Information Fusion 55 (2020) 127–149

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

Multi-view diffusion maps

Ofir Lindenbaum [a,*], Arie Yeredor [a], Moshe Salhov [b], Amir Averbuch [b]

[a] School of Electrical Engineering, Tel Aviv University, Israel
[b] School of Computer Science, Tel Aviv University, Israel

ARTICLE INFO

Keywords:
Dimensionality reduction
Manifold learning
Diffusion maps
Multi-view

ABSTRACT

In this paper, we address the challenging task of achieving multi-view dimensionality reduction. The goal is to effectively use the availability of multiple views for extracting a coherent low-dimensional representation of the data. The proposed method exploits the intrinsic relation within each view, as well as the mutual relations between views. The multi-view dimensionality reduction is achieved by defining a cross-view model in which an implied random walk process is restrained to hop between objects in the different views. The method is robust to scaling and insensitive to small structural changes in the data. We define new diffusion distances and analyze the spectra of the proposed kernel. We show that the proposed framework is useful for various machine learning applications such as clustering, classification, and manifold learning. Finally, by fusing multi-sensor seismic data we present a method for automatic identification of seismic events.

1. Introduction

high-dimensional big data are becoming ubiquitous in a growing variety of fields, and pose new challenges in their analysis. Extracted features are useful in analyzing these datasets. However, some prior knowledge or modeling is required in order to identify the essential features. Unsupervised dimensionality reduction methods, on the other hand, aim to find low-dimensional representations based on the *intrinsic geometry* of the analyzed data. A "good" dimensionality reduction methodology reduces the complexity of the data, while preserving its coherency, thereby facilitating data analysis tasks (such as clustering, classification, manifold learning and more) in the reduced space. Many methods such as Principal Component Analysis (PCA) [1], Multidimensional Scaling (MDS) [2], Local Linear Embedding [3], Laplacian Eigenmaps (LE) [4], Diffusion Maps (DM) [5], p-Laplacian Regularization [6], T-distributed Stochastic Neighbor Embedding (t-SNE) [7], Uniform manifold approximation [8] and more have been proposed to achieve robust dimensionality reduction. low-dimensional representations have been shown to be useful in various applications such as face recognition based on LE [9], Non-linear independent component analysis using DM [10], Musical Key extraction employing DM [11], and many more.

Frameworks such as [1–8] do not consider the possibility to exploit more than one view representing the same process. Such multiple views can be obtained, for example, when the same underlying process is observed using several different modalities, or measured with different instrumentations. An additional view can provide meaningful insights regarding the dynamical process behind the data, whenever the data is indeed generated or governed by such a latent process.

This study is devoted to the development of a framework for obtaining a low-dimensional parametrization from multiple measurements organized as "views". Our approach essentially relies on quantifying of the speed of a random walk restricted to "hop" between views. The analysis of such a random walk provides a natural representation of the data using joint l organization of the multiple measurements.

The problem of learning from multiple views has been studied in several domains. Some prominent statistical approaches, for addressing this problem are Bilinear Models [12], Partial Least Squares [13] and Canonical Correlation Analysis (CCA) [14]. These methods are powerful for learning the relations among the different views, but are restricted to linear transformation for each view. Some methods such as [15–20] extend the CCA to nonlinear transformations by introducing a kernel, and demonstrate the advantages of fusing multiple views for clustering and classification. A Hessian multiset canonical correlations is presented in [21], the authors propose to overcome limitations of the Laplacian by defining local Hessians on the samples to capture structures of multiple views. Markov based methods such as [22–24] use diffusion distances for classification, clustering or retrieval tasks. The "agreement" (also called "consensus") between different views is used in [25] to extract the geometric information from all views. A sparsity-based learning algorithm for cross-view dimensionality reduction is proposed in [26]. A neural network which extracts maximally correlated representations is studied in [27]. A review comprehensive article [28] presents recent progress and challenges in this domain.

* Corresponding author.
E-mail address: ofirlin@gmail.com (O. Lindenbaum).

https://doi.org/10.1016/j.inffus.2019.08.005
Received 1 July 2018; Received in revised form 27 May 2019; Accepted 17 August 2019
Available online 19 August 2019

In this work we extend the concepts established in [19,29] to devise a diffusion-based learning framework for fusing multiple views, by seeking the implied underlying low-dimensional structure in the ambient space. Our contributions can be summarized as follows.

1. We present a natural generalization of the DM method [5] for handling multiple views. The generalization is attained by combining the intrinsic relations within each view with the mutual relations between views, so as to construct a multi-view kernel matrix. The proposed kernel defines a cross-view diffusion process, and related diffusion distances, which impose a structured random walk between the various views. In addition, we show that the spectral decomposition of the proposed kernel can be used to find an embedding of the data that offers improved exploitation of the information encapsulated in the multiple views.
2. For the coupled views setting, we analyze theoretical properties of the proposed method. The associated parametrization is justified by being recast as a minimizer of a kernel-based objective function. Then, we relate the spectrum of the proposed kernel to the spectrum of a single-view kernel which is based on naïve concatenation of the views. Finally, under some simplification, we find the infinitesimal generator of the proposed kernel.
3. For practical use, we present an automated method for setting the kernel bandwidth parameter and a proposed procedure for recursively augmenting the representation with each new data point.
4. We demonstrate the applicability of our multi-view method to classification, clustering and manifold learning. Furthermore, by fusing data from multiple seismic sensors we demonstrate an automatic extraction of latent seismic parameters using real-world data.

The paper is structured as follows. Some essential background is provided in Section 2. In Section 3 we formulate the multi-view dimensionality reduction problem and discuss prior work. Then, in Section 4 we present our proposed multi-view DM method and discuss some of its basic properties. Section 5 studies additional theoretical properties of the proposed kernel for the particular (more simple) case of a coupled setting (where only two views are available). Section 6 presents the experimental results. Potential applications are described in Section 7 and concluding remarks appear in Section 8 and Appendix (respectively).

2. Background

2.1. General dimensionality reduction

Consider a high-dimensional dataset $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_M\} \in \mathbb{R}^{D\times M}$, $\boldsymbol{x}_i \in \mathbb{R}^D, i = 1, 2, \ldots, M$. The goal is to find a low-dimensional representation $\boldsymbol{Z} = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_M\} \in \mathbb{R}^{r\times M}, \boldsymbol{z}_i \in \mathbb{R}^r, i = 1, 2, \ldots, M$, such that $r \ll D$, while latent "inner relations" (if any) among the multidimensional data points are preserved as closely as possible, in some sense. This problem setup is based on the assumption that the data is represented (viewed) in a single vector space (single view).

2.2. Diffusion maps (DM)

DM [5] is a dimensionality reduction method which aims at extracting the intrinsic geometry of the data. The DM framework is highly effective when the data is densely sampled from some low-dimensional manifold, so that its "inner relations" are the local connectivities (or proximities) on the manifold. Given a high-dimensional dataset $\boldsymbol{X}$, the DM framework consists of the following steps:

1. A kernel function $\mathcal{K} : \boldsymbol{X} \times \boldsymbol{X} \to \mathbb{R}$ is chosen, so as to construct a matrix $\boldsymbol{K} \in \mathbb{R}^{M\times M}$ with elements $K_{ij} \triangleq \mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j)$, satisfying the following properties: (i) Symmetry: $\boldsymbol{K} = \boldsymbol{K}^{\mathrm{T}}$; (ii) Positive semi-defeniteness: $\boldsymbol{K} \succeq \boldsymbol{0}$, namely $\forall \boldsymbol{v} \in \mathbb{R}^M : \ \boldsymbol{v}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{v} \geq 0$; and (iii) Non-negativity: $\boldsymbol{K} \geq \boldsymbol{0}$, namely $K_{i,j} \geq 0 \ \forall i, \ j \in [1, \ M]$. These properties guarantee that the matrix $\boldsymbol{K}$ has real-valued eigenvectors and non-negative eigenvalues. A *Gaussian kernel* is a common example, in which $\mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) \triangleq \exp\left\{-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\sigma_x^2}\right\}$, where $\|\cdot\|$ denotes the L_2 norm and σ_x^2 is a user-selected width (scale) parameter.
2. By normalizing the rows of $\boldsymbol{K}$, the matrix
$$\boldsymbol{P}^x \triangleq \boldsymbol{D}^{-1}\boldsymbol{K} \in \mathbb{R}^{M\times M} \tag{1}$$
is obtained, where $\boldsymbol{D} \in \mathbb{R}^{M\times M}$ is a diagonal matrix with $D_{i,i} = \sum_j K_{i,j}$. $\boldsymbol{P}^x$ can be interpreted as the transition probabilities of a (fictitious) Markov chain on $\boldsymbol{X}$, such that $\left[(\boldsymbol{P}^x)^t\right]_{i,j} \triangleq p_t(\boldsymbol{x}_i, \boldsymbol{x}_j)$ (where t is an integer power) describes the implied probability of transition from point $\boldsymbol{x}_i$ to point $\boldsymbol{x}_j$ in t steps.
3. Spectral decomposition is applied to $\boldsymbol{P}^x$, obtaining a set of M eigenvalues $\{\lambda_m\}$ (in descending order) and associated normalized eigenvectors $\{\boldsymbol{\psi}_m\}$ satisfying $\boldsymbol{P}^x\boldsymbol{\psi}_m = \lambda_m\boldsymbol{\psi}_m$, $m = 0, \ldots, M-1$, or $\boldsymbol{P}^x\boldsymbol{\Psi} = \boldsymbol{\Psi}\boldsymbol{\Lambda}$, where $\boldsymbol{\Psi}$ and $\boldsymbol{\Lambda}$ are (resp.) the eigenvectors and diagonal eigenvalues matrices.
4. A new representation is defined for the dataset $\boldsymbol{X}$, representing each $\boldsymbol{x}_i$ by the ith row of $(\boldsymbol{P}^x)^t\boldsymbol{\Psi} = \boldsymbol{\Psi}\boldsymbol{\Lambda}^t$, namely:
$$\boldsymbol{\Psi}_t(\boldsymbol{x}_i) : \ \boldsymbol{x}_i \mapsto \left[\lambda_1^t\psi_1[i], \ldots, \lambda_{M-1}^t\psi_{M-1}[i]\right]^{\mathrm{T}} \in \mathbb{R}^{M-1}, \ \ i \in [1, M] \tag{2}$$
where t is a selected number of steps and $\psi_m[i]$ denotes the ith element of $\boldsymbol{\psi}_m$. Note that the trivial eigenvector $\boldsymbol{\psi}_0 = \boldsymbol{1}$ (with corresponding eigenvalue $\lambda_0 = 1$) was omitted from the representation as it does not carry information about the data. The main idea behind this representation is that the Euclidean distance between two data points in the new representation equals a weighted L_2 distance between the conditional probability vectors $p_t(\boldsymbol{x}_i, :)$ and $p_t(\boldsymbol{x}_j, :)$, $i, j \in [1, M]$ (the ith and jth rows of $(\boldsymbol{P}^x)^t$). This weighted Euclidean distance is referred to as the *diffusion distance*, denoted
$$\begin{aligned} \mathcal{D}_t^2(\boldsymbol{x}_i, \boldsymbol{x}_j) &\triangleq \|\boldsymbol{\Psi}_t(\boldsymbol{x}_i) - \boldsymbol{\Psi}_t(\boldsymbol{x}_j)\|^2 \\ &= \sum_{m=1}^{M-1} \lambda_m^{2t}(\psi_m[i] - \psi_m[j])^2 = \|p_t(\boldsymbol{x}_i, :) - p_t(\boldsymbol{x}_j, :)\|_{W^{-1}}^2, \end{aligned} \tag{3}$$
where $\boldsymbol{W} = \boldsymbol{D}/\mathrm{Trace}\{\boldsymbol{D}\}$ and where $\|z^T\|_Q \triangleq \boldsymbol{z}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{z}$ denotes a $\boldsymbol{Q}$-weighted Euclidean norm (the last equality is shown in [5]).
5. A desired accuracy level $\delta \geq 0$ is chosen for the diffusion distance defined by Eq. (3), such that $r(\delta, t) = \max\{\ell : |\lambda_\ell|^t > \delta|\lambda_1|^t\}$. The new (truncated) $r(\delta, t)$-dimensional mapping, which leads to the desired representation $\boldsymbol{Z}$, is then defined as
$$\boldsymbol{\Psi}_t^\delta(\boldsymbol{x}_i) : \ \boldsymbol{x}_i \mapsto \left[\lambda_1^t\psi_1[i], \lambda_2^t\psi_2[i]. \ldots, \lambda_r^t\psi_r[i]\right]^{\mathrm{T}} \triangleq \boldsymbol{z}_i \in \mathbb{R}^{r(\delta,t)}. \tag{4}$$

This dimensionality reduction approach was found to be useful in various applications in diverse fields. However, as previously noted, it is limited to a single-view representation.

3. Multi-view dimensionality reduction - Problem formulation and prior work

Assume now that we are given multiple sets (views) of observations $\boldsymbol{X}^\ell, \ell = 1, \ldots, L$. Each view is a high-dimensional dataset $\boldsymbol{X}^\ell = \{\boldsymbol{x}_1^\ell, \boldsymbol{x}_2^\ell, \ldots, \boldsymbol{x}_M^\ell\} \in \mathbb{R}^{D_\ell\times M}$, where D_ℓ is the dimension of each feature space. Note that a bijective correspondence between views is assumed. These views can be a result of different measurement devices or different types of features extracted from raw data. For each view $\ell = 1, \ldots, L$, we seek a lower-dimensional representation that preserves the "inner relations" between multidimensional data points within each view $\boldsymbol{X}^\ell$, as well as among all views $\{\boldsymbol{X}^1, \ldots, \boldsymbol{X}^L\}$. Our goal is to find L representations $\boldsymbol{\Phi}_\ell(\boldsymbol{X}^1, \ldots, \boldsymbol{X}^L) : \mathbb{R}^{D_\ell} \to \mathbb{R}^r$, such that $r \ll D_\ell, \ \ell = 1, \ldots, L$.

Before turning to present our proposed approach, we describe existing alternative approaches for incorporating multiple views. In particular, we detail here the methods which we would later use as benchmarks

for comparison to our proposed framework. All of these methods begin by computing the kernel matrix $\boldsymbol{K}^{\ell}$ for each (ℓth) view ($\ell = 1, \ldots, L$), and differ in the way of combining these kernels for generating the DM.

3.1. Kernel product DM (KP)

A naïve generalization of DM [5] may be computed using an element-wise kernel product, namely by constructing $\boldsymbol{K}^{\circ} \triangleq \boldsymbol{K}^1 \circ \boldsymbol{K}^2 \circ \ldots \circ \boldsymbol{K}^L \in \mathbb{R}^{M\times M}$, where $\circ$ denotes Hadamard's (element-wise) matrix product, such that $K^{\circ}_{i,j} \triangleq K^1_{i,j} \cdot K^2_{i,j} \cdot \ldots \cdot K^L_{ij}$, followed by row-normalization: $\boldsymbol{P} = (\boldsymbol{D}^{\circ})^{-1}\boldsymbol{K}^{\circ} \in \mathbb{R}^{M\times M}$, where $\boldsymbol{D}^{\circ}$ is diagonal, with $D^{\circ}_{i,i} \triangleq \sum_j K^{\circ}_{i,j}$.

Note that in the special case of using Gaussian kernels with equal width parameters σ, the resulting matrix $\boldsymbol{K}^{\circ}$ is equal to the matrix $\boldsymbol{K}^{w}$ constructed from the concatenated observations vector

$$\boldsymbol{w}_i = \left[\left(\boldsymbol{x}_i^1\right)^{\mathrm{T}}, \ldots, \left(\boldsymbol{x}_i^L\right)^{\mathrm{T}}\right]^{\mathrm{T}} \tag{5}$$

such that

$$K^{w}_{i,j} = \exp\left\{-\frac{\left\|\boldsymbol{w}_i - \boldsymbol{w}_j\right\|^2}{2\sigma^2}\right\}. \tag{6}$$

3.2. Kernel sum DM (KS)

An average diffusion process was used in [23], where the *sum kernel* is defined as

$$\boldsymbol{K}^{+} \triangleq \sum_{\ell=1}^{L} \boldsymbol{K}^{\ell}, \tag{7}$$

followed by row-normalization: $\boldsymbol{P}^{+} = \left(\boldsymbol{D}^{+}\right)^{-1}\boldsymbol{K}^{+}$, where $D^{+}_{i,i} = \sum_j K^{+}_{i,j}$. The implied random walk sums up (and normalizes) the step probabilities from each view.

3.3. Kernel canonical correlation analysis (KCCA)

The frameworks [15,16] extend the well-known Canonical Correlation Analysis (CCA) by applying a kernel function prior to the application of CCA. Kernels $\boldsymbol{K}^1$ and $\boldsymbol{K}^2$ are constructed for each view, and the canonical vectors $\boldsymbol{\nu}_1$ and $\boldsymbol{\nu}_2$ are obtained by solving the following generalized eigenvalue problem

$$\begin{bmatrix} \boldsymbol{0}_{M\times M} & \boldsymbol{K}^1 \cdot \boldsymbol{K}^2 \\ \boldsymbol{K}^2 \cdot \boldsymbol{K}^1 & \boldsymbol{0}_{M\times M} \end{bmatrix} \begin{pmatrix} \boldsymbol{\nu}_1 \\ \boldsymbol{\nu}_2 \end{pmatrix} = \rho \cdot \begin{bmatrix} (\boldsymbol{K}^1 + \gamma \boldsymbol{I})^2 & \boldsymbol{0}_{M\times M} \\ \boldsymbol{0}_{M\times M} & (\boldsymbol{K}^2 + \gamma \boldsymbol{I})^2 \end{bmatrix} \begin{pmatrix} \boldsymbol{\nu}_1 \\ \boldsymbol{\nu}_2 \end{pmatrix}, \tag{8}$$

where $\gamma \boldsymbol{I}$ are regularization terms added to prevent overfitting and thus improve generalization of the method. Usually the Incomplete Cholesky Decomposition (ICD) [15,16,30] is used to reduce the run time required for solving (8). For clustering tasks, K-means clustering is applied to the set of generalized eigenvectors.

3.4. Spectral clustering with two views

The approach in [19] generalizes the traditional normalized graph Laplacian for two views. Kernels $\boldsymbol{K}^1$ and $\boldsymbol{K}^2$ are computed in each view and multiplied, yielding $\boldsymbol{W} = \boldsymbol{K}^1 \cdot \boldsymbol{K}^2$, from which

$$\boldsymbol{A} \triangleq \begin{bmatrix} \boldsymbol{0}_{M\times M} & \boldsymbol{W} \\ \boldsymbol{W}^T & \boldsymbol{0}_{M\times M} \end{bmatrix} \in \mathbb{R}^{2M\times 2M} \tag{9}$$

is obtained. Symmetric normalization is applied by using the diagonal matrix $\bar{\boldsymbol{D}}$ with diagonal elements $\bar{D}_{i,i} = \sum_j W_{i,j}$, such that the normalized fused kernel is defined as

$$\bar{\boldsymbol{A}} = \bar{\boldsymbol{D}}^{-0.5} \cdot \boldsymbol{A} \cdot \bar{\boldsymbol{D}}^{-0.5}. \tag{10}$$

Assuming that the data can be clustered into N_C clusters, and denoting the $2M$ eigenvectors of $\bar{\boldsymbol{A}}$ as $\boldsymbol{\phi}_i, i = 1, \ldots, 2M$ (in descending order of the corresponding eigenvalues), the mapping

$$\boldsymbol{\Phi}[i] \triangleq \frac{1}{s[i]}\left[\phi_1[i], \ldots, \phi_{N_C}[i]\right]^{\mathrm{T}} \in \mathbb{R}^{N_C} \tag{11}$$

(where $s[i] = \sum_{j=1}^{N_C}(\phi_j^2[i])$) is useful for applying K-means clustering. The work in [19] is focused on spectral clustering, however, a similar version of the kernel from Eq. (10) is suitable for manifold learning, as we demonstrate in the sequel. In Section 6 this approach is referred to as de Sa's.

As We show in the next section, our proposed approach essentially uses a stochastic matrix version of this kernel, and extends the construction to multiple views. We shall show that our stochastic matrix version is useful, as it provides various theoretical justifications for the implied multi-view diffusion process.

4. Our proposed approach for multi-view diffusion maps

Here we propose our generalization of the DM framework for handling a multi-view scenario. This is done by imposing an implied (fictitious) random walk model using the local connectivities between data points within all views. Our way to generalize the DM framework is by restraining the random walker to "hop" between different views in each step. The construction requires to choose a set of L symmetrical positive semi-definite, non-negative kernels, one for each view$\mathcal{K}^l : \boldsymbol{X}^l \times \boldsymbol{X}^l \to \mathbb{R}$, $l = 1, \ldots, L$. We use the Gaussian kernel function, so that the (i, j)th element of each $\boldsymbol{K}^l \in \mathbb{R}^{M\times M}$ is given by

$$K^l_{i,j} = \exp\left\{-\frac{||\boldsymbol{x}_i^l - \boldsymbol{x}_j^l||^2}{2\sigma_l^2}\right\}, \quad i,j = 1, \ldots, M, \quad l = 1, \ldots, L, \tag{12}$$

where $\left\{\sigma_l^2\right\}_{l=1}^{L}$ are a set of selected parameters. The decaying property of the Gaussian is useful, as it removes the influence of large Euclidean distances. The *multi-view kernel* is formed by constructing the following matrix

$$\hat{\boldsymbol{K}} = \begin{bmatrix} \boldsymbol{0}_{M\times M} & \boldsymbol{K}^1\boldsymbol{K}^2 & \boldsymbol{K}^1\boldsymbol{K}^3 & \ldots & \boldsymbol{K}^1\boldsymbol{K}^L \\ \boldsymbol{K}^2\boldsymbol{K}^1 & \boldsymbol{0}_{M\times M} & \boldsymbol{K}^2\boldsymbol{K}^3 & \ldots & \boldsymbol{K}^2\boldsymbol{K}^L \\ \boldsymbol{K}^3\boldsymbol{K}^1 & \boldsymbol{K}^3\boldsymbol{K}^2 & \boldsymbol{0}_{M\times M} & \ldots & \boldsymbol{K}^3\boldsymbol{K}^L \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ \boldsymbol{K}^L\boldsymbol{K}^1 & \boldsymbol{K}^L\boldsymbol{K}^2 & \boldsymbol{K}^L\boldsymbol{K}^3 & \ldots & \boldsymbol{0}_{M\times M}. \end{bmatrix} \in \mathbb{R}^{LM\times LM}. \tag{13}$$

Finally, using the diagonal row-normalization matrix $\hat{\boldsymbol{D}} \in \mathbb{R}^{LM\times LM}$ with $\hat{D}_{i,i} = \sum_j \hat{K}_{i,j}$, the normalized row-stochastic matrix is defined as

$$\hat{\boldsymbol{P}} = \hat{\boldsymbol{D}}^{-1}\hat{\boldsymbol{K}} \in \mathbb{R}^{LM\times LM}. \tag{14}$$

We refer to the (l, m)th block of $\hat{\boldsymbol{P}}$ as the square $M\times M$ matrix starting at

$[1 + (l-1)M, 1 + (m-1)M]$, $l, m = 1, \ldots, L$. Thus, the (i, j)th element of the (l, m)th block describes a (fictitious) probability of transition from $\boldsymbol{x}_i^l$ to $\boldsymbol{x}_j^m$. This construction takes into account all possibilities to "hop" between views, under the constraint that staying in the same view (namely, a transition from $\boldsymbol{x}_i^l$ to $\boldsymbol{x}_j^l$) is forbidden.

4.1. probabilistic interpretation of $\hat{\mathbf{P}}^t$

Subsequent to our proposed construction (Eqs. (12)–(14)), each element of the power-t matrix $\hat{\boldsymbol{P}}^t$,

$$\left[\hat{\boldsymbol{P}}^t\right]_{i+(l-1)M, j+(m-1)M} \triangleq \hat{p}_t(\boldsymbol{x}_i^l, \boldsymbol{x}_j^m) \tag{15}$$

denotes the probability of transition from $\boldsymbol{x}_i^l$ to $\boldsymbol{x}_j^m$ in t time-steps. Note that, as mentioned above, due to the block-off-diagonal structure of $\hat{\boldsymbol{P}}$, which forbids a transition into the same view within a single time-step, we have $\hat{p}_1(\boldsymbol{x}_i^l, \boldsymbol{x}_j^l) = 0$, $l = 1, \ldots, L$, although for $t > 1$ $\hat{p}_t(\boldsymbol{x}_i^l, \boldsymbol{x}_j^l)$ may be non-zero.

