

# An Interpretable Data Embedding under Uncertain Distance Information

Nikolaos M. Freris

University of Science and Technology of China  
China  
nfr@ustc.edu.cn

Michalis Vlachos

University of Lausanne  
Switzerland  
michalis.vlachos@unil.ch

Ahmad Ajalloeian

University of Lausanne  
Switzerland  
ahmad.ajalloeian@unil.ch

**Abstract**—A common assumption in embedding methodologies is the availability of exact pairwise distances. In this paper, we propose a 2D embedding that overcomes this limitation. It can operate on distances that are represented as a range of lower and upper bounds. Such bounds are typically available when objects are compressed, whence our approach is highly applicable in the case of big compressed datasets. We establish linear convergence (i.e., exponential decay of distance to optimality) for the proposed scheme, with a rate characterized by the topology of the data graph. We compare with prevalent embedding methodologies (ISOMAP, t-SNE, MDS) and illustrate that our approach can provide faithful preservation of distances, correlations, and object ranks, even in the presence of inexact distance information.

**Index Terms**—Dimensionality Reduction, Visualization

## I. INTRODUCTION

Dimensionality reduction is instrumental for processing large volumes of data. Embedding algorithms map high-dimensional data into low-dimensional representations so as to alleviate the run-time of machine learning operations, while aiming to retain object relations between the original data. We present a two-dimensional (2D) embedding, i.e., the mapping of a high-dimensional dataset onto the Euclidean plane; a notable application being data visualization. The literature comprises a wide range of dimensionality reduction methods: Multi-Dimensional Scaling (MDS) [1], ISOMAP [2], Locally Linear Embedding (LLE) [3], t-distributed Stochastic Neighbor Embedding (t-SNE) [4], random projections [5], and Uniform Manifold Approximation and Projection (UMAP) [6] enlist some of the most commonly used techniques. All of these operate using *exact* distance information between data pairs.

A distinct trait of the proposed embedding method is that it can accommodate *inexact* distance estimates between object pairs, represented by lower and upper bounds. In specific, our method has access only to distance *ranges* between pairs of objects. Such distance estimates can be derived, for example, when data are represented and compressed in a *lossy* manner using orthonormal transforms (Fourier, wavelets, etc.) [7], or compressed sensing methodologies [8]. In such scenarios, exact distances between pairs of objects are no longer computable, but upper/lower bounds can be estimated instead; cf. Fig. 1 for a visual illustration.

Corresponding author: Nikolaos M. Freris, Email: nfr@ustc.edu.cn.

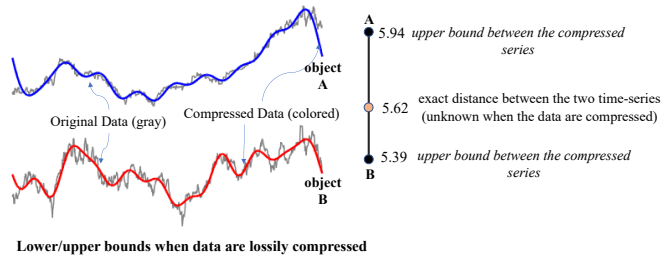


Fig. 1. The embedding proposed does not require exact distances but only ranges of lower/upper distance estimates. One case where such estimates are available is when compressing data using a lossy compression mechanism.

Our data embedding method, called *Multi-objective 2D Embedding* (MoDE), can capture with high fidelity multiple facets of the data relationships: correlations, distances, and orders/importance rankings. MoDE may serve as an effective big-data visualization tool in applications where the amount of collected data is so large that it must be compressed in a lossy manner, in which case only bounds on distances can be deduced such as in [9]–[11]. MoDE organizes the dataset as a graph, across the edges of which (i.e., between pairs of data objects) lower/upper bounds on pairwise distances are obtained. The method leverages these bounds to cast the 2D embedding problem as a set of *linear inequalities* exploiting also the (partial) ordering of objects. It solves the inequality system in a least-squares sense to obtain the angular values of the embedded data points.

Our **contributions** enlist: (1) an iterative, anytime [12] embedding algorithm that operates on distance ranges; the anytime nature provides early embeddings even before complete execution of the algorithm; (2) preservation of distance/correlation relations with competitive accuracy to the state-of-the-art; (3) preservation of (partial) orders present in the dataset, thus promoting the *interpretability* of the embedding outcome; and (4) theoretical analysis establishing linear convergence (i.e., exponential decay of the approximation error with the number of iterations) despite lack of strong convexity in the objective.

To the best of our knowledge, this is the first embedding approach that can accommodate **uncertain distance information** in the form of distance ranges.

