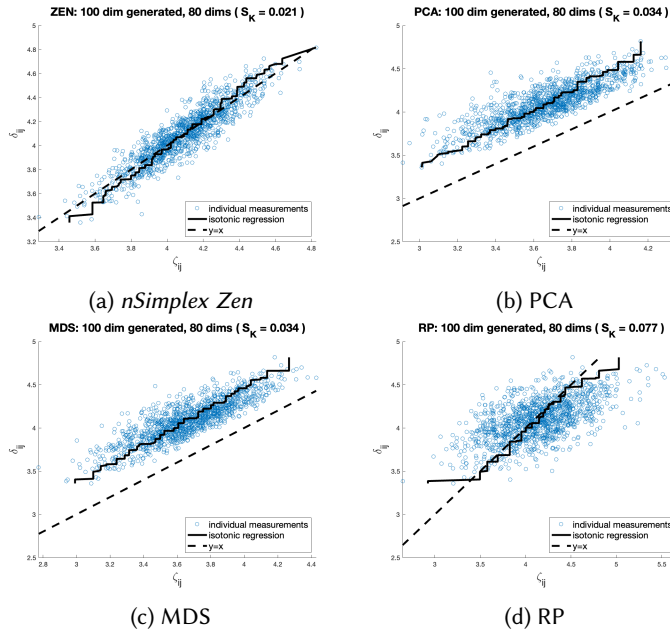
(a) *nSimplex Zen*

(b) PCA

(c) MDS

(d) RP

Fig. 5. Shepard plots for the reduction transforms, each having reduced 100-dimensional generated data to 80 dimensions. For 50 randomly selected values, all pairwise distances are plotted in both original and transformed spaces. The Y-axis represents true distance, and the X-axis is the distance measured in the reduced space. The solid black line shows the fitted least-squares monotonic regression function from which the Kruskal stress ($S_K$) is measured. It can be seen that *nSimplex Zen* and RP point clouds are centred around the true distance function ($y = x$, the dashed line), whereas PCA is a contraction mapping. While MDS gives the appearance of a contraction mapping, in fact this is not a guarantee.

created without reference to the domain, and the *nSimplex* transform is created from a set of $k$ objects randomly selected from the witness set.

For the majority of quality tests a further (non-intersecting) subset of objects is used as the domain of the transform. For Shepard plots, a subset of just 50 objects is used to avoid overcrowding the plot. A set of $10^4$ objects is used to calculate the Kruskal stress used to annotate the plot, and the other quality measures other than recall. For recall experiments, a set of $10^6$ elements is used, against which the ground truth of 1,000 nearest neighbours is calculated for 100 elements of this subset. All experiments have been multiply repeated with different random selections to ensure the results shown are representative and repeatable.

## 5.3 Generated Euclidean spaces

Uniformly distributed Euclidean spaces[11] were generated in 100 and 500 dimensions. Reduced-dimension versions were produced using RP, PCA, MDS, and *nSimplex*, and tested using the quality measures outlined in Section 5.1.

In the case of generated data, the witness set contains no useful information about the data, as there is no lower-dimensional manifold contained within the representational space. Both PCA and MDS therefore effectively apply a random projection to the experimental data. The PCA transform

---

[11]The experiments have been repeated for generated data with a Gaussian distribution, the results are not significantly different from those shown here.
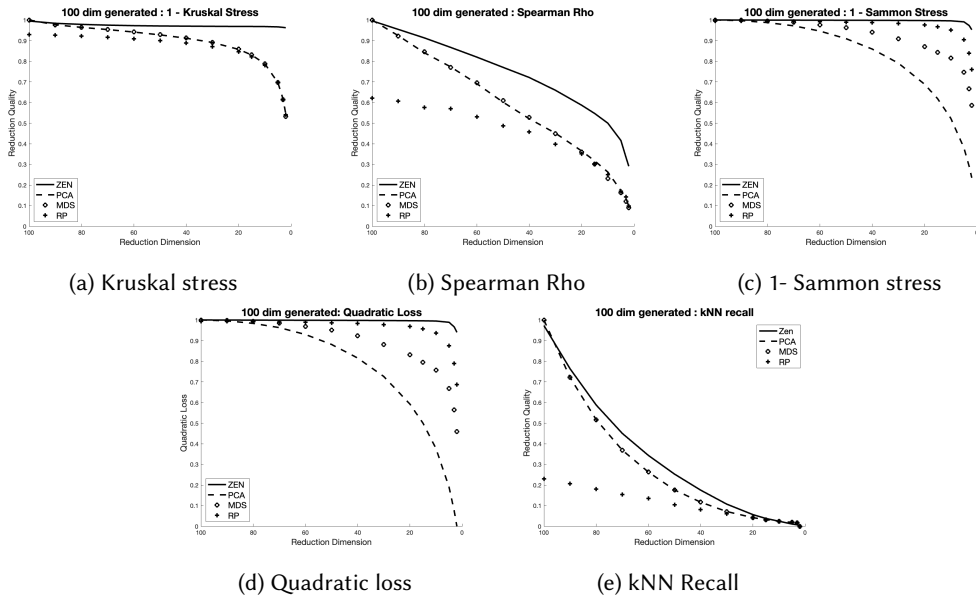
Fig. 6. Five measures of quality as the reduction dimension decreases. The X-axis in each plot is the dimension of the reduction used, in this case starting from 100 on the left and ending at 2 on the right. The Y-axis shows the measure of quality, for each transform, at each dimension. All quality measures are normalised into $[0, 1]$ to make the comparisons clearer, as recommended in [26]. For all measures, a value of 1 implies a perfect representation of the original space, a value of 0 means the transform has no effective value.

is guaranteed to be orthonormal, and while the MDS transform is not, it is always close to this given a uniform distribution of the witness data. The RP technique used in these experiments is much further from orthonormal, especially with lower dimensionalities.

It is generally perceived that there is no value in applying non-random dimensionality reduction to uniformly distributed data, but these experiments demonstrate that the *Zen* function preserves distances better than the other methods, due to the geometric model described in Section 4.3, even in the absence of a lower-dimensional manifold.

*5.3.1   100 dimensional generated space.* Figure 5 shows Shepard plots for the 100-dimensional data reduced to 80 dimensions using various reduction transforms. The reduction to 80 dimensions has been chosen as it represents the number of dimensions that explain 80% of the variance using PCA analysis and Eq. (3).

As can be seen, *nSimplex Zen* is the best transform according to the Kruskal stress criterion, giving a significantly better outcome than either PCA or MDS. In this example, PCA gives a slightly better outcome than MDS, but this is not significant. It is however significant than *nSimplex Zen* gives a better outcome: this function relies upon properties of the domain geometry which are not available to a linear transform. RP, as expected, performs the worst of the four transforms.

Figure 6 shows the outcomes of the various quality measures, as the reduction dimension is reduced from the dimension of the original data down to 2 dimensions. Each quality measure has been normalised into the range $[0, 1]$, where 1 implies a perfect outcome and 0 implies that the transform has no effective value. The expectation is that, for each measure, the outcome will start high and monotonically reduce as the dimension of the reduction decreases.
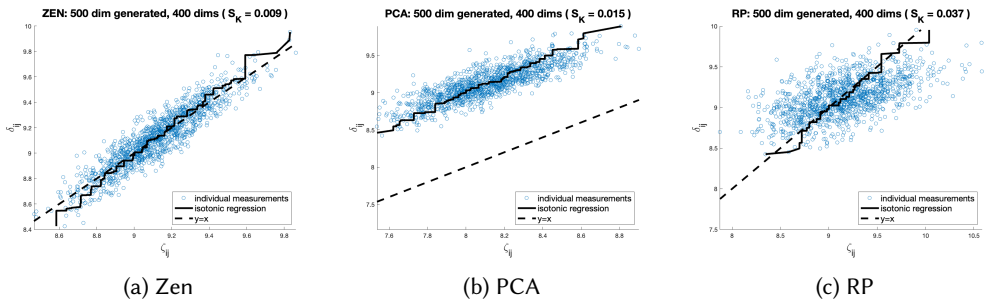
(a) Zen      (b) PCA      (c) RP

Fig. 7. Shepard plots for the DR mechanisms applied to 500-dimensional space, reduced to 400 dimensions. MDS, as before, gives very similar results to PCA, and from now on we omit that figure from this analysis.



(a) Kruskal stress      (b) Spearman Rho      (c) Sammon stress

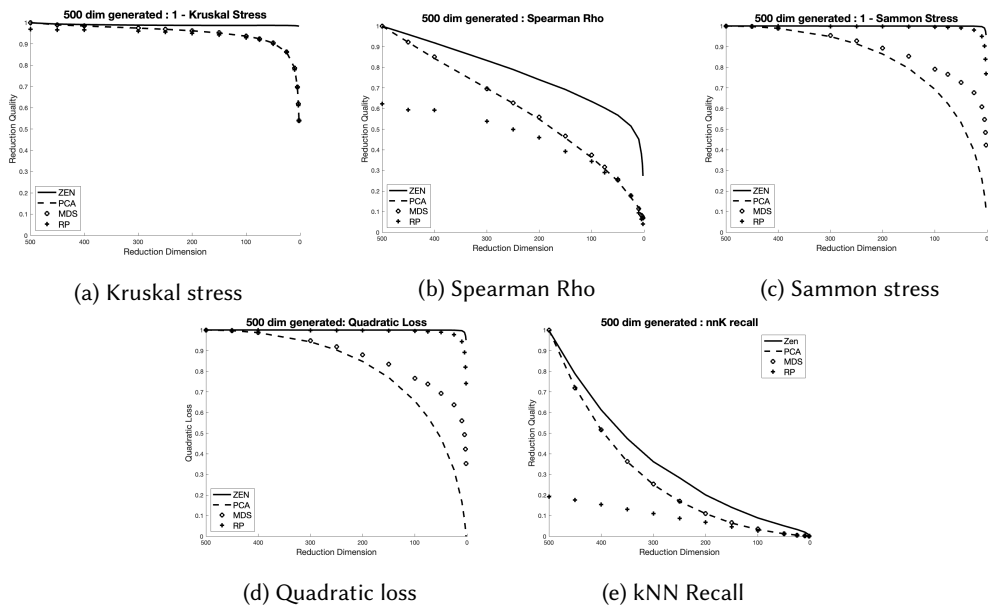(d) Quadratic loss      (e) kNN Recall

Fig. 8. Quality metrics for 500 dimensional Euclidean spaces reduced to between 500 and 2 dimensions.

It is clear that for this data set the *nSimplex Zen* transform consistently performs better than any of the other techniques for all measures and for all reduction dimensions. It is also particularly evident that the Kruskal quality of *nSimplex Zen* does not appear to significantly degrade as the reduction dimension is reduced to surprisingly low dimensions. Extraordinarily, the Kruskal stress of *nSimplex Zen* in the 2-dimensional reduction is less than that of the other techniques at 80 dimensions. This aspect will again be discussed further in Section 7.