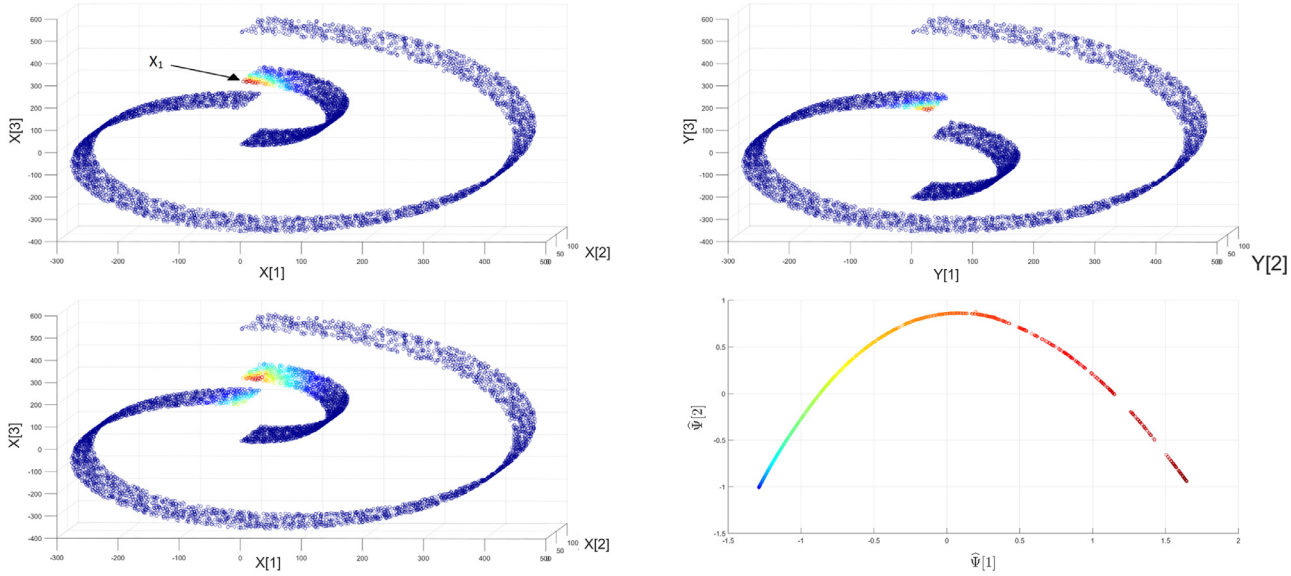

Fig. 1. Top-left: A non-smooth Swiss Roll sampled from View-I ($\boldsymbol{X}^1$), colored by the single-view probability of transition ($t = 1$) from $\boldsymbol{x}_1$ to $\boldsymbol{x}_:$. Top-right: A second Swiss Roll, sampled from View-II ($\boldsymbol{X}^2$), colored by the single-view probability of transition ($t = 1$) from $\boldsymbol{y}_1$ to $\boldsymbol{y}_:$. Bottom-left: the first Swiss Roll, colored by the multi-view probabilities of transition ($t = 1$) from $\boldsymbol{x}_i$ to $\boldsymbol{y}_:$. In the top-left figure, we highlight $\boldsymbol{x}_1$ using an arrow. Bottom-right: a low-dimensional representation extracted based on the multi-view transition matrix $\widehat{\boldsymbol{P}}$ (Eq. (14)).

4.1.1. *Smoothing effect* $t = 1$

For simplicity let us examine the term $\widehat{p}_t(\boldsymbol{x}_i^l, \boldsymbol{x}_j^m), l \neq m$ for $t = 1$. The transition probability for $t = 1$ is

$$\widehat{p}_1(\boldsymbol{x}_i^l, \boldsymbol{x}_j^m) = \frac{\sum_s K_{i,s}^l K_{s,j}^m}{\widehat{D}_{i,i}}.$$

This probability takes into consideration all the various connectivities of node $\boldsymbol{x}_i^l$ to node $\boldsymbol{x}_s^l$ and the connectivities of the corresponding node $\boldsymbol{x}_s^m$ to the destination node $\boldsymbol{x}_j^m$. The proposed multi-view approach has a "smoothing effect" in terms of the transition probabilities, meaning that the probability of transitioning from $\boldsymbol{x}_i^l$ to $\boldsymbol{x}_j^m$ could be positive even if $K_{i,j}^l = 0$ and $K_{i,j}^m = 0$: Assume that a non-empty subset $S = \{s_1, \ldots, s_F\} \subseteq [1, M]$ exists, such that $K_{i,s_f}^l > 0$ and $K_{s_f,j}^m > 0, f = 1, \ldots, F$. Then by definition of the multi-view probability we have $\widehat{p}_1(\boldsymbol{x}_i^l, \boldsymbol{x}_j^m) > 0$. Note that although with the Gaussian kernel all elements of $\boldsymbol{K}^l$ are positive, the smoothing effect can still be significant when despite negligibly low values in $K_{i,j}^l$ and in $K_{i,j}^m$, $\widehat{p}_1(\boldsymbol{x}_i^l, \boldsymbol{x}_j^m)$ can become relatively high.

Fig. 1 illustrates the multi-view transition probabilities compared to a single-view approach using two ($L = 2$) deformed "Swiss Roll" manifolds. In each view, there is no probability of transition from one side of its observed "gap" to the other side. And yet, the multi-view (one-step) transition probability is non-zero for points at both sides of the gaps. This smoothing effect occurs because the gaps are located at a different position (near different points) on each view, thus allowing the multi-view kernel to smooth out the nonlinear gaps.

4.1.2. *Increasing the diffusion step* t

Under the stochastic Markov model assumption, raising $\widehat{\boldsymbol{P}}$ to a higher power (by increasing the diffusion step t) spreads the probability mass function along its rows based on the connectivities in all views. As described in [5], this probability spread reduces the influence of eigenvectors associated with high-indexed (smaller) eigenvalues on the diffusion distance (Eq. (16)). This implies that the eigenvectors corresponding to low-indexed (large) eigenvalues have a low-frequency content, whereas the eigenvectors corresponding to the high-indexed (small) eigenvalues describe the oscillatory behavior of the data [5]. In Fig. 2, we present the eigenvalues of the matrix $\widehat{\boldsymbol{P}}^t$ with different values of t. For this experiment we generated $L = 3$ Swiss Rolls with $M = 1200$ data points

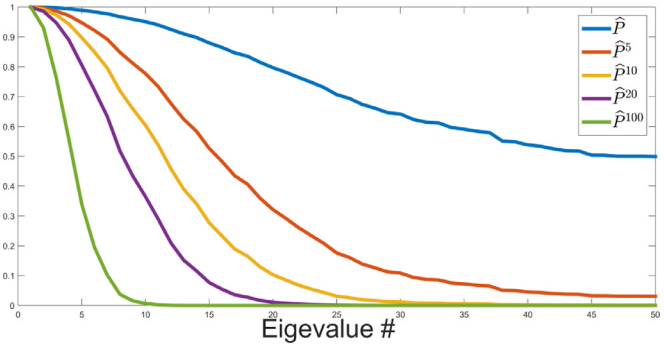

Fig. 2. The decay of the eigenvalues for increasing powers of the matrix $\widehat{\boldsymbol{P}}$.

each. It is evident that the numerical rank of $\widehat{\boldsymbol{P}}^t$ decreases for higher values of t.

4.2. *Multi-view diffusion distance*

In a variety of real-world data types, the Euclidean distance between observed data points (vectors) does not provide sufficient information about the intrinsic relations between these vectors, and is highly sensitive to non-unitary transformations. Common tasks such as classification, clustering or system identification often require a measure for the intrinsic connectivity between data points, which is only locally expressed by the Euclidean distance in the high-dimensional ambient space. The multi-view diffusion kernel (defined in Section 4) describes all the small local connections between data points. The row stochastic matrix $\widehat{\boldsymbol{P}}^t$ (Eq. (14)) accounts for all possible transitions between data points in t time steps while hopping between the views. For a fixed value $t > 0$, two data points are *intrinsically similar* if the conditional distributions $\widehat{p}_t(\boldsymbol{x}_i, :) = [\widehat{\boldsymbol{P}}^t]_{i,:}$ and $\widehat{p}_t(\boldsymbol{x}_j, :) = [\widehat{\boldsymbol{P}}^t]_{j,:}$ are similar. This type of similarity measure indicates that the points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are similarly connected (in the sense of similar probabilities of transitions) to several mutual points. Thus, they are connected by some geometrical path. In many cases, a small Euclidean distance can be misleading due to the fact that two data points can be "close" without having any geodesic path that connects them. Comparing the transition probabilities is more

robust, as it takes into consideration all of the local connectivities between the compared points. Therefore, even if two points seem far from one another in the Euclidean sense, they may still share many common neighbors and thus be "close" in the sense of having a small diffusion distance.

Based on this observation, by expanding the single-view construction given in [5], we define the weighted inner view diffusion distances for the first view as

$$D_t^2(\boldsymbol{x}_i^1,\boldsymbol{x}_j^1) \triangleq \sum_{k=1}^{L\cdot M}\frac{1}{\tilde{\phi}_0(k)}\left(\left[\widehat{\boldsymbol{P}}^t\right]_{i,k}-\left[\widehat{\boldsymbol{P}}^t\right]_{j,k}\right)^2=\left\|(\boldsymbol{e}_i-\boldsymbol{e}_j)^{\mathrm{T}}\widehat{\boldsymbol{P}}^t\right\|^2_{\widehat{\boldsymbol{D}}^{-1}}, \tag{16}$$

where $1\le i,j\le M$, $\boldsymbol{e}_i$ is the ith column of an $L\cdot M\times L\cdot M$ identity matrix, $\tilde{\boldsymbol{\phi}}_0$ is the (non-normalized) first left eigenvector of $\widehat{\boldsymbol{P}}$, whose kth element is $\tilde{\phi}_0(k)=\widehat{D}_{k,k}$. A similarly weighted norm is defined for the lth view

$$D_t^2(\boldsymbol{x}_i^l,\boldsymbol{x}_j^l) \triangleq \sum_{k=1}^{L\cdot M}\frac{1}{\tilde{\phi}_0(k)}\left(\left[\widehat{\boldsymbol{P}}^t\right]_{i+\tilde{l},k}-\left[\widehat{\boldsymbol{P}}^t\right]_{j+\tilde{l},k}\right)^2=\left\|(\boldsymbol{e}_{\tilde{l}+i}-\boldsymbol{e}_{\tilde{l}+j})^{\mathrm{T}}\widehat{\boldsymbol{P}}^t\right\|^2_{\widehat{\boldsymbol{D}}^{-1}}, \tag{17}$$

where $\tilde{l}=(l-1)\cdot M$. The main advantage of these distances (Eqs. (16) and (17)) is that they can be expressed in terms of the eigenvalues and eigenvectors of the matrix $\widehat{\boldsymbol{P}}$. This insight allows us to use a representation (defined in Section 4.3 below) where the induced Euclidean distance is proportional to the diffusion distances defined in Eqs. (16) and (17). Indeed, let $\widehat{\boldsymbol{P}}=\boldsymbol{\Psi}\boldsymbol{\Lambda}\boldsymbol{\Phi}^{\mathrm{T}}$ denote the eigenvalues decomposition of $\widehat{\boldsymbol{P}}$, where $\boldsymbol{\Psi}=\left[\psi_0,\ldots,\psi_{L\cdot M-1}\right]$ and $\boldsymbol{\Phi}=\left[\phi_0,\ldots,\phi_{L\cdot M-1}\right]$ denote the matrices of (normalized) right- and left-eigenvectors (resp.), and $\boldsymbol{\Lambda}\triangleq\mathrm{Diag}(\lambda_0,\ldots,\lambda_{L\cdot M-1})$ denotes the diagonal matrix of respective eigenvalues. Then

Theorem 1. *The inner view diffusion distance defined by Eqs. (16) and (17) can also be expresses as*

$$D_t^2(\boldsymbol{x}_i^l,\boldsymbol{x}_j^l)=\sum_{k=1}^{L\cdot M-1}\lambda_k^{2t}\left(\psi_k\left[i+\tilde{l}\right]-\psi_k\left[j+\tilde{l}\right]\right)^2, i,j=1,\ldots,M, \tag{18}$$

where $\tilde{l}=(l-1)\cdot M$.

Proof. Using the eigenvalues decomposition of $\widehat{\boldsymbol{P}}$ we have

$$\widehat{\boldsymbol{P}}^t\widehat{\boldsymbol{D}}^{-1}\left(\widehat{\boldsymbol{P}}^t\right)^{\mathrm{T}}=\boldsymbol{\Psi}\boldsymbol{\Lambda}^t\boldsymbol{\Phi}^{\mathrm{T}}\widehat{\boldsymbol{D}}^{-1}\boldsymbol{\Phi}\boldsymbol{\Lambda}^t\widehat{\boldsymbol{\Psi}}^{\mathrm{T}}. \tag{19}$$

Define the symmetric matrix $\widehat{\boldsymbol{P}}_s\triangleq\widehat{\boldsymbol{D}}^{-1/2}\widehat{\boldsymbol{K}}\widehat{\boldsymbol{D}}^{-1/2}=\widehat{\boldsymbol{D}}^{1/2}\widehat{\boldsymbol{P}}\widehat{\boldsymbol{D}}^{-1/2}$, and note that this matrix is algebraically similar to $\widehat{\boldsymbol{P}}$. Therefore, both matrices share the same set of eigenvalues $\boldsymbol{\Lambda}$. Additionally, let $\boldsymbol{\Pi}$ denote the (left- and right-) orthonormal eigenvectors matrix of $\widehat{\boldsymbol{P}}_s$, namely $\widehat{\boldsymbol{P}}_s=\boldsymbol{\Pi}\boldsymbol{\Lambda}\boldsymbol{\Pi}^{\mathrm{T}}$. The left- and right-eigenvectors matrices of $\widehat{\boldsymbol{P}}=\widehat{\boldsymbol{D}}^{-1/2}\widehat{\boldsymbol{P}}_s\widehat{\boldsymbol{D}}^{1/2}$ are then easily identified as $\boldsymbol{\Phi}=\widehat{\boldsymbol{D}}^{1/2}\boldsymbol{\Pi}$ and $\boldsymbol{\Psi}=\widehat{\boldsymbol{D}}^{-1/2}\boldsymbol{\Psi}$ (resp.), so that $\boldsymbol{\Phi}^{\mathrm{T}}\widehat{\boldsymbol{D}}^{-1}\boldsymbol{\Phi}=\boldsymbol{\Pi}^{\mathrm{T}}\boldsymbol{\Pi}=\boldsymbol{I}$. Therefore,

$$\begin{aligned}D_t^2(\boldsymbol{x}_i^l,\boldsymbol{x}_j^l)&=\left\|(\boldsymbol{e}_{\tilde{l}+i}-\boldsymbol{e}_{\tilde{l}+j})^{\mathrm{T}}\widehat{\boldsymbol{P}}^t\right\|^2_{\widehat{\boldsymbol{D}}^{-1}}\\&=(\boldsymbol{e}_{\tilde{l}+i}-\boldsymbol{e}_{\tilde{l}+j})^{\mathrm{T}}\widehat{\boldsymbol{P}}^t\widehat{\boldsymbol{D}}^{-1}\left(\widehat{\boldsymbol{P}}^t\right)^{\mathrm{T}}(\boldsymbol{e}_{\tilde{l}+i}-\boldsymbol{e}_{\tilde{l}+j})\\&=(\boldsymbol{e}_{\tilde{l}+i}-\boldsymbol{e}_{\tilde{l}+j})^{\mathrm{T}}\boldsymbol{\Psi}\boldsymbol{\Lambda}^{2t}\boldsymbol{\Psi}^{\mathrm{T}}(\boldsymbol{e}_{\tilde{l}+i}-\boldsymbol{e}_{\tilde{l}+j})\\&=\sum_{k=1}^{L\cdot M-1}\lambda_k^{2t}\left(\psi_k\left[i+\tilde{l}\right]-\psi_k\left[j+\tilde{l}\right]\right)^2.\end{aligned} \tag{20}$$

The term with $k=0$ can be excluded from the sum since $\boldsymbol{\psi_0}=\mathbf{1}$ (an all-ones vector) is always the first right-eigenvector (with eigenvalue 1) for any stochastic matrix, so the term corresponding to $k=0$ in the sum would vanish anyway. □

4.3. Multi-view data parametrization

Tasks such as classification, clustering or regression in a high-dimensional feature space are considered to be computationally expensive. In addition, the performance in such tasks is highly dependent on the distance measure used. As explained in Section 4.2, distance measures in the original ambient space are often meaningless in many real life situations. Interpreting Theorem 1 in terms of Euclidean distance enables us to define mappings for every view $\boldsymbol{X}^l, l=1,\ldots,L$, using the right eigenvectors of $\widehat{\boldsymbol{P}}$ (Eq. (14)) weighted by λ_i^t. The representation for instances in $\boldsymbol{X}^l$ is given by

$$\widehat{\boldsymbol{\Psi}}_t(\boldsymbol{x}_i^l):\boldsymbol{x}_i^l\longmapsto\left[\lambda_1^t\psi_1[i+\tilde{l}],\ldots,\lambda_{M-1}^t\psi_{M-1}[i+\tilde{l}]\right]^T\in\mathbb{R}^{M-1}, \tag{21}$$

where $\tilde{l}=(l-1)\cdot M$. These L mappings capture the intrinsic geometry of the views as well as the mutual relations between them. As shown in [31], the set of eigenvalues λ_m has a decaying property such that $1=|\lambda_0|\ge|\lambda_1|\ge\cdots\ge|\lambda_{M-1}|$. Exploiting the decaying property enables us to represent data up to a dimension r where $r\ll D_1,\ldots,D_l$. The dimension $r\equiv r(\delta)$ is determined by approximating the diffusion distance (Eq. (18)) up to a desired accuracy δ (we elaborate on this issue in Section 5.4). The reduced dimension version of $\widehat{\boldsymbol{\Psi}}_t(\boldsymbol{X})$ is denoted by $\widehat{\boldsymbol{\Psi}}_t^r(\boldsymbol{X})$.

Using the inner view diffusion distances defined in Eqs. (16) and (17), we define a multi-view diffusion distance as the sum of inner views distances,

$$\mathcal{D}_t^{(MV)^2}(i,j)\triangleq\sum_{l=1}^{L}\left|\widehat{\boldsymbol{\Psi}}_t^r(\boldsymbol{x}_i^l)-\widehat{\boldsymbol{\Psi}}_t^r(\boldsymbol{x}_j^l)\right|^2. \tag{22}$$

This distance is the induced Euclidean distance in a space constructed from the concatenation of all low-dimensional multi-view mappings

$$\widehat{\boldsymbol{\Psi}}_t(\boldsymbol{X})=\left[\widehat{\boldsymbol{\Psi}}_t^r(\boldsymbol{X}^1);\widehat{\boldsymbol{\Psi}}_t^r(\boldsymbol{X}^2);\ldots;\widehat{\boldsymbol{\Psi}}_t^r(\boldsymbol{X}^L)\right]\in\mathbb{R}^{L\cdot r\times M}. \tag{23}$$

This mapping is used in Section 7.1 for the experimental evaluation of clustering.

4.4. Multi-view kernel bandwidth

When constructing the Gaussian kernels $\boldsymbol{K}^l, l=1,\ldots,L$, in Eq. (12), the values of the scale (width) parameter σ_l^2 have to be set. Setting these values to be too small may result in very small local neighborhoods that are unable to capture the local structures around the data points. Conversely, setting the values to be too large may result in a fully connected graph that may generate a too coarse description of the data. Several approaches have been proposed in the literature for determining the kernel width (scale):

In [32], a max-min measure is suggested such that the scale becomes

$$\sigma_l^2=C\cdot\max_j\left[\min_{i,i\neq j}(||\boldsymbol{x}_i^l-\boldsymbol{x}_j^l||^2)\right]. \tag{24}$$

Based on empirical results, the authors in [32] suggest to set C within the range [1,1.5]. The specific choice of C should be sufficient to capture all local connectivities. This single-view approach could be relaxed in the context of our multi-view scenario. The multi-view kernel $\widehat{\boldsymbol{K}}$ (Eq. (13)) consists of products of single-view kernel matrices $\boldsymbol{K}^l, l=1,\ldots,L$ (Eq. (12)). The diagonal values of each kernel matrix $\boldsymbol{K}^l$ are all 1's, therefore, in order to have connectivity in all rows of $\widehat{\boldsymbol{K}}$ a connectivity in only one of the views is sufficient. By connectivity, we mean that there is at least one nonzero value in the affinity kernel. This insight suggests that a smaller value for the parameter C could be used in the multi view setting. In [33], the authors provide a probabilistic interpretation of the choice of C, and present a grid-search algorithm for optimizing the choice of C such that each point in the dataset is connected to at least one other point.

Another scheme, proposed in [34], aims to find a range of values for σ_l. The idea is to compute the kernel $\boldsymbol{K}^l$ (Eq. (12)) for various values of σ and search for the range of values where the Gaussian bell shape is more pronounced. To find such range of valid values for σ, first a logarithmic function is applied to the sum of all elements of the kernel matrix, next the range is identified as the maximal range where the logarithmic plot is linear. We expand this idea for a multi-view scenario based on the following algorithm:

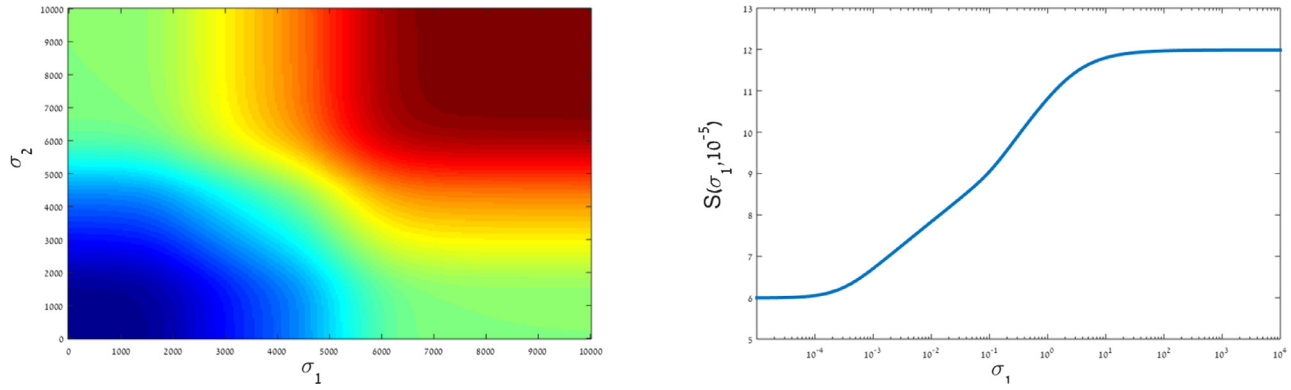

Fig. 3. Left: an example of the two dimensional function $S(\sigma_l,\ \sigma_m)$. Right: a slice at the first row ($\sigma_2 = 10^{-5}$). The asymptotes are clearly visible in both figures. Algorithm 1 exploits the multi-view to set a small scale parameter for both views.

Note that the two dimensional function $S^{lm}(\sigma_l,\ \sigma_m)$ consists of two asymptotes, $S^{lm}(\sigma_l,\sigma_m) \overset{\sigma_l,\sigma_m\to 0}{\longrightarrow} \log(N)$, and $S^{lm}(\sigma_l,\sigma_m) \overset{\sigma_l,\sigma_m\to\infty}{\longrightarrow} \log(N^3) = 3\log(N)$, since for $\sigma_l,\ \sigma_m \to 0$, both $\boldsymbol{K}^l$ and $\boldsymbol{K}^m$ approach the Identity matrix, and for $\sigma_l,\ \sigma_m \to \infty$, both $\boldsymbol{K}^l$ and $\boldsymbol{K}^m$ approach all-ones matrices. An example of the plot $S^{lm}(\sigma_l,\ \sigma_m)$ for two views ($L = 2$) is presented in Fig. 3. The range of each σ_l should reflect the asymptotic behaviour, and may be determined empirically by choosing $\min(\sigma_l) \ll median\{||\boldsymbol{x}_i^l - \boldsymbol{x}_j^l||\}$ and $\max(\sigma_l) \gg \text{median}\{||\boldsymbol{x}_i^l - \boldsymbol{x}_j^l||\}$. In practice the range $[10^{-5}, 10^5]$ is sufficient.

4.5. Computational complexity

Let us now consider the computational complexity required for extracting $\boldsymbol{\Phi}_l(\boldsymbol{X}^1,\ldots,\boldsymbol{X}^L) \in \mathbb{R}^{r\times M}$ from $\boldsymbol{X}^l = \{\boldsymbol{x}_1^l, \boldsymbol{x}_2^l, \boldsymbol{x}_3^l, \ldots, \boldsymbol{x}_M^l\} \in \mathbb{R}^{D_l\times M}$. The computational complexity of computing L kernels is $\mathcal{O}(\sum_l M^2 D_l)$. Computing the proposed normalized kernel $\widehat{\boldsymbol{P}}$ (Eq. 14) adds a complexity of $\mathcal{O}(\frac{L(L-1)}{4}M^3)$. The final step requires the spectral decomposition of $\widehat{\boldsymbol{P}}$, thus it is the most computationally expensive step and adds a complexity of $\mathcal{O}(L^3M^3)$. However, due to the low rank, sparse nature of $\widehat{\boldsymbol{P}}$, this spectral decomposition could be approximated using random projection methods such as in [35–37]. Using random projection the complexity of the spectral decomposition is improved to $\mathcal{O}(L^2M^2\log r)$. For the optional step proposed in Algorithm 1, if n_0 values are chosen for σ, the additional complexity cost is of $\mathcal{O}(MLn_0)$.

Algorithm 1 Multi-view kernel bandwidth selection.

Input: Multiple sets of observations (views) $\boldsymbol{X}^l, l = 1,\ldots,L$.
Output: Scale parameters $\{\sigma_1,\ldots,\sigma_L\}$.

1: Compute Gaussian kernels $\boldsymbol{K}^l(\sigma_l), l = 1,\ldots,L$ for several values of $\sigma_l \in [10^{-5}, 10^5]$.
2: Compute for all pairs $l \neq m$: $S^{lm}(\sigma_l,\sigma_m) = \sum_i\sum_j K^{lm}_{i,j}(\sigma_l,\sigma_m)$, where $\boldsymbol{K}^{lm}(\sigma_l,\sigma_m) = \boldsymbol{K}^l(\sigma_l)\cdot\boldsymbol{K}^m(\sigma_m)$.
3: **for** $l = 1 : L$ **do**
4: Find the minimal value for σ_l such that $\boldsymbol{S}^{lm}(\sigma_l,\sigma_m)$ is linear for all $m \neq l$.
5: **end for**

5. Coupled views $L = 2$

In this section we provide analytical results for the special case of a coupled data set (i.e $L = 2$). Some of the results could be expanded to a larger number of views but not in a straightforward manner. To simplify the notation in the rest of this section we denote $\boldsymbol{X} \triangleq \boldsymbol{X}^1$ and $\boldsymbol{Y} \triangleq \boldsymbol{X}^2$.