## A. Notation

For  $n \in \mathbb{Z}_+$ , we denote  $[n] := \{1, \dots, n\}$ . We use boldface lower-case variables for vectors (represented as column vectors) and denote matrices with upper-case variables. The standard inner product in  $\mathbb{C}^d$  is denoted by  $\langle \cdot, \cdot \rangle$ , whereas  $\|\cdot\|$  denotes the Euclidean norm. For a matrix  $A$ , we use  $\mathbf{a}_i$ ,  $A^T$ ,  $\text{Range}(A)$ , and  $\|A\| = \sigma_{\max}(A)$  for the  $i$ -th column, Moore–Penrose pseudoinverse, range (column space), and maximum singular value, respectively. For  $\ell < u$ , we define  $(x)_\ell^u := \min(\max(x, \ell), u)$ , the projection of  $x$  to the interval  $[\ell, u]$ . We extend the notation for vectors, with  $\mathbf{x}[\ell, u]$ , and  $(\mathbf{x})_\ell^u := ((x_i)_{\ell_i}^{u_i})$  meant entry-wise.

## II. PROBLEM FORMULATION

Given a dataset  $X \in \mathbb{C}^{d \times n}$ , where  $n$  is the number of data points and  $d$  their dimension, a primary goal of the proposed 2D embedding is to generate  $X_{2d} \in \mathbb{R}^{2 \times n}$  aiming to preserve (as accurately as possible) the distance structure, that is:

**Objective 1:**  $\|\mathbf{x}_i - \mathbf{x}_j\| = \|\mathbf{x}_{2d,i} - \mathbf{x}_{2d,j}\|$ .

In our setting, *direct access to  $X$  is assumed unavailable*, and the only information accessible encompasses *ranges of distances* (i.e., lower and upper bounds) between a subset of object pairs; for example, this is the case when data are lossily compressed using orthonormal transforms. Nonetheless, norms of original points are assumed known—one scalar value per object is stored—and MoDE is *norm-preserving*, in that it satisfies  $\|\mathbf{x}_i\| = \|\mathbf{x}_{2d,i}\|$  for all  $i \in [n]$ . Then, we express the embedding in polar coordinates using  $\theta_i$  to denote the angle of  $\mathbf{x}_{2d,i}$ , which are the decision variables in our method (one scalar value per object).

In the following, we leverage pairwise *correlations* to obtain linear constraints on angular values  $\theta \in \mathbb{R}^n$ . For simplicity, we assume for now that exact pairwise distances are known, and showcase how to utilize them for formulating the embedding problem as a linear system. Next, we illustrate how to incorporate bounds in obtaining a set of linear inequalities (i.e., a linear system with ranges of distance values).

a) *Relative angle from correlation.*: From the basic relation

$$\|\mathbf{x}_i - \mathbf{x}_j\|^2 = \|\mathbf{x}_i\|^2 + \|\mathbf{x}_j\|^2 - 2c_{\mathbf{x}_i, \mathbf{x}_j} \|\mathbf{x}_i\| \|\mathbf{x}_j\|, \quad (1)$$

where  $c_{\mathbf{x}_i, \mathbf{x}_j} := \frac{\text{Re}\langle \mathbf{x}_i, \mathbf{x}_j \rangle}{\|\mathbf{x}_i\| \|\mathbf{x}_j\|} \in [-1, 1]$  is the *correlation* (coefficient), and the fact that MoDE is norm-preserving it becomes apparent that *distance preservation is equivalent to preservation of pairwise correlations*, i.e.,

**Objective 2:**  $c_{\mathbf{x}_i, \mathbf{x}_j} = c_{\mathbf{x}_{2d,i}, \mathbf{x}_{2d,j}}$ .

Note that for a single pair  $(\mathbf{x}_i, \mathbf{x}_j)$ , objectives 1 & 2 can be achieved *perfectly* when  $c_{\mathbf{x}_i, \mathbf{x}_j}$  is known, by setting:

$$\theta_j - \theta_i = \arccos(c_{\mathbf{x}_i, \mathbf{x}_j}), \quad (2)$$

where the sign indeterminacy is due the fact that  $\cos(\cdot)$  is an even function. A distinctive attribute of MoDE, not present in other embedding techniques, is that it further seeks to *preserve an order* on data points.

**Definition 1.** (Partial order) A strict *partial order* is a binary relation that is non-symmetric ( $x \prec y$  implies  $y \not\prec x$ ) and transitive ( $x \prec y$  and  $y \prec z$  imply  $x \prec z$ ). This generalizes the notion of (total) ordering because for  $x, y$  it may hold that neither  $x \prec y$  nor  $y \prec x$ , i.e., two points may not be *comparable*. A non-strict partial order differs in that it does not exclude both  $x \prec y$  and  $y \prec x$  for  $x \neq y$ . A set equipped with a partial order is called a partially ordered set (*poset*).