The poor Quadratic Loss and Sammon Stress outcomes for PCA are due to the mechanism being a contraction transform, which therefore introduces a consistent error across all measurements. These quality measures punish any absolute, rather than relative, error. The *nSimplex Zen* transform gives much better results in these tests, as the underlying geometry holds the transformed distances close to those measured in the original space as seen in Figure 5.
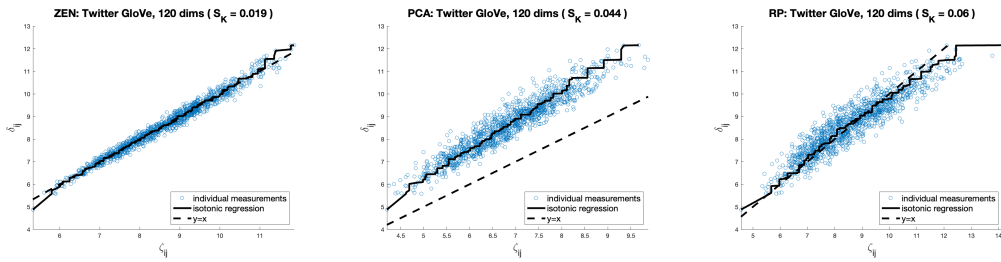
Fig. 9.  *nSimplex Zen*, PCA and RP transforms mapping GloVe from 200 to 120 dimensions. MDS gives, as before, very similar results to PCA and is omitted from the Figure.

*5.3.2   500-dimensional generated space.* We repeat the above analysis for a higher dimension space. Figure 7 gives Shepard plots for *nSimplex Zen*, PCA and RP reduced to 400 dimensions.

Even although the reduction is again to 80% of the original dimensions, it can be seen that the higher dimensions give relatively better outcomes, as predicted by Johnson-Lindenstrauss. This is particularly evident in the quality charts shown in Figure 8, where it can be further seen that, as dimensions are reduced, RP starts to give equal performance to both PCA and MDS in all quality measures, and is better for Sammon stress and Quadratic Loss. However, from our perspective, the key result is substantially better performance of the *nSimplex Zen* transform for all quality measures across all dimensions.

## 5.4   Euclidean spaces from other applications

In this section we examine the reduction of some high-dimensional Euclidean data sets produced in the application of representational techniques to real-world data. These spaces are known to lie within complex manifolds of the Euclidean space in which they are embedded, and are therefore better subjects for dimensionality reduction than the uniform spaces of the previous section.

*5.4.1   Twitter GloVe 200.* GloVe [42] is an unsupervised learning algorithm for obtaining vector representations for words, with the intent that the distance between vectors is semantically significant. Twitter GloVe is the outcome of this algorithm applied to $2 \times 10^9$ individual short texts from Twitter, from which $10^6$ individual tokens are assigned vector values. The achieved semantic similarity is quite striking, for example the closest vectors to the term *frog* are, in order: *frogs, toad, litoria* and *leptodactylidae*.

Linear substructures are also preserved, for example the relative difference between the word pair (*man, king*) is similar to that between the pair (*woman, queen*). The authors have considered both Euclidean and Cosine distances over the records and have found no significant advantage to either metric. Pre-trained word vectors with 25, 50, 100 and 200 dimensions are available online[12], here we have used the 200 dimension version.

The same experiments as above were performed on the Twitter GloVe data set. Before creating the Shepard plots, PCA was used to find the number of dimensions necessary to explain 80% of the variance (according to Eq. 3). This value is 120, significantly less than the 160 dimensions that would be required for uniform data. This was selected as the reduction dimension to illustrate using Shepherd plots in in Figure 9. As can be seen, in this context the *nSimplex Zen* transform performs by far the best of those tested, the Kruskal stress now being less than half of that obtained using PCA or MDS.

---

[12]https://nlp.stanford.edu/projects/glove_

(a) Kruskal stress

(b) Spearman Rho

(c) Sammon stress
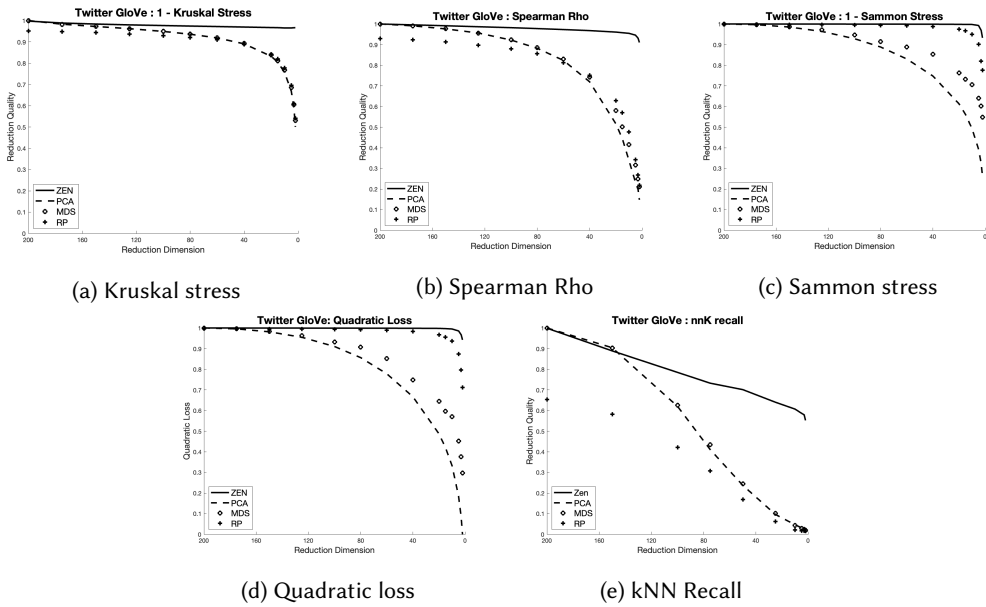
(d) Quadratic loss

(e) kNN Recall

Fig. 10. Quality metrics for Twitter GloVe reduced to between 200 and 2 dimensions.

The plots of transform quality with reducing dimensions are shown in Figure 10. Some of these results are quite startling: the *nSimplex Zen* transform is almost always best, for all quality measures, and in some cases maintains high quality values down to tiny reduction dimensions compared to the other techniques.

The reason for this relative increase in performance is, we believe, due to the nature of the manifold in which the data lie. PCA, MDS and RP all produce linear transforms of this manifold, whereas *nSimplex Zen*'s transform is non-linear, allowing it to respect the geometry of the original manifold with respect to each object mapped into the lower dimension. The *nSimplex Zen* transform shown in Figure 9 is produced with reference to only 120 reference objects, as opposed to the 1,000 objects used for PCA, and in Figure 10 the transform at each dimension is produced using only that number of reference points to represent the manifold in which the domain lies. Even for example with a random selection of only 20 reference objects, mapping to 20 dimensions, it is clear that *nSimplex Zen* performs far better than the linear transforms which use many more reference objects, even when mapped to many more dimensions.

*5.4.2 MirFlickr 1M / Alexnet.* This data derives from the AlexNet convolutional neural network [31] applied to the set of one million images available from the MirFlickr project [28]. While this network is starting to be considered a little dated, as its categorisation performance is less good than some more modern networks, the combination gives a highly available network applied to a highly available large image collection: the purpose here is just to provide a realistic set of data, with meaningful semantics, in high dimensions. The data is taken from the first fully-connected (*DeCAF, fc6*) layer of the network, after the initial convolutional layers and before the remaining fully-connected layers of the network. Euclidean distance over this representation has been shown to give an excellent proxy to image similarity even for categories of image that are not included in the original classification [23, 39]. In this experiment we apply Euclidean distance to the data as
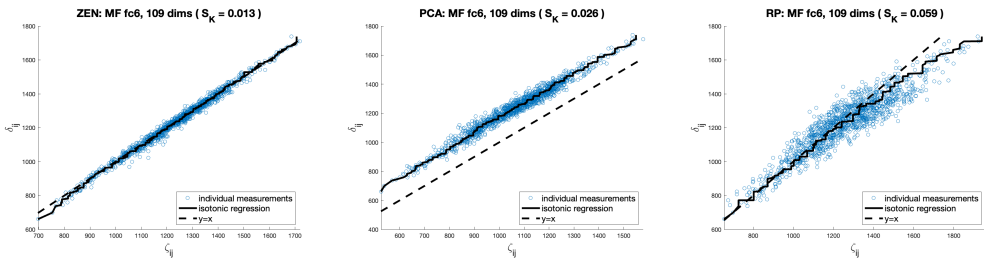
Fig. 11. MirFlickr fc6 representations, reduced from 4,096 dimensions to 109.



(a) Kruskal stress

(b) Spearman Rho

(c) Sammon stress



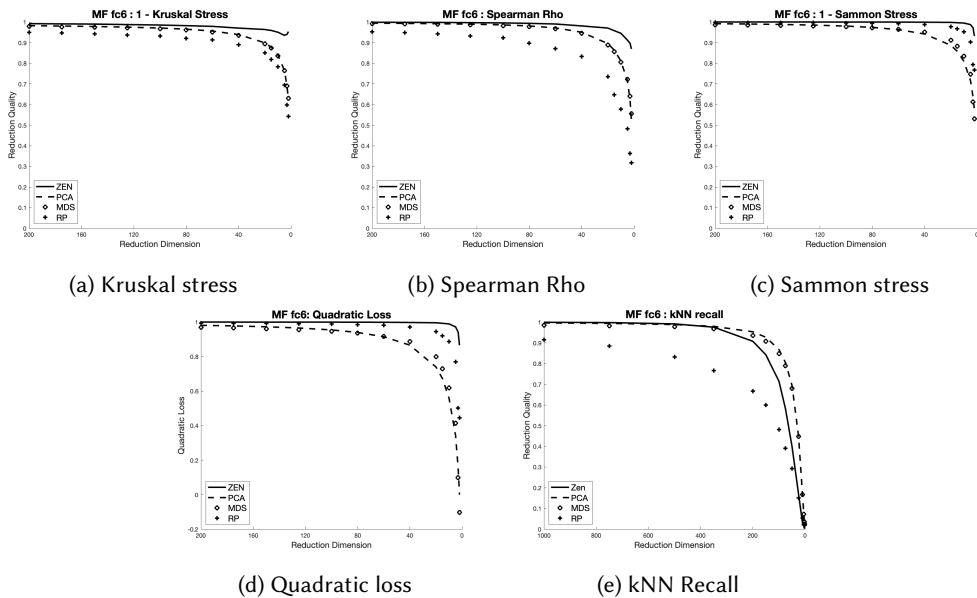(d) Quadratic loss

(e) kNN Recall

Fig. 12. Quality metrics for Mirflickr fc6. For most plots, the X-axis (dimensionality of the reduction) goes between 200 and 2, as there is very little loss of quality for any of the measurements above 200 dimensions. The recall experiment is run from 1,000 dimensions down to 2.

extracted, and in Section 5.5.2 we use the same data with Cosine distance applied to the post-RELU filtered version.

The data used for this section therefore comprises 4,096 Euclidean dimensions, including both positive and negative values, and lies within a complex manifold where the PCA eigenvalues determine that only 109 dimensions are required to explain 80% of its variance (Eq. 3). As before we compare RP, PCA and *nSimplex Zen* at this reduction dimension, the results of which are shown in Figure 11. Again MDS and PCA give almost indistinguishable outcomes for this test.

Again, it is visually evident that the *Zen* function gives a much tighter fit to the true distances than either RP or PCA, borne out by the lower value of Kruskal stress, and is generally much closer than either to the true distance.

