Nonlinear intrinsic variables and state reconstruction in multiscale simulations

Carmeline J. Dsilva, Ronen Talmon, Neta Rabin, Ronald R. Coifman, and Ioannis G. Kevrekidis

Citation: The Journal of Chemical Physics **139**, 184109 (2013); doi: 10.1063/1.4828457
View online: http://dx.doi.org/10.1063/1.4828457
View Table of Contents: http://scitation.aip.org/content/aip/journal/jcp/139/18?ver=pdfcov
Published by the AIP Publishing

Nonlinear intrinsic variables and state reconstruction in multiscale simulations

Carmeline J. Dsilva,[1,a)] Ronen Talmon,[2,b)] Neta Rabin,[3,c)] Ronald R. Coifman,[2,d)] and Ioannis G. Kevrekidis[1,4,e)]

[1]*Department of Chemical and Biological Engineering, Princeton University, Princeton, New Jersey 08544, USA*
[2]*Department of Mathematics, Yale University, New Haven, Connecticut 06520, USA*
[3]*Department of Exact Sciences, Afeka Tel-Aviv Academic College of Engineering, Tel-Aviv, Israel*
[4]*Program in Applied and Computational Mathematics, Princeton University, Princeton, New Jersey 08544, USA*

(Received 22 July 2013; accepted 8 October 2013; published online 12 November 2013)

Finding informative low-dimensional descriptions of high-dimensional simulation data (like the ones arising in molecular dynamics or kinetic Monte Carlo simulations of physical and chemical processes) is crucial to understanding physical phenomena, and can also dramatically assist in accelerating the simulations themselves. In this paper, we discuss and illustrate the use of nonlinear intrinsic variables (NIV) in the mining of high-dimensional multiscale simulation data. In particular, we focus on the way NIV allows us to functionally merge different simulation ensembles, and *different partial observations of these ensembles*, as well as to infer variables not explicitly measured. The approach relies on certain simple features of the underlying process variability to filter out measurement noise and systematically recover a unique reference coordinate frame. We illustrate the approach through two distinct sets of atomistic simulations: a stochastic simulation of an enzyme reaction network exhibiting both fast and slow time scales, and a molecular dynamics simulation of alanine dipeptide in explicit water. © *2013 AIP Publishing LLC*. [http://dx.doi.org/10.1063/1.4828457]

I. INTRODUCTION

The last decade has witnessed extensive advances in dimensionality reduction techniques: finding meaningful low-dimensional descriptions of high-dimensional data.[1–5] Often, high-dimensional data do not uniformly fill the entire ambient space, but rather lie on a lower-dimensional manifold. Discovering a parameterization of the manifold therefore yields a compact representation of the data, and may lead to efficient analysis and processing schemes. In particular, these developments have the potential to significantly enable the computational exploration of physicochemical problems. Traditionally, state reduction in chemical reaction networks has been done using some *a priori* knowledge about separation of time scales, for example, using the quasi-steady state approximation to reduce a network of chemical reactions.[6] Alternatively, empirical knowledge about the system of interest has been used to develop coarse grained descriptions, such as grouping atoms of a macromolecule into larger "beads" to accelerate simulations.[7–10] The use of manifold learning allows for the systematic discovery of data-driven, low-dimensional descriptions that circumvent the need for such *a priori* knowledge of the system. If the (high-dimensional) data $\mathbf{Y}(t)$ arise from, for example, a molecular dynamics simulation of a macromolecule in solution, or from the stochastic simulation of a complex chemical reaction scheme, the detection of a few good, coarse-grained "reduction coordinates" $\mathbf{x}(t)$ can be invaluable in understanding and predicting system behavior. For example, such reduction coordinates may be the dihedral angles inferred from trajectories of the atoms of a macromolecule.

While the benefits from such reduced descriptions are manifest, a crucial shortcoming of data-driven reduction coordinates is their dependence on the specific data set processed, and not only on the physical model in question. It is well known that, even in the simple linear case of Principal Component Analysis,[11] different data sets on the same low-dimensional hyperplane in the ambient space will lead to different bases spanning the hyperplane—in effect, to different reduction coordinates $\mathbf{x}$. While this can easily be rectified by an affine transformation (see, by analogy, the discussion in Lafon *et al.*[12]), the problem becomes exacerbated when the low-dimensional space is curved (a manifold, rather than a hyperplane) and when different data sets are obtained using different instrumental modalities (such as when one wants to merge molecular dynamics data with, for example, spectral information).

In this paper, we propose a new method to embed data in a low-dimensional space. This method is based on a kernel constructed from a Riemannian metric that is specially designed to exploit the noise/diffusion induced variability of the data in local neighborhoods. By solving the appropriate eigenvector problem, local affinities from the kernel are integrated into a global coordinate system for the data.

[a)]Electronic mail: cdsilva@princeton.edu
[b)]Electronic mail: ronen.talmon@yale.edu
[c)]Electronic mail: netar@afeka.ac.il
[d)]Electronic mail: coifman@math.yale.edu
[e)]Electronic mail: yannis@princeton.edu

We will demonstrate this can produce a *unique* and *consistent* reduction coordinate set, shared by all measurement ensembles and observation modalities. We will call these coordinates Nonlinear Intrinsic Variables (NIV). Embedding data in such a coordinate system allows us to naturally merge different observations of the same system; more importantly, it enables the construction of an empirical mapping between these different observation ensembles, allowing us to complete partial measurements in a test data set from a training data set that consists of *different* observations. To construct this empirical mapping and the associated observers, accurate interpolation tools must be available in the embedding space; to this end, we will demonstrate the use of a multiscale Laplacian Pyramid approach.[13]

We will illustrate our methodologies with two distinct examples. The first is a simulation of two Goldbeter-Koshland modules in an enzyme kinetics model using the Gillespie Stochastic Simulation Algorithm (SSA);[14] in certain parameter regimes, separation of time scales is known to reduce the ordinary differential equation (ODE) model of this kinetic scheme to an effective two-dimensional description.[15] Although this example is rather simple, it will serve as an introduction to our techniques and highlight the main features of the algorithms. The second example is a molecular dynamics simulation (in explicit water) of a simple peptide fragment (alanine dipeptide) whose folding dynamics are known to be described through a small set of physical observables.[16] This example will allow us to compare our approach to more common techniques, such as diffusion maps[17] for dimensionality reduction and nearest neighbor interpolation for observation reconstruction. The remainder of the paper is structured as follows: in Sec. II we present the Nonlinear Intrinsic Variable formulation and the associated inference method. Section III contains our discussion of Laplacian Pyramids that is used for the completion of partial observations. In Sec. IV, the results of the application of the approach to simulation data from our two illustrative examples are presented and discussed. We conclude with a summary and our perspective on open issues in Sec. V.

II. NONLINEAR INTRINSIC VARIABLES

A. Problem formulation

Let $\mathbf{Y}(t)$ be a high-dimensional measured process in $\mathbb{R}^n$ consisting of n observable variables. We impose two critical assumptions. First, the measured process is assumed to be a manifestation, in an observable domain, of a low-dimensional diffusion process. Thus, it can be expressed by

$$\mathbf{Y}(t) = \mathbf{f}(\mathbf{x}(t)), \tag{1}$$

where $\mathbf{f} : \mathbb{R}^d \to \mathcal{M}$ is an unknown (possibly nonlinear) function, $\mathcal{M} \subset \mathbb{R}^n$ is a d-dimensional manifold, and $\mathbf{x}(t)$ is a diffusion process that consists of d underlying variables (with $d \ll n$). Second, the dynamics of the diffusion process in each of its underlying variables are described by normalized stochastic differential equations as

$$dx_i(t) = a_i(\mathbf{x}(t))dt + dw_i(t),\ i = 1, \ldots, d, \tag{2}$$

where a_i are unknown drift functions and $w_i(t)$ are independent Brownian motions. The independence is our second critical assumption.

Given a sequence of samples $\mathbf{Y}(t),\ t = 1, \ldots, T$, we present an empirical method to construct a unique and consistent reduction coordinate set, represented here by $\mathbf{x}(t)$.[18] Because the empirical method we will describe is independent of the observation function $\mathbf{f}$, we refer to the coordinates of $\mathbf{x}(t)$ as NIV. The available samples $\mathbf{Y}(t)$ may be the result of different measurement functions $\mathbf{f}$ in various observable domains, or they may be partial measurements consisting of merely a subset of the coordinates of the observable domains. The idea is to empirically construct a NIV coordinate system driven entirely by measurements that is invariant to the observation function $\mathbf{f}$ (see Figure 1 for a schematic illustration). We remark that the available data should be "rich enough," i.e., consist of a sufficient amount of historical data with adequate variability, in order to obtain the full empirical model; this is further discussed and demonstrated in the experimental study.

The method consists of the following main principles. (1) The underlying diffusion process implies that a short trajectory of successive samples mainly consists of diffusion noise, and hence, creates a "sphere" of samples in the underlying domain $\mathbb{R}^d$. This sphere is mapped to an ellipse in the observable domain by the measurement function $\mathbf{f}$. In this work, the identification of the associated ellipse of samples according to the time trajectory of our data enables us to estimate the

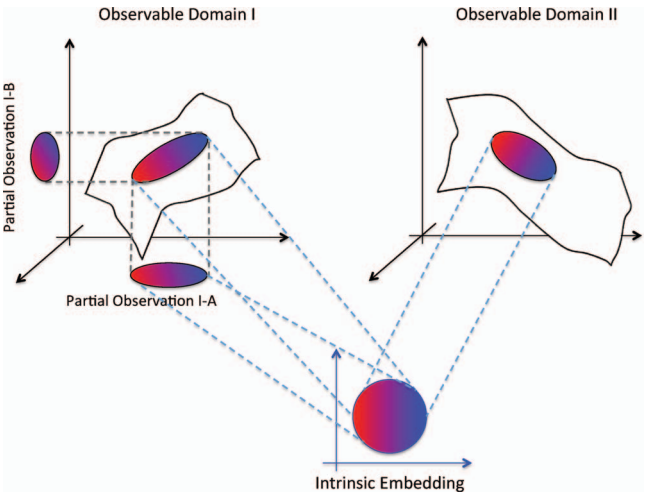

FIG. 1. Illustration of the nonlinear embedding that yields an intrinsic representation independent of the measurement function $\mathbf{f}$. (Bottom) The underlying variables in which the noises are independent with unit variance. The circle illustrates samples from, say, a short trajectory in time that sample a disc on the manifold. (Top left) The first set of observed variables. The ellipse illustrates the mapping of the sphere of the underlying samples into the observable domain via the first observation function. In this sketch, we illustrate that the observations might be *partial*, i.e., might consist of merely a subset of the observed domain variables. (Top right) Second set of observable variables. The ellipse illustrates the mapping of the sphere of the underlying samples into this (different) observable domain via a second observation function.

tangent planes of the observable manifolds *via the principal components of the covariance matrices of the samples in these ellipses*. (2) The principal directions of the tangent planes are utilized to define a Riemannian metric that is shown to be *locally invariant to the measurement function* **f**. (3) The NIV are constructed through the eigenvalue decomposition of a Laplace operator that is built upon a pairwise affinity between the samples, defined using this Riemannian metric.