Given a (strict) partial order on  $[n]$ , the distinct objective of MoDE is to plot points such that:

**Objective 3:**

$$i \prec j \Rightarrow \theta_i < \theta_j. \quad (3)$$

MoDE embodies this objective by means of adopting  $\theta_j$   $\theta_i = \arccos(c_{\mathbf{x}_i, \mathbf{x}_j}) \in [0, \pi]$ , when  $i \prec j$  [13]. Our method organizes the dataset as a (directed) data graph  $G = (V, E)$ , where  $V$  is the set of vertices (points) and  $E$  is the set of edges. The ordered pair  $(i, j) \in E$  if and only if  $i \prec j$ . We let  $A \in \mathbb{R}^{m \times n}$  denote the incidence matrix ( $m = |E|, n = |V|$ ), where each row corresponds to a directed edge  $(i, j) \in E$  and takes values  $-1, 1$  at the  $i$ -th and  $j$ -th entry, respectively, and is zero elsewhere. Consequently, it follows that (2), (3) can be written compactly as a linear system  $\mathbf{y} = A\theta$ , where  $\mathbf{y} \in \mathbb{R}^m$  stacks the values  $\arccos(c_{\mathbf{x}_i, \mathbf{x}_j}) \mathbf{1}_{(i,j) \in E}$  computed across the edges of the data graph  $G$ . The following remark discusses means of constructing the data graph, when not defined *a priori* alongside a strict partial order.

*Remark 1* (Ordering the dataset). A typical embedding will try to preserve only the pairwise distances in the lower dimensionality. However, in many applications, the notion of *importance ranking* of a data point is explicitly given by means of a *score function*. For example, in a recommender system application for movies, in addition to the movie similarities, it is advantageous to also highlight the importance of the movie (e.g., the average user rating can be used as a proxy for that). Similarly, when visualizing similarities in financial time-series data, one can use the market capitalization of a company, to highlight its importance. *The question that we address here, is how to incorporate in our embedding this importance of a data-point.*

The score mechanism incurs a total ordering in  $V$ , i.e., a non-strict partial order on  $V \times V$ . MoDE aims to plot more “important” points at higher angles (cf. (3)), thus yielding an *interpretable* embedding outcome. In practice, it may be beneficial to consider only a subset of relations, e.g., based on the  $K$  Nearest Neighbors (K-NN) of each point, for the sake of computational savings (this is the option we adopt in our experiments), which incurs a partial order. MoDE does not *require* a score per object, but if available it is used to create a more interpretable embedding. In the absence of scores, *random scores are used*. Our experiments reveal that preservation of the objectives of MoDE are not compromised in the absence of object score values.

**Operating on distance ranges.** When the original dataset is unavailable, exact distance/correlation information between a

pair of points in the original space is lost. We restrict our attention to cases for which it is possible to infer *lower/upper bounds* on pairwise distances/correlations between original points. Besides, note that one may further use these in constructing a K-NN graph (K-NNG), for example with respect to the average of lower/upper distance bounds. This was the case for our experiments, where we assumed that only the compressed data are available.

Obtaining such lower and upper bounds can be regarded as a pre-processing step, during which for every pair of points  $(i, j) \in E$ , one deduces a correlation uncertainty of the form  $c_{ij} \in [c_{ij}^-, c_{ij}^+]$ , where  $c_{ij} = c_{\mathbf{x}_i, \mathbf{x}_j}$  is the correlation of original points  $\mathbf{x}_i, \mathbf{x}_j$  (this can be inferred from distance bounds immediately from (1), given knowledge of norms). These bounds directly yield lower/upper bounds of  $u_{ij}$  and  $\ell_{ij}$  on the angular difference  $\theta_j - \theta_i$ , namely  $u_{ij} := \arccos(c_{ij}^-)$  and  $\ell_{ij} := \arccos(c_{ij}^+)$ , in light of the fact that  $\arccos(\cdot) : [ -1, 1 ] \rightarrow [ 0, \pi ]$  is a decreasing function. To conclude, we stack the lower/upper bounds in vectors  $\mathbf{u}$  with  $\mathbf{u} \in [ 0, \pi ]^m$  to capture the uncertainty in correlation via a linear system of inequalities on the angular domain:

$$\mathbf{A} \mathbf{u} \leq \mathbf{b}. \quad (4)$$

Given that the embedding is from a high-dimensional space  $\mathbb{R}^d$  to the 2D plane, it is plausible that the inequality system is infeasible, i.e.,  $\mathbf{f}^T \mathbf{u} \leq \mathbf{g}$ . For this reason, we consider a solution in the least-squares sense, that is:

$$\text{minimize } f(\mathbf{u}) := \frac{1}{2} \|\mathbf{A} \mathbf{u} - \mathbf{b}\|^2. \quad (5)$$

This is equivalent to computing  $\text{dist}(\text{Range}(\mathbf{A}); [\mathbf{b}])$ , i.e., the closest point to  $[\mathbf{b}]$  that belongs in the range space (i.e., the linear span of columns) of  $\mathbf{A}$ . In the next section, we devise an efficient algorithm to solve this problem.

