

# Local Canonical Correlation Analysis for Nonlinear Common Variables Discovery

Or Yair, *Student Member, IEEE*, and Ronen Talmon, *Member, IEEE*

**Abstract**—In this paper, we address the problem of hidden common variables discovery from multimodal data sets of nonlinear high-dimensional observations. We present a metric based on local applications of canonical correlation analysis (CCA) and incorporate it in a kernel-based manifold learning technique. We show that this metric discovers the hidden common variables underlying the multimodal observations by estimating the Euclidean distance between them. Our approach can be viewed both as an extension of CCA to a nonlinear setting as well as an extension of manifold learning to multiple data sets. Experimental results show that our method indeed discovers the common variables underlying high-dimensional nonlinear observations without assuming prior rigid model assumptions.

**Index Terms**—CCA, diffusion maps, metric learning, multimodal.

## I. INTRODUCTION

THE need to study and analyze complex systems arises in many fields. Nowadays, in more and more applications and devices, many sensors are used to collect and to record multiple channels of data, a fact that increases the amount of information available to analyze the state of the system of interest. In such cases, it is typically insufficient to study each channel separately. Yet, the ability to gain a deep understanding of the true state of the system from the overwhelming amount of collected data from multiple (usually different) sources of information is challenging; it calls for the development of new technologies and novel ways to observe the system of interest and to fuse the available information [1]. For example, the study of human physiology in many fields of medicine is performed by simultaneously monitoring various medical features through electroencephalography (EEG) signals, electrocardiography (ECG) signals, respiratory signals, etc. Each type of measurement carries different and specific information, while our purpose is to systematically discover an accurate description of the state of the patient/person.

A commonly-used method that has the ability to reveal correlations between multiple different sets, which often furthers our understanding of the system, is the *Canonical Correlation*

Manuscript received April 27, 2016; revised September 5, 2016 and October 16, 2016; accepted October 16, 2016. Date of publication November 14, 2016; date of current version December 15, 2016. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Olivier Lezoray. This work was supported by the Israel Science Foundation under Grant 1490/16.

The authors are with the Viterbi Faculty of Electrical Engineering, Technion–Israel Institute of Technology, Haifa 32000, Israel (e-mail: oryair@campus.technion.ac.il; ronen@ee.technion.ac.il).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TSP.2016.2628348

*Analysis* (CCA) [2]–[4]. CCA is a well known and studied algorithm, where linear projections maximizing the correlation between the two data sets are constructed. The main limitation of the CCA algorithm is the inherent restriction to linear relationships, whereas in medical recordings, for example, it is unlikely that the collected data carry only linear information on the human physiological features. To circumvent this linear restriction and to accommodate nonlinearities, kernel-based extensions of CCA (KCCA) have been developed, e.g., [5], which allow for the discovery of nonlinear relationships between data sets. Indeed, KCCA has proven to be beneficial in many cases [6]–[8]. Recently, a gamut of work extending CCA based on various combinations and manipulations of kernels has been presented, e.g., [9]–[13].

In this paper, we use a different approach using manifold learning [14]–[16]. The core of manifold learning resides in the construction of a kernel representing affinities between data samples based on pairwise distance metrics. Such distance metrics define local relationships, which are then aggregated into a global *nonlinear* representation of the entire data set. Indeed, in recent studies, various local distance metrics extending the usage of the prototypical Euclidean metric in the context of kernel-based manifold learning have been introduced, e.g. [10], [17]–[30]. Along this line of research, our focus in the present work has been on the construction of a local Riemannian metric for sensor data fusion, which in turn, can be incorporated in a kernel-based manifold learning technique. Note that a nonlinear variant of CCA was presented in [31] using deep neural networks. This algorithm detects linear correlations after a nonlinear transform (using a deep neural network), whereas in our proposed method we detect nonlinear relations among shared views.

The contribution of our work is two-fold. First, we present a metric for discovering the hidden common variables underlying multiple data sets of observations. Second, we devise a data-driven method based on this metric that extends manifold learning to multiple data sets and gives a nonlinear parametrization of the hidden common variables.

Here, we consider the following setting. We assume a system of interest, observed by two (or more) observation functions. Each observation captures via a nonlinear and high-dimensional function the system intrinsic variables. These variables are common to all the observations. In addition, each observation may introduce additional (noise) variables, which are specific to each function. In other words, we assume that our system of interest is monitored via several observations, each observation, in addition to observing the system itself, observes additional features which are not related to the system. Consequently, our

focus is on obtaining the common hidden variables which hopefully represent the true state of the system. Our method includes two main steps. (i) The construction of a local metric. This metric estimates the Euclidean distance between any two realizations of the hidden common variables among nonlinear and high-dimensional data sets. This is accomplished by using a “local” application of CCA, which emphasizes the common variables underlying the collected data sets while suppressing the observation-specific features which tend to mask the important information on the system. We show that the local metric computed from multiple data sets is a natural extension of a modified *Mahalanobis distance* presented in [17]–[19], which is computed from only on a single data set. (ii) The usage of a manifold learning method, Diffusion Maps [16], which recovers a nonlinear global parametrization of the common variables based on the constructed local metric. Initially, we focus on a setting with only two data sets, and then, we present an extension of our method for multiple sets using multi-linear algebra involving tensor product and tensor decomposition [32], [33].

Experimental results demonstrate that our method is indeed able to identify the hidden common variables in simulations. In particular, we present an example of a dynamical system with a definitive underlying model and demonstrate that without any prior model knowledge, our method obtains an accurate description of the state of the system solely from high-dimensional nonlinear observations. Moreover, the experiments demonstrate the ability of the algorithm to successfully cope with observations that are only weakly related to the system, a situation in which we show that KCCA and another recently introduced method fail.

This paper is organized as follows. In Section II we formulate the problem. Section III gives a brief scientific background presenting CCA and defining the notation used throughout this paper. In Section IV we derive the local Riemannian metric that estimates the Euclidean distance between realizations of the hidden common variable, and we present several results regarding the equality of the estimation. We also compare the proposed metric to a metric which was recently introduced in [34] and show the advantages of the present one. In Section V we incorporate the metric into Diffusion Maps and construct a global parametrization of the common variables. Section VI presents an extension of our method for the case of multiple (more than two) data sets. In Section VII, experimental results demonstrate the ability to discover an accurate parametrization of the hidden common variables from multiple data sets of observations in three different experiments and simulations. Finally, in Section VIII, we conclude with several insights and directions for future work.

## II. PROBLEM FORMULATION

We consider a system of interest whose hidden state is governed by  $d_z$  isotropic variables  $\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{I}_{d_z})$ ,  $\mathbf{z} \in \mathbb{R}^{d_z}$ . We assume that the hidden variables  $\mathbf{z}$  can only be accessed via some observation functions. In this paper, we focus on the case where the hidden state of the system is accessed using two (or more) observation functions, which are possibly nonlinear and

are assumed to be locally invertible. For simplicity, the exposition here focuses on two observation functions. In Section VI, we present an extension for more than two. The observations are given by

$$\mathbf{x} = f(\mathbf{z}, \boldsymbol{\epsilon}), \quad \mathbf{x} \in \mathbb{R}^{d_x}, \quad (1)$$

$$\mathbf{y} = g(\mathbf{z}, \boldsymbol{\eta}), \quad \mathbf{y} \in \mathbb{R}^{d_y} \quad (2)$$

where  $\boldsymbol{\epsilon} \in \mathbb{R}^{d_\epsilon}$  and  $\boldsymbol{\eta} \in \mathbb{R}^{d_\eta}$  are (hidden) observation-specific variables which depend on the observation mechanism and are assumed as not related to the system of interest. The two observation functions can represent, for example, two different sensors, each introducing additional variables. The probability densities of the hidden variables  $\boldsymbol{\epsilon}$  and  $\boldsymbol{\eta}$  are unknown. We assume that the common variables  $\mathbf{z}$  and the observation-specific variables  $\boldsymbol{\epsilon}$  and  $\boldsymbol{\eta}$  are uncorrelated, i.e.,  $\boldsymbol{\Sigma}_{z\epsilon} = \boldsymbol{\Sigma}_{z\eta} = \boldsymbol{\Sigma}_{\epsilon\eta} = \mathbf{0}$ , where  $\boldsymbol{\Sigma}_{ab} = \mathbb{E}[\mathbf{a}\mathbf{b}^T]$ . Finally, we assume that the observations are in higher dimension, i.e.,  $d_z + d_\epsilon \leq d_x$ ,  $d_z + d_\eta \leq d_y$ .

Given  $N$  realizations of the hidden variables,  $\{\mathbf{z}_i, \boldsymbol{\epsilon}_i, \boldsymbol{\eta}_i\}_{i=1}^N$ , we obtain two data sets of observations:

$$\mathcal{X} = \left\{ \mathbf{x}_i \mid \mathbf{x}_i = f(\mathbf{z}_i, \boldsymbol{\epsilon}_i) \right\}_{i=1}^N$$

$$\mathcal{Y} = \left\{ \mathbf{y}_i \mid \mathbf{y}_i = g(\mathbf{z}_i, \boldsymbol{\eta}_i) \right\}_{i=1}^N$$

Our goal in this paper is to devise a method which builds a parametrization of the hidden common variables  $\mathbf{z}$  from the two observation sets  $\mathcal{X}$  and  $\mathcal{Y}$ . The method consists two main steps. First, a local metric for the hidden variables is constructed. More specifically, we derive a pairwise distance function  $D_{ij}$  from the sets  $\mathcal{X}$  and  $\mathcal{Y}$  that corresponds to the Euclidean distance between the common variables  $\mathbf{z}$  and neglects the observation-specific variables,  $\boldsymbol{\epsilon}$  and  $\boldsymbol{\eta}$ , i.e.,

$$D_{ij} \approx \|\mathbf{z}_i - \mathbf{z}_j\|_2^2, \quad i, j = 1, \dots, N. \quad (3)$$

Second, manifold learning is applied with a kernel based on the local distances  $D_{ij}$ .

## III. BACKGROUND AND NOTATION

Let  $v_x \triangleq \langle \mathbf{p}_x, \mathbf{x} \rangle$  be the inner product between the vector  $\mathbf{x}$  and a direction  $\mathbf{p}_x$ , defined by  $\langle \mathbf{p}_x, \mathbf{x} \rangle = \mathbf{p}_x^T \mathbf{x}$ . Analogously, let  $v_y \triangleq \langle \mathbf{p}_y, \mathbf{y} \rangle$ . CCA is traditionally applied to two zero mean random vectors  $\mathbf{x}$  and  $\mathbf{y}$  and finds the directions that maximize the correlation between  $v_x$  and  $v_y$ . The first direction is obtained by solving the following optimization problem:

$$\rho^* = \max_{\mathbf{p}_x, \mathbf{p}_y} \rho(v_x, v_y) \quad (4)$$

where the correlation between  $v_x$  and  $v_y$  is given by:

$$\rho(v_x, v_y) = \frac{\mathbb{E}[v_x v_y]}{\sqrt{\mathbb{E}[v_x^2] \mathbb{E}[v_y^2]}}$$

In a similar manner,  $d$  directions are obtained iteratively, where  $d \triangleq \min(\text{rank}(\boldsymbol{\Sigma}_{xx}), \text{rank}(\boldsymbol{\Sigma}_{yy}))$ . In each iteration, an additional direction is computed by solving (4), with the restriction

that the projection of the random vector on the current direction is orthogonal to the projections on the directions attained in previous iterations. Since the correlation between  $v_x$  and  $v_y$  is invariant to (nonzero) scalar multiplication, we have

$$\begin{aligned} \rho(\alpha v_x, v_y) &= \frac{\mathbb{E}[\alpha v_x \cdot v_y]}{\sqrt{\mathbb{E}[\alpha^2 v_x^2] \mathbb{E}[v_y^2]}} \\ &= \frac{\mathbb{E}[v_x \cdot v_y]}{\sqrt{\mathbb{E}[v_x^2] \mathbb{E}[v_y^2]}} = \rho(v_x, v_y) \end{aligned} \quad (5)$$

Thus, CCA constrains the projected variable to have a unit variance, namely  $\mathbb{E}[v_x^2] = \mathbb{E}[v_y^2] = 1$ .

Using Lagrange multipliers the problem is reduced to an eigenvalue problem given by

$$\mathbf{\Gamma} \mathbf{p}_x = \lambda^2 \mathbf{p}_x \quad (6)$$

where  $\mathbf{\Gamma} \triangleq \mathbf{\Sigma}_{xx}^{-1} \mathbf{\Sigma}_{xy} \mathbf{\Sigma}_{yy}^{-1} \mathbf{\Sigma}_{yx}$  and  $\lambda \in \mathbb{R}$  is an unknown scalar. The  $d$  right eigenvectors  $\mathbf{p}_x$  of  $\mathbf{\Gamma}$  corresponding to the largest  $d$  eigenvalues of  $\mathbf{\Gamma}$  are solutions of the optimization problem, where the eigenvalues  $\lambda^2$  are the maximal correlations. Thus, the  $d$  directions of CCA can be computed via the eigenvalue decomposition problem (6), circumventing the iterative procedure.

In summary, the application of CCA to two random vectors  $\mathbf{x} \in \mathbb{R}^{d_x}$  and  $\mathbf{y} \in \mathbb{R}^{d_y}$  results in two matrices  $\mathbf{P}_x \in \mathbb{R}^{d_x \times d}$  and  $\mathbf{P}_y \in \mathbb{R}^{d_y \times d}$  and a diagonal matrix  $\mathbf{\Lambda} \in \mathbb{R}^{d \times d}$ , where  $\mathbf{P}_x$  consists of the  $d$  directions  $\mathbf{p}_x$ ,  $\mathbf{P}_y$  consists of the  $d$  directions  $\mathbf{p}_y$ , and  $\mathbf{\Lambda}$  consists of the  $d$  eigenvalues  $\lambda^2$  on the diagonal. As a result, the random vector  $\mathbf{v}_x = \mathbf{P}_x^T \mathbf{x}$  satisfies  $\mathbb{E}[\mathbf{v}_x \mathbf{v}_x^T] = \mathbf{I}$ . In the same manner,  $\mathbf{v}_y = \mathbf{P}_y^T \mathbf{y}$ . In addition, the correlation between the  $i$ th entry in  $\mathbf{v}_x$  and the  $i$ th entry in  $\mathbf{v}_y$  is greater or equal than the correlation between the  $(i+1)$ th entry. For more details, see [2].