For the first time, we show a result where the *nSimplex Zen* transform is less good than either PCA or MDS across the range of reduced dimensions: while *nSimplex Zen* remains the highest-scoring mechanism for almost all the quality measures across all reduction dimensions, Figure 12 shows

Fig. 13. ANN-SIFT $\ell_2$-normed representations, reduced from 128 dimensions to 28.

that the recall test for *nSimplex Zen* is worse for recall than either PCA or MDS when the reduction dimension is less than around 300. This is initially surprising, as the Spearman Rho measure, which tests the preservation of ordering among distances, shows a better performance for *nSimplex Zen*. The phenomenon being displayed is that *Zen*'s performance in this test is less good over very small distances, which we examine in more detail in Section 7.1.

## 5.5 Other Hilbert spaces - Cosine Distance

Cosine similarity is frequently applied over large high-dimensional spaces in the context of Information Retrieval [35]. As noted in Appendix A, the most common interpretation of Cosine distance, the complement of the normal cosine similarity (the cosine of the angle between vectors) is not a proper metric. The angle itself does give a proper metric, and can be used as a proxy which gives the same ordering within a space. To avoid the potentially expensive arccos function, Euclidean distance measured over the end-points of $\ell_2$-normalised vectors is another proper metric, with the same ordering, and which also has the Hilbert properties. While the Euclidean metric is used, such spaces are however very different from general Euclidean spaces in terms of the distribution of distances.

*5.5.1 ANN SIFT.* The ANN_SIFT1M [29] dataset comprises vectors of Angular Quantisation-based Binary Codes (AQBC) [25] deriving from the SIFT [33] feature analysis of one million images. The similarity of such representations is intended to be assessed using Cosine similarity. SIFT is no longer state-of-the-art in image similarity, but the benchmark is still widely used and provides a valuable set of data for this purpose. In these experiments the 128-dimensional data is $\ell_2$-normalised and Euclidean distance is used to provide a semantic proxy for Cosine distance.

As before, PCA and Eq. (3) are used to determine the number of dimensions required to explain 80% of the variance in distances, which turns out to be only 28 dimensions. Figure 13 shows the Shepard plots of this data for *nSimplex Zen*, PCA and RP. In this case there is little visual difference between the plots for PCA and *nSimplex Zen*, although the Kruskal score for *nSimplex Zen* is significantly better.

Figure 14 shows the quality measures for the data, for dimensions reducing from 100 down to 2 as, with the exception of RP, there is almost no quality loss at above 100 dimensions. As can be seen, in these tests *nSimplex Zen* performs best for Kruskal stress, Sammon stress and quadratic loss, but is marginally less good than either PCA or MDS at lower dimensions for Spearman Rho, and is strikingly less good than either for recall at all dimensions.

We do not have a categorical reason for these observations, but believe that there are two main reasons for this difference in performance: first, the data set to start with has a relatively low intrinsic dimensionality, and second, the data lies on a relatively regular linear manifold within the representational space. These observations are certainly consistent with the relatively small number
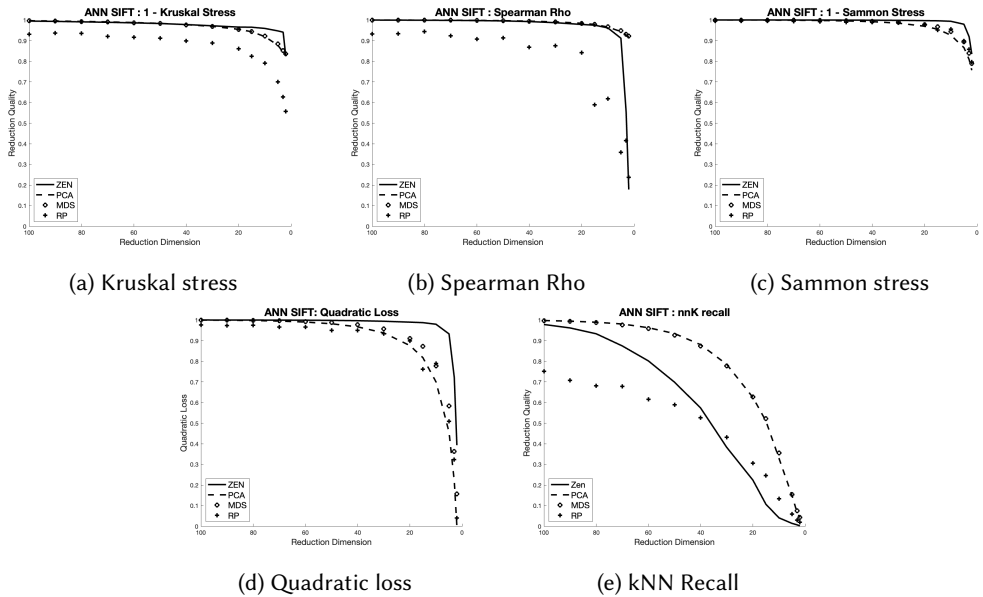
(a) Kruskal stress                           (b) Spearman Rho                           (c) Sammon stress



(d) Quadratic loss                           (e) kNN Recall

Fig. 14. Quality metrics for ANN SIFT reduced from 128 to between 100 and 2 dimensions.



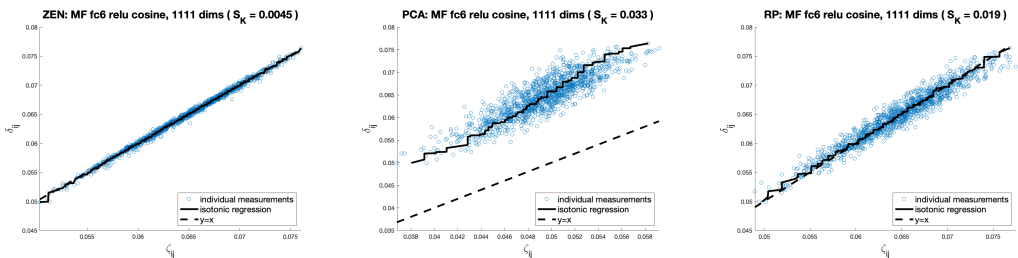Fig. 15. fc6 RELU cos $\ell_2$-normed representations, reduced from 4096 dimensions to 1111.

of dimensions shown by PCA to capture the majority of the distance variance. The relatively poor
performance of RP is also explained by these observations.

*5.5.2   MirFlickr 1m / Alexnet.* In this section we use the same raw data as that of Section 5.4.2,
but with the data converted to give a proxy for the classic "cosine" distance after the RELU filter
has been applied. RELU is applied by zeroing out the negative values, then the resulting points
are $\ell_2$-normalised by dividing each vector component by the magnitude of the resulting vector.
After this transformation, the Euclidean distance between the values, which now represent the
end-points of unit vectors, gives the same rank ordering as Cosine distance over the post-RELU
space.

The value of this metric is that it is that used for training the original network, with this transform
being applied at each fully-connected layer of the categorisation section of the CNN architecture. It
might therefore be expected to give an improved performance in terms of semantic similarity over
the simpler *DeCAF* measure, although testing this in practice for a large data set is challenging.

(a) Kruskal stress            (b) Spearman Rho            (c) Sammon stress



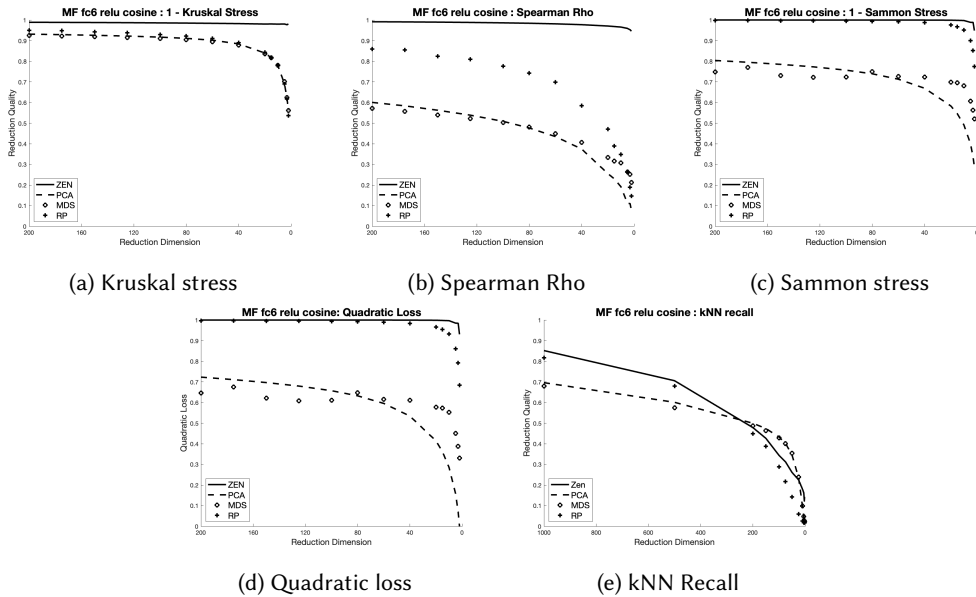(d) Quadratic loss            (e) kNN Recall

Fig. 16. Quality metrics for fc6 RELU Cosine distance over MirFlickr fc6 data. Apart from recall, all charts plot reductions from the original 4096 to between 200 and 2 dimensions, as *nSimplex Zen* gives almost perfect results at 200 dimensions. Recall is plotted from between 1000 and 2 dimensions. 1111 dimensions capture 80% of the variance when using PCA.

However, as before, our purpose for this data is to give a convincingly realistic large set of values which derive from some application and which require the use of the Cosine metric, which is thus achieved. As previously mentioned, the characteristics of the metric space produced in this way are quite different to those for the same raw data under simple Euclidean distance.

For this data the PCA eigenvalues show that $1,111$ dimensions are required to explain 80% of the variance in distance according to Eq. (3), a surprising departure from the 109 dimensions required for the non-RELU $\ell_2$ metric version. Figure 15 shows Shepard plots at this dimension. For the first time we see that RP outperforms PCA and MDS, implying that no useful information about the manifold containing this data is gleaned from either of these analyses. In fact both can in fact be seen to be harmful, as the randomly generated RP transform, which is approximately orthonormal at this higher dimension, outperforms both. In this case the relatively small stress caused by the reduction transform can only be attributed to the effect highlighted by the Johnson-Lindenstrauss lemma, and this effect is somehow being lessened by linear analysis of the original manifold.

Again, however, our main purpose is to compare the *nSimplex Zen* transform, which in this case significantly outperforms any of the other three mechanisms. As can be seen, the Kruskal stress in this case is almost an order of magnitude less than for PCA.

In fact, the high quality of the *nSimplex Zen* transform is maintained down to much lower dimensions. Figure 16 shows the first four quality measures applied to reductions of between 200 and 2 dimensions; the starting point of 200 is used as, with the exception of recall, there is almost no loss of quality with *nSimplex Zen* at any of these dimensions. In all cases it can be seen that *nSimplex Zen* is the best mechanism, along with the observation that RP is better than either PCA or MDS again across the whole range of reduction dimensions. For Kruskal stress and Spearman Rho measures, *nSimplex Zen* is much better than RP.
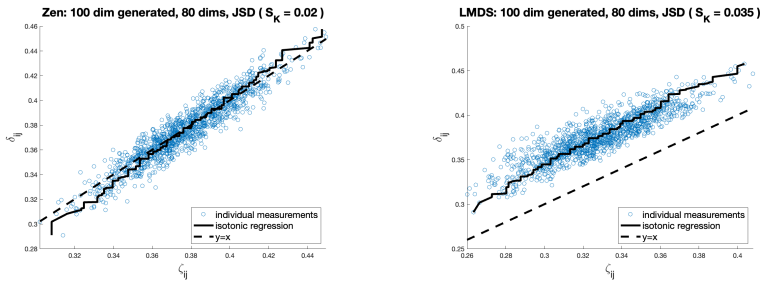
Fig. 17. 100-dimensional generated probability space with Jensen-Shannon metric, reduced to 80 dimensions using *nSimplex Zen* and LMDS.

