ADVANCED INTELLIGENT SYSTEMS
Open Access
www.advintellsyst.com

Dimension Reduction of Multidimensional Structured and Unstructured Datasets through Ensemble Learning of Neural Embeddings

Juan Carlos Alvarado-Pérez, Miguel Angel Garcia, and Domenec Puig*

Dimension reduction aims to project a high-dimensional dataset into a low-dimensional space. It tries to preserve the topological relationships among the original data points and/or induce clusters. NetDRm, an online dimensionality reduction method based on neural ensemble learning that integrates different dimension reduction methods in a synergistic way, is introduced. NetDRm is designed for datasets of multidimensional points that can be either structured (e.g., images) or unstructured (e.g., point clouds, tabular data). It starts by training a collection of deep residual encoders that learn the embeddings induced by multiple dimension reduction methods applied to the input dataset. Subsequently, a dense neural network integrates the generated encoders by emphasizing topological preservation or cluster induction. Experiments conducted on widely used multidimensional datasets (point-cloud manifolds, image datasets, tabular record datasets) show that the proposed method yields better results in terms of topological preservation (R_{NX} curves), cluster induction (V measure), and classification accuracy than the most relevant dimension reduction methods.

1. Introduction

The significant increase in the data volumes generated by the integration of multiple technologies and information sources entails dealing with complex, high-dimensional data, where dimension is understood as the number of variables or attributes that characterize each instance of an object or phenomenon. Processing multiple variables is a challenge to be faced by the data analysis and pattern recognition communities. Additionally, the task of presenting and/or representing data in an understandable, intuitive, and dynamic way is not trivial.[1]

Dimensionality (or dimension) reduction (DR) aims at developing efficient ways of representing and interrelating data, such that the information can be more usable and intelligible to the user.[2,3] Specifically, this is done by mapping (embedding) the original high-dimensional data into a low-dimensional space while preserving the original information as faithfully as possible. Frequently, it is necessary to represent the data in an efficient way closer to how humans understand and process information.[4] DR tackles this problem by transforming the data with the least information loss, allowing the user or processing algorithm to explore and extract useful information present in the original representation.[5,6] The goal is to be able to summarize and describe the original contents, exposing their main features. In particular, the representation of N-dimensional spaces in the 2D or 3D space while either inducing clusters or preserving the topological structure of the given data is consistent with the human perceptual system, which is limited to 3D. In addition, DR also reduces computational costs (processing time and storage capacity), since it is more efficient to process and understand a smaller number of variables. Therefore, DR becomes a fundamental stage in the design of robust data analysis systems.

An enormous variety of DR methods have been proposed based on different principles:[1,7] statistical concepts, spectral approaches, graphs, predefined structures (e.g., lattices), and neural networks. Depending on their underlying principle, they generate mappings (embeddings) that preserve either global or local characteristics, exhibit intrusive or extrusive behavior, or which are either more efficient in preserving the high-dimensional structure of data or more discriminative, hence allowing for better cluster induction.

In machine learning, a combination approach known as ensemble learning[8,9] applies the principle of "two heads think better than one." This approach is utilized in several techniques, such as random forests, which allow the integration of several

J. C. Alvarado-Pérez
Departament d'Enginyeria Informàtica i Matemàtiques, Universitat Rovira i Virgili and Department of Engineering
CESMAG University
Carrera 20A No. 14–54 Centro, Pasto 52001, Colombia
E-mail: jcalvaradop@sgc.gov.co

M. A. Garcia
Department of Electronic and Communications Technology
Autonomous University of Madrid
Francisco Tomas y Valiente 11, 28049 Madrid, Spain

D. Puig
Departament d'Enginyeria Informàtica i Matemàtiques
Universitat Rovira i Virgili
Paisos Catalans, 26E-43007 Tarragona, Spain

The ORCID identification number(s) for the author(s) of this article can be found under https://doi.org/10.1002/aisy.202400178.

DOI: 10.1002/aisy.202400178

classification trees to improve the prediction of labels of a target variable. Neural networks are another example. In this case, several linear hyperplanes, each generated by one neuron, are synergistically combined in a network to generate nonlinear solutions. One of the approaches to ensemble learning known as *stacking* obtains an integrated method (*metamodel*) that improves the performance of several heterogeneous algorithms that have been previously trained independently. This approach can be used to combine dimension reduction algorithms, synergistically integrating the features inherited from the different heuristics of the combined DR methods.

Most DR methods are classified as batch processing (offline). They require the entire dataset to compute the low-dimensional space. In addition, if new data instances must be projected, the whole process must be redone from scratch. This can be a serious drawback, as the full dataset and the results of the different operations must fit into main computer memory, which is insufficient in many cases. Alternatively, other DR methods, such as those based on neural networks, apply online processing (also known as incremental or adaptive processing). This form of processing data does not require the complete set, but instances or subsets of data called minibatches, thus making computer memory management much more efficient.

In the present work, an intuitive method is presented for DR with either topological preservation or cluster induction. It is based on online processing of minibatches and combination of DR methods using the ensemble learning approach. The proposed method applies multiple precursor DR methods to a set of high-dimensional data for generating respective embeddings (mappings). These low-dimensional representations are learnt in a self-supervised way through neural encoders.[10] These encoders allow the online adaptation of the traditional, offline methods from which they were trained, thus being able to project new data points very efficiently. Once the encoders are obtained, they are optimized using manifold approximation based on Uniform Manifold Approximation and Projection (UMAP) net.[11,12]

Subsequently, the optimized encoders are integrated under the ensemble learning approach using a dense residual neural network that processes data in blocks to generate a new embedding. We propose two variations of the integration network, referred to as NetDRm_T and NetDRm_D, depending on whether the final aim is to achieve topological preservation[13–15] or cluster induction (i.e., discrimination).[16–18] For topological preservation, the integration network uses unsupervised learning based on a computational cost function that attempts to preserve the local topology by means of a probabilistic neighborhood graph and the global topology through high- and low-dimensional Euclidean distance matrices. In turn, the discriminative version uses a deep network trained in a supervised fashion with a composite cost function that attempts to minimize intracluster differences and maximize intercluster differences.

The experimental validation was conducted on a variety of multidimensional structured (image datasets) and unstructured datasets (manifold-type point datasets and tabular datasets). Due to space limitations, this article presents results corresponding to four representative datasets: Swiss Roll and Sphere (point datasets), ECG Arrhythmia (tabular dataset), Fashion-MNIST, and COIL-20 (image datasets). Three widely used measures were utilized to assess the performance of the proposed method: the R_{NX}[19] curves to evaluate topological preservation, the V-measure[20] to evaluate cluster induction, and Accuracy[21] to assess classification performance. The average of those measures was also considered.

The neural encoders were trained with relevant methods in the DR literature: principal component analysis (PCA),[22] classical multidimensional scaling (CMDS),[23] Laplacian eigenmaps (LE),[24] and locally linear embedding (LLE).[25] In addition to those classical methods, the experimental validation compared the proposed method with the most relevant DR methods in the literature: linear discriminant analysis (LDA),[26] ISOMAP,[27,28] multiple kernel learning (MKL),[29] deep multiple kernel learning (DMKL),[30] GraphEncoder,[13,31] and factor analysis (FA),[32] as well as with recent DR methods based on projections and manifold approximations: SliseMap,[33] TriMap,[34] DensMap,[35,36] ParametricUMAP,[11] t-distributed stochastic neighbor embedding (t-SNE),[37] context-relevant self-organizing maps (CRSOM),[38] soft-supervised topological autoencoder (STA),[39] and ScaledPCA.[40]

The rest of this article is organized as follows. Section 2 presents an overview of the main concepts related to DR methods and the main characteristics that make a DR method versatile and efficient. Section 3 discusses the combination of DR methods using MKL and ensemble learning. In Section 4, the proposed method, referred to as NetDRm, is described, including its two constituent stages. Section 5 and 6 present the experimental setup and the obtained results, respectively. Finally, conclusions and future research lines are given in Section 7.

2. Dimensionality Reduction

The goal of DR is to generate embeddings in a low-dimensional space in such a way that the relevant information of the high-dimensional input dataset is preserved. Its advantages are multiple. On the one hand, data is compressed by focusing on relevant information and discarding irrelevant information. This also implies a reduced computational cost in the application of various machine learning algorithms, since the calculations are performed over a lower number of dimensions. Another benefit for machine learning algorithms, and in particular for neural networks, is that overfitting is reduced, leading to more simple and general models.

DR is also very useful as an information visualization technique, since it allows rearranging multidimensional data in a space of two or three dimensions compatible with the human perception capabilities, while maintaining the intrinsic structure of the data, that is, neighboring points in the high dimension keep being neighbors in the low dimension, just as distant points retain their topological difference. In addition, DR allows the induction of groupings (clusters). This ensures the homogeneity and completeness of the data and prevents wrong point overlaps. Finally, DR is being used as a generative approach: instead of using generative adversarial networks (GANs), it is possible to use different low-dimensional embeddings in a multimodal and supervised manner. The underlying idea is that nearby points of image, audio, or text embeddings under the same

concept can generate new points through a vector integration of those points, hence transmitting the features of each other.[41]

DR methods have traditionally followed two approaches: feature selection and feature projection (also known as feature extraction). The goal of feature selection is determining a subset of the original dimensions that optimizes the performance of a certain data analysis algorithm (classification, regression, etc.). This can be done by following either filter, wrapper, or embedded strategies.[42] However, suppressing dimensions may not preserve the underlying structures of the input data in general. The alternative is feature projection, which maps the original dimensions into a lower dimensional space.

2.1. Online Methods

There are two main approaches for dataset processing: offline and online. Offline (also discontinuous) processing involves having access to the entire dataset (full batch). This approach is unsuitable when large datasets must be processed due to the lack of storage capacity in main computer memory. In addition, data may be constantly changing over time, which implies processing algorithms that must continuously adapt to such variations.

Alternatively, online (also continuous, incremental, or adaptive) processing is used when it is not feasible to process the entire dataset, or when the full dataset is not directly available, or it is available as small subsets at a time. In this approach, data points are processed individually or in small subsets called minibatches.[43] The main advantage of online processing is the versatility in managing computer memory. However, there are also disadvantages: the results may depend on the order in which individual data points or minibatches are presented, and the applied optimization procedures may be trapped in local optima without guaranteeing optimal solutions.

2.2. Topological Preservation

Topology is the underlying structure of the distribution of data points in a known D-dimensional space from a discrete (finite) sample. Topological preservation refers to the ability to maintain neighborhoods of points, by minimizing both extrusions (points moving away from nearby points in a neighborhood) and intrusions (points moving toward distant points in a neighborhood).

In topology, local and global qualities are studied and different properties are extracted from both. For example, two surfaces (topological varieties of dimension 2) are locally homeomorphic if the surface is homeomorphic to a piece of plane around any point. However, two homeomorphic surfaces can be completely different globally. For instance, if we zoomed in on a surface and moved onto it, the separation between sections would be indistinguishable, as only a small surrounding environment would be observed. A local analysis could thus be carried out. On the other hand, if we zoomed out from the same surface, it would be possible to distinguish general properties and look at the surface in its context. Therefore, a global analysis of the surface could be carried out.

The goal of DR methods is to extract relevant information from high-dimensional data in a low-dimensional space while preserving the topology through deformations or isometries (e.g., rotations, translations, reflections, stretching, bending, shrinking). The intrinsic structure of the manifold connectivity must not be altered. Therefore, the measures of angle, area, length, or volume should be preserved. For example, a circle is topologically equivalent to an ellipse, just as a triangle is to a square, since it is possible to transform one into another continuously.

Several types of DR methods preserve the topological structure of the data in low dimensions. Some of them have an intrusive behavior that leads to a better global performance, making distant points become neighbors. In sum, they "crush" the manifold. The global aggregate topology shows all the related nodes and their shared properties.[44] In turn, other methods have an extrusive behavior that tends to "tear" the manifold. In other words, some close neighbors may be embedded far away from each other. These methods have a better local performance.[45]

For example, if the goal is to map the well-known Swiss Roll 3D point dataset, **Figure 1**a, from 3D to 2D with a pure global approach, the result would crush the structure, as shown in Figure 1b. If a pure local approach was used instead, the structure would tend to expand, as shown in Figure 1c. In this case, heterogeneous, global and local approaches, such as the proposed technique or ISOMAP, are necessary. In the end, unrolling such a complex structure is just a matter of applying an appropriate tradeoff between attractive and repulsive forces.

An emerging approach for preserving the topology of data uses similarity matrices. From the point of view of graph theory, data can be represented by a nondirected, weighted graph, in which nodes represent data points, and a similarity or affinity matrix keeps the edge weights. Two pioneering methods that

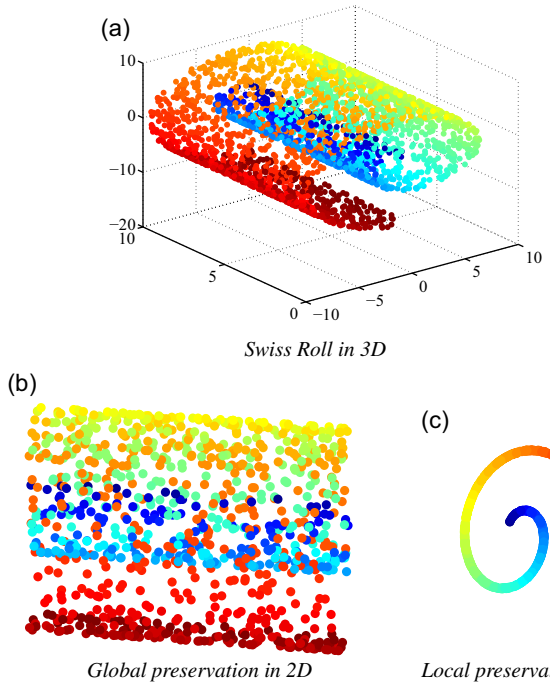

Figure 1. DR of the Swiss Roll 3D point dataset, with DR methods featuring global and local topological preservation. a) shows the data in the original dimension, b) presents the 2-dimensional embedding with global topology, c) presents the 2-dimensional embedding with local topology.

exhibit a local behavior and apply similarity matrices are L[24] and LLE.[25] They follow a spectral approach based on kernels and exhibit local behavior.

2.3. Cluster Induction

It is necessary to differentiate between the clustering task and the iterative process of cluster induction. Clustering algorithms do not transform the input data. They identify clusters based on different principles: calculating distances or similarities between cluster members, recognizing dense areas of the data space, using intervals or particular statistical distributions, finding common characteristics or attributes.

Alternatively, cluster induction aims to generate dense, non-overlapping clusters. In general, clustering can be formulated as a multiobjective optimization problem, where one seeks to maximize the similarity within clusters and minimize the similarity between clusters. On the other hand, cluster induction applies transformations to the given data, arranging closely and/or densely those points that are similar, so that a clustering method can perform a more efficient grouping later. Cluster induction can be applied to a wide range of fields, such as customer segmentation in marketing, social network analysis, bioinformatics, computer vision, and many other areas.

An example of cluster induction is dimension reduction. It transforms the high-dimensional input space into a low-dimensional space, and in the process, it generates a latent characteristic space that represents the original space. The new embedded space is obtained by preserving the topological affinity of elements with similar characteristics. This is the fundamental idea of clustering.

Deep cluster induction is an advanced variation that combines the power of deep learning with the ability to discover latent and complex structures in datasets. Neural models are able to learn relevant and nonlinear features of objects, thus capturing more complex intrinsic patterns and relationships. In deep cluster induction, a neural network is trained to learn a compact and expressive representation of a given dataset. Once the model is trained, a traditional clustering technique can be applied to the learnt features to group objects into clusters.