#### IV. LEARNING THE LOCAL METRIC

To obtain a parametrization of the hidden common variables, we construct a metric that satisfies (3), which simultaneously implies good approximation of the Euclidean distance between any two realizations of the hidden state variables  $\mathbf{z}_i$  and  $\mathbf{z}_j$ , as well as the attenuation of any effect caused by the observation-specific variables  $\boldsymbol{\epsilon}$  and  $\boldsymbol{\eta}$ .

##### A. Linear Case

We first describe a special case where  $f$  and  $g$  are linear functions, namely:

$$\mathbf{x} = \mathbf{J}_x \begin{bmatrix} \mathbf{z} \\ \boldsymbol{\epsilon} \end{bmatrix}, \quad \mathbf{y} = \mathbf{J}_y \begin{bmatrix} \mathbf{z} \\ \boldsymbol{\eta} \end{bmatrix} \quad (7)$$

where  $\mathbf{J}_x \in \mathbb{R}^{d_x \times (d_z + d_\epsilon)}$  and  $\mathbf{J}_y \in \mathbb{R}^{d_y \times (d_z + d_\eta)}$ . Note that the assumption  $d_z + d_\epsilon \leq d_x$ ,  $d_z + d_\eta \leq d_y$  entails that the set of equations (7) are overdetermined. By the notation of Section III, applying CCA to the random vectors  $\mathbf{x}$  and  $\mathbf{y}$

results in the following projection matrices:

$$\mathbf{P}_x = \left( \begin{bmatrix} \mathbf{U}_z & \mathbf{0} \\ \mathbf{0} & \mathbf{U}_\epsilon \end{bmatrix} \mathbf{J}_x^\dagger \right)^T, \quad \mathbf{P}_y = \left( \begin{bmatrix} \mathbf{V}_z & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_\eta \end{bmatrix} \mathbf{J}_y^\dagger \right)^T \quad (8)$$

and with the following correlation matrix:

$$\mathbf{\Lambda} = \begin{bmatrix} \mathbf{I}_{d_z} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \quad \mathbf{\Lambda} \in \mathbb{R}^{d \times d} \quad (9)$$

where  $\mathbf{U}_z, \mathbf{V}_z, \mathbf{U}_\epsilon, \mathbf{V}_\eta$  are arbitrary unitary matrices, and  $\mathbf{J}_x^\dagger$  and  $\mathbf{J}_y^\dagger$  are the Moore-Penrose pseudoinverse of  $\mathbf{J}_x$  and  $\mathbf{J}_y$ , respectively, i.e.,  $\mathbf{J}_x^\dagger \mathbf{J}_x = \mathbf{I}_{(d_z + d_\epsilon) \times (d_z + d_\epsilon)}$  and  $\mathbf{J}_y^\dagger \mathbf{J}_y = \mathbf{I}_{(d_z + d_\eta) \times (d_z + d_\eta)}$ .

*Proposition 1:* In the linear case, the Euclidean distance between any two realizations  $\mathbf{z}_i$  and  $\mathbf{z}_j$  of the random variable  $\mathbf{z}$  is given by:

$$\|\mathbf{z}_i - \mathbf{z}_j\|_2^2 = (\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{P}_x \mathbf{\Lambda} \mathbf{P}_x^T (\mathbf{x}_i - \mathbf{x}_j)$$

*Proof:*

$$\begin{aligned} \Delta \mathbf{x}^T \mathbf{P}_x \mathbf{\Lambda} \mathbf{P}_x^T \Delta \mathbf{x} &= \left\| \mathbf{\Lambda}^{\frac{1}{2}} \mathbf{P}_x^T \Delta \mathbf{x} \right\|_2^2 \\ &= \left\| \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{U}_z & \mathbf{0} \\ \mathbf{0} & \mathbf{U}_\epsilon \end{bmatrix} \mathbf{J}_x^\dagger \mathbf{J}_x \begin{bmatrix} \Delta \mathbf{z} \\ \Delta \boldsymbol{\epsilon} \end{bmatrix} \right\|_2^2 \\ &= \left\| \begin{bmatrix} \mathbf{U}_z & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{z} \\ \Delta \boldsymbol{\epsilon} \end{bmatrix} \right\|_2^2 = \|\Delta \mathbf{z}\|_2^2 \end{aligned}$$

where  $\Delta \mathbf{x} = \mathbf{x}_i - \mathbf{x}_j$ ,  $\Delta \mathbf{z} = \mathbf{z}_i - \mathbf{z}_j$ , and  $\Delta \boldsymbol{\epsilon} = \boldsymbol{\epsilon}_i - \boldsymbol{\epsilon}_j$ . ■

Note that the resulting distance takes into account *only* the common hidden variables  $\mathbf{z}$  and filters out the observation-specific variables  $\boldsymbol{\epsilon}$  using the matrix  $\mathbf{\Lambda}$ . The Euclidean distance between realizations of  $\mathbf{z}$  can be expressed in an analogous manner based on realizations of  $\mathbf{y}$ ; one can also take the average of the two metrics using both realizations of  $\mathbf{x}$  and  $\mathbf{y}$ .

##### B. Nonlinear Case

In the general case, where  $f(\mathbf{z}, \boldsymbol{\epsilon})$  and  $g(\mathbf{z}, \boldsymbol{\eta})$  are nonlinear, we use a linearization approach to obtain a similar result to the linear case (up to some bounded error). We denote  $\mathbf{v}_j \triangleq [\mathbf{z}_j^T \quad \boldsymbol{\epsilon}_j^T]^T$ , such that  $\mathbf{x}_j = f(\mathbf{v}_j)$ , and expand  $f$  via its Taylor series around  $\mathbf{v}_j$ :

$$\mathbf{x}_i = \mathbf{x}_j + \mathbf{J}_x(\mathbf{v}_j) [\mathbf{v}_i - \mathbf{v}_j] + \mathcal{O}(\|\mathbf{v}_i - \mathbf{v}_j\|^2)$$

where  $\mathbf{J}_x(\mathbf{v}_j)$  is the Jacobian of  $f$  at  $\mathbf{v}_j$ . Reorganizing the expression above yields:

$$f(\mathbf{v}_i) = \underbrace{\mathbf{x}_j - \mathbf{J}_x(\mathbf{v}_j) \mathbf{v}_j}_{\text{Constant}} + \underbrace{\mathbf{J}_x(\mathbf{v}_j) \mathbf{v}_i}_{\text{Linear Part}} + \mathcal{O}(\|\mathbf{v}_i - \mathbf{v}_j\|^2).$$

Thus, when the higher-order terms are negligible, this form is similar to the linear function considered in Section IV-A with an additional constant term. Define the matrices  $\mathbf{P}_x(\mathbf{x}_j)$  and  $\mathbf{\Lambda}(\mathbf{x}_j)$  similarly to (8) and (9), respectively, using  $\mathbf{J}_x(\mathbf{v}_j)$  (instead of  $\mathbf{J}_x$ ) as the linear function. The following Proposition

uses the fact that the projection matrix obtained by the local CCA is a rotated version of the local Jacobian matrix of the observations. Note that the local Jacobian can be helpful in revealing the common variable from observations locally by itself. Here, as in the linear case, we use it to form a metric between the observations.

*Proposition 2:* In the nonlinear case, the Euclidean distance between any two realizations  $\mathbf{z}_i$  and  $\mathbf{z}_j$  of the random variable  $\mathbf{z}$  is given by:

$$\begin{aligned} \|\mathbf{z}_i - \mathbf{z}_j\|_2^2 &= (\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{A}(\bar{\mathbf{x}}_{ij}) (\mathbf{x}_i - \mathbf{x}_j) \\ &\quad + \mathcal{O}(\|\mathbf{x}_i - \mathbf{x}_j\|^4) \end{aligned} \quad (10)$$

where  $\bar{\mathbf{x}}_{ij} \triangleq (\mathbf{x}_i + \mathbf{x}_j)/2$  denotes the middle point, and  $\mathbf{A}(\bar{\mathbf{x}}_{ij}) \triangleq \mathbf{P}_x(\bar{\mathbf{x}}_{ij}) \mathbf{\Lambda}(\bar{\mathbf{x}}_{ij}) \mathbf{P}_x^T(\bar{\mathbf{x}}_{ij})$ .

*Proof:* In the proof of Proposition 1 we show that one can write the common variables  $\mathbf{z}$  up to some rotation by  $\mathbf{U}_z \mathbf{z} = \tilde{\mathbf{P}}_x^T \mathbf{x}$ , where  $\tilde{\mathbf{P}}_x$  are the  $d_z$  leftmost columns of  $\mathbf{P}_x$ . Notice that since  $\mathbf{U}_z$  is unitary, we have  $\|\mathbf{z}_i - \mathbf{z}_j\|_2^2 = \|\mathbf{U}_z \mathbf{z}_i - \mathbf{U}_z \mathbf{z}_j\|_2^2$ . Thus, since the norm is invariant to rotation, we can recover  $\mathbf{z}$  up to rotation. With a slight abuse of notation, let  $f^{-1}$  denote the local inverse function of  $f$  restricted to  $\mathbf{z}$ . The linearization of  $f^{-1}$  around the point  $\bar{\mathbf{x}}_{ij} \triangleq (\mathbf{x}_j + \mathbf{x}_i)/2$  is given by:

$$\begin{aligned} \mathbf{z}_i &= \bar{\mathbf{z}} + \tilde{\mathbf{P}}_x^T(\bar{\mathbf{x}}_{ij}) \left[ \mathbf{x}_i - \frac{\mathbf{x}_j + \mathbf{x}_i}{2} \right] \\ &= \bar{\mathbf{z}} + \tilde{\mathbf{P}}_x^T(\bar{\mathbf{x}}_{ij}) \left[ \frac{\mathbf{x}_i - \mathbf{x}_j}{2} \right] \end{aligned}$$

where  $\bar{\mathbf{z}} = f^{-1}(\bar{\mathbf{x}}_{ij})$ . Thus, the Taylor expansion of the  $l$ th element of  $\mathbf{z}_i$  is given by:

$$\begin{aligned} (z_i)_l &= (\bar{z})_l + \frac{1}{2} \left( \mathbf{p}_x^{(l)}(\bar{\mathbf{x}}_{ij}) \right)^T (\mathbf{x}_i - \mathbf{x}_j) \\ &\quad + \frac{1}{8} (\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{H}_x^{(l)}(\bar{\mathbf{x}}_{ij}) (\mathbf{x}_i - \mathbf{x}_j) \\ &\quad + \mathcal{O}(\|\mathbf{x}_i - \mathbf{x}_j\|^3) \end{aligned} \quad (11)$$

where  $\mathbf{p}_x^{(l)}(\mathbf{x})$  is the  $l$ -th column of  $\mathbf{P}_x(\mathbf{x})$ , and  $\mathbf{H}_x^{(l)}(\mathbf{x})$  is the Hessian of the  $l$ -th entry of  $f^{-1}$ . In a similar way, expanding the  $l$ -th element of  $\mathbf{z}_j$  around the same point gives:

$$\begin{aligned} (z_j)_l &= (\bar{z})_l + \frac{1}{2} \left( \mathbf{p}_x^{(l)}(\bar{\mathbf{x}}_{ij}) \right)^T (\mathbf{x}_j - \mathbf{x}_i) \\ &\quad + \frac{1}{8} (\mathbf{x}_j - \mathbf{x}_i)^T \mathbf{H}_x^{(l)}(\bar{\mathbf{x}}_{ij}) (\mathbf{x}_j - \mathbf{x}_i) \\ &\quad + \mathcal{O}(\|\mathbf{x}_j - \mathbf{x}_i\|^3) \end{aligned} \quad (12)$$

Subtracting (12) from (11) yields:

$$\begin{aligned} (z_i)_l - (z_j)_l &= \left( \mathbf{p}_x^{(l)}(\bar{\mathbf{x}}_{ij}) \right)^T (\mathbf{x}_i - \mathbf{x}_j) \\ &\quad + \mathcal{O}(\|\mathbf{x}_i - \mathbf{x}_j\|^3) \end{aligned}$$

Thus:

$$\begin{aligned} \|\mathbf{z}_i - \mathbf{z}_j\|_2^2 &= (\mathbf{x}_i - \mathbf{x}_j)^T \tilde{\mathbf{P}}(\bar{\mathbf{x}}_{ij}) \tilde{\mathbf{P}}^T(\bar{\mathbf{x}}_{ij}) (\mathbf{x}_i - \mathbf{x}_j) \\ &\quad + \mathcal{O}(\|\mathbf{x}_i - \mathbf{x}_j\|^4) \\ &= (\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{A}(\bar{\mathbf{x}}_{ij}) (\mathbf{x}_i - \mathbf{x}_j) \\ &\quad + \mathcal{O}(\|\mathbf{x}_i - \mathbf{x}_j\|^4) \end{aligned}$$

where  $\mathbf{A}(\bar{\mathbf{x}}_{ij}) \triangleq \mathbf{P}_x(\bar{\mathbf{x}}_{ij}) \mathbf{\Lambda}(\bar{\mathbf{x}}_{ij}) \mathbf{P}_x^T(\bar{\mathbf{x}}_{ij})$ .  $\blacksquare$

We note that similarly to Proposition 1, Proposition 2 can be analogously formulated based on realizations of  $\mathbf{y}$  instead of realizations of  $\mathbf{x}$ .