5.1. Coupled mapping

The mappings provided by our approach (Eq. (21)) are justified by the relations given by Eq. (18). In this section we provide some intuition on the relation between the mappings of $\boldsymbol{X}$ and $\boldsymbol{Y}$. We focus on the analysis of a 1-dimensional mapping for each view. Let $\boldsymbol{\rho}^x \triangleq \boldsymbol{\rho}(\boldsymbol{X}) = [\rho(\boldsymbol{x}_1),\rho(\boldsymbol{x}_2),\ldots,\rho(\boldsymbol{x}_M)]^{\mathrm{T}}$ and $\boldsymbol{\rho}^y \triangleq \boldsymbol{\rho}(\boldsymbol{Y}) = [\rho(\boldsymbol{y}_1),\rho(\boldsymbol{y}_2),\ldots,\rho(\boldsymbol{y}_M)]^{\mathrm{T}}$ denote such 1-dimensional mappings (one for each view) and define $\boldsymbol{K}^z \triangleq \boldsymbol{K}^x\cdot\boldsymbol{K}^y$, where $\boldsymbol{K}^x$, $\boldsymbol{K}^y$ are computed from $\boldsymbol{X}$ and $\boldsymbol{Y}$ (resp.) based on Eq. (12). The following theorem characterizes the desirable 1-dimensional mappings $\boldsymbol{\rho}^x$ and $\boldsymbol{\rho}^y$.

Theorem 2. *A 1-dimensional zero-mean representation which minimizes the following objective function*

$$\min_{\boldsymbol{\rho}^x,\boldsymbol{\rho}^y}\sum_{i,j}(\rho(\boldsymbol{x}_i)-\rho(\boldsymbol{y}_j))^2K^z_{i,j}$$
$$\text{s.t. } \|\boldsymbol{\rho}^x\|^2+\|\boldsymbol{\rho}^y\|^2=1,\ \sum_i\left(\rho(\boldsymbol{x}_i)+\rho(\boldsymbol{y}_i)\right)=0, \tag{25}$$

is obtained by setting $\widehat{\boldsymbol{\rho}} \triangleq [\boldsymbol{\rho}^{x\mathrm{T}}\ \boldsymbol{\rho}^{y\mathrm{T}}]^{\mathrm{T}} = \boldsymbol{\psi}_1$, where $\boldsymbol{\psi}_1$ is the first non-trivial normalized eigenvector of the graph Laplacian $\widehat{\boldsymbol{D}} - \widehat{\boldsymbol{K}}$. The scaling constraint avoids arbitrary scaling of the representation, while the zero-mean constraint (implying orthogonality of $\widehat{\boldsymbol{\rho}}$ to the constant all-ones vector $\boldsymbol{1}$) eliminates any presence of a trivial (constant) component in the representations.

Note that $\boldsymbol{K}^z$ is an affinity measure of points based on $\boldsymbol{K}^x\cdot\boldsymbol{K}^y$. This means that a small value of $K^z_{i,j}$ indicates weak connectivity between data points i and j (possibly through an intermediate point $\ell \in [1, M]$), implying that the distance between $\rho(\boldsymbol{x}_i)$ and $\rho(\boldsymbol{y}_j)$ can be large. Conversely, if $K^z_{i,j}$ is large, indicating strong connectivity (possibly through an intermediate point) between points i and j, the distance between $\rho(\boldsymbol{x}_i)$ and $\rho(\boldsymbol{y}_j)$ should be small when aiming to minimize the objective function.

Proof. By expanding the objective function in Eq. (25) we get

$$\begin{aligned}
&\textstyle\sum_{i,j}(\rho(\boldsymbol{x}_i)-\rho(\boldsymbol{y}_j))^2K^z_{i,j} = \sum_{i,j}\rho^2(\boldsymbol{x}_i)K^z_{i,j}+\sum_{i,j}\rho^2(\boldsymbol{y}_j)K^z_{i,j}\\
&\textstyle-2\sum_{i,j}\rho(\boldsymbol{x}_i)\rho(\boldsymbol{y}_j)K^z_{i,j} = \sum_i\rho^2(\boldsymbol{x}_i)\sum_j K^z_{i,j}+\sum_j\rho^2(\boldsymbol{y}_j)\sum_i K^z_{i,j}\\
&\textstyle-\sum_{i,j}\rho(\boldsymbol{x}_i)\rho(\boldsymbol{y}_j)K^z_{i,j}-\sum_{i,j}\rho(\boldsymbol{x}_j)\rho(\boldsymbol{y}_i)K^z_{j,i} = \sum_i\rho^2(\boldsymbol{x}_i)D^{\text{rows}}_{i,i}\\
&\textstyle+\sum_j\rho^2(\boldsymbol{y}_j)D^{\text{cols}}_{j,j}-\sum_{i,j}\rho(\boldsymbol{x}_i)\rho(\boldsymbol{y}_j)K^z_{i,j}-\sum_{j,i}\rho(\boldsymbol{x}_j)\rho(\boldsymbol{y}_i)K^z_{j,i}\\
&=\begin{bmatrix}\boldsymbol{\rho}^{x\mathrm{T}} & \boldsymbol{\rho}^{y\mathrm{T}}\end{bmatrix}\left[\begin{bmatrix}\boldsymbol{D}^{\text{rows}} & \boldsymbol{0}_{M\times M}\\ \boldsymbol{0}_{M\times M} & \boldsymbol{D}^{\text{cols}}\end{bmatrix}-\begin{bmatrix}\boldsymbol{0}_{M\times M} & \boldsymbol{K}^z\\ (\boldsymbol{K}^z)^{\mathrm{T}} & \boldsymbol{0}_{M\times M}\end{bmatrix}\right]\begin{bmatrix}\boldsymbol{\rho}^x\\ \boldsymbol{\rho}^y\end{bmatrix},
\end{aligned} \tag{26}$$

where $D^{\text{rows}}_{i,i} = \sum_{j=1}^M K^z_{i,j}$ and $D^{\text{cols}}_{j,j} = \sum_{i=1}^M K^z_{i,j}$ are diagonal matrices.

The same minimization problem (Eq. (25)) can therefore be rewritten as

$$\min_{\hat{\rho}} \hat{\rho}^{\mathrm{T}}(\hat{\boldsymbol{D}}-\hat{\boldsymbol{K}})\hat{\rho} \quad \text{s.t.} \quad \|\hat{\rho}\|=1,\ \hat{\rho}^{\mathrm{T}}\mathbf{1}=0. \tag{27}$$

Without the orthogonality constraint, this minimization problem could be solved by finding the minimal eigenvalue of $(\hat{\boldsymbol{D}}-\hat{\boldsymbol{K}})\hat{\rho}^{T}=\bar{\lambda}\hat{\rho}^{T}$. This eigenproblem has a trivial solution which is the all-ones eigenvector $\boldsymbol{\psi}_0=\mathbf{1}$ with $\bar{\lambda}=0$. However, to satisfy the orthogonality constraint, we must use a different eigenvector (which would naturally be orthogonal to $\boldsymbol{\psi}_0$ due to the symmetry of the matrix), leading to the second-smallest (smallest non-zero) eigenvalue λ_1 with its corresponding eigenvector $\boldsymbol{\psi}_1$. □

5.2. Spectral decomposition

In this section, we show how to efficiently compute the spectral decomposition of $\hat{\boldsymbol{P}}$ (Eq. (14)) when only two view exist ($L=2$). As already mentioned above, the matrix $\hat{\boldsymbol{P}}$ is algebraically similar (conjugation) to the symmetric matrix $\hat{\boldsymbol{P}}_s \triangleq \hat{\boldsymbol{D}}^{1/2}\hat{\boldsymbol{P}}\hat{\boldsymbol{D}}^{-1/2}=\hat{\boldsymbol{D}}^{-1/2}\hat{\boldsymbol{K}}\hat{\boldsymbol{D}}^{-1/2}$. Therefore, both $\hat{\boldsymbol{P}}$ and $\hat{\boldsymbol{P}}_s$ share the same set of eigenvalues. Due to symmetry of the matrix $\hat{\boldsymbol{P}}_s$, it has a set of $2M$ real eigenvalues $\{\lambda_i\}_{i=0}^{2M-1}\in\mathbb{R}$ and corresponding real orthogonal eigenvectors $\{\boldsymbol{\pi}_m\}_{m=0}^{2M-1}\in\mathbb{R}^{2M}$, thus, $\hat{\boldsymbol{P}}_s=\boldsymbol{\Pi}\boldsymbol{\Lambda}\boldsymbol{\Pi}^{T}$. By denoting $\boldsymbol{\Psi}=\hat{\boldsymbol{D}}^{-1/2}\boldsymbol{\Pi}$ and $\boldsymbol{\Phi}=\hat{\boldsymbol{D}}^{1/2}\boldsymbol{\Pi}$, we conclude that the set $\{\boldsymbol{\psi}_m,\boldsymbol{\phi}_m\}_{m=0}^{2M-1}\in\mathbb{R}^{2M}$ are the right and the left eigenvectors of $\hat{\boldsymbol{P}}=\boldsymbol{\Psi}\boldsymbol{\Lambda}\boldsymbol{\Phi}^{T}$, respectively, satisfying ${\boldsymbol{\psi}_i}^{T}\boldsymbol{\phi}_j=\delta_{i,j}$ (Kronecker's delta function). In the sequel, we use the symmetric matrix $\hat{\boldsymbol{P}}_s$ to simplify the analysis.

To avoid the spectral decomposition of a $2M\times 2M$ matrix $\hat{\boldsymbol{P}}_s$, the spectral decomposition of $\hat{\boldsymbol{P}}_s$ can be computed using the Singular Value Decomposition (SVD) of the matrix $\bar{\boldsymbol{K}}^{z}=(\boldsymbol{D}^{\mathrm{rows}})^{-1/2}\boldsymbol{K}^{z}(\boldsymbol{D}^{\mathrm{cols}})^{-1/2}$ of size $M\times M$ where $D^{\mathrm{rows}}_{i,i}=\sum_{j=1}^{M}K^{z}_{i,j}$ and $D^{\mathrm{cols}}_{j,j}=\sum_{i=1}^{M}K^{z}_{i,j}$ are diagonal matrices. Theorem 3 enables us to form the eigenvectors of $\hat{\boldsymbol{P}}$ as a concatenation of the singular vectors of $\boldsymbol{K}^{z}=\boldsymbol{K}^{x}\cdot\boldsymbol{K}^{y}$.

Theorem 3. *By using the left and right singular vectors of $\boldsymbol{K}^{z}=\boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{U}^{T}$, the eigenvectors and the eigenvalues of $\hat{\boldsymbol{K}}$ are given explicitly by*

$$\boldsymbol{\Pi}=\frac{1}{\sqrt{2}}\begin{bmatrix}\boldsymbol{V} & \boldsymbol{V}\\ \boldsymbol{U} & -\boldsymbol{U}\end{bmatrix},\boldsymbol{\Lambda}=\begin{bmatrix}\boldsymbol{\Sigma} & \mathbf{0}_{M\times M}\\ \mathbf{0}_{M\times M} & -\boldsymbol{\Sigma}\end{bmatrix}. \tag{28}$$

Proof. Both $\boldsymbol{V}$ and $\boldsymbol{U}$ are orthonormal sets, therefore, ${\boldsymbol{u}_i}^{T}\boldsymbol{u}_j=\Delta_{i,j}$, and ${\boldsymbol{v}_i}^{T}\boldsymbol{v}_j=\Delta_{i,j}$, thus, the set $\{\boldsymbol{\pi}_m\}$ is orthonormal. Therefore, $\boldsymbol{\Pi}\boldsymbol{\Pi}^{T}=\boldsymbol{I}$. By direct substitution of Eq. (28), $\boldsymbol{\Pi}\boldsymbol{\Lambda}\boldsymbol{\Pi}^{T}$ can be computed explicitly as

$$\begin{aligned}\boldsymbol{\Pi}\boldsymbol{\Lambda}\boldsymbol{\Pi}^{T}&=\frac{1}{2}\begin{bmatrix}\boldsymbol{V} & \boldsymbol{V}\\ \boldsymbol{U} & -\boldsymbol{U}\end{bmatrix}\begin{bmatrix}\boldsymbol{\Sigma} & \mathbf{0}_{M\times M}\\ \mathbf{0}_{M\times M} & -\boldsymbol{\Sigma}\end{bmatrix}\begin{bmatrix}\boldsymbol{V}^{T} & \boldsymbol{U}^{T}\\ \boldsymbol{V}^{T} & -\boldsymbol{U}^{T}\end{bmatrix}\\ &=\frac{1}{2}\begin{bmatrix}\boldsymbol{V}\boldsymbol{\Sigma} & -\boldsymbol{V}\boldsymbol{\Sigma}\\ \boldsymbol{U}\boldsymbol{\Sigma} & \boldsymbol{U}\boldsymbol{\Sigma}\end{bmatrix}\begin{bmatrix}\boldsymbol{V}^{T} & \boldsymbol{U}^{T}\\ \boldsymbol{V}^{T} & -\boldsymbol{U}^{T}\end{bmatrix}=\frac{1}{2}\begin{bmatrix}\mathbf{0}_{M\times M} & 2\boldsymbol{K}^{z}\\ (2\boldsymbol{K}^{z})^{T} & \mathbf{0}_{M\times M}\end{bmatrix}=\hat{\boldsymbol{K}}.\end{aligned} \tag{29}$$

□

Thus the proposed mapping in Eq. (21) could be computed for $L=2$ using the SVD of $\boldsymbol{K}^{z}$, Eq. (28) and $\boldsymbol{\Psi}=\hat{\boldsymbol{D}}^{-1/2}\boldsymbol{\Pi}$.

5.3. Cross-view (CV) diffusion distance

In some physical systems the observed dataset $\boldsymbol{X}$ may depend on some underlying parameter, say α. Under this model, multiple snapshots (views) can typically be obtained for various values of α, each denoted $\boldsymbol{X}^{\alpha}$. An example of such a scenario occurs in hyper-spectral images that change over time. Quantifying the amount of change in the datasets due to a change in α in such models is often desired, but if the datasets are high-dimensional, this can be quite a challenging task. This scenario was recently studied in [38], where the DM framework was applied to each fixed value of α. Then, by using the extracted low-dimensional mappings, the Euclidean distance enables to quantify the extent of changes due to α. However, this approach may be rather unstable, since every small change in the data can result in different mappings and the mappings are extracted independently of any mutual influence.

Our multiview approach, on the other hand, incorporates the mutual relations of data within each view, as well as the relations between views. This point of view facilitates a more robust measure of the extent of variation between two datasets that correspond to a small variation in α. To this end, we define a new diffusion distance, which measures the relation between two views, i.e. between all the data points obtained for different values of α. We measure the distance between all the coupled data points among the mappings of the snapshots $\boldsymbol{X}^{\alpha_l}$ and $\boldsymbol{X}^{\alpha_m}$ by using the expression

$$\mathcal{D}_t^{(\mathrm{CV})^2}(\boldsymbol{X}^{\alpha_l},\boldsymbol{X}^{\alpha_m})\triangleq\sum_{i=1}^{M}\left\|\hat{\boldsymbol{\Psi}}_t\left(\boldsymbol{x}_i^{\alpha_l}\right)-\hat{\boldsymbol{\Psi}}_t\left(\boldsymbol{x}_i^{\alpha_m}\right)\right\|^2. \tag{30}$$

Our kernel matrix is a product of the Gaussian kernel matrices in each view. If these values of the kernel matrices $(\boldsymbol{K}^{x^{\alpha_l}},\boldsymbol{K}^{x^{\alpha_m}})$ are similar, this corresponds to similarity between the views' inner geometry. The right- and left-singular vectors of the matrix $\boldsymbol{K}^{x^{\alpha_l}}\boldsymbol{K}^{x^{\alpha_m}}$ will be similar, thus, $\mathcal{D}_t^{(\mathrm{CV})}$ will be small.

Theorem 4. *Using Gaussian kernels with identical σ values, the cross manifold distance (defined in Eq. (30)) is invariant to orthonormal transformations between the ambient spaces $\boldsymbol{X}^{\alpha_l}$ and $\boldsymbol{X}^{\alpha_m}$.*

Proof. Denote an orthonormal transformation matrix $\boldsymbol{R}:\boldsymbol{X}^{\alpha_l}\to\boldsymbol{X}^{\alpha_m}$, w.l.o.g. by $\boldsymbol{x}_i^{\alpha_m}=\boldsymbol{R}\boldsymbol{x}_i^{\alpha_l}$. Then

$$\begin{aligned}K_{i,j}^{x^{\alpha_m}}&=\exp\left\{-\frac{\|\boldsymbol{x}_i^{\alpha_m}-\boldsymbol{x}_j^{\alpha_m}\|^2}{2\sigma_m^2}\right\}=\exp\left\{-\frac{\|\boldsymbol{R}\boldsymbol{x}_i^{\alpha_l}-\boldsymbol{R}\boldsymbol{x}_j^{\alpha_l}\|^2}{2\sigma_m^2}\right\}\\ &=\exp\left\{-\frac{\|\boldsymbol{x}_i^{\alpha_l}-\boldsymbol{x}_j^{\alpha_l}\|^2}{2\sigma_l^2}\right\}=K_{i,j}^{x^{\alpha_l}}.\end{aligned} \tag{31}$$

The penultimate transition is due to the orthonomality of $\boldsymbol{R}$ and to the identical $\sigma_l=\sigma_m$ values. Therefore, the matrix $\boldsymbol{K}^{z}=(\boldsymbol{K}^{x^{\alpha_m}})^2$ from Eq. (12) is symmetric and its right- and left-singular vectors are equal, i.e. $\boldsymbol{U}=\boldsymbol{V}$ in Eq. (28). This induces a repetitive form in $\boldsymbol{\Psi}=\hat{\boldsymbol{D}}^{-1/2}\boldsymbol{\Pi}\to\psi_l[i]=\psi_l[M+i],\ 1\le i,l\le M-1\to\Psi_t(\boldsymbol{x}_i^{\alpha_l})=\Psi_t(\boldsymbol{x}_i^{\alpha_m})$, thus, $\mathcal{D}_t^{(CM)^2}(\boldsymbol{X}^{\alpha_l},\boldsymbol{X}^{\alpha_m})=0$. □

5.4. Spectral decay of $\hat{\mathbf{K}}$

The power of kernel based methods for dimensionality reduction stems from the spectral decay of the kernel matrix' eigenvalues. In this section we study the relation between the spectral decay of the Kernel Product (Eq. (6)) and our multi-view kernel (Eq. (14)).

Theorem 5. *All $2M$ eigenvalues of $\hat{\boldsymbol{P}}$ (Eq. (14)) are real-valued and bounded, $|\lambda_i|\le 1$, $i=0,\dots,2M-1$.*

Proof. [1] As shown in Section 5.2, $\hat{\boldsymbol{P}}$ is algebraically similar to a symmetric matrix, thus its eigenvalues are guaranteed to be real-valued. Denote by λ and $\boldsymbol{\psi}$ an eigenvalue and an eigenvector, resp., such that $\lambda\boldsymbol{\psi}=\hat{\boldsymbol{P}}\boldsymbol{\psi}$.

Define $i_0\triangleq\arg\max_i|\psi[i]|$ (the index of the largest element of $\boldsymbol{\psi}$). The maximal value $\psi[i_0]$ can be computed using $\hat{\boldsymbol{P}}$ from Eq. (14),

$$\begin{aligned}\lambda\psi[i_0]&=\sum_{j=0}^{2M-1}\hat{P}_{i_0j}\psi[j]\Rightarrow|\lambda|=\left|\sum_{j=0}^{2M-1}\hat{P}_{i_0j}\frac{\psi[j]}{\psi[i_0]}\right|\le\sum_{j=0}^{2M-1}\hat{P}_{i_0j}\frac{|\psi[j]|}{|\psi[i_0]|}\\ &\le\sum_{j=1}^{2M}\hat{P}_{i_0j}=1.\end{aligned} \tag{32}$$

[1] A similar proof is given in https://sites.google.com/site/yoelshkolnisky/teaching.

The first inequality is due to the triangle inequality and the second equality is due to the definition of i_0.

□

Although, according to this Theorem, the eigenvalues are all real-valued and bounded, their mere boundedness is generally insufficient for dimensionality reduction. Dimensionality reduction is meaningful only in the presence of a significant spectral decay.

Definition 1. Let $\mathcal{M}$ be a manifold. The intrinsic dimension d of the manifold is a positive integer determined by how many independent "coordinates" are needed to describe $\mathcal{M}$. Using a parametrization to describe a manifold, the dimension of $\mathcal{M}$ is the smallest integer d such that a smooth map $\boldsymbol{f}(\xi) = \mathcal{M}$ describes the manifold, where $\xi \in \mathbb{R}^d$.

Our framework employs a Gaussian kernel, the spectral decay of which was studied in [5]. We use Lemma 1 (which is based on Weyl's asymptotic law and appears in [31]) to evaluate the spectral decay of our kernel.

Lemma 1. *Assume that the data is sampled from a manifold with intrinsic dimension $d \ll M$. Let $\boldsymbol{K}^{\circ} \in \mathbb{R}^{M\times M}$ (a kernel constructed based on the concatenation of views) denote the kernel with an exponential decay as a function of the Euclidean distance. For $\delta > 0$, the number of eigenvalues of $\boldsymbol{K}^{\circ}$ above δ is proportional to $(\log(\frac{1}{\delta}))^d$.*

Assume that the eigenvalues of $\boldsymbol{K}^{\circ}$ are arranged in descending order of their absolute values, $|\lambda_0| \geq |\lambda_1| \geq \cdots \geq |\lambda_{M-1}|$, set $\delta \in (0, 1)$ and define $r_\delta \triangleq \max\{\ell \in [1, M] : |\lambda_{\ell-1}| > \delta\}$ (denoting the number of eigenvalues of $\boldsymbol{K}^{\circ}$ above δ). Recall now that $\boldsymbol{K}^{\circ} = \boldsymbol{K}^x \circ \boldsymbol{K}^y$ corresponds to a single DM view (formed of the concatenation of two views) as addressed in [5]. Theorem 6 relates the spectral decay of our proposed kernel $\widehat{\boldsymbol{P}}$ (Eq. (14)) to the decay of the Kernel Product-based DM ($\boldsymbol{P}^{\circ}$).

Lemma 2. *Let $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{M\times M}$ be any two positive semi-definite (PSD) matrices. Then for any $1 \leq k \leq M-1$*

$$\prod_{\ell=k}^{M-1} \lambda_\ell(\boldsymbol{A}\cdot\boldsymbol{B}) \leq \prod_{\ell=k}^{M-1} \lambda_\ell(\boldsymbol{A}\circ\boldsymbol{B}) \tag{33}$$

where $\lambda_\ell(\cdot)$ denotes the ℓth eigenvalue (in descending order) of the enclosed matrix.

This inequality is proved in [39] and [40].

Theorem 6. *The product of the last $M-1-r_\delta$ eigenvalues of $\boldsymbol{K}^z$ is smaller or equal to δ^{M-1-r_δ}. Formally, $\prod_{\ell=r_\delta}^{M-1} \lambda_\ell(\boldsymbol{K}^x\cdot\boldsymbol{K}^y) \leq \delta^{M-1-r_\delta}$.*

Proof. Substitute the PSD matrices $\boldsymbol{A} = \boldsymbol{K}^x$ and $\boldsymbol{B} = \boldsymbol{K}^y$ in Lemma 2 and choose $\ell = r_\delta$ in Eq. (33) to obtain

$$\prod_{\ell=r_\delta}^{M-1} \lambda_\ell(\boldsymbol{K}^x\cdot\boldsymbol{K}^y) \leq \prod_{\ell=r_\delta}^{M-1} \lambda_\ell(\boldsymbol{K}^{\circ}) \leq \delta^{M-1-r_\delta}. \tag{34}$$

□

Relying on the spectral decay of the kernel matrix, we can approximate Eq. (18) by neglecting all eigenvalues smaller than δ. Thus, we can compute a low dimensional mapping such that

$$\widehat{\boldsymbol{\Psi}}_t^r(\boldsymbol{x}_i) : \boldsymbol{x}_i \longmapsto \left[\lambda_1^t\psi_1[i], \lambda_2^t\psi_2[i], \lambda_3^t\psi_3[i], \ldots, \lambda_{r-1}^t\psi_{r-1}[i]\right]^T \in \mathbb{R}^{r-1}. \tag{35}$$

The following Lemma introduces an error bound for using this low-dimensional mapping:

Lemma 3. *The truncated diffusion distance up to coordinate r defined as*

$$[D_t^r(\boldsymbol{x}_i, \boldsymbol{x}_j)]^2 \triangleq \left\|\widehat{\boldsymbol{\Psi}}_t^r(\boldsymbol{x}_i) - \widehat{\boldsymbol{\Psi}}_t^r(\boldsymbol{x}_j)\right\|^2 = \sum_{s=1}^{r} \lambda_s^{2t}(\psi_s[i] - \psi_s[j])^2, \tag{36}$$

is bounded by the inner view diffusion distance (defined in Eq. (16))

$$2\cdot\left[\sum_{s=1}^{M-1}\lambda_s^{2t}(\psi_s[i]-\psi_s[j])^2 - \delta^{2t}\cdot\left(\frac{1-\Delta_{i,j}}{\widehat{D}_{i,j}^{\min}}\right)\right] \leq \left[D_t^r(\boldsymbol{x}_i,\boldsymbol{x}_j)\right]^2$$
$$\leq 2\cdot\sum_{s=1}^{M-1}\lambda_s^{2t}(\psi_s[i]-\psi_s[j])^2,$$

where $\widehat{D}_{i,j}^{\min}$ is the minimal value of $\widehat{D}_{i,i}$ and $\widehat{D}_{j,j}$, $\Delta_{i,j}$ is the Kronecker delta function and δ is the accuracy threshold.