The only exception to this is the recall measure, where it can be seen that *nSimplex Zen* starts to perform less well than either PCA or MDS when the reduction dimensions is less than around 300. While *nSimplex Zen* is better than RP across the whole range, the advantage is only relatively small; again, RP gives a surprisingly good outcome for this test.

### 5.6 Other Hilbert spaces - Jensen-Shannon Distance

In the final experimental section, we apply *nSimplex Zen* to Hilbert spaces where there is no available coordinate system. As explained in Section 4, the underlying *nSimplex* transform can be applied to any metric space which is isometrically embeddable in a Hilbert space. As the $n$-dimensional simplexes are constructed in Euclidean space using only the pairwise distances measured in the original space, then any metric space which allows a finite $n$-embedding into $(n-1)$ Euclidean dimensions can be used as the domain, and all metric spaces which are isometrically embeddable in Hilbert space have this property.

One of the most interesting classes in this category is that of metric spaces governed by the Jensen-Shannon distance, an information-theoretic distance metric which has some interesting properties, and can reasonably be regarded as a distance which should always be preferred to the semantics-free Cosine distance [16]. One possible reason for its relatively low uptake may be that its calculation requires many log calculations, and can be two orders of magnitude slower than Cosine distance over the same dimensions. It is therefore intriguing that a dimensionality reduction transform exists which not only reduces the size of the representations, but also converts the distance metric from an expensive calculation to a much cheaper one.

The absence of a coordinate system means that neither PCA nor RP can be applied. While MDS can be applied to a small space, it is not possible to use the extended version described in Section 3.3 which is necessary to allow its application to a large space. However in [44] it is shown that LMDS, a different extension of the MDS principles, can be applied to any metric space, including those without a coordinate system. In this section we therefore compare *nSimplex Zen* and LMDS in use against two spaces governed by the Jensen-Shannon metric.

*5.6.1 100-dimensional generated space.* The first experiment uses a 100 dimensional generated space. 100-dimensional vectors are generated using a uniform random generator with each dimension, bounded in [0, 1], and each vector is $\ell_1$-normed in order to simulate a probability distribution over 100 independent variables, thus giving an appropriate domain for Jensen-Shannon distance.

Figure 17 shows Shepard plots for *nSimplex Zen* and LDMS at 80 dimensions (now an arbitrary figure as PCA is not possible over the data), from which it can be seen that *nSimplex Zen* gives less Kruskal stress than LMDS.

(a) Kruskal stress  (b) Spearman Rho  (c) Sammon stress
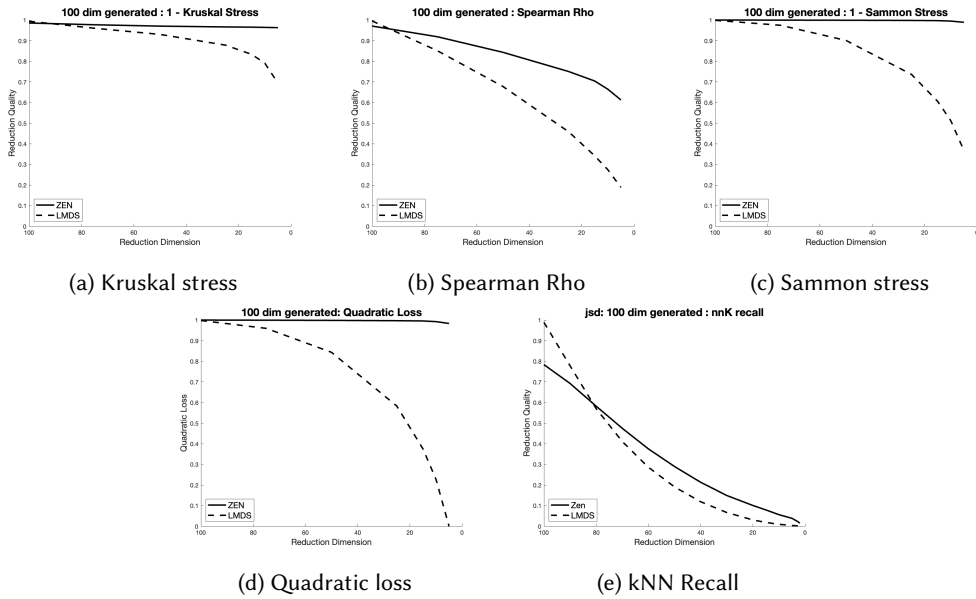
(d) Quadratic loss  (e) kNN Recall

Fig. 18. Quality metrics 100-dimensional generated probability space with Jensen-Shannon Distance.

Figure 18 shows the usual quality charts across the range of 100 down to 2 dimensions. Again, *nSimplex Zen* is generally the better of the mechanisms. It is interesting to note that *nSimplex Zen* does not give perfect results even at 100 dimensions, and that the only cases where LMDS outperforms *nSimplex Zen* are at 100 dimensions for the Spearman Rho test, and at about 80 dimensions for recall. While one property of a Hilbert space is a finite $n$-embeddability in $(n-1)$ Euclidean dimensions, this does not of course imply that an $n$-dimensional Jensen-Shannon space should in general be isometrically embeddable in an $n$-dimensional Euclidean space. Thus there is no reason to expect perfect performance when any space with Hilbert properties is "reduced" to a Euclidean space with the same physical dimensions.

*5.6.2  GIST.* Our final experiment is with Jensen-Shannon distance applied to GIST image descriptors. GIST [40] is a representation of the image based on a set of perceptual dimensions that represent the dominant spatial structure of a scene. Although again GIST is no longer the state of the art in image similarity, it has been shown that GIST representations used in conjunction with Jensen-Shannon distance gives an excellent technique for finding near-duplicate images for forensic purposes [17], a specialist application quite different from more general image similarity.

The MirFlickr 1M image set was again used, and GIST representations were obtained. Each representation is a 480-dimensional vector, again $\ell_1$-normalisation is applied to achieve a set of one million values suitable as a domain for the Jensen-Shannon distance metric.

After initial analysis it was found that these representations are quite amenable to dimensionality reduction using both *nSimplex Zen* and LMDS, so Shepard diagrams were produced at the 100-dimensional reduction. As shown in Figure 19, both techniques give a relatively low Kruskal stress even at around one-fifth or their initial size, and again the *nSimplex Zen* transform gives a significantly lower stress than LMDS.
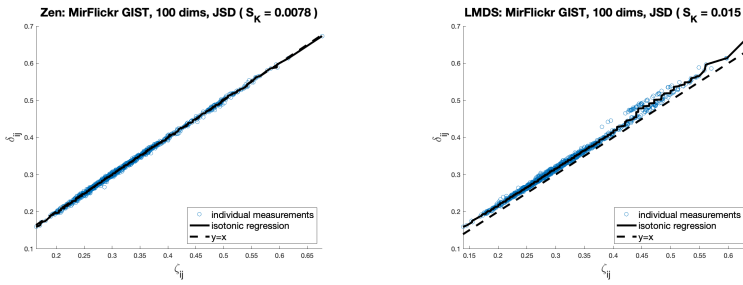
Fig. 19. MirFlickr GIST with Jensen-Shannon metric, reduced to 400 dimensions using *nSimplex Zen* and Landmark MDS.



(a) Kruskal stress

(b) Spearman Rho

(c) Sammon stress
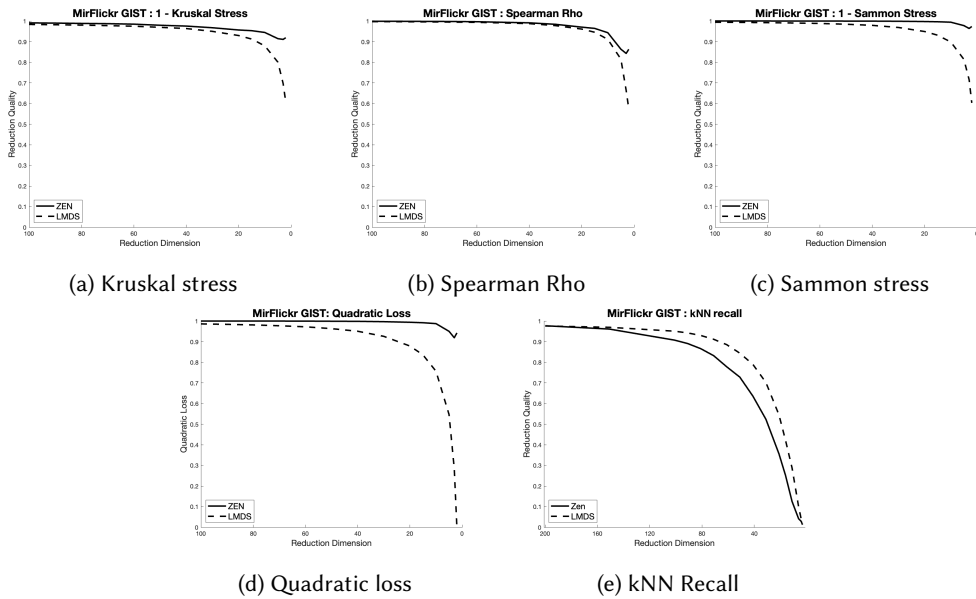
(d) Quadratic loss

(e) kNN Recall

Fig. 20. Quality metrics for GIST/JSD. The data is 480 dimensions, however there is very little quality loss when reduced to 100 dimensions, therefore most charts are plotted from 100 down to 2 dimensions; kNN recall is plotted from 200 dimensions downwards.

Figure 20 repeats the usual quality analysis over these descriptors. Most charts are from 100 down to 2 dimensions again as there is little quality loss at higher dimensions; recall is measured between 200 and 2 dimensional reductions.

For this space, *nSimplex Zen* is better across all dimensionalities for all tests other than for recall, where it is not as good as LMDS. It is noteworthy that both techniques, relying on distances alone, perform much better for the "real" data than for the uniformly generated data. The reason for this is presumably that the data is contained within a manifold contained within the representational space whose characteristics are being usefully captured by the distance-based analysis.
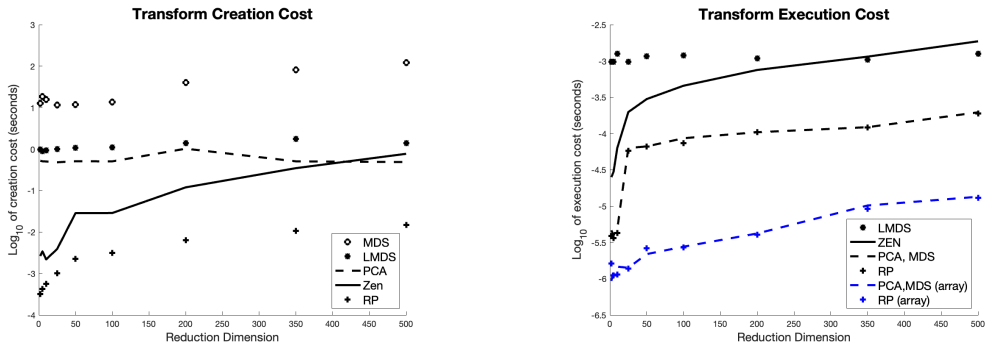
Fig. 21. Log-scaled costs of the creation and execution of the different DR transforms over a synthetic Euclidean space of 1000 dimensions.