In [34] we presented a different way to calculate the Euclidean distance between two realization:

$$\begin{aligned} \|\mathbf{z}_i - \mathbf{z}_j\|_2^2 &= \frac{1}{2} (\mathbf{x}_i - \mathbf{x}_j)^T [\mathbf{A}(\mathbf{x}_i) + \mathbf{A}(\mathbf{x}_j)] (\mathbf{x}_i - \mathbf{x}_j) \\ &\quad + \mathcal{O}(\|\mathbf{x}_i - \mathbf{x}_j\|^4) \end{aligned}$$

where  $\mathbf{A}(\mathbf{x}_i) \triangleq \mathbf{P}_x(\mathbf{x}_i) \mathbf{\Lambda}(\mathbf{x}_i) \mathbf{P}_x^T(\mathbf{x}_i)$ . Both expressions approximate  $\|\mathbf{z}_i - \mathbf{z}_j\|_2$  up to the second-order. In Section V, we discuss the advantage of the metric using the middle point  $\bar{\mathbf{x}}_{ij}$  in terms of computational complexity. In addition, we address the case where the middle points  $(\mathbf{x}_i + \mathbf{x}_j)/2$  and  $(\mathbf{y}_i + \mathbf{y}_j)/2$  are inaccessible. In Section VII-A, we compare the proposed metric using the middle point (13) with the metric proposed in [34] in a toy problem, which demonstrates that the computation based on the middle point attains a better estimation for the Euclidean distance.

### C. Implementation

Given  $\mathcal{X}$ , we define a pairwise metric between the  $N$  realizations based on Proposition 2.

*Definition 3:* Let  $D_{ij}$  be the distance between each pair of realizations  $\mathbf{x}_i$  and  $\mathbf{x}_j$  in  $\mathcal{X}$  based on the matrix (metric)  $\mathbf{A}(\bar{\mathbf{x}}_{ij})$ , which is given by:

$$D_{ij} \triangleq (\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{A}(\bar{\mathbf{x}}_{ij}) (\mathbf{x}_i - \mathbf{x}_j) \quad (13)$$

where  $\bar{\mathbf{x}}_{ij} \triangleq \frac{\mathbf{x}_i + \mathbf{x}_j}{2}$ . We can also define an analogous metric between any two realizations  $\mathbf{y}_i$  and  $\mathbf{y}_j$  in  $\mathcal{Y}$  or define a metric which is the average of the two.

We now describe the computation of the matrices  $\mathbf{A}(\bar{\mathbf{x}}_{ij})$  from the sets  $\mathcal{X}$  and  $\mathcal{Y}$  provided that we have access to the neighborhoods of the middle points; for the case where they are inaccessible, see Section V. For any point  $\mathbf{x}_i$  (including the middle point), by Proposition 2,  $\mathbf{A}(\mathbf{x}_i)$  can be computed from the matrices  $\mathbf{P}_x(\mathbf{x}_i)$  and  $\mathbf{\Lambda}(\mathbf{x}_i)$ . Let  $\mathcal{X}_i \subset \mathcal{X}$  and  $\mathcal{Y}_i \subset \mathcal{Y}$  be two subsets of realizations defining a small neighborhood  $(\mathcal{X}_i, \mathcal{Y}_i)$  around  $(\mathbf{x}_i, \mathbf{y}_i)$ . The definition of the neighborhoods is application-specific. For example, for time series data, we could use a time window around each point to define its neighbors. Finding the  $k$  nearest neighbors for each realization could be another possibility. The size of the subsets is set according to the following trade-off. On the one hand, there should be a sufficient number of data points in each neighborhood for an accurate

estimation of the local covariance and correlation matrices. On the other hand, the neighborhood should be sufficiently small so that the local linearization results in small error terms in (10). For more details and rigorous analysis of the choice of the neighborhood size, we refer the readers to [35]. This issue was also addressed in the context of the local Mahalanobis distance in [19]. When considering subsets  $(\mathcal{X}_i, \mathcal{Y}_i)$  consisting of only samples  $(\mathbf{x}_i, \mathbf{y}_i)$  within the neighborhood of  $\mathbf{x}_i$  and  $\mathbf{y}_i$ , then in particular the distance  $\|\mathbf{x}_i - \mathbf{x}_j\|_2^4$  between any two realizations in the neighborhood is indeed negligible, and applying CCA to the two sets  $\mathcal{X}_i$  and  $\mathcal{Y}_i$  results in the estimation of  $\mathbf{P}_x(\mathbf{x}_i)$  and  $\mathbf{\Lambda}(\mathbf{x}_i)$ , which leads to the estimation of  $\mathbf{A}(\mathbf{x}_i)$  as desired.

*Proposition 4:* In the absence of the observation-specific variables, the distance  $D_{ij}$  can be written as

$$D_{ij} = (\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{\Sigma}_{xx}^{-1}(\bar{\mathbf{x}}_{ij}) (\mathbf{x}_i - \mathbf{x}_j) \quad (14)$$

where  $\bar{\mathbf{x}}_{ij} \triangleq \frac{\mathbf{x}_i + \mathbf{x}_j}{2}$  and  $\mathbf{\Sigma}_{xx}(\mathbf{x}_i)$  is the covariance of the random variable  $\mathbf{x}$  at the point  $\mathbf{x}_i$  (noting that the covariance changes from point to point due to the nonlinearity of the observation function  $f$ ).

In other words, when there are no observation-specific variables, i.e.,  $\epsilon = \boldsymbol{\eta} = \mathbf{0}$ , the metric we build based on local applications of CCA is a modified *Mahalanobis distance*, which was presented and analyzed in [18], [19], [36] for the purpose of recovering the intrinsic representation from nonlinear observation data.

*Proof:* In the absence of the observation-specific variables, the matrices  $\mathbf{\Lambda}(\mathbf{x})$  become the identity, namely,  $\mathbf{\Lambda}(\mathbf{x}) = \mathbf{I}$  for all  $\mathbf{x}$ . In addition, a known property of CCA links between the matrix  $\mathbf{P}_x$  and the covariance matrix  $\mathbf{\Sigma}_{xx}$  [37]:

$$\mathbf{P}_x^T(\bar{\mathbf{x}}_{ij}) = \mathbf{U}^T(\bar{\mathbf{x}}_{ij}) \mathbf{\Sigma}_{xx}^{-\frac{1}{2}}(\bar{\mathbf{x}}_{ij}) \quad (15)$$

where  $\mathbf{U}(\mathbf{x})$  is a unitary matrix. We recall that

$$D_{ij} \triangleq (\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{A}(\bar{\mathbf{x}}_{ij}) (\mathbf{x}_i - \mathbf{x}_j) \quad (16)$$

and

$$\mathbf{A}(\bar{\mathbf{x}}_{ij}) \triangleq \mathbf{P}_x(\bar{\mathbf{x}}_{ij}) \mathbf{\Lambda}(\bar{\mathbf{x}}_{ij}) \mathbf{P}_x^T(\bar{\mathbf{x}}_{ij}) \quad (17)$$

Substituting  $\mathbf{\Lambda}(\mathbf{x}) = \mathbf{I}$  and (15) into (17) results in

$$\begin{aligned} \mathbf{A}(\bar{\mathbf{x}}_{ij}) &= \mathbf{P}_x(\bar{\mathbf{x}}_{ij}) \mathbf{\Lambda}(\bar{\mathbf{x}}_{ij}) \mathbf{P}_x^T(\bar{\mathbf{x}}_{ij}) \\ &= \mathbf{\Sigma}_{xx}^{-\frac{1}{2}}(\bar{\mathbf{x}}_{ij}) \mathbf{U}(\bar{\mathbf{x}}_{ij}) \mathbf{I} \mathbf{U}^T(\bar{\mathbf{x}}_{ij}) \mathbf{\Sigma}_{xx}^{-\frac{1}{2}}(\bar{\mathbf{x}}_{ij}) \\ &= \mathbf{\Sigma}_{xx}^{-1}(\bar{\mathbf{x}}_{ij}) \end{aligned}$$

where we used  $\mathbf{U}(\bar{\mathbf{x}}_{ij}) \mathbf{I} \mathbf{U}^T(\bar{\mathbf{x}}_{ij}) = \mathbf{I}$ .  $\blacksquare$

## V. GLOBAL PARAMETRIZATION

In Section IV, we proposed a metric that approximates the Euclidean distance between two realizations. The estimation of the Euclidean distance is accurate for small distances, whereas the overall goal is to obtain a global parametrization which corresponds to the hidden common variables  $\mathbf{z}$ . For this purpose, i.e., for obtaining a global parametrization from the local metric, we use a kernel-based manifold learning method, Diffusion Maps [16]. Following common practice, we use a Gaussian

---

**Algorithm 1:** Diffusion Maps of Two Datasets With Middle Points.

---

**Input:** Two sets of observations  $\mathcal{X}$  and  $\mathcal{Y}$ .

**Output:** Low dimensional parametrization of the common variables  $\mathbf{z}$ .

- 1) For each pair of realizations points  $(\mathbf{x}_i, \mathbf{y}_i), (\mathbf{x}_j, \mathbf{y}_j) \in (\mathcal{X}, \mathcal{Y})$ :
    - a) Define the middle points  $\bar{\mathbf{x}}_{ij} \triangleq \frac{1}{2}(\mathbf{x}_i + \mathbf{x}_j)$  and  $\bar{\mathbf{y}}_{ij} \triangleq \frac{1}{2}(\mathbf{y}_i + \mathbf{y}_j)$ .
    - b) Construct the subsets  $\bar{\mathcal{X}}_{ij}, \bar{\mathcal{Y}}_{ij}$  by collecting all pairs  $(\mathbf{x}_k, \mathbf{y}_k)$  such that  $\mathbf{x}_k$  is in the neighborhood of  $\bar{\mathbf{x}}_{ij}$  and  $\mathbf{y}_k$  is in the neighborhood of  $\bar{\mathbf{y}}_{ij}$ .
    - c) Apply (linear) CCA to the sets  $\bar{\mathcal{X}}_{ij}, \bar{\mathcal{Y}}_{ij}$  and obtain the matrices  $\mathbf{P}_x(\bar{\mathbf{x}}_{ij})$  and  $\mathbf{\Lambda}(\bar{\mathbf{x}}_{ij})$ .
    - d) Set  $\mathbf{A}(\bar{\mathbf{x}}_{ij}) \triangleq \mathbf{P}_x(\bar{\mathbf{x}}_{ij}) \mathbf{\Lambda}(\bar{\mathbf{x}}_{ij}) \mathbf{P}_x^T(\bar{\mathbf{x}}_{ij})$
    - e) Compute the distances  $D_{ij} = (\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{A}(\bar{\mathbf{x}}_{ij}) (\mathbf{x}_i - \mathbf{x}_j)$ .
  - 2) Apply Diffusion Maps:
    - a) Construct the kernel:  $W_{ij} = \exp(-D_{ij}/\sigma)$ , where  $\sigma$  is set to the median value of  $\{D_{ij}\}, \forall i, j$ .
    - b) Normalize the kernel  $\mathbf{M} = \mathbf{\Omega}^{-1} \mathbf{W}$ , where  $\mathbf{\Omega}$  is a diagonal matrix with  $\Omega_{ii} = \sum_j W_{ij}$
    - c) Compute the eigenvectors and eigenvalues of the matrix  $\mathbf{M}$ , i.e.,  $\mathbf{M} = \mathbf{\Psi} \mathbf{S} \mathbf{\Psi}^{-1}$ .
  - 3) Form the parametrization of  $\mathbf{z}_i, \forall i = 1, \dots, N$  using the  $d_z$  eigenvectors (the columns of  $\mathbf{\Psi}$ ) associated with the largest  $d_z$  eigenvalues (without the first trivial one), i.e.,  $(\Psi_{i1}, \dots, \Psi_{id_z})^T$  for  $i = 1, \dots, N$ .
- 

kernel  $W_{ij} = \exp(-D_{ij}/\sigma)$ , which emphasizes the notion of locality using the kernel scale  $\sigma$ : for  $D_{ij} \gg \sigma$ , the kernel value  $W_{ij}$  is negligible. Therefore, a proper selection of  $\sigma$  entails that only (sufficiently) small distances  $D_{ij}$  are taken into account in the kernel. The matrix  $\mathbf{W}$  can be considered as a weights matrix of a graph where the observations are the nodes of the graph and the weight of the edge between observations  $i$  and  $j$  is  $W_{ij}$ . One should set  $\sigma$  to the smallest possible value which still maintains a connected graph. See [38] for more details. In this work, we follow common practice and set  $\sigma$  to the median of the distances in  $D_{ij}$  for all  $i$  and  $j$ . By appropriately tuning the value  $\sigma$  to correspond to the linear part of (10),  $W_{ij}$  accurately represents an affinity between the common variables, since the higher-order error terms in (10) disappear. For more details, see [35]. The entire method is presented in Algorithm 1.

In Step 1-b, Algorithm 1 assumes that the neighborhoods of the middle points  $\frac{1}{2}(\mathbf{x}_i + \mathbf{x}_j)$  and  $\frac{1}{2}(\mathbf{y}_i + \mathbf{y}_j)$  are accessible. In addition, for sets of size  $|\mathcal{X}| = |\mathcal{Y}| = N$ , Step 1 is repeated  $\frac{N(N+1)}{2}$  times. This entails that in Step 1-d, the matrix  $\mathbf{A}(\bar{\mathbf{x}}_{ij})$  is computed for every possible middle point  $\bar{\mathbf{x}}_{ij}$ , and overall  $\frac{N(N+1)}{2}$  such CCA matrices are computed, one for each possible pair  $(\mathbf{x}_i, \mathbf{x}_j)$ .