3. Ensemble Learning

As discussed above, a large variety of DR methods based on different operating principles have been proposed in the literature. Each method exhibits particular characteristics, among which topological preservation and cluster induction are the most relevant. By synergistically combining different DR methods, it is possible to obtain an enriched embedding capable of preserving the inherent properties of each method. The aim is that the combined method (ensemble) outperforms the scores achieved by the individual methods.

A widely used approach for combining DR methods is m*MKL*.[29] It combines several kernel methods into a single embedding matrix, improving the individual representation. MKL is based on a basic principle of kernels: the sum of kernels yields a new kernel. Under the MKL approach, it is possible to combine traditional spectral methods, such as kernel LE (KLE), kernel LLE (KLLE), kernel PCA (KPCA), kernel CMDS (KCMDS), among others. Each combined kernel is associated with a weighting factor that determines the implication of the corresponding method in the final result. The final kernel combination is usually determined by iterative optimization procedures. Different neural approaches have also been proposed for MKL. For example, ref. [30] uses multilayer networks to combine sets of kernel methods shown to be effective in DR.

In machine learning, the three main approaches for combining methods (i.e., ensemble learning) are bagging, boosting, and stacking.[8,9] In *bagging*, different subsets of the same dataset are defined, and several instances of a same method are trained on each subset. The different instances are finally combined by averaging their results or applying majority voting. A representative example of bagging is the Random Forest algorithm. In turn, boosting sequentially trains multiple instances of a same learning method with the entire dataset, although every new instance focuses on the points misclassified by the previous instance. Afterward, the predictions of all instances are combined by averaging or voting. An example of this approach is AdaBoost. Finally, *stacking* combines a set of methods independently pretrained on a same dataset by means of a metamodel. The latter can be any machine learning method. A representative example of stacking is neural networks, which combine in a nonlinear way linear models associated with every neuron.

4. Synergistic Integration of DR Methods through Deep Neural Networks

We present NetDRm, a DR method based on deep networks that allows the integration of multiple DR methods. In particular, we integrate four well-known spectral DR methods with NetDRm: PCA/KPCA, CMDS/KCMDS, LLE/KLLE, and LE/KLE. We chose those precursor methods for several reasons. First, they are suitable to be combined through MKL,[29] which is the reference integration tool in this field. They are also simple linear algorithms that preserve the global topology (CMDS/KCMDS, PCA/KPCA) or the local topology (LE/KLE, LLE/KLLE). Global and local topological preservation are desirable features for any embedding. Finally, the embeddings generated by these methods are typically used as the starting point for more recent DR methods, such as t-SNE, TriMap, and UMAP.

NetDRm is a neural DR method with two variations: a discriminative version aimed at cluster induction and a topological version that preserves the structure of the high-dimensional data.[46] NetDRm uses ensemble learning to synergistically integrate DR methods through deep neural combination, with the goal of endowing the new embedding with the features of its precursor methods, along with the features of deep learning. The latter efficiently uses computer memory by processing data into subsets and allows new points to be projected based on the learned model.

NetDRm has two stages. In the first stage, the behavior of the base DR methods is learnt using dense neural encoders from the embeddings generated by each precursor method. Those encoders are subsequently optimized with UMAP-based approximation procedures. In the second stage, a deep neural metamodel integrates the embeddings generated by the previously

trained encoders. Depending on the configuration of the metamodel, two versions are generated: a discriminative version that applies a supervised approach to separate the given points into clusters and a topological version that applies an unsupervised approach to preserve the structure of the high-dimensional data. The two stages are fully described below.

4.1. Learning DR Methods

In the first stage, M deep encoders $\{\varepsilon_1, \ldots, \varepsilon_M\}$ learn the M embeddings $\mathbf{Y}_{DR} \in \mathbb{R}^{N\times d}$ generated by M DR methods $\{DR_1, \ldots, DR_M\}$, all applied to the same high-dimensional dataset ${}_s\mathbf{X} \in \mathbb{R}^{N\times D}$, as shown in **Figure 2**.

Since NetDRm is applicable to both structured and unstructured datasets, a preprocessing stage guarantees that the different data types are transformed into matrix form. Each basis DR method to be combined is individually applied to the high-dimensional input set $\mathbf{X}$ to generate its respective $\mathbf{Y}_{DR}$ embedding. Subsequently, the input set $\mathbf{X}$ of N points is partitioned into subsets of b points $\overline{\mathbf{X}} \in \mathbb{R}^{b\times D}$, generating N/b *minibatches*. Each minibatch $\overline{\mathbf{X}}$ gives rise to the corresponding embedding $\overline{\mathbf{Y}}_{DR} \in \mathbb{R}^{b\times d}$. Working with minibatches instead of the full dataset reduces overfitting, gradient calculation and parameter update is more effective, and memory management is optimal.

A different neural encoder ε_{DR} learns each of the previously generated $\mathbf{Y}_{DR}$ embeddings, yielding the online version of each embedding, $\overline{\mathbf{Y}}_{\varepsilon} \in \mathbb{R}^{b\times d}$, which allows new data to be projected without recomputing the whole embedding. The encoder ε_{DR} must be general purpose, that is, capable of processing a wide variety of structured and unstructured datasets. Therefore, a dense neural network based on the structure shown in **Figure 3** is utilized.

A loss function compares the result obtained by the encoder $\overline{\mathbf{Y}}_{\varepsilon}$ and the embedding generated by the DR method $\overline{\mathbf{Y}}_{DR}$:

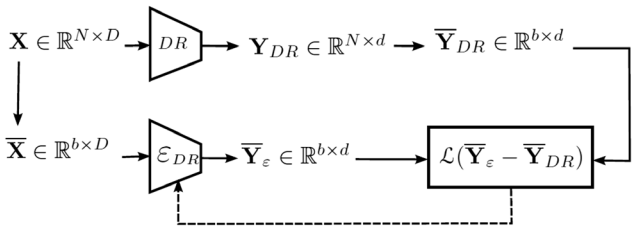

Figure 2. Learning a DR method through an associated neural encoder ε_{DR}.

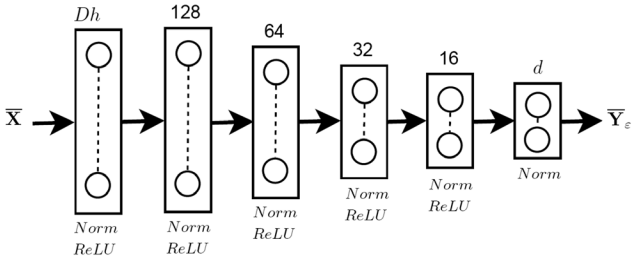

Figure 3. Structure of general purpose encoder ε_{DR}.

$\mathscr{L}(\overline{\mathbf{Y}}_{\varepsilon} - \overline{\mathbf{Y}}_{DR})$. $\mathscr{L}$ can be any regression loss function.[47] The experimental validation conducted in this work was performed with the Smooth L1 function,[48] which shows more stable results in the presence of outliers. The Adam[49] algorithm was used for minimizing the difference between both embeddings at every iteration.

Four classical DR methods were used to train the encoders in this work: PCA, CMDS, LLE, and LE. They are the most relevant and representative spectral DR methods. Their online versions were generated by means of the respective encoders: ε_{PCA}, ε_{CMDS}, ε_{LLE}, and ε_{LE}.

After training the four encoders, they are optimized by applying ParametricUMAP.[11] The goal is to maintain the integrity of high-dimensional clusters. Initially, the algorithm computes distances between every pair of high-dimensional points. Subsequently, a graph is created to represent the nearest neighbors, assigning probabilities to potential edges between points and establishing a local concept of distance. ParametricUMAP applies the following processes. First, it calculates a distance matrix $\mathbf{D}_{\mathbf{X}} \in \mathbb{R}^{N\times N}$.

$$\mathbf{D}_{\mathbf{X}}(i,j) = \|\mathbf{X}_i - \mathbf{X}_j\| \tag{1}$$

A graph consisting of the k nearest neighbors for each point in $\mathbf{X}$ is also generated, where k is a predetermined hyperparameter. Using the graph and the distance matrix $\mathbf{D}_{\mathbf{X}}$, a matrix of similarity scores $\mathbf{P}_{\mathbf{X}} \in \mathbb{R}^{N\times N}$ is computed for each point and its neighbors as follows.

$$\mathbf{P}_{\mathbf{X}}(i,j) = e^{-(\mathbf{D}_{\mathbf{X}}(i,j)-\rho_i)^2/\sigma_i} \tag{2}$$

where ρ_i represents the distance to the nearest neighbor of $\mathbf{X}_i$ in the graph, and σ_i is a parameter used to standardize the similarity scores of each point. Specifically, σ_i is determined such that the total sum of similarity scores $\mathbf{P}_{\mathbf{X}}(i,j)$ between $\mathbf{X}_i$ and its k nearest neighbors in the diagram is less than $\log_2(k)$. It is important to note that the matrix of similarity scores $\mathbf{P}_{\mathbf{X}}$ is asymmetric. To avoid this, a new symmetric matrix $\mathbf{G}_{\mathbf{X}} \in \mathbb{R}^{N\times N}$ of similarity scores for the entire dataset is constructed using the element-wise (Hadamard) product o.

$$\mathbf{G}_{\mathbf{X}} = (\mathbf{P}_{\mathbf{X}} + \mathbf{P}_{\mathbf{X}}^T) - \mathbf{P}_{\mathbf{X}} \,\mathrm{o}\, \mathbf{P}_{\mathbf{X}}^T \tag{3}$$

Both $\mathbf{D}_{\mathbf{X}}$ and $\mathbf{G}_{\mathbf{X}}$ are calculated once for the entire dataset. However, the neural encoders are applied to different subsets of b data points $\overline{\mathbf{X}} \in \mathbb{R}^{b\times D}$ that are randomly sampled from $\mathbf{X} \in \mathbb{R}^{N\times D}$. Therefore, each time an encoder is utilized, we focus on the distance submatrix and symmetric similarity score submatrix corresponding to the sampled points within the minibatch: $\mathbf{D}_{\overline{\mathbf{X}}} \in \mathbb{R}^{b\times b}$ and $\mathbf{G}_{\overline{\mathbf{X}}} \in \mathbb{R}^{b\times b}$. These submatrices are directly extracted from $\mathbf{D}_{\mathbf{X}}$ and $\mathbf{G}_{\mathbf{X}}$, respectively.

In the next step, a low-dimensional graph is created to represent the connection between high-dimensional points. This is achieved by repositioning the low-dimensional points to mimic the groups established in the original space. ParametricUMAP selects two points within a group based on their high-dimensional scores and brings one closer to the other. Afterward, the method identifies a point that should move away, selecting the one with the lowest score in its respective

dimensional group. The probabilities in the embedded space are determined as follows.

$$\mathbf{Q}_{\overline{\mathbf{Y}}_\varepsilon} = \left(1 + \alpha \|\overline{\mathbf{Y}}_\varepsilon - \overline{\mathbf{Y}}_\varepsilon^T\|^{2\beta}\right)^{-1} \tag{4}$$

Low-dimensional similarity scores are derived from a constant bell-shaped t-distribution curve. All edges are considered to have a probability of 1, while negative samples have a probability of 0. Hyperparameter α controls the shape of the t-distribution: as α increases, the t-distribution approaches a standard normal distribution. In turn, hyperparameter β determines how the contribution of the squared norm is weighted in the overall expression, allowing the sensitivity of the model to be adapted to differences in the data.

Finally, a crossentropy cost function C is computed, and optimization is performed through gradient descent to minimize the difference between both similarity scores, $\mathbf{G}_{\overline{\mathbf{X}}}$ and $\mathbf{Q}_{\overline{\mathbf{Y}}_\varepsilon}$.

$$\begin{aligned}\mathcal{C}\left(\mathbf{G}_{\overline{\mathbf{X}}}, \mathbf{Q}_{\overline{\mathbf{Y}}_\varepsilon}\right) = &-\sum_{i \neq j} \mathbf{G}_{\overline{\mathbf{X}}}(i,j) \log \frac{\mathbf{G}_{\overline{\mathbf{X}}}(i,j)}{\mathbf{Q}_{\overline{\mathbf{Y}}_\varepsilon}(i,j)} \\ &+ \left(1 - \mathbf{G}_{\overline{\mathbf{X}}}(i,j)\right) \log \frac{1 - \mathbf{G}_{\overline{\mathbf{X}}}(i,j)}{1 - \mathbf{Q}_{\overline{\mathbf{Y}}_\varepsilon}(i,j)}\end{aligned} \tag{5}$$

The cost function is applied to all pairs of points within the minibatch. Pairs exhibiting a high similarity score $\mathbf{G}_{\overline{\mathbf{X}}}(i,j)$ are indicative of nearby points in the high-dimensional (HD) space. Conversely, pairs with a low $\mathbf{G}_{\overline{\mathbf{X}}}(i,j)$ represent distant points in the HD space. By minimizing this crossentropy function, nearby points in the HD space are encouraged to converge in the low-dimensional (LD) space (attraction), while distant points in the HD space are prompted to spread apart in the LD space (repulsion). Consequently, in this initial stage, an enhanced embedding $\overline{\mathbf{Y}}_\varepsilon \in \mathbb{R}^{b \times d}$ is derived via the neural encoder associated with each specific DR technique under consideration.

In the end, each considered DR method yields an enhanced embedding denoted as $\overline{\mathbf{Y}}_{\mathrm{DR}}$.

4.2. Combining DR Methods

The second stage applies a metamodel, α, based on neural ensemble learning for integrating the embeddings generated by the M encoders $\{\varepsilon_1, \ldots, \varepsilon_{\mathrm{M}}\}$ previously trained in the first stage. The integration network, α, represented in **Figure 4**, is a residual neural network with skip connections to reduce vanishing gradient. The network processes data through densely connected layers.

H_{b} is the number of residual hidden blocks. It is a hyperparameter that determines the depth of the network. In turn, H_n is a hyperparameter that defines the number of hidden neurons. The residual network, α, receives the M neural embeddings $\overline{\mathbf{Y}}_{\mathrm{M}} \in \mathbb{R}^{b \times d}$, each of d dimensions. Thus, the input layer consists of $M \times d$ neurons, followed by a sequence of residual blocks. Each residual block consists of two layers of H_n hidden neurons. H_n is determined according to the number of methods to be combined, the desired embedding dimension, and a regularization parameter η as follows: $H_n = \eta M d$.

The result of each residual block is normalized and a nonlinearity is added through the rectified linear unit activation function. Finally, the output layer has d neurons, which is the desired embedding dimension. As a result of the ensemble, a low-dimensional integrated output $\overline{\mathbf{Y}} \in \mathbb{R}^{b \times d}$ is obtained, which can be either discriminative or topological depending on the metamodel configuration as described below.

4.2.1. Discriminative Version: NetDRm$_D$

For cluster induction, the metamodel, α, must be configured with a large number of layers and neurons. The following parameters were utilized in this work (see Figure 4) based on the analysis described in Section 6.2: $H_{\mathrm{b}} = 10$ and $H_n = 10\mathrm{Md}$, that is, 22 fully connected layers corresponding to: one input layer, one output layer, and 20 hidden layers (10 residual blocks, with 2 hidden layers each). In addition, each hidden layer contains 10Md neurons. The depth and number of neurons cause a characteristic effect that yields a better discrimination of the projected points.

