One of the challenges in data analysis is to distinguish between different sources of variability manifested in data. In this paper, we consider the case of multiple sensors measuring the same physical phenomenon, such that the properties of the physical phenomenon are manifested as a hidden common source of variability (which we would like to extract), while each sensor has its own sensor-specific effects. We present a method based on alternating products of diffusion operators, and show that it extracts the common source of variability. Moreover, we show that this method extracts the common source of variability in a multi-sensor experiment as if it were a standard manifold learning algorithm used to analyze a simple single-sensor experiment, in which the common source of variability is the *only* source of variability.

Common Manifold Learning Using Alternating-Diffusion

Roy R. Lederman[†], Ronen Talmon[‡],
Technical Report YALEU/DCS/TR-1497
September 15, 2014
Updated March 13, 2015

[†] Applied Mathematics Program, Yale University, New Haven CT 06511.
[‡] Department of Electrical Engineering, Technion - Israel Institute of Technology, Haifa 32000, Israel.

Approved for public release: distribution is unlimited.
Keywords: *Common Variable, Alternating-Diffusion, Diffusion-Maps.*

1 Introduction

Measurement systems typically have many sources of variability. When multiple sensors are used to measure the same physical phenomenon, some sources of variability are related to the actual physical phenomenon, whereas other sources of variability are irrelevant, sensor-specific effects. In this case, extracting the common source of variability and discarding the sensor-specific sources may uncover the essence of the data, separating the relevant information from the irrelevant information. We note that sensor-related sources of variability are not restricted to noise and interferences, but also include variables with "structures", such as the position and orientation of a sensor, environmental effects, and channel characteristics.

Unsupervised Manifold Learning is a class of nonlinear data-driven methods, e.g. ISOMAP [1], locally linear embedding (LLE) [2], Hessian Maps [3], and Laplacian Eigenmaps [4], often used to extract the sources of variability in given data sets. Of particular interest in the context of this paper is Diffusion Geometry [5, 6, 7, 8, 9], a manifold learning framework, in which discrete diffusion processes are constructed on the given data points; these diffusion processes are designed to capture the structure of the sources of variability. In the case of multiple sensors, despite having more information, adding sensors adds sources of variability, making it more difficult to extract the common source of variability. Various methods have been proposed to analyze data from multiple sensors within the framework of Manifold Learning. Often, the vectors representing the data are concatenated into one vector, but in this case it is not clear how the data from each sensor should be scaled, especially if the sensors are of very different nature. It has been proposed in [10] to use Diffusion Maps to obtain a low-dimensional representation of data from each sensor, and then to concatenate the low dimensional representations. However, this method does not overcome the general problem of many sources of variability.

A different approach designed to extract the common source of variability from two sensors is Canonical Correlation Analysis (CCA) [11], which recovers highly correlated linear projections in linear systems, but has limited applicability to non-linear problems. Kernel CCA (KCCA), the generalization of CCA to the kernel feature space (e.g. [12, 13]), treats some aspects of nonlinearity, but it relies on inversion of covariance matrices, an operation that raises statistical and numerical issues in applications. Another related method [14] also assumes certain linearities in the problem.

In this paper, we propose a data-driven method based on a product of diffusion operators. In the context of supervised learning, linear combinations of kernels have been the subject of considerable work on Multi Kernel Learning (e.g. [15]). In the literature of multi-view problems, several approaches have been proposed for metric-fusion, clustering and classification, based on various manipulation of affinity matrices (e.g. [16, 17, 18, 19]), Markov and diffusion matrices (e.g. [20, 21]), graph Laplacians (e.g. [22, 23]) and sets of nearest neighbors (e.g. [24, 25]). Tensor products of Markov matrices have been proposed in [26] and products of Markov matrices and their transposes have been proposed in [27]. A recent work on products of kernels in [28] considers the fusion of different manifestations of the same variables, in the absence of sensor-specific variability. The models and goals in these studies are different, and the algorithms, theoretical justifications and proofs have only limited applicability to the common variable problem in unsupervised manifold learning.

The main contributions of this paper are the presentation of an alternating-diffusion method and showing that it solves the common variable extraction problem. We show that the common source of variability is extracted by this method from multiple sensors as if it were the only source of variability in a single sensor, extracted by a diffusion method.

In the analysis of the algorithm we distinguish between two types of objects: *observable* objects, which are quantities that can be approximated based on the measurements (following the standard practices of Manifold Learning), and *hidden* objects, which are not approximated/accessible directly. We discuss a hidden effective diffusion process and use it to develop a manifold learning method for extracting the hidden common variable. While the hidden effective diffusion is merely a formal object that is not accessible directly, we present an algorithm that is based on observables that is equivalent to computing the hidden manifold.

The structure of this paper is as follows. In the remainder of Section 1, we formulate the common variable problem and present some intuitive motivation for the algorithm. In Section 2, we briefly describe some of the notation and mathematical methods used in this paper. In Section 3, we present the alternating-diffusion method. In Section 4, the method is analyzed and the theoretical results are presented. In Section 5, we present several modifications that are useful in implementation. In Section 6, we demonstrate the performance of the method on simulated and real data. Finally, in Section 7 we summarize our conclusions.

The data sets used in this paper with code examples and additional results are available online at `http://roy.lederman.name/alternating-diffusion/`
.

1.1 Problem formulation

Consider three hidden random variables $(X, Y, Z) \sim \pi_{x,y,z}(X, Y, Z)$, from the (possibly high dimensional) spaces $\mathcal{X}$, $\mathcal{Y}$ and $\mathcal{Z}$, where, given X, the variables Y and Z are independent. In other words, the joint probability density of the hidden variables can be factorized as

$$\pi_{x,y,z}(X, Y, Z) = \pi_x(X)\pi_{y|x}(Y|X)\pi_{z|x}(Z|X), \tag{1}$$

where π_x is the marginal density of X, $\pi_{y|x}$ is the conditional density of Y given X, and $\pi_{z|x}$ is the conditional density of Z given X.

We have access to these hidden variables through two observable random variables

$$S^{(1)} = g(X, Y) \tag{2}$$

and

$$S^{(2)} = h(X, Z) \tag{3}$$

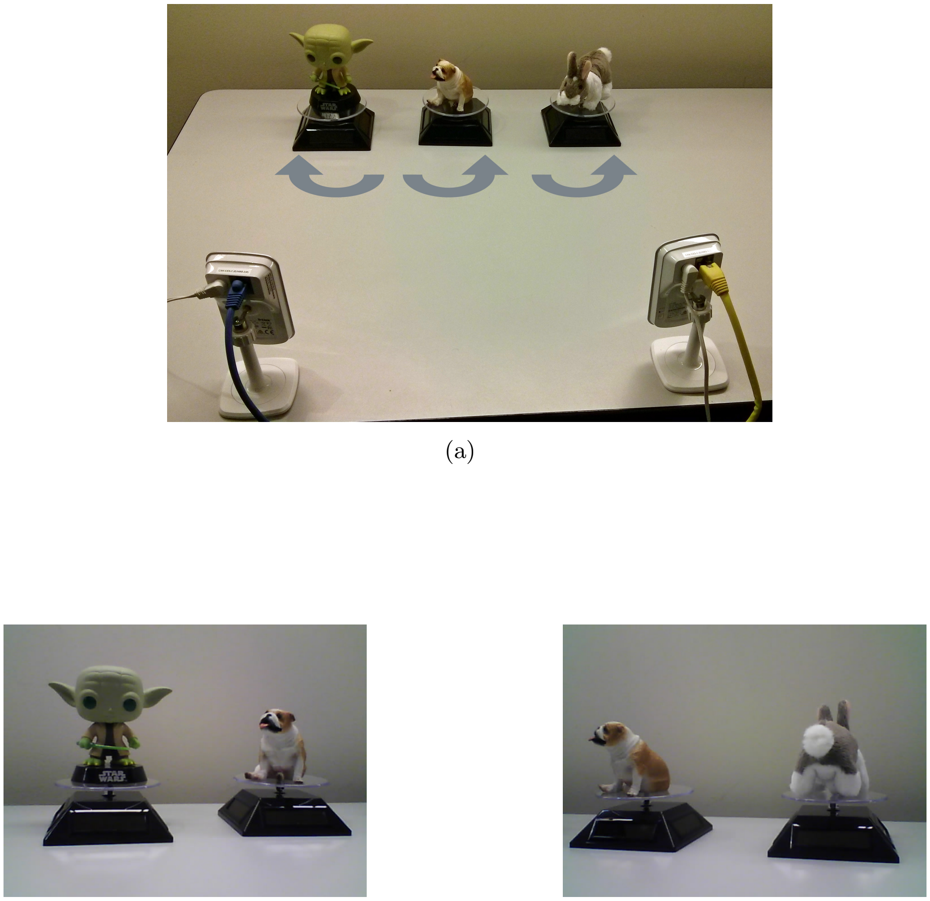
from the (possibly high dimensional) spaces $\mathcal{S}^{(1)}$ and $\mathcal{S}^{(2)}$, where g and h are bilipschitz functions.

One realization of the system consists of the hidden triplet (x_i, y_i, z_i) and the corresponding measurements $(s_i^{(1)}, s_i^{(2)})$; while x_i, y_i and z_i are hidden and not available to us directly, $s_i^{(1)} = g(x_i, y_i)$ and $s_i^{(2)} = h(x_i, z_i)$ are observable. We refer to $s_i^{(1)}$ and $s_i^{(2)}$ as the measurement in Sensor 1 and the measurement in Sensor 2, respectively. We note that both $s_i^{(1)}$ and $s_i^{(2)}$ are functions of the hidden common x_i, whereas y_i and z_i are the sensor-specific components.

From n realizations of the system $\{(x_i, y_i, z_i)\}_{i=1}^n$, we have n pairs of corresponding measurements $\left\{(s_i^{(1)}, s_i^{(2)})\right\}_{i=1}^n$. Our goal is to recover a parametrization of the samples of the common variable $\{x_i\}_{i=1}^n$.

1.2 Illustrative toy problem

To illustrate this setup, we consider the following toy problem. We placed three objects: Yoda (the green action figure), a bulldog, and a bunny, on three rotating displays, as depicted in Fig. 1(a). Two cameras were used to take simultaneous snapshots of the rotating objects: the field of view of Camera 1 included Yoda and the bulldog, as presented in Fig. 1(b), and the



(a)

(b)

(c)

Figure 1: (a) The experiment setup of the toy problem. (b) Sample snapshot taken by Camera 1, where only Yoda (the green action figure) and the bulldog are visible. (c) Sample snapshot taken by Camera 2, where only the bunny and the bulldog are visible.

field of view of Camera 2 included the bulldog and the bunny, as presented in Fig. 1(c).

In this problem, the rotation angles of the bulldog, Yoda and the bunny are the hidden variables X, Y and Z, respectively, and the snapshots from Camera 1 and Camera 2 are the measurements $S^{(1)} = g(X, Y)$ and $S^{(2)} = h(X, Z)$, respectively. The goal is to find a parametrization of the common variable X, i.e., the rotation angle of the bulldog, from the snapshots.

The proposed algorithm is data-driven and it does not rely on prior knowledge of the problem setup; in particular, the algorithm does not assume that the common variable X is the rotation angle of an object. The setup is given here for the purpose of illustration.

To demonstrate the underlying concept, we describe a caricature of the alternating-diffusion based on the toy problem. We start with an arbitrary pair of simultaneous snapshots, for example, the pair presented at the top row in Fig. 2. On the left is the snapshot taken by Camera 1, and on the right is the corresponding snapshot taken by Camera 2.