In order to relax the above assumption and to reduce the computational complexity, we present an algorithm based on [36]. The more efficient algorithm is presented in Algorithm 2, where

---

**Algorithm 2:** Diffusion Maps of Two Datasets Without Middle Points.
 

---

**Input:** Two sets of observations  $\mathcal{X}$  and  $\mathcal{Y}$ .

**Output:** Low dimensional parametrization of the common variables  $\mathbf{z}$ .

- 1) Construct some subsets  $\mathcal{X}_L \subseteq \mathcal{X}$  and  $\mathcal{Y}_L \subseteq \mathcal{Y}$  with  $|\mathcal{X}_L| = |\mathcal{Y}_L| = L \leq N$  such that  $\mathbf{x}_i \in \mathcal{X}_L$  iff  $\mathbf{y}_i \in \mathcal{Y}_L$
  - 2) For each pair of points  $(\mathbf{x}_i, \mathbf{y}_i) \in (\mathcal{X}_L, \mathcal{Y}_L)$ :
    - a) Construct the subsets  $\mathcal{X}_i, \mathcal{Y}_i$  by choosing all pairs  $(\mathbf{x}_j, \mathbf{y}_j)$  such that  $\mathbf{x}_j$  is in the neighborhood of  $\mathbf{x}_i$  and  $\mathbf{y}_j$  is in the neighborhood of  $\mathbf{y}_i$ .
    - b) Apply (linear) CCA to the sets  $\mathcal{X}_i, \mathcal{Y}_i$  and obtain the matrices  $\mathbf{P}_x(\mathbf{x}_i)$  and  $\mathbf{\Lambda}(\mathbf{x}_i)$ .
    - c) Set  $\mathbf{A}(\mathbf{x}_i) \triangleq \mathbf{P}_x(\mathbf{x}_i) \mathbf{\Lambda}(\mathbf{x}_i) \mathbf{P}_x^T(\mathbf{x}_i)$
  - 3) For each two observations  $\mathbf{x}_i \in \mathcal{X}_L$  and  $\mathbf{x}_j \in \mathcal{X}$ , compute the distances  $\tilde{D} \in \mathbb{R}^{L \times N}$  according to
 
$$\tilde{D}_{ij} = (\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{A}(\mathbf{x}_i) (\mathbf{x}_i - \mathbf{x}_j)$$
  - 4) Apply Diffusion Maps:
    - a) Construct the kernel:  $W_{ij} = \exp(-\tilde{D}_{ij}/\sigma)$ , where  $\sigma$  is set to the median value of  $\{\tilde{D}_{i,j}\}, \forall i, j$ .
    - b) Normalize the kernel  $\mathbf{M} = \mathbf{\Omega}^{-\frac{1}{2}} \mathbf{W}^T \mathbf{W} \mathbf{\Omega}^{-\frac{1}{2}}$ , where  $\mathbf{\Omega}$  is a diagonal matrix with  $\Omega_{ii} = \sum_j (W^T W)_{ij}$ .
    - c) Compute the eigenvectors and eigenvalues of the matrix  $\mathbf{M}$ , i.e.,  $\mathbf{M} = \mathbf{\Psi} \mathbf{S} \mathbf{\Psi}^{-1}$ .
  - 5) Form the parametrization of  $\mathbf{z}_i, \forall i = 1, \dots, N$  using the  $d_z$  eigenvectors (the columns of  $\mathbf{\Psi}$ ) associated with the largest  $d_z$  eigenvalues (without the first trivial one), i.e.,  $(\Psi_{i1}, \dots, \Psi_{id_z})^T$  for  $i \in 1, \dots, N$ .
- 

the CCA matrices  $\mathbf{A}(\mathbf{x}_i)$  are constructed only for a subset of  $L \leq N$  points  $\mathbf{x}_i \in \mathcal{X}_L \subseteq \mathcal{X}$  (without the need to directly address the middle point), i.e.,  $\mathbf{A}(\mathbf{x}_i)$  is computed only  $L$  times. This modification does not affect the algorithm; Theorem 3.2 presented in [36] states that the entries of matrix  $\mathbf{M}$  calculated in Step 3 in Algorithm 2 are approximations of the entries of the matrix  $\mathbf{M}$  calculated in Step 2 in Algorithm 1. For more details, see [36]. The modification gives rise to two benefits. First, it circumvents the need to have access to the middle points (and their respective neighborhoods). Second, in Algorithm 2 one can reduce the computational load by setting  $\mathcal{X}_L \subset \mathcal{X}$  and then by calculating  $\mathbf{A}(\mathbf{x}_i)$  only at  $L < N$  different points. Overall, Algorithm 2 reduces the number of local CCA applications from  $N^2$  to only  $L$ . The complexity of a single CCA application is  $O(nd^2) + O(d^3)$ , where  $n$  is the size of the subsets, and  $d$  is the maximum of the dimensions of the input vectors from the two views. We note that there are types of data, e.g. patches in images or time series, where the local CCA neighborhood is determined using a sliding window. In such cases, an additional computational complexity improvement can be attained (without loss of accuracy) as suggested in [39]. This can be achieved by rearrangement of the terms in the numerical estimation of the multiple correlation equations, and then by forming the suitable structure for box filters implementation.

## VI. MULTIPLE OBSERVATION SCENARIO

The proposed method can be extended to the case where there are more than two sets of observations. The core of the proposed method relies on the local metric (13). As described in IV, this metric requires the computation of the matrix  $\mathbf{A}$  (at the middle points in Algorithm 1 or at the observations in Algorithm 2). In either case, this matrix is computed based on the canonical directions extracted by a local application of CCA to two sets of observations from two (possibly different) observation functions. Consequently, the extension to more than two sets involves an extension of the local CCA application that enables to compute the canonical directions from more than two sets of observations. Once such canonical directions are identified,  $\mathbf{A}$  can be constructed analogously, and the remainder of the algorithm remains unchanged.

Therefore, this multiple observations case requires a suitable alternative to CCA, which is not restricted to two sets. The algorithm proposed in [40] is a generalized version of the CCA algorithm. This algorithm relies on the construction of a generalized block cross-correlation matrix by concatenating all the data from all the multiple views. In our case, by the definition of the setting of the problem, since we apply CCA locally, we have a relatively small number of realizations per CCA application. Therefore, the generalized cross-correlation matrix comprises coarse estimates with relatively large estimation errors. This poses a problem for the algorithm proposed in [40], since the generalized eigenvalue decomposition applied to this matrix is unstable. A possible approach for circumventing this is to add a small regularization term to the diagonal of the auto-covariance matrices (known as canonical ridge). For more details regarding this instability see [41]. We exploit the method presented in [32], which extends CCA for the case of more than two data sets. As mentioned in [32], this method is limited to finding only the dominant canonical direction for each observation, since finding multiple directions that satisfy the orthogonality constraint is still an open problem [33].

In the remainder of this section, we extend our notation to support multiple observations. Then, we briefly describe the (linear) Tensor CCA (TCCA) method based on [32] using multi-linear algebra, i.e. using tensor products and tensor decompositions. Finally, we present an algorithm for the general multimodal scenario, which extends the algorithm presented in Section V for more than two sets of observations.

Let  $f^{(k)}$  denote the  $k$ th observation function, i.e.,  $\mathbf{x}^{(k)} = f^{(k)}(\mathbf{z}, \epsilon^{(k)})$ , where  $\epsilon^{(k)}$  is a  $k$ th observation-specific variable. Let  $\mathcal{X}^{(k)} = \{\mathbf{x}_i^{(k)}\}_{i=1}^N$  denote the  $k$ th set of observations, where  $1 \leq k \leq K$ .

As mentioned above, extending the derivation of the local metric presented in Section V for  $K > 2$  observation sets requires the use of tensors instead of matrices. The notation that is used throughout this section is as follows.

*Definition 5:* The  $k$ th ( $1 \leq k \leq K$ ) mode product between a  $K$ th order tensor  $\mathcal{T} \in \mathbb{R}^{d_1 \times d_2 \times \dots \times d_K}$  and a matrix  $\mathbf{M} \in \mathbb{R}^{d_k \times D}$

is defined by

$$\begin{aligned} & \mathcal{K}(m_1, \dots, m_{k-1}, n, m_{k+1}, \dots, m_K) \\ &= \sum_{m_k=1}^{d_k} \mathcal{T}(m_1, \dots, m_K) M(m_k, n). \end{aligned}$$

In matrix form this product can be expressed by  $\mathcal{K} = \mathcal{T} \times_k M$ , where

$$\mathcal{K} \in \mathbb{R}^{d_1 \times d_2 \times \dots \times d_{k-1} \times D \times d_{k+1} \times \dots \times d_K}$$

In a similar manner, we define by

$$\mathcal{K} = \mathcal{T} \times_1 M_1 \times_2 M_2 \cdots \times_K M_K$$

the product of  $\mathcal{T}$  with a sequence of matrices  $M_k \in \mathbb{R}^{d_k \times D_k}$  for  $1 \leq k \leq K$ . Note that in this case  $\mathcal{K} \in \mathbb{R}^{D_1 \times D_2 \times \dots \times D_K}$ .

*Definition 6:* Given a  $K$ th order tensor  $\mathcal{T}_1 \in \mathbb{R}^{d_1 \times d_2 \times \dots \times d_K}$ , and a  $J$ th order tensor  $\mathcal{T}_2 \in \mathbb{R}^{D_1 \times D_2 \times \dots \times D_J}$ , their outer product  $\mathcal{T}_1 \otimes \mathcal{T}_2$  is a  $(K+J)$ th order tensor  $\mathcal{K} = \mathcal{T}_1 \otimes \mathcal{T}_2 \in \mathbb{R}^{d_1 \times \dots \times d_K \times D_1 \times \dots \times D_J}$  which holds:

$$\begin{aligned} & \mathcal{K}(m_1, m_2, \dots, m_K, m_{K+1}, \dots, m_{K+J}) \\ &= \mathcal{T}_1(m_1, \dots, m_K) \mathcal{T}_2(m_{K+1}, \dots, m_{K+J}) \end{aligned}$$

In Section IV-B, we estimate the Euclidean distance  $\|z_i - z_j\|_2$  by projecting the observations from each set on the respective canonical directions obtained by a local application of CCA. In the case of multiple sets, we aim to find the generalized canonical directions which maximize the correlation of observations  $\mathbf{x}^{(k)}$  from all sets  $k = 1, \dots, K$ . Assuming zero mean random variables for simplicity, the corresponding optimization problem can be written as follows:

$$\begin{aligned} & \arg \max_{\{\mathbf{p}^{(k)}\}_{k=1}^K} \rho(v_1, v_2, \dots, v_K) \\ & \text{s.t. } \mathbb{E}[v_k^2] = 1, k \in \{1, \dots, K\} \end{aligned} \quad (18)$$

where  $v_k \triangleq \langle \mathbf{p}^{(k)}, \mathbf{x}^{(k)} \rangle$  and  $\rho(v_1, v_2, \dots, v_K) = \mathbb{E}[v_1 v_2 \cdots v_K]$ .

A summary of the algorithm presented in [32] for obtaining the generalized canonical directions is outlined in Algorithm 3. Note that in Step 4, Algorithm 3 uses a low-rank tensor decomposition to solve the optimization problem (18). To compute this decomposition, one can use the alternating least squares (ALS) algorithm [42].

We repeat the same steps done in Section IV-B to obtain the generalized canonical direction  $\mathbf{p}^{(k)}$  at the point  $\mathbf{x}_i^{(k)}$ , namely,  $\mathbf{p}^{(k)}(\mathbf{x}_i^{(k)})$ . In other words, we apply Algorithm 3 only to the neighborhood of the  $i$ th realizations,  $\mathcal{X}_i^{(k)}$ . In addition, by repeating the same steps as in the proof of Proposition 2, we arrive to the following result.

*Corollary 7:* In the multiple observation case, the Euclidean distance between any two (scalar) realizations  $z_i$  and  $z_j$  of the

---

### Algorithm 3: Linear TCCA.

---

**Input:**  $K$  sets of observations  $\mathcal{X}^{(k)}$ ,  $1 \leq k \leq K$

**Output:** The canonical directions  $\mathbf{p}^{(1)}, \mathbf{p}^{(2)}, \dots, \mathbf{p}^{(K)}$

---

- 1) For each set  $\mathcal{X}^{(k)} = \{\mathbf{x}_i^{(k)}\}_{i=1}^N$ , compute the covariance matrix  $\Sigma_{kk}$ .
- 2) Compute the covariance tensor

$$\mathcal{C}_{12\dots K} = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i^{(1)} \otimes \mathbf{x}_i^{(2)} \otimes \dots \otimes \mathbf{x}_i^{(K)}$$

- 3) Compute:

$$\mathcal{T} = \mathcal{C}_{12\dots K} \times_1 \Sigma_{11}^{-\frac{1}{2}} \times_2 \Sigma_{22}^{-\frac{1}{2}} \times \dots \times_K \Sigma_{KK}^{-\frac{1}{2}}$$

- 4) Apply a rank-1 tensor approximation to  $\mathcal{T}$  by solving:

$$\arg \min_{\rho, \{\mathbf{p}^{(k)}\}_{k=1}^K} \left\| \mathcal{T} - \rho \mathbf{p}^{(1)} \otimes \mathbf{p}^{(2)} \otimes \dots \otimes \mathbf{p}^{(K)} \right\|_F$$

where  $\|\cdot\|_F$  is the Frobenius norm, and obtain the canonical directions.

---

random variable  $z$  is given by:

$$\begin{aligned} \|z_i - z_j\|_2^2 &= \left( \mathbf{x}_i^{(1)} - \mathbf{x}_j^{(1)} \right)^T \mathbf{A} \left( \bar{\mathbf{x}}_{ij}^{(1)} \right) \left( \mathbf{x}_i^{(1)} - \mathbf{x}_j^{(1)} \right) \\ &+ \mathcal{O} \left( \left\| \mathbf{x}_i^{(1)} - \mathbf{x}_j^{(1)} \right\|^4 \right) \end{aligned}$$

where  $\bar{\mathbf{x}}_{ij}^{(1)} \triangleq (\mathbf{x}_i^{(1)} + \mathbf{x}_j^{(1)})/2$  and  $\mathbf{A}(\bar{\mathbf{x}}_{ij}) \triangleq (\mathbf{p}^{(1)}(\bar{\mathbf{x}}_{ij}^{(1)})) (\mathbf{p}^{(1)}(\bar{\mathbf{x}}_{ij}^{(1)})^T$ .

Note that the common variable  $z$  in this case is restricted to be a scalar, due to the limitation of the TCCA algorithm. We note that similarly to Proposition 2, Corollary 7 can be analogously formulated based on realizations of  $\mathbf{x}^{(k)}$  (instead of  $\mathbf{x}^{(1)}$ ) for any  $1 \leq k \leq K$ .