The M encoders and the metamodel constitute a deep residual encoder, ε. It is trained in a supervised way with an encoder–decoder topology shown in Figure > 6. That supervision assumes that every original point $\mathbf{X}_i$ belongs to a known class c out of C different classes. The structure of the decoder, δ, is depicted in **Figure 5**. It is a shallow feed-forward network with two successive fully connected layers and a final softmax function. The decoder is fed with a minibatch $\overline{\mathbf{Y}} \in \mathbb{R}^{b \times d}$ generated by the metamodel and yields b class probability vectors, $\mathbf{C} \in \mathbb{R}^{b \times C}$, such that $\mathbf{C}_i(c)$, $c \in [1, C]$, is the estimated probability that $\overline{\mathbf{Y}}_i$ belongs to class c. The ground-truth class of every high-dimensional point $\overline{\mathbf{X}}_i$ and its corresponding embedded point $\overline{\mathbf{Y}}_i$ is defined through a

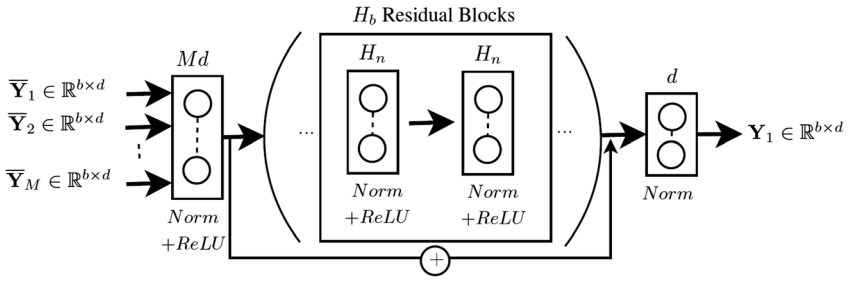

Figure 4. Structure of the deep integration network (metamodel)α.

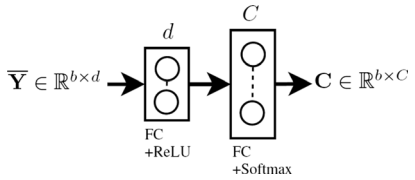

Figure 5. Structure of decoder δ.

one-hot vector $\mathbf{O}_i$, such that $\mathbf{O}_i(c)$ is 1 if $\overline{\mathbf{Y}}_i$ belongs to class c and 0 otherwise.

Ideally, each $\mathbf{C}_i$ should match its corresponding one-hot vector $\mathbf{O}_i$. To achieve this goal, both the metamodel and the decoder are trained by minimizing a crossentropy cost function $\mathscr{L}_C$ given by

$$\mathscr{L}_C = \mathcal{C}'(\mathbf{C},\mathbf{O}) = -\sum_{i=1}^{b}\sum_{c=1}^{C} \mathbf{O}_i(c)\log \mathbf{C}_i(c) + (1-\mathbf{O}_i(c))\log(1-\mathbf{C}_i(c)) \tag{6}$$

The entire process is illustrated in **Figure 6**. The parameters of the M encoders were kept fixed. However, comparable results were obtained by unfreezing all encoders.

4.2.2. Topological Version: NetDRm$_T$

For topological preservation, the metamodel, α, must be configured with a small number of layers and neurons. The following parameters were chosen in this work (see Figure 4) based on the analysis described in Section 6.2: $H_b = 1$ and $H_n = 2\text{Md}$, that is, four fully-connected layers corresponding to one input layer, one output layer, and two hidden layers (one residual block with two hidden layers). In addition, each hidden layer contains 2Md neurons.

The network is trained in an unsupervised manner by considering a weighted compound loss $\mathscr{L}_t$.

$$\mathscr{L}_t = w\mathscr{L}_L + (1-w)\mathscr{L}_G \tag{7}$$

The first term in Equation (7) aims at preserving local topology: $\mathscr{L}_L = \mathcal{C}\left(\mathbf{G}_{\overline{\mathbf{X}}}, \mathbf{Q}_{\overline{\mathbf{Y}}}\right)$ is the crossentropy loss defined in Equation (5). It compares $\mathbf{G}_{\overline{\mathbf{X}}}$ and $\mathbf{Q}_{\overline{\mathbf{Y}}}$ based on the process proposed in ParametricUMAP,[11] which we also apply to the optimization of the pretrained encoders (Section 4.1). $\mathbf{G}_{\overline{\mathbf{X}}}$ is the symmetric matrix of similarity scores defined in Equation (3). It is computed once for the original dataset $\mathbf{X}$ by considering the distance matrix $\mathbf{D}_{\mathbf{X}}$ defined in Equation (1) and a graph with the k nearest neighbors of each point in $\mathbf{X}$. In turn, $\mathbf{Q}_{\overline{\mathbf{Y}}}$ is the conversion to probabilities of the low-dimensional distance matrix $\mathbf{D}_{\overline{\mathbf{Y}}}$, similarly to Equation (4). $\mathbf{Q}_{\overline{\mathbf{Y}}}$ is formulated to assign higher values to shorter distances and lower values to longer distances, with adjustments made based on the provided parameters α and β. It gives values between 0 and 1.

$$\mathbf{Q}_{\overline{\mathbf{Y}}} = \left(1 + \alpha \mathbf{D}_{\overline{\mathbf{Y}}}^{2\beta}\right)^{-1} \tag{8}$$

The second term in Equation (7) aims at preserving global topology: $\mathscr{L}_G = \mathcal{D}\left(\mathbf{D}_{\overline{\mathbf{X}}} - \mathbf{D}_{\overline{\mathbf{Y}}}\right)$ is a regression loss function that aims to minimize the error between the Euclidean distance matrices of both the input $\overline{\mathbf{X}}$ and the output $\overline{\mathbf{Y}}$ data points (i.e., $\mathbf{D}_{\overline{\mathbf{X}}}, \mathbf{D}_{\overline{\mathbf{Y}}} \in \mathbb{R}^{b\times b}$). In our experiments, we chose the *Smooth L1* loss[48] due to its robustness to outliers.

Hyperparameter w in Equation (7) is a weighting factor between 0 and 1 that determines the amount of local ($\mathscr{L}_L$) and global ($\mathscr{L}_G$) topological preservation. If $w = 1$, the system only preserves the local topology, whereas only global topology is preserved for $w = 0$. A tradeoff value $w = 0.5$ was considered in the experiments conducted in this work. Only the metamodel, α, was trained by minimizing Equation (7). Comparable results were achieved by unfreezing all encoders. The entire process is depicted in **Figure 7**.

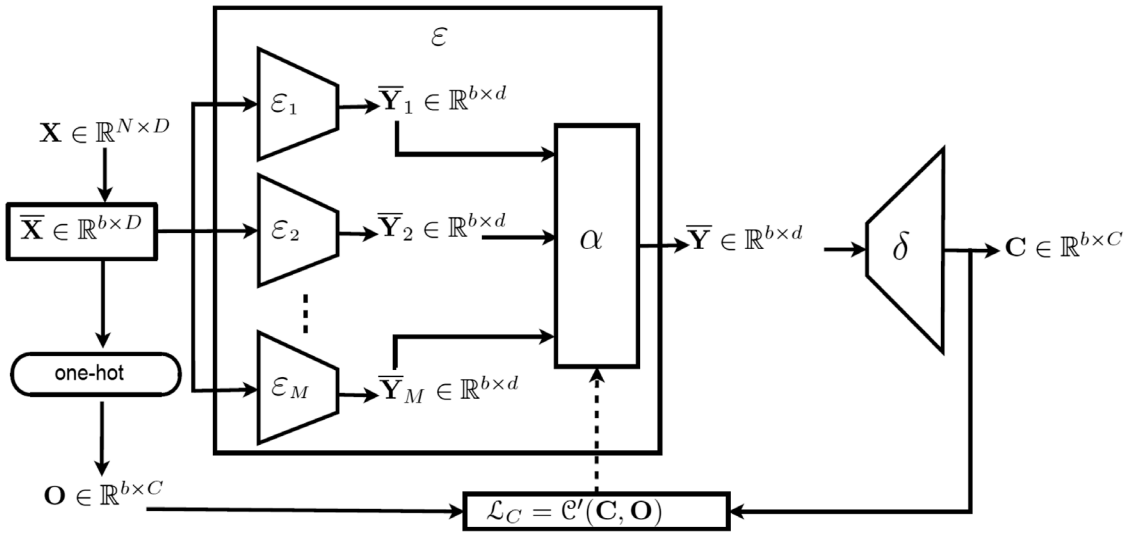

Figure 6. Combination of DR methods through multilayer feedforward network trained in a supervised manner, which exhibits cluster induction.

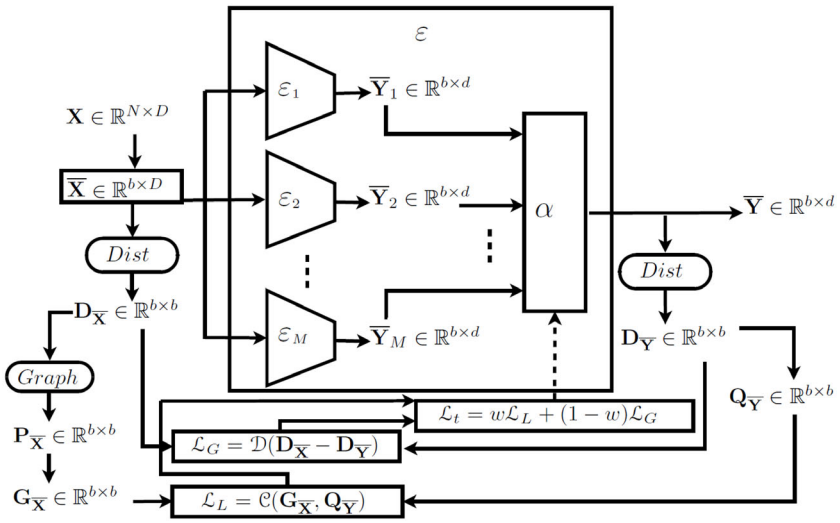

Figure 7. Combination of DR methods through multilayer feedforward network trained with Euclidean distance matrices for global topological preservation and with a neighborhood graph for local topological preservation.

5. Experimental Section

The effectiveness of the proposed DR approach was experimentally assessed on a variety of multidimensional structured (image datasets) and unstructured datasets (manifold-type point datasets and tabular datasets). Due to space limitations, this article presents results corresponding to five representative datasets: Swiss Roll and Sphere (point datasets), ECG Arrhythmia (tabular dataset), and COIL-20 and Fashion-MNIST (image datasets). They are described in Section 5.1. Three widely used measures were utilized to assess the performance of the proposed method: R_{NX} curves to evaluate topological preservation, the V-measure to evaluate cluster induction, and Accuracy (Ac) to evaluate the classification capabilities. Finally, an average (Av) of the previous three measures was also computed. They are described in Section 5.2. The architecture of the encoders and decoders was designed by taking into account the aforementioned datasets, as already explained in Section 4.

5.1. Datasets

We presented both quantitative and qualitative results for two unstructured manifold-type point datasets: Swiss Roll and Sphere. They contained $N = 3000$ 3D points (i.e., $D = 3$) sampled from two well-defined surfaces. Therefore, they are suitable for assessing the behavior of dimension reduction methods when those points are projected to 2D (i.e., $d = 2$). The Swiss Roll can easily be deployed to a 2D surface by unrolling the spiral structure. Conversely, the Sphere is a compact set difficult to deploy, as the points are closely related to each other. Each data point in those datasets has an associated color from a 15 color palette. We used these colors as class labels to evaluate the classification accuracies.

An unstructured, tabular dataset with features extracted from signals for ECG Arrhythmia detection was also considered. The dataset has $N = 30\,000$ records with $D = 34$ dimensions. The records belong to $C = 4$ different classes: N (Normal), S (Supraventricular ectopic beat), V (Ventricular ectopic beat), and F (Fusion beat) (https://www.kaggle.com/datasets/sadmansakib7/ecg-arrhythmia-classification-dataset).

We also showed the results for two structured image datasets: COIL-20 and Fashion-MNIST. COIL-20 (Columbia Object Image Library)[50] contains $N = 1440$ images of $C = 20$ simple objects, such that every object was rotated 360 degrees on a turntable in 5-degree steps, yielding 72 images per object. The images were monochrome with a resolution of 32×32 pixels (i.e., $D = 32 \times 32 = 1024$). In turn, Fashion-MNIST[51] consisted of a training set with 60 000 images and a test set with 10 000 images, all belonging to 10 different categories corresponding to various Zalando's articles. In this case, $N = 60\,000$, with a resolution of 28×28 pixels (i.e., $D = 28 \times 28 = 784$). **Figure 8** shows examples of those point and image datasets.

5.2. Quality Measures

As mentioned above, DR methods aim to find a mapping (embedding) that projects the original high-dimensional points into a low-dimensional space. As a result of that mapping, the data points undergo topological transformations (e.g., stretching, breaking, approximations, distancing) that modify the structural relationships among them. Therefore, one of the desirable features in any embedding is the ability to preserve the topological structure of the high-dimensional data. Furthermore, another desirable feature of DR methods is their ability to induce clusters by modifying the topological structure in the low-dimensional space. This is not to do with clustering methods nor classification algorithms. Clustering methods aim at identifying or recognizing clusters in the original data, without modifying their topological structure. In turn, classification algorithms aim to assign a label to each original data point in a supervised manner, without modifying the topological structure of the input data. In turn, a DR method can be applied to either a clustering method or a

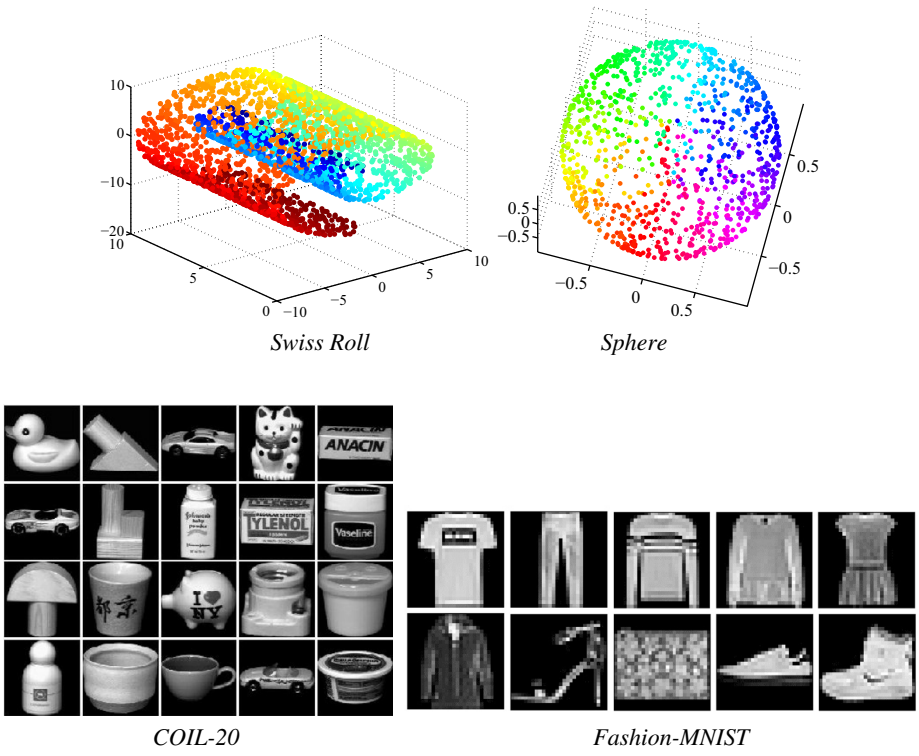

Figure 8. Illustration of the point-cloud and image datasets tested in this work: Swiss Roll, Sphere, COIL-20, and Fashion-MINIST.

classification algorithm as a preprocessing stage, since it condenses the most relevant information in such a way that the subsequent clustering or classification stage can be more efficient and less prone to overfitting. In sum, differently to clustering and classification methods, DR methods embed the original points into a low-dimensional space while attempting to preserve their original spatial relationships. That preservation ability can be directly evaluated through suitable topological preservation measures.