B. Mahalanobis distance

Let $\mathbf{C}(t)$ be the covariance matrix associated with the measured sample $\mathbf{Y}(t)$. In practice, the covariance matrix can be estimated from a short trajectory of samples in time around the sample $\mathbf{Y}(t)$ by

$$\widehat{\mathbf{C}}(t) = \sum_{\tau=t-L}^{t+L} (\mathbf{Y}(\tau) - \widehat{\boldsymbol{\mu}}(t))(\mathbf{Y}(\tau) - \widehat{\boldsymbol{\mu}}(t))^T, \tag{3}$$

where $\widehat{\boldsymbol{\mu}}(t)$ is the empirical mean of the short trajectory of samples. We define a Riemannian metric between a pair of samples using the associated covariance matrices as

$$d^2(\mathbf{Y}(t), \mathbf{Y}(\tau)) = 2(\mathbf{Y}(t) - \mathbf{Y}(\tau))^T(\widehat{\mathbf{C}}(t) + \widehat{\mathbf{C}}(\tau))^\dagger(\mathbf{Y}(t) - \mathbf{Y}(\tau)); \tag{4}$$

this is the Mahalanobis distance (and † denotes a pseudoinverse, as discussed below). As previously described, the covariance matrices convey the local variability of the measurements and are utilized to explore and learn the tangent planes of the observable manifold. This information is then utilized in (4) to compare a pair of points according to the directions of their respective tangent planes. The Mahalanobis distance is invariant under affine transformations. Thus, by assuming that the observation function **f** is bi-Lipschitz and smooth, and by using local linearization of the function, i.e., $\mathbf{Y}(t) = \mathbf{J}(t)\mathbf{x}(t) + \boldsymbol{\epsilon}(t)$ where $\mathbf{J}(t)$ is the Jacobian of $\mathbf{f}(\mathbf{x}(t))$ and $\boldsymbol{\epsilon}(t)$ is the residual consisting of higher-order terms, it was shown by Singer and Coifman[18] that $\mathbf{C}(t) = \mathbf{J}(t)\mathbf{J}^T(t)$ and that the Mahalanobis distance approximates the Euclidean distance between the corresponding samples of the underlying process to second order, i.e.,

$$\|\mathbf{x}(t) - \mathbf{x}(\tau)\|^2 = d^2(\mathbf{Y}(t), \mathbf{Y}(\tau)) + \mathcal{O}(\|\mathbf{Y}(t) - \mathbf{Y}(\tau)\|^4). \tag{5}$$

This result implies that the Mahalanobis distance is invariant to the measurement function **f**, and hence, it yields the same distances between samples obtained under different observation functions or even partial observations. We would like to note that, in general, **f** being bi-Lipschitz implies that **f** is invertible (on the d-dimensional manifold $\mathcal{M}$). However, in practice, determining whether **f** contains sufficient information and is "rich enough" to completely determine the underlying process is a non-trivial task. In this work, we exploit the fact that $\mathbf{C}(t) = \mathbf{J}(t)\mathbf{J}^T(t)$, which implies that $\mathbf{C}(t)$ is an $n \times n$ positive semidefinite matrix of rank d, to empirically infer the dimension d. According to the spectrum of the local covariance matrices and their corresponding spectral gaps, we approximate the rank of the matrices. Consistent rank estimates among these local covariance matrices are taken to imply that the measurements are "rich enough," and hence, may be good indicators for the dimension d. Since the dimension d of the underlying process is typically considerably smaller than the dimension of the measured process n, the covariance matrix is singular and non-invertible; thus, we use the pseudo-inverse in (4).

C. Laplace operator

The Mahalanobis distance described in Sec. II B enables us to compare observations in terms of the intrinsic variables of the associated underlying diffusion process. In this section, we show how to recover the underlying process itself from the pairwise Euclidean distances through the eigenvectors of a Laplace operator.

Let $\mathbf{W}$ be a pairwise affinity matrix (kernel) based on a Gaussian, whose (t, τ)-th element is given by

$$W_{t,\tau} = \exp\left\{-\frac{d^2(\mathbf{Y}(t), \mathbf{Y}(\tau))}{\varepsilon}\right\}, \tag{6}$$

where ε is the kernel scale, which can be set according to Hein and Audibert[19] and Coifman *et al.*[20] Based on the kernel, we form a weighted graph, where the measurements $\mathbf{Y}(t)$ are the graph nodes and the weight of the edge connecting node $\mathbf{Y}(t)$ to node $\mathbf{Y}(\tau)$ is $W_{t,\tau}$. In particular, a Gaussian kernel exhibits a notion of locality by defining a neighborhood around each measurement $\mathbf{Y}(t)$ of radius ε, i.e., measurements $\mathbf{Y}(\tau)$ such that $d^2(\mathbf{Y}(t), \mathbf{Y}(\tau)) > \varepsilon$ are weakly connected to $\mathbf{Y}(t)$. In practice, we set ε to be the median of the pairwise distances. According to the graph interpretation, this implies a well-connected graph because each measurement is effectively connected to half of the other measurements.[21]

Let $\mathbf{D}$ be a diagonal matrix whose elements are the row sums of $\mathbf{W}$, and let

$$\mathbf{W}^{\text{norm}} = \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2} \tag{7}$$

be a normalized kernel that shares its eigenvectors with the normalized graph-Laplacian $\mathbf{I} - \mathbf{W}^{\text{norm}}$.[22] The diagonal entries of $\mathbf{D}$ serve as estimates for the density of data around each point. The eigenvectors of $\mathbf{W}^{\text{norm}}$, denoted ψ_j, reveal the underlying structure of the data.[17] Specifically, the ith coordinate of the jth eigenvector can be associated with an intrinsic coordinate j of the sample $\mathbf{x}(i)$ of the underlying process. The eigenvectors are ordered such that $|\lambda_1| \geq |\lambda_2| \geq \ldots \geq |\lambda_n|$, where λ_j is the eigenvalue associated with eigenvector ψ_j. Because $\mathbf{W}^{\text{norm}} \sim \mathbf{D}^{-1}\mathbf{W}$, and $\mathbf{D}^{-1}\mathbf{W}$ is row-stochastic, $\lambda_1 = 1$ and ψ_1 is the diagonal of $\mathbf{D}^{1/2}$. The next few eigenvectors can be argued to describe the geometry of the underlying manifold.[17] However, some eigenvectors can be higher harmonics of the same principal direction along the data manifold. This is analogous to how the eigenfunctions $\cos x$ and $\cos 2x$ of the usual Laplacian in one spatial dimension and with no flux boundary conditions are one-to-one with the values of x for $0 \leq x \leq 1$; one must check for correlations between the eigenvectors before selecting those that describe the underlying manifold geometry. The above steps to construct the nonlinear intrinsic variables are summarized in Algorithm I.

ALGORITHM I. Nonlinear Intrinsic Variables construction.

1. Obtain a sequence of high-dimensional observation samples $\mathbf{Y}(t)$.
2. Compute the empirical covariance matrix $\widehat{\mathbf{C}}(t)$ of each sample $\mathbf{Y}(t)$ *in a short window in time* according to (3).
3. Using the samples and their associated covariance matrices, compute the Mahalanobis distance between the observations (4).
4. Build the pairwise affinity matrix $\mathbf{W}$ and the corresponding normalized kernel $\mathbf{W}^{\text{norm}}$ (7).
5. Apply eigenvalue decomposition to the normalized kernel and view the values of its principal eigenvectors (modulo the possibility of "higher harmonics," see text) as the Nonlinear Intrinsic Variables (NIV) of the given observations.

Ignoring the higher harmonics, each retained eigenvector then describes an intrinsic variable for the data set of interest. We must normalize the eigenvectors from different data sets so that the resulting embeddings are consistent. We first scale the eigenvectors so that $\|\psi_i\| = T$, where T is the number of data points, to make the embedding coordinates invariant to the size of the data set. Still, the computed embedding eigenvectors, even for two identical data sets, may differ by a sign. Reconciling the signs for the embeddings of different data sets can be rationally done in several ways and is somewhat problem-specific. For example, if the mean of the embedding is sufficiently far from 0, we can require $\langle\psi_i\rangle > 0$; alternatively, if there is a common region sampled by both data sets, the sign of each eigenvector can be chosen to optimize the consistency of the embeddings of the common region data. We will return to the issue of embedding consistency for different data sets in our concluding discussion; for the moment, we will assume that our different sets sample the same region of data space in a representative enough way such that the correspondence between the sequences of retained eigenvectors for different embeddings is obvious.

D. Limiting diffusion operator

Assume that the dynamics of the underlying variables in (2) can be rewritten as

$$d\mathbf{x}(t) = -\nabla U(\mathbf{x}(t))dt + d\mathbf{w}(t), \tag{8}$$

where the drift term is expressed as a gradient of a potential U and $\mathbf{w}(t) \in \mathbb{R}^d$ is a vector consisting of independent Brownian motions. Nadler *et al.*[23] showed that in the limit $\varepsilon \to 0$ and $T \to \infty$, the kernel $\mathbf{W}^{\text{norm}}$ converges to the continuous backward Fokker-Planck operator

$$\mathcal{L} = \Delta - \nabla U \cdot \nabla. \tag{9}$$

We remark that, while in diffusion maps,[17] the limiting operator is the Laplacian on the manifold on which the observations lie, here the Laplacian operator on the data that is based on the Mahalanobis distance converges to a limiting Laplacian operator on the manifold of the intrinsic variables $\mathbf{x}$.

The convergence to the limiting operator implies that the eigenvectors of $\mathbf{W}^{\text{norm}}$, which are used to represent the NIV, approximate the eigenfunctions of the limiting Fokker-Planck operator on the NIV manifold. As such, according to (9), they depend on the potential. Thus, the constructed embedding is consistent as long as the subsets of observations $\mathbf{Y}(t)$ are adequate samples of the same (equilibrium) density. For example, when looking at multiple data sets from a molecular simulation, the embeddings will be consistent provided each simulation sufficiently samples the same underling free energy surface.

In order to merge observations from different regions, the fundamental question is, to what extent the respective sampling densities affect the coordinates we obtain. In principle, we can decouple diffusion dynamics (and thus, equilibrium densities) from geometry by normalizing the diffusion kernel with the sample density. This has been explored for diffusion maps independently of NIV.[17] When the normalized kernel is defined as

$$\widetilde{\mathbf{W}}^{\text{norm}} = \mathbf{D}^{-1}\mathbf{W}\mathbf{D}^{-1}, \tag{10}$$

the limiting diffusion operator is the Laplace-Beltrami operator,[23] given by $\widetilde{\mathcal{L}} = \Delta$. In other words, this diffusion captures only the geometry of the observations and is invariant to the potential (or, alternatively, to the details of the sampling density). This kernel is similar to the kernel in (7), but $\mathbf{D}$ appears with coefficient -1 rather than $-1/2$.

Since local coordinates are based on the noise term in (8), sets of measurements from different regions will essentially detect the same system of coordinates (the same variable "axes"); yet different sampling densities (arising, for example, from the drift term in (8)) will result in different *scalings* of these axes. Applying the Laplace-Beltrami normalization in the NIV context will thus eliminate sampling density effects. An alternative approach would find the scaling transformations that optimally "register" data sampled from common regions (in the spirit of histogram reweighting[24]).