## 6  RUN-TIME COSTS

As explained in [30], it is difficult or impossible to make fundamental comparisons on computational cost for a novel mechanism. In this case, we note that mechanisms such as PCA have had many years of study as to their optimisation, and specialist mathematical programming systems provide extremely fast versions, whereas the LMDS and *nSimplex Zen* transforms reported here are (possibly naively) coded by ourselves following the high-level definitions. However measurements of the systems as used may be pragmatically useful, although these caveats should be taken into account.

As a further caveat, the experimental results reported here are performed using MatLab[13], and some of the tests performed in Java implementations give quite different outcomes. The MatLab system is highly optimised for array manipulation, and provides optimised implementations of PCA and MDS.

There are two key aspects to the performance of the mechanisms: the cost of producing the transform, and the cost of applying it to a data item, or a data set.

For PCA, MDS and RP, the transform function comprises a matrix which is multiplied by the data in order to produce the reduced-dimensional form. So in all cases, when an $m$-dimensional Euclidean space is to be reduced to $k$ dimensions, the transform takes the shape of an $m \times k$ matrix. The data to be reduced is an $n \times m$ matrix, comprising $n$ rows of $m-$dimensional data. The outcome of multiplying the data by the transform is then an $n \times k$ matrix representing the reduced-dimension data. Although the cost of producing the transform matrix varies widely with the technique, the application to the data is thus largely constant for a given $m$ and $k$. In theory, this cost with the RP mechanism we have used throughout could be much lower, as the transform matrix is deliberately constructed to contain many zero values and thus reduce the potential cost of matrix multiplication. We have not observed this in our tests, using standard matrix multiplication as provided by MatLab, but the potential exists.

For *nSimplex Zen* and LMDS, the execution of the transform in both cases depends on distances measured between each element of the data and a set of reference objects. For LMDS this is a typically large fixed set of reference objects, whereas for *nSimplex Zen* the magnitude of the set of reference objects is the same as the reduction dimension.

Figure 21 shows a set of experimental results for the creation and execution of the different reduction transforms. The context is a generated Euclidean space of 1000 dimensions, and each

---

[13]R2022a update 4, 64-bit (Mac i64)

transform is used to reduce this to between 2 and 500 dimensions. The charts show the cost in each case of creating the transform, and the per-object cost of applying it to the data once the transform is created.

As can be seen, the cost of the transform creation varies widely, and this we believe to show fundamental differences in each approach. However in most cases the cost of transform creation is relatively unimportant compared to the cost of its execution.

The execution times show the cost of transformation per object. Two major effects are visible here; first, the mechanisms using matrix multiplication are approximately an order of magnitude faster than the mechanisms using object distances. In terms of the number of individual arithmetic operations performed, there is no such order of magnitude difference: in fact both *nSimplex Zen* and LMDS require fewer individual arithmetic operations for these examples, and the difference seen is due to the optimisation of array multiplication.

Secondly, two execution times are shown for each of PCA, MDS and RP. The greater of these apply the matrix multiplication inside an iterative loop, to give a fair comparison with *nSimplex Zen* and LMDS where this is necessary. The faster outcomes, again approximately an order of magnitude better, apply the transform via a single matrix multiplication over the entire data set as would normally be possible.

In summary, the application of any of PCA, MDS or RP is around two orders of magnitude faster than *nSimplex Zen* or LMDS, when our naive implementations are compared against a professional matrix-optimised programming system. It is possible that *nSimplex Zen* could be optimised to reach an equivalent performance, but whether this is the case or not, and indeed how to achieve it, are open questions. Meantime, our provided version of *nSimplex Zen* can perform reduction at a typical rate of between $10^{-4}s$ and $10^{-3}s$ per object, which may be fast enough for many useful purposes.

## 7 DISCUSSION

In almost all quality measures, the *nSimplex Zen* transform outperforms all of the other well-known general techniques for Euclidean spaces. As its application requires only the measurement of pairwise distances, rather than inspection of a coordinate space, it can also be applied to any metric space which is isometrically embeddable in a Hilbert space, where the necessary Euclidean properties also exist. In this context, it again outperforms LMDS in almost all measures.

There are two main reasons why the technique can perform better than other linear methods. First, it uses a well-known feature of high-dimensional spaces, namely the high probability of two sampled vectors being nearly orthogonal, to build a more accurate geometric model. Secondly, for a data set within a complex manifold, the use of a small number of sampled reference points, as opposed to a much larger number required to produce a linear transform, seems to give a better reference model for the transform. It is noteworthy that, for example, the two-dimensional reductions shown in the example are derived from a set of 1000 reference points for the PCA transform, but only two randomly-selected reference points for the *nSimplex Zen* transform; it seems scarcely credible that the latter almost always give much better outcomes.

One drawback of the technique is that the range of the *nSimplex Zen* transform is not a Euclidean space, and therefore cannot be used for low-dimensional visualisations of data. However, small projected sets can be re-modelled using MDS to produce such a visualisation if desired. The *nSimplex Zen* function does possess the triangle inequality property, and the reduced space can therefore be used with metric indexing techniques.

## 7.1 Very small distances

The *nSimplex Zen* function has been seen in some spaces to be less good at preserving ordering over very small distances than other techniques, in particular when applied to relatively low-dimensional spaces, or spaces which lie within a relatively uniform low-dimensional manifold. This is a significant drawback as it means the technique may not be the best for performing similarity search over a large reduced-dimension space. The reason for this is understood, and explained in Section 4.3. In pragmatic terms, there is an absolute lower-bound on any distance measured within the reduced space based on the altitude of the last derived component of the representative simplex.

As the simplest case, the *nSimplex Zen* distance between any object and itself, i.e. $d(u, u)$, projected into any dimension, is calculated as $\sqrt{2x_k^2}$, where $x_k$ is the value of the final-dimension coordinate in the transformed space. This *nSimplex Zen* distance may well be greater than to another object $y$ where the other components are similar and $y_k$ is coincidentally smaller than $x_k$.

In high-dimensional spaces very small distances are very rare, and the problem is more evident in the lower-dimensional spaces we have tested. The probability of it occurring however is currently beyond our full understanding; in some of our experiments it presents a problem, in others it does not. We would be reasonably optimistic that the effect could be at least partly overcome with further research. One factor that does make a difference is the choice of reference objects, which in all experiments we have reported has been random. Reference points which are mutually very close improves this particular outcome.

## 7.2 Choice of Reference Objects

As would be expected, the choice of reference objects used to construct the simplex has a significant effect on the quality of the reduction transforms.

In particular, it is possible for a pathological choice to result in the formation of a simplex which can be embedded in less than the required number of Euclidean dimensions. This does not lead to an incorrect situation, but one where the vectors comprising the individual vertex points do not form a basis for the desired projection space, therefore leading to a loss of information potential. In a high-dimensional space the probability of this happening by chance is in fact vanishingly small, and it is easy to check during simplex construction at which point a different choice of reference object can be made. The problem is only likely to occur in practice if the space is contained in a manifold whose intrinsic dimensionality is close to the dimensionality of the projection.

In all other cases, the quality of the transform can still be greatly affected by the choice of objects. We have spent some effort in seeking an optimal strategy, and have so far failed to improve, in general, on a random selection, other than when the projection is to very low dimensions. We used the random strategy in all of the reported experimental results, and consider this point as further work. In outline, other than for a very small selection of reference objects, a random choice is highly likely to reflect the properties of the manifold in which the data is contained, and thus form a natural basis for the projection of the rest of that manifold.

## 8 CONCLUSIONS

We have presented a novel dimensionality reduction technique based on a geometric model of high-dimensional metric spaces. In an extensive range of tests, it outperforms any of the other well-known general techniques for Euclidean spaces. It gives particularly good relative performance when reductions from high to low dimensions are performed. Furthermore, it can be applied to a wide range of Hilbert spaces.

While there are still many unanswered questions as to its improvement, and the optimisation of its performance, we are convinced that the *nSimplex Zen* transform provides an exciting new tool to the dimensionality reduction toolbox.

## ACKNOWLEDGMENTS

## REFERENCES

[1] Large powers of sine appear gaussian — why? https://math.stackexchange.com/questions/2293330, Accessed: 2023-01-19.

[2] Dimitris Achlioptas. Database-friendly random projections. In *Proceedings of the Twentieth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '01, page 274–281, New York, NY, USA, 2001. Association for Computing Machinery.

[3] Christian Beecks, Merih Seran Uysal, and Thomas Seidl. Signature quadratic form distances for content-based similarity. In *Proceedings of the 17th ACM International Conference on Multimedia*, MM '09, page 697–700, New York, NY, USA, 2009. Association for Computing Machinery.

[4] Avrim Blum, John Hopcroft, and Ravindran Kannan. *Foundations of Data Science*. Cambridge University Press, 2020.

[5] Avrim Blum, John Hopcroft, and Ravindran Kannan. *High-Dimensional Space*, page 4–28. Cambridge University Press, 2020.

[6] L. M. Blumenthal. *Theory and applications of distance geometry*. Clarendon Press, 1953.

[7] Leonard M Blumenthal. A note on the four-point property. *Bulletin of the American Mathematical Society*, 39(6):423–426, 1933.

[8] Michael M. Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: Going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.

[9] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML '05, page 89–96, New York, NY, USA, 2005. Association for Computing Machinery.

[10] T Tony Cai, Jianqing Fan, and Tiefeng Jiang. Distributions of angles in random packing on spheres. *Journal of Machine Learning Research*, 14:1837, 2013.

[11] Edgar Chávez, Gonzalo Navarro, Ricardo Baeza-Yates, and José Luis Marroquín. Searching in metric spaces. *ACM Comput. Surv.*, 33(3):273–321, September 2001.

[12] R. Connor, F. A. Cardillo, L. Vadicamo, and F. Rabitti. Hilbert Exclusion: Improved metric search through finite isometric embeddings. *ACM Transactions on Information Systems (TOIS)*, 35(3):17:1–17:27, December 2016.

[13] R. Connor, L. Vadicamo, F. A. Cardillo, and F. Rabitti. Supermetric search with the four-point property. In *Proceedings of 9th International Conference on Similarity Search and Applications (SISAP 2016)*, Lecture Notes in Computer Science, pages 51–64. Springer International Publishing, 2016.

[14] R. Connor, L. Vadicamo, F. A. Cardillo, and F. Rabitti. Supermetric search. *Information Systems*, 80:108–123, 2019.

[15] R. Connor, L. Vadicamo, and F. Rabitti. High-dimensional simplexes for supermetric search. In *Proceedings of 10th International Conference on Similarity Search and Applications (SISAP 2017)*, Lecture Notes in Computer Science, pages 96–109. Springer International Publishing, 2017.

[16] Richard Connor. A tale of four metrics. In Laurent Amsaleg, Michael E. Houle, and Erich Schubert, editors, *Similarity Search and Applications*, pages 210–217, Cham, 2016. Springer International Publishing.