Topological preservation and cluster induction are not mutually exclusive: one can affect the other. For example, if the objective is data segmentation, the generation of dense subgroups will cause similar points to concentrate in certain regions, leading to extrusions or intrusions,[52] which will affect the preservation of the original structure. For this reason, a specific measure must be applied to each feature.

In this work, the topological preservation of high-dimensional data has been evaluated with the R_{NX} curves. The R_{NX} curves generate scores in terms of percentages. However, in order to obtain a score comparable to the other metrics, the R_{NX} score was normalized in such a way that 1 represents the best result. In turn, the *V*-measure has been used to evaluate the ability to induce clusters.[53] In practical terms, the well-known K-means algorithm was applied to the low-dimensional points generated by each evaluated DR method in order to generate clusters. The points contained in each cluster were then given a same label. Finally, those automatically generated labels were compared with the real labels. The *V*-measure ranged between 0 and 1, with 1 being the optimal score. Finally, Accuracy (Ac) was used to evaluate the benefits of the generated low-dimensional embeddings for subsequent classification applications. In this case, all datasets were split into a training subset (80% of data) and an evaluation subset (20% of data). The well-known decision tree classifier was then trained to learn the classes of the low-dimensional points contained in the training subset. Afterward, that classifier was applied to predict the class labels of the low-dimensional points in the test subset. Accuracy was computed by comparing the predicted labels with the ground-truth labels. Due to the strong DR applied to the original data in our experiments (typically 2D embeddings), the obtained

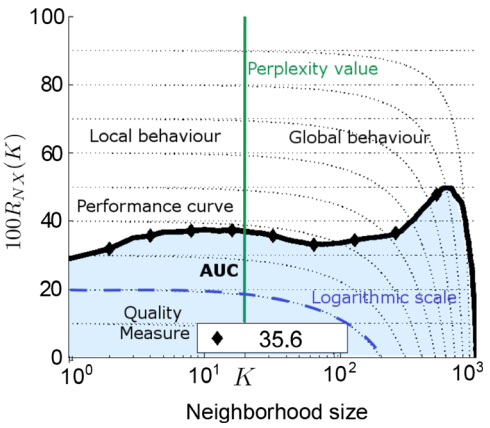

Figure 9. Example of $R_{NX}(K)$ curve and associated AUC measure for assessing the local and global topological preservation of a DR method.

Table 1. R_{NX} score, *V* measure, and Accuracy for different combinations of precursor DR methods with NetDRm.

sNetDRm	Measure	*COIL-20*		*Swiss Roll*		*Sphere*		*Arrhythmia*		*Fashion-MNIST*	
Combination		$NetDRm_T$	$NetDRm_D$	$NetDRm_T$	$NetDRm_D$	$NetDRm_T$	$NetDRm_D$	$NetDRm_T$	$NetDRm_D$	$NetDRm_T$	$NetDRm_D$
PCA/KPCA[1]	*V*	0.815	0.924	0.578	0.932	0.454	0.743	0.426	0.696	0.526	0.655
LE/KLE[1]	R_{NX}	44.9	23.6	46.8	24.1	49.9	22.4	42.3	21.0	31.7	19.4
	Ac	0.247	0.220	0.254	0.195	0.280	0.317	0.573	0.662	0.426	0.278
CMDS/KCMDS[1]	*V*	0.811	0.930	0.594	0.940	0.448	0.756	0.430	0.708	0.524	0.645
LLE/KLLE[1]	R_{NX}	44.4	23.8	48.5	23.6	51.3	21.1	43.4	20.5	32.8	20.3
	Ac	0.244	0.233	0.243	0.186	0.278	0.318	0.563	0.686	0.428	0.280
CMDS/KCMDS[1]	*V*	0.814	0.917	0.591	0.941	0.459	0.762	0.423	0.714	0.531	0.653
PCA/KPCA[1]	R_{NX}	44.8	24.1	50.2	23.2	50.6	21.9	41.4	23.1	32.3	19.6
	Ac	0.238	0.338	0.248	0.201	0.287	0.319	0.576	0.691	0.435	0.274
LE/KLE[1]	*V*	8	0.952	0.582	0.937	0.461	0.752	0.428	0.694	0.518	0.649
LLE/KLLE[1]	R_{NX}	44.1	23.8	46.5	25.0	48.7	20.9	43.0	20.3	31.5	18.3
	Ac	0.246	0.324	0.256	0.193	0.283	0.319	0.569	0.684	0.429	0.261
CMDS/KCMDS[1]	*V*	0.793	0.955	0.623	0.968	0.482	0.771	0.435	0.763	0.554	0.691
PCA/KPCA[1]	R_{NX}	45.3	28.3	59.3	25.6	54.8	25.4	51.2	24.6	35.0	21.1
LE/KLE[1]	Ac	0.275	0.341	0.283	0.214	0.311	0.341	0.607	0.745	0.483	0.311
LLE/KLLE[1]	*V*	0.823	0.941	0.618	0.964	0.483	0.764	0.435	0.768	0.546	0.683
LE/KLE[1]	R_{NX}	46.6	28.4	57.7	27.3	55.6	24.8	53.3	24.7	0.33.7	20.8
CMDS/KCMDS[1]	Ac	0.273	0.338	0.291	0.209	0.308	0.342	0.602	0.724	0.478	0.307
LLE/KLLE[1]	*V*	0.839	0.960	0.634	0.989	0.514	0.839	0.475	3	0.583	0.714
LE/KLE[1]	R_{NX}	50.2	30.6	64.2	28.9	61.8	23.4	56.5	22.8	38.6	22.7
CMDS/KCMDS[1]	Ac	0.290	0.351	0.307	0.225	0.339	0.355	0.636	0.770	0.525	0.350
PCA/KPCA[1]											

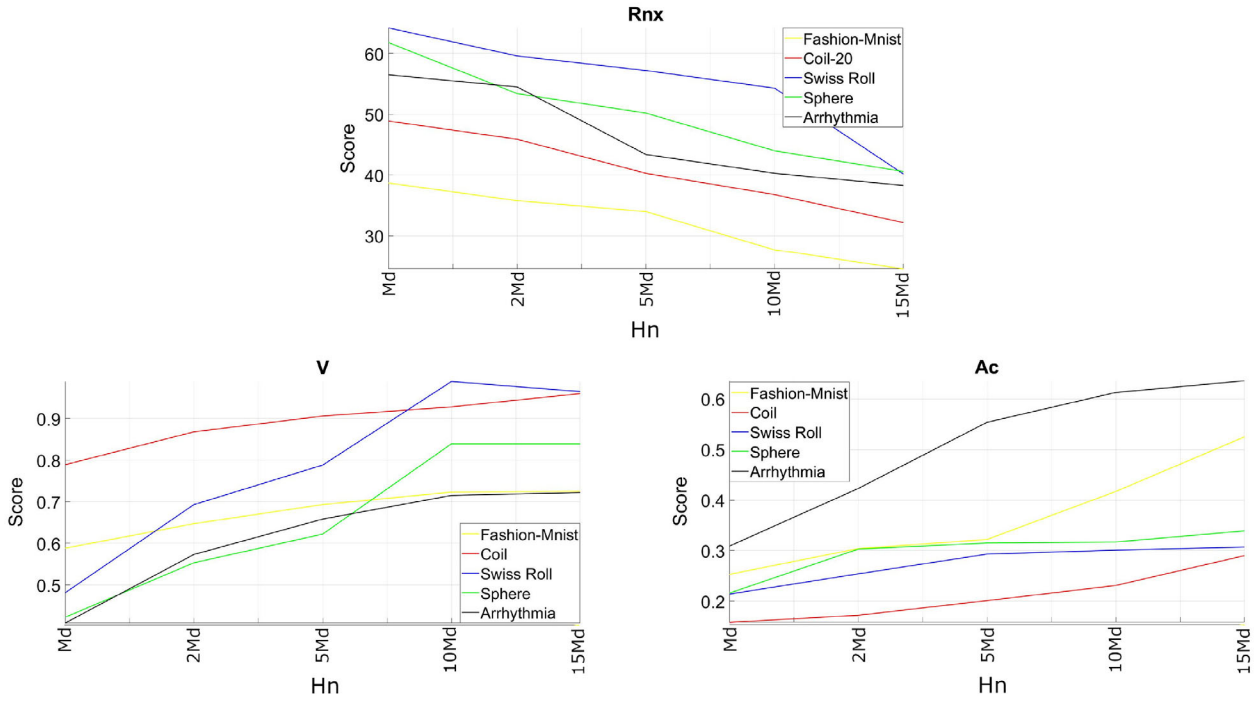

Figure 10. Performance measures for different numbers of neurons in the hidden layers of the metamodel,H_n. R_{NX} scores by considering 2 hidden layers (topological metamodel), *V*-measures by considering 20 hidden layers (discriminative metamodel), Ac scores by considering 2 hidden layers (topological metamodel).

accuracy scores were expected to be much lower that when this measure was applied to the original data. An ideal embedding would be the one that yields the best scores for the three aforementioned measures (i.e., R_{NX}, V, and Ac). For this reason, we finally considered a holistic measure (Av), which was simply obtained as their arithmetic average. These three quality measures are fully described below.

5.2.1. Topological Preservation Measures

To measure the topological preservation of a DR method, the intersection between neighborhoods in the high-dimensional space and their associated neighborhoods in the low-dimensional space are taken into account. If the intersection is complete, and both neighborhoods have the same data points, the embedding is perfect and the topology of the original space is fully preserved. On the contrary, if the intersection between associated neighborhoods is null, there is no topological preservation at all.

Function $Q_{NX}(K)$[53] measures the quality of the topological preservation of an embedding depending on the neighborhood size K (also known as *perplexity*).

$$Q_{NX}(K) = \sum_{i=1}^{N} \frac{\left|\nu_i^K \cap n_i^K\right|}{KN} \tag{9}$$

where ν_i^K is the high-dimensional neighborhood of the i-th data point (with $K-1$ nearest neighbors), and n_i^K is the associated neighborhood in the low-dimensional space.

Unfortunately, $Q_{NX}(K)$ does not give 0 for a random embedding. This is solved by defining the $R_{NX}(K)$ measure, which gives 0 for a random embedding and 1 for perfect topological preservation.

$$R_{NX}(K) = \frac{(N-1)Q_{NX}(K) - K}{N-1-K} \tag{10}$$

If a DR method preserves the topology for low values of K (small neighborhoods), it is said to preserve the local topology. In turn, if it keeps the topology for big values of K (big neighborhoods), it preserves the global topology. The $R_{NX}(K)$ values are plotted for increasing values of K, giving rise to the $R_{NX}(K)$ curves. **Figure 9** shows a typical example. Therefore, an $R_{NX}(K)$ curve graphically shows the topological preservation of a DR method both locally and globally. The overall topological preservation associated with a single $R_{NX}(K)$ curve is finally summarized with a measure known as area under curve (AUC), which is the discrete integral of the curve.

$$\mathrm{AUC}_{\log K}(R_{NX}(K)) = \frac{\sum_{K=1}^{N-2} R_{NX}(K)/K}{\sum_{K=1}^{N-2} 1/K} \tag{11}$$

Table 2. Evaluation of R_{NX} score, V-measure, and Accuracy for different numbers of hidden layers in the metamodel. Results in bold indicate the highest scores obtained in each metric.

Hidden	Measure	*COIL-20*		*Swiss Roll*		*Sphere*		*Arrhythmia*		*Fashion-MNIST*	
blocks		NetDRm_T	NetDRm_D	NetDRm_T	NetDRm_D	NetDRm_T	NetDRm_D	NetDRm_T	NetDRm_D	NetDRm_T	NetDRm_D
$H_b = 1$	V	0.839	0.866	0.634	0.851	0.514	0.612	0.475	0.521	0.583	0.594
($L = 4$)	R_{NX}	48.9	37.2	64.2	34.3	61.8	29.7	56.5	32.4	38.6	33.5
	Ac	0.290	0.318	0.307	0.209	0.339	0.349	0.636	0.770	0.525	0.323
$H_b = 2$	V	0.859	0.874	0.645	0.862	0.575	0.663	0.419	0.544	0.595	0.613
($L = 6$)	R_{NX}	45.6	35.9	62.7	33.4	59.4	27.0	55.3	31.6	34.3	31.2
	Ac	0.304	0.315	0.311	0.218	0.342	0.340	0.642	0.778	0.493	0.341
$H_b = 4$	V	0.884	0.896	0.693	0.885	0.523	0.608	0.452	0.568	0.593	0.620
($L = 10$)	R_{NX}	42.3	34.1	62.2	32.7	58.4	26.9	53.5	30.3	32.5	30.0
	Ac	0.297	0.329	0.319	0.227	0.328	0.347	0.654	0.784	0.517	0.347
$H_b = 6$	V	0.909	0.922	0.739	0.943	0.642	0.745	0.508	0.611	0.602	0.633
($L = 14$)	R_{NX}	40.6	32.5	58.4	30.7	58.1	24.5	51.9	28.9	30.8	27.3
	Ac	0.313	0.345	0.322	0.219	0.344	0.350	0.661	0.783	0.503	0.358
$H_b = 8$	V	0.915	0.946	0.781	0.972	0.636	0.818	0.532	0.691	0.612	0.689
($L = 18$)	R_{NX}	38.9	32.9	57.4	30.3	57.6	24.7	49.8	27.6	28.4	26.1
	Ac	0.323	0.348	**0.336**	0.225	0.351	0.353	0.672	0.788	0.523	0.364
$H_b = 10$	V	0.923	**0.960**	3	**0.989**	0.718	**0.839**	0.552	0.720	0.632	0.703
($L = 22$)	R_{NX}	34.7	30.6	56.3	28.9	56.8	23.4	47.8	24.5	25.1	23.4
	Ac	0.319	0.351	0.324	0.225	0.350	**0.355**	0.677	**0.791**	**0.528**	0.372
$H_b = 12$	V	0.921	0.949	0.796	0.978	0.588	0.812	0.556	**3**	0.654	**0.714**
($L = 26$)	R_{NX}	33.7	29.8.0	56.9	26.5	49.3	22.1	46.3	22.8	23.6	22.7
	Ac	0.320	**0.353**	0.327	0.232	0.346	0.353	0.684	0.770	0.512	0.350

Improvements of $Q_{NX}(K)$ and $R_{NX}(K)$ imply that a DR method can better preserve both the local and global structure of the original data when projected into a lower-dimensional space. This means that the relationships between nearby and distant data points in the high-dimensional space are more faithfully preserved in the low-dimensional representation. The ability to maintain relationships between data points in the low-dimensional projection helps analysts understand patterns, clusters, and structures in the data that would otherwise be difficult to discern. It can also improve the performance of clustering and classification algorithms that process the projected data. By improving $Q_{NX}(K)$ and $R_{NX}(K)$, a DR method can reveal intrinsic structures in the data that other DR methods might miss. This can include subgroup identification, anomaly detection, and other relevant features of the original data.