III. LAPLACIAN PYRAMIDS FOR DATA EXTENSION

In this work, we are not only interested in extracting the underlying variables $\mathbf{x}$ from some (partial) observations $\mathbf{Y}$, but also interested in extending high-dimensional functions on a set of points which lie in a low-dimensional space. More specifically, viewing the ambient space coordinates $\mathbf{Y}$ as functions on the low-dimensional data $\mathbf{x} \in \mathbb{R}^d$, we want to estimate $\mathbf{Y}$ for new points $\mathbf{x}$. Laplacian Pyramids (LP) is a multiscale algorithm for extending an empirical function $\mathbf{f}$ defined on a set of points to new points not in the data set. The algorithm uses Laplacian kernels of decreasing widths to create multiscale representations of $\mathbf{f}$; these representations can be easily extended to new data points. This type of multiscale representation was introduced by Burt and Adelson[25] for image coding, and was later shown to be a tight frame by Do and Veterli.[26] Recently, LP was used to extend nonlinear embedding coordinates to new high-dimensional data points.[13] We will first review the LP algorithm for approximating and extending a one-dimensional function, and then describe the application of LP in extending high-dimensional functions.

Let $\mathbf{f} : \Gamma \to \mathbb{R}^n$ be a function that is known on a subset of points $S \subset \Gamma$. A coarse representation of $\mathbf{f}$ is generated using a coarse smoothing operator P_0. The smoothing operator P_0

is a normalized, coarse Laplacian kernel, defined by

$$p_0(i,j) = s_0^{-1}(i) w_0(i,j),\ i,j \in \Gamma, \tag{11}$$

where $w_0(i,j) = e^{-d^2(i,j)/\sigma_0}$ and $s_0(i) = \sum_{j\in S} w_0(i,j)$ is the normalizing term. The pairwise distance $d(i,j)$ is typically the Euclidean distance, and the parameter σ_0 is set to be large compared to the values of $d^2(i,j)$. The application of P_0 to $\mathbf{f}$ yields a coarse representation of the function, which we denote by $\mathbf{f}_0 = P_0(\mathbf{f})$.

The difference $\delta_1 = \mathbf{f} - P_0(\mathbf{f})$ is the input for the next iteration of the algorithm, which uses the smoothing operator P_1, $P_1 \propto e^{-d^2(i,j)^2/2^{-1}\sigma_0}$, to construct a coarse representation of δ_1. The obtained representation of δ_1, $P_1(\delta_1)$ together with the result of the previous iteration $\mathbf{f}_0 = P_0(\mathbf{f})$ yields a new, finer representation of $\mathbf{f}$, $\mathbf{f}_1 = P_0(\mathbf{f}) + P_1(\delta_1)$. In an iterative manner, multiscale representations of the function $\mathbf{f}$, denoted $\mathbf{f}_l$, are constructed.

$$\begin{aligned} &\text{scale } 0\text{: } \mathbf{f}_0 = P_0(\mathbf{f}),\\ &\text{scale } 1\text{: } \mathbf{f}_1 = P_0(\mathbf{f}) + P_1(\delta_1),\\ &\quad\vdots \quad\ \vdots\\ &\text{scale } l\text{: } \mathbf{f}_l = P_0(\mathbf{f}) + \textstyle\sum_{k=1}^{l} P_k(\delta_k). \end{aligned} \tag{12}$$

As l increases, the approximation becomes more refined because $P_l \propto e^{-d^2(i,j)/2^{-l}\sigma_0}$ uses a Laplacian kernel of a finer width. The iterations stop when the difference between $\mathbf{f}$ and $\mathbf{f}_l$ is smaller than a pre-defined error threshold.

The representations $\mathbf{f}_0, \mathbf{f}_1, \ldots, \mathbf{f}_l$ can be extended to a new point $\tilde{i} \in \Gamma\backslash S$ by extending the operators P_0, P_1, …, P_l. For example, $\mathbf{f}_0(\tilde{i}) = \sum_{i\in S} p_0(\tilde{i}, i)\mathbf{f}(i)$ and $\mathbf{f}_1(\tilde{i}) = \mathbf{f}_0(\tilde{i}) + \sum_{(i\in S)} p_1(\tilde{i}, i)\delta_1(i)$.

Figure 2 displays an illustrative example of the algorithm, when applied to the function

$$\mathbf{f}(x) = \begin{cases} -0.02(x-4\pi)^2 + \sin(x), & 0 \le x \le 4\pi,\\ -0.02(x-4\pi)^2 + \sin(x) + 0.5\sin(3x), & 4\pi < x \le 7.5\pi,\\ -0.02(x-4\pi)^2 + \sin(x) + 0.5\sin(3x) + 0.25\sin(9x), & 7.5\pi < x \le 10\pi, \end{cases} \tag{13}$$

that contains several scales. The coarse regions of the function ($0 \le x \le 4\pi$) are well approximated by a small number of scales. As the function becomes more oscillatory ($\pi \le x \le 7.5\pi$ and $7.5\pi \le x \le 10\pi$), a finer representation, and a larger number of scales l, is required to capture its behavior.

In this work, LP is applied to extend a high-dimensional function $\mathbf{f} : \mathbb{R}^d \to \mathcal{M}$, which maps a set of points in the NIV space to their values $\mathbf{Y}(i)$ in the observable space. Let $\Psi(i) = (\psi_1(i), \psi_2(i), \ldots, \psi_d(i))$ be the set of NIV that were constructed from the data samples $\mathbf{Y}(i)$, as described in Algorithm I. The values of function $\mathbf{f} : \mathbb{R}^d \to \mathcal{M}$ are known on the subset $S = \{\Psi(i)\}$, with $\mathbf{f}(\Psi(i)) = \mathbf{Y}(i)$.

A naïve way to extend $\mathbf{f}$ to a new data point $\Psi(\tilde{i})$ is to find the point's nearest neighbors in NIV space and average their function values. A different, point-wise adaptive approach is described by Buchman *et al.*:[27] high-dimensional hurricane tracks were estimated from low dimensional embedding coordinates using a weighted average of the points close to $\Psi(\tilde{i})$ in the embedded space. However, this point-wise adaptation requires setting the nearest neighborhood radius parameter for every point. The LP algorithm finds the appropriate nearest neighborhood radius for each new point $\Psi(\tilde{i})$. This radius will be large in smooth regions of the function, and small in regions in which $\mathbf{f}$ contains higher frequency components. The LP approximation of a new, high dimensional point $\mathbf{Y}(\tilde{i})$ is calculated by a weighted average of the function values that belong to the neighboring points. The weights are based on the pairwise distances in the intrinsic, low-dimensional space. In practice, a set of smoothing operators $P_0, P_1, \ldots, P_l$, with

$$P_l \propto e^{-d^2(\Psi(i),\Psi(j))/2^{-l}\sigma_0}, \tag{14}$$

are constructed and later extended to create the multiscale approximations as defined in (12). The LP algorithm for the inverse mapping is summarized in Algorithm II.

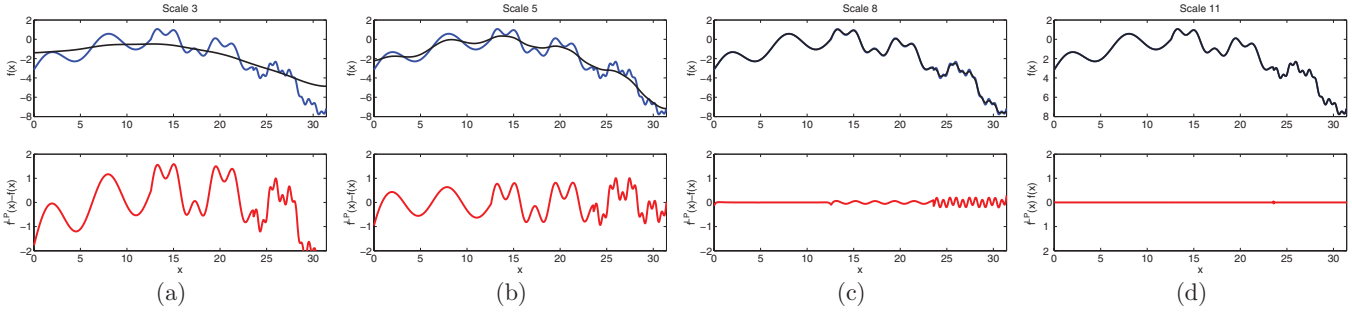

FIG. 2. Approximation of the function $\mathbf{f}(x)$ defined in (13) using Laplacian Pyramids for "scales" (a) 3, (b) 5, (c) 8, and (d) 11. (Top) The true function in blue and the LP approximation in black. (Bottom) The residual error in the LP approximation.

ALGORITHM II. Laplacian Pyramids for inverse mapping.

1. Construct a set of smoothing operators $P_0, P_1, \ldots, P_l$ based on intrinsic pairwise distances $d(\Psi(i), \Psi(j))$ (where d is typically the Euclidean distance).
2. Use the smoothing operators to obtain a multiscale representation (see (14)) of $\mathbf{f} : \Psi(i) \to Y(i)$.
3. Given a new point $\Psi(\tilde{i})$ in NIV, extend the smoothing operators $P_0, P_1, \ldots, P_l$ by $p_l(\Psi(\tilde{i}), \Psi(i)) = s_l^{-1}(\Psi(i)) w_l(\Psi(\tilde{i}), \Psi(i))$.
4. Use the extended smoothing operators to approximate the value of $Y(\tilde{i})$ as $Y(\tilde{i}) \approx \mathbf{f}_l(\Psi(\tilde{i})) = \mathbf{f}_0(\Psi(\tilde{i})) + \sum_k \sum_{(i \in S)} p_k(\Psi(\tilde{i}), \Psi(i)) \delta_k(\Psi(i))$.

IV. MODELS AND RESULTS

A. A chemical reaction network

We first consider a chemical reaction network involving multiple enzyme-substrate interactions.[15] The reaction steps that comprise the network are

$$\begin{aligned} E + S &\underset{e_{-1}}{\overset{e_1}{\rightleftharpoons}} E:S \overset{e_2}{\to} E + S^*, \\ S + E &\underset{b_{-1}}{\overset{b_1}{\rightleftharpoons}} S:E \overset{b_2}{\to} S + E^*, \\ D + S^* &\underset{d_{-1}}{\overset{d_1}{\rightleftharpoons}} D:S^* \overset{d_2}{\to} D + S, \\ F + E^* &\underset{f_{-1}}{\overset{f_1}{\rightleftharpoons}} F:E^* \overset{f_2}{\to} F + E. \end{aligned} \tag{15}$$

The "*" denotes an activated form of a species, and the ":" denotes a complex formed between two species; the complexes $E : S$ and $S : E$ are not equivalent. There are 10 species in this reaction system. However, one can write four conservation equations (since total E, S, D, and F are all conserved) to reduce the system to 6 dimensions (which we order as S, E, $E : S$, $S : E$, $D : S^*$, $F : E^*$). We consider a parameter regime in which the ODE approximation of this scheme exhibits a separation of time scales, so that initial conditions quickly approach a two-dimensional manifold. Details about the specific parameter values can be found in the Appendix.