Proof. For the last inequality, clearly

$$[D_t^r(\boldsymbol{x}_i,\boldsymbol{x}_j)]^2 \leq [D_t^{2M}(\boldsymbol{x}_i,\boldsymbol{x}_j)]^2 = 2\cdot\sum_{s=1}^{M-1}\lambda_s^{2t}(\psi_s[i]-\psi_s[j])^2,$$

the equality is a result of the repetitive form of $\boldsymbol{\Psi}$ and $\boldsymbol{\Lambda}$, which were defined in Theorem 3 for $L = 2$. Note that $s = 0$ was excluded from the sum as $\boldsymbol{\psi}_0 = \mathbf{1}$ is constant. For the first inequality, using $\boldsymbol{\Psi} = \widehat{\boldsymbol{D}}^{-1/2}\boldsymbol{\Pi}$ where $\boldsymbol{\Pi}$ is an orthonormal basis defined in Eq. (28), we get

$$\boldsymbol{\Psi}\boldsymbol{\Psi}^T = \widehat{\boldsymbol{D}}^{-1/2}\boldsymbol{\Pi}\boldsymbol{\Pi}^T\widehat{\boldsymbol{D}}^{-1/2} = \widehat{\boldsymbol{D}}^{-1},$$

which means that

$$\sum_{s=0}^{2M-1}(\psi_s[i]-\psi_s[j])^2 = \frac{1}{\widehat{D}_{i,i}} + \frac{1}{\widehat{D}_{j,j}} - \frac{2\Delta_{i,j}}{\widehat{D}_{i,i}}. \tag{37}$$

Using the definition of the truncated diffusion distance we have

$$\left[D_t^r(\boldsymbol{x}_i,\boldsymbol{x}_j)\right]^2 = \sum_{s=0}^{2M-1}\lambda_s^{2t}(\psi_s[i]-\psi_s[j])^2 - \sum_{s=r+1}^{2M-1}\lambda_s^{2t}(\boldsymbol{\psi}_s[i]-\boldsymbol{\psi}_s[j])^2$$
$$\geq 2\sum_{s=1}^{M-1}\lambda_s^{2t}(\psi_s[i]-\psi_s[j])^2 - \delta^{2t}\sum_{s=0}^{2M-1}(\psi_s[i]-\psi_s[j])^2$$
$$\geq 2\left[\sum_{s=1}^{2M-1}\lambda_s^{2t}(\psi_s[i]-\psi_s[j])^2 - \delta^{2t}\cdot\left(\frac{1-\Delta_{i,j}}{\widehat{D}_{i,j}^{\min}}\right)\right]. \tag{38}$$

In the same way, a similar bound for the truncated diffusion distance between $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ can be derived. □

The dimension r is typically determined by looking at the decay rate of the eigenvalues and on choosing a threshold level δ. In this section, we demonstrated how the dimension r and the threshold δ are related to the approximation of the diffusion distance. By removing all coordinates with eigenvalues smaller than δ, the error of the diffusion distance is bounded according to Lemma 3. The decaying property of the eigenvalues could be quantified by the numerical rank of the Gaussian kernel matrix. Next, we present some existing results that shed some light onto the practicality of the truncation of the diffusion coordinates up to dimension r. In practice, there usually is no single optimal value for r, and the number of necessary coordinates depends on the application.

5.4.1. Intrinsic dimension and numerical rank

In [41] Bermanis et al. prove that the numerical rank of a Gaussian kernel matrix is typically independent of its size. Their results state that the numerical rank is proportional to the volume of the data. By using boxes with side-length of $\epsilon^{D/2} = \sigma^D$ (D is the dimension of the data), it is shown that the numerical rank is bounded from above by the minimal number of such cubes required to cover the data. As the number of eigenvalues $|\lambda_\ell| > \delta$ is proportional to the numerical rank, this means that for a prescribed δ, the number of coordinates r_δ required is independent of the number of samples N.

Practically speaking, this result is useful if the numerical rank is small. Furthermore, as there are several methods for estimating the numerical rank, using the numerical rank to choose r_δ is not very expensive. Another practical method, which we found useful for setting r without directly choosing δ, is based on estimating the intrinsic dimension of the data d. Various methods have been proposed for estimating

the intrinsic dimension of high-dimensional data. Here, we provide a brief description of a few. Popular methods, such as [42,43] apply local PCA to the data, and estimate the intrinsic dimension as the median of the number of principal components corresponding to singular values larger than some threshold. Based on our experience, we found that a method called Dimensionality from Angle and Norm Concentration (DANCo) [44] provides a robust estimation of the intrinsic dimension d. DANCo generates artificial data of dimension d and attempts to minimize the KullbackLeibler divergence between the estimated probability distribution functions (pdf-s) of the observed data and the artificially-generated samples.

5.5. Out-of-sample extension

To extend the diffusion coordinates to new data points without re-applying a large-scale eigendecomposition [32], the Nyström extension [45] is widely used. Here we formulate the extension method for a multi-view scenario. Given the data sets $\boldsymbol{X}$ and $\boldsymbol{Y}$ and new points $\tilde{\boldsymbol{x}} \notin \boldsymbol{X}$ and $\tilde{\boldsymbol{y}} \notin \boldsymbol{Y}$, we want to extend the multi-view diffusion mapping to $\tilde{\boldsymbol{x}}$ and $\tilde{\boldsymbol{y}}$ without re-applying the batch procedure. First, we describe the explicit form for the eigenvalue problem for two views $\boldsymbol{X}$ and $\boldsymbol{Y}$. The eigenvector $\boldsymbol{\psi}_k$ and its associated eigenvalue λ_k satisfy $\lambda_k \boldsymbol{\psi}_k = \widehat{\boldsymbol{P}} \boldsymbol{\psi}_k$. Substituting the definition of $\widehat{\boldsymbol{P}}$ from Eqs. (12) and (14) we get that

$$\lambda_k \boldsymbol{\psi}_k = \widehat{\boldsymbol{D}}^{-1} \begin{bmatrix} \mathbf{0}_{M\times M} & \boldsymbol{K}^x \cdot \boldsymbol{K}^y \\ \boldsymbol{K}^y \cdot \boldsymbol{K}^x & \mathbf{0}_{M\times M} \end{bmatrix} \cdot \boldsymbol{\psi}_k \in \mathbb{R}^{2M},$$

due to the block form of the matrix $\widehat{\boldsymbol{P}}$

$$\lambda_k \boldsymbol{\psi}_k^x[i] = \sum_j \hat{p}(\boldsymbol{x}_i, \boldsymbol{y}_j) \boldsymbol{\psi}_k^y[j] \in \mathbb{R}^M,$$
$$\lambda_k \boldsymbol{\psi}_k^y[i] = \sum_j \hat{p}(\boldsymbol{y}_i, \boldsymbol{x}_j) \boldsymbol{\psi}_k^x[j] \in \mathbb{R}^M,$$

where $\boldsymbol{\psi}_k^x[i] \triangleq \boldsymbol{\psi}_k[i], i = 1, \ldots, M$ and $\boldsymbol{\psi}_k^y[i] \triangleq \boldsymbol{\psi}_k[i+M], i = 1, \ldots, M$. The transition matrices are

$$\hat{p}(\boldsymbol{x}_i, \boldsymbol{y}_j) = \frac{\sum_s K_{i,s}^x K_{s,j}^y}{\hat{D}_{i,i}^{\text{rows}}}, \text{ and } \hat{p}(\boldsymbol{y}_i, \boldsymbol{x}_j) = \frac{\sum_s K_{i,s}^y K_{s,j}^x}{\hat{D}_{i,i}^{\text{cols}}}.$$

The Nyström extension is similarly obtained by computing a weighted sum of the original eigenvectors. The weights are computed by applying the kernel $\mathcal{K}$ to the extended data points, followed by row-normalization. For the proposed mapping, the extension is defined by

$$\hat{\psi}_k(\tilde{\boldsymbol{x}}) = \frac{1}{\lambda_k} \sum_j \hat{p}(\tilde{\boldsymbol{x}}, \boldsymbol{y}_j) \psi_k^y[j] \tag{39}$$

$$\hat{\psi}_k(\tilde{\boldsymbol{y}}) = \frac{1}{\lambda_k} \sum_j \hat{P}(\tilde{\boldsymbol{y}}, \boldsymbol{x}_j) \psi_k^x[j] \tag{40}$$

where the "missing" probabilities $\hat{p}(\tilde{\boldsymbol{x}}, \boldsymbol{y}_j)$ and $\hat{p}(\tilde{\boldsymbol{y}}, \boldsymbol{x}_j)$ are approximated using the original kernel function as

$$\hat{p}(\tilde{\boldsymbol{x}}, \boldsymbol{y}_j) = \sum_s \exp\left\{ -\frac{||\tilde{\boldsymbol{x}} - \boldsymbol{x}_s||^2}{2\sigma_x^2} \right\} \exp\left\{ -\frac{||\boldsymbol{y}_s - \boldsymbol{y}_j||^2}{2\sigma_y^2} \right\} \frac{1}{\hat{D}_{j+M,j+M}} \tag{41}$$

$$\hat{p}(\tilde{\boldsymbol{y}}, \boldsymbol{x}_j) = \sum_s \exp\left\{ -\frac{||\tilde{\boldsymbol{y}} - \boldsymbol{y}_s||^2}{2\sigma_y^2} \right\} \exp\left\{ -\frac{||\boldsymbol{x}_s - \boldsymbol{x}_j||^2}{2\sigma_x^2} \right\} \frac{1}{\hat{D}_{j+M,j+M}} \tag{42}$$

and where $\psi_k^x[j] = \psi_k[j]$ and $\psi_k^y[j] = \psi_k[m+j]$.

The new mapping vector for the new data point is then given by

$$\hat{\boldsymbol{\Psi}}(\tilde{\boldsymbol{x}}) = \left[\lambda_1 \hat{\psi}_1(\tilde{\boldsymbol{x}}), \lambda_2 \hat{\psi}_2(\tilde{\boldsymbol{x}}), \lambda_3 \hat{\psi}_3(\tilde{\boldsymbol{x}}), \ldots, \lambda_{M-1} \hat{\psi}_{M-1}(\tilde{\boldsymbol{x}})\right] \in \mathbb{R}^{M-1}. \tag{43}$$

The new coordinates in the diffusion space are approximated and the new data points $\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}$ have no effect on the original map's structure.

5.6. Infinitesimal generator

In the subsequent analysis we only consider kernel functions operating on the scaled difference between their two arguments, namely kernel functions which can be written as

$$\boldsymbol{\mathcal{K}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = K\left(\frac{\boldsymbol{x}_i - \boldsymbol{x}_j}{\sqrt{\epsilon}} \right), \tag{44}$$

where ϵ is a positive constant, and $K(\boldsymbol{z}) : \mathbb{R}^D \mapsto \mathbb{R}$ is a non-negative symmetric function, namely $R(\boldsymbol{z}) \geq 0$ and $R(\boldsymbol{z}) = R(-\boldsymbol{z})\ \forall \boldsymbol{z} \in \mathbb{R}^D$. For example, the Gaussian kernel (see Section 2.2) $\mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left\{ -\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\sigma_x^2} \right\}$ satisfies this property with $\epsilon = \sigma^2$.

A family of differently normalized diffusion operators was introduced in [5]. If appropriate limits are taken such that $M \to \infty$, $\epsilon \to 0$, then from Coifman and Lafon [5] it follows that the DM kernel operator will converge to one of the following differential operators: 1. Normalized graph Laplacian; 2. Laplace-Beltrami diffusion; or 3. Heat kernel equation. These are proved in [5]. The operators are all special cases of the diffusion equation. This convergence provides not only a physical justification for the DM framework, but allows in some cases to distinguish between the geometry and the density of the data points. In this section we study the asymptotic properties of the proposed kernel $\widehat{\boldsymbol{K}}$ (Eq. (13)), limiting the discussion to only two views, i.e. $L = 2$.

We are interested in understanding the properties of the eigenfunctions of the proposed multi-view kernel $\widehat{\boldsymbol{P}}$ (Eqs. (13) and (14)) for two views. We assume that there is some unknown mapping $\boldsymbol{\beta} : \mathbb{R}^d \to \mathbb{R}^d$ from view $\boldsymbol{X}$ to view $\boldsymbol{Y}$ that satisfies $\boldsymbol{y}_i = \boldsymbol{\beta}(\boldsymbol{x}_i), i = 1, \ldots, M$. Each view-specific kernel applies the same function, namely $K^x(\boldsymbol{z}) = K^y(\boldsymbol{z}) = K(\boldsymbol{z})$, and $K(\boldsymbol{z})$ is normalized such that $\int_{\mathbb{R}^d} K(\boldsymbol{z}) \mathrm{d}\boldsymbol{z} = 1$. Note that the Gaussian kernel can be properly normalized to satisfy this requirement. The analysis relates to data points $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M\} \in \mathbb{R}^D$ sampled from a uniform distribution over a bounded domain in $\mathbb{R}^d$. The image of the function $\boldsymbol{\beta}$ is a bounded domain in $\mathbb{R}^d$ with distribution $\alpha(\boldsymbol{z})$.

Theorem 7. *The infinitesimal generator induced by the proposed kernel matrix $\widehat{\boldsymbol{K}}$ (Eq. (13)) after row-normalization, denoted in here as $\widehat{\boldsymbol{P}}$, converges when $M \to \infty$, $\epsilon \to 0$ (with $\epsilon = \sigma_x^2 = \sigma_y^2$) to a "cross domain Laplacian operator". The functions $f(\boldsymbol{x})$ and $g(\boldsymbol{y})$ converge to eigenfunctions of $\widehat{\boldsymbol{P}}$. These functions are the solutions of the following diffusion-like equations:*

$$(\widehat{\boldsymbol{P}} f)(\boldsymbol{x}_i) = g(\boldsymbol{\beta}(\boldsymbol{x}_i)) + \epsilon \triangle \gamma(\boldsymbol{\beta}(\boldsymbol{x}_i)) / \alpha(\boldsymbol{\beta}(\boldsymbol{x}_i)) + \mathcal{O}(\epsilon^{3/2}), \tag{45}$$

$$(\widehat{\boldsymbol{P}} g)(\boldsymbol{y}_i) = f(\boldsymbol{\beta}^{-1}(\boldsymbol{y}_i)) + \epsilon \triangle \eta(\boldsymbol{\beta}^{-1}(\boldsymbol{y}_i)) / \alpha(\boldsymbol{\beta}(\boldsymbol{y}_i) + \mathcal{O}(\epsilon^{3/2}), \tag{46}$$

where the functions γ, η are defined as $\gamma(\boldsymbol{z}) \triangleq g(\boldsymbol{z})\alpha(\boldsymbol{z}), \eta(\boldsymbol{z}) \triangleq f(\boldsymbol{z})\alpha(\boldsymbol{z})$.

The proof of Theorem 7 is deferred to the Appendix. Although the interpretation of the result is hardly intuitive, it provides evidence that the mapping $\boldsymbol{f}$ and $\boldsymbol{g}$ are indeed coupled and are related to the second derivative on the manifold. However, the distribution of the points α has an impact on the mapping, which we hope to reduce in future research.

5.7. The convergence rate

In Theorem 7, we assume the number of data points $M \to \infty$ while the scale parameter $\epsilon \to 0$. In practice we cannot expect to have an infinite number of data points. It was shown in [5,46] (and elsewhere) that a single-view graph Laplacian converges to the laplacian operator on a manifold. It is demonstrated in [47,48] that the variance of the error for such an operator decreases as $M \to \infty$, but increases as $\epsilon \to 0$. The study in [47] proves that for a uniform distribution of data points, the variance of the error is bounded by $\mathcal{O}(\frac{1}{M^{1/2}\epsilon^{1+d/4}}, \epsilon^{1/2})$. This bound was improved in [48] by an asymptotic factor of $\sqrt{\epsilon}$ based on the correlation between $\boldsymbol{D}^{-1}$ and $\boldsymbol{K}$.

We now turn our attention to the variance of the multi-view kernel for a finite number of points. Given $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M$ independent uniformly

distributed data points sampled from a bounded domain in $\mathbb{R}^d$, define the multi-view Parzen Window density estimator by

$$\dot{\boldsymbol{K}}_{M,\epsilon}(\boldsymbol{x}) \triangleq \frac{1}{M^2}\sum_{\ell=1}^{M}\sum_{j=1}^{M}\frac{1}{\epsilon^d}K\left(\frac{\boldsymbol{x}-\boldsymbol{x}_\ell}{\sqrt{\epsilon}}\right)K\left(\frac{\boldsymbol{y}_\ell-\boldsymbol{y}_j}{\sqrt{\epsilon}}\right). \tag{47}$$

We are interested in finding a bound for the variance of $\dot{\boldsymbol{K}}_{M,\epsilon}(\boldsymbol{x})$ for a finite number of data points:

$$\begin{aligned}&\mathrm{var}(\dot{\boldsymbol{K}}_{M,\epsilon}(\boldsymbol{x})) = \tfrac{1}{M^4\epsilon^{2d}}\cdot M\cdot \mathrm{var}\left(\textstyle\sum_\ell^M K\left(\frac{\boldsymbol{x}-\boldsymbol{x}_\ell}{\sqrt{\epsilon}}\right)K\left(\frac{\boldsymbol{y}_\ell-\boldsymbol{y}_j}{\sqrt{\epsilon}}\right)\right)\\ &\le \tfrac{1}{M^4\epsilon^{2d}}\cdot M^3\cdot \mathrm{var}\left(K_\epsilon^x K_\epsilon^y\right) \le \tfrac{1}{M\epsilon^{2d}}\left[\mathrm{var}\left(K_\epsilon^x\right)\cdot ||K_\epsilon^y\Big|_\infty + \mathrm{var}\left(K_\epsilon^y\right)\cdot ||K_\epsilon^x\Big|_\infty\right]\\ &\le \tfrac{1}{M\epsilon^{2d}}\cdot\left[\epsilon^{d/2}\cdot m_1\cdot 1+\epsilon^{d/2}\cdot m_2\cdot\alpha(\boldsymbol{x})\right] \le \tfrac{m_1+m_2\cdot\alpha(\boldsymbol{x})}{M\cdot\epsilon^{1.5d}},\end{aligned} \tag{48}$$

where the constants m_1 and m_2 are functions of the chosen kernels and of the pdf $\alpha(\cdot)$ of the points $\boldsymbol{y}_i, i=1,\ldots,M$. This bound helps to choose an optimal value for the scaling factor ϵ given the number of data points M and the intrinsic dimension d.

5.8. Generalized multi-view kernel

One can consider a more general multi-view kernel which does not preclude a transition within views $\boldsymbol{X}$ and $\boldsymbol{Y}$ in each time step. Such a kernel will take the form

$$\widehat{\widehat{\boldsymbol{K}}} = \begin{bmatrix}\eta\cdot(\boldsymbol{K}^x)^2 & (1-\eta)\cdot\boldsymbol{K}^x\boldsymbol{K}^y\\ (1-\eta)\cdot\boldsymbol{K}^y\boldsymbol{K}^x & \eta\cdot(\boldsymbol{K}^y)^2\end{bmatrix}, \tag{49}$$

where the parameter $\eta\in[0,1]$ prescribes the (implied) probability of within-view transitions. This kernel is normalized using the sum of rows diagonal matrix $\widehat{\widehat{\boldsymbol{D}}}$, such that $\widehat{\widehat{\boldsymbol{P}}} = \widehat{\widehat{\boldsymbol{D}}}^{-1}\widehat{\widehat{\boldsymbol{K}}}$. For large values of η, the kernel favors the within-view transitions, thereby sharing characteristics with the single-view diffusion process. For small values of η, the kernel tends to behave like our multi-view kernel $\widehat{\boldsymbol{K}}$ (Eq. (13)).

5.9. Additive noise

In this section we explore the effect of noise by following the analysis presented in [18]: Assuming that the noise is additive, we evaluate how the proposed method preserves the statistics of the latent model parameters $\boldsymbol{\Theta}\in\mathbb{R}^{d\times M}$. We begin by considering the case of a single-view linear model, then consider a multi-view (linear model) and finally use the kernel trick to relax the linearity assumption.

The single-view linear model is defined by $\boldsymbol{x}_i = \boldsymbol{A}\boldsymbol{\theta}_i + \boldsymbol{\xi}_i, i=1,\ldots,M$, where the vectors $\boldsymbol{\theta}_i\in\mathbb{R}^d, i=1,\ldots,M$ (columns of $\boldsymbol{\Theta}$) describe latent parameters, modeled as i.i.d., zero-mean random vectors with a finite covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}$, and $\boldsymbol{\xi}_i\in\mathbb{R}^d$ are i.i.d., zero-mean random "noise" vectors with a finite covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\xi}}$. The observations are $\boldsymbol{x}_i\in\mathbb{R}^D$ and the matrix $\boldsymbol{A}\in\mathbb{R}^{D\times d}$ has full column rank.

A standard approach for reducing the dimension is to apply PCA. However, in the presence of noise, even if the covariance of $\boldsymbol{\Theta}$ is full rank PCA will provide a biased representation of $\boldsymbol{\Theta}$. In applying PCA to the set $\boldsymbol{X}$, one first computes the sample covariance $\frac{1}{M}\boldsymbol{X}\boldsymbol{X}^T$, which at the limit $M\to\infty$ converges to $\boldsymbol{A}\boldsymbol{\Sigma}_{\boldsymbol{\theta}}\boldsymbol{A}^T+\boldsymbol{\Sigma}_{\boldsymbol{\xi}}$. This means that using the top principal components of $\boldsymbol{X}$ embeds the data into a biased representation of $\boldsymbol{\Theta}$. Nonetheless, by introducing a second set of measurements $\boldsymbol{y}_i = \boldsymbol{B}\boldsymbol{\theta}_i+\boldsymbol{\eta}_i, i=1,\ldots,M$, where $\boldsymbol{B}\in\mathbb{R}^{D\times d}$ is another full-rank matrix and $\boldsymbol{\eta}_i$ are i.i.d. zero-mean noise vectors which are all uncorrelated with the respective $\boldsymbol{\xi}_i$ (namely, $E\left[\boldsymbol{\xi}_i\cdot\boldsymbol{\eta}_i^{\mathrm{T}}\right]=\boldsymbol{0}, i=1,\ldots,M$), one can remove the bias term. Essentially, the corresponding additional measurements, along with the noise decorrelation assumption, provide the information required for retrieving an unbiased representation of $\boldsymbol{\Theta}$. Removing the bias term is possible by applying CCA [49] to $\boldsymbol{X}$ and $\boldsymbol{Y}$. CCA not only removes the bias, but enables to uncover a reduced representation that preserves the statistics of $\boldsymbol{\theta}$. This is because the sample cross covariance of $\boldsymbol{X}$ and $\boldsymbol{Y}$ is an unbiased estimate of $\boldsymbol{A}\boldsymbol{\Sigma}_{\boldsymbol{\theta}}\boldsymbol{B}^T$. The top d right- and left-singular vectors of $\frac{1}{M}\boldsymbol{X}\boldsymbol{Y}^T$ are used to embed the data into a d dimensional space, such that the statistics of $\boldsymbol{\Theta}$ are preserved. By denoting the top d singular vectors as $(\hat{\boldsymbol{U}},\boldsymbol{S},\hat{\boldsymbol{V}}^T) = \mathrm{SVD}_d(\frac{1}{M}\boldsymbol{X}\boldsymbol{Y}^T)$, the reduced representations are defined as $\hat{\boldsymbol{U}}^T\boldsymbol{X}$ and $\hat{\boldsymbol{V}}^T\boldsymbol{Y}$, respectively.

Although CCA is an effective, powerful tool in this context, it is based on a linear model, and thus limited to linear transformations of the data. A widely used solution to capture non-linear relations in the data are Kernel matrices [50–52]. Using a kernel matrix in the ambient space is a natural way to capture the sample-covariance in some unknown high-dimensional feature space [53]. Therefore, using a kernel generalizes the results provided by CCA to the nonlinear case. We now demonstrate how under mild assumptions the proposed kernel $\boldsymbol{K}^z = \boldsymbol{K}^x\boldsymbol{K}^y$ mitigates the bias effect of the additive uncorrelated noise.

Assume that $\boldsymbol{x}_i, \boldsymbol{y}_i, i=1,\ldots,M$ are noisy observations of the same low-dimensional latent variable $\boldsymbol{\theta}_i$, such that $\boldsymbol{x}_i = \boldsymbol{r}(\boldsymbol{\theta}_i)+\boldsymbol{\xi}_i$ and $\boldsymbol{y}_i = \boldsymbol{h}(\boldsymbol{\theta}_i)+\boldsymbol{\eta}_i$. The functions $\boldsymbol{r},\boldsymbol{h}:\mathbb{R}^d\to\mathbb{R}^D$. The affinity values of $\boldsymbol{K}^x$, $\boldsymbol{K}^y$ are the sample covariance in two high-dimensional feature spaces $\boldsymbol{\Gamma}$, $\boldsymbol{\Delta}$. The inaccesible feature maps for both views are defined by $\boldsymbol{\gamma}(\boldsymbol{x}_i)$ and $\boldsymbol{\delta}(\boldsymbol{y}_i)$. This means that by computing the SVD of $\boldsymbol{K}^z = \boldsymbol{K}^x\boldsymbol{K}^y$, we would essentially be applying CCA in the inaccessible features spaces $\boldsymbol{\Gamma}$, $\boldsymbol{\Delta}$. This implies that if the following conditions hold

1. The functions $\boldsymbol{r},\boldsymbol{h}\in\mathbb{C}^\infty$ are invertible.
2. The feature representations $\boldsymbol{\Gamma}(\boldsymbol{X})$ and $\boldsymbol{\Delta}(\boldsymbol{Y})$ are centered.
3. The noise terms are uncorrelated: $E\left[\boldsymbol{\xi}_i\cdot\boldsymbol{\eta}_i^{\mathrm{T}}\right]=\boldsymbol{0}, i=1,\ldots,M$.
4. $E[\boldsymbol{\gamma}(\boldsymbol{x}_i)|\boldsymbol{\theta}_i] = a_0\boldsymbol{\theta}_i + a_1$, with some real valued constants a_0 and a_1.
5. $E[\boldsymbol{\delta}(\boldsymbol{y}_i)|\boldsymbol{\theta}_i] = b_0\boldsymbol{\theta}_i + b_1$, with some real valued constants b_0 and b_1.

then, applying SVD to $\boldsymbol{K}^z$ enables to cancel out the view-specific noise terms. Condition (1) implies that the latent parameters $\boldsymbol{\Theta}$ lie on a noisy manifold in both observed spaces. Based on condition (2) and on the choice of kernels, the empirical covariance matrices in the feature space are $\boldsymbol{K}^x = \frac{1}{M}\boldsymbol{\Gamma}\boldsymbol{\Gamma}^T$ and $\boldsymbol{K}^y = \frac{1}{M}\boldsymbol{\Delta}\boldsymbol{\Delta}^T$. We remind that $\boldsymbol{\Gamma}$ and $\boldsymbol{\Delta}$ are inaccessible and the kernels are computed based on $\boldsymbol{X}$ and $\boldsymbol{Y}$. As in the linear case, using $\boldsymbol{K}^x$ or $\boldsymbol{K}^y$ alone is insufficient for obtaining an unbiased estimate of the representation of $\boldsymbol{\Theta}$.