5.2.2. Cluster Induction Measures

The cluster induction capability of a DR method has been assessed in this work through the *V*-measure.[20] It is necessary to have a partition of the original dataset $\mathbf{X}$ into C classes (clusters). That induces a corresponding partition in the embedded dataset $\mathbf{Y}$. Let $\mathbf{Y}_{GT} = \{\mathbf{Y}_{GT_i}\}_{i=1}^{C}$ be the family of C clusters induced in $\mathbf{Y}$ by the ground-truth partition. The k-means clustering algorithm with $k = C$ was then applied to $\mathbf{Y}$. It yielded a second family of C clusters: $\mathbf{Y}_C = \{\mathbf{Y}_{C_i}\}_{i=1}^{C}$.

Two complementary clustering measures are defined based on the Shannon entropy H: homogeneity and completeness. In this scope, H measures the dispersion of the N given points into C clusters. The maximum entropy occurs if all points are equally distributed among all clusters. Conversely, entropy is zero if all data points fall into a same cluster (no dispersion at all).

Homogeneity is defined as $h = 1 - H(\mathbf{Y}_{GT}|\mathbf{Y}_C)/H(\mathbf{Y}_{GT})$. $H(\mathbf{Y}_{GT})$ is the dispersion of the ground-truth partition, whereas the conditional entropy $H(\mathbf{Y}_{GT}|\mathbf{Y}_C)$ is the dispersion of the points belonging to every k-means cluster among the C ground-truth clusters. When all points in each k-means cluster are concentrated in the same ground-truth cluster, $H(\mathbf{Y}_{GT}|\mathbf{Y}_C)$ is zero, resulting in maximum homogeneity ($h = 1$). In turn, *completeness* is defined as $c = 1 - H(\mathbf{Y}_C|\mathbf{Y}_{GT})/H(\mathbf{Y}_C)$. Here, $H(\mathbf{Y}_C)$ is the dispersion of the k-means partition, whereas the conditional entropy $H(\mathbf{Y}_C|\mathbf{Y}_{GT})$ is the dispersion of the points belonging to every ground-truth cluster among the C k-means clusters. In this case, when all points of each ground-truth cluster are concentrated in a single k-means cluster, $H(\mathbf{Y}_C|\mathbf{Y}_{GT})$ is zero, yielding maximum completeness ($c = 1$).

Finally, both measures are combined into the *V*-measure through their weighted harmonic mean: $V_\beta = (1+\beta)hc/(\beta h + c)$. If β is greater than one, completeness receives more weight. Conversely, if β is less than one, homogeneity is strengthened.

5.2.3. Accuracy

The accuracy measure is one of the most common and basic ways of evaluating classification methods in machine learning. It is defined as the ratio of correct predictions made by the model

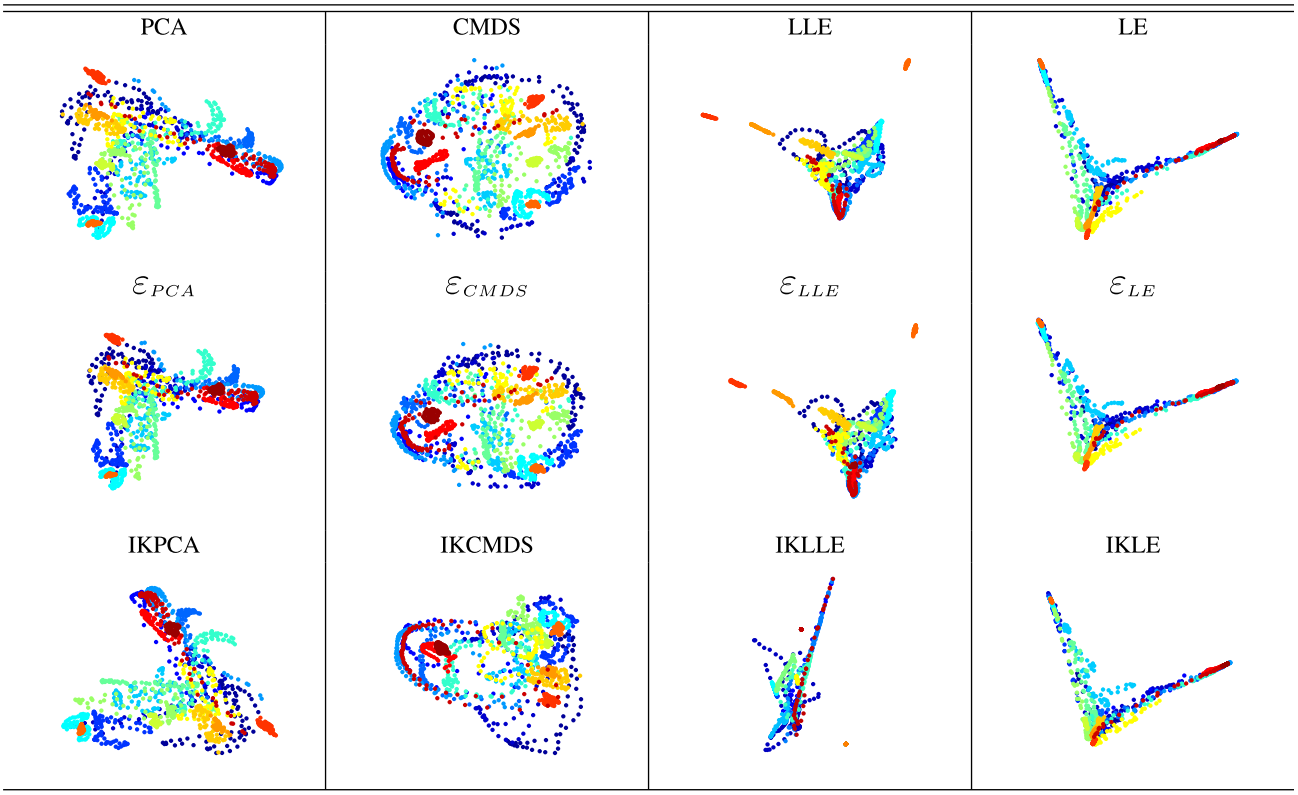

Figure 11. 2D embeddings of COIL-20 generated by PCA, CMDS, LLE, LE, the encoders trained with them, and their incremental versions.

to the total number of predictions. This measure indicates how well the model is classifying the data points. It is defined as

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{12}$$

where TP (True Positives) is the number of correctly classified positive examples, TN (True Negatives) is the number of correctly classified negative examples, FP (False Positives) is the number of negative examples incorrectly classified as positive, and FN (False Negatives) is the number of positive examples incorrectly classified as negative.

6. Experimental Validation

The experimental validation of NetDRm comprises two phases. The first phase is described in Section 6.3. It involves the evaluation of the encoders presented in Section 4.1. The second phase is described in Section 6.4. It evaluates the performance of the proposed integration metamodel detailed in Section 4.2. Previously, Section 6.2 describes the experiments conducted in order to determine the best configuration of the metamodel.

6.1. Different Combinations of DR Methods

The rationale behind ensemble learning is that the combination of diversity is synergistic. **Table 1** shows the results after applying the proposed NetDRm method to different combinations of precursor DR methods. As described in Section 4, we considered four precursor methods, PCA/KPCA, CMDS/KCMDS, LE/KLE, and LLE/KLLE, and combined them in subgroups of 2, 3, and 4 methods. These quantitative results show that the best performances (R_{NX}, V, and Ac) were obtained for the combination of the four base methods. The reason is that each method leaves its footprint in the generated embedding, hence enforcing its own particular features (basically, global or local topological preservation).

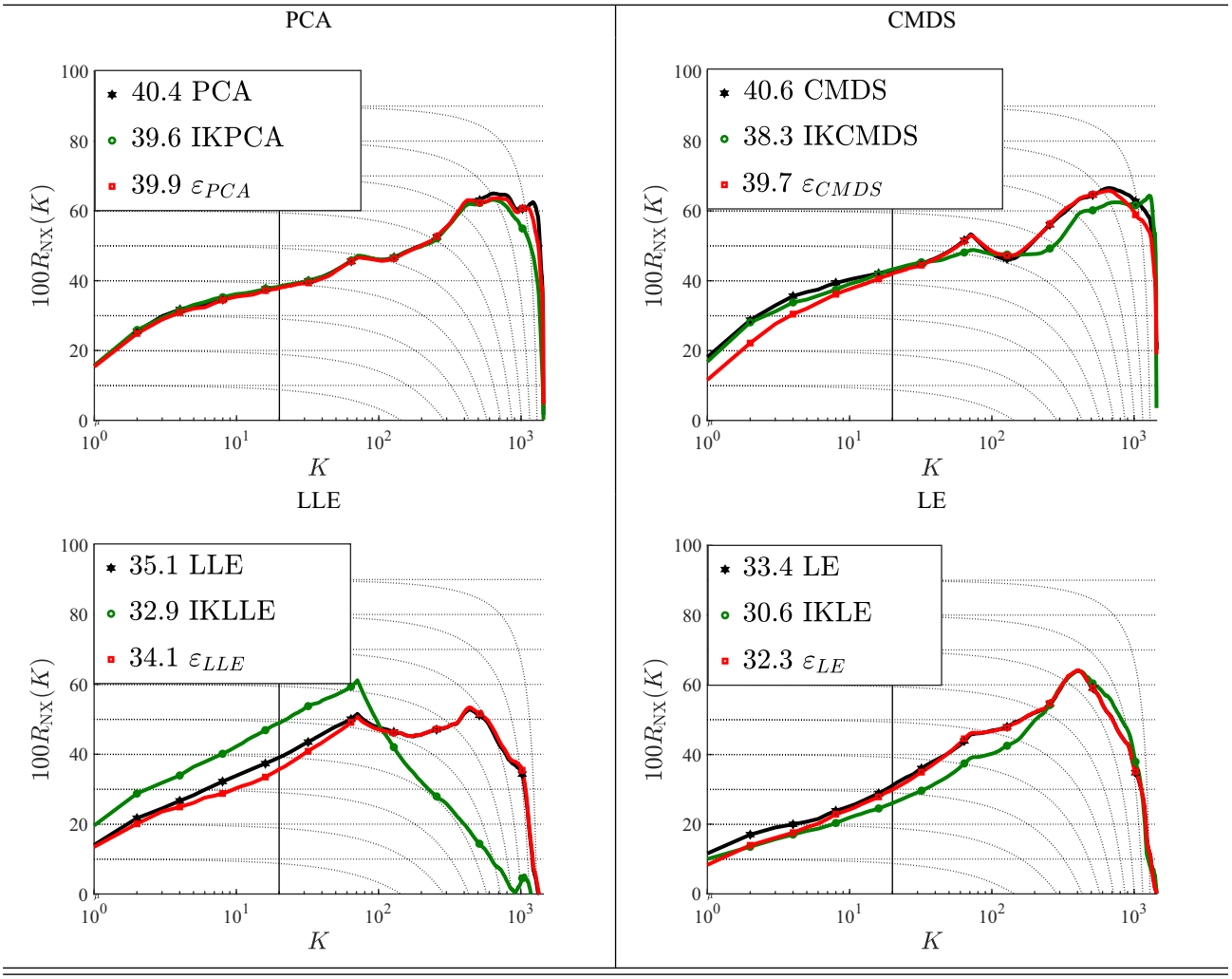

Figure 12. R_{NX} scores for PCA, CMDS, LLE, LE, the encoders trained with them, and their incremental versions for 2D embeddings of COIL-20.

6.2. Best Configuration of the Residual Integration Network (Metamodel)

This section describes the experiments conducted to determine the best configuration of the metamodel described in Section 4.2 (see Figure 4), that is, the number of neurons per hidden layer, H_n, as well as the number of hidden blocks, H_b. The number of neurons per hidden layer was defined as $H_n = \eta M d$, where M is the number of DR methods to be combined, d is the desired embedding dimension, and η is a hyperparameter respectively

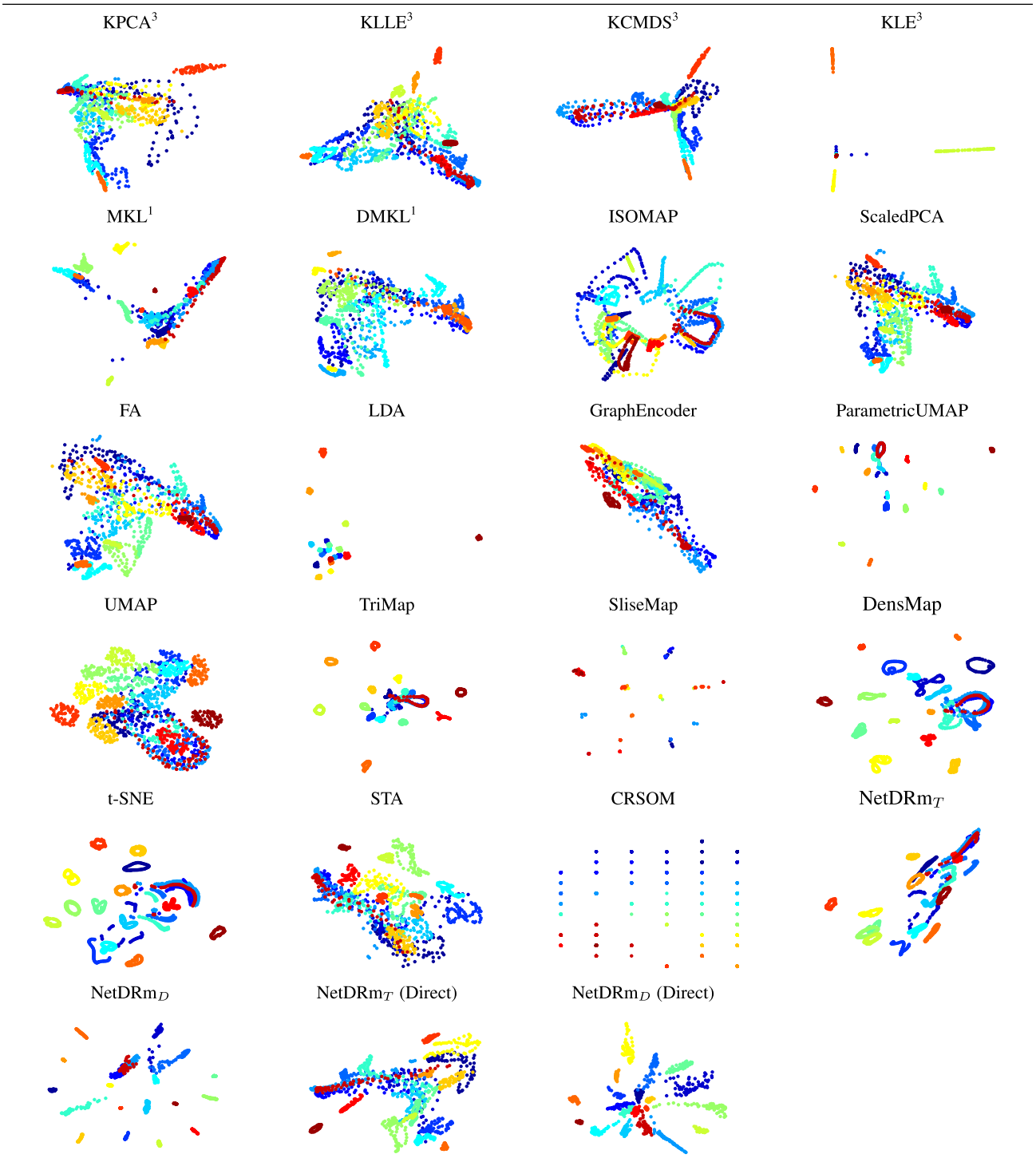

Figure 13. 2D embeddings of tested DR methods applied to the COIL-20 dataset.

chosen for the topological and discriminative versions of NetDRm according to the results presented in **Figure 10**. These experiments assume $M = 4$ (i.e., integration of four DR methods), $d = 2$, and the five evaluated datasets.