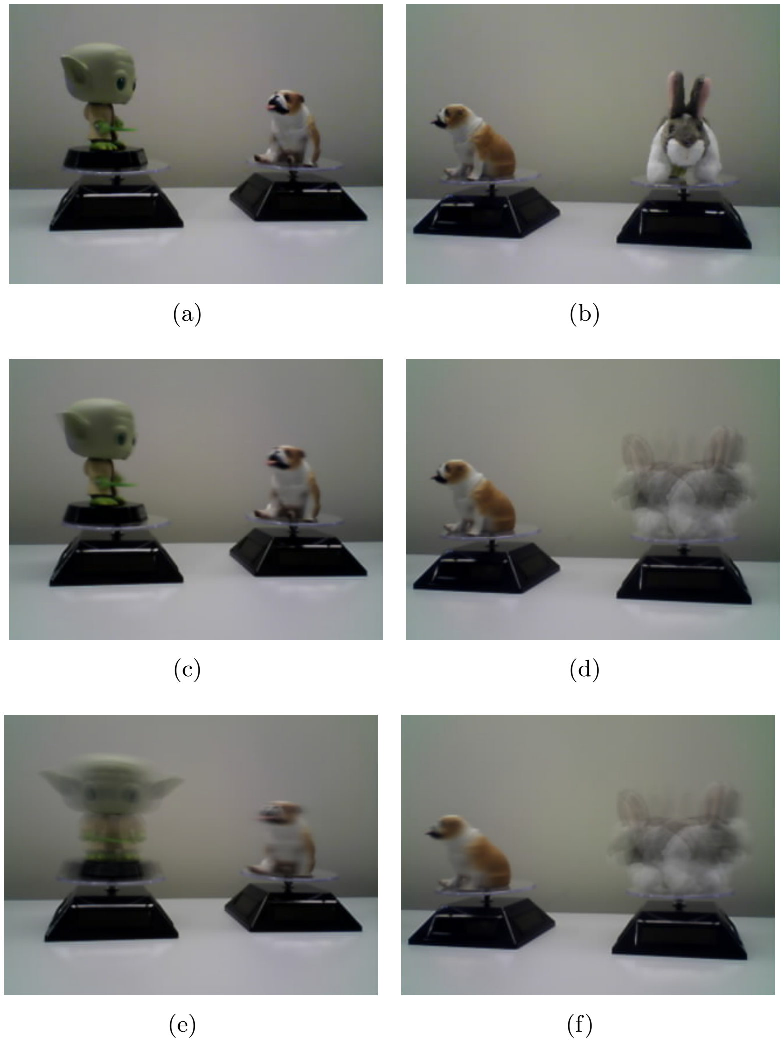
We find the nearest neighbors of this initial pair in the entire sequence. For this nearest neighbors search, we consider only the snapshots taken by Camera 1, ignoring the snapshots taken by Camera 2, i.e., in this step, two pairs are similar if and only if their respective snapshots from Camera 1 are similar.

We compute the average of these neighbors and present it in the middle row of Fig. 2: the average image of the snapshots taken by Camera 1 is presented on the left, and the average image of the corresponding snapshots taken by Camera 2 at the same times is presented on the right.

The average image on the left is sharp and very similar to the initial image taken by Camera 1, implying that both the bulldog and Yoda are at similar rotation angles in the nearest neighbors snapshots. However, in the average image on the right, the bunny is blurred, implying that it was in different rotation angles in the nearest neighbors snapshots. This result stems from the fact that the bunny was visible only to Camera 2, and it was completely ignored when we computed the nearest neighbors based on snapshots taken by Camera 1.

Now, we take each one of the nearest neighbors that we found in the previous step, and find its nearest neighbors, however, this time we search for the new nearest neighbors based on similarity in the snapshots taken by Camera 2. We refer to the nearest neighbors in Camera 2 of all the nearest neighbors in Camera 1 as the *indirect nearest neighbors*.

We aggregate all the indirect nearest neighbors, and compute their average images, presented in the bottom row of Fig. 2. In the two new average



(a) (b)

(c) (d)

(e) (f)

Figure 2: Demonstration of the alternating-diffusion. The images on the left column are snapshots taken by Camera 1 (or average snapshots taken by Camera 1) and the images on the right column are snapshots taken by Camera 2 (or average snapshots taken by Camera 2). Two actual snapshots taken by Camera 1 are presented in the top row. The averages of neighbors are presented in the middle row. The averages of indirect neighbors are presented in the bottom row.

images, only the bulldog is sharp, while Yoda and the bunny are blurred. This implies that the bulldog was in very similar rotation angles in all the indirect nearest neighbors, whereas Yoda and the bunny were oriented in “random” directions. In other words, the rotation angle of the bulldog is coherent across the indirect nearest neighbors, while the angles of the camera-specific objects, Yoda and the bunny, are incoherent.

In summary, the coherence in the rotation angle of the bunny is suppressed when we process the nearest neighbors in Camera 1, and the coherence in the rotation angle of Yoda is suppressed when we process the indirect nearest neighbors in Camera 2. However, the coherence in the rotation angle of the bulldog is largely preserved.

In this paper we will use these coherence and incoherence properties across indirect nearest neighbors to recover the structure of the coherent common variable, while discarding the incoherent sensor-specific variables.

We note that the averaging of images in this example is performed only for the purpose of illustration. While the averaging of images is convenient for visualization in the case of consistent specially separated objects, it is inadequate in other models. The algorithm does not assume special separation and does not compute average samples, but rather essentially compares the sets of indirect nearest neighbors of different points in order to measure the similarity between points.

2 Preliminaries

2.1 Notation

We find it convenient to use the expected value notation for integration; the underlying probability density $\pi_{x,y,z}$ generating the data is implied by the expected value notation, whereas operators, kernels and functions are presented explicitly.

Unless specifically stated otherwise, the underlying probability density throughout this paper is assumed to be the probability density $\pi_{x,y,z}$ specified in the problem formulation in section 1.1.

We denote by $\mathbb{E}_{x,y,z}$ the expected value with respect to the probability density $\pi_{x,y,z}$; the expected value of the function $f(x,y,z)$ is defined by

$$\mathbb{E}_{x,y,z}\left(f(x,y,z)\right) \quad = \quad \int_{\mathcal{X},\mathcal{Y},\mathcal{Z}} f(x,y,z)\pi_{x,y,z}(x,y,z)dxdydz. \tag{4}$$

The expected value with respect to a marginal probability density, such as π_x of X, will be denoted by $\mathbb{E}_x$; the expected value $\mathbb{E}_x$ is defined by

$$\mathbb{E}_x\left(f(x)\right) = \int_{\mathcal{X}} f(x)\pi_x(x)dx \tag{5}$$

where

$$\pi_x(x) = \int_{\mathcal{Y},\mathcal{Z}} \pi_{x,y,z}(x,y,z)dzdy. \tag{6}$$

The expected value with respect to a conditional probability density, such as $\pi_{y|x}$ of Y given X, will be denoted by $\mathbb{E}_{y|x}$; for a given value of $x \in \mathcal{X}$, the expected value $\mathbb{E}_{y|x}$ is defined by

$$\mathbb{E}_{y|x}\left(f(x,y)\right) = \int_{\mathcal{Y}} f(x,y)\pi_{y|x}(x,y)dy \tag{7}$$

where

$$\pi_{y|x}(y|x) = \frac{\int_{\mathcal{Z}} \pi_{x,y,z}(x,y,z)dz}{\int_{\mathcal{Y},\mathcal{Z}} \pi_{x,y,z}(x,y,z)dzdy}. \tag{8}$$

2.2 Diffusion geometry

This section provides a short overview of a version of diffusion geometry that is convenient in the context used in this paper. For more detailed descriptions and additional variations and generalizations see, for example, [5, 7, 8]. We will discuss both the asymptotic continuous form of diffusion geometry and its discrete counterpart on sample sets.

Suppose $\mathcal{A}$ is a metric space and π_a is a probability density defined on $\mathcal{A}$. The operators in diffusion geometry do not have direct access to the variables in $A \in \mathcal{A}$, but rather through $S = \rho(A)$, where $\mathcal{S}$ is a metric space and $\rho : \mathcal{A} \to \mathcal{S}$ is a mapping from $\mathcal{A}$ to $\mathcal{S}$.

The primary component in diffusion geometry is a diffusion operator; for any function $f : \mathcal{A} \to \mathbb{R}$, the diffusion operator is defined by

$$\left(D(f)\right)(a) = \mathbb{E}_{a'}\left(K(a,a')f(a')\right),\ \forall a \in \mathcal{A} \tag{9}$$

where the kernel $K(a,a')$ is a "local" kernel; for simplicity, we assume a "local" Markov kernel in this paper, so that

$$\mathbb{E}_a\left(\left(D(f)\right)(a)\right) = \mathbb{E}_a\left(f(a)\right). \tag{10}$$

The "local" kernel is defined based on the observable variable, often using a weighted Gaussian kernel, e.g.

$$K(a,a') = \frac{1}{\omega(a')} e^{-\|\rho(a)-\rho(a')\|^2_{\mathcal{S}}/\epsilon}, \tag{11}$$

where $\|\rho(a) - \rho(a')\|_{\mathcal{S}}$ is the distance in the space of observations $\mathcal{S}$ and

$$\omega(a') = \mathbb{E}_{a''} \left(e^{-\|\rho(a')-\rho(a'')\|^2_{\mathcal{S}}/\epsilon} \right). \tag{12}$$

Intuitively, the diffusion operator "smoothens" functions; suppose that p_0 is the Dirac delta function at $\alpha \in \mathcal{A}$, then the function $p_1 = D(p_0)$ is a "bump" around α and $p_2 = D(p_1) = D^2(p_0)$ is a wider "bump." This way, a sequence of increasingly wider "bumps" $\{p_t\}_{t=0}^{\infty}$ is defined by successive "smoothening" operations:

$$p_t(a) = \left(D^t(p_0)\right)(a), \;\; a \in \mathcal{A}. \tag{13}$$

We refer to this sequence of functions as the *propagation* of p_0.

The other component used in diffusion geometry is a norm. The typical choice in the literature is a weighted L^2 norm. In this paper, we consider a more general form; we define the seminorm of a function $f : \mathcal{A} \to \mathbb{R}$ using a quadratic form:

$$\|f\|_M = \sqrt{\mathbb{E}_a \mathbb{E}_{a'} \left(f(a) M(a,a') f(a')\right)}, \tag{14}$$

where $M(a,a')$ is a positive semidefinite kernel.

The diffusion operator and the seminorm are used to define the *diffusion distance* $d_t(\alpha, \alpha')$ at the propagation time t between any two points $\alpha, \alpha' \in \mathcal{A}$;

$$d_t(\alpha, \alpha') = \|p_t - q_t\|, \tag{15}$$

where the initial functions p_0 and q_0 of the sequences are delta functions at α and α' ,respectively.

In the discrete setting, we consider n samples $\{a_i\}_{i=1}^n$ of the random variable $A \in \mathcal{A}$, sampled from π_a. These samples are not accessible directly, but rather via $s_i = \rho(a_i)$.

The discrete counterpart of the diffusion operator D is an $n \times n$ matrix $\mathbf{K}$ typically of the form

$$\mathbf{K}(i,j) = \frac{\mathbf{W}(i,j)}{w(j)}, \tag{16}$$

where

$$\mathbf{W}(i,j) = e^{-\|s_i - s_j\|^2_{\mathcal{S}}/\epsilon}, \tag{17}$$

and

$$w(j) = \sum_{l=1}^{n} e^{-\|s_j - s_l\|^2_{\mathcal{S}}/\epsilon}. \tag{18}$$

In these expressions, $\|s_i - s_j\|_{\mathcal{S}}$ is the distance between s_i and s_j in the metric space $\mathcal{S}$.

The discrete diffusion can be interpreted as a Markov chain on a graph $G = (V, E)$. Each of the n samples is represented by one of the vertices in $V = \{1, \ldots, n\}$, i.e., the i-th vertex represents the i-th realization of the system a_i and the i-th measurement $s_i = \rho(a_i)$. The weight of the edge $e_{ji} \in E$ between the vertices i and j is $\mathbf{W}(i,j)$. Then, $\mathbf{K}$ can be viewed as the transition probability matrix on this graph: the probability of transition to vertex i from vertex j in a single step is $\mathbf{K}(i,j)$.