Obtaining the global metric from the local metric is achieved similarly to the case where  $K = 2$  and is described in Section IV-B. Here as well, we use Diffusion Maps with a Gaussian kernel. The overall algorithm for the multimodal case ( $K > 2$ ) is presented in Algorithm 4.

## VII. EXPERIMENTAL RESULTS

In this section we provide several simulations to illustrate the theory and present the results attained by the proposed algorithms. In all the simulations, we use Matlab's built-in CCA implementation 'canoncorr'. We note that this implementation uses a stable implementation of the CCA algorithm as proposed in [41], rather than the more naïve implementation which is described in Section III.

### A. Local Metric Comparison

We generate  $N = 400$  realizations  $\{z_i\}_{i=1}^N$  of a two dimensional random variable with uniform distribution in the  $[0, 2]^2$  square. To compare only the metric estimation, we use an observation with no observation-specific variables. According to

**Algorithm 4:** Diffusion Maps of  $K$  Datasets.**Input:**  $K$  sets of observations  $\mathcal{X}^{(k)}$ ,  $1 \leq k \leq K$ **Output:** Low dimensional parametrization of the common variable  $z$ .

- 1) For each sample  $(\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)}, \dots, \mathbf{x}_i^{(K)}) \in (\mathcal{X}^{(1)}, \mathcal{X}^{(2)}, \dots, \mathcal{X}^{(K)})$ :
  - a) Construct the subsets  $(\mathcal{X}_i^{(1)}, \mathcal{X}_i^{(2)}, \dots, \mathcal{X}_i^{(K)})$  by choosing all samples  $(\mathbf{x}_j^{(1)}, \mathbf{x}_j^{(2)}, \dots, \mathbf{x}_j^{(K)})$  such that the  $k$ th coordinate  $\mathbf{x}_j^{(k)}$  is in the neighborhood of  $\mathbf{x}_i^{(k)}$ .
  - b) Apply (linear) TCCA to the sets  $(\mathcal{X}_i^{(1)}, \mathcal{X}_i^{(2)}, \dots, \mathcal{X}_i^{(K)})$  and obtain the vector  $\mathbf{p}^{(k)}(\mathbf{x}_i)$ .
- 2) For each two observations  $\mathbf{x}_i^{(k)}, \mathbf{x}_j^{(k)} \in \mathcal{X}^{(k)}$ , compute the distances  $\tilde{D}_{ij}$  according to

$$\tilde{D}_{ij} = (\mathbf{x}_i^{(k)} - \mathbf{x}_j^{(k)})^T \mathbf{A}^{(k)} (\mathbf{x}_i^{(k)} - \mathbf{x}_j^{(k)})$$

where  $\mathbf{A}^{(k)}(\mathbf{x}_i^{(k)}) \triangleq (\mathbf{p}^{(k)}(\mathbf{x}_i^{(k)}))(\mathbf{p}^{(k)}(\mathbf{x}_i^{(k)}))^T$ .

- 3) Apply Diffusion Maps:
  - a) Construct the kernel:  $W_{ij} = \exp(-\tilde{D}_{ij}/\sigma)$ , where  $\sigma$  is set to the median values  $\tilde{D}_{i,j}$ ,  $\forall i, j$ .
  - b) Normalize the kernel  $\mathbf{M} = \mathbf{\Omega}^{-\frac{1}{2}} \mathbf{W}^T \mathbf{W} \mathbf{\Omega}^{-\frac{1}{2}}$ , where  $\mathbf{\Omega}$  is a diagonal matrix with  $\Omega_{ii} = \sum_j (W^T W)_{ij}$ .
  - c) Compute the eigenvectors and eigenvalues of the matrix  $\mathbf{M}$ , i.e.,  $\mathbf{M} = \mathbf{\Psi} \mathbf{S} \mathbf{\Psi}^{-1}$ .
- 4) Form the parametrization of  $z_i$ ,  $\forall i = 1, \dots, N$  using the the eigenvector (the left most column of  $\mathbf{\Psi}$ ) associated with the largest eigenvalue (excluding the trivial eigenvector).

Proposition 4, only a single observation is needed, and we can compare between  $D_{ij}$  as defined in (16) and the distance defined in [34], which we denote here as  $Q_{ij}$ , namely:

$$Q_{ij} = \frac{1}{2} (\mathbf{x}_i - \mathbf{x}_j)^T [\mathbf{A}(\mathbf{x}_i) + \mathbf{A}(\mathbf{x}_j)] (\mathbf{x}_i - \mathbf{x}_j).$$

By simulating the following nonlinear observation function:

$$\mathbf{x} = f(\mathbf{z}) = \begin{bmatrix} z_1^2 - z_2 \\ z_1 + \sqrt{z_2} \end{bmatrix}$$

we obtain a set of  $N = 400$  observations  $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^N$ . Fig. 1 depicts (a) the hidden variables  $\mathcal{Z} = \{z_i\}$  and (b) the observations  $\mathcal{X} = \{\mathbf{x}_i\}$ .

Fig. 2 shows the estimated distances as a function of the true distances. In Fig. 2(a), we plot the estimated metric based on [34], and in Fig. 2(b) we plot the estimated metric based on 16. We can see that for small Euclidean distances (small values on the x-axis), both estimated metrics are accurate. For large Euclidean distances, the estimated metric based on the mid-

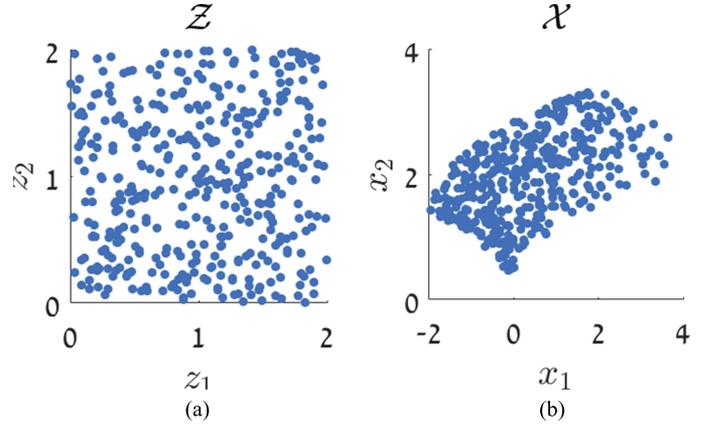


Fig. 1. (a) The hidden random variables  $\mathcal{Z} = \{z_i\}_{i=1}^N$ . (b) The observations  $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^N$ .

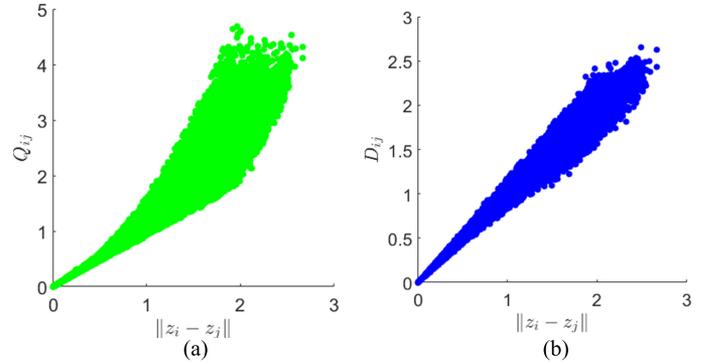


Fig. 2. Comparisons of the Euclidean metric estimations. (a) The metric estimation based on [34]. (b) The metric estimation based on the middle point (16).

dle point maintains a linear correlation with the true distance, whereas the estimated metric proposed in [34] exhibits large error.

### B. Neighborhood Selection

In this simulation, we test the derived metric 16 using two implementations. We draw  $N = 500$  realizations of

$$z, \epsilon, \eta \sim U[0, 1]$$

and then, we obtain the  $N$  data points using two observation functions:

$$\mathbf{x} = \begin{bmatrix} \epsilon - 2z^2 \\ z + \epsilon^2 \end{bmatrix}, \mathbf{y} = \begin{bmatrix} z - \eta \\ \eta + 2z^3 \end{bmatrix}$$

A visualization of the data appears in Fig. 3, where the data are colored by the common hidden variable  $z$ .

The first implementation is based on neighborhoods, which are defined according to K-nn ( $K = 50$ ). The second implementation uses the analytic expression of the local CCA. Fig. 4 shows the performance of both simulations. Indeed, using the analytic expression results in superior performance compared to using the K-nn neighborhoods. Yet, the implementation based

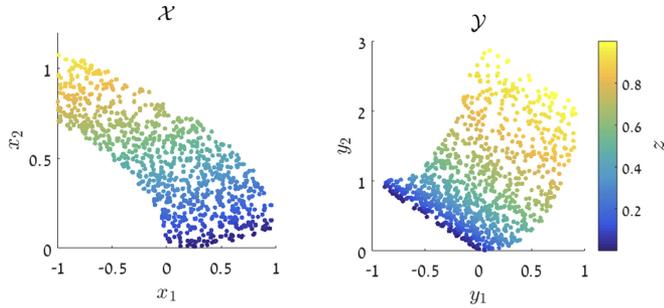


Fig. 3. The non-linear views  $\boldsymbol{x}$  (left) and  $\boldsymbol{y}$  (right). The data are colored by the common hidden variable  $z$ .

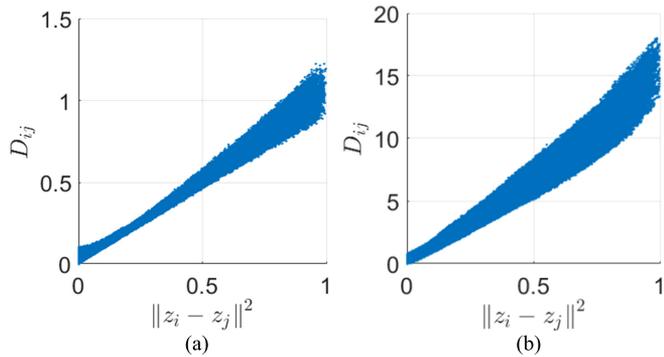


Fig. 4. (a) The estimated distances using the analytic expression of the Local CCA. (b) The estimated distances using K-nn ( $K = 50$ ) neighborhoods.

on K-nn attains an accurate Euclidean estimates, comparable to the estimation based on the analytic expression.

### C. Sleep Stage Recovery

In this subsection we demonstrate the applicability of Algorithm 2 to real medical signals. Specifically, we address the problem of sleep stage identification. Typically, for this purpose, data are collected in sleep clinics in multiple multimodal sensors, and then, analyzed by a human expert. The data we used is available online in [43] and described in detail in [44]. A single patient’s night recording contains several measurements including two electroencephalogram (EEG) channels and one electrooculography (EOG) channel sampled at 100 [Hz]. To each set of observations from each channel, of length  $N = 2^{20}$ , we apply the scattering transform [45] as proposed in [46] and obtain 4,096 short-time frames of observations of dimension 12. In the first experiment we applied both Diffusion Maps and Algorithm 2 to two EEG channels (FpzCZ and PzOZ). In the Diffusion Maps application, data from the two channels were concatenated to form a single set of observations.

Fig. 5 displays the embedding using the 3 dominant eigenvectors. In Fig. 4(a) we present the embedding obtained by Diffusion Maps with two concatenated EEG channels, and in Fig. 4(b) we present the embedding attained by Algorithm 2. The points in the figures are colored by the 6 sleep stages labeled by experts. Sleep stages 1 to 4 represent the sleep depth (1 - shallow sleep, 4 - deep sleep), and sleep stages 5 and 6 represent awake and rapid eye movement (REM) stages, respec-

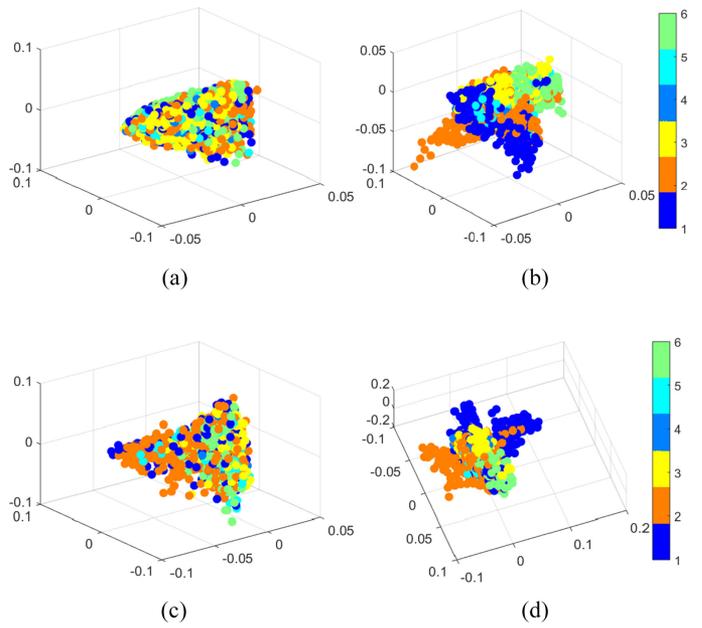


Fig. 5. The embedding of two EEG recordings (a)-(b) and of EEG and EOG recording (c)-(d) via the three dominant eigenvectors. (a) The representation obtained by Diffusion Maps. (b) The representation obtained by Algorithm 2. (c) The representation obtained by Diffusion Maps. (d) The representation obtained by Algorithm 2. The points in the figures are colored by the 6 different sleep stages.