For the discriminative version, NetDRm_D, different values of H_n were tested by assuming a configuration with $H_\text{b} = 10$ hidden blocks (i.e., 20 hidden layers). In this configuration, the V and Ac measures were evaluated. The results are shown in Figure 10 (V and Ac). Notice that the scores for both V and Ac stabilize around $\eta = 10$. Therefore, the number of neurons per hidden layer was set to $H_n = 10 \times 4 \times 2 = 80$.

As for the topological version, NetDRm_T, different values of H_n were evaluated by assuming a configuration with $H_\text{b} = 1$ hidden block (i.e., 2 hidden layers). These results are presented in Figure 10 (*Rnx*). In this case, the best R_NX score was attained for $\eta = 1$. Thus, the number of neurons per hidden layer was set to $H_n = 1 \times 4 \times 2 = 8$.

In order to find the best number of hidden blocks, H_b, in the metamodel, a second set of experiments was run by testing different values of H_b. The number of layers of the metamodel is $L = 2H_\text{b} + 2$, that is, two hidden layers per block plus an input and an output layer. Again, we distinguished between the discriminative and topological versions by considering hidden layers with $H_n = 80$ neurons for NetDRm_D and with $H_n = 8$ neurons for NetDRm_T. In all cases, we also assumed $M = 4$, $d = 2$, and the four evaluated datasets. **Table 2** presents the performance measures for all the experiments. For the discriminative version, NetDRm_D, the best V-measure was obtained with $H_\text{b} = 10$ hidden blocks ($L = 22$ layers). In turn, for the topological verison, NetDRm_T, the best R_NX

Table 3. R_NX, V-measure, and Accuracy of tested DR methods applied to the Coil-20 dataset.

Method	R_NX	V	Ac	Av	Method	R_NX	V	Ac	Av
PCA	0.404	0.646	0.250	0.433	GraphEncoder	0.196	0.442	0.283	0.307
CMDS	0.406	0.628	0.314	0.449	MKL[1]	0.376	0.702	0.227	0.435
LE	0.334	0.558	0.311	0.401	DMKL[1]	0.380	0.659	0.224	0.421
LLE	0.351	0.626	0.241	0.406	FA	0.408	0.661	**0.461**	0.510
KPCA[3]	0.366	0.601	0.290	0.419	LDA	0.174	0.883	0.248	0.435
KCMDS[3]	0.337	0.578	0.279	0.398	T-SNE	**0.554**	0.853	0.359	**0.582**
KLE[3]	0.277	0.454	0.245	0.325	STA	0.370	0.707	0.224	0.434
KLLE[3]	0.365	0.651	0.230	0.415	NetDRm_T^1	0.502	0.839	0.290	0.557
ISOMAP	0.341	0.644	0.387	0.457	NetDRm_D^1	0.306	**0.960**	0.351	0.539
DensMap	0.465	0.878	0.231	0.525	NetDRm_T^1(Direct)	0.449	0.797	0.271	0.505
UMAP	0.401	0.707	0.262	0.457	NetDRm_D^1(Direct)	0.287	0.905	0.364	0.519
TriMap	0.456	0.863	0.325	0.548	SliseMap	0.149	0.661	0.245	0.352
ScaledPCA	0.396	0.637	0.352	0.462	ParametricUMAP	0.446	0.882	0.321	0.550
CRSOM	0.284	0.848	0.252	0.461					

Table 4. R_NX, V-measure, and Accuracy of tested DR methods applied to the Fashion-MNIST dataset.

Method	R_NX	V	Ac	Av	Method	R_NX	V	Ac	Av
PCA	0.282	0.421	0.459	0.387	GraphEncoder	0.129	0.368	0.211	0.236
CMDS	0.282	0.422	0.414	0.373	MKL[1]	0.253	0.508	0.462	0.408
LE	0.264	0.557	0.513	0.445	DMKL[1]	0.250	0.355	0.315	0.307
LLE	0.212	0.511	0.338	0.354	FA	0.279	0.32	0.461	0.353
KPCA[3]	0.252	0.427	0.405	0.361	LDA	0.142	0.561	0.520	0.408
KCMDS[3]	0.240	0.400	0.337	0.326	T-SNE	**0.407**	0.563	0.514	0.495
KLE[3]	0.209	0.479	0.107	0.265	STA	0.187	0.424	0.275	0.295
KLLE[3]	0.145	0.366	0.105	0.205	NetDRm_T^1	0.386	0.583	**0.525**	**0.498**
ISOMAP	0.266	0.486	0.424	0.435	NetDRm_D^1	0.227	**0.714**	0.350	0.430
DensMap	0.336	0.599	0.464	0.466	NetDRm_T^1(Direct)	0.298	0.515	0.426	0.413
UMAP	0.314	0.538	0.471	0.441	NetDRm_D^1(Direct)	0.283	0.652	0.322	0.419
TriMap	0.332	0.589	0.523	0.484	SliseMap	0.134	0.631	0.185	0.317
ScaledPCA	0.277	0.415	0.438	0.377	ParametricUMAP	0.340	0.632	0.146	0.373
CRSOM	0.195	0.674	0.107	0.325					

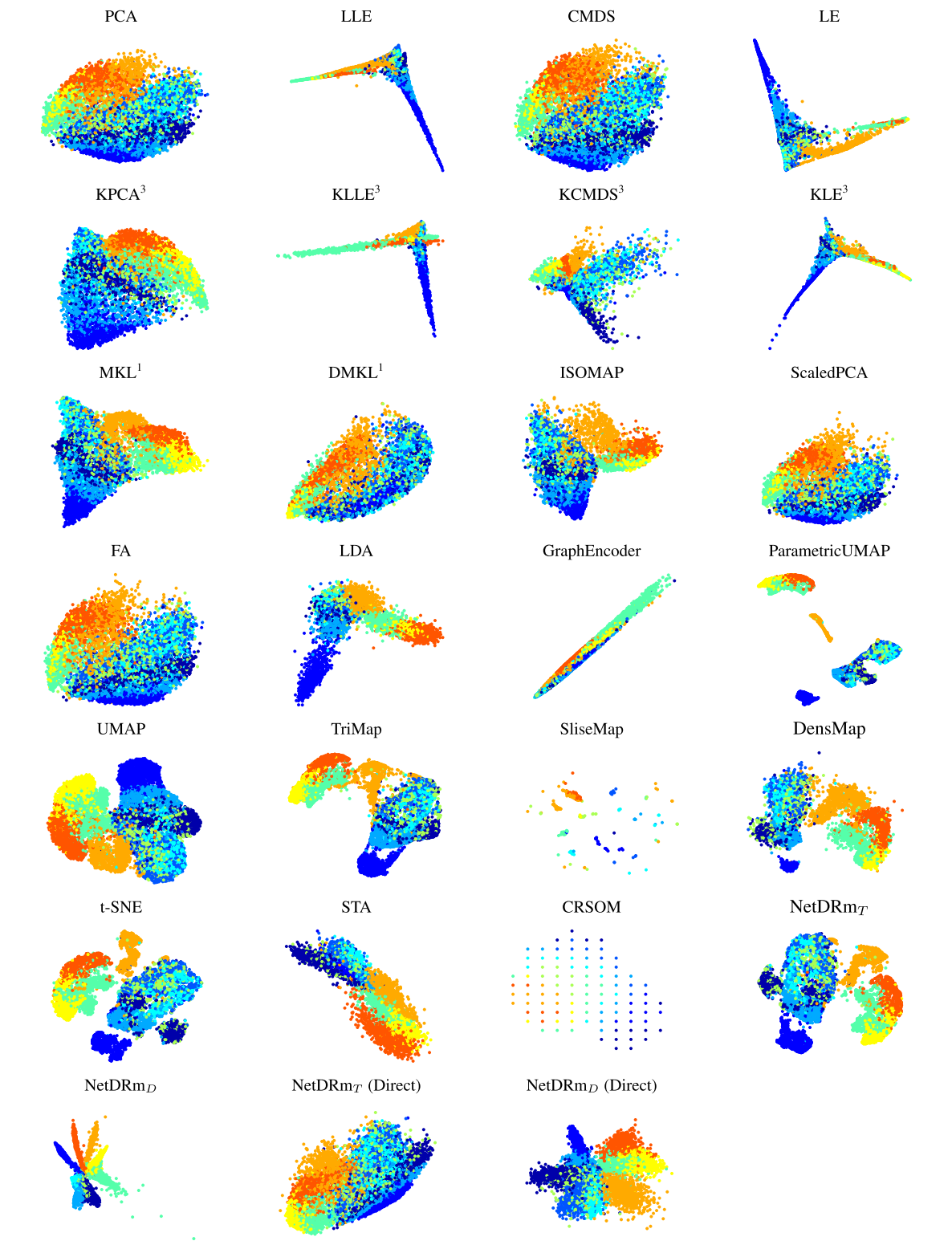

Figure 14. 2D embeddings of tested DR methods applied to the Fashion-MNIST dataset.

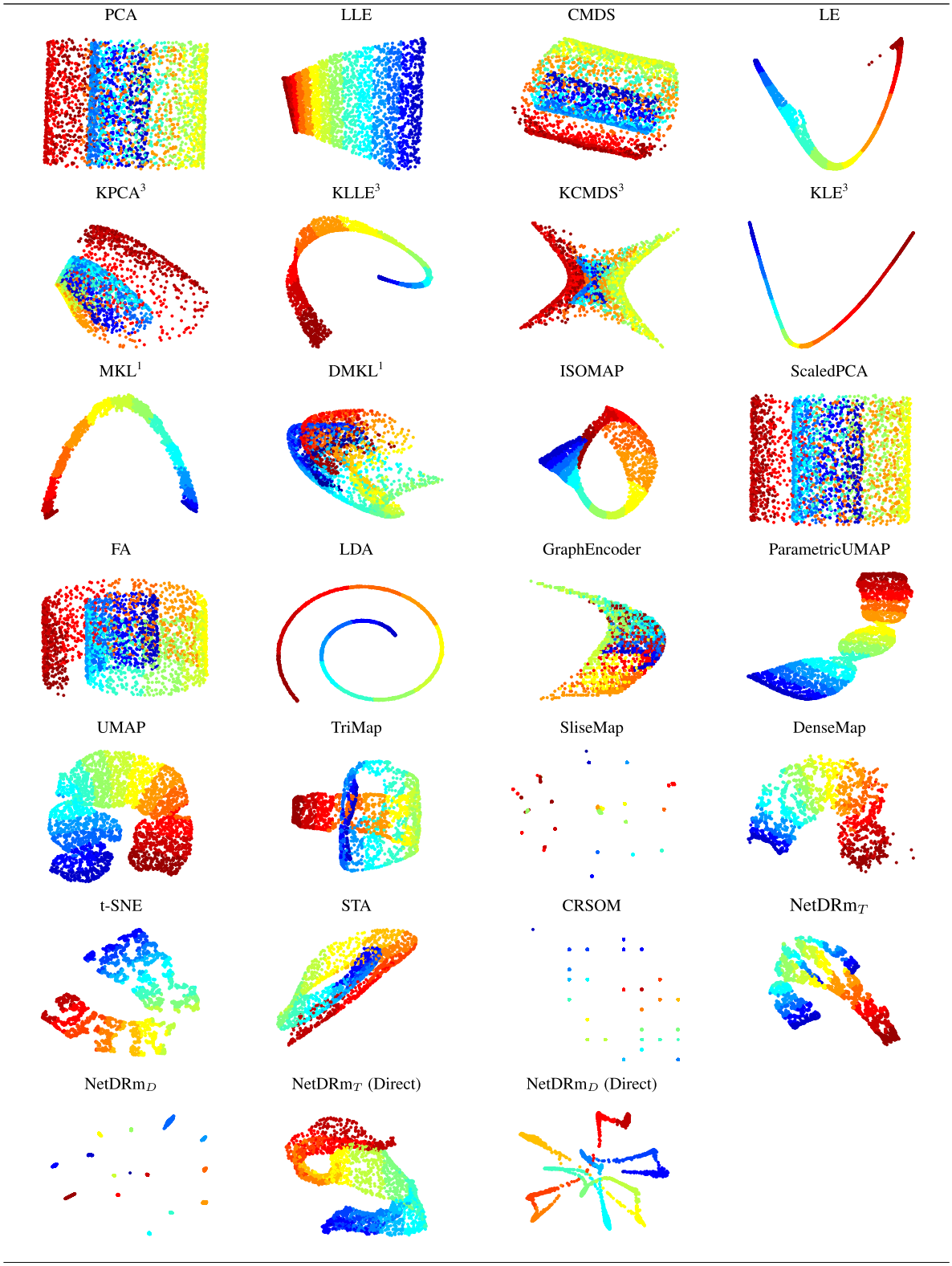

Figure 15. 2D embeddings of tested DR methods applied to the Swiss Roll dataset.

score was attained with a single hidden block: $H_b = 1$ ($L = 4$ layers).

6.3. Performance of Deep Encoders Trained with DR Spectral Methods

The performance of the proposed model has been tested by integrating four classical DR methods: PCA, CMDS, LLE, and LE. In the first phase described in Section 4.1, a neural encoder is trained to learn the embedding generated by each precursor DR method (ε_{PCA}, ε_{CMDS}, ε_{LLE}, and ε_{LE}). The COIL-20 dataset has been chosen in this section to illustrate the performance of the trained deep encoders and compare them against the original methods and their incremental versions (IKPCA, IKCMDS, IKLLE, and IKLE). The generated 2D embeddings and topological preservation measures are shown in **Figure 11** and **12**, respectively. Notice that the encoders accurately reproduce the embeddings of the original DR methods, while allowing new data to be projected into the low-dimensional space in an online manner. In addition, all their inherent features are also replicated, including the topological behavior.

6.4. Performance of Combination of DR Deep Encoders

The second stage of the proposed model synergistically combines the deep encoders trained in the previous stage using a metamodel α. As described in Section 4.2, we proposed two configurations of NetDRm: NetDRm_T (topological) and NetDRm_D (discriminative). They were applied to the Swiss Roll, Sphere, COIL-20, Fashion-MNIST, and Arrhythmia datasets introduced in Section 5.1. The performance of both configurations was evaluated through the *V*-measure, Ac, $R_{NX}6$ score explained in Section 5.2, as well as the average of these three measures, which allows a holistic view of the performance of the evaluated methods. In order to do that, the scores of the R_{NX} topological curves were normalized, with 1 being the best score and 0 the worst.

The proposed model was compared against the most relevant DR methods in the literature: LDA, FA, t-SNE, kernel methods[54] with degree-3 polynomial kernel basis functions (KPCA3, KCMDS3, KLLE3 and KLE3), MKL approaches with degree-1 polynomial kernel basis functions (MKL, DMKL), neural methods (GraphEncoder, ISOMAP, STA, CRSOM), and manifold approximation and projection methods (ScaledPCA, UMAP, TriMap, DensMap, ParametricUMAP, SliseMap). The topological and discriminant versions of NetDRm were also applied to the original datasets to perform a direct DR, that is, without going through the encoder combination process, just applying the neural learning process to the original high-dimensional data. These configurations are identified in the sequel as NetDRm (Direct).