Note that the weights of the edges and the transition probabilities are computed based on the observable measured samples, whereas the graph vertices represent realizations of the entire system, so that the i-th sample is associated with the observable s_i and the hidden a_i.

Let $v_0 = (0, 0, \ldots, 0, 1, 0, \ldots)^T$ be the vector of dimensionality n, of all zeros, except for the i-th position. The propagation from the i-th point is defined as a sequence of vectors $\{v_t\}_{t=0}^{\infty}$ such that

$$v_{t+1} = \mathbf{K} v_t. \tag{19}$$

The vectors v_0 and v_t are the discrete counterparts of the functions p_0 and p_t; the j-th element of v_t is viewed as the sample of the function $p_t(a_j)$ at the point a_j.

Intuitively, the transition matrix $\mathbf{K}$ can be viewed as allowing a transition from the vertex j to the vertex i only when $\|s_i - s_j\|_{\mathcal{S}} \leq \sqrt{\epsilon}$. For a bilipschitz function ρ, it implies that the transition is allowed when $\|a_i - a_j\|$ is small. As a result, after the first step of the discrete diffusion initialized at the j-th vertex, the vector v_1 has non negligible values in elements i, such that a_i are a small neighborhood around a_j. Following a similar argument, each additional step extends this neighborhood of elements with non-negligible values to a larger neighborhood around a_j.

The discrete diffusion distance between sample i and sample j is defined using a Euclidean distance (or a weighted Euclidean distance, see [7])

$$d_t(i,j) = \|v_t - u_t\|_2, \tag{20}$$

where v_t and u_t are vectors in the propagation sequences (defined in (19)) from the sample i and the sample j, respectively.

The diffusion distance has been shown to be a powerful metric of comparing samples that capture the structure of the graph, e.g., it is invariant to small topological distortions and moderate noise [7, 29]. While the Euclidean distance $\|s_i - s_j\|_{\mathcal{S}}$ compares two individual samples, the diffusion distance integrates other samples and measures the "connectivity" between the two samples via the entire sample set.

3 Algorithm

The alternating-diffusion method discussed in this paper is detailed in Algorithm 1.

3.1 Description of the algorithm

In the common variable problem discussed in this paper, we have two sample sets, $\{s_i^{(1)}\}_{i=1}^n$ and $\{s_i^{(2)}\}_{i=1}^n$ from Sensor 1 and Sensor 2, respectively. For each sample set, we construct the affinity matrices $\mathbf{W}^{(\mathbf{s}_1)}$ and $\mathbf{W}^{(\mathbf{s}_2)}$, specified in (21), and the associated diffusion operators $\mathbf{K}^{(\mathbf{s}_1)}$ and $\mathbf{K}^{(\mathbf{s}_2)}$, specified in (22). By construction, $\mathbf{K}^{(\mathbf{s}_1)}$ and $\mathbf{K}^{(\mathbf{s}_2)}$ are Markov matrices.

The propagation from the i-th sample is defined as a sequence of vectors $\{v_t\}_{t=0}^{\infty}$ such that

$$v_{t+1} = \begin{cases} \mathbf{K}^{(\mathbf{s}_1)} v_t, & t = 2m \\ \mathbf{K}^{(\mathbf{s}_2)} v_t, & t = 2m+1 \end{cases} \tag{25}$$

for every integer $m \geq 0$, where the initial vector $v_0 = (0, 0, \ldots, 0, 1, 0, \ldots)^T$ is a vector of dimensionality n, of all zeros, except for the i-th position. It follows that the even steps of the propagation defined in (25) can be restated as

$$v_{2m} = \mathbf{K}^m v_0. \tag{26}$$

where $\mathbf{K}$ is the alternating-diffusion Markov matrix defined in (23).

We define the diffusion distance between sample i and sample j based on the alternating-diffusion as the following Euclidean distance

$$d_t(i,j) = \|v_t - u_t\|_2, \tag{27}$$

where v_t and u_t are vectors in the propagation sequences (defined in (25)) from the sample i and the sample j, respectively.

Algorithm 1 Alternating-diffusion

Input: *aligned* samples from the two sensors $\left\{(s_i^{(1)}, s_i^{(2)})\right\}_{i=1}^{n}$.

Output: diffusion distances $d_{2m}(i,j)$. Optionally, refined diffusion distances and diffusion maps embedding.

1. Calculate two pairwise affinity matrices (kernels) $\mathbf{W}^{(\mathbf{s}_1)}$ and $\mathbf{W}^{(\mathbf{s}_2)}$ based on a Gaussian as follows:

   $$W_{ij}^{(\mathbf{s}_1)} = \exp\left(-\frac{\|s_i^{(1)} - s_j^{(1)}\|^2}{\varepsilon^{(1)}}\right);\ W_{ij}^{(\mathbf{s}_2)} = \exp\left(-\frac{\|s_i^{(2)} - s_j^{(2)}\|^2}{\varepsilon^{(2)}}\right) \tag{21}$$

   for all $i, j = 1, \ldots, n$, where $\varepsilon^{(1)}$ and $\varepsilon^{(2)}$ are the kernel scales.

2. Create two diffusion operators $\mathbf{K}^{(\mathbf{s}_1)}$ and $\mathbf{K}^{(\mathbf{s}_2)}$:

   $$K_{ij}^{(\mathbf{s}_1)} = \frac{W_{ij}^{(\mathbf{s}_1)}}{\sum\limits_{l=1}^{n} W_{lj}^{(\mathbf{s}_1)}}; K_{ij}^{(\mathbf{s}_2)} = \frac{W_{ij}^{(\mathbf{s}_2)}}{\sum\limits_{l=1}^{n} W_{lj}^{(\mathbf{s}_2)}} \tag{22}$$

3. Build an alternating-diffusion kernel:

   $$\mathbf{K} = \mathbf{K}^{(\mathbf{s}_2)}\mathbf{K}^{(\mathbf{s}_1)}. \tag{23}$$

4. Compute the diffusion distance at time $2m$ between each two points i, j:

   $$d_{2m}(i,j) = \sum_{l=1}^{n} \left((\mathbf{K}^m)_{l,i} - (\mathbf{K}^m)_{l,j}\right)^2. \tag{24}$$

5. (Optionally:) Compute refined diffusion distances and refined diffusion maps using a standard diffusion maps algorithm (see A), by substituting the diffusion distance $d_{2m}(i,j)$ computed in the previous step into the distance between measurements in the input of the diffusion maps algorithm.

The diffusion distance defined in (27) is computed according to (24) using the columns of the matrix $\mathbf{K}^m$. The two expressions (27) and (24) are equivalent since the i-th column of $\mathbf{K}^m$ is equal to the vector v_t.

We will show in Section 4 that the new alternating-diffusion operator $\mathbf{K}$ is related to an "effective" diffusion operator, and that it captures the structure of the common variable and ignores the variables specific to either sensor. It follows that the diffusion distances (24) inherit the properties of diffusion distances computed from data where the common variable is the only source of variability.

Optionally, the results can be refined using a standard diffusion maps algorithm on the diffusion distances $d_{2m}(i,j)$ computed in the previous step as the distances between measurements, thereby obtaining a low-dimensional embedding of the data.

3.2 Intuitive interpretation

Applying standard diffusion to the set of measurements $\{s_i^{(1)}\}$ builds a Markov chain on a graph $G^{(1)} = (V, E^{(1)})$. Each of the n samples is represented by one of the vertices in $V = \{1, \ldots, n\}$, i.e., the i-th vertex represents the i-th realization of the system. The weight of the edge $e_{ij}^{(1)} \in E^{(1)}$ between the vertices i and j is $\mathbf{W}^{(\mathbf{s}_1)}(i,j)$. Therefore, the diffusion matrix $\mathbf{K}^{(\mathbf{s}_1)}$ can be viewed as the transition probability matrix on this graph: the probability of transition to vertex i from vertex j in a single step is $\mathbf{K}^{(\mathbf{s}_1)}(i,j)$. Similarly, a separate application to the set $\{s_i^{(2)}\}$ builds a Markov chain with a different transition probability matrix $\mathbf{K}^{(\mathbf{s}_2)}$ on a graph $G^{(2)} = (V, E^{(2)})$. In other words, we obtain two graphs with the same set of vertices V, representing the samples, and two different sets of weighted edges $E^{(1)}$ and $E^{(2)}$, determined by the distances between samples within each separate set of measurements.

The alternating-diffusion operates on the same set of vertices V. However, the transition probabilities are defined as follows: the transition probability matrix in odd steps is $\mathbf{K}^{(\mathbf{s}_1)}$, and the transition probability matrix in even steps is $\mathbf{K}^{(\mathbf{s}_2)}$. Therefore, the combination of two consecutive steps is a Markov chain on a new (directed) graph $G = (V, E)$, where the transition probabilities are determined by the matrix K.

To illustrate the diffusion in more detail, we revisit the example in Fig. 2 and describe it more detail. We consider a sequence of vectors $\{v_t\}_{v=0}^{\infty}$ (defined in (25)) of alternating-diffusion propagation from sample i; this state is represented by the initial vector v_0 in the sequence, which is nonzero only at the i-th coordinate. At this initial state, the bulldog, Yoda, and the bunny are oriented in angles x_i, y_i and z_i, respectively; the snapshots $s_i^{(1)}$ and $s_i^{(2)}$ are depicted in the top row of Fig. 2.

In the middle row of Fig. 2, we present the weighted average $\sum_j v_1(j)s_j^{(1)}$ and $\sum_j v_1(j)s_j^{(2)}$ of the snapshots, based on the propagated vector v_1. On the left, we observe that the average image $\sum_j v_1(j)s_j^{(1)}$ is almost identical to the initial snapshot $s_i^{(1)}$. It implies that the $v_1(j)$ is non-negligible only if both the rotation angles of the bulldog x_j and of Yoda y_j are close to the initial rotation angles x_i and y_i respectively.

On the right, we observe that the bunny in the average image $\sum_j v_1(j)s_j^{(2)}$ is blurred; this implies that $v_1(j)$ is non-negligible at some of the coordinates that correspond to samples where the rotation angle of the bunny z_j is significantly different from the initial rotation angle z_i.

The middle image demonstrates that the propagated vector v_1 (after the first step of the alternating-diffusion) is non-negligible only in coordinates that correspond to small neighborhoods of X and Y around the initial values x_i and y_i, but to a large range of values of the variable Z, ignoring the initial value z_i.

In the bottom row of Fig. 2 we present the weighted average of the snapshots $\sum_j v_2(j)s_j^{(1)}$ and $\sum_j v_2(j)s_j^{(2)}$ based on the propagated vector v_2. We observe that both Yoda and the bunny are now blurred, whereas the bulldog remains sharp in both the right and the left images. This implies that the second step of the alternating-diffusion leads to a small neighborhood of X around the initial x_i, but to a large range of the variables Y and Z, ignoring the initial values y_i and z_i.

In a similar manner, the following alternating-diffusion steps propagate the values of the common variable X to gradually growing neighborhoods around the initial x_i. At the same time, the variables Y and Z are distributed across large ranges, as evident from the first two alternating-diffusion steps. We will show that this alternating-diffusion process can be viewed as a diffusion process on the common variable X, ignoring the values of Y and Z. In other words, the alternating-diffusion extracts the common variable X from the given sample sets.

3.3 More than two sensors

The method applies to the problem of finding the common source of variability in measurement from more than two sensors. One approach to integrating more than two sensors is to replace (23) with

$$\mathbf{K} = \mathbf{K}^{(\mathbf{s}_m)}...\mathbf{K}^{(\mathbf{s}_2)}\mathbf{K}^{(\mathbf{s}_1)}, \tag{28}$$

and optionally average over all the possible permutations in the order of the operators. The extension of the algorithm and its analysis, which will be presented in Section 4, is straightforward.