We now turn our attention to demonstrate that the matrix $\boldsymbol{K}^z = \boldsymbol{K}^x\boldsymbol{K}^y = \frac{1}{M^2}\boldsymbol{\Gamma}\boldsymbol{\Gamma}^T\boldsymbol{\Delta}\boldsymbol{\Delta}^T$ enables to remove the uncorrelated additive noise. A right eigenvector of $\boldsymbol{K}^z$, denoted by $\boldsymbol{\tau}_i$, satisfies

$$\boldsymbol{K}^z\boldsymbol{\tau}_i = \frac{1}{M^2}\boldsymbol{\Gamma}\boldsymbol{\Gamma}^T\boldsymbol{\Delta}\boldsymbol{\Delta}^T\boldsymbol{\tau}_i = \lambda_i\boldsymbol{\tau}_i, \tag{50}$$

by multiplying Eq. (50) on the left by $\boldsymbol{\Delta}^T$ we get

$$\frac{1}{M^2}\boldsymbol{\Delta}^T\boldsymbol{\Gamma}\boldsymbol{\Gamma}^T\boldsymbol{\Delta}\boldsymbol{\Delta}^T\boldsymbol{\tau}_i = \lambda_i\boldsymbol{\Delta}^T\boldsymbol{\tau}_i. \tag{51}$$

Substituting $\boldsymbol{u}_i = \boldsymbol{\Delta}^T\boldsymbol{\tau}_i$, $\bar{\boldsymbol{\Sigma}}_{\boldsymbol{\gamma\delta}} = \frac{1}{M}\boldsymbol{\Gamma}^T\boldsymbol{\Delta}$ and $\bar{\boldsymbol{\Sigma}}_{\boldsymbol{\delta\gamma}} = \frac{1}{M}\boldsymbol{\Delta}^T\boldsymbol{\Gamma}$ yields

$$\bar{\boldsymbol{\Sigma}}_{\boldsymbol{\delta\gamma}}\bar{\boldsymbol{\Sigma}}_{\boldsymbol{\gamma\delta}}\boldsymbol{u}_i = \lambda_i\boldsymbol{u}_i, \tag{52}$$

so up to scaling the left- and right-singular vectors of $\bar{\boldsymbol{\Sigma}}_{\boldsymbol{\gamma\delta}}$ provide a low-dimensional representation that captures the statistics of $\boldsymbol{\Theta}$. Thus, based on conditions (3–5), by taking the number of points to infinity, one can extract an unbiased estimate of the representation of $\boldsymbol{\Theta}$.

6. Experimental results

In this section we present experimental results to evaluate our proposed framework. First we empirically evaluate the theoretical properties derived in Section 5. Then, we demonstrate how the proposed framework can be used for learning coupled manifolds even in the presence of noise.

6.1. Empirical evaluations of theoretical aspects

In the first group of experiments we provide empirical evidence corroborating the theoretical analysis from Section 5.

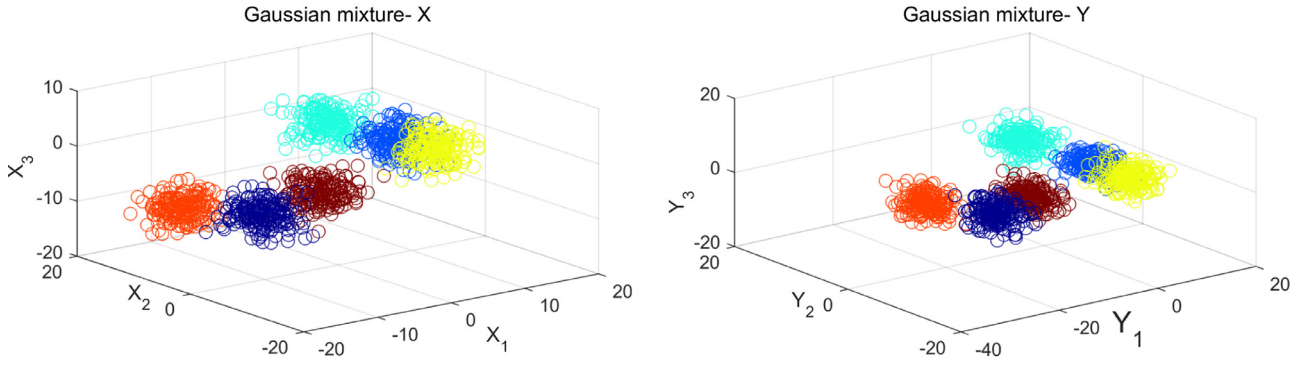

Fig. 4. The first 3 dimensions of the Gaussian mixture. Both views share the center of masses of the Gaussian spread. Left: first view denoted as $\boldsymbol{X}$. Right: second view denoted as $\boldsymbol{Y}$. The variance of the Gaussian in each dimension is 8.

6.1.1. Spectral decay

In Section 5.4, an upper bound on the eigenvalues' rate of decay for our multi-view-based approach (matrix $\widehat{\boldsymbol{P}}$ Eq. (14)) was presented. In order to empirically evaluate the spectral decay for $\widehat{\boldsymbol{P}}$, $\boldsymbol{P}^{\circ}$ (Eq. (6) and [5]) and $\boldsymbol{P}^{+}$, we generated synthetically-clustered data drawn from Gaussian-Mixtures distributions. The following steps describe the generation of both views, denoted ($\boldsymbol{X}$, $\boldsymbol{Y}$) and referred to as View-I ($\boldsymbol{X}$) and View-II ($\boldsymbol{Y}$), resp.:

1. Six vectors $\boldsymbol{\mu}_j \in \mathbb{R}^9$, $j = 1, \ldots, 6$ were drawn from a Gaussian distribution $N(\boldsymbol{0}, 8 \cdot \boldsymbol{I}_{9\times 9})$. These vectors would serve as the centers of masses of the generated classes.
2. One hundred data points were drawn for each cluster $j = 1, \ldots, 6$ from a Gaussian distribution $N(\boldsymbol{\mu}_j, 2 \cdot \boldsymbol{I}_{9\times 9})$. Denote these 600 data points $\boldsymbol{X} \in \mathbb{R}^{9\times 600}$.
3. One hundred additional data points were similarly drawn from each of the six Gaussian distributions $N(\boldsymbol{\mu}_j, 2 \cdot \boldsymbol{I}_{9\times 9})$. Denote these 600 data points$\boldsymbol{Y} \in \mathbb{R}^{9\times 600}$.

The first 3 dimensions of both views are depicted in Fig. 4. We compute the probability matrix for each view $\boldsymbol{P}^{x}$ and $\boldsymbol{P}^{y}$, the Kernel Sum approach probability matrix $\boldsymbol{P}^{+}$, the Kernel Product approach $\boldsymbol{P}^{\circ}$ (Eq. (6)) and the proposed approach $\widehat{\boldsymbol{P}}$. The eigendecomposition is computed for all matrices. The resulting eigenvalues' decay rate are compared with the eigenvalues product from both views. To get a fair comparison between all the methods, we set the Gaussian scale parameters σ_x and σ_y in each view and then use these scales in all the methods. The vectors' variance in the concatenation approach is the sum of variances since we assume statistical independence. Therefore, the following scale parameters $\sigma_{\circ}^2 = \sigma_x^2 + \sigma_y^2$ are used.

The experiment is repeated but this time $\boldsymbol{X}$ contains 6 clusters whereas $\boldsymbol{Y}$ contains only 3. For $\boldsymbol{Y}$, we use only the first 3 centers of masses and generate 200 points in each cluster. Fig. 5 presents a logarithmic scale of the spectral decay for eigenvalues extracted from all methods. It is evident that our proposed kernel has the strongest spectral decay.

6.1.2. Cross view diffusion distance

In this section, we examine the proposed Cross View Diffusion Distance (Section 5.3). A Swiss Roll is generated by using the function

$$\text{View I: } \boldsymbol{x}_i = \begin{bmatrix} x_i[1] \\ x_i[2] \\ x_i[3] \end{bmatrix} = \begin{bmatrix} 6\theta_i \cos(\theta_i) \\ h_i \\ 6\theta_i \sin(\theta_i) \end{bmatrix} + \boldsymbol{n}_i^{(1)}, \tag{53}$$

with $\theta_i = (1.5\pi)s_i$, $i = 1, 2, \ldots, 1{,}000$, where s_i are 1000 data points evenly spread along the segment [1,3], and where $\boldsymbol{n}_i^{(1)}$ are i.i.d. zero-mean Gaussian noise vectors with covariance $\sigma_N^2 \cdot \boldsymbol{I}_{3\times 3}$. The second view is generated by applying an orthonormal transformation to the (noiseless) Swiss Roll and similarly adding Gaussian noise:

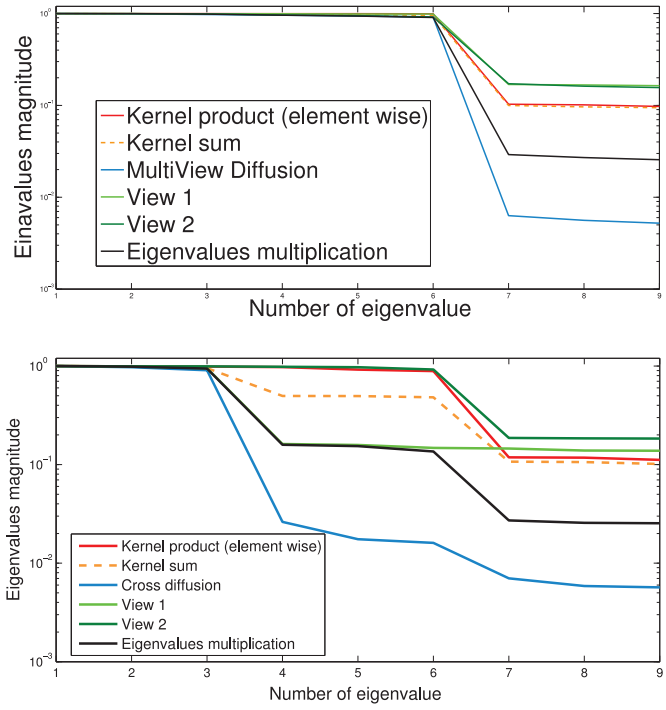

Fig. 5. Eigenvalues decay rate. Comparison between different mapping methods. Top: 6 clusters in each view. Bottom: 6 clusters in $\boldsymbol{X}$ and 3 clusters in $\boldsymbol{Y}$.

$$\text{View II: } \boldsymbol{y}_i = \begin{bmatrix} y_i[1] \\ y_i[2] \\ y_i[3] \end{bmatrix} = \boldsymbol{R} \begin{bmatrix} 6\theta_i \cos(\theta_i) \\ h_i \\ 6\theta_i \sin(\theta_i) \end{bmatrix} + \boldsymbol{n}_i^{(2)}, \tag{54}$$

where $\boldsymbol{R} \in \mathbb{R}^{3\times 3}$ is a random orthonormal transformation matrix, and where $\boldsymbol{n}_i^{(2)}$ are i.i.d. $N(\boldsymbol{0}, \sigma_N^2 \cdot \boldsymbol{I}_{3\times 3}$. The matrix $\boldsymbol{R}$ is generated by independently drawing its elements from a standard Gaussian distribution, followed by applying the Gram-Schmidt orthogonalization procedure. The variables $h_i, i = 1, \ldots, 1000$ are drawn from a uniform distribution in the interval [0,100]. An example for both Swiss Rolls is shown in Fig. 6.

A standard DM is applied to each view and a 2-dimensional embedding of the Swiss Roll is extracted. The sum of distances between all the data points in the embedding spaces is denoted as a single-view diffusion distance (SVDD). The distance is computed using the measure

$$\mathcal{D}_t^{(\mathrm{SV})^2}(X, Y) = \sum_{i=1}^{M} ||\Psi_t(\boldsymbol{x}_i) - \Psi_t(\boldsymbol{y}_i)||^2, \tag{55}$$

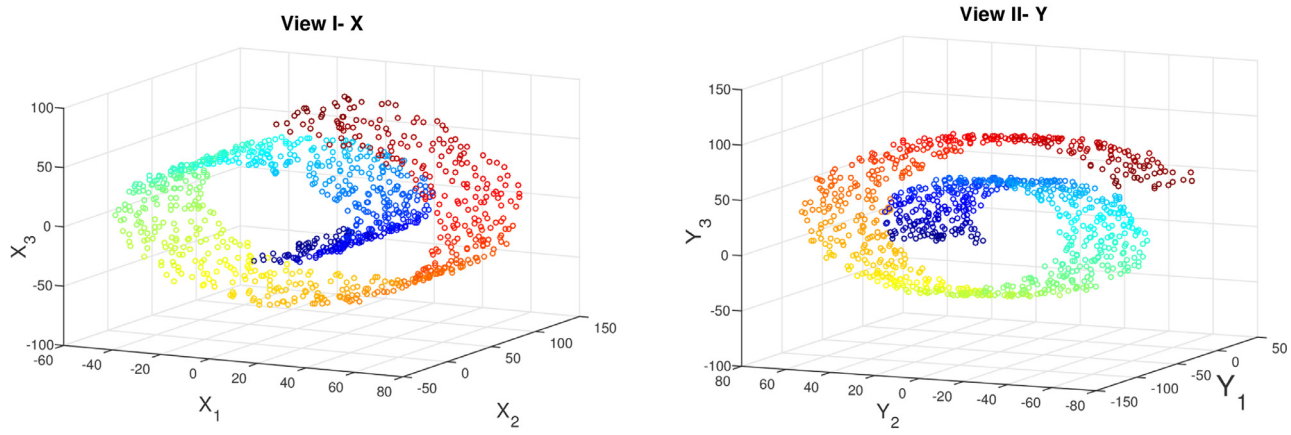

Fig. 6. The two Swiss Rolls, Left - a Swiss Roll generated by Eq. (53), Right - a Swiss Roll generated by Eq. (54).

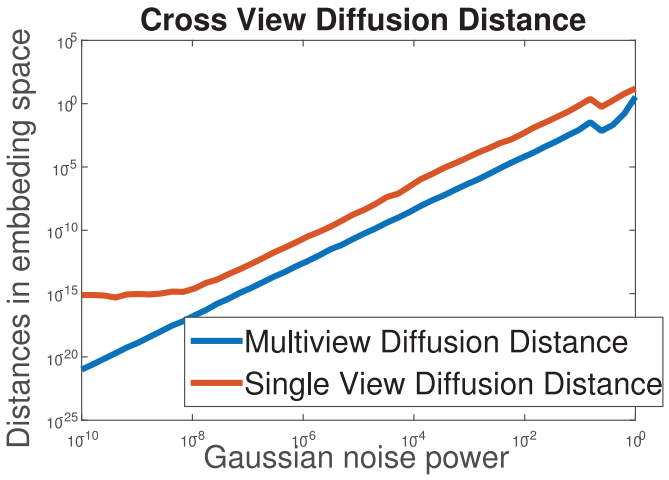

Fig. 7. Comparison between two cross view diffusion based distances. Simulated on two Swiss Rolls with additive Gaussian noise. The results are the median of 100 simulations.

where $\Psi_t(\boldsymbol{x}_i), \Psi_t(\boldsymbol{y}_i), i = 1, \ldots, M$ are the single-view diffusion mappings. Then, the proposed framework is applied to extract the coupled embedding. A Cross View Diffusion Distance (CVDD) is computed using Eq. (30). This experiment was executed 100 times for various values of the Gaussian noise variance σ_N^2.

In about 10% of the single-view trials the embeddings' axis are flipped. This generates a large SVDD although the embeddings share similar structures. In order to mitigate the effect of this type of errors we used the Median of the measures taken from the 100 trials, presented in Fig. 7.

6.2. Manifold learning

In this section we demonstrate how our proposed Multi-view DM framework allows to simultaneously extract L low-dimensional representations for L datasets.

6.2.1. Artificial manifold learning

The general DM approach is based on an underlying assumption, that the sampled space describes a single low-dimensional manifold. However, this assumption may be incorrect if the sampled space describes the existence of redundancy in the manifold, or more generally, if the sampled space can describe two or more manifolds generated by a common physical process. In this section we consider such cases. We examine the extracted embedding computed using our method and compare it to the Kernel Product approach (Section 3.1).

Helix A

Two coupled manifolds with a common underlying open circular structure are generated. The helix shaped manifolds were generated by the application of a 3-dimensional function to $M = 1{,}000$ data points $\{a_i, b_i\}_{i=1}^{M}$, such that the $\{a_i\}$ are evenly spread in $[0, 2\pi]$ and $b_i = (a_i + 0.5\pi) \mod 2\pi$, $i = 1, \ldots, M$. The following functions are used to generate the datasets for View-I and View-II denoted as $\boldsymbol{X}$ and $\boldsymbol{Y}$, resp.:

$$\text{View I: } \boldsymbol{x}_i = \begin{bmatrix} x_i[1] \\ x_i[2] \\ x_i[3] \end{bmatrix} = \begin{bmatrix} 4\cos(0.9a_i) + 0.3\cos(20a_i) \\ 4\sin(0.9a_i) + 0.3\sin(20a_i) \\ 0.1(6.3a_i^2 - a_i^3) \end{bmatrix}, i = 1, 2, \ldots, 1{,}000, \tag{56}$$

$$\text{View II: } \boldsymbol{y}_i = \begin{bmatrix} y_i[1] \\ y_i[2] \\ y_i[3] \end{bmatrix} = \begin{bmatrix} 4\cos(0.9b_i) + 0.3\cos(20b_i) \\ 4\sin(0.9b_i) + 0.3\sin(20b_i) \\ 0.1(6.3b_i - b_i^2) \end{bmatrix}, i = 1, 2, \ldots, 1{,}000. \tag{57}$$

The resulting 3-dimensional Helix-shaped manifolds $\boldsymbol{X}$ and $\boldsymbol{Y}$ are shown in Fig. 8.

The Kernel Product mapping (Eq. (6)) separates the manifold to a bow and a point as shown in Fig. 10. This structure neither represents any of the original structures nor reveals the underlying parameters a_i, b_i. On the other hand, our embedding (Eq. (21)) captures the two structures. one for each view. As shown in Fig. 9, one structure represents the angle of a_i while the other represents the angle of b_i. The Euclidean distance in the new spaces preserves the mutual relations between data points based on the geometrical relation in both views. Moreover, both manifolds are in the same coordinate system and this is a strong advantage as it enables to compare the manifolds in the lower-dimensional space. The Euclidean distance in the new spaces preserves the mutual relations between data points that are based on the geometrical structure of both views.

Helix B

The previous experiment was repeated using the following alternative functions:

$$\text{View I: } \boldsymbol{x}_i = \begin{bmatrix} x_i[1] \\ x_i[2] \\ x_i[3] \end{bmatrix} = \begin{bmatrix} 4\cos(5a_i) \\ 4\sin(5a_i) \\ 4a_i \end{bmatrix}, \tag{58}$$

$$\text{View II: } \boldsymbol{y}_i = \begin{bmatrix} y_i[1] \\ y_i[2] \\ y_i[3] \end{bmatrix} = \begin{bmatrix} 4\cos(5b_i) \\ 4\sin(5b_i) \\ 4b_i \end{bmatrix}. \tag{59}$$

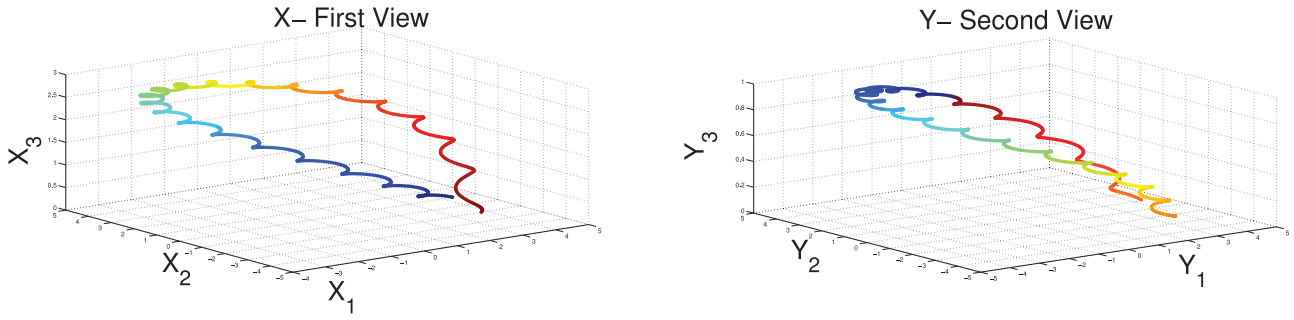

Fig. 8. Left: first Helix $\boldsymbol{X}$ (Eq. (56)). Right: second Helix $\boldsymbol{Y}$ (Eq. (57)). Both manifolds have some circular structure governed by the angle parameter $a[i]$ and $b[i]$, $i = 1, 2, \ldots, 1000$ colored by the points index i.

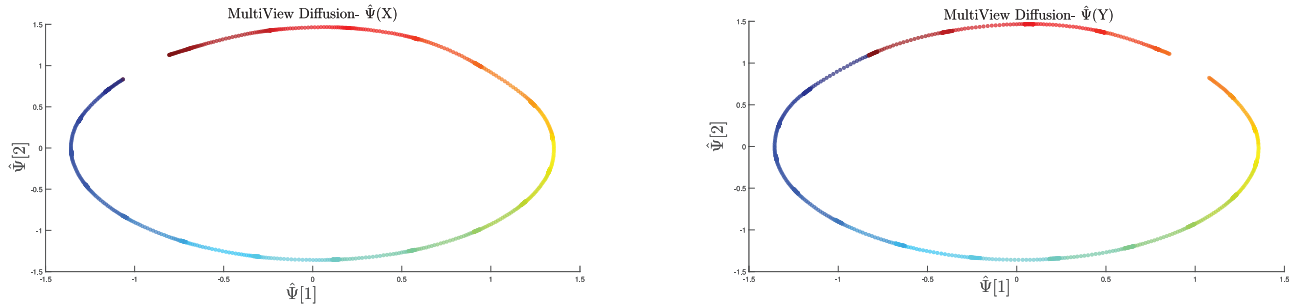

Fig. 9. Left: Multi-View based embedding of the first view $\hat{\boldsymbol{\Psi}}(\boldsymbol{X})$. Right: Multi-View based embedding of the second view $\hat{\boldsymbol{\Psi}}(\boldsymbol{Y})$. They were computed by using Eq. (21)), respectively.

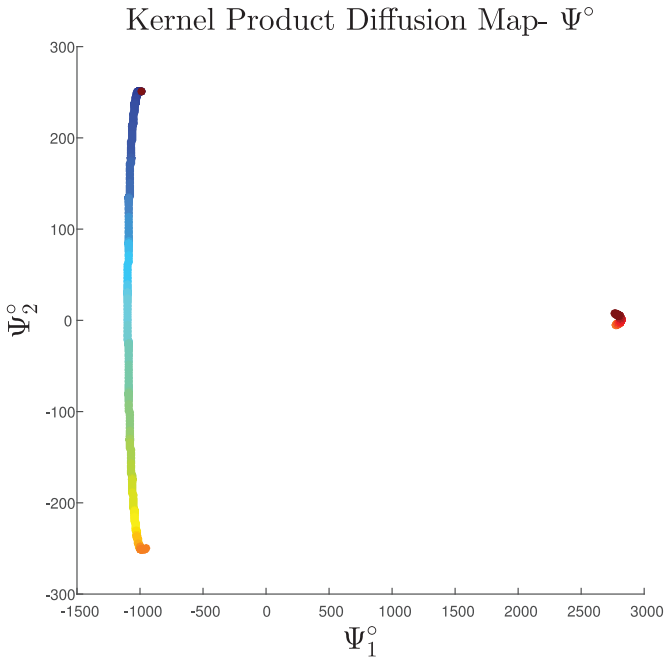

Fig. 10. 2-dimensional DM-based mapping of the Helix computed using the concatenated vector from both views that correspond to the kernel $\boldsymbol{P}^{*}$ (Eq. (6)).

Again, $M = 1000$ points were generated using $a_i \in [0,\ 2\pi]$, $b_i = (a_i + 0.5\pi) \mod 2\pi$, $i = 1, \ldots, M$. The generated manifolds are presented in Fig. 11.

As can be viewed in Fig. 12, our proposed embeddings (Eq. (21) has successfully captured the governing parameters a_i and b_i. The Kernel Product based embedding (Eq. (6)), as evident in Fig. 13, again separated the data points into two unconnected structures that do not represent well the parameters.

6.2.2. *Multiview video sequence*

Various examples involving datasets in diverse fields, such as images, audio, MRI ([11,54,55], resp.) have demonstrated the power of DM for the extraction of underlying changing physical parameters from real datasets. In this experiment, the multi-view approach is tested on a real video data, in what can literally be termed a "toy example":

Two web cameras and a toy train with preset tracks are used. The train's tracks have an "eight" shape structure. Extracting the underlying manifold from the set of images enables to organize the images according to the location along the train's path and thus reveals the true underlying parameters of the processes.