### III. ALGORITHM

We invoke the gradient method for solving (5); The iterates ( $k \in \mathbb{Z}_+$ ) are:

$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} - \gamma \mathbf{A}^T (\mathbf{A} \mathbf{u}^{(k)} - \mathbf{b}). \quad (6)$$

---

#### Algorithm 1 Multi-objective 2D Embedding (MoDE)

---

**Input:**  $\mathcal{X}, \mathbf{s}, K$   $\triangleright$   $\mathcal{X}$ : ‘inexact’ dataset;  $\mathbf{s}$ : scores;  $K$ : # NNs  
1: Construct  $G$  (incidence matrix  $A$ )  
2: Obtain  $\mathbf{u}$  in (4)  $\triangleright$  Lower/Upper correlation bounds  
3:  $\text{DAE}(A, \mathbf{u}, \epsilon)$   $\triangleright$  Compute angular values  
4: **for**  $i = 1, \dots, n$  **do**  
5:  $\mathbf{x}_{2d,i} = (k \mathbf{x}_i \cos \theta_i, k \mathbf{x}_i \sin \theta_i)$   $\triangleright$  Embedding  
6: **end for**

**Output:**  $X_{2d}$

---

The algorithmic description of MoDE is illustrated in Alg. 1. In Steps 1 & 2, respectively, the algorithm constructs a data graph  $G$  and computes the relevant lower/upper correlation

---

#### Algorithm 2 Distributed Angle Estimator (DAE)

---

**Input:**  $A, \mathbf{u}, \epsilon$   $\triangleright$   $A$ : incidence matrix;  $\mathbf{u}$ : lower/upper bounds;  $\epsilon$ : tolerance

1:  $\mathbf{0}; k = 0$   $\triangleright$  Initialization  
2:  $\mathbf{d} = \text{diag}(A^T A)$   $\triangleright$  Degrees  
3:  $d_{\max} = \max_{i \in [n]} d_i$   $\triangleright$  Maximum degree  
4:  $\gamma = \frac{1}{2d_{\max}}$   $\triangleright$  Step size  
5: **repeat**  
6:  $k = k + 1$   
7: **for**  $i = 1, \dots, n$  **do**  $\triangleright$  Updates  
8:

$$\phi_i^{(k+1)} = \sum_{j \in N_i^{\text{in}}} \left[ \theta_j^{(k)} + \left( \theta_i^{(k)} - \theta_j^{(k)} \right) \frac{u_{ij}}{\ell_{ij}} \right] + \sum_{j \in N_i^{\text{out}}} \left[ \theta_j^{(k)} - \left( \theta_j^{(k)} - \theta_i^{(k)} \right) \frac{u_{ij}}{\ell_{ij}} \right]$$

9:  $\theta_i^{(k+1)} = (1 - \gamma d_i) \theta_i^{(k)} + \gamma \phi_i^{(k+1)}$   
10: **end for**  
11: **until**  $\left\| \mathbf{u}^{(k+1)} - \mathbf{u}^{(k)} \right\| \leq \epsilon$   $\triangleright$  Termination criterion  
**Output:**  $\mathbf{u}^{(k+1)}$

---

bounds (in case the data graph is available step 1 is omitted). In accordance with our experiments, we focus on the special case when the dataset is not organized a priori as a strict poset; in such case, one may compute lower/upper distance bounds for each pair of points and use the average of these bounds to construct a K-NNG.

The direction of each edge is then determined based on a partial order, e.g., score values, that also constitute an input to the algorithm. If no score exists for the dataset objects, a random score is used. Step 3 uses gradient descent (Alg. 2) to compute the angular values, while Steps 4–6 produce the (norm-preserving) embedding outcome.

The gradient method translates to Alg. 2, referred to as *Distributed Angle Estimator* (DAE). The update equations for each point’s angular value are *distributed* in the sense that a point uses solely angular values pertaining to its neighbors to update its own, cf. Steps 8–9 (we define  $N_i^{\text{in}} := \{j \in [n] \mid (i, j) \in E\}$ ,  $N_i^{\text{out}} := \{j \in [n] \mid (j, i) \in E\}$  and  $d_i = |N_i^{\text{in}}| + |N_i^{\text{out}}|$  as the in-/out neighborhood, respectively, and (total) degree of point  $i$ ). We deem this a favorable attribute of MoDE, in that updating a single point’s angular estimate does not require processing the entire dataset, but rather a subset of neighboring points. The algorithm terminates when the norm of the gradient of the objective in (5) falls below a given tolerance (Step 11).

Because of its iterative implementation, MoDE is an **anytime algorithm** that provides an embedding of all the points (which progressively improves), even before the full execution of the algorithm when the termination criterion is reached. The following theorem establishes linear convergence for Alg. 2 (the proof is omitted for length considerations.).

**Theorem 1** (Convergence of Alg. 2). *For tolerance  $\epsilon > 0$ ,*

Alg. 2 takes  $O(d_{\max}^2 \eta^2 \log(\frac{1}{\epsilon}))$  iterations and outputs an  $\epsilon^\theta$  optimal solution where  $\eta > 0$  is the metric subregularity constant of  $r f$  [14],  $d_{\max}$  is the maximum degree of the graph, and  $\epsilon^\theta := 2d_{\max} \eta \epsilon$ .