The first dataset considered in the experiments was *COIL-20*. Figure 11 shows the 2D embeddings generated by the original PCA, CMDS, KLLE, and KLE methods. In turn, **Figure 13** shows the embeddings generated by the rest of tested DR methods.

Table 3 shows the scores obtained for *COIL-20*. In topological terms, it shows that the highest performance is obtained by t-SNE ($R_{NX} = 55.4$.), followed by the topological version of the proposed method (NetDRm_T), with $R_{NX} = 50.2$. The local behavior of the generated embedding can also be appreciated: the proposed method has higher R_{NX} values than the majority of methods for small neighborhoods (low values of perplexity *K*). This significantly differs from the rather global behavior of the other methods (high scores for large neighborhoods). The lowest performance for preserving the high-dimensional structure is attained by SliseMap ($R_{NX} = 14.9$), followed by LDA ($R_{NX} = 17.4$).

When cluster induction is considered instead, NetDRm_D gives the best performance ($V = 0.960$), followed by LDA ($V = 0.883$), as shown in Table 3 (*V*). This demonstrates the discriminative characteristics of both methods. Other methods with performances closely related to LDA are ParametricUMAP ($V = 0.882$) and DensMap ($V = 0.878$).

Regarding the Ac measure, the best method is FA with 0.461, followed by NetDRm_D for direct reduction, with a score of 0.364.

Table 5. R_{NX}, *V*-measure, and Accuracy of tested DR methods applied to the Swiss Roll dataset.

Method	R_{NX}	*V*	Ac	Av	Method	R_{NX}	*V*	Ac	Av
PCA	0.461	0.420	0.267	0.383	GraphEncoder	0.297	0.217	0.204	0.239
CMDS	0.494	0.439	0.352	0.428	MKL^1	0.287	0.789	0.230	0.435
LE	0.355	0.794	0.210	0.453	DMKL^1	0.262	0.313	0.224	0.266
LLE	0.461	0.463	0.247	0.390	FA	0.439	0.573	0.260	0.424
KPCA^3	0.440	0.442	0.250	0.377	LDA	0.279	0.963	0.210	0.484
KCMDS^3	0.411	0.372	0.287	0.357	T-SNE	**0.661**	0.652	0.220	0.511
KLE^3	0.261	0.798	0.225	0.428	STA	0.450	0.509	0.245	0.401
KLLE^3	0.340	0.782	0.286	0.469	NetDRm_T^1	0.642	0.634	0.307	**0.528**
ISOMAP	0.474	0.701	0.295	0.490	NetDRm_D^1	0.289	**0.989**	0.225	0.501
DensMap	0.555	0.641	0.295	0.497	NetDRm_T^1 (Direct)	0.606	0.627	0.291	0.508
UMAP	0.621	0.692	0.271	**0.528**	NetDRm_D^1 (Direct)	0.293	0.881	0.255	0.476
TriMap	0.569	0.552	**0.357**	0.493	SliseMap	0.126	0.945	0.261	0.444
ScaledPCA	0.464	0.421	0.320	0.402	ParametricUMAP	0.630	0.688	0.203	0.507
CRSOM	0.189	0.878	0.312	0.460					

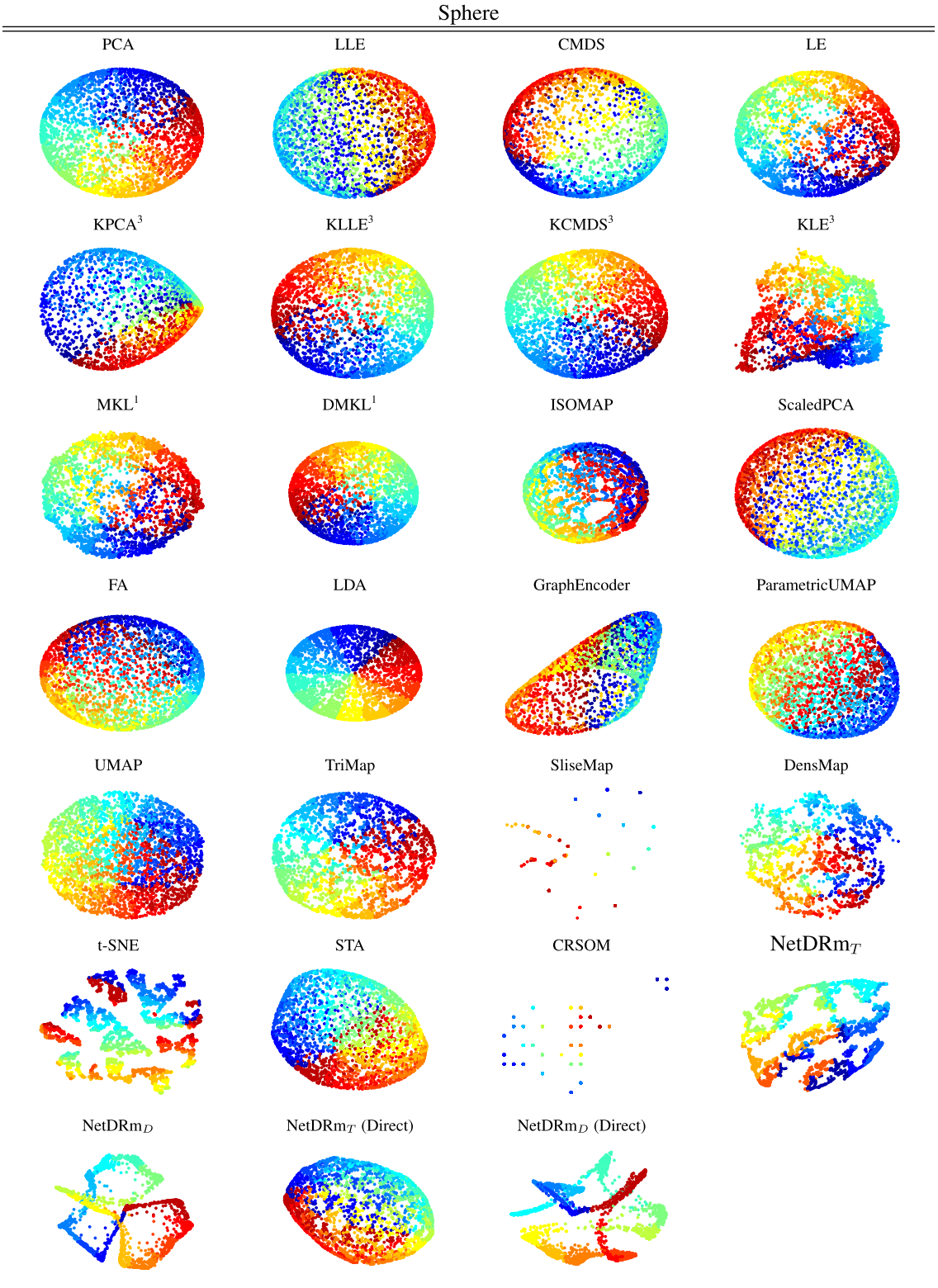

Figure 16. 2D embeddings of tested DR methods applied to the Sphere dataset.

The methods with the best average performance are t-SNE, NetDRm_T, and ParametricUMAP, with scores of 0.582, 0.557, and 0.550, respectively.

Regarding the *Fashion-MNIST* dataset, the best topological performance is achieved by t-SNE ($R_{NX} = 40.7$), followed by NetDRm_T, with a score of 38.6. In the cluster induction evaluation, the highest V score was for NetDRm_D (0.714), followed by CRSOM (0.674) and ParametricUMAP, with a score of (0.632). Regarding the Accuracy measure, NetDRm_T yields the best performance (0.525), followed by TriMap, with a score of (0.523), and LDA with 0.520. In general terms, the best average is obtained by NetDRm_T (0.498). The scores can be seen in **Table 4**. The resulting embeddings for *Fashion-MNIST* are shown in **Figure 14**.

Figure 15 shows the 2D embeddings generated by the evaluated DR methods applied to the Swiss Roll dataset. In turn, the R_{NX} scores presented in **Table 5** indicate that t-SNE yields the highest performance ($R_{NX} = 66.1$) in structural preservation of the Swiss Roll dataset. NetDRm_T is the method with the second best performance ($R_{NX} = 64.2$). NetDRm_D has a significantly better discriminative performance than the other methods, achieving the best score ($V = 0.989$). In terms of Accuracy, the best score is achieved by TriMap (0.357), followed by ScaledPCA and CRSOM, with scores of 0.320 and 0.312, respectively. The best average of the evaluated measures is achieved by the UMAP and NetDRm_T methods.

Figure 16 shows the 2D embeddings of the evaluated DR methods applied to the *Sphere* dataset. In turn, **Table 6** shows the scores of the various DR methods for such a highly compact dataset. There is not a big variation between the local and global sections of the topological curves of the different DR methods, which also present similar scores. Notwithstanding, t-SNE gives the best performance ($R_{NX} = 64.4$). FA ($R_{NX} = 13.5$), CRSOM ($R_{NX} = 18.0$), and SliseMap ($R_{NX} = 20.3$) have the lowest topological preservation. However, SliseMap ($V = 0.838$) and NetDRm_D ($V = 0.839$) have the best cluster induction. In terms of Accuracy, NetDRm_D ($\text{Ac} = 0.355$) yields the best performance. NetDRm_T represents the best balance between measures, with $\text{Av} = 0.490$.

Finally, **Figure 17** shows the 2D embeddings generated by the evaluated DR methods for the Arrhythmia tabular dataset. In turn, **Table 7** shows that NetDRm_T yields the best topological performance ($R_{NX} = 56.5$). SliseMap ($R_{NX} = 10.1$) has the lowest topological performance. Notwithstanding, SliseMap yields the best cluster induction score ($V = 0.872$), followed by NetDRm_D ($V = 0.803$). However, NetDRm_D achieves the best average performance ($\text{Av} = 0.600$), followed by SliseMap ($\text{VR}_{NX} = 0.596$). In terms of Accuracy, ISOMAP achieved the highest score (0.851).

6.4.1. Online Performance

The proposed DR model based on neural networks can be applied online, that is, new high-dimensional points can be embedded right away without retraining the system. This online feature distinguishes the proposed method from many of the DR methods proposed in the literature, which are offline. The latter means that those methods must be recomputed whenever new points must be projected.

In order to assess the online capabilities of the proposed DR model, we run experiments aimed at measuring its performance when embedding new data points, that is, points never considered during training. For the Swiss Roll and Sphere datasets, those new data points were generated in a straightforward way from the generative mathematical formulation of each manifold. For the COIL-20 and Fashion-MNIST image datasets, new synthetic images were generated by training a DCGAN network.[55] Those new images mimic the original ones, although being sufficiently different due to random rotations and noise. The new points for the Arrhythmias dataset were the ones contained in its validation subset provided in the source repository.

Table 8 shows the cluster induction (V), Accuracy (Ac), and topological preservation (R_{NX}) measures after embedding those new data points with the two proposed variations of NetDRm, as

Table 6. R_{NX}, V-measure, and Accuracy of tested DR methods applied to the Sphere dataset.

Method	R_{NX}	V	Ac	Av	Method	R_{NX}	V	Ac	Av
PCA	0.481	0.611	0.241	0.444	GraphEncoder	0.483	0.279	0.225	0.329
CMDS	0.484	0.452	0.238	0.391	MKL[1]	0.461	0.521	0.235	0.406
LE	0.492	0.531	0.221	0.415	DMKL[1]	0.483	0.589	0.230	0.434
LLE	0.482	0.361	0.243	0.362	FA	0.135	0.323	0.216	0.225
KPCA[3]	0.438	0.596	0.240	0.425	LDA	0.481	0.634	0.248	0.454
KCMDS[3]	0.488	0.597	0.238	0.441	T-SNE	**0.644**	0.482	0.290	0.472
KLE[3]	0.464	0.486	0.206	0.385	STA	0.474	0.484	0.238	0.399
KLLE[3]	0.482	0.586	0.228	0.432	NetDRm_T^1	0.618	0.514	0.339	**0.490**
ISOMAP	0.500	0.397	0.218	0.372	NetDRm_D^1	0.234	**0.839**	**0.355**	0.476
DensMap	0.537	0.490	0.250	0.426	NetDRm_T^1 (Direct)	0.521	0.439	0.331	0.430
UMAP	0.519	0.478	0.218	0.405	NetDRm_D^1 (Direct)	0.337	0.749	0.323	0.469
TriMap	0.521	0.549	0.228	0.433	SliseMap	0.203	0.838	0.308	0.450
ScaledPCA	0.479	0.360	0.223	0.354	ParametricUMAP	0.512	0.397	0.251	0.387
CRSOM	0.180	0.833	0.213	0.409					

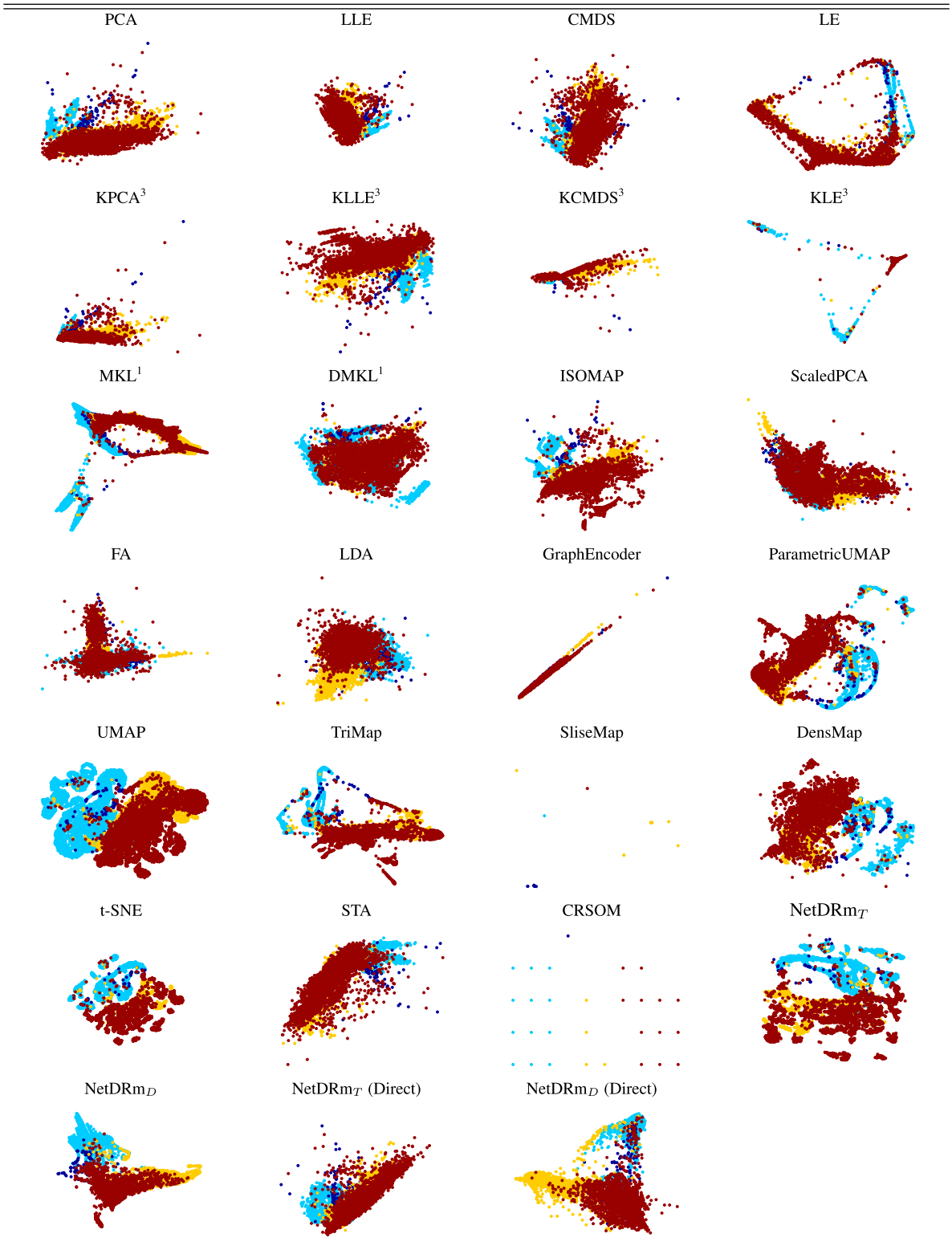

Figure 17. 2D embeddings of tested DR methods applied to the Electrocardiogram (ECG) Arrhythmia dataset.