The setting of the experiment is as follows: each camera records a set of images from a different angle. A sample frame from each view is shown in Fig. 14. The video is sampled at 30 frames per second with a resolution of 640×480 pixels per frame. $M = 220$ images were collected from each view (camera). Then, the R,G,B values were averaged and downsampled to 160×120 pixels resolution. The matrices were reshaped into column vectors. The resulted set of vectors are denoted by $\boldsymbol{X}$ and $\boldsymbol{Y}$ where $\boldsymbol{x}_i, \boldsymbol{y}_i \in \mathbb{R}^{19200}$, $1 \le i \le 220$. The sequential order of the images is not important for the algorithm. In a normal setting, one view is sufficient to extract the parameters that govern the movement of the train and thus extract the natural order of the images. However, we use two types of interferences to create a scenario in which each view by itself is insufficient for the extraction of the underlying parameters. The first interference is a gap in the recording of each camera. We remove 20 consecutive frames from each view at different time intervals. By removing frames, the bijective correspondence of some of the images in the sequence is broken. However, even an approximated correspondence is sufficient for our proposed manifold extraction. A standard 2-dimensional DM based mapping of each view was extracted. The results are bow-shaped manifolds as presented in Fig. 15. Applying DM

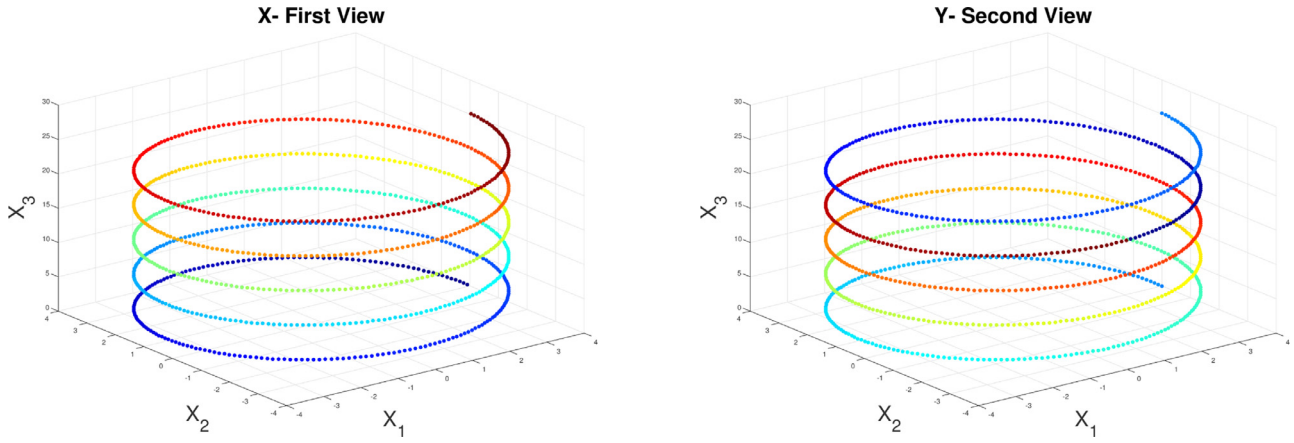

Fig. 11. Left: first Helix $\boldsymbol{X}$ (Eq. (58)). Right: second Helix $\boldsymbol{Y}$ (Eq. (59)). Both manifolds have some circular structure governed by the angle parameter $a[i]$ and b_i, $i = 1, 2, \ldots, 1000$, as colored by the point's index i.

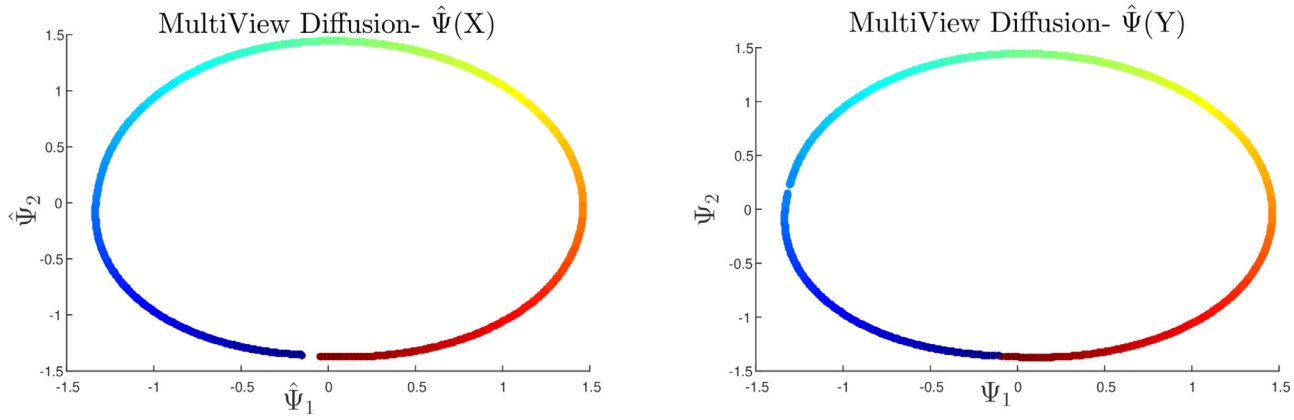

Fig. 12. The coupled mappings computed using our proposed parametrization in Eq. (21).

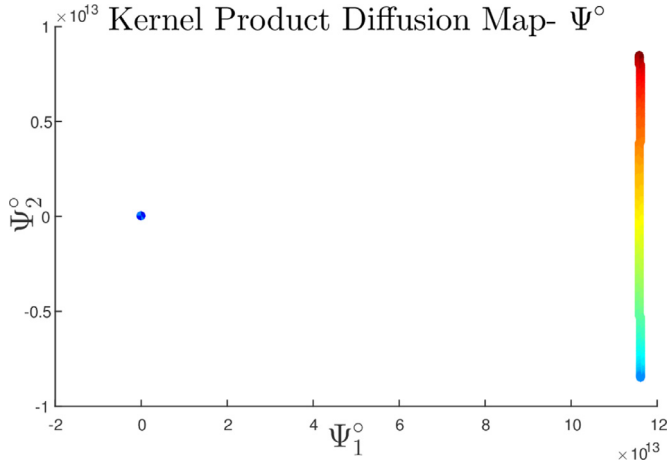

Fig. 13. A 2-dimensional mapping, extracted based on $\boldsymbol{P}^*$ (Eq. (6)).

separately to each view extracts the correct order of the data points (images) along the path. However, the "missing" data points broke the circular structure of the expected manifold and resulted in a bow-shaped embedding. We use the multi-view based methodology to overcome this interference by application of the multi-view framework to extract two coupled mappings (Eq. (21)). The results are shown in Fig. 16. The proposed approach overcomes the interferences by smoothing out the gap inherited in each view through the use of connectivities from the "unonbstructed" view. Finally, we concatenate the vectors from both views and compute the Kernel Product embedding The results are presented in Fig. 17. Again, the structure of the manifold is distorted and incomplete due to the missing images.

This experiment was then repeated, replacing 10 frames from each view with "noise frames" consisting of pixel-wise i.i.d. zero-mean Gaussian noise with variance 10. A single-view DM-based mapping was computed. The Kernel Product-based DM and the multi-view based DM mappings were computed as well. As presented in Fig. 18, the Gaussian noise distorted the manifolds extracted in each view. The multi-view approach extracted two circular structures presented in Fig. 19. Again, the data points are ordered according to the position along the path. This time, the circular structure is unfolded and the gaps are visible in both embeddings. Applying the Kernel Product approach (Eq. 6) has yielded a distorted manifold as presented in Fig. 20.

7. Applications

7.1. Multi-view clustering

The task of clustering has been in the core of machine learning for many years. The goal is to divide a given dataset into subsets based on the inherited structure of the data. We use the multi-view construction to extract low-dimensional mappings from multiple sets of

Fig. 14. Left: a sample image from the first camera ($\boldsymbol{X}$). Right: a sample image from the second camera ($\boldsymbol{Y}$).

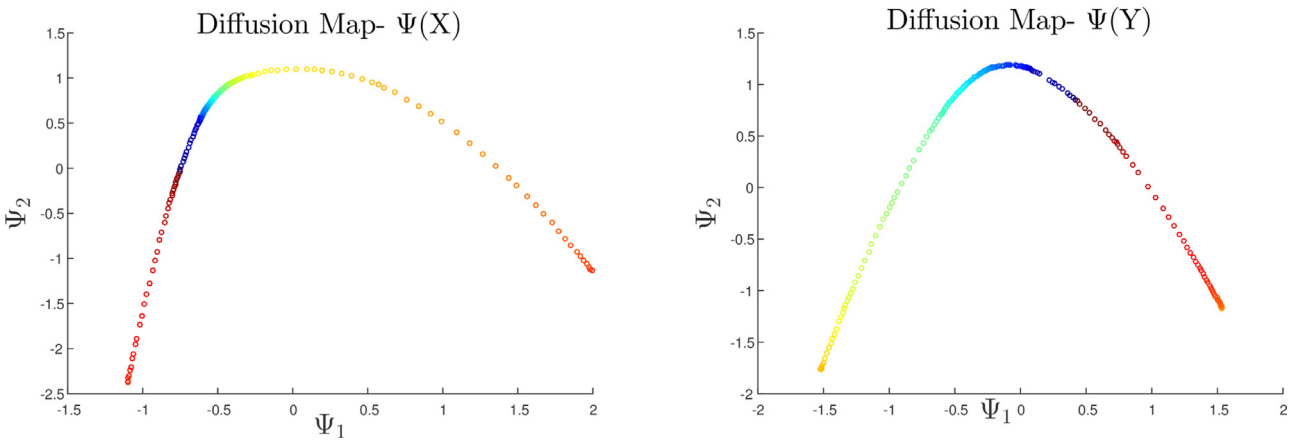

Fig. 15. Left: DM-based single-view mapping $\Psi(\boldsymbol{X})$. Right: DM-based single-view mapping $\Psi(\boldsymbol{Y})$). The removed images caused a bow shaped structure.

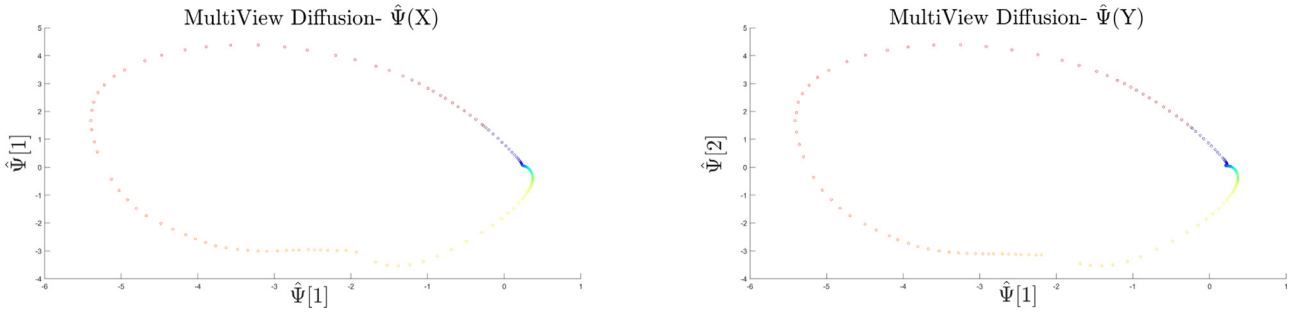

Fig. 16. Left: Mapping $\hat{\Psi}(\boldsymbol{X})$. Right: Mapping $\hat{\Psi}(\boldsymbol{Y})$ as extracted by the multi-view based framework. Two small gaps, which correspond to the removed images, are visible.

high-dimensional data points. In the following experiments we expand the examples presented in [56] to cluster artificial and real data sets. For the real data sets applying the multi-view approach requires an eigen decomposition of large matrices. To reduce the runtime of experiments we use an approximate matrix decomposition based on sparse random projections [35].

7.1.1. Two circles clustering

Spectral properties of data sets are useful for clustering since they reveal information about the unknown number of clusters. The characteristic of the eigenvalues of $\widehat{\boldsymbol{P}}$ (Eq. (14)) can provide insight into the number of clusters within the data set. The study in [57] relates the number of clusters to the multiplicity of the eigenvalue 1. A different approach in [58] provides an analysis about the relation between the eigenvalue drop to the number of clusters. In this section, we evaluate how our proposed method captures the clusters' structure when two views are available.

We generate two circles that represent the original clusters using the function

$$\boldsymbol{z}_i = \begin{bmatrix} z_i[1] \\ z_i[2] \end{bmatrix} = \begin{bmatrix} r \cdot \cos(\theta_i)) \\ r \cdot \sin(\theta_i) \end{bmatrix}, \tag{60}$$

where $M = 1{,}600$ points θ_i, $1 \le i \le M$, are evenly spread in $[0, 4\pi]$. The clusters are created by changing the radius as follows:

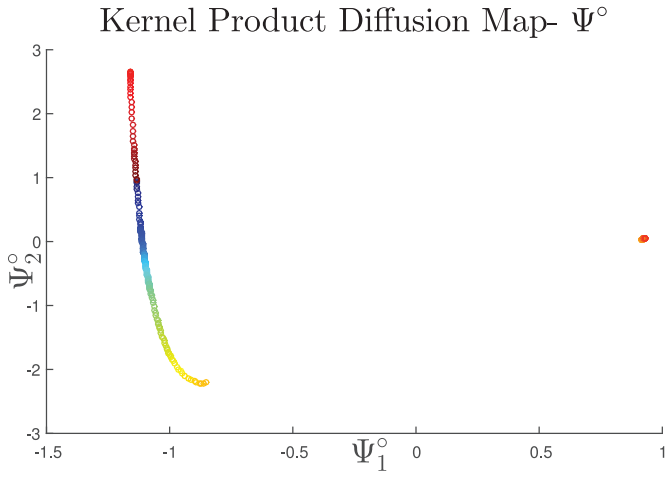

Fig. 17. A standard diffusion mapping (Kernel Product-based) that was computed by using the concatenated vector from both views that correspond to kernel $\boldsymbol{K}^{\circ}$.

$r = 2, 1 \le i \le 800$ (first cluster), $r = 4{,}801 \le i \le 1{,}600$ (second cluster). The views $\boldsymbol{X}$ (Eq. (61)) and $\boldsymbol{Y}$ (Eq. (62)) are generated by the application of the following non-linear functions that produce the distorted views

$$x_i[1] = \left\{ \begin{array}{c} z_1[i] + 1 + n_i[2] | z_i[2] \ge 0 \\ z_1[i] + n_i[3] | z_i[2] < 0 \end{array} \right\}, x_i[2] = z_i[2] + n_i[1] \tag{61}$$

and

$$y_i[1] = z_i[1] + n_i[4], y_i[2] = \left\{ \begin{array}{c} z_i[2] + 1 + n_i[6] | z_i[1] \ge 0 \\ z_i[2] + n_i[6] | z_i[1] < 0 \end{array} \right\}, \tag{62}$$

where $n_i[\ell], 1 \le \ell \le 6$, are i.i.d. random variables drawn from a Gaussian distribution with $\mu = 0$ and $\sigma_n^2 \in [0.03, 0.6]$. This data is referred to as the Coupled Circles dataset.

In Fig. 21, the views $\boldsymbol{X}$ and $\boldsymbol{Y}$, which were generated by Eqs. (61) and (62), are shown. Color and shape indicate the ground truth clusters. Initially, DM is applied to each view and clustering is applied using K-means ($K = 2$) within the first diffusion coordinate. The kernel bandwidths σ_x and σ_y for all methods are set using the min-max method described in Eq. (24). We use $t = 1$ since it is optimal for clustering tasks. For the kernel product method we use $\sigma_\circ = \sqrt{\sigma_x^2 + \sigma_y^2}$. We further extract a 1-dimensional representation using the proposed multi-view framework (Eq. (21)), the Kernel Sum DM (Eq. (7)), Kernel Product DM (Eq. (6)), de Sa's approach (Eq. (10)) and Kernel CCA (Eq. (8)) described in Section 3. The regularization parameter is $\gamma = 0.01$ for KCCA and we use 100 components for the Incomplete Cholesky Decomposition [15,16]. Clustering is performed in the representation space by the application of K-means where $K = 2$. To evaluate the performance of our proposed map 100 simulations with various values of the Gaussian's noise variance (all with zero mean) were performed. The average clustering success rate is presented in Fig. 22. It is evident that the multi-view based approach outperforms the DM-based single-view and the Kernel Product approaches.

Table 1
Accuracy of clustering and the normalized mutual information (NMI) on the infinity MNIST dataset. Each view consists a noisy version based on $40K$ images of handwritten 2's and 3's. The first view $\boldsymbol{X}^1$ is generated by adding Gaussian noise, while $\boldsymbol{X}^2$ is generated by zeroing out random pixels with probability 0.5.

Method	NMI	Accuracy
DM $\boldsymbol{X}^1$	0.38	84.4%
DM $\boldsymbol{X}^2$	0.38	84.1%
Kernel Prod	0.41	85.2%
Kernel Sum	0.39	84.6%
Kernel CCA	0.54	90.5%
de Sa	0.59	91.2%
Multiview	**0.70**	**94.7%**

The performance of kernel methods is highly dependent on setting an appropriate kernel bandwidth σ_x, σ_y, in Algorithm 1 we have presented a method for setting such parameters. To evaluate the influence of these parameters on the clustering quality we set $\sigma_n = 0.16$ and extract the multi-view, Kernel Sum and Kernel Product diffusion mapping for various values of σ_x, σ_y. The average clustering performance using K-means ($K = 2$) are presented in Fig. 23.

7.1.2. *Handwritten digits*

For the following clustering experiment, we use the Multiple Features database [2] from the UCI repository. The data set consists of 2, 000 handwritten digits from 0 to 9 that are equally spread. The extracted features from these images are the profile correlations (FAC), Karhunen-Loéve coefficients (KAR), Zerkine moment (ZER), morphological (MOR), pixel averages in 2×3 windows and the Fourier coefficients (Fou) as our feature spaces $\boldsymbol{X}^1, \boldsymbol{X}^2, \boldsymbol{X}^3, \boldsymbol{X}^4, \boldsymbol{X}^5, \boldsymbol{X}^6$ respectively. We apply dimensionality reduction using a single-view DM, Kernel Product DM, Kernel Sum DM and the proposed Multi-view. We apply K-means to the re-

[2] http://archive.ics.uci.edu/ml.

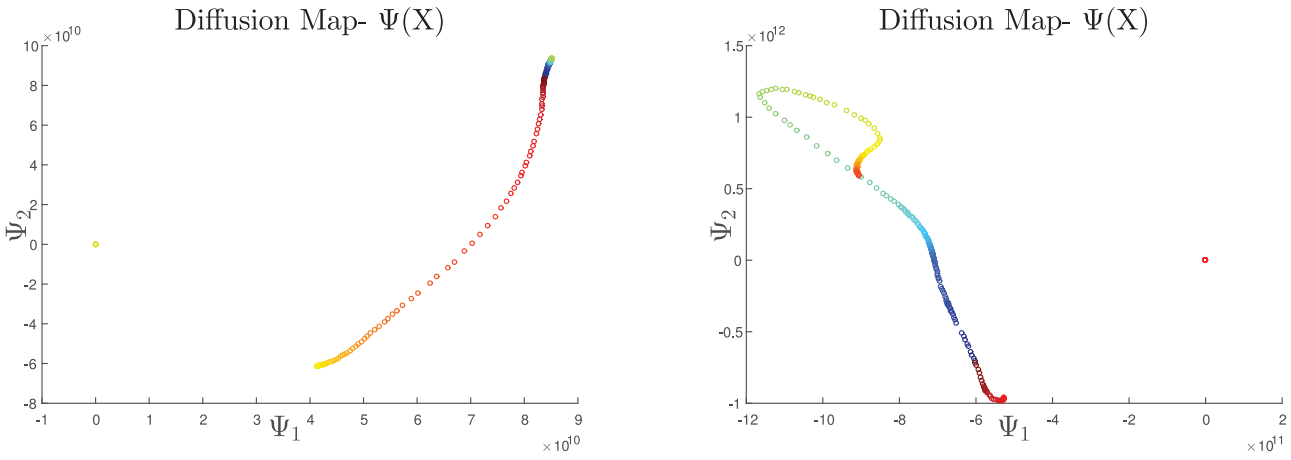

Fig. 18. Left: DM-based single-view mapping $\Psi(\boldsymbol{X})$. Right: DM-based single-view mapping $\Psi(\boldsymbol{Y})$). The Gaussian noise deformed the circular structure.

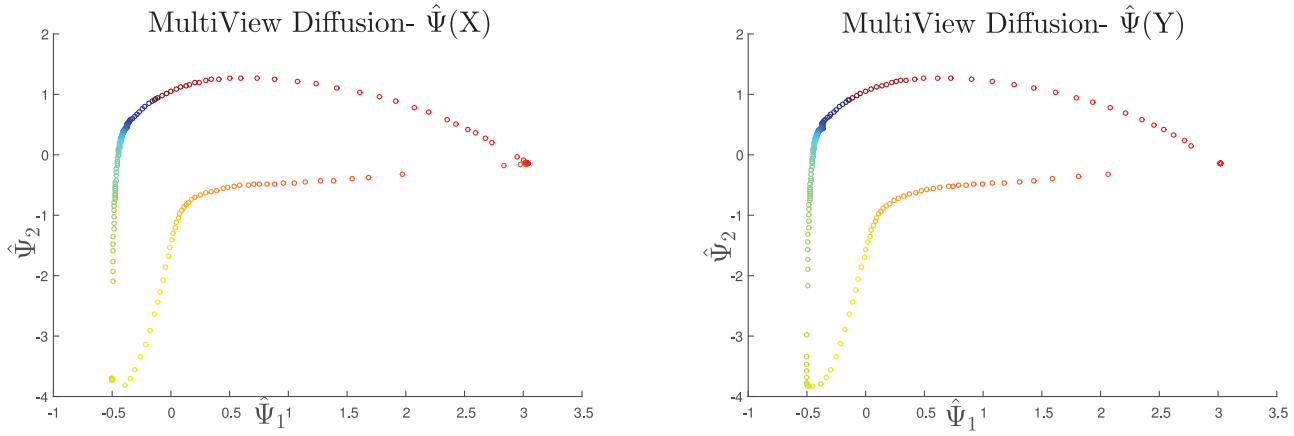

Fig. 19. Left: Mapping $\hat{\Psi}(\boldsymbol{X})$. Right: Mapping $\hat{\Psi}(\boldsymbol{Y})$ as extracted by the multi-view framework. Two gaps are visible that correspond to Gaussian noise.

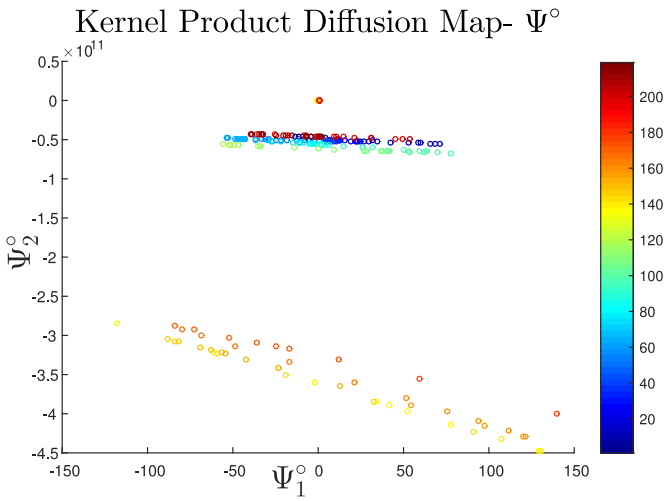

Fig. 20. Computation of a standard diffusion mapping (Kernel Product) by using the concatenation vector from both views (corresponding to kernel $\boldsymbol{P}^{*}$ Eq. (6)).

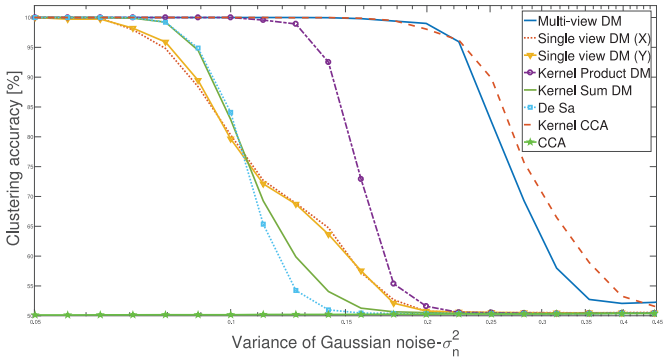

Fig. 22. Clustering results from averaging 200 trials vs. the variance of the Gaussian noise. The simulation performed on the Coupled Circles data (Eqs. (60)–(62)).

duced mapping using 6 to 20 coordinates. The clustering performance is measured using the Normalized Mutual Information [59] (NMI). Fig. 24 presents the average clustering results using K-Means.

Next, we attempt to cluster 40K grey scale images of handwritten 2's and 3's. The images with dimension 28×28 were collected from the infinity MNIST dataset [60]. We generate two independent noisy versions of the 40K samples. By adding pixel Gaussian noise $N(0, 0.5)$ to each image we create the first noisy view which is denoted as $\boldsymbol{X}^1$. The second view $\boldsymbol{X}^2$ is created by randomly and independently zeroing each pixel with probability 0.5. In Fig. 25. we present 25 examples from each view. To evaluate the clustering performance, we apply K-means 100 times to the reduced representations with dimensions 5,10,15 and 20 and report the top results for each method. In Table 1 we present the top Normalized Mutual Information (NMI) and clustering accuracy's the proposed multi-view and various alternative methods.