4 Analysis

4.1 Observable diffusion on Sensor 1 and Sensor 2

We denote the observable diffusion operator of Sensor 1 by $D^{(\mathbf{s}_1)}$. For a function $p : \mathcal{S}^{(1)} \to \mathbb{R}$, the operation of $D^{(\mathbf{s}_1)}$ is defined by

$$\left(D^{(\mathbf{s}_1)}(p)\right)(s) = \mathbb{E}_{s'}\left(\frac{e^{-\|s-s'\|^2/\epsilon}}{\mathbb{E}_{s''}\left(e^{-\|s''-s'\|^2/\epsilon}\right)}\right). \tag{29}$$

This operator is approximated by $\mathbf{K}^{(\mathbf{s}_1)}$ defined in (22). Since by (2) each point in $\mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$ is mapped to $\mathcal{S}^{(1)}$ by the function g, the following definition is equivalent to (29):

$$\left(D^{(\mathbf{s}_1)}(p)\right)(x, y, z) = \mathbb{E}_{x',y',z'}\left(K^{(\mathbf{s}_1)}((x, y), (x', y'))p(x', y', z')\right), \tag{30}$$

where $K^{(\mathbf{s}_1)}((x, y), (x', y'))$ is defined by

$$K^{(\mathbf{s}_1)}((x, y), (x', y')) = \frac{1}{\omega^{(\mathbf{s}_1)}(x', y')}e^{-\|g(x,y)-g(x',y')\|^2/\epsilon}, \tag{31}$$

and

$$\omega^{(\mathbf{s}_1)}(x', y') = \mathbb{E}_{x'',y'',z''}\left(e^{-\|g(x'',y'')-g(x',y')\|^2/\epsilon}\right). \tag{32}$$

We note that the variable z appears on the left side of (30), but does not appear on the right side of the equation, because the measurement from Sensor 1 is independent of Z given X; the significance is discussed in Lemma 1 below.

Similarly, we denote the observable diffusion operator of Sensor 2 by $D^{(\mathbf{s}_2)}$, and its operation is defined by

$$\left(D^{(\mathbf{s}_2)}(p)\right)(x, y, z) = \mathbb{E}_{x',y',z'}\left(K^{(\mathbf{s}_2)}((x, z), (x', z'))p(x', y', z')\right) \tag{33}$$

where $K^{(\mathbf{s}_2)}$ is

$$K^{(\mathbf{s}_2)}((x,z),(x',z')) = \frac{1}{\omega^{(\mathbf{s}_2)}(x',z')} e^{-\|h(x,z)-h(x',z')\|^2/\epsilon}, \tag{34}$$

and

$$\omega^{(\mathbf{s}_2)}(x',z') = \mathbb{E}_{x'',y'',z''}\left(e^{-\|h(x'',z'')-h(x',z')\|^2/\epsilon}\right). \tag{35}$$

We note that the variable y appears on the left side of (33), but does not appear on the right side of the equation, because the measurement from Sensor 2 is independent of Y given X; the significance is discussed in Lemma 2 below.

For a function $p_0 : \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \to \mathbb{R}$, we define the sequence of observable *propagated functions* $\{p_t\}_{t=0}^{\infty}$ by

$$p_{t+1}(x,y,z) = \begin{cases} \left(D^{(\mathbf{s}_1)}(p_t)\right)(x,y,z), & \forall t = 2m \\ \left(D^{(\mathbf{s}_2)}(p_t)\right)(x,y,z), & \forall t = 2m+1 \end{cases} \tag{36}$$

where $m \geq 0$ is a nonnegative integer.

We note that the subsequence of the odd steps takes the form of standard diffusion:

$$p_{2m+3}(x,y,z) = \left(D(p_{2m+1})\right)(x,y,z), \tag{37}$$

where D is the product of the standard diffusion operators $D^{(\mathbf{s}_1)}$ and $D^{(\mathbf{s}_2)}$

$$D \quad = \quad D^{(\mathbf{s}_1)} \circ D^{(\mathbf{s}_2)}. \tag{38}$$

Let $\|\cdot\|_\pi$ denote the observable norm of functions on $\mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$; for a function $p : \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \to \mathbb{R}$, the norm $\|\cdot\|_\pi$ is defined by

$$\|p\|_\pi = \left(\mathbb{E}_{x,y,z}\left(p^2(x,y,z)\right)\right)^{1/2} \tag{39}$$

For $t > 0$, we define the observable diffusion distance $d_t^{(\pi)}$ between (x_i, y_i, z_i) and (x_j, y_j, z_j) by

$$d_t^{(\pi)}\left((x_i,y_i,z_i),(x_j,y_j,z_j)\right) = \|p_t - q_t\|_\pi, \tag{40}$$

where p_0 is a delta function at (x_i, y_i, z_i) and q_0 is a delta function at (x_j, y_j, z_j), and $\{p_t\}_{t=0}^{\infty}$ and $\{p_t\}_{t=0}^{\infty}$ are the sequences of propagated functions from p_0 and q_0, respectively. $d_t^{(\pi)}$ is approximated by d_t, defined in (24)

$$d_t^{(\pi)}\left((x_i,y_i,z_i),(x_j,y_j,z_j)\right) \approx C \cdot d_t(i,j). \tag{41}$$

4.2 Properties of the diffusion operators $D^{(\mathbf{s}_1)}$ and $D^{(\mathbf{s}_2)}$ and propagated functions

In this subsection we present some of the properties of the diffusion operators $D^{(\mathbf{s}_1)}$ and $D^{(\mathbf{s}_2)}$ that will be used to show the main results. We show these properties in the context of the sequence of functions $p_t(x,y,z),\ t=1,2,\dots$ propagated from $p_0(x,y,z)$ by (36).

As an intermediate step towards the definition of the effective functions on X, we define the hidden *intermediate functions* $p_t^{(\mathbf{i}_1)}(x,y)$ on $\mathcal{X}\times\mathcal{Y}$ and $p_t^{(\mathbf{i}_2)}(x,z)$ on $\mathcal{X}\times\mathcal{Z}$ by

$$p_t^{(\mathbf{i}_1)}(x,y) = \mathbb{E}_{z|x,y}\left(p_t(x,y,z)\right), \tag{42}$$

and

$$p_t^{(\mathbf{i}_2)}(x,z) = \mathbb{E}_{y|x,z}\left(p_t(x,y,z)\right). \tag{43}$$

Given X, the random variables Y and Z are independent, therefore,

$$p_t^{(\mathbf{i}_1)}(x,y) = \mathbb{E}_{z|x}\left(p_t(x,y,z)\right), \tag{44}$$

and

$$p_t^{(\mathbf{i}_2)}(x,z) = \mathbb{E}_{y|x}\left(p_t(x,y,z)\right). \tag{45}$$

In the remainder of this subsection, $m\geq 0$ denotes a nonnegative integer.

Lemma 1 (Properties of the image of $D^{(\mathbf{s}_1)}$)**.** *For every odd step $t=2m+1$, and for all $z\in\mathcal{Z}$, the propagated function $p_{2m+1}(x,y,z)=\left(D^{(\mathbf{s}_1)}\left(p_{2m}\right)\right)(x,y,z)$ satisfies*

$$p_{2m+1}(x,y,z) = p_{2m+1}^{(\mathbf{i}_1)}(x,y) \tag{46}$$

where $p_{2m+1}^{(\mathbf{i}_1)}(x,y)$ is the intermediate function of p_{2m+1} on $\mathcal{X}\times\mathcal{Y}$, specified in (44).

Proof. By (36) and (30),

$$p_{2m+1}(x,y,z) = \mathbb{E}_{x',y',z'}\left(K^{(\mathbf{s}_1)}((x,y),(x',y'))p_{2m}(x',y',z')\right). \tag{47}$$

Since z does not appear in the right side of (47), $p_{2m+1}(x,y,z)$ does not depend on the value of z. Therefore, by (42), we have

$$p_{2m+1}(x,y,z) = \mathbb{E}_{z'|x,y}\left(p_{2m+1}(x,y,z')\right) = p_{2m+1}^{(\mathbf{i}_1)}(x,y). \tag{48}$$

□

Intuitively, Lemma 1 states that the operator $D^{(\mathbf{s}_1)}$ uses only the information available at Sensor 1, i.e. $g(X, Y)$, and it has no information on the variable Z. Therefore, it assigns equal values to the propagated function p_{2m+1} at any two points (x, y, z) and (x, y, ζ), regardless of the values of z and ζ.

The following lemma repeats the statements in Lemma 1 for the even steps. The proof is analogous to the proof of Lemma 1 and therefore it is omitted for the sake of brevity.

Lemma 2 (Properties of the image of $D^{(\mathbf{s}_2)}$)**.** *For every even step $t = 2m+2$, and for all $y \in \mathcal{Y}$, the propagated function $p_{2m+2}(x, y, z) = \left(D^{(\mathbf{s}_2)}\left(p_{2m+1}\right)\right)(x, y, z)$ satisfies*

$$p_{2m+2}(x, y, z) = p^{(\mathbf{i}_2)}_{2m+2}(x, z) \tag{49}$$

where $p^{(\mathbf{i}_2)}_{2m+2}(x, z)$ is the intermediate function of p_{2m+2} on $\mathcal{X} \times \mathcal{Z}$, specified in (45).

4.3 The hidden effective propagated functions $p^{(\mathbf{e})}_j(x)$

We introduce the hidden *effective function* $p^{(\mathbf{e})}_t(x) : \mathcal{X} \to \mathbb{R}$, which is defined by

$$p^{(\mathbf{e})}_t(x) = \mathbb{E}_{y,z|x}\left(p_t(x, y, z)\right), \tag{50}$$

where $p_t(x, y, z)$ is a function in a sequence of propagated functions $\{p_t(x, y, z)\}_{t=1}^{\infty}$, defined in (36).

We remark that by (42), (43) and (50), the intermediate functions $p^{(\mathbf{i}_1)}_t(x, y)$ and $p^{(\mathbf{i}_2)}_t(x, z)$ defined in (42) and (43) are related to the effective functions via:

$$p^{(\mathbf{e})}_t(x) = \mathbb{E}_{y|x}\left(p^{(\mathbf{i}_1)}_t(x, y)\right) \tag{51}$$

$$= \mathbb{E}_{z|x}\left(p^{(\mathbf{i}_2)}_t(x, z)\right) \tag{52}$$

We introduce two hidden *effective diffusion operators*, $D^{(\mathbf{e}_1)}$ and $D^{(\mathbf{e}_2)}$, defined by

$$\left(D^{(\mathbf{e}_1)}(p^{(\mathbf{e})})\right)(x) = \mathbb{E}_{x'}\left(K^{(\mathbf{e}_1)}\left(x, x'\right) p^{(\mathbf{e})}(x')\right), \tag{53}$$

and

$$\left(D^{(\mathbf{e}_2)}(p^{(\mathbf{e})})\right)(x) = \mathbb{E}_{x'}\left(K^{(\mathbf{e}_2)}\left(x, x'\right) p^{(\mathbf{e})}(x')\right), \tag{54}$$

with the following *effective kernels*

$$K^{(\mathbf{e}_1)}(x, x') \quad = \quad \mathbb{E}_{y|x}\left(\mathbb{E}_{y'|x'}\left(K^{(\mathbf{s}_1)}((x, y), (x', y'))\right)\right), \tag{55}$$

and

$$K^{(\mathbf{e}_2)}(x, x') \quad = \quad \mathbb{E}_{z|x}\left(\mathbb{E}_{z'|x'}\left(K^{(\mathbf{s}_2)}((x, z), (x', z'))\right)\right). \tag{56}$$

Theorem 3. *For every $t \geq 1$, the effective function $p^{(\mathbf{e})}_{t+1}(x)$ (defined in (50)) is related to the preceding effective function $p^{(\mathbf{e})}_t(x)$ by the effective alternating-diffusion*

$$p^{(\mathbf{e})}_{t+1}(x) = \begin{cases} \left(D^{(\mathbf{e}_1)}\left(p^{(\mathbf{e})}_t\right)\right)(x), & t = 2m \\ \left(D^{(\mathbf{e}_2)}\left(p^{(\mathbf{e})}_t\right)\right)(x), & t = 2m+1 \end{cases} \tag{57}$$

where $D^{(\mathbf{e}_1)}$ and $D^{(\mathbf{e}_1)}$ are defined in (53) and (54), respectively.