Although the dynamics of chemical reaction networks are typically described by a system of ODEs, the ODEs are only an approximation that holds in the limit of a large number of molecules. When the number of molecules is small, the system is inherently stochastic and its dynamics can be simulated using the Gillespie SSA;[14] at intermediate molecule counts, the chemical Langevin approximation[28] becomes useful. We can control the level of noise in our simulation by adjusting the volume V, and therefore, adjusting the number of molecules, in the system. We take the volume small enough so that we can still observe appreciable stochasticity in small simulation bursts, but large enough (in our simulations, we take $V = 10^5$) so that the underlying two-dimensional manifold is (relatively) smooth.

We generate 3000 initial conditions $\mathbf{Y}_0(1), \ldots, \mathbf{Y}_0(3000) \in \mathbb{R}^6$ uniformly at random from the region of state space where all concentrations (respecting conservation laws) are non-negative. We evolve each point $\mathbf{Y}_0(t)$ forward for 10 time units using the SSA to obtain a point $\mathbf{Y}(t) \in \mathbb{R}^6$; according to the time scales calculated from the linearized ODEs, 10 time units is sufficiently long for the initial points in the ODE system to converge to the two-dimensional manifold, but not long enough for the points to converge to a one-dimensional curve or to the final steady state (see the Appendix for more details). In our stochastic simulations, the initial points appear to converge to an approximate two-dimensional manifold (in expected value, see Figure 3). We consider $\mathcal{Y} = \{\mathbf{Y}(t) : t = 1, \ldots, 3000\}$ to be representative points "on" this apparent two-dimensional manifold. From each manifold point $\mathbf{y} \in \mathcal{Y}$, we run 20 short simulation "bursts," each for 0.2 time units. We denote the endpoints from the short simulations as $\mathcal{Y}^{burst}(\mathbf{y})$.

We consider two different data sets from our simulations. Data set 1, denoted $\mathcal{Y}_1$, consists of $\mathbf{Y}(1), \ldots, \mathbf{Y}(2000)$, restricted to components S, E, $S : E$, and $F : E^*$, i.e.,

$$\mathcal{Y}_1 = \left\{ \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \mathbf{Y}(t) \in \mathbb{R}^4 : t = 1, \ldots, 2000 \right\}.$$

Data set 2, denoted $\mathcal{Y}_2$, consists of $\mathbf{Y}(1500), \ldots, \mathbf{Y}(3000)$, restricted to components S, E, $E : S$, and $D : S^*$, i.e.,

$$\mathcal{Y}_2 = \left\{ \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \mathbf{Y}(t) \in \mathbb{R}^4 : t = 1500, \ldots, 3000 \right\}.$$

The endpoints of the simulation bursts for the two data sets, $\mathcal{Y}_1^{burst}(\mathbf{y})$ and $\mathcal{Y}_2^{burst}(\mathbf{y})$, are defined analogously. We then estimate the covariances for each point in each data set as

$$\widehat{\mathbf{C}}_i(\mathbf{y}) = \sum_{\mathbf{z} \in \mathcal{Y}_i^{burst}(\mathbf{y})} (\mathbf{z} - \hat{\mu}_i(\mathbf{y}))(\mathbf{z} - \hat{\mu}_i(\mathbf{y}))^T, \mathbf{y} \in \mathcal{Y}_i, i \in \{1, 2\}, \tag{16}$$

where $\hat{\mu}_i(\mathbf{y})$ is the empirical mean of $\mathcal{Y}_i^{burst}(\mathbf{y})$.

We first demonstrate that NIV produces the same embeddings for $\mathcal{Y}_1$ and $\mathcal{Y}_2$, even though the two data sets contain information of different chemical species. Figure 4 shows the two-dimensional NIV embeddings for the two different data sets; the embeddings appear visually consistent. We also note that both $\mathcal{Y}_1$ and $\mathcal{Y}_2$ contain points that are projections of $\mathbf{Y}(1500), \ldots, \mathbf{Y}(2000)$. We therefore compute the correlation between the embedding coordinates for these points common to $\mathcal{Y}_1$ and $\mathcal{Y}_2$. We obtain a correlation of 0.97 and 0.95 for the first and second NIV, respectively, indicating that the two embeddings are in quantitative agreement with each other. We would like to note that both $\mathcal{Y}_1$ and $\mathcal{Y}_2$ are sufficiently high-dimensional ("rich enough") to allow us to recover the common underlying two-dimensional manifold.

To determine whether a sufficient amount of data is available to obtain an accurate embedding, we compute the NIV embeddings for different numbers of data points (arising from different numbers of trajectories initialized randomly in the full concentration space). Figure 5 shows

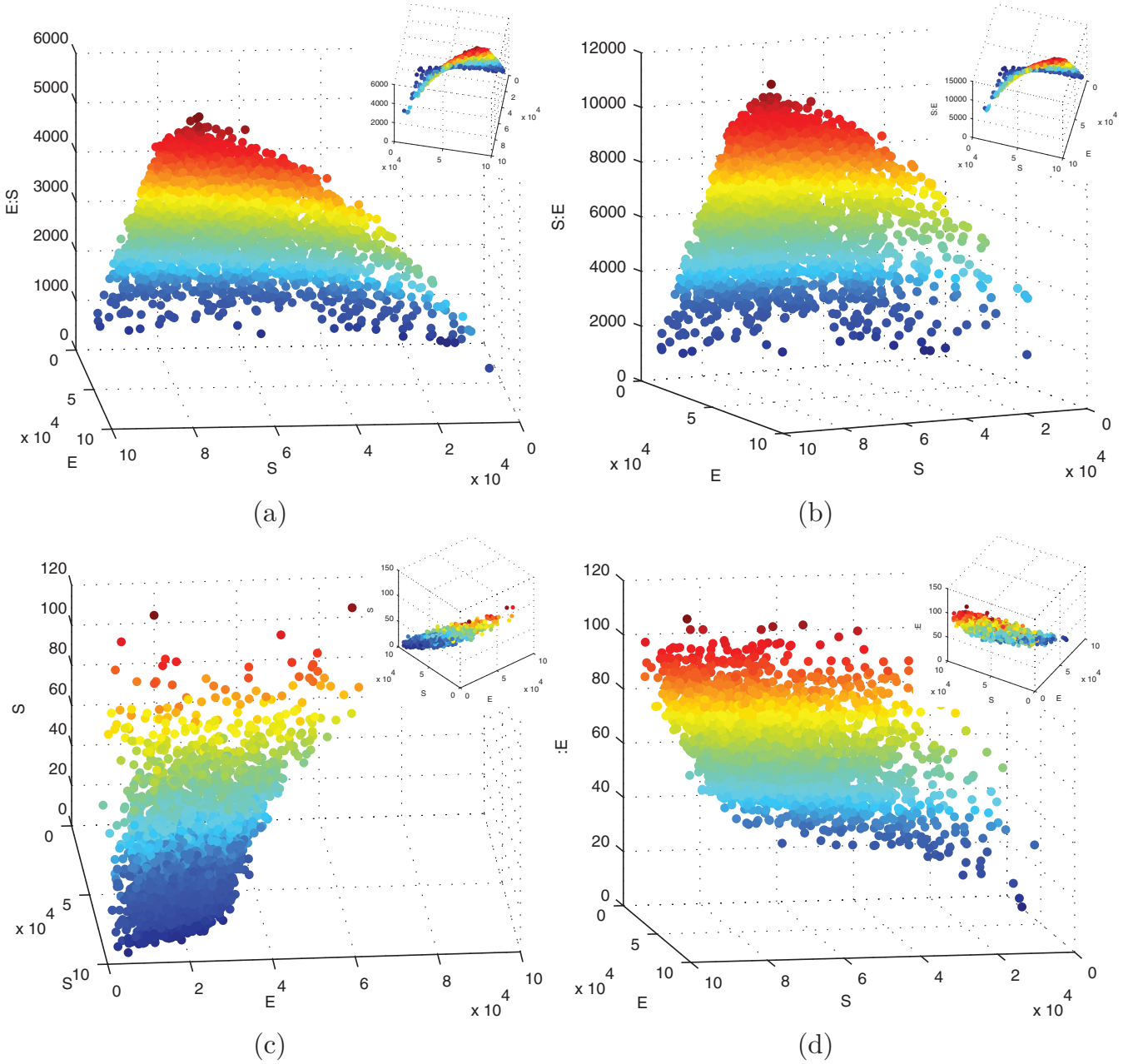

FIG. 3. (a)–(d) Projections of the data obtained from stochastic simulation of the chemical reaction network described in Sec. IV A. The insets show rotations of the projections to illustrate the approximate two-dimensionality of the "slow manifold."

the two-dimensional NIV embeddings and corresponding eigenvalue spectra of the kernel for the chemical reaction network example computed from 10, 100, 500, and 1500 data points. We observe that the eigenvalue spectra appear converged for 100 data points and above. This implies that a sufficient amount of data is available merely to discover the intrinsic variables. However, we observe that 100 points are insufficient to construct a self-consistent embedding *in the two variables discovered*. In order to account for the different scaling of the NIV coordinate axes, we require at least 500 data points. This last convergence is exemplified in Figures 5(c) and 5(d). The convergence of the spectrum of the kernel and the ultimate self-consistency of the NIV embedding is our empirical indicator that a sufficient amount of data is available.

We then use NIV together with Laplacian Pyramids to estimate the values of $S : E$ and $F : E^*$ for $\mathcal{Y}_2$. Because $\mathcal{Y}_1$ and $\mathcal{Y}_2$ are measured for different components, there is no simple way to estimate $S : E$ and $F : E^*$ directly in the observation space. Instead, we must first embed the data into the NIV space so that we can compute neighbors *between the two data sets*. We use $\mathcal{Y}_1$ to train an LP function from the two-dimensional NIV embedding to $S : E$ and $F : E^*$. We then use this function to predict the values of $S : E$ and $F : E^*$ for $\mathcal{Y}_2$, using the computed NIV embedding for $\mathcal{Y}_2$. In this way, we are exploiting the fact that the NIV embedding is intrinsic and consistent