tively. We observe that Algorithm 2 obtains a distinct clustering of the different sleep stages, whereas Diffusion Maps does not recover any particular structure. To further demonstrate the applicability of Algorithm 2 to multiple multimodal signals, we repeated the experiment and applied the two algorithms to data arising from an EEG channel and an EOG channel. The results are depicted in Fig. 4(c) and (d). Here as well, we observe that Algorithm 2 enables distinct clustering according to the sleep stage, whereas the embedding attained by Diffusion Maps is unstructured. To test the obtained representation we apply a leave-one-out simple K-nn classification, with  $K = 9$ . Although we only display the dominant 3 eigenvectors, for the classification we use the 100 dominant eigenvectors. Fig. 6 depicts the classification confusion matrices. In Fig. 6(a) the classification was applied to the embedding obtained by Algorithm 2, and in Fig. 6(b) the classification was applied to the embedding obtained by Diffusion Maps. In the diffusion maps application, the multi-modal data was concatenate. One can notice that the classification based on Algorithm 2 achieves a 90% success rate, whereas the naïve Diffusion Maps approach achieves only 55.2%. We note that although the classification performance is comparable to the results reported in the literature on sleep stages classification, the leave-one-out K-nn classification is not practical for classifying streaming data and it was done only to evaluate the structure of the obtained embedding by our method in comparison to a baseline embedding. We emphasize that our goal in these two experiments is merely to demonstrate the applicability of the proposed algorithm to real signals. Indeed, we have shown that Algorithm 2 is capable of extracting a physiological process underlying two (possibly multimodal) channels

Output Class	Stage 1	343 4.2%	22 0.3%	7 0.1%	1 0.0%	13 0.2%	3 0.0%	88.2% 11.8%
	Stage 2	53 0.6%	2985 36.5%	193 2.4%	110 1.3%	16 0.2%	35 0.4%	88.0% 12.0%
	Stage 3	17 0.2%	73 0.9%	1672 20.5%	89 1.1%	10 0.1%	0 0.0%	89.8% 10.2%
	Stage 4	1 0.0%	43 0.5%	106 1.3%	1841 22.5%	7 0.1%	8 0.1%	91.8% 8.2%
	Awake	4 0.0%	5 0.1%	1 0.0%	0 0.0%	107 1.3%	0 0.0%	91.5% 8.5%
	REM	0 0.0%	4 0.0%	0 0.0%	0 0.0%	0 0.0%	406 5.0%	99.0% 1.0%
	Stage 1	82.1% 17.9%	95.3% 4.7%	84.5% 15.5%	90.2% 9.8%	69.9% 30.1%	89.8% 10.2%	90.0% 10.0%
		S1	S2	S3	S4	Awake	REM	S1

(a)

Output Class	Stage 1	108 1.3%	58 0.7%	43 0.5%	6 0.1%	19 0.2%	29 0.4%	41.1% 58.9%
	Stage 2	167 2.0%	2189 26.8%	762 9.3%	492 6.0%	59 0.7%	116 1.4%	57.8% 42.2%
	Stage 3	84 1.0%	440 5.4%	767 9.4%	294 3.6%	46 0.6%	38 0.5%	46.0% 54.0%
	Stage 4	4 0.0%	336 4.1%	349 4.3%	1232 15.1%	2 0.0%	58 0.7%	62.2% 37.8%
	Awake	7 0.1%	9 0.1%	10 0.1%	3 0.0%	9 0.1%	2 0.0%	22.5% 77.5%
	REM	48 0.6%	100 1.2%	49 0.6%	14 0.2%	18 0.2%	209 2.6%	47.7% 52.3%
	Stage 1	25.8% 74.2%	69.9% 30.1%	38.7% 61.3%	60.4% 39.6%	5.9% 94.1%	46.2% 53.8%	55.2% 44.8%
		S1	S2	S3	S4	Awake	REM	S1

(b)

Fig. 6. The confusion matrices of a leave-one-out K-nn ( $K = 9$ ) classification based on (a) the eigenvectors obtained by Algorithm 2 (b) the eigenvectors obtained by Diffusion Maps.

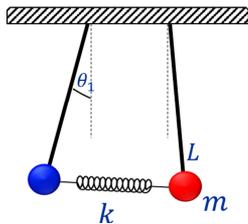


Fig. 7. Illustration of the setup of the coupled pendulums system.

without any prior knowledge. Further examination as well as comparison to state of the art automatic identification methods will be performed in future work.

#### D. Coupled Pendulum

In this experiment we simulate a coupled pendulum model. This model consists of two simple pendulums with lengths  $L_1$  and  $L_2$  and masses  $m_1$  and  $m_2$ , which are connected by a spring as shown in Fig. 7. For simplicity we set the same length and

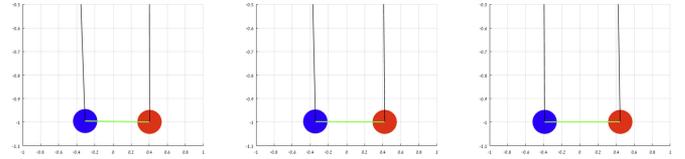


Fig. 8. An example of 3 snapshots of the coupled pendulum system.

the same mass for both pendulums, namely  $L_1 = L_2 = L$  and  $m_1 = m_2 = m$ . Let  $u^{(i)}(t)$  and  $w^{(i)}(t)$  denote the horizontal position and vertical position of the  $i$ th pendulum, respectively. Note that a close-form expression for the positions cannot be derived for the general case. Yet, in the case of small perturbations around the equilibrium point, we can consider a linear regime. Accordingly, let  $\theta^{(i)}(t) = \arctan(u^{(i)}(t)/w^{(i)}(t))$  be the angle between the  $i$ th pendulum and the vertical axis, and assume that  $w^{(1)}(t) = w^{(2)}(t) = -L$ , and  $\sin(\theta^{(i)}) \approx \theta^{(i)}$ . The ordinary differential equation (ODE) representing the horizontal position under the linear regime is given by:

$$\begin{cases} m\ddot{u}^{(1)} = -\frac{mg}{L}u^{(1)} - k(u^{(1)} - u^{(2)}) \\ m\ddot{u}^{(2)} = -\frac{mg}{L}u^{(2)} + k(u^{(1)} - u^{(2)}) \end{cases} \quad (19)$$

where  $\ddot{u}$  is the second derivative of  $u$ ,  $g$  is the gravity of earth, and  $k$  is the spring constant. For the following initial conditions:

$$\dot{u}_1(0) = 0, u_1(0) = \delta, \dot{u}_2(0) = 0, u_2(0) = 0$$

where  $0 < \delta \in \mathbb{R}$  is assumed to be sufficiently small to satisfy the linear regime, the closed-form solution of the ODE (19) is given by:

$$\begin{cases} u^{(1)}(t) = \frac{1}{2}\delta \cos(\omega_1 t) + \frac{1}{2}\delta \cos(\omega_2 t) \\ u^{(2)}(t) = \frac{1}{2}\delta \cos(\omega_1 t) - \frac{1}{2}\delta \cos(\omega_2 t) \end{cases} \quad (20)$$

where

$$\omega_1 = \sqrt{\frac{g}{L}}, \quad \omega_2 = \sqrt{\frac{g}{L} + \frac{2k}{m}}$$

This example suits our purposes, since the horizontal displacement of each pendulum  $u^{(i)}(t)$  is a linear combination of two harmonic motions with the *common* frequencies  $\omega_1$  and  $\omega_2$ . In other words, the horizontal displacement of each pendulum can be viewed as a different observation of the same common harmonic motion.

To further demonstrate the power of our method, we assume that we do not have direct access to the horizontal displacement. Instead, we generate movies of the motion of the coupled pendulum in the linear regime. Consequently, on the one hand, we have a definitive ground truth described by the solution of the ODE of the system (the harmonic motion with the two frequencies  $\omega_1$  and  $\omega_2$ ). On the other hand, we only have access to high-dimensional nonlinear observations of the system, and we do not assume any prior model knowledge. Three snapshots of the entire system are displayed in Fig. 8.

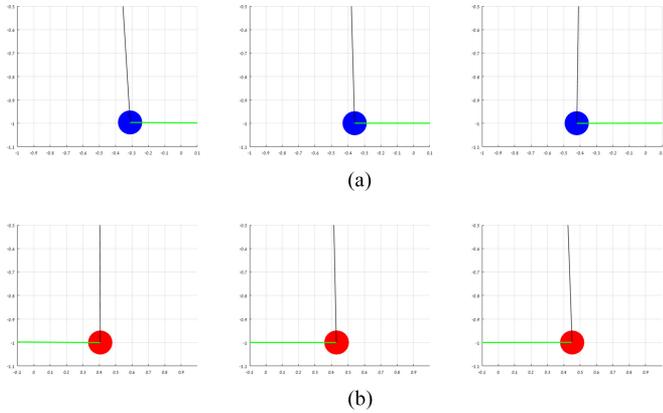


Fig. 9. An example of 3 frames of the each movie: (a) the left pendulum, and (b) the right pendulum.

This model is used to test our method in two scenarios. In the first scenario, we generate two movies of the two pendulums without any other features. In the second scenario, we generate two movies of the two pendulums, where each movie also contains an additional pendulum that represents an observation-specific geometric noise.

1) *Case I – Coupled Pendulum*: We generate two movies, each is 5 seconds long with  $N = 400$  frames (namely, sampling interval of  $T_s = 0.0125$  s) of each of the pendulums in the couple pendulum system oscillating in a linear regime as demonstrated in Fig. 9.

Let  $\mathbf{m}_i^{(1)} \in \mathbb{R}^{800}$  and  $\mathbf{m}_i^{(2)} \in \mathbb{R}^{800}$  be two column stack vectors consisting of the pixels of the  $i$ th frame of the movies of the left and right pendulums, respectively. Let  $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^N$  and  $\mathcal{Y} = \{\mathbf{y}_i\}_{i=1}^N$  be two sets of observations, which are random projections of the frames of the movies. In the words, each observation is given by  $\mathbf{x}_i = \mathbf{F}\mathbf{m}_i^{(1)}$  and  $\mathbf{y}_i = \mathbf{G}\mathbf{m}_i^{(2)}$ , where  $\mathbf{F} \in \mathbb{R}^{200 \times 800}$  and  $\mathbf{G} \in \mathbb{R}^{200 \times 800}$  are two fixed matrices with (approximately) orthonormal columns, drawn independently (once) from a Gaussian distribution. These observations/projections represent two different modalities in two different spaces.

We apply Algorithm 2 to  $\mathcal{X}$  and  $\mathcal{Y}$ , where we use 8 adjacent projected frames (in time), i.e.,  $\mathcal{X}_i = \{\mathbf{x}_j\}_{j=i-3}^{i+4}$  and  $\mathcal{Y}_i = \{\mathbf{y}_j\}_{j=i-3}^{i+4}$ , as the subset of each observation.

We compare our method to 3 different algorithms: (i) Diffusion Maps with Euclidean metric (once with a single view  $\mathcal{X}$  and once with the average kernel of both views  $\mathcal{X}$  and  $\mathcal{Y}$ ), (ii) KCCA, and (iii) Alternating Diffusion Maps (with Euclidean metric) [12]. In the Diffusion Maps algorithms, we view the nontrivial eigenvector associated with the largest eigenvalue as the parametrization of the system. In the KCCA algorithm we use the obtained projections as the system parametrization. We display the Fourier transform of the derived parametrization in Fig. 10. In each sub-figure the blue line is the Fourier transform of the parametrization and the two vertical red dashed lines are the two frequencies of the coupled pendulum system:  $\omega_1$  and  $\omega_2$ . Fig. 10(a) displays the output of Diffusion Maps with the Euclidean metric using the set of observations  $\mathcal{X}$  from only one

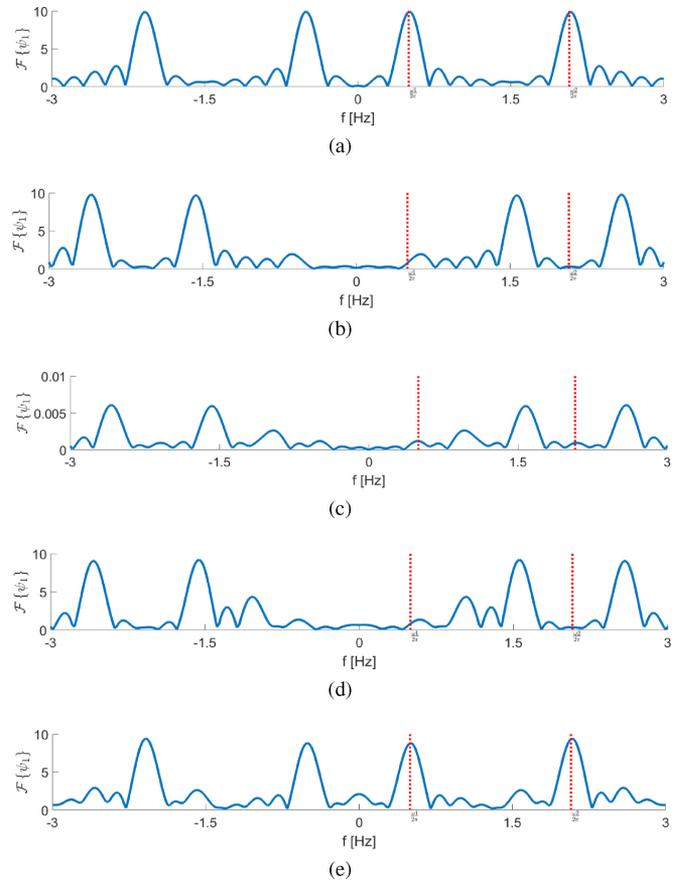


Fig. 10. The Fourier transform of (a) the dominant eigenvector attained by Diffusion Maps based on a single set  $\mathcal{X}$ , (b) the dominant eigenvector attained by Diffusion Maps based on the average kernel of both sets  $\mathcal{X}$ ,  $\mathcal{Y}$ , (c) the projection attained by KCCA, (d) the dominant eigenvector attained by Alternating Diffusion Maps, and (e) the dominant eigenvectors attained by Algorithm 2. The blue curves are the Fourier transforms of the representations and the vertical dashed lines represent the two frequencies of the system  $\omega_1$  and  $\omega_2$ .

movie. Fig. 10(b) displays the output of Diffusion Maps with the Euclidean metric using the two sets of observations  $\mathcal{X}$  and  $\mathcal{Y}$ . In this case we average the two kernels into one kernel, by forming  $W_{ij} = \frac{1}{2}(W_{ij}^{(1)} + W_{ij}^{(2)})$  where  $\mathbf{W}^{(1)}$  and  $\mathbf{W}^{(2)}$  are the kernels obtain from each view respectively. Fig. 10(c) displays the output of KCCA. Fig. 10(d) displays the output of Alternating Diffusion Maps. Finally, Fig. 10(e) displays the output of Algorithm 2, i.e., Diffusion Maps with  $D_{ij}$  (13) as its metric. We note that only one nontrivial eigenvector is presented, since, as shown in (20), the displacement of each pendulum, and hence, the projected frames of each movie, can be represented by a single scalar  $\theta$ , which is the angle of the pendulum with respect to the equilibrium axis. In other words, the coupled pendulum system can be described using a low-dimensional representation conveyed by the dominant nontrivial eigenvector.