Proof. By (36) and (30),

$$p_{2m+1}(x, y, z) = \mathbb{E}_{x',y',z'}\left(K^{(\mathbf{s}_1)}\left((x, y), (x', y')\right) p_{2m}(x', y', z')\right). \tag{58}$$

By Lemma 2, for $m > 0$, the propagated function $p_{2m}(x', y', z')$ does not depend on the variable y', and

$$p_{2m}(x', y', z') = p^{(\mathbf{i}_2)}_{2m}(x', z') \tag{59}$$

where $p^{(\mathbf{i}_2)}_{2m}(x, z)$ is the intermediate function of p_{2m} on $\mathcal{X} \times \mathcal{Z}$, specified in (45).

Substituting (59) into (58), and rearranging the expression yields

$$p_{2m+1}(x, y, z) = \mathbb{E}_{x',y'}\left(K^{(\mathbf{s}_1)}\left((x, y), (x', y')\right) \mathbb{E}_{z'|x'}\left(p^{(\mathbf{i}_2)}_{2m}(x', z')\right)\right). \tag{60}$$

Since z' does not appear in the kernel $K^{(\mathbf{s}_1)}\left((x, y), (x', y')\right)$, it is simply integrated out, so that by (51),

$$p_{2m+1}(x, y, z) = \mathbb{E}_{x',y'}\left(K^{(\mathbf{s}_1)}\left((x, y), (x', y')\right) p^{(\mathbf{e})}_{2m}(x')\right). \tag{61}$$

Taking the expected value given X over Y and Z yields,

$$\mathbb{E}_{y,z|x}\left(p_{2m+1}(x,y,z)\right) = \mathbb{E}_{y,z|x}\left(\mathbb{E}_{x',y'}\left(K^{(\mathbf{s}_1)}\left((x,y),(x',y')\right)p^{(\mathbf{e})}_{2m}(x')\right)\right). \tag{62}$$

But by (50), the left side of (62) is the effective function $p^{(\mathbf{e})}_{2m+1}(x)$,

$$p^{(\mathbf{e})}_{2m+1}(x) = \mathbb{E}_{y,z|x}\left(\mathbb{E}_{x',y'}\left(K^{(\mathbf{s}_1)}\left((x,y),(x',y')\right)p^{(\mathbf{e})}_{2m}(x')\right)\right). \tag{63}$$

Rearranging the right side of (63), and using the independence given X of Y and Z,

$$p^{(\mathbf{e})}_{2m+1}(x) = \mathbb{E}_{x'}\left(\mathbb{E}_{y|x}\left(\mathbb{E}_{y'|x'}\left(K^{(\mathbf{s}_1)}\left((x,y),(x',y')\right)\right)\right)p^{(\mathbf{e})}_{2m}(x')\right), \tag{64}$$

so that by (55),

$$p^{(\mathbf{e})}_{2m+1}(x) = \mathbb{E}_{x'}\left(K^{(\mathbf{e}_1)}(x,x')p^{(\mathbf{e})}_{2m}(x')\right). \tag{65}$$

Equation (57), in the case $t = 2m$, is obtained from (65) using (53). The proof for the case $t = 2m+1$ is analogous and is omitted for the sake of brevity.

□

Finally, we combine the hidden effective diffusion operators and define the hidden *effective alternating-diffusion operator* $D^{(\mathbf{e})}$ by

$$D^{(\mathbf{e})} = D^{(\mathbf{e}_1)} \circ D^{(\mathbf{e}_2)}. \tag{66}$$

The following corollary is an immediate result from Theorem 3.

Corollary 4. *The subsequence of effective propagated function $p^{(\mathbf{e})}_t(x)$ at odd steps $t = 2m+1$, is given by*

$$p^{(\mathbf{e})}_{2m+3}(x) = \left(D^{(\mathbf{e})}(p^{(\mathbf{e})}_{2m+1})\right)(x). \tag{67}$$

where $m \geq 0$ is a nonnegative integer.

In other words, the subsequence of effective functions $p^{(\mathbf{e})}_{2m+1}(x)$ at odd steps is analogous to a standard diffusion propagation on the common hidden variable X as defined in (13):

$$p^{(\mathbf{e})}_{2m+1}(x) = \left(\left(D^{(\mathbf{e})}\right)^m (p^{(\mathbf{e})}_1)\right)(x). \tag{68}$$

where m is a nonnegative integer. Although the effective functions on $\mathcal{X}$ are not computed, the appropriate seminorm of the difference between effective functions can be computed, as we will show in Section 4.4.

4.4 The hidden effective alternating-diffusion distance

Let $\|\cdot\|_M$ be the hidden seminorm of effective functions $p^{(\mathbf{e})}(x)$ on $\mathcal{X}$, defined by

$$\|p^{(\mathbf{e})}\|_M = \sqrt{\mathbb{E}_{x,x'}\left(p^{(\mathbf{e})}(x)M(x,x')p^{(\mathbf{e})}(x')\right)} \tag{69}$$

where $M(x,x')$ is given by

$$M(x,x') = \mathbb{E}_{x'',z''}\left(\mathbb{E}_{z|x}\left(K^{(\mathbf{s}_2)}\left((x'',z''),(x,z)\right)\right)\mathbb{E}_{z'|x'}\left(K^{(\mathbf{s}_2)}\left((x'',z''),(x',z')\right)\right)\right) \tag{70}$$

and $K^{(\mathbf{s}_2)}$ is defined in (34). We note that z, z', z'' and x'' appear in (70) only as dummy variables in integration.

The following theorem shows that the hidden seminorm of a difference between two hidden effective propagated functions $p^{(\mathbf{e})}_{2m+1}(x)$ and $q^{(\mathbf{e})}_{2m+1}(x)$ defined in (36) can be computed without explicitly using these functions on $\mathcal{X}$, but rather via the corresponding observable propagated functions $p_{2m+2}(x,y,z)$ and $q_{2m+2}(x,y,z)$ on $\mathcal{X}\times\mathcal{Y}\times\mathcal{Z}$.

Theorem 5. *Suppose that $\{p_t(x,y,z)\}_{t=0}^{\infty}$ and $\{q_t(x,y,z)\}_{t=0}^{\infty}$ are two sequences of propagated functions as defined in (36), and that $\{p^{(\mathbf{e})}_t(x)\}_{t=1}^{\infty}$ and $\{q^{(\mathbf{e})}_t(x)\}_{t=1}^{\infty}$ are the corresponding sequences of effective propagated functions as defined in (50). Then,*

$$\|p^{(\mathbf{e})}_{2m+1}(x) - q^{(\mathbf{e})}_{2m+1}(x)\|_M = \|p_{2m+2}(x,y,z) - q_{2m+2}(x,y,z)\|_\pi, \tag{71}$$

where $\|\cdot\|_M$ is defined in (69) and $\|\cdot\|_\pi$ is defined in (39).

Proof. We introduce the notation $w_t(x,y,z)$ to denote the difference

$$w_t(x,y,z) = p_t(x,y,z) - q_t(x,y,z). \tag{72}$$

Due to the linearity of the diffusion operators, the sequence w_j can be expressed as a sequence of propagated functions, defined in (36):

$$w_{t+1}(x,y,z) = \begin{cases} \left(D^{(\mathbf{s}_1)}(w_t)\right)(x,y,z), & t = 2m \\ \left(D^{(\mathbf{s}_2)}(w_t)\right)(x,y,z), & t = 2m+1 \end{cases}. \tag{73}$$

Similarly, let $w^{(\mathbf{e})}_t(x)$ be the difference $w^{(\mathbf{e})}_t(x) = p^{(\mathbf{e})}_t(x) - q^{(\mathbf{e})}_t(x)$. Then, due to linearity it is the effective function of $w_t(x,y,z)$ as defined in (50):

$$w^{(\mathbf{e})}_t(x) = \mathbb{E}_{y,z|x}\left(w_t(x,y,z)\right). \tag{74}$$

By a similar argument to (61) in the proof of Theorem 3,

$$w_{2m+2}(x'', y'', z'') = \mathbb{E}_{x', z'} \left(K^{(\mathbf{s}_2)} \left((x'', z''), (x', z') \right) w^{(\mathbf{e})}_{2m+1}(x') \right). \tag{75}$$

Therefore, by (50),

$$\begin{aligned} \left(\| w_{2m+2} \|_{\pi} \right)^2 = \mathbb{E}_{x'', y'', z''} & \Big(\mathbb{E}_{x,z} \left(K^{(\mathbf{s}_2)} \left((x'', z''), (x, z) \right) w^{(\mathbf{e})}_{2m+1}(x) \right) \\ & \cdot \mathbb{E}_{x', z'} \left(K^{(\mathbf{s}_2)} \left((x'', z''), (x', z') \right) w^{(\mathbf{e})}_{2m+1}(x') \right) \Big). \end{aligned} \tag{76}$$

Due to the linearity of the operations, we have

$$\begin{aligned} \left(\| w_{2m+2} \|_{\pi} \right)^2 = \mathbb{E}_{x} \mathbb{E}_{x'} & \Big(w^{(\mathbf{e})}_{2m+1}(x) w^{(\mathbf{e})}_{2m+1}(x') \\ & \cdot \mathbb{E}_{x'', z''} \Big(\mathbb{E}_{z|x} \left(K^{(\mathbf{s}_2)} \left((x'', z''), (x, z) \right) \right) \\ & \cdot \mathbb{E}_{z'|x'} \left(K^{(\mathbf{s}_2)} \left((x'', z''), (x', z') \right) \right) \Big) \Big). \end{aligned} \tag{77}$$

Using (69) and (70), we obtain

$$\| w^{(\mathbf{e})}_{t} \|_{M} = \| w_{t+1} \|_{\pi}. \tag{78}$$

□

The importance of Theorem 5 is that the expression $\|p_{2m+2} - q_{2m+2}\|_{\pi}$ has a discrete counterpart, and is approximated by the Euclidean norm of the difference between propagated discrete functions (see (27)). In other words, the theorem gives a method for approximating $\|p^{(\mathbf{e})}_{2m+1} - q^{(\mathbf{e})}_{2m+1}\|_{M}$ based on the given sample set of measurements from the two sensors.

We define the hidden *effective alternating-diffusion distance* $d^{(\mathbf{e})}_{2m+1}$ by

$$d^{(\mathbf{e})}_{2m+1} \left((x_i, y_i, z_i), (x_j, y_j, z_j) \right) = \| p^{(\mathbf{e})}_{2m+1}(x'') - q^{(\mathbf{e})}_{2m+1}(x'') \|_{M}. \tag{79}$$

where $\{p^{(\mathbf{e})}_{t}\}_{t=1}^{\infty}$ and $\{p^{(\mathbf{e})}_{t}\}_{t=1}^{\infty}$ are sequences of effective functions defined in (50) associated with the sequences of propagated functions $\{p_t\}_{t=0}^{\infty}$ and $\{p_t\}_{t=0}^{\infty}$, respectively, where p_0 and q_0 are delta functions at (x_i, y_i, z_i) and (x_j, y_j, z_j), respectively.

It follows from Theorem 5, (40), (41) and (79) that the effective alternating-diffusion distance is approximated by the distance d_{2m+2}, computed by the algorithm:

$$d^{(\mathbf{e})}_{2m+1} \left((x_i, y_i, z_i), (x_j, y_j, z_j) \right) \approx C \cdot d_{2m+2}(i, j). \tag{80}$$

In conclusion, the distance computed by the algorithm approximates a diffusion distance corresponding to a diffusion operator on the common variable X.