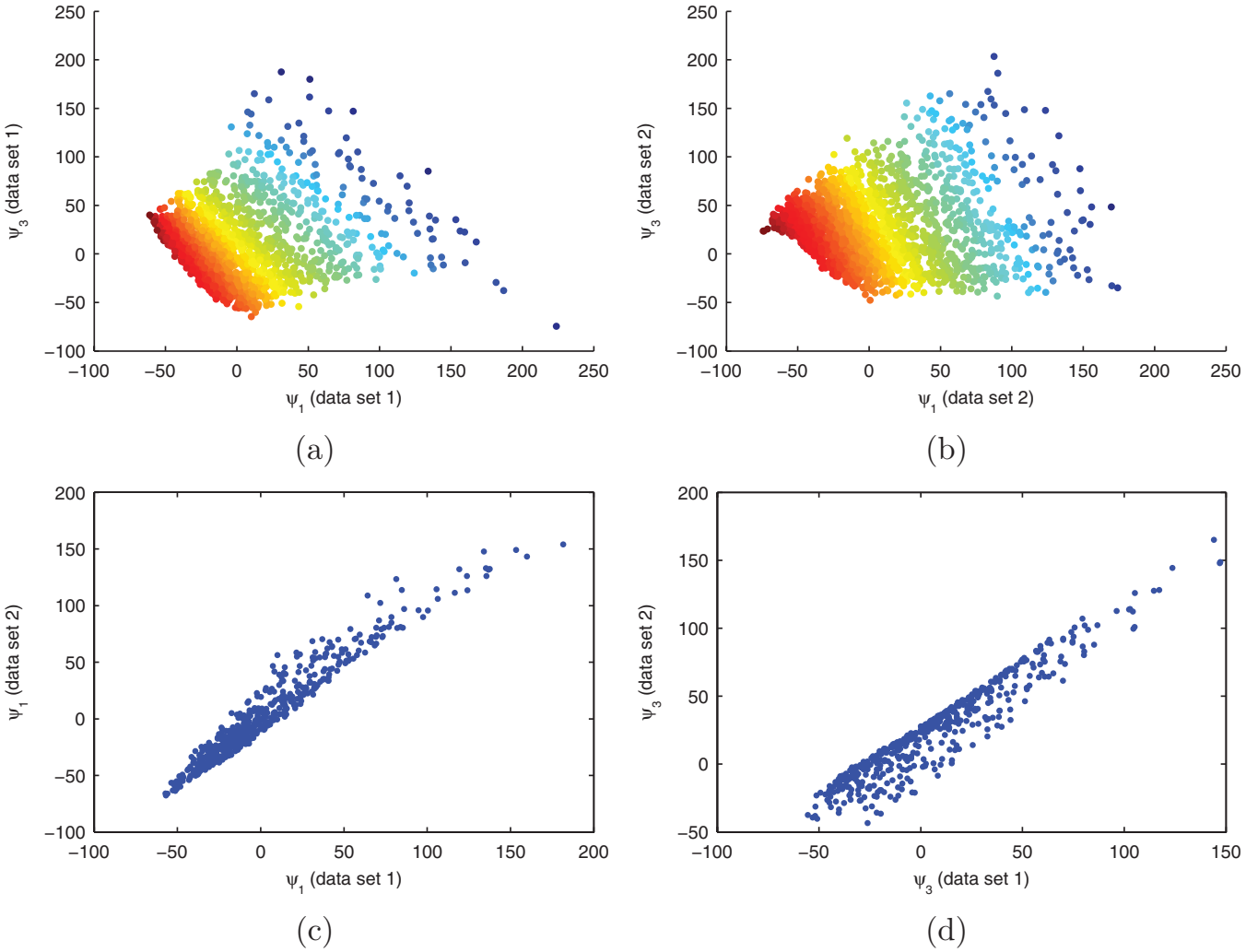

FIG. 4. (a) NIV embedding obtained from $\mathcal{Y}_1$ (observations of components E, S, S:E, and F:E*), colored by S. (b) NIV embedding obtained from $\mathcal{Y}_2$ (observations of components E, S, E:S, and D:S*), colored by S. Visually, we can see that the embeddings obtained from $\mathcal{Y}_1$ and $\mathcal{Y}_2$ are consistent, even though the two data sets consist of observations of different chemical species. (c) Correlation of first NIV between two different embeddings (correlation = 0.97). (d) Correlation of second NIV between two different embeddings (correlation = 0.95). We obtain a good quantitative agreement between the embedding coordinates for the two data sets.

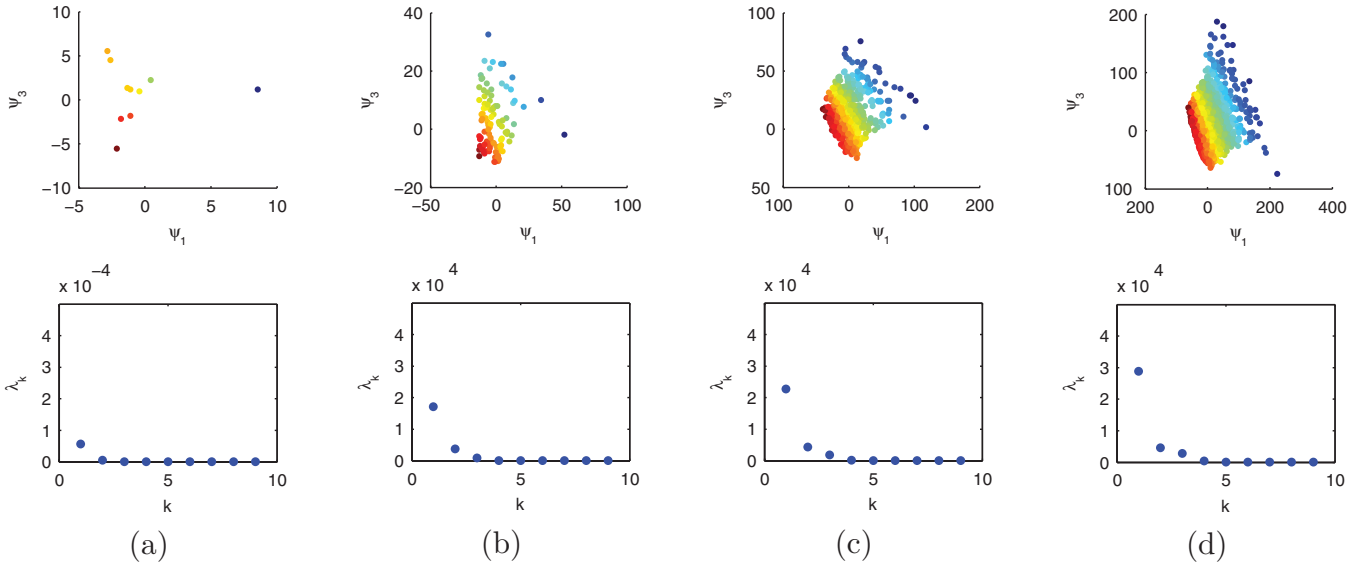

FIG. 5. Two-dimensional NIV embeddings (top) and corresponding eigenvalue spectra (bottom) calculated from subsets of $\mathcal{Y}_1$ containing (a) 10 data points, (b) 100 data points, (c) 500 data points, and (d) 1500 data points. One can see that the embedding and spectrum in (a) is inconsistent with the other embeddings, and that, although the spectrum in (b) appears converged, the embedding is not converged until (c).

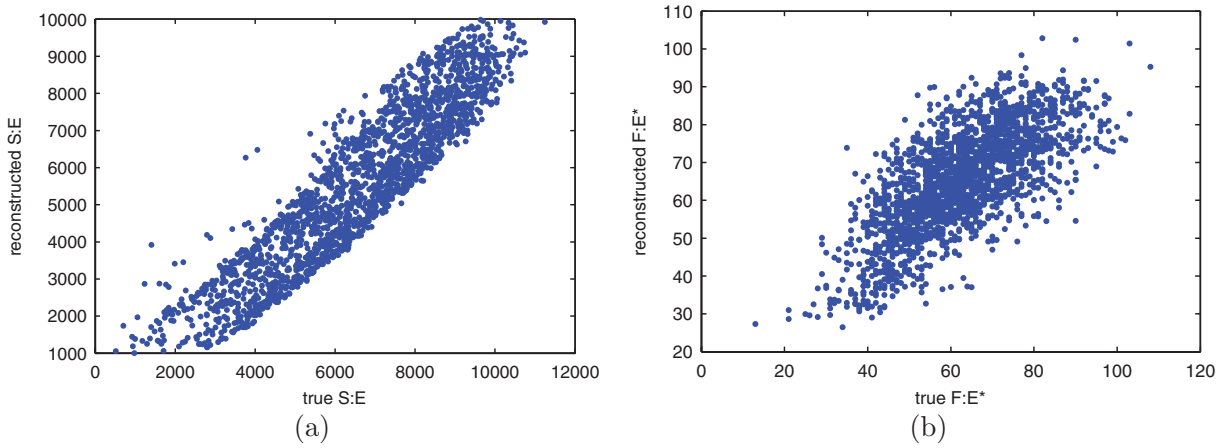

FIG. 6. LP reconstructions of (a) $S : E$ and (b) $F : E^*$ for $\mathcal{Y}_2$, using $\mathcal{Y}_1$ as training data.

between the two data sets, even though the two data sets contain measurements of different chemical species.

The results of the LP prediction are shown in Figure 6. The normalized mean-squared errors between the true and estimated values for $S : E$ and $F : E^*$, defined as $\frac{\langle (y_{true} - y_{pred})^2 \rangle}{\langle y_{true}^2 \rangle}$, are 0.0372 and 0.0287 , respectively. Therefore, we can effectively estimate the unobserved components in the reaction network using NIV together with LP.

B. Alanine dipeptide

Our second example comes from the molecular dynamics simulation of a small peptide fragment. Alanine dipeptide (Ala2) is often used as a "prototypical" protein caricature for simulation studies.[16,29–33] We simulate the motion of Ala2 in explicit solvent using the AMBER 10 molecular simulation package[34] with an optimized version[35] of the AMBER ff03 force field.[36] The molecule is solvated with 638 TIP3P water molecules[37] with periodic boundary conditions, and the particle mesh Ewald method is used for long-range electrostatic interactions.[38] The simulation is performed at constant volume and temperature (NVT ensemble), with the temperature being maintained at 300 K with a Langevin thermostat.[39] Hydrogen bond lengths are fixed using the SHAKE algorithm.[40] The two dihedral angles ϕ and ψ are known to parameterize the free energy surface, which contains three important minima (labeled A, B, and C, see Figure 7). Our simulations are concentrated around minimum B in the free energy surface, located at $\phi \approx -65^\circ$, $\psi \approx 150^\circ$. We choose to sample only around minimum B, as each data set must sample the same region of the potential energy surface to obtain consistent NIV, as described in Sec. II D. We start many simulations at 10° away from the minimum, and allow the simulations to each

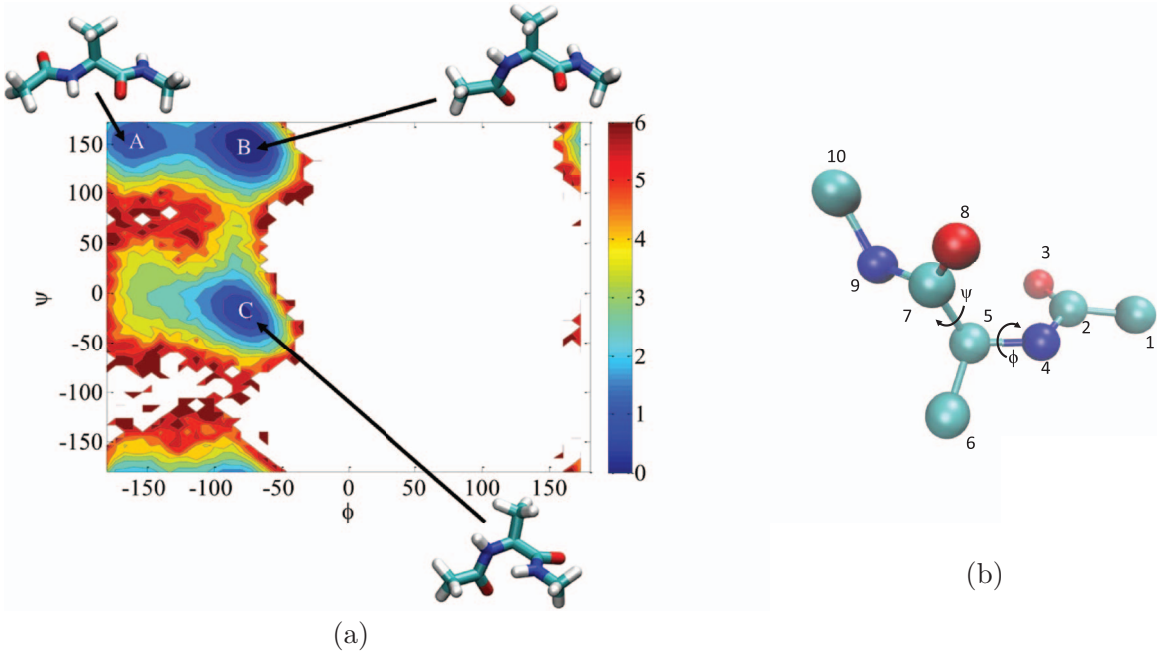

FIG. 7. (a) Free energy surface for Ala2. The relevant minima are labeled A, B, and C, and the corresponding molecular configurations are shown. (b) Sample representative molecular structure of Ala2, excluding the hydrogens. The atoms are numbered and the two dihedral angles ϕ and ψ are indicated.