[17] Richard Connor and Franco Alberto Cardillo. Quantifying the specificity of near-duplicate image classification functions. In *VISAPP 2016*, 2016.

[18] Richard Connor, Al Dearle, and Lucia Vadicamo. Modelling string structure in vector spaces. In *Proceedings of the 27th Italian Symposium on Advanced Database Systems (SEBD 2019)*, volume 2400. CEUR-WS.org, 2019.

[19] Richard Connor and Alan Dearle. Sampled angles in high-dimensional spaces. In *International Conference on Similarity Search and Applications*, pages 233–247. Springer, 2020.

[20] Richard Connor, Lucia Vadicamo, and Fausto Rabitti. High-dimensional simplexes for supermetric search. *CoRR*, abs/1707.08370, 2017.

[21] Michael AA Cox and Trevor F Cox. Multidimensional scaling. In *Handbook of data visualization*, pages 315–347. Springer, 2008.

[22]  Sanjoy Dasgupta. Experiments with random projection. *arXiv preprint arXiv:1301.3849*, 2013.

[23]  Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, pages 647–655. PMLR, 2014.

[24]  Karl Pearson F.R.S. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[25]  Yunchao Gong, Sanjiv Kumar, Vishal Verma, and Svetlana Lazebnik. Angular quantization-based binary codes for fast similarity search. *Advances in neural information processing systems*, 25, 2012.

[26]  Antonio Gracia, Santiago González, Victor Robles, and Ernestina Menasalvas. A methodology to compare dimensionality reduction algorithms in terms of loss of quality. *Information Sciences*, 270:1–27, 2014.

[27]  Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933.

[28]  Mark J. Huiskes and Michael S. Lew. The mir flickr retrieval evaluation. In *MIR '08: Proceedings of the 2008 ACM International Conference on Multimedia Information Retrieval*, New York, NY, USA, 2008. ACM.

[29]  Herve Jégou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1):117–128, 2011.

[30]  Hans-Peter Kriegel, Erich Schubert, and Arthur Zimek. The (black) art of runtime evaluation: Are we comparing algorithms or implementations? *Knowledge and Information Systems*, 52(2):341–378, 2017.

[31]  Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, page 1097–1105, Red Hook, NY, USA, 2012. Curran Associates Inc.

[32]  Joseph B Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, 1964.

[33]  D.G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157 vol.2, 1999.

[34]  Prasanta Chandra Mahalanobis. On the generalized distance in statistics. *Proceedings of the National Institute of Sciences (Calcutta)*, 2:49–55, 1936.

[35]  Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[36]  Jiri Matousek. *Lectures on discrete geometry*, volume 212. Springer Science & Business Media, 2013.

[37]  Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[38]  K. Menger. Untersuchungen ber allgemeine metrik. *Mathematische Annalen*, 100:75–163, 1928.

[39]  David Novak, Jan Cech, and Pavel Zezula. Efficient image search with neural net features. In *Proceedings of the 8th International Conference on Similarity Search and Applications - Volume 9371*, SISAP 2015, page 237–243, Berlin, Heidelberg, 2015. Springer-Verlag.

[40]  Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2004.

[41]  Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 2(11):559–572, 1901.

[42]  Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[43]  Aviad Rubinstein. Hardness of approximate nearest neighbor search. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2018, page 1260–1268, New York, NY, USA, 2018. Association for Computing Machinery.

[44]  Vin Silva and Joshua Tenenbaum. Sparse multidimensional scaling using landmark points. *Technology*, 01 2004.

[45]  Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[46]  Joshua B Tenenbaum, Vin de Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

[47]  B. Thomee, B. Elizalde, D. .A Shamma, K. Ni, G. Friedland, D. Poland, D. Borth, and L-J Li. YFCC100M: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73, 2016.

[48]  Lucia Vadicamo, Giuseppe Amato, and Claudio Gennaro. Induced permutations for approximate metric search. *Information Systems*, 119:102286, 2023.

[49] Lucia Vadicamo, Richard Connor, Fabrizio Falchi, Claudio Gennaro, and Fausto Rabitti. SPLX-perm: A novel permutation-based representation for approximate metric search. In *Similarity Search and Applications: 12th International Conference, SISAP 2019, Newark, NJ, USA, October 2–4, 2019, Proceedings 12*, pages 40–48. Springer, 2019.

[50] Lucia Vadicamo, Claudio Gennaro, Fabrizio Falchi, Edgar Chávez, Richard Connor, and Giuseppe Amato. Re-ranking via local embeddings: A use case with permutation-based indexing and the nsimplex projection. *Information Systems*, 95:101506, 2021.

[51] Lucia Vadicamo, Vladimir Mic, Fabrizio Falchi, and Pavel Zezula. Metric embedding into the hamming space with the n-simplex projection. In *Similarity Search and Applications: 12th International Conference, SISAP 2019, Newark, NJ, USA, October 2–4, 2019, Proceedings 12*, pages 265–272. Springer, 2019.

[52] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[53] Aaron R. Voelker, Jan Gosmann, and Terrence C. Stewart. Efficiently sampling vectors and coordinates from the n-sphere and n-ball. Technical report, Centre for Theoretical Neuroscience, Waterloo, ON, 01 2017.

[54] Wallace A Wilson. A relation between metric and euclidean spaces. *American Journal of Mathematics*, 54(3):505–517, 1932.

[55] Pavel Zezula, Giuseppe Amato, Vlastislav Dohnal, and Michal Batko. *Similarity search: the metric space approach*, volume 32 of *Advances in Database Systems*. Springer, 2006.

## APPENDICES

## A HILBERT-EMBEDDABLE DISTANCE METRICS

Distance metrics are usually referred to by name, but these names are often subject to details of context and may mean subtly different things to different readers. The following gives unambiguous definitions of metrics to which we refer in the text, all of which are isometrically embeddable in Hilbert space. In all cases we refer to a domain of vectors $\mathbf{v}, \mathbf{w} \in \mathbb{R}^n$ indexed as $v_i, w_i, \ 1 \leq i \leq n$.

### A.1 Euclidean Distance

$$\ell_2(\mathbf{v}, \mathbf{w}) = \sqrt{\sum_{i=1}^{n} (v_i - w_i)^2} \tag{10}$$

### A.2 Cosine Distance

This term is particularly problematic; in some contexts it refers to simply the complement of the cosine of the angle between vectors, which is not a proper metric; in some it refers to the angle between vectors, which is a proper metric, and in some cases it means the Euclidean distance between the $\ell_2$-normalised vectors, which is a proper, Hilbert-embeddable metric. Note that all three forms give the same rank ordering. We use the last form:

$$D_{cos}(\mathbf{v}, \mathbf{w}) = \sqrt{\sum_{i=1}^{n} \left( \frac{v_i}{\|\mathbf{v}\|} - \frac{w_i}{\|\mathbf{w}\|} \right)^2} \tag{11}$$

which is generally efficient to evaluate as it is equivalent to Euclidean distance over $\ell_2$-normalised data. Note however that the general properties of such spaces are generally very different to Euclidean spaces due to this tight constraint over the data distribution.

### A.3 Jensen-Shannon Distance

This distance is applicable only to $\ell_1$-normalised positive vectors, as it derives from a metric over probability distributions. It is defined as

$$D_{jsd}(\mathbf{v}, \mathbf{w}) = \sqrt{\mathcal{K}(\mathbf{v}, \mathbf{w})} \tag{12}$$

---

**Algorithm 1:** nSimplexBuild

---

**Input:** $n + 1$ reference points $r_1, \ldots, r_{n+1} \in (\mathcal{U}, d)$
**Output:** $n$-dimensional simplex in $\ell_2^n$ represented by the matrix $\Sigma \in \mathbb{R}^{(n+1) \times n}$

1  $\Sigma = 0 \in \mathbb{R}^{(n+1) \times n}$;
2  **if** $n = 1$ **then**
3  $\quad$ $\delta = d(r_1, r_2)$;
4  $\quad$ $\Sigma = \begin{bmatrix} 0 \\ \delta \end{bmatrix}$;
5  $\quad$ **return** $\Sigma$;
6  **end**
7  $\Sigma_{Base} = \text{nSimplexBuild}(r_1, \ldots, r_n)$;
8  $Distances = \mathbf{0} \in \mathbb{R}^n$;
9  **for** $1 \leq i \leq n$ set $Distances[i] = d(r_i, r_{n+1})$;
10 $newApex = \text{ApexAddition}(\Sigma_{Base}, Distances)$;
11 **for** $1 \leq i \leq n$ and $1 \leq j \leq i - 1$ set $\Sigma[i][j]$ to $\Sigma_{Base}[i][j]$;
12 **for** $1 \leq j \leq n$ set $\Sigma[n + 1][j]$ to $newApex[j]$;
13 **return** $\Sigma$;

---

where

$$\mathcal{K}(\mathbf{v}, \mathbf{w}) = 1 - \tfrac{1}{2} \sum_{i=1}^{n} (h(v_i) + h(w_i) - h(v_i + w_i)) \tag{13}$$

$$h(x) = -x \log_2 x \tag{14}$$

In sparse spaces the term $0 \log 0$ may occur; this is taken as 0, rather than undefined. This is a reasonable interpretation as this is the limit of the term $e \log e$ as $e$ tends to 0 from above.

### A.4 Triangular Distance

This distance is applicable only to $\ell_1$-normalised positive vectors. Its main value is as a (much cheaper and very accurate in high dimensions) estimator for Jensen-Shannon distance [16].

$$D_{tri}(\mathbf{v}, \mathbf{w}) = \sqrt{\tfrac{1}{2} \sum_{i=1}^{n} \frac{(v_i - w_i)^2}{v_i + w_i}} \tag{15}$$

In sparse spaces the term $0/0$ may occur; this is taken as 0, rather than undefined.

### A.5 Quadratic Form Distance

The Quadratic Form Distance associated to a symmetric semi-definite positive matrix $M \in \mathbb{R}^{n \times n}$ is defined as

$$D_M(\mathbf{v}, \mathbf{w}) = \sqrt{(\mathbf{v} - \mathbf{w})^T M (\mathbf{v} - \mathbf{w})} \tag{16}$$

When the matrix $M$ is diagonal the corresponding distance is a weighted Euclidean distance.

Notable examples include Mahalanobis distance [34], and the Signature Quadratic Form distance [3].

---

**Algorithm 2:** ApexAddition

---

**Input:** A $(n-1)$-dimensional base simplex and the distances between a new (unknown) apex point and the vertices of the base simplex:

$$\Sigma_{\text{Base}} = \begin{bmatrix} 0 & & & & \\ v_{2,1} & 0 & & \mathbf{0} & \\ v_{3,1} & v_{3,2} & \ddots & & \\ \vdots & & \ddots & 0 & \\ v_{n,1} & & \cdots & & v_{n,n-1} \end{bmatrix} \in \mathbb{R}^{n \times n-1}$$

$$Distances = \begin{bmatrix} \delta_1 & \cdots & \delta_n \end{bmatrix} \in \mathbb{R}^n$$

**Output:** The cartesian coordinates of the new apex point

1  $Output = \begin{bmatrix} \delta_1 & 0 & \cdots & 0 \end{bmatrix} \in \mathbb{R}^n$;
2  **for** $i = 2$ **to** $n$ **do**
3  $\quad l = \ell_2(\Sigma_{Base}[i], Output)$;
4  $\quad \delta = Distances[i]$;
5  $\quad x = \Sigma_{Base}[i][i-1]$;
6  $\quad y = Output[i-1]$;
7  $\quad Output[i-1] = y - (\delta^2 - l^2)/2x$;
8  $\quad Output[i] = +\sqrt{y^2 - (Output[i-1])^2}$;
9  **end**
10 **return** $Output$

---

## B   SIMPLEX CONSTRUCTION

This section gives an inductive algorithm (Algorithm 1) to construct a simplex in $n$ dimensions based only on the distances measured among $n + 1$ points.