5 Implementation

In this section, we briefly describe several modifications to the algorithm presented in the paper. Some of these modifications are common in the implementation of diffusion algorithms.

In our discussion and analysis we have required the normalized kernels, such as, $\mathbf{K}^{(\mathbf{s}_1)}$ and $\mathbf{K}^{(\mathbf{s}_2)}$ defined in (22), to be Markovian. This choice is convenient because it can be interpreted as a random process on a graph. We have also assumed a Gaussian form of the affinity matrix. However, we note that these assumptions can be relaxed and various "local" kernels, not necessarily Gaussian or Markovian, can be used. In particular, in diffusion geometry, row-stochastic matrices are often used rather than column-stochastic matrices.

In some datasets, where the scales of the distances between samples varies significantly, the graph specified by the affinity matrices (such as (21)) may have both highly connected vertices and vertices that are effectively isolated. This may imply that the constant kernel scaling ε is inadequate. Therefore, the expression for the affinity matrices is often replaced with

$$W_{ij} = \exp\left(-\frac{\|s_i - s_j\|^2}{\sqrt{\varepsilon_i}\sqrt{\varepsilon_j}}\right), \tag{81}$$

where ε_i is a local scaling factor around the i point. Typically, $\sqrt{\varepsilon_i}$ is taken to be in the order of the distance from s_i to some of its nearest neighbors.

In practice, mainly due to noise, the data samples often do not lie on the underlying manifold, but rather scattered around it. We note that according to the noise analysis presented in [30], setting the diagonal terms of the affinity matrix to zero makes the manifold learning in kernel-based method more robust to noise.

6 Experimental results

6.1 Simulated data I

We simulate the three hidden variables (X, Y, Z) as three statistically independent uniform random variables, i.e.,

$$X \sim \pi_x(X) = U[0,1],\ Y \sim \pi_y(Y) = U[0,1],\ Z \sim \pi_z(Z) = U[0,1] \quad (82)$$

We generate $n = 1000$ samples (x_i, y_i, z_i) of (X, Y, Z), and define two sets of simulated samples in $\mathbb{R}^3$ by

$$s_i^{(1)} = \begin{pmatrix} (R + r^{(1)}\cos(2\pi y_i))\cos(2\pi x_i) \\ (R + r^{(1)}\cos(2\pi y_i))\sin(2\pi x_i) \\ r^{(1)}\sin(2\pi y_i) \end{pmatrix} \quad (83)$$

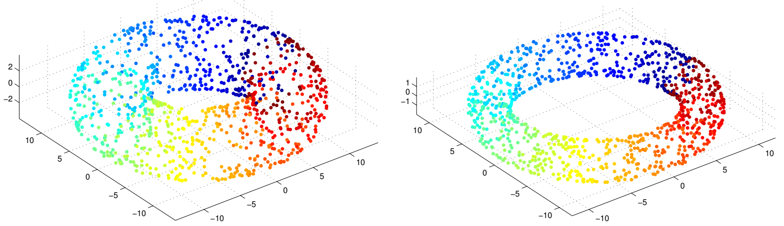
and

$$s_i^{(2)} = \begin{pmatrix} (R + r^{(2)}\cos(2\pi z_i))\cos(2\pi x_i) \\ (R + r^{(2)}\cos(2\pi z_i))\sin(2\pi x_i) \\ r^{(2)}\sin(2\pi z_i) \end{pmatrix} \quad (84)$$

where $R = 10$, $r^{(1)} = 4$ and $r^{(2)} = 2$, for $i = 1, \dots, n$. Thus, each set of measurements lies on a different 2-dimensional torus embedded in $\mathbb{R}^3$, where the "major angle" X is common and the respective "minor angle", Y or Z, is different. In other words, $s_i^{(1)}$ and $s_i^{(2)}$ are two samples in $\mathbb{R}^3$ on two different tori, where their major angle parametrization the two tori is common, and their respective minor angles are independent. See Fig. 3 for illustration.

We apply Algorithm 1 to the two sets $\left\{s_i^{(1)}, s_i^{(2)}\right\}_{i=1}^{1000}$ and obtain a corresponding diffusion maps embedding $\{\tilde{x}_i\}_{i=1}^{1000}$ based on the diffusion distances from (24) (For details about diffusion maps see A). Figure 4 presents a scatter plot of the 2-dimensional embedding. The embedded samples are colored according to the ground truth/common variable X – the "major angle". We observe that the two new coordinates $\tilde{x}_i = (\tilde{x}_{i,1}, \tilde{x}_{i,2})$ parametrize the common variable, approximately via $(\tilde{x}_{i,1}, \tilde{x}_{i,2}) \approx (a\cos(2\pi x_i + c), a\sin(2\pi x_i + c))$, where a and c are some constants.

To demonstrate the challenges stemming from the nonlinearity in this example, we apply CCA to $\left\{s_i^{(1)}, s_i^{(2)}\right\}_{i=1}^{1000}$. Figure 5 depicts the first two common components obtained by CCA. We observe that the CCA parametrization merely projects the samples onto the first two coordinates of the measurements, without extracting the "major angle" (the true common variable) nor without suppressing the "minor angle".



(a) (b)

Figure 3: (Simulation I) The two sets of measurement samples lying on two tori. The samples are colored according to their common variable – the major angle parametrizes each of the tori. (a) $s_i^{(1)}, i = 1, \ldots, 1000$. (b) $s_i^{(2)}, i = 1, \ldots, 1000$.

To illustrate the alternating-diffusion procedure, we plot in Fig. 6 a propagated function after 10 alternating-diffusion steps initialized at an arbitrary sample (marked in green). The color represents the value of the propagated function on each point, where red denotes large values and blue denotes small values. Figure 6 demonstrates that after 10 alternating-diffusion steps, the propagated function "covers" the entire range of the minor angle, but only a small neighborhood of the major angle is covered. To further demonstrate the alternating-diffusion, in Fig. 7, we map the tori back to the hidden variable (X, Y) and (X, Z).

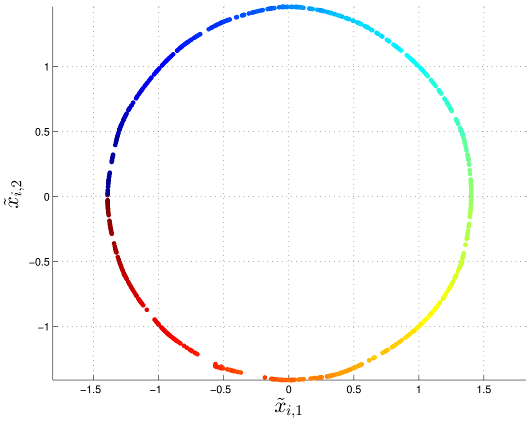
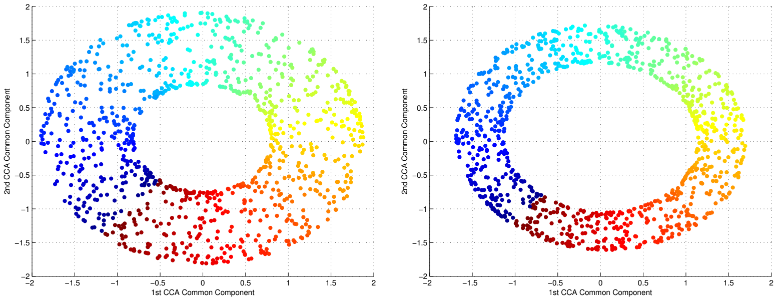


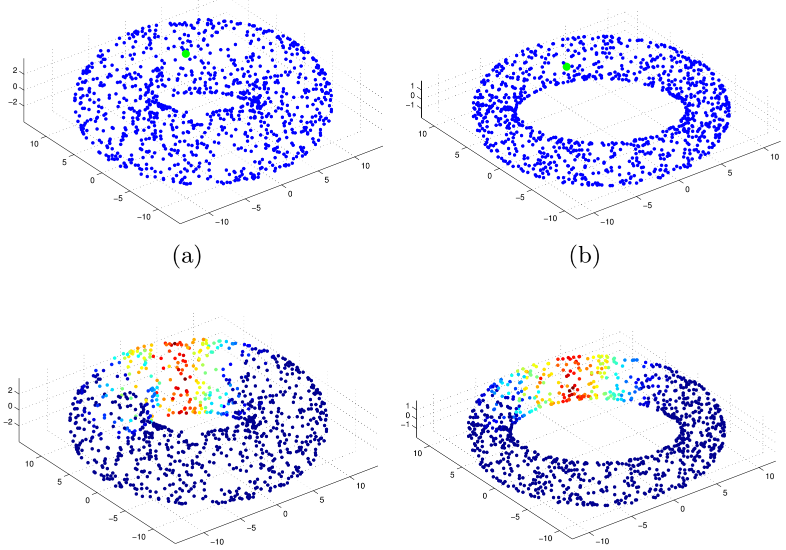
Figure 4: (Simulation I) A 2-dimensional parametrization of the common variable obtained by Algorithm 1 applied to the two measurement sets $\left\{s_i^{(1)}, s_i^{(2)}\right\}_{i=1}^{1000}$ of samples lying on two tori. The parametrized samples are colored according to the true value of the common variable.



(a)

(b)

Figure 5: (Simulation I) The parametrization of the common variable extracted from the two measurement sets $\left\{s_i^{(1)}, s_i^{(2)}\right\}_{i=1}^{1000}$. The parametrized samples are colored according to the true value of the common variable. (a) The samples $s_i^{(1)}$ projected on the first two common components obtained by CCA. (b) The samples $s_i^{(2)}$ projected on the first two common components obtained by CCA..



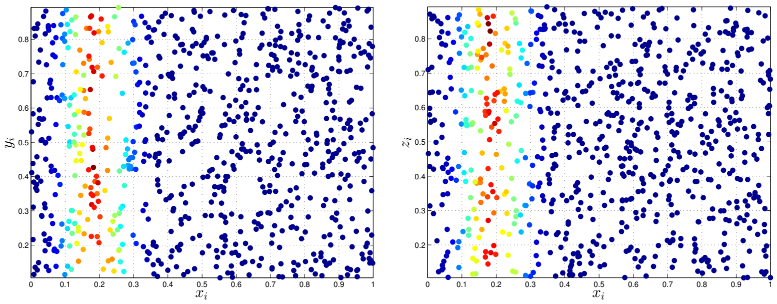
(a)

(b)

(c)

(d)

Figure 6: (Simulation I) The propagation of functions. (a) The initial sample in the space of the measurement set $\left\{s_i^{(1)}\right\}$ marked in green. (b) The initial sample in the space of the measurement set $\left\{s_i^{(2)}\right\}$ marked in green. (c) The propagated function after 10 alternating-diffusion steps in the space of the measurement set $\left\{s_i^{(1)}\right\}$. (b) The propagated function after 10 alternating-diffusion steps in the space of the measurement set $\left\{s_i^{(2)}\right\}$.



(a) (b)

Figure 7: (Simulation I) The propagated distribution after 10 alternating-diffusion steps. (a) The alternating-diffusion in (X, Y). (b) The alternating-diffusion in (X, Z).