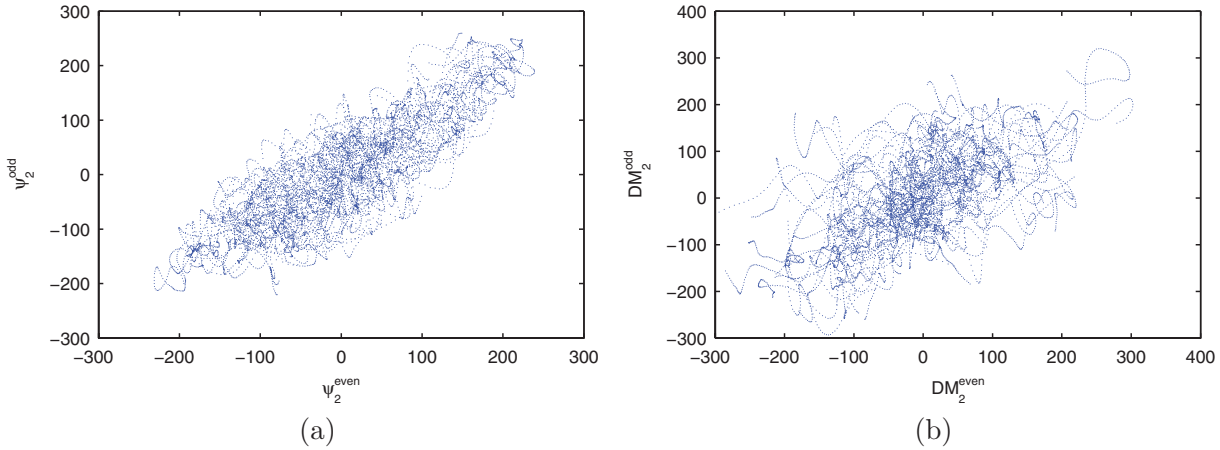

FIG. 8. (a) Correlation between the second NIV computed using the atoms 2, 4, 6, 8, and 10 (ψ_2^{even}) and the second NIV computed using atoms 5, 7, and 9 (ψ_2^{odd}). (b) Correlation between the second DM computed using the atoms 2, 4, 6, 8, and 10 (DM_2^{even}) and the second DM computed using atoms 5, 7, and 9 (DM_2^{odd}). The correlations for the first (not shown) and second NIV coordinates are found to be 0.62 and 0.84, respectively. The correlations for the first (not shown) and second DM coordinates are found to be 0.54 and 0.60, respectively.

run for 0.1 ps, while recording the configuration of Ala2 every 1 fs (therefore, each trajectory is 100 points long). Configurations are recorded with all atoms except the hydrogens.

We first compare NIV with diffusion maps,[17] an established nonlinear dimensionality reduction technique. We consider 10 000 data points from our simulation $\mathbf{Y}(1), \ldots, \mathbf{Y}(10\,000) \in \mathbb{R}^{30}$; every 100 data points comes from a continuous simulation trajectory. We construct two data sets: $\mathcal{Y}_{even} = \{\mathbf{Y}(t)$ restricted to atoms 2, 4, 6, 8, and $10 : t = 1, \ldots, 10\,000\}$, and $\mathcal{Y}_{odd} = \{\mathbf{Y}(t)$, restricted to atoms 5, 7, and $9 : t = 1, \ldots, 10\,000\}$ (see Figure 7 for the atom indexing). We then compute the NIV and diffusion maps embeddings for $\mathcal{Y}_{even}$ and $\mathcal{Y}_{odd}$; for NIV, we compute the covariances as in (3), with $L = 10$.

The correlation between the NIV coordinates for the two data sets and the diffusion map (DM) coordinates for the two data sets are shown in Figure 8. The correlation between the two NIV embeddings is higher than the correlation between the two diffusion map embeddings. Therefore, it appears advantageous to use NIV over diffusion maps if one wishes to obtain a consistent embedding and merge data sets from different observation domains (as long as the two main assumptions underpinning the NIV algorithm hold).

We then use NIV together with LP to predict the conformation of Ala2 when we only observe some of the atoms. We have 20 000 data points $\mathbf{Y}(1), \ldots, \mathbf{Y}(20\,000)$, where every 100 data points come from one continuous simulation trajectory. Our first data set (which will serve as our training data for LP), $\mathcal{Y}_{all}$, consists of the first 10 000 data points ($\mathcal{Y}_{all} = \{\mathbf{Y}(t) : t = 1, \ldots, 10\,000\}$). Our second data set (which will serve as our test data), $\mathcal{Y}_{odd}$, consists of the last 12 000 data points restricted to *only* the odd atoms ($\mathcal{Y}_{odd} = \{\mathbf{Y}(t)$ restricted to the odd atoms : $t = 8001, \ldots, 20\,000\}$). We compute the covariances as in (3) with $L = 15$. We compute the NIV embedding for the training data $\mathcal{Y}_{all}$ and the test data $\mathcal{Y}_{odd}$; we then use LP interpolation from the training data to predict the location of all the atoms for each point in the test data.

The NIV embedding for the training data $\mathcal{Y}_{all}$ is shown in Figure 9. The embedding is three-dimensional, and visual inspection reveals that each coordinate can be directly linked with one physical variable: the first coordinate

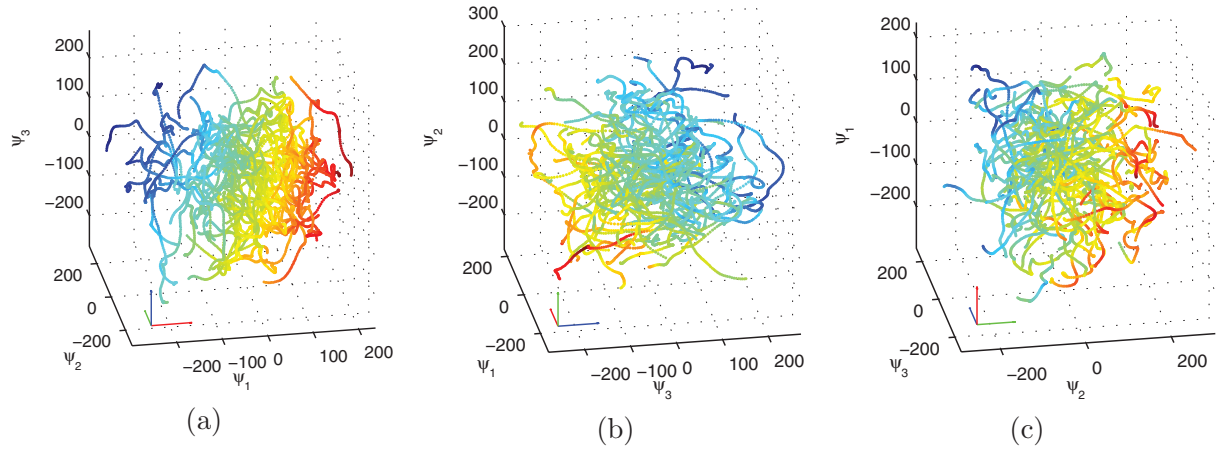

FIG. 9. The 3-dimensional NIV embedding for Ala2 computed using $\mathcal{Y}_{all}$, colored by (a) the y-coordinate of the first atom, (b) the dihedral angle ϕ, and (c) the dihedral angle ψ. Each embedding is rotated so that the correlation between the colors and the relevant NIV can easily be seen.