For the base case of a one-dimensional simplex (i.e. two points with a single distance $\delta$) the construction is simply

$$\Sigma = \begin{bmatrix} 0 \\ \delta \end{bmatrix} \qquad (17)$$

For an $n$-dimensional simplex, where $n \geq 2$, the distances among $n + 1$ points are given. In this case, an $(n-1)$-dimensional simplex is first constructed using the first $n$ points. This simplex is used as a simplex base to which a new apex, the $(n+1)^{th}$ point, is added by the following *ApexAddition* algorithm (Algorithm 2).

For an arbitrary set of objects $s_i \in \mathcal{U}$, the apex $\sigma(s_i)$ can be pre-calculated. When a query is performed, only $n$ distances in the metric space require to be calculated to discover the new apex $\sigma(q)$ in $\ell_2^n$.

In essence, the *ApexAddition* algorithm is derived from exactly the same intuition as the lower-bound property explained earlier. Proofs of correctness for both the construction and the lower-bound property are included hereafter for the interested reader.

## C  PROOF OF CORRECTESS, *APEXADDITION* AND *NSIMPLEX LWB*

LEMMA C.1 (CORRECTNESS OF THE APEXADDITION ALGORITHM). *Let $\Sigma_{Base} \in \mathbb{R}^{n \times n-1}$ representing a $(n-1)$-dimensional simplex of vertices $\Sigma_{Base}[i] \in \ell_2^{n-1}$, with $\Sigma_{Base}[i][j] = 0$ for all $j \geq i$ and $\Sigma_{Base}[n][n-1] \geq 0$. Let $v_i$ the corresponding vertices in $\ell_2^n$ (obtained from $\Sigma_{Base}[i]$ by adding a zero to the end of the vector) and let $\delta_i$ the distance between an unknown apex point and the vertex $v_i$. Let $o = \begin{bmatrix} o_1 & \ldots & o_n \end{bmatrix}$ the output of the ApexAddition Algorithm. Then $o$ is a feasible apex, i.e. it is a point in $\mathbb{R}^n$ satisfying $\ell_2(o, v_i) = \delta_i$ for all $1 \leq i \leq n$. The last component $o_n$ is non-negative and represents the* altitude *of $o$ with respect to a base face $\Sigma_{Base}$.*

PROOF. It is sufficient to prove that the output $o = \begin{bmatrix} o_1 & \ldots & o_n \end{bmatrix}$ of the Algorithm 2 has distance $\delta_i$ from the vertex $v_i$, i.e. satisfies the following equations

$$\begin{cases} o_1^2 + \cdots + o_n^2 = \delta_1^2 & (18.1) \\ \quad\quad \vdots \\ \sum_{j=1}^{i-1}(v_{i,j} - o_j)^2 + \sum_{j=i}^{n} o_j^2 = \delta_i^2 & (18.i) \\ \quad\quad \vdots \\ \sum_{j=1}^{n-1}(v_{n,j} - o_j)^2 + o_n^2 = \delta_n^2 & (18.n) \end{cases} \quad (18)$$

Note that the $i$-th component of the output $o$ is updated only at the iteration $i$ and $i+1$ of the *ApexAddition* Algorithm. So, if we denote with $o^{(i)}$ the output at the end of iteration $i$ we have:

$$o^{(1)} = \begin{bmatrix} \delta_1 & 0 & \ldots & 0 \end{bmatrix} \quad (19)$$

$$o_i = o_i^{(h)}, \quad o_n = o_n^{(n)}, \quad o_h^{(i)} = 0 \quad\quad\quad 1 \leq i < h \leq n \quad (20)$$

$$o_{i-1} = o_{i-1}^{(i-1)} - \frac{\delta_i^2 - \sum_{j=1}^{i-2}(v_{i,j} - o_j)^2 - (v_{i,i-1} - o_{i-1}^{(i-1)})^2}{2v_{i,i-1}} \quad\quad 2 \leq i \leq n \quad (21)$$

$$(o_{i-1})^2 = (o_{i-1}^{(i-1)})^2 - (o_i^{(i)})^2 \quad\quad\quad 1 \leq i \leq n-1 \quad (22)$$

By combining Eq. (20) and (22) we obtain $\sum_{j=i}^{n} o_j^2 = (o_i^{(i)})^2$ for all $1 \leq i \leq n-2$, and so Eq. (18.1) clearly holds (case $i = 1$). Moreover, it follows that $o$ satisfies Eq. (18.i) for all $i = 2, \ldots, n$:

$$\sum_{j=1}^{i-1}(v_{i,j} - o_j)^2 + \sum_{j=i}^{n} o_j^2 = v_{i,i-1}^2 - 2v_{i,i-1} o_{i-1} + \sum_{j=1}^{i-2}(v_{i,i-1} - o_j)^2 + (o_{i-1}^{(i-1)})^2 \overset{(21)}{=} \delta_i^2$$

□                                                                               □

LEMMA C.2 (N-SIMPLEX DISTANCE CONSTRAINT). *Let $(\mathcal{U}, d)$ a space $(n+2)$-embeddable in $\ell_2^{n+1}$. Let $r_1, \ldots, r_n \in \mathcal{U}$ and, for any $m \leq n$, let $\sigma_m$ the $(m-1)$-dimensional simplex generated from $r_1, \ldots, r_m$ by using the nSimplexBuild Algorithm. For any $x \in \mathcal{U}$, let $x^{(m)} \in \ell_2^m$ the apex point with distance $d(x, r_1), \ldots, d(x, r_m)$ from the vertices of $\sigma_m$, computed using the ApexAddition Algorithm. Then for all $q, s \in \mathcal{U}$,*

(1) $\ell_2^{m-1}(s^{(m-1)}, q^{(m-1)}) \leq \ell_2^{m}(s^{(m)}, q^{(m)})$             *for*   $2 \leq m \leq n$

(2) $g(s^{(m-1)}, q^{(m-1)}) \geq g(s^{(m)}, q^{(m)})$             *for*   $2 \leq m \leq n$

(3) $\ell_2^{n}(s^{(n)}, q^{(n)}) \leq d(s, q) \leq g(s^{(n)}, q^{(n)})$

*where, for any $k \in \mathbb{N}$, $g : \ell_2^k \rightarrow \ell_2^k$ is defined as $g(x, y) = \sqrt{\sum_{i=1}^{k-1}(x_i - y_i)^2 + (x_k + y_k)^2}$.*

PROOF. By construction, for any $m \leq n$ we have

$$x_i^{(m)} = x_i^{(m-1)} \qquad\qquad\qquad i = 1, \ldots, m-2 \qquad (23)$$

$$x_i^{(i)} \geq 0 \qquad\qquad\qquad\qquad\quad i = 1, \ldots, m \qquad (24)$$

$$(x_{m-1}^{(m)})^2 + (x_m^{(m)})^2 = (x_{m-1}^{(m-1)})^2 \qquad (25)$$

Condition 1 directly follows from Eq. (23)-(25):

$$
\begin{aligned}
\ell_2^m(\mathbf{s}^{(m)}, \mathbf{q}^{(m)})^2 &= \ell_2^{m-1}(\mathbf{s}^{(m-1)}, \mathbf{q}^{(m-1)})^2 - (s_{m-1}^{(m-1)} - q_{m-1}^{(m-1)})^2 + \sum_{i=m-1}^{m} (s_i^{(m)} - q_i^{(m)})^2 \\
&= \ell_2^{m-1}(\mathbf{s}^{(m-1)}, \mathbf{q}^{(m-1)})^2 + 2\Bigg[ -s_{m-1}^{(m)} q_{m-1}^{(m)} - s_m^{(m)} q_m^{(m)} \\
&\qquad\qquad + \sqrt{(s_{m-1}^{(m)})^2 + (s_m^{(m)})^2} \sqrt{(q_{m-1}^{(m)})^2 + (q_m^{(m)})^2} \Bigg] \\
&\geq \ell_2^{m-1}(\mathbf{s}^{(m-1)}, \mathbf{q}^{(m-1)})^2
\end{aligned}
$$

where the last passage follows from the Cauchy–Schwarz inequality[14].

Similarly, Condition 2 also holds:

$$
\begin{aligned}
g(\mathbf{s}^{(m)}, \mathbf{q}^{(m)})^2 &= g(\mathbf{s}^{(m-1)}, \mathbf{q}^{(m-1)})^2 + 2\Bigg[ -s_{m-1}^{(m)} q_{m-1}^{(m)} + s_m^{(m)} q_m^{(m)} \\
&\qquad\qquad - \sqrt{(s_{m-1}^{(m)})^2 + (s_m^{(m)})^2} \sqrt{(q_{m-1}^{(m)})^2 + (q_m^{(m)})^2} \Bigg] \\
&\leq g(\mathbf{s}^{(m-1)}, \mathbf{q}^{(m-1)})^2.
\end{aligned}
$$

Now we prove that $\ell_2^n(\mathbf{s}^{(n)}, \mathbf{q}^{(n)})$ and $g(\mathbf{s}^{(n)}, \mathbf{q}^{(n)})$ are, respectively, a lower bound and an upper bound for the actual distance $d(s, q)$. The main idea is using the simplex $\sigma_n$ spanned by $r_1, \ldots, r_n$ as a base face to build the simplex $\sigma_{n+1}$ spanned by $r_1, \ldots, r_n, s$ and then use the latter as base face to build the simplex $\sigma_{n+2}$ spanned by $r_1, \ldots, r_n, s, q$. In this way, we have an isometric embedding of $r_1, \ldots, r_n, s, q$ into $\ell_2^{n+1}$ that is the function that maps $r_1, \ldots, r_n, s, q$ into the vertices of $\sigma_{n+2}$. So, given the base simplex $\sigma_n$ (represented by the matrix $\Sigma_n$), and the apex $\mathbf{s}^{(n)}, \mathbf{q}^{(n)} \in \ell_2^n$ we have that the simplex $\sigma_{n+2}$ is represented by

$$
\Sigma_{n+2} = \left[
\begin{array}{ccc|cc}
 & & & & \\
 & \Sigma_n & & & 0 \\
 & & & & \\
\hline
s_1^{(n)} & \cdots & s_{n-1}^{(n)} & s_n^{(n)} & 0 \\
q_1^{(n)} & \cdots & q_{n-1}^{(n)} & q_n^{(n+1)} & q_{n+1}^{(n+1)}
\end{array}
\right] \in \mathbb{R}^{n+2 \times n+1} \qquad (26)
$$

where, by construction, $(q_{n+1}^{(n+1)})^2 = (q_n^{(n)})^2 - (q_n^{(n+1)})^2$, $s_n^{(n)}, q_{n+1}^{(n+1)} \geq 0$, and $d(q, s)$ equals the Euclidean distance between the two last rows of $\Sigma_{n+2}$.