7.1.3. Isolet data set

The Isolet data set was constructed by recording 150 people pronouncing each letter twice for all 26 letters. The feature vector available is a concatenation of the following features: spectral coefficients, contour, sonorant, pre-sonorant and post-sonorant. The authors do not provide the feature's separation, therefore, the dimension of the feature vector is 617. We use a subset of the data with 1599 instances, thus the features space is $\boldsymbol{X}\in\mathbb{R}^{1559\times 617}$. To apply the multi-view approach we compute 3 different kernels and fuse them together. The first kernel $\boldsymbol{K}^1$

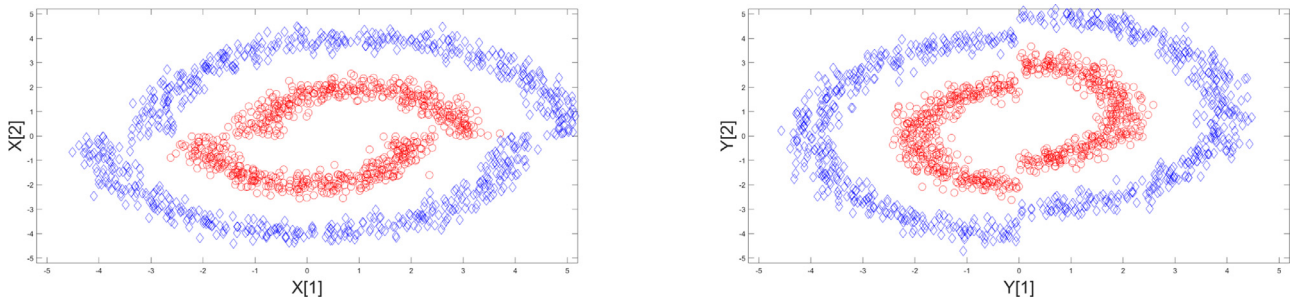

Fig. 21. Left: first view $\boldsymbol{X}$. Right: second view $\boldsymbol{Y}$. The ground truth clusters are represented by the marker's shape and color.

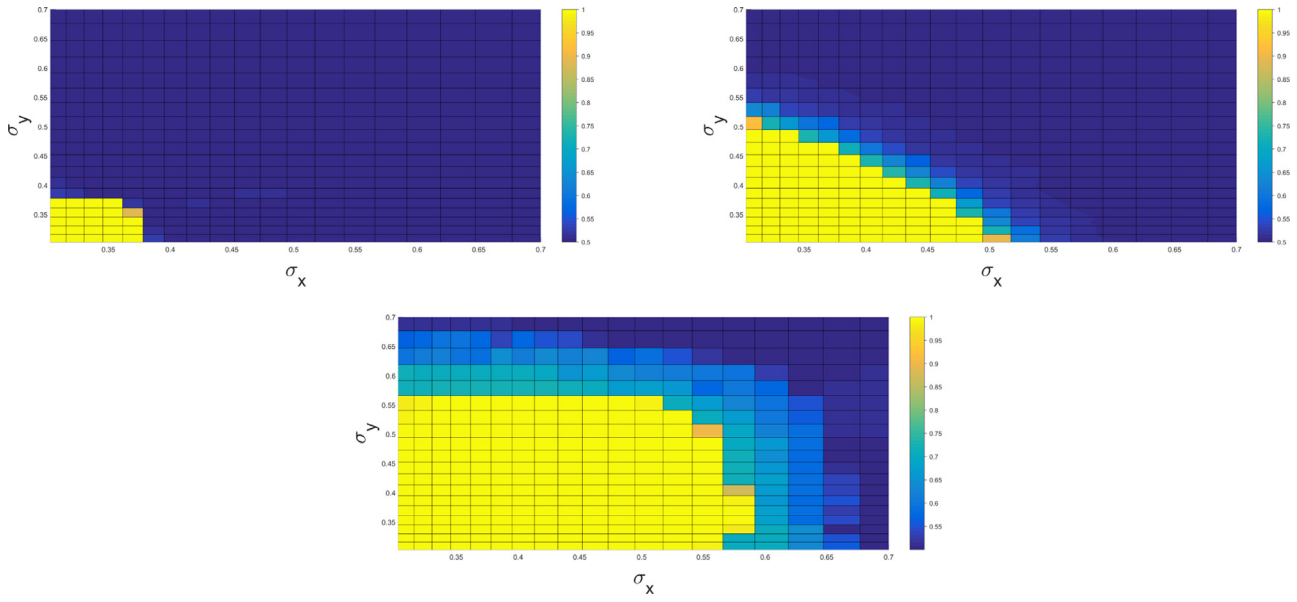

Fig. 23. Clustering results from averaging 20 trails using various values of σ_x, σ_y based on different mappings. The standard deviation of the noise is $\sigma_n = 0.16$. Top left- Kernel Sum DM, top right- Kernel Product DM, bottom- Multi View DM.

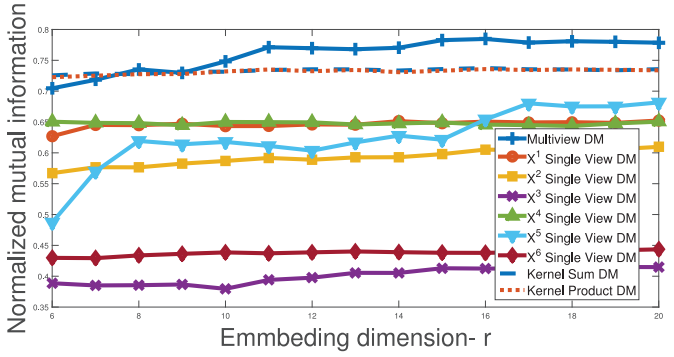

Fig. 24. Average clustering accuracy running 100 simulations on the Handwritten data set. Accuracy is measured using the Normalized Mutual Information (NMI).

is the standard Gaussian kernel defined in Eq. (12). $\boldsymbol{K}^2$ is a Laplacian kernel defined by

$$K^2_{i,j} \triangleq \exp\left(\frac{-|\boldsymbol{x}_i - \boldsymbol{x}_j|}{\sigma_2}\right). \tag{63}$$

The third kernel $\boldsymbol{K}^3$ is an exponent with a correlation distance as the affinity measure, given by

$$K^3_{i,j} \triangleq \exp\left(\frac{T_{i,j} - 1}{2\sigma^2}\right), i, j = 1, \ldots, M, \tag{64}$$

where $T_{i,j}$ is the correlation coefficient between the ith and jth feature vectors, computed by

$$T_{i,j} \triangleq \frac{\tilde{\boldsymbol{x}}_i{}^T \cdot \tilde{\boldsymbol{x}}_j}{\sqrt{(\tilde{\boldsymbol{x}}_i{}^T \cdot \tilde{\boldsymbol{x}}_i)(\tilde{\boldsymbol{x}}_j{}^T \cdot \tilde{\boldsymbol{x}}_j)}}, i, j = 1, \ldots, M. \tag{65}$$

The average subtracted features are $\tilde{\boldsymbol{x}}_i \triangleq \boldsymbol{x}_i - \eta_i \cdot \mathbf{1}$, where η_i is the average of the features for instance i. We fuse the kernels using multi-view, kernel product and kernel sum approach, we then apply K-Means to the extracted space. The average NMI for 26 classes is presented in Fig. 26.

Table 2
Clustering results on two subsets of Caltech-101 data set. Columns represent measures for the quality of clustering. First two rows are scores based on a single-view DM using different features. The last four rows present scores based on the proposed and alternative schemes for a Multiview DM based mapping.

Method	Silhouette(10)	Calinski(10)	Silhouette(15)	Calinski(15)
DM $\boldsymbol{X}^1$ (SIFT)	−0.27	183	−0.27	293
DM $\boldsymbol{X}^2$ (PHOG)	−0.23	169	−0.19	329
Kernel Prod	−0.08	348	−0.02	683
Kernel Sum	−0.11	322	−0.05	650
de Sa	−0.35	134	−0.34	227
Multiview	**0.32**	**727**	**0.32**	**1275**

7.1.4. Caltech 101

For this experiment we use an image dataset which consists of 101 categories collected in [61] for an object recognition problem. We use two subsets of the dataset, with 10 and 15 instances in each category. The views $\boldsymbol{X}^1$, $\boldsymbol{X}^2$ are represented by the following features: Bag-of-words SIFT descriptors [62] and Pyramid Histogram of Oriented Gradients (PHOG) [63]. We use Kernel matrices computed by [64], therefore, we do no set the scale parameters. The quality of the clustering is again measured using Calinski-Harabasz Criterion [65] and the average silhouette width [66], for both metrics higher values indicates better separation. The silhouette is defined within the range of $[-1, 1]$. The average clustering results based on 10 to 15 coordinates are presented in Table 2.

7.2. Learning from multi sensor seismic data

Automatic detection and identification of seismic events is an important task, it is carried out constantly for seismic and nuclear monitoring. The monitoring process results in a seismic event bulletin that contains information about the detected events, their locations, magnitudes and type (natural or man made event). Seismic stations usually consist of multiple sensors recording continuously at a low frequency. The amount of available data is huge and only a fraction of the recordings contains the signal of interest. Thus, automatic tools for monitoring are of great interest. A suspect event is usually identified based on the energy of the signal, then, typical discrimination algorithms extract seismic parameters. A simple seismic parameter is the focal depth. Its drawback is

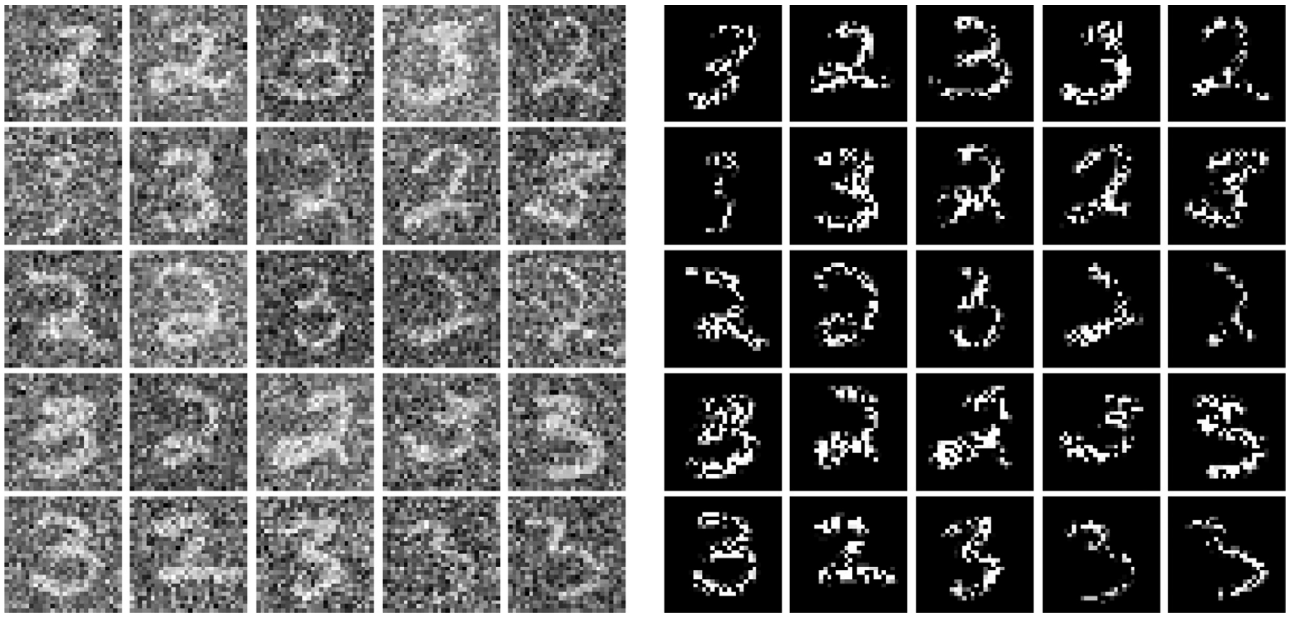

Fig. 25. Random samples from both views. Left- $\boldsymbol{X}^1$ generated by adding Gaussian noise. Right- $\boldsymbol{X}^2$ generated by randomly dropping out 50% of the pixels.

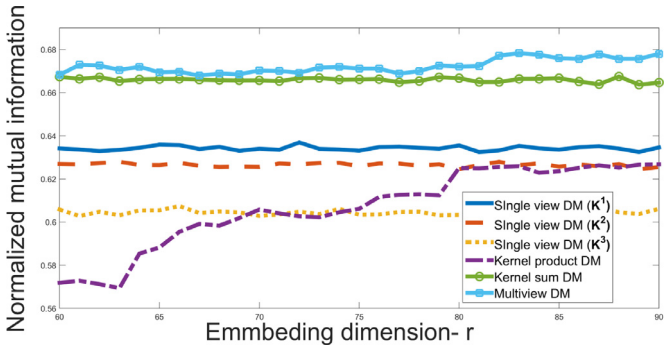

Fig. 26. Clustering accuracy measured with Normalized Mutual Information (NMI) on the Isolet data set by using 3 different kernel matrices. Clustering was performed in the r dimensional embedding space.

that its estimation is usually inaccurate without the depth phases. Other widely used seismic discrimination methods are Ms:mb (surface wave magnitude versus body wave magnitude) and spectral amplitude ratios of different seismic phases [67,68]. Current automatic seismic bulletins comprise a large number of false alarms, which have to be manually corrected by an analyst.

In this section we apply the proposed method to extract essential latent seismic parameters. A suspected event is identified based on a short and long time average ratio (STA/LTA) [69]. Then, a time-frequency representation is computed to which multi-view is applied to fuse the data from multiple seismic sensors. Using the multi-view low-dimensional embedding and simple classifiers, we demonstrate capabilities classification of event type (earthquakes vs. explosions) and quarry source for explosions.

7.2.1. Description of the data set

The dataset consists of recordings from two different broad band seismic stations MMLI (Malkishua) and HRFI (Harif). Both stations are operated by the Geophysical Institute of Israel (GII) and they are part of the Israel National Seismic Network [70]. MMLI and HRFI stations are located to the north and to the south of the analyzed region, respectively. Each station is equipped with a three component STS-2 seismometer, thus the total number of views is $L = 6$. All recording are sampled at $F_s = 40$ Hz. The HRFI data includes 1654 explosions and 105 earthquakes, while MMLI includes a subset of 46 earthquakes, 62 explosions. The explosions occurred in the south of Israel between the years 2005 – 2015.

7.2.2. Feature extraction by normalized sonograms

A seismic event typically generates two underground traveling waves. A primary wave (P) and secondary wave (S). The two waves (P-S) arrive at the recording station with some time delay. A time frequency representation of the recording captures the spectral properties of the event while maintaining the P-S time gap. Here we use a time frequency representation termed sonogram [71] with some modifications. Each single-trace seismic waveform, denoted by $y[n] \in \mathbb{R}^N$ is a time series signal sampled at the rate of $F_s = 40$Hz. The waveform $y[n]$ is decomposed into a set of overlapping windows of length $N_0 = 256$ using an overlap of $s = 0.9$. Thus, the sift between consecutive windows is $N_S = \lfloor 0.1 \cdot 256 \rfloor = 25$. A short-time Fourier transform (STFT) is applied to $y[n]$ in each time window. Then, power spectral densities are computed. The resulting spectrogram is denoted by $R(f, t)$, where f is a frequency bin and t is a number of time window (bin). Thus, it contains $T = 192$ time bins and $N_0/2 = 128$ frequency bins.

The sonogram is obtained by summing the spectrogram in equally tempered logarithmically scaled frequency bands, this is done for every time bin. Finally the sonogram is normalized such that the sum of energy in every frequency band is equal to 1. The result is a normalized sonogram and it is denoted by $S(k, t)$, where k is the frequency band number and t is the time window number. The resulting set of sonograms are denoted $\boldsymbol{X}^1$, $\boldsymbol{X}^2$, $\boldsymbol{X}^3$, $\boldsymbol{X}^4$, $\boldsymbol{X}^5$, $\boldsymbol{X}^6$. These are the input views for our framework.

7.2.3. Event classification

Each sensor records information from the seismic event as well as nuisance noise. We can assume that the noise at each station is independent. Thus, by fusing the measurements from different sensors we may be able to improve detection level. To evaluate how well the proposed approach fuses the information, we use a set which includes 46 earthquakes, 62 explosions recorded at MMLI (Malkishua) and HRFI (Harif). After we extract the sonogram, the proposed MVDM framework is applied, as well as the single-view DM, kernel product DM and kernel sum DM. Classification is performed by using K-NN (K=1), based on 3 or

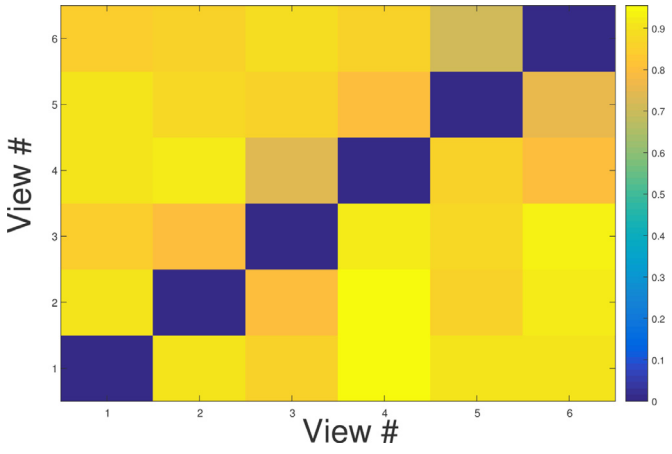

Fig. 27. Classification accuracy using K-nn (K=1) for all pairs of views $\boldsymbol{X}^l$, $\boldsymbol{X}^m$, $l \neq m$. The y-axis is the number of the first view used, while the x-axis is the number of the second view. Classification is performed in the multi-view low-dimensional embedding ($r = 4$). The diagonal terms are presented as zero since we did not simulate for $l = m$.

Table 3
Classification accuracy using 1-fold cross validation. r is the number of coordinates used in the embedding space.

Method	Accuracy [%] ($r = 3$)	Accuracy [%] ($r = 4$)
single-view DM ($\boldsymbol{X}^1$)	89.9	93.0
single-view DM ($\boldsymbol{X}^2$)	88.6	92.4
single-view DM ($\boldsymbol{X}^3$)	89.3	91.1
single-view DM ($\boldsymbol{X}^4$)	89.3	89.2
single-view DM ($\boldsymbol{X}^5$)	89.3	90.5
single-view DM ($\boldsymbol{X}^6$)	88.6	91.1
Kernel Sum DM	93.7	94.9
Kernel Product DM	94.3	93.0
Multi-view DM	97.5	98.1

4 coordinates from the reduced mapping. The results are presented in Table 3. This experiment demonstrates that applying multi-view DM to seismic recordings extracts a meaningful representation.

In the following test we check how each view affects the detection rate. We do this by applying the MV to subsets of the $L = 6$ views. We perform classification using representation computed based on all pairs of view $\boldsymbol{X}^l, \boldsymbol{X}^m, l, m = 1, \ldots, 6, l \neq m$. The accuracy of classification based on the multi-view representation $\hat{\boldsymbol{\Psi}}(\boldsymbol{X}^l), l = 1, \ldots, 6$ given that $\boldsymbol{X}^m, m = 1, \ldots, 6, m \neq l$ is presented in Fig. 27.

Table 4
Classification accuracy using 1-fold cross validation. r is the number of coordinates used in the embedding space.

Method	Accuracy [%] ($r = 3$)	Accuracy [%] ($r = 4$)
single-view DM ($\boldsymbol{X}^1$)	80.3	80.6
single-view DM ($\boldsymbol{X}^2$)	79.1	79.2
single-view DM ($\boldsymbol{X}^3$)	76.4	77.8
Kernel Sum DM	82.2	82.8
Kernel Product DM	80.8	81.2
Multi-view DM	86.2	86.4

Identification and separation of quarries by attributing the explosions to the known sources is a challenging task [72,73]. Quarry blast have a similar spectral properties, they are usually classified by a triangulation process. Such a process requires to compare the arrival time between distinct seismic stations. Here we attempt to classify the source of the explosions, using 3-channels from the same station.

For this experiment 602 seismograms of explosions are used. The explosions occurred in 4 quarry clusters in Israel and 1 quarry in Jordan. All events were recorded in HRFI station, the distances from the event to the station vary between 50and 130 Km. The association of each blast to quarry (labeling) is performed manually by an analyst from the GII. After extracting the sonograms, we apply multi-view DM and present the first 2 coordinates in Fig. 28. In Table 4 we summarize the classification results by applying K-NN ($k = 1$) in a leave one out procedure. Our method is compared to a single-view DM, kernel product and kernel sum.

8. Discussion

We presented a multi-view based framework for dimensionality reduction. The framework enables to extract simultaneous embeddings from coupled embeddings. Our approach is based on imposing an implied cross-domain transition in each single time-step. The transition probabilities depend on the connectivities in both views. We reviewed various theoretical aspects of the proposed method and demonstrated their applicability to both synthetic and real data. The experimental results demonstrate the strength of the proposed framework in cases where data is missing in each view or each of the manifolds is deformed by an unknown function. The framework is applicable to various real life machine learning tasks that consist of multiple views or multiple modalities.

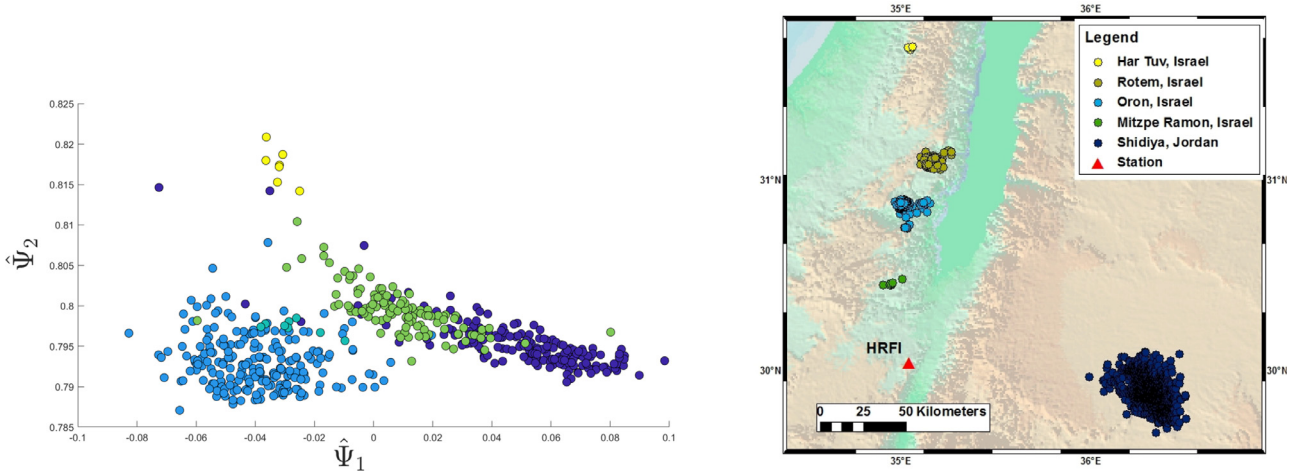

Fig. 28. Left- 2 dimensional multi-view DM of the 602 quarry blasts. Points are colored by quarry label. Right- a map with the approximated source location.

Appendix

We prove Theorem 7 (Section 5.6), repeated here for convenience:

Theorem 7. *The infinitesimal generator induced by the proposed kernel matrix $\widehat{\boldsymbol{K}}$ (Eq. (13)) after row-normalization, denoted in here as $\widehat{\boldsymbol{P}}$, converges when $M\to\infty$, $\epsilon\to 0$ (with $\epsilon=\sigma_x^2=\sigma_y^2$) to a "cross domain Laplacian operator". The functions $f(\boldsymbol{x})$ and $g(\boldsymbol{y})$ converge to eigenfunctions of $\widehat{\boldsymbol{P}}$. These functions are the solutions of the following diffusion-like equations:*

$$(\widehat{\boldsymbol{P}}f)(\boldsymbol{x}_i)=g(\boldsymbol{\beta}(\boldsymbol{x}_i))+\epsilon\triangle\gamma(\boldsymbol{\beta}(\boldsymbol{x}_i))/\alpha(\boldsymbol{\beta}(\boldsymbol{x}_i))+\mathcal{O}(\epsilon^{3/2}), \tag{66}$$

$$(\widehat{\boldsymbol{P}}g)(\boldsymbol{y}_i)=f(\boldsymbol{\beta}^{-1}(\boldsymbol{y}_i))+\epsilon\triangle\eta(\boldsymbol{\beta}^{-1}(\boldsymbol{y}_i))/\alpha(\boldsymbol{\beta}(\boldsymbol{y}_i)+\mathcal{O}(\epsilon^{3/2}), \tag{67}$$

where the functions γ, η are defined as $\gamma(\boldsymbol{z})\triangleq g(\boldsymbol{z})\alpha(\boldsymbol{z})$, $\eta(\boldsymbol{z})\triangleq f(\boldsymbol{z})\alpha(\boldsymbol{z})$.