#### IV. EXPERIMENTS

We compare the quality of our embedding methodology with several widely used embedding techniques, namely ISOMAP, MDS, and t-SNE. The comparisons are in terms of:

- 1) Quality of embedding
- 2) Classification accuracy

Moreover, we highlight solely for MoDE its ability to provide an accurate embedding even in the absence of object orders/scores in the dataset, and its linear scalability. The code and datasets used are available at: <https://github.com/ahmadajal/MoDE>. All experiments have been conducted on a 2.5 GHz 14-Core Intel Xeon W with 256 GB of RAM.

**Experimental setup.** Our method assumes as input lower and upper bounds on the Euclidean distance for a given pair of objects. Any methodology that provides such bounds can be used. Here, we derive such bounds by lossily compressing time series using the approach of [7], which has been proven to compute optimally tight bounds. Therefore, for each pair of objects, we do not compute the exact distance (as this information is lost in compression) but rather a lower bound  $\ell$  and an upper bound  $u$ . Our methodology uses both of these bounds, cf. (5). The techniques with which we compare our methodology assume an exact distance, and for those we use the mid-point  $\frac{1}{2}(\ell + u)$  as a surrogate. For techniques that work on the K-NN graph (all except MDS),  $K = 20$  was chosen. MDS, which is equivalent to Principal Component Analysis (PCA), is a global technique and uses all pairwise distances.

##### A. Embedding Quality

**Comparison Metrics.** We evaluate the embedding quality on 2D of various techniques across a variety of metrics that highlight the quality in distance, correlation, and score order preservation. We define and evaluate metrics on the K-NNG with respect to *original* distances on uncompressed data so as to simultaneously assess the impact of both compression and embedding on a lower-dimensional space (2D) on the retention of relations. We define the generic formula  $R := 1 - \frac{1}{m} \sum_{(i,j) \in E} C_{ij}$ , where  $C_{ij}$  is the cost of preservation accuracy on the pair  $(i, j) \in E$ . In all cases, a higher metric value implies a more accurate preservation, with 1 corresponding to perfect preservation. Specifically, we introduce: (1)  $R_d \in [0, 1]$  for *distance*, by setting  $C_{ij} = \frac{|d_{ij} - \hat{d}_{ij}|}{d_{ij} + \hat{d}_{ij}}$ , where  $d_{ij} := \|\mathbf{x}_i - \mathbf{x}_j\|$  denotes the original distance on the high-dimensional data, and  $\hat{d}_{ij} := \|\mathbf{x}_{2d,i} - \mathbf{x}_{2d,j}\|$  denotes the distance between corresponding embeddings in 2D; (2)  $R_c \in [0, 1]$  for *correlation*, by setting  $C_{ij} = |c_{ij} - \hat{c}_{ij}|$ , where  $c_{ij} := c_{\mathbf{x}_i, \mathbf{x}_j}$ ,  $\hat{c}_{ij} := c_{\mathbf{x}_{2d,i}, \mathbf{x}_{2d,j}}$  refer to the correlation between original and embedded points, respectively; (3)  $R_o \in [0, 1]$  for *order* by setting  $C_{ij} = 1$  when  $i > j$  and  $\theta_i > \theta_j$ , i.e., when order is not preserved, and 0 otherwise; in other

words, we compute the fraction of preserved order relations. Finally, for tests involving scored datasets (i.e., a total ordering) we further assess the *Spearman* correlation metric [15, p. 508]  $R_s \in [-1, 1]$ , which assesses how well the total order is preserved in the K-nearest neighborhoods of data points.

Understandably, metrics involving order preservation are mainly used to capture the additional objective of our methodology and are meant to highlight that our approach can indeed satisfy its multi-objective desiderata. To this end, the corresponding values for the baseline methods are only presented for the sake of completeness.

**Quality of embedding.** For the first set of experiments, we have used two *time-series* datasets that we compiled ourselves: Small-Stock (436 stocks of length 128) and Big-Stock (2252 stocks of length 1024). The nature of these datasets allows us to try small and large compression ratios using Fourier coefficients (by dropping low-energy coefficients). The datasets were created using historical prices from equities in the NASDAQ stock index, from which data were extracted using the Investors Exchange (IEX) API (<https://iextrading.com/developer/docs>), and were compressed in the Fourier basis by storing the highest magnitude Fourier coefficients. As score, to order the data objects, we use the market capitalization of each stock. Therefore, in our method, stocks with greater capitalization are aimed to be placed at a “greater” angle on a 2D plot, which provides an additional attribute of *interpretability* not present in any of the other baseline methods. We compress the Small-Stock dataset using a 16:1 compression ratio, and the Big-Stock dataset using a 128:1 compression ratio.

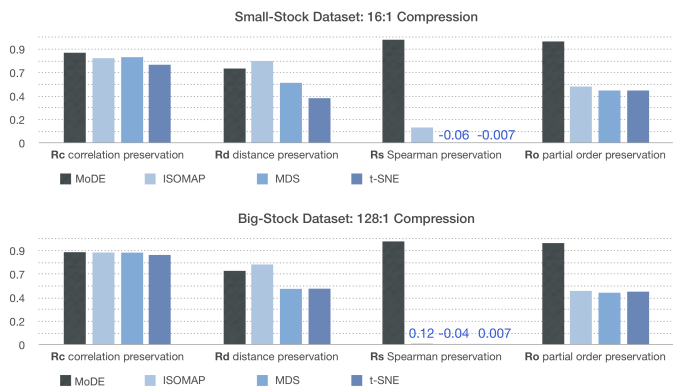


Fig. 2. Comparison of MoDE with other embedding techniques.