Table 7. R_{NX}, *V*-measure, and Accuracy of tested DR methods applied to the Arrhythmia dataset.

Method	R_{NX}	*V*	Ac	Av	Method	R_{NX}	*V*	Ac	Av
PCA	0.459	0.338	0.819	0.539	GraphEncoder	0.118	0.258	0.700	0.359
CMDS	0.387	0.332	0.813	0.511	MKL[1]	0.323	0.402	0.617	0.447
LE	0.375	0.427	0.344	0.382	DMKL[1]	0.246	0.262	0.416	0.308
LLE	0.418	0.432	0.339	0.396	FA	0.430	0.323	0.501	0.306
KPCA[3]	0.398	0.425	0.252	0.358	LDA	0.156	0.672	0.745	0.524
KCMDS[3]	0.369	0.141	1	0.437	T-SNE	0.554	0.400	0.780	0.578
KLE[3]	0.374	0.245	0.356	0.325	STA	0.382	0.363	0.524	0.423
KLLE[3]	0.429	0.428	0.327	0.395	NetDRm_T^1	**0.565**	0.475	0.636	0.559
ISOMAP	0.455	0.378	**0.851**	0.561	NetDRm_D^1	0.228	3	0.770	**0.600**
DensMap	0.448	0.467	0.472	0.462	NetDRm_T^1 (Direct)	0.485	0.426	0.645	0.518
UMAP	0.443	0.428	0.256	0.376	NetDRm_D^1 (Direct)	0.217	0.764	0.771	0.584
TriMap	0.490	0.356	0.812	0.553	SliseMap	0.101	**0.872**	0.814	0.596
ScaledPCA	0.153	0.626	0.631	0.470	ParametricUMAP	0.480	0.469	0.669	0.539
CRSOM	0.188	0.852	0.362	0.467					

Table 8. R_{NX} and *V*-measure for NetDR_T, NetDR_D, and neural encoders applied to never seen data points.

Dataset	Measure	ε_{PCA}	ε_{CMDS}	ε_{LE}	ε_{LLE}	NetDRm_T	NetDRm_D
cm*Fashion-MNIST*	*V*	0.417	0.415	0.546	0.501	0.522	0.696
	R_{NX}	25.9	26.3	24.7	21.1	36.2	20.3
cm*Coil-20*	*V*	0.622	0.616	0.515	0.609	0.778	0.917
	R_{NX}	37.6	37.5	28.2	31.1	41.5	28.5
	Ac	0.226	0.288	0.296	0.235	0.276	0.323
cm*Swiss Roll*	*V*	0.411	0.416	0.758	0.421	0.653	0.932
	R_{NX}	43.6	46.2	29.3	44.1	58.7	26.3
	Ac	0.207	0.303	0.209	0.229	0.284	0.217
cm*Sphere*	*V*	0.589	0.405	0.508	0.332	0.506	0.796
	R_{NX}	45.6	45.3	46.2	45.4	57.1	22.0
	Ac	0.230	0.204	0.220	0.224	0.325	0.329
cm*Arrhythmia*	*V*	0.311	0.318	0.401	0.412	0.388	0.667
	R_{NX}	44.2	44.8	32.1	39.5	50.3	20.2
	Ac	0.792	0.787	0.324	0.311	0.615	0.752

well as with the four neural encoders on their own. Notice that the neural encoders fully replicate the behavior of the original methods, not only in terms of embeddings, but also regarding the measures. Furthermore, the experiment clearly shows the online capabilities of NetDRm, which is able to embed new data points in a fast and effective way. In addition, it also shows that the integration of those four encoders with the proposed metamodels outperforms the individual performance of every encoder on its own, thus proving the significant synergistic effect of ensemble learning in this application scope.

7. Conclusion

A new approach for online DR based on neural ensemble learning has been proposed. NetDRm is intended to integrate multiple well-known dimension reduction methods in a synergistic way. It has been designed for datasets of multidimensional points, either structured (e.g., images) or unstructured (e.g., point clouds, tabular data). NetDRm starts by training a collection of deep residual encoders that learn the embeddings induced by the precursor methods. Every encoder is then optimized by applying manifold approximation. Those optimized embeddings inherit the genuine topological preservation properties of the original methods being combined, behaving as seeds of the final embedding. Subsequently, a dense neural network (metamodel) combines the optimized neural encoders. Two variations of the integration network have been proposed, depending on whether the final aim is topological preservation or cluster induction (i.e., discrimination). For topological preservation, a shallow integration network applies unsupervised learning with a compound

cost function that aims at preserving the local topology through a probabilistic neighborhood graph and the global topology through high- and low-dimensional Euclidean distance matrices. For cluster induction, a deep integration network is trained in a supervised way with a compound cost function that minimizes intracluster differences and maximizes intercluster differences.

Extensive experiments have been conducted on widely used heterogeneous datasets (point-cloud manifolds, images, and tabular records) and the most relevant methods in the dimension reduction literature. In order to assess the overall performance of the tested DR methods in terms of topological preservation, cluster induction, as well as classification accuracy, we averaged the corresponding three evaluation measures (VR_{NX}, V, and Accuracy). The obtained results show that $NetDRm_T$ yields the best average (overall) performance for the Fashion-MNIST, Swiss Roll, and Sphere datasets. In turn, $NetDRm_D$ gives the best overall performance for Arrhythmia, whereas t-SNE is the best overall technique for COIL-20. Furthermore, $NetDRm_D$ consistently yielded the best results in terms of cluster induction (V measure) for all datasets except for Arrythmia. In addition, NetDRm is an online method that does not require recomputing the full embedding when new high-dimensional data points must be projected. Once it has been trained, new data points can be processed right away, taking advantage of the good properties of the generated embedding. Finally, experiments also show that the integration of the neural encoders with the proposed metamodels outperforms the individual performance of every encoder on its own, thus proving the significant synergistic effect of ensemble learning in this scope.

Immediate work will consist of exploring new neural topologies for the proposed encoders and integration networks. We also aim to extend the experimental validation to other types of structured and unstructured datasets. In particular, the ability to deal with heterogeneous tabular datasets opens new research lines in the field of tabular data processing with deep neural networks, which may have a strong impact in several exciting fields, such as medical diagnosis. Currently, the adaptation of neural networks to tabular data for inference or data generation tasks remains highly challenging. The proposed heterogeneous dimension reduction technique can thus conform a backbone for new tabular data processing models.

Acknowledgements

CESMAG University. The Spanish Government partly supported this research through Project TED2021-130081B-C21, Project PDC2022-133383-I00, and Project PID2019-105789RB-I00.

Conflict of Interest

The authors declare no conflict of interest.

Data Availability Statement

The data that support the findings of this study are openly available in Coil-20 at https://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php, reference number 0. These data were derived from the following resources available in the public domain: [Coil20], https://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php[Coil20]; [Arrhythmia], https://www.kaggle.com/datasets/sadmansakib7/ecg-arrhythmia-classification-dataset.[Arrhythmia].

Keywords

cluster inductions, dimensionality reductions, ensemble learning, manifold approximations, online processing, topological preservations, unsupervised deep networks

Received: March 6, 2024
Revised: July 17, 2024
Published online: August 5, 2024

[1] M. Kumbhkar, P. Shukla, Y. Singh, R. A. Sangia, D. Dhabliya, in *2023 IEEE Int. Conf. on Integrated Circuits and Communication Systems (ICICACS)*. IEEE, Piscataway, NJ **2023**, pp. 1–7.
[2] A. R. Javed, W. Ahmed, S. Pandya, P. K. R. Maddikunta, M. Alazab, T. R. Gadekallu, *Electronics* **2023**, *12*, 1020.
[3] I. M. Enholm, E. Papagiannidis, P. Mikalef, J. Krogstie, *Inf. Syst. Front.* **2022**, *24*, 1709.
[4] E. Mosqueira-Rey, E. Hernández-Pereira, D. Alonso-Ros, J. Bobes-Bascarán, Á. Fernández-Leal, *Artif. Intell. Rev.* **2023**, *56*, 3005.
[5] W. Jia, M. Sun, J. Lian, S. Hou, *Complex Intell. Syst.* **2022**, *8*, 2663.
[6] J. A. Lee, M. Verleysen, in *Nonlinear Dimensionality Reduction*, Springer Science & Business Media, Berlin, Germany, **2007**.
[7] L. Meneghetti, N. Demo, G. Rozza, *Appl. Intell.* **2023**, *53*, 22818.
[8] L. Wang, Z. Wang, H. Qu, S. Liu, *Appl. Soft Comput.* **2018**, *66*, 1.
[9] D. Denisko, M. M. Hoffman, *Proc. Natl. Acad. Sci.* **2018**, *115*, 1690.
[10] G. Hinton, R. Salakhutdinov, *Science* **2006**, *313*, 504.
[11] T. Sainburg, L. McInnes, T. Q. Gentner, *Neural Comput.* **2021**, *33*, 2881.
[12] B. Xu, G. Zhang, *bioRxiv* **2023**, 11.
[13] Z. Luo, C. Xu, Z. Zhang, W. Jin, *Sci. Rep.* **2021**, *11*, 20028.
[14] M. Alswaitti, K. Siddique, S. Jiang, W. Alomoush, A. Alrosan, *Symmetry* **2022**, *14*, 1282.
[15] L. Jiang, X. Fang, W. Sun, N. Han, S. Teng, *Signal Proc.* **2023**, *204*, 108817.
[16] T. Guo, K. Yu, M. Aloqaily, S. Wan, *Future Gener. Comput. Syst.* **2022**, *128*, 381.
[17] E. Eskandarnia, H. M. Al-Ammal, R. Ksantini, *Sustainable Cities Soc.* **2022**, *78*, 103618.
[18] L. Wu, L. Noels, *Comput. Methods Appl. Mech. Eng.* **2022**, *390*, 114476.
[19] J. A. Lee, E. Renard, G. Bernard, P. Dupont, M. Verleysen, *Neurocomputing* **2013**, *112*, 92.
[20] A. Rosenberg, J. Hirschberg, in *Proc. of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, Prague, Czech Republic, **2007**, pp. 410–420.
[21] T. Fawcett, *Pattern Recognit. Lett.* **2006**, *27*, 861.
[22] H. Hotelling, *J. Educ. Psychol.* **1933**, *24*, 417.
[23] I. Borg, in *Modern Multidimensional Scaling: Theory and Applications*, Springer, Berlin, Germany, **2005**.
[24] M. Belkin, P. Niyogi, *Neural Comput.* **2003**, *15*, 1373.
[25] S. T. Roweis, L. K. Saul, *Science* **2000**, *290*, 2323.
[26] A. J. Izenman, in *Modern Multivariate Statistical Techniques*, Springer, Berlin, Germany, **2013**, pp. 237–280.
[27] J. B. Tenenbaum, V. de Silva, J. C. Langford, *Science* **2000**, *290*, 2319.
[28] H. Qu, L. Li, Z. Li, J. Zheng, *Expert Syst. Appl.* **2021**, *180*, 115055.
[29] M. Gönen, E. Alpaydn, *J. Mach. Learn. Res.* **2011**, *12*, 2211.

[30] E. V. Strobl, S. Visweswaran, in *2013 12th Int. Conf. on Machine Learning and Applications*, Miami, FA, **2013**, *1*, pp. 414–417.
[31] F. Tian, B. Gao, Q. Cui, E. Chen, T.-Y. Liu, in *Proc. of the AAAI Conf. on Artificial Intelligence*, Palo Alto, CA, **2014**, *28*.
[32] C. Spearman, *Am. J. Psychol.* **1904**, *15*, 201.
[33] A. Björklund, J. Mäkelä, K. Puolamäki, *Mach. Learn.* **2023**, *112*, 1.
[34] E. Amid, M. K. Warmuth (Preprint) arXiv:1910.00204, v1, Submitted: Oct. 2019.
[35] A. Narayan, B. Berger, H. Cho, Density-preserving data visualization unveils dynamic patterns of single-cell transcriptomic variability, *bioRxiv* **2020**, *05*.
[36] B. Ghojogh, A. Ghodsi, F. Karray, M. Crowley (Preprint) arXiv:2109.02508, v1, Submitted: Aug. 2021.
[37] L. Van der Maaten, G. Hinton, *J. Mach. Learn. Res.* **2008**, *9*, 85.
[38] P. Hartono, P. Hollensen, T. Trappenberg, *IEEE Trans. Neural Networks Learn. Syst.* **2014**, *26*, 2323.
[39] P. Hartono, *IEEE Access* **2020**, *8*, 105301.
[40] D. Huang, F. Jiang, K. Li, G. Tong, G. Zhou, *Manage. Sci.* **2022**, *68*, 1678.
[41] R. Girdhar, A. El-Nouby, Z. Liu, M. Singh, K. V. Alwala, A. Joulin, I. Misra, in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition,* Vancouver, BC, **2023**, pp. 15180–15190.
[42] L. Molina, L. Belanche, A. Nebot, in *2002 IEEE Int. Conf. on Data Mining*, Maebashi City, Japan, **2002**, pp. 306–313.
[43] P. Joshi, P. Kulkarni, *Int. J. Data Min. Knowl. Manage. Process* **2012**, *2*, 43.
[44] Z. Sun, Y. Wang, P. Liu, Y. Luo, *Mater. Des.* **2022**, *220*, 110885.
[45] M. Rovira, K. Engvall, C. Duwig, *Chem. Eng. J.* **2022**, *438*, 135250.
[46] R. K. Sanodiya, J. Mathew, *Image Vision Comput.* **2019**, *90*, 103802.
[47] Q. Wang, Y. Ma, K. Zhao, Y. Tian, *Ann. Data Sci.* **2020**, *9*, 187.
[48] P. J. Huber, *Ann. Math. Stat.* **1964**, *35*, 73.
[49] D. P. Kingma, J. Ba, presented at *the 3rd Int. Conf. for Learning Representations*, San Diego, **2015**.
[50] S. A. Nene, S. K. Nayar, H. Murase, Columbia Object Image Library (COIL-20) **1996**, 62.
[51] H. Xiao, K. Rasul, R. Vollgraf (Preprint) arXiv:1708.07747, v1, Submitted: Aug. 2017.
[52] A. Shetty, N. AV, *Earth Sci. Inf.* **2023**, *16*, 25.
[53] J. A. Lee, M. Verleysen, in *2014 IEEE Symp. on Computational Intelligence and Data Mining (CIDM)*, IEEE, Piscataway, NJ **2014**, pp. 163–170.
[54] C. Carmeli, E. De Vito, A. Toigo, *Anal. Appl.* **2006**, *4*, 377.
[55] A. Radford, L. Metz, S. Chintala, **2015**, *1511*, 06434.