---

[14] Cauchy–Schwarz inequality in two dimension is: $(a_1 b_1 + a_2 b_2)^2 \leq (a_1^2 + a_2^2)(b_1^2 + b_2^2) \ \forall a_1, b_1, a_2, b_2 \in \mathbb{R}$, which implies

$$(a_1 b_1 + a_2 b_2) \leq \sqrt{(a_1^2 + a_2^2)} \sqrt{(b_1^2 + b_2^2)} \quad \forall a_1, b_1, a_2, b_2 \in \mathbb{R}$$

It follows that

$$d(q,s)^2 = \sum_{i=1}^{n-1} (s_i^{(n)} - q_i^{(n)})^2 + (s_n^{(n)})^2 + (q_n^{(n)})^2 - 2s_n^{(n)} q_n^{(n+1)}; \tag{27}$$

and, since $q_n^{(n)} \geq |q_n^{(n+1)}|$, we have

$$d(q,s)^2 = \ell_2^n(\mathbf{s}^{(n)}, \mathbf{q}^{(n)})^2 + 2s_n^{(n)}(q_n^{(n)} - q_n^{(n+1)}) \geq \ell_2^n(\mathbf{s}^{(n)}, \mathbf{q}^{(n)})^2,$$

and

$$d(q,s)^2 = g(\mathbf{s}^{(n)}, \mathbf{q}^{(n)})^2 - 2s_n^{(n)}(q_n^{(n)} + q_n^{(n+1)}) \leq g(\mathbf{s}^{(n)}, \mathbf{q}^{(n)})^2$$

Finally, we observe that since $(q_{n+1}^{(n+1)})^2 + (q_n^{(n+1)})^2 = (q_n^{(n)})^2$, $q_{n+1}^{(n+1)} \geq 0$, and $q_n^{(n)} \geq 0$, there exists an angle $\theta \in [0, \pi]$ such that

$$\begin{cases} q_n^{(n+1)} & = q_n^{(n)} \cos \theta \\ q_{n+1}^{(n+1)} & = q_n^{(n)} \sin \theta \end{cases} \tag{28}$$

Therefore, Eq. 27 can be rewritten as

$$d(q,s)^2 = \sum_{i=1}^{n-1} (s_i^{(n)} - q_i^{(n)})^2 + (s_n^{(n)})^2 + (q_n^{(n)})^2 - 2s_n^{(n)} q_n^{(n)} \cos \theta;$$

In other words, if $\sigma : D \to \mathbb{R}^n$ is the *nSimplex* transform defined by a set of $n$ reference points then for any $s, q \in D$ given the transformed points $\mathbf{x} = \sigma(s)$ an $\mathbf{y} = \sigma(q)$ it holds

$$d(q,s) = \sqrt{\sum_{i=1}^{n-1} (x_i - y_i)^2 + x_n^2 + y_n^2 - 2x_n y_n \cos \theta} \tag{29}$$

□

## D DATA SETS USED IN EXPERIMENTS

While all the software used in experiments described is available from https://github.com/richardconnor/dr-matlab-code, the data sets are typically too large to provide conveniently and we therefore provide brief descriptions of their provenance.

**Generated uniform data** All generated data is created using the MatLab *rand* function from the Statistics and Machine Learning toolbox. For example a set of one thousand objects of one hundred dimensions is created by the single line

```
data = rand(1000,100);
```

**Twitter GloVe** The GloVe data used derives from https://nlp.stanford.edu/projects/glove_ where the data and instructions for downloading it can be found. We used the 200-dimensional vectors.

**MF1M** The images from which this data derives are available from https://press.liacs.nl/mirflickr/mirdownload.html. We use the one million image set. For input to AlexNet, whole images were reduced to 227 x 227 using ImageMagick.

The 4096-dimensional vectors were obtained by applying the MatLab release of AlexNet, which is also available from other domains in other languages. In MatLab, the fc6 layer used is simply extracted by code such as

```
fc6 = activations(net,thisImage,"fc6","OutputAs","rows");
```

**ANN SIFT** The ANN SIFT data, along with code to extract it, is available from http://corpus-texmex.irisa.fr/.

**GIST** Again the Mir Flickr one million image collection was used to produce the GIST data. Although not fully documented, GIST representations of the images are also available from https://press.liacs.nl/mirflickr/mirflickr1m.v3b/; alternatively MatLab code to create GIST descriptors is available at https://people.csail.mit.edu/torralba/code/spatialenvelope/

# E MEASURING THE QUALITY OF DIMENSIONALITY REDUCTION

## E.1 Global Structure

**Shepard Diagrams** give a visual overview of the quality of a transform. For a given set of data all distances $\delta_{ij}$ are plotted against the reduced dimensional distances $\zeta_{ij}$. In most general terms, the closer the plot lies to the $y = x$ diagonal, the better the reduction.

**Kruskal Stress** To quantify the visual effect, Shepard diagrams are usually overlaid with an isotonic regression function calculated from the original and reduced spaces, as used to calculate Kruskal's stress function. The stress function is given by:

$$S_K = \sqrt{\frac{\sum_{i<j}(\zeta_{ij} - d_{ij}^*)^2}{\sum_{i<j}\zeta_{ij}^2}} \tag{30}$$

where $d^*$ is the value given by a function implied from the least-squares isotonic regression calculated from the finite data presented in the chart.

The point of fitting an isotonic regression is that the stress function assigns a value according to the monotonicity of the reduction transform, rather than the absolute values produced. PCA, for example, is a reduction mapping: in all cases, $\zeta_{ij} \leq \delta_{ij}$. However if the function $\tau : \delta_{ij} \to \zeta_{ij}$ is perfectly monotonic, the stress will be zero despite the reduction in individual values, even if this is non-linear.

## E.2 Distance Preservation

**Sammon Stress** derives from Sammon Mapping, a non-linear dimensionality reduction technique similar to MDS which minimises the stress function:

$$S_S = \frac{1}{\sum_{i<j}\delta_{ij}} \sum_{i<j} \frac{(\delta_{ij} - \zeta_{ij})^2}{\delta_{ij}} \tag{31}$$

Unlike Kruskal's stress function, Sammon stress is affected by the absolute differences between $\zeta_{ij}$ and $\delta_{ij}$ rather than their isotonic relationship, and so gives a further useful perspective on the quality of a reduction transform.

**Quadratic Loss** is a purely distance-based technique, the quadratic function used to punish the production of outliers:

$$S_Q = \sum_{i<j}(\delta_{ij} - \zeta_{ij})^2 \tag{32}$$

In fact we view this function as somewhat of a blunt instrument, as it punishes even regular deviations from original distances. For example, PCA is a contraction function, while RP is not; RP often performs better for quadratic loss even when the quality of the reduction is clearly, overall, lower. Even if the absolute deviation from original distance is important, it may often be possible to apply a scaling function to the reduced space in order to minimise this.

As discussed in Appendix E.4, we require all quality measures to be bounded in $[0, 1]$ to allow visual comparisons over different reduction dimensions. There is no natural upper-bound on this measure, so we simply convert an outcome of $q$ to $\frac{q_{max} - q}{q_{max}}$ to produce the desired range, where $q_{max}$ is the greatest value obtained in the context of the visualisation.

## E.3 Topology Preservation

**Spearman Rho** measures the preservation of rank ordering among pairwise distances measured between corresponding objects in the domain and range of the reduction transform. This is a useful measure for many applications, such as nearest-neighbour analysis, where the absolute distances among values are of no interest other than for the ordering which they induce over other elements of the set.

Pairwise distances from a sample set of $n$ objects are used to construct an ordering $z$ of size $T = \binom{n}{2}$ of pairs $\delta_{ij}$. A ranking $z'$ is then created according to the relative distances of the same pairs of objects after the transform is applied. The Spearman Rho function is then given as

$$S_R = 1 - \frac{6 \sum_{i=1}^{T} (z(i) - \hat{z}(i))^2}{T^3 - T} \tag{33}$$

where the adjusting factors combine to give an output in the range $[-1, 1]$ where 1 implies a perfect preservation of distance ordering and $-1$ implies the inverse correlation.

There are other forms of this formula, but we choose this one to faithfully follow the exact methodology of [26]. Although all of our other quality measures are normalised into $[0, 1]$, the fact that the outcome of zero implies an effectively random ordering suits our purpose in this respect.

**kNN Query Recall** When the purpose of dimension reduction is to speed up similarity search, the most important outcome is the nearest-neighbour topology. To an extent this is tested by both Kruskal stress and Spearman Rho quality measures, but with the crucial difference that in the context of query recall it is only the smallest distances, relative to a query, that are relevant. Thus, a transform which preserves very small distances well, but is less good over larger distances, will be preferable to one which preserves distances overall, although the latter may score better in these quality measures.

To measure this quality, it is necessary to construct a nearest-neighbour ground truth over a representative sample of queries for a large data set, and then compare the nearest neighbours of those queries in the reduced-dimension space. One problem is that the pattern of nearest neighbours depends on specific details of the space being considered, as well as more general properties of the reduction. The larger the space, the smaller the nearest-neighbour distances will be, and the higher the probability of having very close matches which are not representative of the general space.

To overcome some of these issues we measure recall using the following assumptions:
(1) only a small percentage of the true nearest neighbours are of any significance
(2) the nearer true neighbours are considerably more significant than the further neighbours
(3) preservation of order in the results is also important

Our recall measurement therefore uses a discounted cumulative gain (DCG) function over a relevance function based on nearest-neighbour rank.

The ranking function is constructed to give significantly higher importance to the closer neighbours by using the logistic function to give an inverse sigmoid function over rank. In our experiments we have collected 1,000 nearest neighbours from a collection of one

million data, and rank the relevance of each true nearest neighbour as

$$R_i = 1 - \frac{1}{\left(1 + e^{-\frac{i-500}{100}}\right)} \tag{34}$$

for the $i$th true nearest neighbour.

We then compare the 1,000 nearest neighbours returned by the DR function using the DCG variant defined in [9]:

$$DCG_{DR} = \sum_{i}^{1000} \frac{2^{DR_i} - 1}{\log_2 i + 1} \tag{35}$$

where $DR_i$ is $R_i$ applied to the position in the true nearest neighbours of the object found in the $i$th position of the nearest neighbours according to the DR transform.

Finally, this function produces an arbitrary maximum value of 66.0435 when lists of length 1,000 are in perfect correlation, and so the outcome is divided by this factor to give a normalised value in the range $[0, 1]$, where 0 means there is no overlap between the lists and 1 means they are in perfect correlation.

### E.4 Quality profiles

As discussed in [26], it is instructive to consider the quality of transforms as a profile over different reduction dimensions. This may be shown as a plot where one or more of the numeric quality functions is plotted against the dimension of the reduction, typically as this is reduced from the original dimensionality of the original domain down to 2. For most mechanisms and useful quality measures, this will result in a monotonic decreasing plot, and will allow the selection of the most useful compromise in terms of quality loss for a given reduction dimension.

To allow presentation of all quality measures within the same bounds, Kruskal and Sammon and stress measurements are subtracted from 1 to give a quality rather than a stress measure, and the results in the range $[0, 1]$ are given. A negative value can arise from either Spearman Rho (which is bounded in $[-1, 1]$ or Sammon stress (which has no formal upper bound) but in reality a value of less than 0 for Spearman Rho, or greater than 1 for Sammon stress, effectively means that the transform has no practical value and a zero quality rating is reasonable. For quadratic loss there are no natural bounds, and this is handled as explained in Appendix E.2.