We assess the performance of the embedding methods using the aforementioned quality metrics. Fig. 2 depicts the values for each of those metrics for the four techniques we compared. For t-SNE, because of its randomized implementation, we report results averaged over 10 runs. We observe that MoDE (darkest bar) depicts the highest preservation across all metrics and across all techniques, with the singular exception of ISOMAP which features more accurate distance preservation ( $R_d$  metric), but lags behind MoDE for all other metrics. It should be noted that ISOMAP incurs a higher computational cost than all other

techniques, because after the K-NN graph computation it builds a minimum spanning tree based on distances. The metrics  $R_o$  and  $R_s$  are depicted for other methods only for the sake of completeness; they are not meant to claim any superiority of MoDE, as it is the only method with an objective of maintaining partial orders. Indeed, note that for baseline methods  $R_o$  and  $R_s$  take values very close to 0.5 and 0, respectively, which reveals no order preservation, in full agreement with the fact that all techniques except MoDE do not consider score information in their embedding process. This fact is further illustrated on the comparative visualization plot for the `Big-Stock` dataset in Fig. 3. The two-dimensional plot reveals that MoDE succeeds in preserving score orders very accurately, as captured by the smooth transition of color-coded points. In this example, the scores/colors encode the market capitalization for each stock.

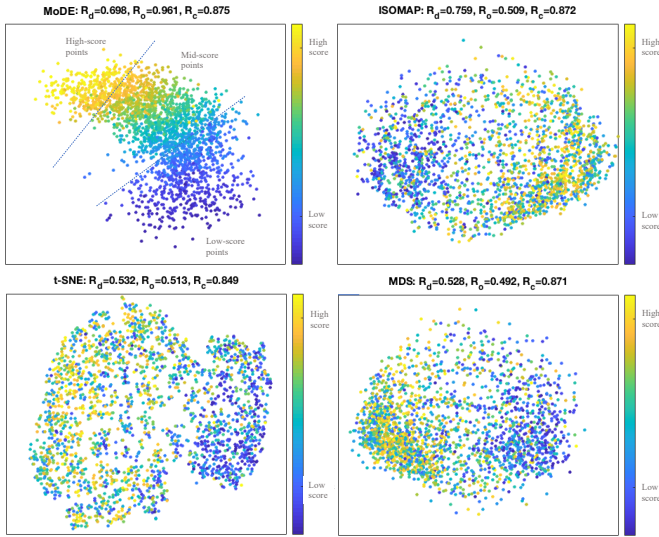


Fig. 3. [`Big-Stock` dataset] Comparative visualization of our method (MoDE) with ISOMAP, t-SNE, and MDS. Color coding demonstrates objects' score values.

### B. Classification Accuracy

We evaluate the embedding quality of MoDE in the context of classification tasks. We assess the prediction accuracy when using MoDE as a pre-processing step to reduce the dimensionality of the data. We compare with t-SNE, ISOMAP, and parametric t-SNE (an extension to t-SNE that learns a parametric mapping between the high-dimensional data space and the latent space [16]). In this experiment, all techniques use the exact distances between the object and not approximate distance information. We note that this is possible to achieve in MoDE by simply providing the same lower and upper bound with value equal to the distance between any two objects.

To conduct this experiment, we split the dataset into training and test (80-20 split). We map the *training* points onto 2D using the previously mentioned techniques and build a classifier on 2D using the dimensionality-reduced training dataset. Subsequently, we map the test points on 2D and use the classifier built to predict their class. We evaluate on two classification methods:

Multinomial Logistic Regression (LR), that generalizes logistic regression to multi-class problems. We used the implementation of scikit-learn [17]. The regularization hyper-parameter was tuned for each dataset and for each of the embedding methods separately.

K-Nearest Neighbors classifier (KNN), which classifies an object by majority vote across its nearest neighbors, i.e., the object is assigned to the class most common among its K-nearest neighbors. We used the implementation of scikit-learn for this task [17]. In the voting phase, we weight the nearest neighbor points by their inverse distance to the query point in order to give more importance to closer points. The number of neighbors for this method is set equal to the number of nearest neighbors used for the embedding methods (MoDE, t-SNE, parametric t-SNE, and ISOMAP).

For each of the datasets and embedding methods, we report the classification accuracy, i.e., the number of correctly classified data points divided by the total number of data points. These datasets do not come with inherent scores, so for MoDE we use the actual class labels as score values for each of the data points. This does not make unfair the comparison to the other methods, because the scores are only used to embed the training set. The test set uses *only* the original object features.