Proof. By extending the single-view construction presented in [5], the eigenfunction of the limit-operator $\widehat{\boldsymbol{P}}$ is defined using the functions $\boldsymbol{f(x)}$ and $\boldsymbol{g(y)}$ by concatenating the vectors such that

$$\boldsymbol{h}=[f(\boldsymbol{x}_1),f(\boldsymbol{x}_2),\ldots,f(\boldsymbol{x}_M),g(\boldsymbol{y}_1),g(\boldsymbol{y}_2),\ldots,g(\boldsymbol{y}_M)]\in\mathbb{R}^{2M}.$$

The limit of the top-half (Eq. (66)) of the characteristic equation is given by

$$\begin{aligned}\lim_{\substack{M\to\infty\\ \epsilon\to 0}}(\widehat{P}h_i)&=\lim_{\substack{M\to\infty\\ \epsilon\to 0}}h_i-\frac{\sum_{j=1}^{2M}\widehat{K}_{i,j}h_j}{\sum_{j=1}^{2M}\widehat{K}_{i,j}}\\&=\lim_{\substack{M\to\infty\\ \epsilon\to 0}}f(\boldsymbol{x}_i)-\frac{\sum_{j=1}^{M}\sum_{\ell=1}^{M}K^x_{i,\ell}K^y_{\ell,j}g(\boldsymbol{y}_j)}{\sum_{j=1}^{M}\sum_{\ell=1}^{M}K^x_{i,\ell}K^y_{\ell,j}},\ i=1,\ldots,M.\end{aligned} \tag{68}$$

We approximate the summations using a Riemann integral. Beginning with the denominator, we have

$$\frac{1}{M^2\epsilon^d}\sum_{j=1}^{M}\sum_{\ell=1}^{M}K^x_{i,\ell}K^y_{\ell,j}\underset{\substack{M\to\infty\\ \epsilon\to 0}}{\longrightarrow}D(\boldsymbol{x})\triangleq\frac{1}{\epsilon^d}\int_{\mathbb{R}^d}\int_{\mathbb{R}^d}K\Big(\frac{\boldsymbol{s}-\boldsymbol{x}}{\sqrt{\epsilon}}\Big)K\Big(\frac{\boldsymbol{y}-\boldsymbol{\beta}(s)}{\sqrt{\epsilon}}\Big)\alpha(\boldsymbol{y})\mathrm{d}\boldsymbol{s}\mathrm{d}\boldsymbol{y}.$$

Using a change of variables $\boldsymbol{z}=\frac{\boldsymbol{y}-\boldsymbol{\beta}(s)}{\sqrt{\epsilon}}$, $\boldsymbol{y}=\boldsymbol{\beta}(\boldsymbol{s})+\sqrt{\epsilon}\boldsymbol{z}$, $\mathrm{d}\boldsymbol{z}=\mathrm{d}\boldsymbol{y}\epsilon^{d/2}$ we get

$$D(\boldsymbol{x})=\frac{1}{\epsilon^{d/2}}\int_{\mathbb{R}^d}\int_{\mathbb{R}^d}K\Big(\frac{\boldsymbol{s}-\boldsymbol{x}}{\sqrt{\epsilon}}\Big)K(\boldsymbol{z})\alpha(\boldsymbol{\beta}(\boldsymbol{s})+\sqrt{\epsilon}\boldsymbol{z})\mathrm{d}\boldsymbol{s}\mathrm{d}\boldsymbol{z}.$$

Using a first order Taylor expansion of $\alpha(\cdot)$ we get

$$D(\boldsymbol{x})\approx\frac{1}{\epsilon^{d/2}}\int_{\mathbb{R}^d}\int_{\mathbb{R}^d}K\Big(\frac{\boldsymbol{s}-\boldsymbol{x}}{\sqrt{\epsilon}}\Big)K(\boldsymbol{z})\left[\alpha(\boldsymbol{\beta}(\boldsymbol{s}))+\frac{\sqrt{\epsilon}}{2}\boldsymbol{z}^{\mathrm{T}}\nabla\alpha(\boldsymbol{\beta}(\boldsymbol{s}))+\mathcal{O}(\epsilon)\right]\mathrm{d}\boldsymbol{s}\mathrm{d}\boldsymbol{z},$$

using the symmetry of the kernel $\boldsymbol{K(z)}$ we have

$$\int_{\mathbb{R}^d}K(\boldsymbol{z})\boldsymbol{z}^{\mathrm{T}}\mathrm{d}\boldsymbol{z}=\boldsymbol{0}^{\mathrm{T}},$$

therefore, applying another change of variables $\boldsymbol{t}=\frac{\boldsymbol{s}-\boldsymbol{x}}{\sqrt{\epsilon}}$, $\boldsymbol{s}=\sqrt{\epsilon}\boldsymbol{t}+\boldsymbol{x}$, $\mathrm{d}\boldsymbol{t}=\mathrm{d}\boldsymbol{s}\epsilon^{d/2}$ we get

$$D(\boldsymbol{x})\approx\int_{\mathbb{R}^d}K(\boldsymbol{t})[\alpha(\boldsymbol{\beta}(\boldsymbol{x}+\sqrt{\epsilon}\boldsymbol{t}))+\mathcal{O}(\epsilon)]\mathrm{d}\boldsymbol{t}\approx\alpha(\boldsymbol{\beta}(\boldsymbol{x}))+\mathcal{O}(\epsilon)$$

(independent of $\boldsymbol{x}$), where the last transition is again based on a Taylor expansion (of $\beta(\cdot)$) and on zeroing out the odd (1st) moment of $\boldsymbol{K(t)}$.

Turning to the numerator (of Eq. (68)),

$$\frac{1}{M^2\epsilon^d}\sum_{j=1}^{M}\sum_{\ell=1}^{M}K^x_{i,\ell}K^y_{\ell,j}g(\boldsymbol{y}_j)\underset{\substack{M\to\infty\\ \epsilon\to 0}}{\longrightarrow}N(\boldsymbol{x})\triangleq\frac{1}{\epsilon^d}\int_{\mathbb{R}^d}\int_{\mathbb{R}^d}K\Big(\frac{\boldsymbol{s}-\boldsymbol{x}}{\sqrt{\epsilon}}\Big)K\Big(\frac{\boldsymbol{y}-\boldsymbol{\beta}(s)}{\sqrt{\epsilon}}\Big)g(\boldsymbol{y})\alpha(\boldsymbol{y})\mathrm{d}\boldsymbol{s}\mathrm{d}\boldsymbol{y}.$$

By applying a change of variables $\boldsymbol{z}=\frac{\boldsymbol{y}-\boldsymbol{\beta}(s)}{\sqrt{\epsilon}}$, $\boldsymbol{y}=\boldsymbol{\beta}(\boldsymbol{s})+\sqrt{\epsilon}\boldsymbol{z}$, $\mathrm{d}\boldsymbol{z}=\mathrm{d}\boldsymbol{y}\epsilon^{d/2}$ we get

$$N(\boldsymbol{x})=\frac{1}{\epsilon^{d/2}}\int_{\mathbb{R}^d}\int_{\mathbb{R}^d}K\Big(\frac{\boldsymbol{s}-\boldsymbol{x}}{\sqrt{\epsilon}}\Big)K(\boldsymbol{z})\gamma(\boldsymbol{\beta}(\boldsymbol{s})+\sqrt{\epsilon}\boldsymbol{z})\mathrm{d}\boldsymbol{s}\mathrm{d}\boldsymbol{z}.$$

Using Taylor's expansion of $\gamma(\cdot)$ we get

$$N(\boldsymbol{x})\approx\frac{1}{\epsilon^{d/2}}\int_{\mathbb{R}^d}\int_{\mathbb{R}^d}K\Big(\frac{\boldsymbol{s}-\boldsymbol{x}}{\sqrt{\epsilon}}\Big)K(\boldsymbol{z})\left[\gamma(\boldsymbol{\beta}(\boldsymbol{s}))+\frac{\sqrt{\epsilon}}{2}\boldsymbol{z}^T\nabla\gamma(\boldsymbol{\beta}(\boldsymbol{s}))+\frac{\epsilon}{2}\boldsymbol{z}^T\boldsymbol{H}\boldsymbol{z}+\mathcal{O}(\epsilon^{3/2})\right]\mathrm{d}\boldsymbol{s}\mathrm{d}\boldsymbol{z}$$

where $H_{i,j}\triangleq\frac{\partial^2\gamma(\boldsymbol{\beta}(\boldsymbol{s}))}{\partial s_i\partial s_j}$ is the Hessian. The first term yields the integral over $\boldsymbol{K(z)}$, while the second term vanishes due to integration over an odd (1st) moment of the symmetric kernel $\boldsymbol{K(z)}$. The last term yields

$$\begin{aligned}&\int_{\mathbb{R}^d}K(\boldsymbol{z})\boldsymbol{z}^{\mathrm{T}}\frac{\partial^2\gamma(\boldsymbol{\beta}(\boldsymbol{s}))}{\partial s_i\partial s_j}\boldsymbol{z}\mathrm{d}\boldsymbol{z}\\&=\sum_{i,j}\frac{\partial^2\gamma(\boldsymbol{\beta}(\boldsymbol{s}))}{\partial s_i\partial s_j}\int_{\mathbb{R}^d}z_iz_jK(\boldsymbol{z})\mathrm{d}\boldsymbol{z}\\&=\sum_{i}\frac{\partial^2\gamma(\boldsymbol{\beta}(\boldsymbol{s}))}{\partial {s_i}^2}\int_{\mathbb{R}^d}z_i^2K(\boldsymbol{z})\mathrm{d}\boldsymbol{z}=\triangle\gamma(\boldsymbol{\beta}(\boldsymbol{s})),\end{aligned}$$

where $\triangle$ denotes the Laplacian operator, and where we assumed that $\int_{\mathbb{R}^d}z_iz_jK(\boldsymbol{z})\mathrm{d}\boldsymbol{z}$ vanishes for $i\neq j$ and equals 1 for $i=j$. Note that this is naturally satisfied, e.g., by the Gaussian kernel function $K(\boldsymbol{z})=c\cdot\exp(-0.5\|\boldsymbol{z}\|^2)$ (with $c=(2\pi)^{-d/2}$, as per the scaling requirement). Substituting into $N(\boldsymbol{x})$ we get

$$N(\boldsymbol{x})\approx\frac{1}{\epsilon^{d/2}}\int_{\mathbb{R}^d}K\Big(\frac{\boldsymbol{s}-\boldsymbol{x}}{\sqrt{\epsilon}}\Big)\Big[\gamma(\boldsymbol{\beta}(\boldsymbol{s}))+\frac{\epsilon}{2}\triangle\gamma(\boldsymbol{\beta}(\boldsymbol{s}))+\mathcal{O}(\epsilon^{3/2})\Big]\mathrm{d}\boldsymbol{s}.$$

Using a change of variables $\boldsymbol{t}=\frac{\boldsymbol{s}-\boldsymbol{x}}{\sqrt{\epsilon}}$, $\boldsymbol{s}=\sqrt{\epsilon}\boldsymbol{t}+\boldsymbol{x}$, $\mathrm{d}\boldsymbol{t}=\mathrm{d}\boldsymbol{s}\epsilon^{d/2}$ and $\gamma(\boldsymbol{y})=g(\boldsymbol{y})\alpha(\boldsymbol{y})$ we get

$$N(\boldsymbol{x})\approx\int_{\mathbb{R}^d}K(\boldsymbol{t})\Big[\gamma(\boldsymbol{\beta}(\boldsymbol{x}+\sqrt{\epsilon}\boldsymbol{t}))+\frac{\epsilon}{2}\triangle\gamma(\boldsymbol{\beta}(\boldsymbol{x}+\sqrt{\epsilon}\boldsymbol{t}))+\mathcal{O}(\epsilon^{3/2})\Big]\mathrm{d}\boldsymbol{t}.$$

Using Taylor's expansion (of $\beta(\cdot)$) once again we get

$$N(\boldsymbol{x})\approx\int_{\mathbb{R}^d}K(\boldsymbol{t})\Big[\gamma(\boldsymbol{\beta}(\boldsymbol{x}))+\frac{\epsilon}{2}\triangle\gamma(\boldsymbol{\beta}(\boldsymbol{x}))+\frac{\epsilon}{2}\boldsymbol{t}^T\boldsymbol{H}\boldsymbol{t}+\mathcal{O}(\epsilon^{3/2})\Big]\mathrm{d}\boldsymbol{t},$$

here we neglected terms involving ϵ to a power higher than 3/2 and terms with odd order of $\boldsymbol{t}$ due to the symmetry of the kernel K. This leads to

$$N(\boldsymbol{x})\approx\gamma(\boldsymbol{\beta}(\boldsymbol{x}))+\epsilon\triangle\gamma(\boldsymbol{\beta}(\boldsymbol{x}))+\mathcal{O}(\epsilon^{3/2}),$$

dividing by the denominator we get

$$(\widehat{\boldsymbol{P}}f)(\boldsymbol{x}_i)\approx g(\boldsymbol{\beta}(\boldsymbol{x}_i))+\epsilon\triangle\gamma(\boldsymbol{\beta}(\boldsymbol{x}_i))/\alpha(\boldsymbol{\beta}(\boldsymbol{x}_i))+\mathcal{O}(\epsilon^{3/2})$$

In the same way, we compute the convergence on $g(\boldsymbol{y}_i)$

$$(\widehat{\boldsymbol{P}}g)(\boldsymbol{y}_i)=f(\boldsymbol{\beta}^{-1}(\boldsymbol{y}_i))+\epsilon\triangle\eta(\boldsymbol{\beta}^{-1}(\boldsymbol{y}_i))/\alpha(\boldsymbol{\beta}(\boldsymbol{y}_i)+\mathcal{O}(\epsilon^{3/2}).$$

□

The following issues were ignored in the proof:

- Errors due to approximating the sum by an integral; An upper bound on the associated errors in the single-view DM is derived in [34].

- Deformation due to the fact that the data is sampled from a non uniform density. This changes the result by some constant.
- The data lies on some manifold. This could be dealt by changing the coordinate system and integrating on the manifold.
- When assuming that the data lies on some manifold, the Euclidean distance should be replaced by the geodesic distance along the manifold. As in the analysis of [5], this introduces a factor to the integral.

References

[1] H.-P. Deutsch, Principle component analysis, in: Derivatives and Internal Models, Springer, 2002, pp. 539–547.
[2] J.B. Kruskal, Nonmetric multidimensional scaling: a numerical method, Psychometrika 29 (2) (1964) 115–129.
[3] S.T. Roweis, L.K. Sau, Nonlinear dimensionality reduction by local linear embedding, Science 290.5500 (2000) 2323–2326.
[4] M. Belkin, P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, Neural Comput. 15 (6) (2003) 1373–1396.
[5] R.R. Coifman, S. Lafon, Diffusion maps, Appl. Comput. Harmon. Anal. 21 (1) (2006) 5–30.
[6] W. Liu, X. Ma, Y. Zhou, D. Tao, J. Cheng, p-laplacian regularization for scene recognition, IEEE Trans. Cybern. 49 (8) (2018) 5120–5129.
[7] L.v.d. Maaten, G. Hinton, Visualizing data using t-sne, J. Mach. Learn. Res. 9 (Nov) (2008) 2579–2605.
[8] L. McInnes, J. Healy, N. Saul, L. Großberger, Umap: uniform manifold approximation and projection, J. Open Source Softw. 3 (29) (2018) 861.
[9] X. He, S. Yan, Y. Hu, P. Niyogi, H.-J. Zhang, Face recognition using laplacianfaces, IEEE Trans. Pattern Anal. Mach. Intell. 27 (3) (2005) 328–340.
[10] A. Singer, R.R. Coifman, Non linear independent component analysis with diffusion maps, Appl. Comput. Harmon. Anal. 25(2) (2008) 226–239.
[11] O. Lindenbaum, A. Yeredor, I. Cohen, Musical key extraction using diffusion maps, Signal Process. 117 (2015) 198–207.
[12] W.T. Freeman, J.B. Tenenbaum, Learning bilinear models for two-factor problems in vision, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE, 1997, pp. 554–560.
[13] I.S. Helland, Partial least squares regression and statistical models, Scand. J. Stat. 17 (1990) 97–114.
[14] K. Chaudhuri, S.M. Kakade, K. Livescu, K. Sridharan, Multi-view clustering via canonical correlation analysis, in: Proceedings of the Twenty-Sixth Annual International Conference on Machine Learning, ACM, 2009, pp. 129–136.
[15] P.L. Lai, C. Fyfe, Kernel and nonlinear canonical correlation analysis, Int. J. Neural Syst. 10 (05) (2000) 365–377.
[16] F.R. Bach, M.I. Jordan, Kernel independent component analysis, J. Mach. Learn. Res. 3 (Jul) (2002) 1–48.
[17] A. Kumar, H. Daumé, A co-training approach for multi-view spectral clustering, in: Proceedings of the Twenty-Eighth International Conference on Machine Learning (ICML-11), 2011, pp. 393–400.
[18] B. Boots, G. Gordon, Two-manifold problems with applications to nonlinear system identification, arXiv preprint arXiv:1206.4648, 2012.
[19] V.R. de Sa, Spectral clustering with two views, in: Proceedings of the ICML Workshop on Learning with Multiple Views, 2005, pp. 20–27.
[20] S. Sun, X. Xie, M. Yang, Multiview uncorrelated discriminant analysis, IEEE Trans. Cybern. 46 (12) (2016) 3272–3284.
[21] W. Liu, X. Yang, D. Tao, J. Cheng, Y. Tang, Multiview dimension reduction via hessian multiset canonical correlations, Inf. Fusion 41 (2018) 119–128.
[22] B. Wang, J. Jiang, W. Wang, Z.-H. Zhou, Z. Tu, Unsupervised metric fusion by cross diffusion, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2012, pp. 2997–3004.
[23] D. Zhou, C. Burges, Spectral clustering and transductive learning with multiple views, in: Proceedings of the Twenty-Fourth International Conference on Machine Learning, 2007, pp. 1159–1166.
[24] R.R. Lederman, R. Talmon, Learning the geometry of common latent variables using alternating-diffusion, Appl. Comput. Harmon. Anal. 44 (3) (2018) 509–536.
[25] W. Wang, M.A. Carreira-Perpinán, The role of dimensionality reduction in classification., in: Proceedings of the AAAI, 2014, pp. 2128–2134.
[26] H. Liu, L. Liu, T.D. Le, I. Lee, S. Sun, J. Li, et al., Non-parametric sparse matrix decomposition for cross-view dimensionality reduction, IEEE Trans. Multimed. 15 (2017) 138–145.
[27] G. Andrew, R. Arora, J. Bilmes, K. Livescu, Deep canonical correlation analysis, in: Proceedings of the International Conference on Machine Learning, 2013, pp. 1247–1255.
[28] J. Zhao, X. Xie, X. Xu, S. Sun, Multi-view learning overview: recent progress and new challenges, Inf. Fusion 38 (2017) 43–54.
[29] O. Lindenbaum, A. Yeredor, M. Salhov, Learning coupled embedding using multiview diffusion maps, in: Proceedings of the International Conference on Latent Variable Analysis and Signal Separation, Springer, 2015, pp. 127–134.
[30] D.S. Kershaw, The incomplete Cholesky conjugate gradient method for the iterative solution of systems of linear equations, J. Comput. Phys. 26 (1) (1978) 43–65.
[31] S. Lafon, Diffusion maps and geometric harmonics, Yale, 2004 Ph.D dissertation.
[32] Y. Keller, R.R. Coifman, S. Lafon, S.W. Zucker, Audio-visual group recognition using diffusion maps, IEEE Trans. Signal Process. 58 (1) (2010) 403–413.
[33] D. Dov, R. Talmon, I. Cohen, Kernel-based sensor fusion with application to audio-visual voice activity detection, IEEE Trans. Signal Process. 64 (24) (2016) 6406–6416.
[34] A. Singer, R. Erban, I.G. Kevrekidis, R.R. Coifman, Detecting intrinsic slow variables in stochastic dynamical systems by anisotropic diffusion maps, Proc. Natl. Acad. Sci. 106 (38) (2009) 16090–16095.
[35] Y. Aizenbud, A. Averbuch, Matrix decompositions using sub-gaussian random matrices, Inf. Inference: J. IMA 8 (3) (2018) 445–469.
[36] M. Holmes, A. Gray, C. Isbell, Fast SVD for large-scale matrices, in: Proceedings of the Workshop on Efficient Machine Learning at NIPS, 58, 2007, pp. 249–252.
[37] M.P. Holmes, J. Isbell, C. Lee, A.G. Gray, QUIC-SVD: fast SVD using cosine trees, in: Proceedings of the Advances in Neural Information Processing Systems, 2009, pp. 673–680.
[38] R.R. Coifman, M.J. Hirn, Diffusion maps for changing data, Appl. Comput. Harmon. Anal. 36 (1) (2014) 79–107.
[39] T. Ando, Majorization relations for Hadamard products., Linear Algebra Appl. 223 (1995) 57–64.
[40] G. Visick, A weak majorization involving the matrices A B and AB, Linear Algebra Appl. 224/224 (1995) 731–744.
[41] A. Bermanis, A. Averbuch, R.R. Coifman, Multiscale data sampling and function extension, Appl. Comput. Harmon. Anal. 34 (1) (2013) 15–29.
[42] K. Fukunaga, D.R. Olsen, An algorithm for finding intrinsic dimensionality of data, IEEE Trans. Comput. 100 (2) (1971) 176–183.
[43] P.J. Verveer, R.P.W. Duin, An evaluation of intrinsic dimensionality estimators, IEEE Trans. Pattern Anal. Mach. Intell. 17 (1) (1995) 81–86.
[44] C. Ceruti, S. Bassis, A. Rozza, G. Lombardi, E. Casiraghi, P. Campadelli, Danco: an intrinsic dimensionality estimator exploiting angle and norm concentration, Pattern Recognit. 47 (8) (2014) 2569–2581.
[45] A. Bermanis, A. Averbuch, R.R. Coifman, Multiscale data sampling and function extension, Appl. Comput. Harmon. Anal. 34 (1) (2013) 15–29.
[46] M. Belkin, P. Niyogi, Convergence of Laplacian eigenmaps, in: Proceedings of the NIPS, 2006, pp. 129–136.
[47] M. Hein, J.-Y. Audibert, U. Von Luxburg, From graphs to manifolds–weak and strong pointwise consistency of graph laplacians, in: Proceedings of the International Conference on Computational Learning Theory, Springer, 2005, pp. 470–485.
[48] A. Singer, From graph to manifold Laplacian: the convergence rate, Appl. Comput. Harmon. Anal. 21 (1) (2006) 128–134.
[49] H. Hotelling, Relations between two sets of variates, Biometrika (1936) 321–377.
[50] K.Q. Weinberger, F. Sha, L.K. Saul, Learning a kernel matrix for nonlinear dimensionality reduction, in: Proceedings of the Twenty-First International Conference on Machine Learning, ACM, 2004, p. 106.
[51] Y. Shiokawa, Y. Date, J. Kikuchi, Application of kernel principal component analysis and computational machine learning to exploration of metabolites strongly associated with diet, Sci. Rep. 8 (1) (2018) 3426.
[52] Q. Zhang, S. Filippi, A. Gretton, D. Sejdinovic, Large-scale kernel methods for independence testing, Stat. Comput. 28 (1) (2018) 113–130.
[53] B. Schölkopf, The kernel trick for distances, in: Proceedings of the Advances in Neural Information Processing Systems, 2001, pp. 301–307.
[54] S. Lafon, Y. Keller, R. Coifman, Data fusion and multicue data matching by diffusion maps, IEEE Trans. Pattern Anal. Mach. Intell. 28 no. 11 (2006) 17841797.
[55] G. Piella, Diffusion maps for multimodal registration, Sensors 14 (6) (2014) 10562–10577.
[56] O. Lindenbaum, A. Yeredor, A. Averbuch, Clustering based on multiview diffusion maps, in: Proceedings of the IEEE Sixteenth International Conference on Data Mining Workshops (ICDMW), IEEE, 2016, pp. 740–747.
[57] A.Y. Ng, M.I. Jordan, Y. Weiss, On spectral clustering1 analysis and an algorithm, in: Proceedings of Advances in Neural Information Processing Systems, Cambridge, MA, 14, MIT Press, 2001, pp. 849–856.
[58] M. Polito, P. Perona, Grouping and dimensionality reduction by locally linear embedding, in: Proceedings of the NIPS, 2001, pp. 1255–1262.
[59] P.A. Estévez, M. Tesmer, C.A. Perez, J.M. Zurada, Normalized mutual information feature selection, IEEE Trans. Neural Netw. 20 (2) (2009) 189–201.
[60] G. Loosli, S. Canu, L. Bottou, Training invariant support vector machines using selective sampling, Large Scale Kernel Mach. 2 (2007) 301–320.
[61] L. Fei-Fei, R. Fergus, P. Perona, Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories, Comput. Vis. Image Underst. 106 (1) (2007) 59–70.
[62] Y.-G. Jiang, C.-W. Ngo, J. Yang, Towards optimal bag-of-features for object categorization and semantic video retrieval, in: Proceedings of the Sixth ACM International Conference on Image and Video Retrieval, ACM, 2007, pp. 494–501.
[63] Y. Bai, L. Guo, L. Jin, Q. Huang, A novel feature extraction method using pyramid histogram of orientation gradients for smile recognition, in: Proceedings of the Sixteenth IEEE International Conference on Image Processing (ICIP), IEEE, 2009, pp. 3305–3308.
[64] P. Gehler, S. Nowozin, On feature combination for multiclass object detection, in: Proceedings of the International Conference on Computer Vision, 2, 2009, p. 6.
[65] M.A. Mouchet, S. Villéger, N.W. Mason, D. Mouillot, Functional diversity measures: an overview of their redundancy and their ability to discriminate community assembly rules, Funct. Ecol. 24 (4) (2010) 867–876.
[66] P.J. Rousseeuw, Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, J. Comput. Appl. Math. 20 (1987) 53–65.
[67] R. Blandford, Seismic event discrimination, Bull. Seismol. Soc. Am. 72 (1982) 569–587.
[68] A.J. Rodgers, T. Lay, W.R. Walter, K.M. Mayeda, A comparison of regional-phase amplitude ratio measurement techniques, Bull. Seismol. Soc. Am. 87 (6) (1997) 1613–1621.
[69] L. Küperkoch, T. Meier, T. Diehl, Automated event and phase identification, New Man. Seismol. Obs. Pract. 2 (2012) 1–52.

[70] R. Hofstetter, Y. Gitterman, V. Pinsky, N. Kraeva, L. Feldman, Seismological observations of the northern dead sea basin earthquake on 11 february 2004 and its associated activity., Israel J. Earth Sci. 57 (2008) 101–124.
[71] M. Joswig, Automated classification of local earthquake data in the bug small array, Geophys. J. Int. 120 (2) (1995) 262–286.
[72] D.B. Harris, A waveform correlation method for identifying quarry explosions, Bull. Seismol. Soc. Am. 81 (6) (1991) 2395–2418.
[73] D.B. Harris, T. Kvaerna, Superresolution with seismic arrays using empirical matched field processing, Geophys. J. Int. 182 (3) (2010) 1455–1477.