As we can see in Fig. 10, both in the result obtained by Diffusion Maps (with single view) as well as in the result obtained by our method, the presented eigenvector contains the frequencies of the coupled pendulum system:  $\omega_1$  and  $\omega_2$ . In contrast, the eigenvector attained by Alternating Diffusion Maps and the projections attained by KCCA do not contain the true frequen-

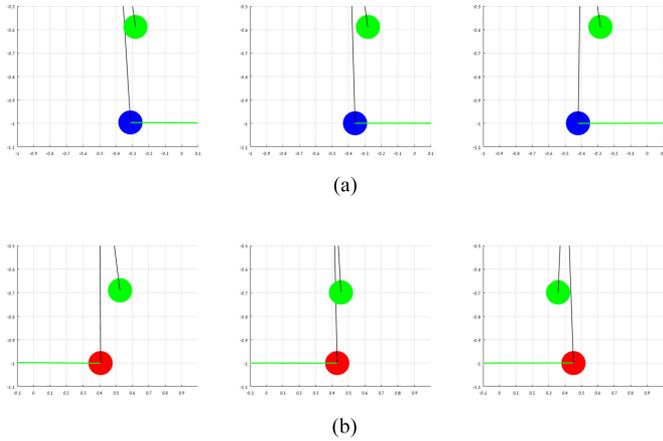


Fig. 11. An example of 3 frames of the noisy movie. (a) The left movie which captures the left coupled pendulum (blue) and an additional simple pendulum (green). (b) The right movie which captures the right coupled pendulum (red) and an additional simple pendulum (green).

cies of the coupled pendulum system. Specifically, we show in Fig. 10(a) that the true frequencies can be extracted simply by applying diffusion maps to one of the observation sets. Consequently, we remark that this experiment serves only as a reference; it implies that in this noiseless case each of the sets carries the full information on the system, and as a result, these frequencies are common to both sets. In the next section, we introduce noise and show that in the noisy case our algorithm is essential.

2) *Case II – Coupled Pendulum With Observation-Specific:* We repeat the experiment described in Section VII-D1 with additional observation-specific. In this experiment, an additional simple pendulum is added to each movie as demonstrated in Fig. 11. Note that the two extra pendulums, one in each movie, oscillate in different frequencies:  $\omega_3 = \frac{1}{5}\omega_1$  and  $\omega_4 = 4\omega_1$ . In other words, we now have 4 different frequencies in the movies, yet only 2 of them ( $\omega_1$  and  $\omega_2$ ) are common to both observations. We further note that for the purpose of recovering the common variables, the presence of the sensor-specific variables is considered as structured noise.

We apply Algorithm 2 with the same selection of subsets  $\mathcal{X}_i$  and  $\mathcal{Y}_i$ . As in Section VII-D1, we compare our method to the same 4 algorithms.

Fig. 12 is similar to Fig. 10, where we display the Fourier transforms of the parametrizations obtained by the algorithms. In each figure the blue curve is the Fourier transform of the parametrization, the two red dashed vertical lines are the two frequencies of the coupled pendulum  $\omega_1$  and  $\omega_2$ , and the two green dashed vertical lines are the two frequencies of the simple pendulums  $\omega_3$  and  $\omega_4$ . Fig. 12(a) displays the output of Diffusion Maps with the Euclidean metric using the set of observations  $\mathcal{X}$  from only one movie. Fig. 10(b) displays the output of Diffusion Maps with the Euclidean metric using the two sets of observations  $\mathcal{X}$  and  $\mathcal{Y}$ . Fig. 12(c) displays the output of KCCA. Fig. 12(d) displays the output of Alternating Diffusion Maps. Finally, Fig. 12(e) displays the output of Algorithm 2, i.e., Diffusion Maps with the distances  $D_{ij}$ . As we can see in

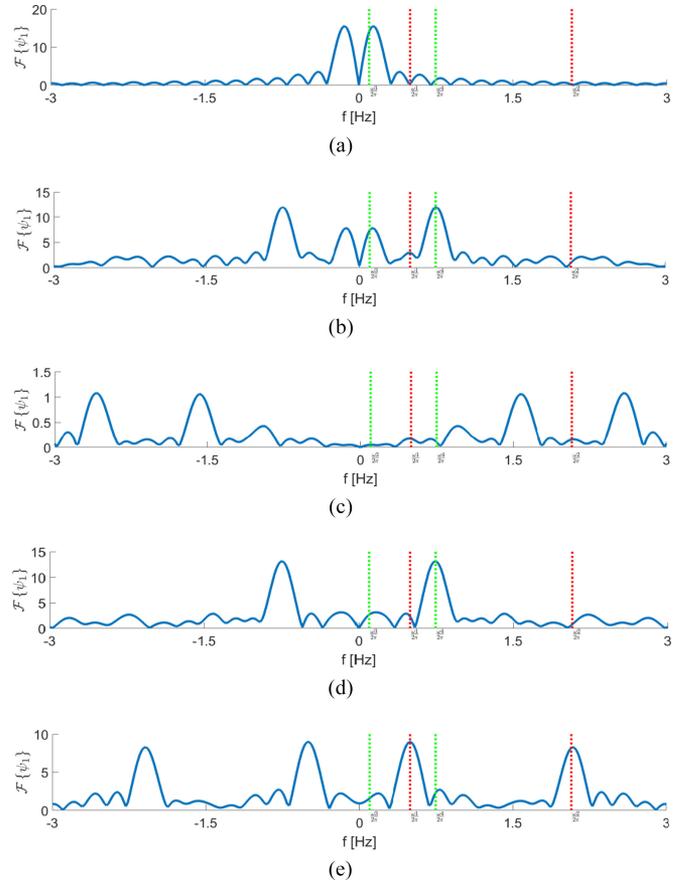


Fig. 12. The output of the 5 algorithms applied to the coupled pendulum movies with additional observation-specific pendulums (variables): (a) Diffusion Maps (single view), (b) Diffusion Maps (both views), (c) KCCA, (d) Alternating Diffusion Maps, and (e) Algorithm 2. The Fourier transforms of the representations are displayed as blue curves. The red and green dashed vertical lines represent the frequencies of the coupled pendulum and the simple uncoupled pendulums, respectively.

Fig. 12, only in the result obtained by Algorithm 2, the dominant eigenvector contains the frequencies of the coupled pendulum system  $\omega_1$  and  $\omega_2$  as desired. The eigenvectors attained by Diffusion Maps, by Alternating Diffusion Maps and the projections attained by KCCA, do not contain the true frequencies of the coupled pendulum system.

3) *AWGN:* In order to demonstrate the ability of our algorithm to cope with AWGN we repeated the experiment described in Subsection VII-D2. In this experiment we added to each observation an additive white Gaussian noise. The SNR in both views is set to 7 [dB]. Here as well, we applied Algorithm 2 to the noisy observations. Fig. 13 displays the Fourier transform of the obtained parametrization using Algorithm 2. In this (noisy) experiment as shown in the figure, the frequency content of the obtained parametrization contains the coupled pendulum frequencies while the observation-specific variables are not detected, as desired.

4) *Run Time:* To test the performance of Algorithm 2 and specifically the relationship between the computational complexity and the algorithm accuracy, we applied it to the same experiment described in Subsection VII-D2. We repeated the

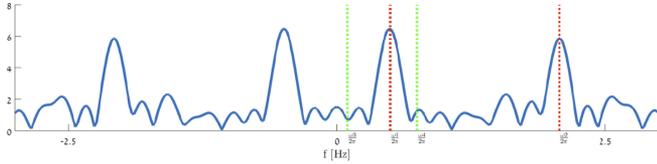


Fig. 13. The output of Algorithm 2 applied to the coupled pendulum movies with both observation-specific pendulums (variables) and AWGN. The blue curve is the Fourier transform of the obtained representation using Algorithm 2. The red dashed lines are the frequencies of the coupled pendulum. The green dashed lines are the frequencies of the observation-specific pendulums.

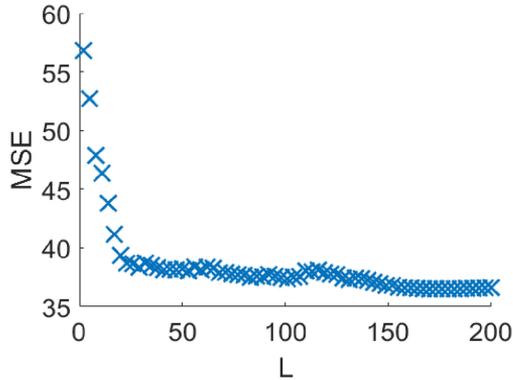


Fig. 14. Algorithm 2 accuracy as a function of  $L$  – the number of local CCA applications in the algorithm.

applications of Algorithm 2 with different values of  $L$ . For each value of  $L$ , the experiment was repeated 10 times with different subsets of measurements of size  $L$ . The complexity of each local CCA is  $O(nd^2) + O(d^3)$ , where  $n$  is the size of the subsets, and  $d$  is the maximum of the dimensions of the input vectors from the two views. In this simulation we have  $N = 400$  frames,  $d = 800$  and  $n = 8$ . Algorithm 1, applies CCA  $N^2$  times, namely, we have an overall complexity:  $O(N^2nd^2) + O(N^2d^3)$ . Algorithm 2, applies CCA only  $L$  times, namely, we have an overall complexity:  $O(Lnd^2) + O(Ld^3)$ . Thus, in this experiment, for  $L = 30$ , the number of local CCA applications in Algorithm 1 compared to Algorithm 2 is larger by a factor of  $\frac{N^2}{L} = \frac{400^2}{30} \approx 5333$ . Fig. 14 shows the MSE (averaged over the different subsets of size  $L$ ) performance as a function of  $L$ . We observe that for a relatively small value of  $L$ , which results in a significant reduction of the computational load (e.g.  $L = 30$ ), we get performance that is comparable to the asymptotic performance obtained by using all the measurements (similarly to Algorithm 1). This demonstrates the potential applicability of Algorithm 2 to problems with large data sets.

### E. Multiple Observations of Rotating Icons

In this simulation we show that Algorithm 4 allows for the accurate parametrization of the common variable underlying  $K = 3$  nonlinear high-dimensional observations. We generate three high-dimensional movies containing four rotating icons: Super Mario, Mushroom, Turtle and Flower. Each movie

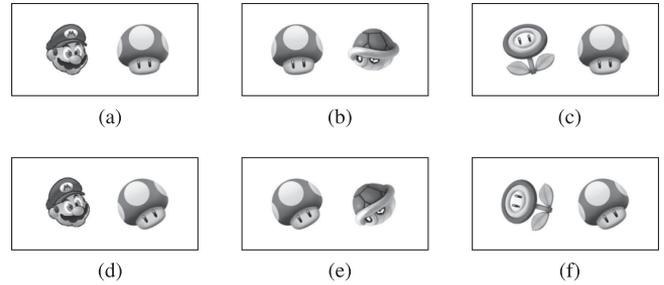


Fig. 15. Two frames of each of the movies: (a) and (d) are from the first movie, (b) and (e) from the second movie, and (c) and (f) from the third movie. Mushroom is common to all the movies while the other icons are specific for each movie.

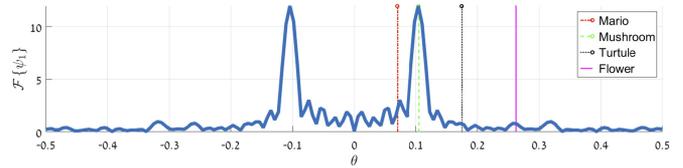


Fig. 16. The Fourier transform of the representation of the common variable obtained by Algorithm 4. The vertical dashed lines represent the true frequencies of the rotating icons.

captures only two icons. In the movies, each icon rotates in a constant angular speed: the angular speeds of Super Mario, Mushroom, Turtle and Flower are  $4^\circ$ ,  $6^\circ$ ,  $10^\circ$ , and  $15^\circ$  per frame, respectively, as demonstrated in Fig. 15. Notice that only Mushroom appears in all the movies, and hence, the angular speed of Mushroom is the hidden common variable  $z$ , whereas the angular speeds of the other icons are the hidden observation-specific variables  $\epsilon^{(k)}$ , for  $1 \leq k \leq K$ . Two frames of each movie are depicted for illustration in Fig. 15.

Each set of nonlinear high-dimensional observations  $\mathcal{X}^{(k)} = \{\mathbf{x}_i^{(k)}\}_{i=1}^N$  consists of  $N = 300$  frames. We apply Algorithm 4, where in step 1(a), the subsets  $\mathcal{X}_i^{(k)}$  consist of the frames in a time window of length 7 around  $\mathbf{x}_i^{(k)}$ . Since, the desired parametrization should convey the fact that the common variable in the sets is the angular speed of Mushroom, and thus, it should be periodic with the same period, we apply the Fourier transform to the first column of  $\Psi$  and present it in Fig. 16. In Fig. 16, the true frequencies of Super Mario, Mushroom, Turtle and Flower are marked by vertical red, green, black and pink dashed lines, respectively.

Fig. 16 shows that indeed the proposed algorithm identifies the frequency of Mushroom (the common variable underlying all observation sets), whereas the frequencies of the observation-specific Super Mario, Turtle and Flower are completely missing, as expected.

To further demonstrate the capabilities of our method, we repeat the simulation in a more complex setting. Here, each pair of movies contains two common icons while only Mushroom is maintained as the common variable of all the movies. Two frames of the new movies are depicted for illustration in Fig. 17.