For MoDE and t-SNE, we embed the test data points in the 2D space by considering the nearest neighbors in the original space and using their location in the new space to map the new points. We take the weighted average of their nearest neighbor embeddings with weights set proportional to the inverse of the distance to these neighbors, that is, given a point  $i$  in the test data, we compute its embedding  $\mathbf{x}_{2d,i}$  as follows:

$$\mathbf{x}_{2d,i} = \frac{1}{\sum_{j \in N_i} \frac{1}{d_{ij}}} \sum_{j \in N_i} \frac{1}{d_{ij}} \mathbf{x}_{2d,j}, \quad (7)$$

where  $N_i$  is the set of K-nearest neighbors of  $i$  in the training set and  $d_{ij}$  is the distance between points  $i$  and  $j$ . For ISOMAP, given that it depends on geodesic neighbors, we first find the nearest neighbors with respect to geodesic distance of each test data point in the training set. Then, we construct a kernel by computing the shortest geodesic distances from each test point to another in the training set. To conclude, the embedding of the test set is taken using this kernel on the embedded vectors of the training set [17].

Table I shows the test accuracy for datasets from the UCR time series archive and the UCI machine learning repository [18]. These results highlight that training machine learning models on the embeddings provided by MoDE typically yields very favorable results than when using t-SNE or ISOMAP for dimensionality reduction.

### C. Additional experiments for MoDE

Here, we present experiments to highlight several desirable traits of MoDE .

**Scalability.** We test the scalability of MoDE under inexact distance information by considering increasing data sizes

TABLE I  
CLASSIFICATION ACCURACY USING LOGISTIC REGRESSION (LR) AND K-NEAREST NEIGHBORS (KNN).

Dataset (# classes)	n	dim	original data		MoDE		t-SNE		parametric t-SNE		ISOMAP	
			LR	KNN	LR	KNN	LR	KNN	LR	KNN	LR	KNN
Arrow (5)	500	1024	0.93	0.87	0.48	0.51	<b>0.72</b>	<b>0.79</b>	0.22	0.37	0.69	0.71
Wafer (2)	1000	128	0.955	0.985	<b>0.915</b>	0.93	0.905	0.985	0.905	0.945	0.905	<b>0.99</b>
Breast Cancer (2)	569	30	0.982	0.982	0.719	0.807	<b>0.964</b>	<b>0.964</b>	<b>0.964</b>	0.956	0.947	0.956
Heart Beat (2)	14545	188	1.0	0.996	0.92	0.932	0.81	0.958	<b>1.0</b>	<b>1.0</b>	0.867	0.898
Madelon (2)	2600	500	0.561	0.586	<b>0.586</b>	0.526	0.463	0.501	0.592	<b>0.613</b>	0.517	0.532
EEG eye state (2)	11853	14	0.682	0.925	<b>0.822</b>	<b>0.839</b>	0.536	0.742	0.545	0.587	0.569	0.581
Wine quality (3)	3961	11	0.606	0.602	<b>0.523</b>	<b>0.537</b>	0.484	0.499	0.442	0.496	0.479	0.503
Phishing websites (2)	5425	30	0.938	0.941	<b>0.748</b>	0.757	0.712	<b>0.884</b>	0.727	0.84	0.536	0.682
cifar-10 (10)	10000	3072	0.392	0.278	0.106	0.111	0.208	0.189	<b>0.224</b>	<b>0.213</b>	0.221	0.205

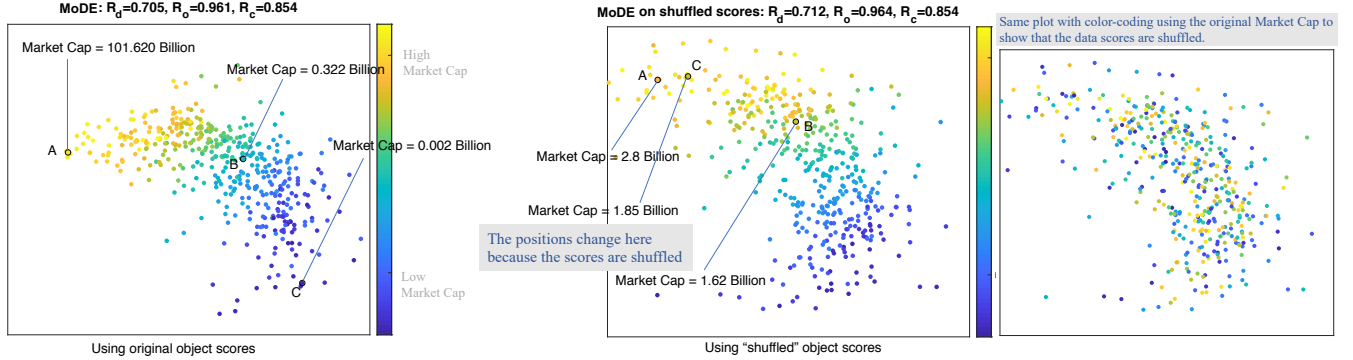


Fig. 4. [Small-Stock dataset] MoDE operates effectively even without object scores. In their absence, random scores can be used without affecting the embedding quality.