6.2 Simulated data II

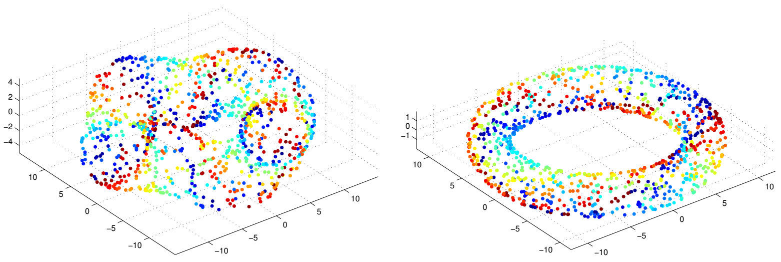
We repeat the experiment with "twisted" tori. Now, the measurement samples are given by

$$s_i^{(1)} = \begin{pmatrix} (R + r^{(1)}(1 + \rho\cos(2\pi x_i)))\cos(2\pi(x_i + \theta y_i))\cos(2\pi y_i) \\ (R + r^{(1)}(1 + \rho\cos(2\pi x_i)))\cos(2\pi(x_i + \theta y_i))\sin(2\pi y_i) \\ r^{(1)}(1 + \rho\cos(2\pi x_i))\sin(2\pi(x_i + \theta y_i)) \end{pmatrix} \quad (85)$$

and

$$s_i^{(2)} = \begin{pmatrix} (R + r^{(2)}\cos(2\pi(x_i - 2z_i)))\cos(2\pi z_i) \\ (R + r^{(2)}\cos(2\pi(x_i - 2z_i)))\sin(2\pi z_i) \\ r^{(2)}\sin(2\pi(x_i - 2z_i))) \end{pmatrix} \quad (86)$$

where $\theta = 4$ and $\rho = 0.25$, for $i = 1, \ldots, n$.



(a)

(b)

Figure 8: (Simulation II) The two sets of measurement samples lying on two "twisted" tori. The samples are colored according to their common variable. (a) $s_i^{(1)}, i = 1, \ldots, 1000$. (b) $s_i^{(2)}, i = 1, \ldots, 1000$.

We generate $n = 1000$ samples of X, Y, and Z and their corresponding measurements $S^{(1)}$ and $S^{(2)}$ in $\mathbb{R}^3$. The new sets of measurements are presented in Fig. 8.

Figure 9 presents the 2-dimensional embedding of the sets obtained by Algorithm 1 and diffusion maps. Similarly to the previous experiment, the obtained new coordinates $\tilde{x}_i = (\tilde{x}_{i,1}, \tilde{x}_{i,2})$ parametrize the common variable. Figure 10 presents the results obtained by CCA in this case. As expected, the linear CCA does not extract the common variable in these nonlinear settings.

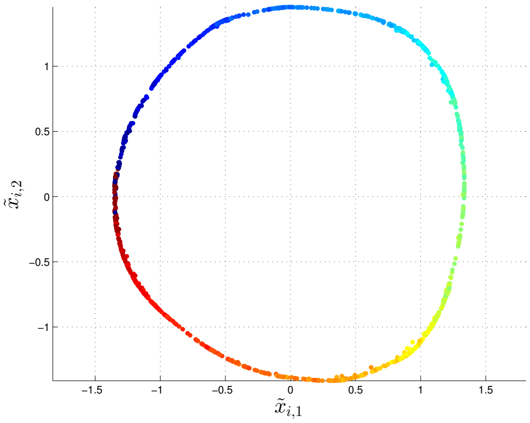
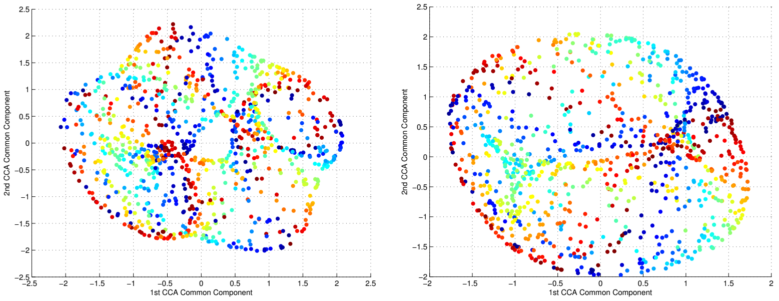


Figure 9: (Simulation II) A 2-dimensional parametrization of the common variable obtained by Algorithm 1 applied to the two measurement sets $\left\{s_i^{(1)}, s_i^{(2)}\right\}_{i=1}^{1000}$ of samples lying on the "twisted" tori. The parametrized samples are colored according to the true value of the common variable.



(a)

(b)

Figure 10: (Simulation II) The parametrization of the common variable extracted from the two measurement sets $\left\{s_i^{(1)}, s_i^{(2)}\right\}_{i=1}^{1000}$ on the "twisted" tori. The parametrized samples are colored according to the true value of the common variable. (a) The samples $s_i^{(1)}$ projected on the first two common components obtained by CCA. (b) The samples $s_i^{(2)}$ projected on the first two common components obtained by CCA.

6.3 Toy problem

We now revisit the toy problem from Section 1.1 and describe it in more detail. Yoda's platform rotates clockwise and completes approximately 310 cycles in the duration of the experiment, the bulldog's platform rotates counterclockwise and approximately completes 450 cycles, and the bunny's platform rotates counterclockwise and approximately completes 270 cycles. See Fig. 1 for the setup illustration.

The sample set of $n = 4000$ pairs of synchronous image snapshots (measurements) taken at the same time by Camera 1 and Camera 2 is denoted by $\{s_i^{(1)}, s_i^{(2)}\}_{i=1}^n$. In other words, $s_i^{(1)}$ is an image taken by Camera 1, and $s_i^{(2)}$ an image taken at the same time by Camera 2.

We apply Algorithm 1 to the two sets of images $\{s_i^{(1)}, s_i^{(2)}\}_i$. We note that the chronological order is not taken into account, although the images are collected in a sequence. For comparison, we also use diffusion maps (see A) to analyze each set separately.

Figure 11 presents 2-D scatter plots of two separate diffusion maps embeddings, i.e., $\tilde{x}_i = (\tilde{x}_{i,1}, \tilde{x}_{i,2})$, created for the snapshots taken by Camera 1 and the snapshots taken by Camera 2. We observe that these diffusion maps are torus-like. Figures 12 and 13 present the embedding generated by Algorithm 1.

To demonstrate the notion of the "intrinsic" distance implied by the embedding, we inset snapshots corresponding to several embedded points. In Fig. 11, we observe that close embedded points correspond to similar snapshots, and that remote points can correspond to snapshots where the bulldog is in the same rotation angle. This suggests that the distance in the embedding does not respect the common variable in this experiment – the rotation angle of the bulldog. In Figs. 12 and 13, we observe that distance in the embedding does respect the rotation angle of the bulldog: close (remote) embedded points correspond to snapshots where the bulldog is in the same (different) rotation angle.

Figures 14(a-c) show the absolute value of the discrete Fourier transform of the principal component $\tilde{x}_{i,1}$ obtained from the diffusion maps applications to the data from Camera 1, Camera 2, and from Algorithm 1. Figure 14(a) shows a sharp peak centered at 310. This suggests that the principal component obtained based on snapshots taken by Camera 1 mainly captures the frequency corresponding to the rotation angle of Yoda. In Fig. 14(b), a sharp peak is obtained at 270 and suggests that the diffusion maps embedding based on the snapshots taken by Camera 2 captures the rotation angle of the bunny. Figure 14(c) shows that the diffusion maps embedding obtained by Algorithm 1 captures the rotation angle of the bulldog, the common source of variability, as we obtain a sharp peak at 450.

The results from Figs. 11, 12, 13 and 14 imply a successful extraction of the common variable by the alternating-diffusion algorithm. The separate analyses of each set suggests that Yoda and the bunny are the dominant objects in the snapshots acquired by Camera 1 and Camera 2, respectively. Nevertheless, the alternating-diffusion is able to extract the "weaker," but common, bulldog.

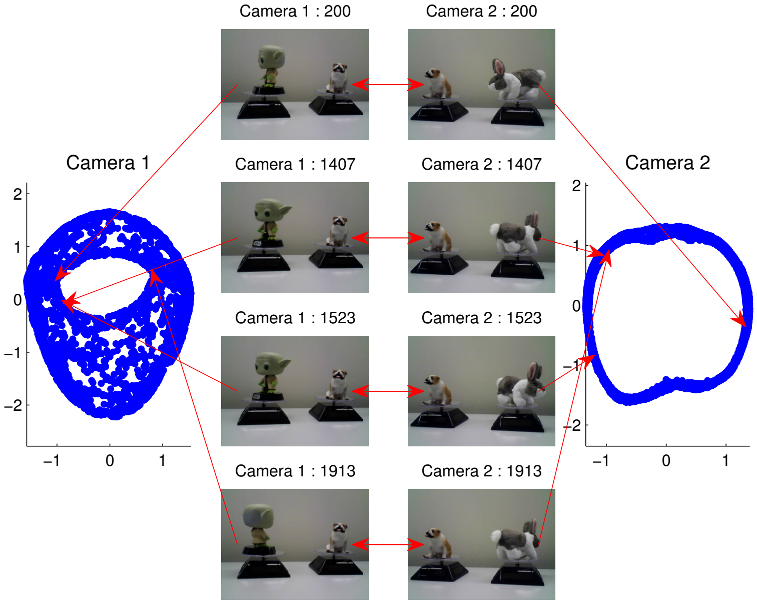


Figure 11: (Toy example) Scatter plots of the diffusion maps embeddings $\tilde{x}_i = (\tilde{x}_{i,1}, \tilde{x}_{i,2})$. A diffusion map of the snapshots taken by Camera 1 is presented on the left, and a diffusion map of the snapshots taken by Camera 2 is presented on the right. In each row in the middle, we present a pair of snapshots taken by Camera 1 and Camera 2 at the same time. the red arrows point to the place on the diffusion map where each snapshot was embedded. The bulldog is approximately in the same position in all the snapshots presented here.

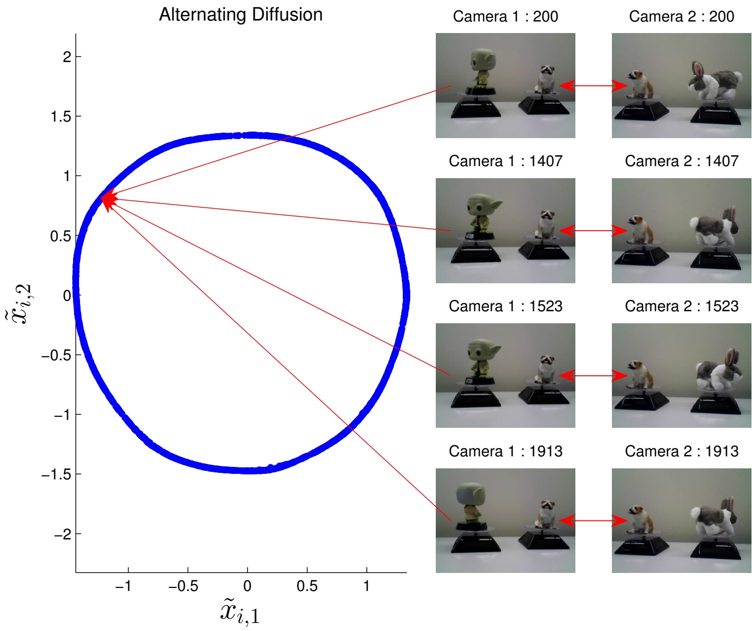


Figure 12: (Toy example) Scatter plot of the diffusion map embeddings $\tilde{x}_i = (\tilde{x}_{i,1}, \tilde{x}_{i,2})$ generated by Algorithm 1. The same pair of images presented in Fig. 11 are presented here, each pair is mapped to a point in the scatter plot. Since the bulldog was in the same position in all the snapshots, all the pairs are mapped to the same point.