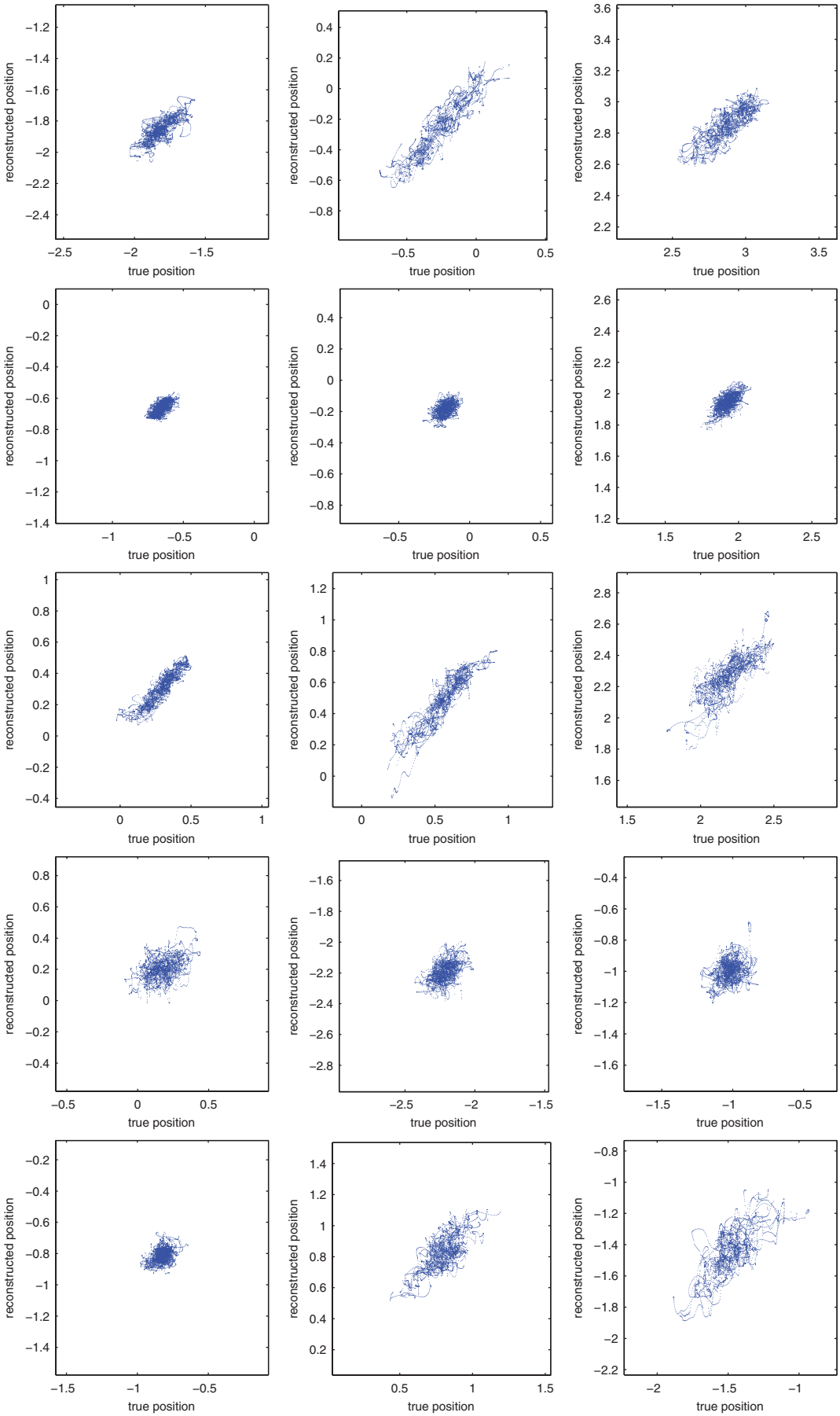

FIG. 10. The correlation between the true position and the reconstructed position (using LP) for the test data. The columns correspond the x-, y-, and z-coordinates, and the rows correspond to atoms 1, 2, 3, 6, and 8.

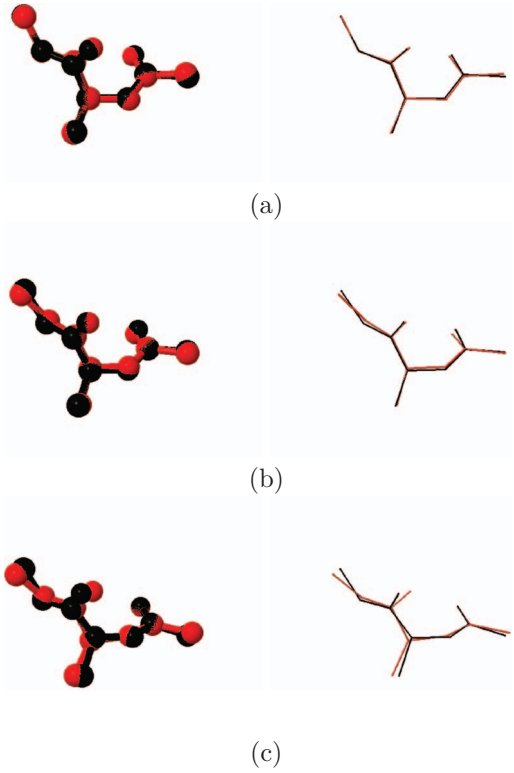

FIG. 11. True structure (black) and reconstructed structure (red) for three different data points. Each data point is shown in "ball-and-stick" representation (left) and "wireframe" representation (right) so that the discrepancies can easily be seen between the different configurations.

describes the flipping of atoms 1 and 3, the second coordinate describes the dihedral angle ϕ, and the third coordinate describes the dihedral angle ψ. We calculate the correlation between the embedding coordinates for the points in $\mathcal{Y}_{all}$ and $\mathcal{Y}_{odd}$ that come from the common simulation data points $\mathbf{Y}(8001), \ldots, \mathbf{Y}(10\,000)$. The embeddings for the two data sets are found to be fairly consistent, with correlations of 0.97, 0.72, 0.85 for the first, second, and third NIV, respectively.

Figure 10 shows the reconstructed position from partial observation versus true position for certain selected atoms. The strong correlation between the true and reconstructed positions is easier to appreciate for atoms that move substantially within the data set (such as atoms 1 and 3). Figure 11 shows molecular structures for the true and reconstructed configurations for selected data points; there is qualitative agreement between the true and reconstructed configurations. The discrepancies between the true and reconstructed positions of atoms 9 and 10 (see Figure 7 for atom indexing) for some of the molecular structures are not surprising. From Figure 9, one can see that the first three NIV describe the flipping of atoms 1 and 3 and the dihedral angles ϕ and ψ. However, atoms 9 and 10 do not participate in any of these physical quantities, and so there is little information about their positions contained in the first three NIV. We suspect that using more NIV would result in a better description of atoms 9 and 10 and lead to more accurate structure reconstructions.

For a brief comparison of LP over other reconstruction techniques, we also reconstruct configurations from the

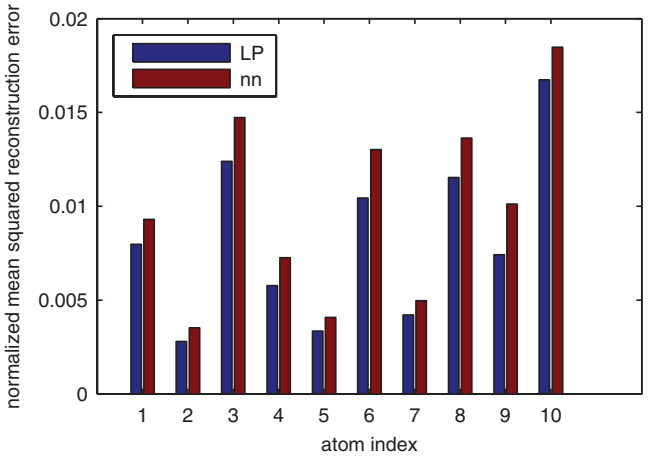

FIG. 12. Mean squared error of reconstructed position, normalized by the average bond length, for each atom in Ala2. The positions were reconstructed using both LP and nearest neighbor (nn) interpolation.

NIV components using simple nearest-neighbor interpolation. The average reconstruction error, scaled by the average bond length within the molecule, is shown in Figure 12; LP arguably outperforms simple nearest neighbor search for all of the atoms. To determine why LP outperforms nearest neighbor search, we examine the errors at different scales in the LP algorithm. The example described in Figure 2 demonstrates the appropriateness of the LP algorithm to signals with an intrinsic multiscale structure. We perform a similar analysis using the data from the alanine dipeptide example, and show that the observed signal does exhibit a similar error structure, with large errors in some regions of NIV space and small errors in other regions; these results are shown in Figure 13. We therefore conclude that the alanine dipeptide data do contain

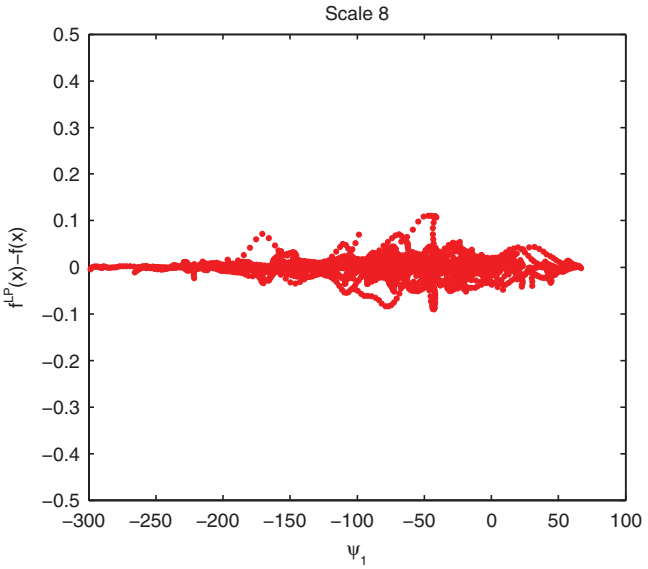

FIG. 13. The error in the reconstructed position of the z-coordinate of atom 10 as a function of ψ_1. One can see that the error is large when ψ_1 is large (similar to the error structure in Figure 2), demonstrating that our data is indeed multiscale and that Laplacian Pyramids is an appropriate interpolation algorithm to use.

a measure of multiscale behavior, and, therefore, that LP is an appropriate algorithm for reconstruction.

V. CONCLUSIONS

We have used Nonlinear Intrinsic Variables to analyze two complex atomistic simulations: a stochastic simulation of a chemical reaction network and a molecular dynamics simulation of alanine dipeptide. In both examples, we were able to uncover the intrinsic variables governing the underlying stochastic process, which are independent of the particular measurement or observation of the system (under the conditions mentioned). The uniqueness of the embedding coordinates allowed us to compare and merge data sets from different measurement functions, and therefore allowed us to use an interpolation/extension scheme (here Laplacian Pyramids) to complete partial observations. Different interpolation techniques (e.g., kriging,[41,42] geometric harmonics,[43] versions of the Nyström extension) can be and should be explored, since the performance of such techniques may well be problem dependent, especially for multiscale, complex simulation data.

There are many open questions leading to interesting research directions to be explored. In this work, we considered data sets that consist of different partial observations, but in which each data set samples the entire underlying manifold in what we loosely referred to as a "representative enough" way. However, NIV could also be used to merge data sets when each data set samples only a portion of the manifold, provided there is enough overlap to "register" the embeddings. Merging data sets that come from different portions of the manifold would not only require scaling the embedding coordinates, but also shifting and possibly permuting the embedding coordinates (in this spirit, see the discussion in Lafon *et al.*[12]). The ability to merge data from different regions would then allow us to analyze systems where complete sampling is computationally intractable, such as molecular systems with several high energy barriers separating regions of state space.

Other issues, such as accurately estimating the covariance matrices required for the computation of the Mahalanobis distance, are also of current research interest. It is clearly necessary to link this type of calculation with modern estimation techniques for (multiscale) diffusions[44–46] to test the appropriateness of the window sampling lengths selected; this will determine the accuracy of the noise covariance estimation by eliminating the bias due to drift variations. We are confident that the exploration of these open questions will enable the use of our methodology in many interesting applications, such as merging data from molecular simulations at different levels of granularity, or merging simulation data with experimental observations.

ACKNOWLEDGMENTS

C.J.D. would like to acknowledge support from the US Department of Energy Computational Science Graduate Fellowship, Grant No. DE-FG02-97ER25308. I.G.K. would like to acknowledge support from the US Department of Energy, Grant Nos. DE-FG02-10ER26024 and DE-FG02-09ER25877.