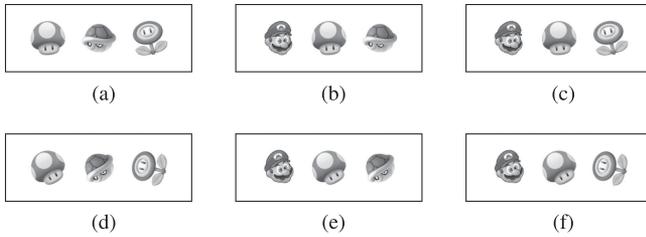


Fig. 17. Two frames from each movie: (a) and (d) are from the first movie, (b) and (e) from the second movie, and (c) and (f) from the third movie. Each two movies contain two common icons. Only Mushroom is common to all the movies.

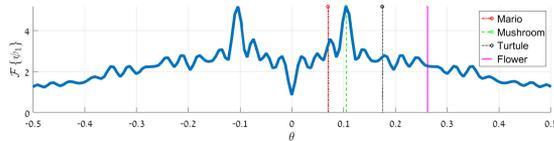


Fig. 18. The Fourier transform of the representation of the common variable obtained by Algorithm 4. The vertical dashed lines represent the true frequencies of the rotating icons.

For similar reasons as in the previous experiment, we apply the Fourier transform to the first column of  $\Psi$  and present it in Fig. 18. In Fig. 18, the true frequencies of Super Mario, Mushroom, Turtle and Flower are marked by vertical dashed red, green, black and pink lines, respectively. Despite the more complex setting, in which any two observation sets contain additional correlated “noise”, Fig. 18 shows that the proposed algorithm identifies the true frequency of the common variable (Mushroom).

In summary, in this simulation without assuming any prior knowledge on the structure and content of the data, our extended method successfully discovers the common variable hidden in *multiple* high-dimensional and nonlinear observations.

## VIII. CONCLUSIONS

In this paper, we have presented a new manifold learning method for extracting the common hidden variables underlying multimodal data sets. Our method does not assume prior knowledge on the system nor on the observed data and relies on a local metric, which is learned from data in an unsupervised manner. Specifically, we proposed a metric between observations based on local CCA and showed that this metric approximates the Euclidean distance between the respective hidden common variables.

The theoretical results were validated in simulations, where we demonstrated the accurate recovery of the hidden common variables from multiple complex and high-dimensional data sets. In addition, we showed that our method can be applied to various different types of observations and attain the same results without adjusting the algorithm to the specific observations at hand. The coupled pendulum system considered here is an example of a dynamical system with a definitive model and a closed-form solution in the linear case. We have shown that without any prior model knowledge our method

can obtain an accurate description of the solution solely from high-dimensional nonlinear observations. Note that our solution was obtained also when the observations contained “structured noise.”

The capability to obtain the close-form solution solely from observations enables us to demonstrate the power of our approach by carrying out empirical modeling of dynamical systems. One can further extend this to the analysis of the coupled pendulum system in more complex scenarios, such as, with different initial conditions and in nonlinear regimes. In such cases, closed-form solutions may no longer be available. Yet, from a data-driven point of view, our method is expected to attain an accurate description of the system from its observations. Importantly, since our method does not require prior rigid model assumptions, it can be applied to a broad variety of multimodal data sets lacking definitive models. Therefore, future work will address the extension of our analysis to various types of dynamical systems and empirical physics experiments.

## REFERENCES

- [1] D. Lahat, T. Adali, and C. Jutten, “Multimodal data fusion: An overview of methods, challenges, and prospects,” *Proc. IEEE*, vol. 103, no. 9, pp. 1449–1477, Sep. 2015.
- [2] H. Hotelling, “Relations between two sets of variates,” *Biometrika*, vol. 28, nos. 3/4, pp. 321–377, Dec. 1936.
- [3] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor, “Canonical correlation analysis: An overview with application to learning methods,” *Neural Comput.*, vol. 16, no. 12, pp. 2639–2664, 2004.
- [4] F. R. Bach and M. I. Jordan, “A probabilistic interpretation of canonical correlation analysis,” Dept. Statist., Univ. California, Berkeley, CA, USA, *Tech. Rep. 688*, 2005.
- [5] P. L. Lai and C. Fyfe, “Kernel and nonlinear canonical correlation analysis,” *Int. J. Neural Syst.*, vol. 10, no. 5, pp. 365–377, 2000.
- [6] W. Zheng, X. Zhou, C. Zou, and L. Zhao, “Facial expression recognition using kernel canonical correlation analysis (KCCA),” *IEEE Trans. Neural Netw.*, vol. 17, no. 1, pp. 233–238, Jan. 2006.
- [7] T. Melzer, M. Reiter, and H. Bischof, “Appearance models based on kernel canonical correlation analysis,” *Pattern Recognit.*, vol. 36, no. 9, pp. 1961–1971, 2003.
- [8] D. R. Hardoon, J. Mourao-Miranda, M. Brammer, and J. Shawe-Taylor, “Unsupervised analysis of fMRI data using kernel canonical correlation,” *NeuroImage*, vol. 37, no. 4, pp. 1250–1259, 2007.
- [9] V. R. de Sa, “Spectral clustering with two views,” in *Proc. Int. Conf. Mach. Learn. Workshop Learn. Multiple Views*, 2005, pp. 20–27.
- [10] B. Wang, J. Jiang, W. Wang, Z. Zhou, and Z. Tu, “Unsupervised metric fusion by cross diffusion,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 2997–3004.
- [11] B. Boots and G. Gordon, “Two-manifold problems with applications to nonlinear system identification,” in *Proc. 29th Int. Conf. Mach. Learn.*, 2012, pp. 623–630.
- [12] R. R. Lederman and R. Talmon, “Learning the geometry of common latent variables using alternating-diffusion,” *Appl. Comput. Harmon. Anal.*, 2016, to be published.
- [13] R. R. Lederman, R. Talmon, H. Wu, Y. Lo, and R. R. Coifman, “Alternating diffusion for common manifold learning with application to sleep stage assessment,” in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2015, pp. 5758–5762.
- [14] S. T. Roweis and L. K. Saul, “Nonlinear dimensionality reduction by locally linear embedding,” *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.
- [15] M. Belkin and P. Niyogi, “Laplacian eigenmaps for dimensionality reduction and data representation,” *Neural Comput.*, vol. 15, no. 6, pp. 1373–1396, 2003.
- [16] R. R. Coifman and S. Lafon, “Diffusion maps,” *Appl. Comput. Harmon. Anal.*, vol. 21, no. 1, pp. 5–30, 2006.
- [17] A. Singer and R. R. Coifman, “Non-linear independent component analysis with diffusion maps,” *Appl. Comput. Harmon. Anal.*, vol. 25, no. 2, pp. 226–239, 2008.

- [18] R. Talmon and R. R. Coifman, "Empirical intrinsic geometry for nonlinear modeling and time series filtering," *Proc. Nat. Acad. Sci.*, vol. 110, no. 31, pp. 12535–12540, 2013.
- [19] R. Talmon and R. R. Coifman, "Intrinsic modeling of stochastic dynamical systems using empirical geometry," *Appl. Comput. Harmon. Anal.*, vol. 39, no. 1, pp. 138–160, 2015.
- [20] R. Talmon, S. Mallat, H. Zaveri, and R. R. Coifman, "Manifold learning for latent variable inference in dynamical systems," *IEEE Trans. Signal Process.*, vol. 63, no. 15, pp. 3843–3856, Aug. 2015.
- [21] T. Berry and T. Sauer, "Local kernels and the geometric structure of data," *Appl. Comput. Harmon. Anal.*, Elsevier, vol. 40, no. 3, pp. 439–469, 2016.
- [22] D. Giannakis, "Dynamics-adapted cone kernels," *SIAM J. Appl. Dyn. Syst.*, vol. 14, no. 2, pp. 556–608, 2015.
- [23] V. R. De Sa, P. W. Gallagher, J. M. L., and V. L. Malave, "Multi-view kernel construction," *Mach. Learn.*, vol. 79, nos. 1/2, pp. 47–71, 2010.
- [24] A. Kumar, P. Rai, and H. Daume, "Co-regularized multi-view spectral clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 1413–1421.
- [25] Y. Y. Lin, T. L. Liu, and C. S. Fuh, "Multiple kernel learning for dimensionality reduction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 6, pp. 1147–1160, Jun. 2011.
- [26] H. Huang, Y. Chuang, and C. Chen, "Affinity aggregation for spectral clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 773–780.
- [27] B. Boots and G. Gordon, "Two-manifold problems with applications to nonlinear system identification," in *Proc. 29th Int. Conf. Mach. Learn. (ICML-12)*, 2012, pp. 623–630.
- [28] O. Lindenbaum, A. Yeredor, M. Salhov, and A. Averbuch, "Multiview diffusion maps," arXiv:1508.05550.
- [29] O. Lindenbaum, A. Yeredor, and M. Salhov, "Learning coupled embedding using multiview diffusion maps," in *Latent Variable Analysis and Signal Separation*. New York, NY, USA: Springer, 2015, pp. 127–134.
- [30] T. Michaeli, W. Wang, and K. Livescu, "Nonparametric canonical correlation analysis," in *Proc. Int. Conf. Mach. Learn.*, 2016.
- [31] G. Andrew, R. Arora, J. A. Bilmes, and K. Livescu, "Deep canonical correlation analysis," in *Proc. 30th Int. Conf. Mach. Learn.*, 2013, pp. 1247–1255.
- [32] Y. Luo, D. Tao, K. Ramamohanarao, C. Xu, and Y. Wen, "Tensor canonical correlation analysis for multi-view dimension reduction," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 11, pp. 3111–3124, Nov. 2015.
- [33] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM J. Matrix Anal. Appl.*, vol. 21, no. 4, pp. 1253–1278, 2000.
- [34] O. Yair and R. Talmon, "Multimodal metric learning with local CCA," in *Proc. IEEE Statist. Signal Process. Workshop*, Jun. 2016, pp. 1–5.
- [35] C. J. Dsilva, R. Talmon, C. W. Gear, R. R. Coifman, and I. G. Kevrekidis, "Data-driven reduction for a class of multiscale fast-slow stochastic dynamical systems," *SIAM J. Appl. Dyn. Syst.*, vol. 15, no. 3, pp. 1327–1351, 2016.
- [36] D. Kushnir, A. Haddad, and R. R. Coifman, "Anisotropic diffusion on sub-manifolds with application to earth structure classification," *Appl. Comput. Harmon. Anal.*, vol. 32, no. 2, pp. 280–294, 2012.
- [37] L. M. Ewerbring and F. T. Luk, "Canonical correlations and generalized SVD: Applications and new algorithms," in *Proc. 32nd Annu. Tech. Symp.*, 1989, pp. 206–222.
- [38] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2005, pp. 1601–1608.
- [39] M. P. Heinrich, B. W. Papież, J. A. Schnabel, and H. Handels, "Multispectral image registration based on local canonical correlation analysis," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervention*, 2014, pp. 202–209.
- [40] F. R. Bach and M. I. Jordan, "Kernel independent component analysis," *J. Mach. Learn. Res.*, vol. 3, pp. 1–48, 2002.
- [41] T. Rainforth and F. Wood, "Canonical correlation forests," arXiv:1507.05444.
- [42] P. Comon, X. Luciani, and A. L. De Almeida, "Tensor decompositions, alternating least squares and other tales," *J. Chemometrics*, vol. 23, nos. 7/8, pp. 393–405, 2009.
- [43] A. L. Goldberger *et al.*, "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, Jun. 2000, PMID:1085218, doi: 10.1161/01.CIR.101.23.e215. [Online]. Available: <http://circ.ahajournals.org/cgi/content/full/101/23/e215>
- [44] B. Kemp, A. H. Zwinderman, B. Tuk, H. A. Kamphuisen, and J. J. Obery, "Analysis of a sleep-dependent neuronal feedback loop: The slow-wave microcontinuity of the EEG," *IEEE Trans. Biomed. Eng.*, vol. 47, no. 9, pp. 1185–1194, Sep. 2000.
- [45] J. Bruna and S. Mallat, "Invariant scattering convolution networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1872–1886, Aug. 2013.
- [46] H.-T. Wu, R. Talmon, and Y.-L. Lo, "Assess sleep stage by modern signal processing techniques," *IEEE Trans. Biomed. Eng.*, vol. 62, no. 4, pp. 1159–1168, Apr. 2015.



**Or Yair** (S'15) received the B.Sc. degree (*summa cum laude*), in 2012, in electrical engineering from the Technion–Israel Institute of Technology, Haifa, Israel, where he is currently working toward the Ph.D. degree. From 2012 to 2015, he worked in the field of image processing and algorithms. Since 2015, he has been a Teaching Assistance with the Viterbi Faculty of Electrical Engineering, Technion–Israel Institute of Technology. His main areas of interest include signal processing, machine learning, and geometric methods. He received the Meyer Fellowship and Zipers Award for 2015 and the Diane and Leonard Sherman Interdisciplinary Fellowship for 2016.



**Ronen Talmon** (M'11) received the B.A. degree (*cum laude*) in mathematics and computer science from the Open University, Ra'anana, Israel, in 2005, and the Ph.D. degree in electrical engineering from the Technion–Israel Institute of Technology, Haifa, Israel, in 2011.

From 2000 to 2005, he was a Software Developer and Researcher at a Technological Unit of the Israeli Defense Forces. From 2005 to 2011, he was a Teaching Assistant in the Department of Electrical Engineering, Technion. From 2011 to 2013, he was a Gibbs Assistant Professor in the Mathematics Department, Yale University, New Haven, CT. In 2014, he joined the Department of Electrical Engineering, Technion, where he is an Assistant Professor of electrical engineering. His research interests include statistical signal processing, analysis and modeling of signals, speech enhancement, biomedical signal processing, applied harmonic analysis, and diffusion geometry.

He received the Irwin and Joan Jacobs Fellowship, the Andrew and Erna Fince Viterbi Fellowship, and the Horev Fellowship.