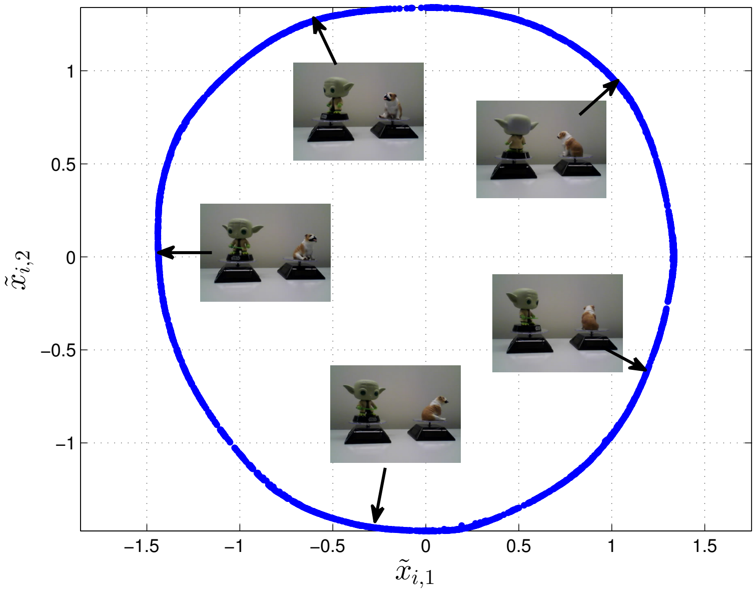
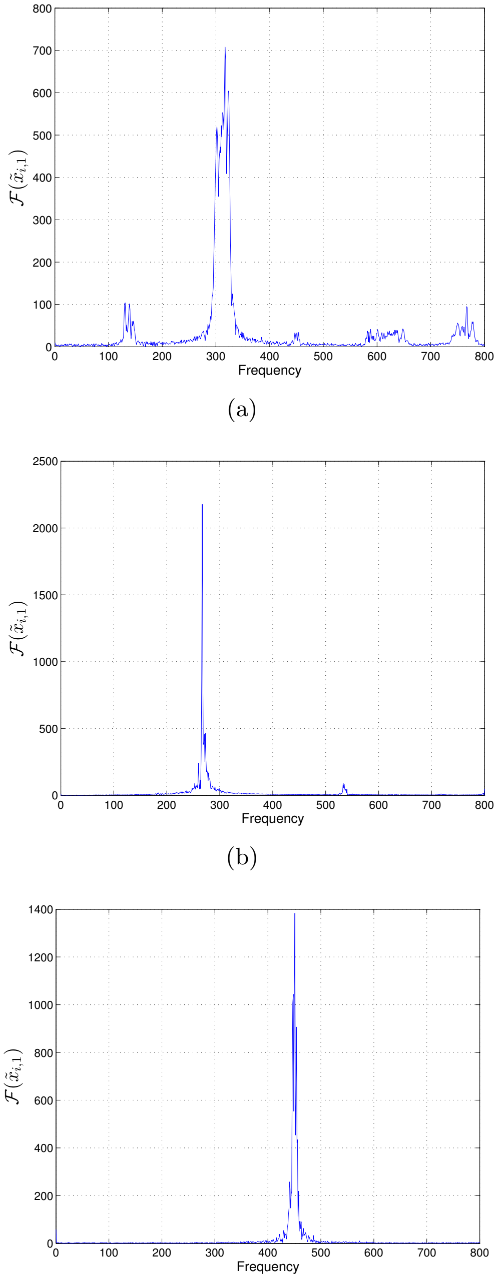


Figure 13: (Toy example) Scatter plot of the diffusion map embeddings $\tilde{x}_i = (\tilde{x}_{i,1}, \tilde{x}_{i,2})$ generated by Algorithm 1. Sample snapshots from Camera 1 are presented to illustrate the embedding where the bulldog is in different rotation angles.



(a)

(b)

(c)

Figure 14: (Toy example) The absolute value of the discrete Fourier transform of the principal component $\tilde{x}_{i,1}$ obtained from the diffusion maps applications to the snapshots taken by Camera 1 (a), Camera 2 (b), and from Algorithm 1 (c).

7 Conclusions

An alternating-diffusion method has been presented for the extraction of the common source of variability from measurements in multiple sensors. The method uses diffusion operators which are computed for each sensor separately from its measurements. The information from the multiple sensors is combined by creating a new alternating-diffusion operator, which is the product of the separate operators. We have shown that this alternating-diffusion operator is, in effect, a diffusion operator computed from data where the common source of variability is the only source of variability.

The method has been demonstrated on simulated data and on an image-processing toy problem. Alternating-diffusion is a data-driven method that does not require strong assumptions on the nature of the sensors. Therefore, this method is not restricted to computer-vision and image processing applications, nor to multi-view problems, and it applies to multimodal problems, where the sensors are not necessarily of the same type. Moreover, the method is not restricted to physical measurements, and the different "sensors" can be replaced by different processing pipelines. Such applications will be demonstrated in future work.

Acknowledgments

The authors would like to thank Raphy Coifman for fruitful discussions and helpful suggestions, and William Leeb for his help.

A Diffusion maps

In this appendix we present a brief description of the diffusion maps algorithm, presented in [31].

Algorithm 2 Diffusion Maps

```
Input: Samples $\{s_i\}_{i=1}^{n}$ and a corresponding distance metric $d(i, j)$.
Output: Embedding $\{\tilde{s}_i\}_{i=1}^{n}$ and diffusion distances $\tilde{d}_t(i, j)$.

1. Compute the affinity matrix $\mathbf{W}$ as follows:
       $W_{ij} = \exp\left(-\frac{\|s_i - s_j\|^2}{\varepsilon}\right), \ \forall i, j = 1, \ldots, n.$
2. Compute the diagonal normalization matrix $Q_{ii} = \left(\sum_{j=1}^{n} W_{ij}\right)^{-1}$.
3. Normalize the kernel $\tilde{K} = QWQ$.
4. Compute the second diagonal normalization matrix $\tilde{Q}_{ii} = \left(\sum_{j=1}^{n} \tilde{K}_{ij}\right)^{-1/2}$.
5. Normalize the kernel $K = \tilde{Q}\tilde{K}\tilde{Q}$ for the second time.
6. Compute the $n$ eigenvectors $u_0, u_1, ...u_{n-1}$ and eigenvalues $\lambda_0, \lambda_1, ..., \lambda_{n-1}$ of $K$.
7. Create the $n - 1$ weighted vectors $\varphi_1, \varphi_2, ...\varphi_{n-1}$ by $\varphi_i(j) = \frac{u_i(j)}{u_0(j)}$.
8. Build the $d$-dimensional embedding for $t \geq 0$ of each point $i = 1, \ldots, n$ by $\tilde{s}_i = \left(\lambda_1^t \varphi_1(i), \lambda_2^t \varphi_2(i), ..., \lambda_{n-1}^t \varphi_{n-1}(i)\right)^T$.
9. Define the diffusion distance between each two points $i, j$ by $\tilde{d}_t(i, j) = \|\tilde{s}_i - \tilde{s}_j\|$.
```

References

[1] J. B. Tenenbaum, V. de Silva, J. C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 260 (2000) 2319–2323.

[2] S. T. Roweis, L. K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 260 (2000) 2323–2326.

[3] D. L. Donoho, C. Grimes, Hessian eigenmaps: New locally linear embedding techniques for high-dimensional data, Proc. Nat. Acad. Sci. 100 (2003) 5591–5596.

[4] M. Belkin, P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, Neural Comput. 15 (2003) 1373–1396.

[5] R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, S. W. Zucker, Geometric diffusions as a tool for harmonic analysis and structure definition of data: diffusion maps, Proc. Nat. Acad. Sci. 102 (21) (2005) 7426–7431.

[6] B. Nadler, S. Lafon, R. Coifman, I. G. Kevrekidis, Diffusion maps, spectral clustering and eigenfunctions of Fokker-Planck operators, Neural Information Process. Systems (NIPS) 18.

[7] R. Coifman, S. Lafon, Diffusion maps, Appl. Comput. Harmon. Anal. 21 (2006) 5–30.

[8] B. Nadler, S. Lafon, R. Coifman, I. G. Kevrekidis, Diffusion maps, spectral clustering and reaction coordinates of dynamical systems, Appl. Comput. Harmon. Anal. 21 (2006) 113–127.

[9] A. Singer, R. Coifman, Non-linear independent component analysis with diffusion maps, Appl. Comput. Harmon. Anal. 25 (2008) 226–239.

[10] Y. Keller, R. Coifman, S. Lafon, S. W. Zucker, Audio-visual group recognition using diffusion maps, IEEE Transactions on Signal Processing 58 (1) (2010) 403–413. `doi:10.1109/TSP.2009.2030861`.

[11] H. Hotelling, Relations between two sets of variates, Biometrika 28 (3/4) (1936) 321. `doi:10.2307/2333955`.

[12] P. L. Lai, C. Fyfe, Kernel and nonlinear canonical correlation analysis, International Journal of Neural Systems 10 (05) (2000) 365–377.

[13] F. R. Bach, M. I. Jordan, Kernel independent component analysis, The Journal of Machine Learning Research 3 (2003) 1–48.

[14] B. Boots, G. J. Gordon, Two-manifold problems with applications to nonlinear system identification, in: Proc. 29th Intl. Conf. on Machine Learning (ICML), 2012.

[15] G. R. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, M. I. Jordan, Learning the kernel matrix with semidefinite programming, The Journal of Machine Learning Research 5 (2004) 27–72.

[16] V. R. de Sa, Spectral clustering with two views, in: ICML workshop on learning with multiple views, 2005.

[17] V. R. de Sa, P. W. Gallagher, J. M. Lewis, V. L. Malave, Multi-view kernel construction, Machine Learning 79 (1-2) (2010) 47–71. doi:10.1007/s10994-009-5157-z.

[18] L. Luo, W. Jia, C. Zhang, Mixed propagation for image retrieval, in: IEEE Fourth International Conference on Multimedia Information Networking and Security (MINES), 2012, pp. 237–240.

[19] Z. Wu, Y. Wang, R. Shou, B. Chen, X. Liu, Unsupervised co-segmentation of 3d shapes via affinity aggregation spectral clustering, Computers & Graphics 37 (6) (2013) 628–637. doi:10.1016/j.cag.2013.05.015.

[20] D. Zhou, C. J. Burges, Spectral clustering and transductive learning with multiple views, in: Proceedings of the 24th international conference on Machine learning, ACM, 2007, pp. 1159–1166.

[21] H.-C. Huang, Y.-Y. Chuang, C.-S. Chen, Affinity aggregation for spectral clustering, in: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE, 2012, pp. 773–780.

[22] A. Kumar, H. Daumé, A co-training approach for multi-view spectral clustering, in: Proceedings of the 28th International Conference on Machine Learning (ICML-11), 2011, pp. 393–400.

[23] J. J.-Y. Wang, Y. Sun, From one graph to many: Ensemble transduction for content-based database retrieval, Knowledge-Based Systems 65 (2014) 31–37.

[24] X. Bai, B. Wang, C. Yao, W. Liu, Z. Tu, Co-transduction for shape retrieval, IEEE Transactions on Image Processing 21 (5) (2012) 2747–2757.

[25] L. Luo, C. Shen, C. Zhang, A. J. van den Hengel, Shape similarity analysis by self-tuning locally constrained mixed-diffusion, IEEE Transactions on Multimedia 15 (5) (2013) 1174–1183.

[26] Y. Zhou, X. Bai, W. Liu, L. J. Latecki, Fusion with diffusion for robust visual tracking, in: Advances in Neural Information Processing Systems, 2012, pp. 2978–2986.

[27] B. Wang, J. Jiang, W. Wang, Z.-H. Zhou, Z. Tu, Unsupervised metric fusion by cross diffusion, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012, pp. 2997–3004.

[28] O. Lindenbaum, A. Yeredor, M. Salhov, Multiview diffusion maps, Preprint.

[29] J. Liu, Y. Yang, M. Shah, Learning semantic visual vocabularies using diffusion distance, Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR) (2009) 461–468.

[30] N. El Karoui, H.-T. Wu, Connection graph Laplacian methods can be made robust to noise, submitted.

[31] S. Lafon, Diffusion maps and geometric harmonics, Ph.D. thesis, Yale University (May 2004).