APPENDIX: CHEMICAL REACTION NETWORK PARAMETERS

We consider the following network of chemical reactions:

$$\begin{aligned} E + S &\underset{e_{-1}}{\overset{e_1}{\rightleftharpoons}} E:S \xrightarrow{e_2} E + S^*, \\ S + E &\underset{b_{-1}}{\overset{b_1}{\rightleftharpoons}} S:E \xrightarrow{b_2} S + E^*, \\ D + S^* &\underset{d_{-1}}{\overset{d_1}{\rightleftharpoons}} D:S^* \xrightarrow{d_2} D + S, \\ F + E^* &\underset{f_{-1}}{\overset{f_1}{\rightleftharpoons}} F:E^* \xrightarrow{f_2} F + E. \end{aligned} \tag{A1}$$

In the limit of a large number of molecules, the dynamics of this network is governed by the following ODEs:

$$\begin{aligned} S' &= -e_1SE + e_{-1}E:S - b_1SE + b_{-1}S:E \\ &\quad + b_2S:E + d_2D:S^*, \\ E' &= -e_1SE + e_{-1}E:S + e_2E:S - b_1SE \\ &\quad + b_{-1}S:E + f_2F:E^*, \\ E:S' &= e_1SE - e_{-1}E:S - e_2E:S, \\ S:E' &= b_1SE - b_{-1}S:E - b_2S:E, \\ S^{*\prime} &= e_2E:S - d_1DS^* + d_{-1}D:S^*, \\ E^{*\prime} &= b_2S:E - f_1FE^* + d_{-1}F:E^*, \\ D' &= -d_1DS^* + d_{-1}D:S^* + d_2D:S^*, \\ F' &= -f_1FE^* + f_{-1}F:E^* + f_2F:E^*, \\ D:S^{*\prime} &= d_1DS^* - d_{-1}D:S^* - d_2D:S^*, \\ F:E^{*\prime} &= f_1FE^* - f_{-1}F:E^* - f_2F:E^*. \end{aligned} \tag{A2}$$

We can write four balance equations for the conservation of total S, E, D, and F.

$$\begin{aligned} S_T &= S^* + S + E:S + S:E + D:S^*, \\ E_T &= E^* + E + E:S + S:E + F:E^*, \\ D_T &= D + D:S^*, \\ F_T &= F + F:E^*. \end{aligned} \tag{A3}$$

We choose to eliminate S^*, E^*, D, and F from the system of ODEs. We therefore obtain a system of 6 ODEs,

$$\begin{aligned}
S' &= -e_1 SE + e_{-1} E : S - b_1 SE \\
&\quad + b_{-1} S : E + b_2 S : E + d_2 D : S^*, \\
E' &= -e_1 SE + e_{-1} E : S + e_2 E : S \\
&\quad - b_1 SE + b_{-1} S : E + f_2 F : E^*, \\
E : S' &= e_1 SE - e_{-1} E : S - e_2 E : S, \\
S : E' &= b_1 SE - b_{-1} S : E - b_2 S : E, \\
D : S^{*\prime} &= d_1 (D_T - D : S^*)(S_T - S - E : S - S : E \\
&\quad - D : S^*) - d_{-1} D : S^* - d_2 D : S^*, \\
F : E^{*\prime} &= f_1 (F_T - F : E^*)(E_T - E - E : S - S : E \\
&\quad - F : E^*) - f_{-1} F : E^* - f_2 F : E^*.
\end{aligned} \tag{A4}$$

Alternatively, we can write the rates for the 12 chemical reactions as

$$\begin{aligned}
r_1 &= e_1 SE, \\
r_2 &= e_{-1} E : S, \\
r_3 &= e_2 E : S, \\
r_4 &= b_1 SE, \\
r_5 &= b_{-1} S : E, \\
r_6 &= b_2 S : E, \\
r_7 &= d_1 (D_T - D : S^*)(S_T - S - E : S - S : E - D : S^*), \\
r_8 &= d_{-1} D : S^*, \\
r_9 &= d_2 D : S^*, \\
r_{10} &= f_1 (F_T - F : E^*)(E_T - E - E : S - S : E - F : E^*), \\
r_{11} &= f_{-1} F : E^*, \\
r_{12} &= f_2 F : E^*.
\end{aligned} \tag{A5}$$

For the Gillespie SSA, we use these rates to adjust the *number* of each molecule, depending on which reaction occurs. We take the volume of the reactor $V = 10^5$. We use the parameters $b_1 = 5/V$, $d_1 = 0.0009/V$, $e_1 = 0.1/V$, $f_1 = 0.1/V$, $b_{-1} = 10.6$, $d_{-1} = 0.05$, $e_{-1} = 0.5$, $f_{-1} = 0.01$, $b_2 = 0.4$, $d_2 = 0.85$, $e_2 = 0.05$, and $f_2 = 2$. We take $S_T = E_T = D_T = 1V$, and $F_T = 0.02V$, where S_T, E_T, D_T, and F_T are total number of S, E, D, and F, respectively. In this parameter regime, the relevant timescales around the steady state ($-1/\lambda_i$, where λ_i are the eigenvalues of the Hessian) are 1176, 9.731, 1.594, 1.111, 0.4975, 0.06498. Therefore, we choose to evolve forward for 10 time units to find points on a perceived two-dimensional manifold.

[1] J. B. Tenenbaum, V. de Silva, and J. C. Langford, Science **290**, 2319 (2000).
[2] S. T. Roweis and L. K. Saul, Science **290**, 2323 (2000).
[3] D. L. Donoho and C. Grimes, Proc. Natl. Acad. Sci. U.S.A. **100**, 5591 (2003).
[4] M. Belkin and P. Niyogi, Neural Comput. **15**, 1373 (2003).
[5] R. Coifman and S. Lafon, Appl. Comput. Harmon. Anal. **21**, 5 (2006).
[6] J. Bowen, A. Acrivos, and A. Oppenheim, Chem. Eng. Sci. **18**, 177 (1963).
[7] L. Monticelli, S. K. Kandasamy, X. Periole, R. G. Larson, D. P. Tieleman, and S.-J. Marrink, J. Chem. Theory Comput. **4**, 819 (2008).
[8] E. Spiga, D. Alemani, M. T. Degiacomi, M. Cascella, and M. Dal Peraro, J. Chem. Theory Comput. **9**, 3515 (2013).
[9] S. Izvekov, A. Violi, and G. A. Voth, J. Phys. Chem. B **109**, 17019 (2005).
[10] M. G. Saunders and G. A. Voth, Annu. Rev. Biophys. **42**, 73 (2013).
[11] I. Jolliffe, *Principal Component Analysis* (Wiley Online Library, 2005).
[12] S. Lafon, Y. Keller, and R. R. Coifman, IEEE Trans. Pattern Anal. Mach. Intell. **28**, 1784 (2006).
[13] N. Rabin and R. Coifman, in *Proceedings of the 12th SIAM International Conference on Data Mining (SDM 2012), Anaheim, California, USA* (SIAM, 2012).
[14] D. T. Gillespie, J. Phys. Chem. **81**, 2340 (1977).
[15] A. Zagaris, C. Vandekerckhove, C. W. Gear, T. J. Kaper, and I. G. Kevrekidis, Discrete Contin. Dyn. Syst., Ser. B **32**, 2759 (2012).
[16] P. G. Bolhuis, C. Dellago, and D. Chandler, Proc. Natl. Acad. Sci. U.S.A. **97**, 5877 (2000).
[17] R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker, Proc. Natl. Acad. Sci. U.S.A. **102**, 7426 (2005).
[18] A. Singer and R. R. Coifman, Appl. Comput. Harmon. Anal. **25**, 226 (2008).
[19] M. Hein and J. Y. Audibert, in *Proceedings of the 22nd International Conference of Machine Learning* (ACM, 2005), p. 289.
[20] R. R. Coifman, Y. Shkolnisky, F. J. Sigworth, and A. Singer, IEEE Trans. Image Process. **17**, 1891 (2008).
[21] M. A. Rohrdanz, W. Zheng, M. Maggioni, and C. Clementi, J. Chem. Phys. **134**, 124116 (2011).
[22] F. R. Chung, *Spectral Graph Theory* (AMS, 1997), Vol. 92.
[23] B. Nadler, S. Lafon, R. R. Coifman, and I. G. Kevrekidis, Appl. Comput. Harmon. Anal. **21**, 113 (2006).
[24] A. M. Ferrenberg and R. H. Swendsen, Phys. Rev. Lett. **61**, 2635 (1988).
[25] P. Burt and E. Adelson, IEEE Trans. Commun. **31**, 532 (1983).
[26] M. N. Do and M. Vetterli, IEEE Trans. Signal Process. **51**, 2329 (2003).
[27] S. M. Buchman, A. B. Lee, and C. M. Schafer, Stat. Methodol. **8**, 18 (2011).
[28] D. T. Gillespie, J. Chem. Phys. **113**, 297 (2000).
[29] J. Apostolakis, P. Ferrara, and A. Caflisch, J. Chem. Phys. **110**, 2099 (1999).
[30] D. S. Chekmarev, T. Ishida, and R. M. Levy, J. Phys. Chem. B **108**, 19487 (2004).
[31] A. Ma and A. R. Dinner, J. Phys. Chem. B **109**, 6769 (2005).
[32] T. A. Frewen, G. Hummer, and I. G. Kevrekidis, J. Chem. Phys. **131**, 134104 (2009).
[33] A. L. Ferguson, A. Z. Panagiotopoulos, P. G. Debenedetti, and I. G. Kevrekidis, J. Chem. Phys. **134**, 135103 (2011).
[34] D. A. Case, T. A. Darden, I. T. E. Cheatham, C. L. Simmerling, J. Wang, R. E. Duke, R. Luo, M. Crowley, R. C. Walker, W. Zhang, K. M. Merz, B. Wang, S. Hayik, A. Roitberg, G. Seabra, I. Kolossváry, K. F. Wong, F. Paesani, J. Vanicek, X. Wu, S. R. Brozell, T. Steinbrecher, H. Gohlke, L. Yang, C. Tan, J. Mongan, V. Hornak, G. Cui, D. H. Mathews, M. G. Seetin, C. Sagui, V. Babin, and P. Kollman, "AMBER 10," University of California, San Francisco, 2008.
[35] R. B. Best and G. Hummer, J. Phys. Chem. B **113**, 9004 (2009).
[36] Y. Duan, C. Wu, S. Chowdhury, M. C. Lee, G. M. Xiong, W. Zhang, R. Yang, P. Cieplak, R. Luo, T. Lee, J. Caldwell, J. M. Wang, and P. Kollman, J. Comput. Chem. **24**, 1999 (2003).
[37] W. L. Jorgensen, J. Chandrasekhar, J. D. Madura, R. W. Impey, and M. L. Klein, J. Chem. Phys. **79**, 926 (1983).
[38] U. Essmann, L. Perera, M. L. Berkowitz, T. Darden, H. Lee, and L. G. Pedersen, J. Chem. Phys. **103**, 8577 (1995).
[39] R. J. Loncharich, B. R. Brooks, and R. W. Pastor, Biopolymers **32**, 523 (1992).
[40] J. P. Ryckaert, G. Ciccotti, and H. J. C. Berendsen, J. Comput. Phys. **23**, 327 (1977).
[41] G. Matheron, Econ. Geol. **58**, 1246 (1963).
[42] G. Matheron, Adv. Appl. Probab. **5**, 439 (1973).
[43] R. R. Coifman and S. Lafon, Appl. Comput. Harmon. Anal. **21**, 31 (2006).
[44] Y. Aït-Sahalia, Econometrica **70**, 223 (2002).
[45] Y. Aït-Sahalia and P. A. Mykland, Econometrica **71**, 483 (2003).
[46] Y. Aït-Sahalia, Ann. Stat. **36**, 906 (2008).