(number of data objects). For this experiment, we use the EEG eye state dataset, with dataset sizes ranging between 1,000 and 11,000 objects. Figure 5 shows the number of iterations for MoDE needed to reach the termination criterion (step 11 in Alg. 2). We have set  $\epsilon = \frac{1}{\sqrt{n}} \cdot 10^{-4}$ , where  $n$  is the number of data points (this is done to ascertain a fair comparison of run-time by maintaining a common RMSE (Rooted-Mean-Square-Error) value across  $n$ ). The linear relation between the number of objects and the iteration counts further supports the scalable nature of MoDE for large datasets. 5.

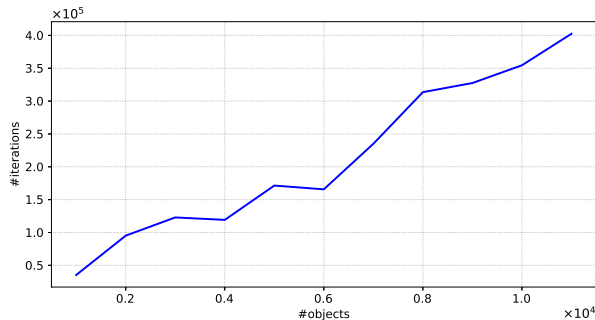


Fig. 5. Scalability of MoDE: Iterations of MoDE until convergence on increasingly larger dataset sizes (EEG eye state dataset).

**Embedding without object scores.** One might be prompt to consider a limitation of MoDE the fact that it asserts the

existence of a score function, but this *is not the case*. In the absence of scores that dictate a partial order of points, *random* orders can be used without affecting the preservation of the given objectives. This is because the points will still be mapped so that distances and correlations are (approximately) preserved, by using random orders. Fig. 4 depicts this for the Small-Stock dataset, where we use both the original and shuffled market capitalization of the stocks, without observing any deterioration in the metrics qualified.

## V. CONCLUSION

We presented the first embedding method that operates on inexact distance information, a scenario encountered when dealing with compressed data. Hence, our method can be instrumental for visualization of very large (compressed) datasets. Moreover, interpretability is enhanced by presenting to the user an *ordered* and also *consistent* outcome of visualization, typically in the shape of a ‘crescent moon’.

## ACKNOWLEDGMENT

Work supported by the Ministry of Science and Technology of China under grant 2019YFB2102200, the Anhui Dept. of Science and Technology under grant 201903a05020049, and Tencent Holdings Ltd under grant FR202003.

## REFERENCES

- [1] T. F. Cox and M. A. Cox, *Multidimensional Scaling*. Chapman and Hall/CRC, 2000.
- [2] J. B. Tenenbaum, V. De Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.
- [3] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.
- [4] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.
- [5] E. Bingham and H. Mannila, "Random projection in dimensionality reduction: applications to image and text data," in *Proceedings of the 7th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2001, pp. 245–250.
- [6] L. McInnes and J. Healy, "UMAP: Uniform Manifold Approximation and Projection for dimension reduction," *arXiv preprint:1802.03426*, 2018.
- [7] M. Vlachos, N. Freris, and A. Kyrillidis, "Compressive mining: fast and optimal data mining in the compressed domain," *The International Journal on Very Large Data Bases (VLDB Journal)*, vol. 24, no. 1, pp. 1–24, 2015.
- [8] E. Candes, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, vol. 59, no. 8, pp. 1207–1223, 2006.
- [9] J. Paratte, N. Perraudin, and P. Vandergheynst, "Compressive embedding and visualization using graphs," *arXiv preprint:1702.05815*, 2017.
- [10] J. Tang, J. Liu, M. Zhang, and Q. Mei, "Visualizing large-scale and high-dimensional data," in *Proceedings of the 25th International Conference on World Wide Web (WWW)*, 2016, pp. 287–297.
- [11] J. Camacho, "Visualizing big data with compressed score plots: Approach and research challenges," *Chemometrics and Intelligent Laboratory Systems*, vol. 135, pp. 110 – 125, 2014.
- [12] S. Zilberstein, "Using anytime algorithms in intelligent systems," *AI magazine*, vol. 17, no. 3, pp. 73–73, 1996.
- [13] N. Freris, C. Yang, and M. Vlachos, "FIDE: Fast and Interpretable 2D Embedding with correlation, distance, and rank considerations," in *Proceedings of the 6th International Conference on Big Data Computing and Communications (BIGCOM)*, 2020, pp. 57–65.
- [14] X. Zheng and K. Ng, "Metric subregularity of piecewise linear multifunctions and applications to piecewise linear multiobjective optimization," *SIAM Journal on Optimization*, vol. 24, pp. 154–174, 2014.
- [15] A. D. Well and J. L. Myers, *Research design & statistical analysis*. Psychology Press, 2003.
- [16] L. van der Maaten, "Learning a parametric embedding by preserving local structure," in *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, vol. 5, 2009, pp. 384–391.
- [17] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [18] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: <http://archive.ics.uci.edu/ml>