Learning Effective SDEs from Brownian Dynamic Simulations of Colloidal Particles

Journal:	*Molecular Systems Design & Engineering*
Manuscript ID	ME-ART-05-2022-000086.R2
Article Type:	Paper
Date Submitted by the Author:	27-Feb-2023

Complete List of Authors:	Evangelou, Nikolaos; Johns Hopkins University - Homewood Campus, Chemical and Biomolecular Engineering Dietrich, Felix; Technical University of Munich Bello-Rivas, Juan; Johns Hopkins University Yeh, Alex; Johns Hopkins University Hendley, Rachel; Johns Hopkins University, Chemical and Biomolecular Engineering Bevan, Michael; Johns Hopkins University, Kevrekidis, Ioannis; Johns Hopkins University

The assembly of nano/micro colloidal particles into ordered materials is a crucial component of several cutting-edge technological applications (e.g., photonic crystals, meta-materials, cloaking devices etc.). Modeling the colloidal assembly process necessitates the ability to design, control and optimize the thermodynamics and kinetics of the process. Our work introduces a data-driven framework that enables the modelling of the colloidal assembly. We discover, through a manifold learning technique (Diffusion Maps), a set of effective collective observables from sampled data of Brownian Dynamics Simulations. We discuss the interpretability of our machine learning observables; we then learn the dynamics of the colloidal assembly process in terms of an effective Stochastic Differential Equation (eSDE). This is accomplished through either the traditional Kramers-Moyal expansion and/or through deep-learning schemes. We illustrate that the current deep-learning schemes allow a computationally efficient and accurate identification of the dynamics. We show that our discovered eSDE encodes accurately the physics not only of the Brownian Simulations but also, qualitatively, of experimental movies of colloidal particle trajectory ensembles. Our data-driven framework is transferable to systems with varying particle shape and to systems involving particle assembly (e.g. on curved surfaces) and can be used to design data-driven controller that accelerate/guide the assembly process.

Journal Name

ARTICLE TYPE

Cite this: DOI: 00.0000/xxxxxxxxxx

Learning Effective SDEs from Brownian Dynamic Simulations of Colloidal Particles

Nikolaos Evangelou,[a] Felix Dietrich[b], Juan M. Bello-Rivas[a], Alex Yeh[a], Rachel Stein[a], Michael A. Bevan[a] Ioannis G. Kevrekidis*[a]

Received Date
Accepted Date

DOI: 00.0000/xxxxxxxxxx

We construct a reduced, data-driven, parameter dependent effective Stochastic Differential Equation (eSDE) for electric-field mediated colloidal crystallization using data obtained from Brownian Dynamics Simulations. We use Diffusion Maps (a manifold learning algorithm) to identify a set of useful latent observables. In this latent space we identify an eSDE using a deep learning architecture inspired by numerical stochastic integrators and compare it with the traditional Kramers-Moyal expansion estimation. We show that the obtained variables and the learned dynamics accurately encode the physics of the Brownian Dynamic Simulations. We further illustrate that our reduced model captures the dynamics of corresponding experimental data. Our dimension reduction/reduced model identification approach can be easily ported to a broad class of particle systems dynamics experiments/models.

1 Introduction

The identification of nonlinear dynamical systems from experimental time series and image series data became an important research theme in the early 1990s[1–3]. After lapsing for almost two decades, it is now experiencing a spectacular rebirth. A key element of the older work was the use of neural architectures[2,4] (recurrent, convolutional, ResNet) motivated by traditional numerical analysis algorithms. Importantly, such architectures allow researchers to identify *effective*, coarse-grained, mean-field type evolution models from *fine-scale* (atomistic, molecular, agent-based) data[5,6].

In this paper, we identify coarse-grained, effective stochastic differential equations (eSDE) for colloidal particle self-assembly based on fine-grained, Brownian dynamics simulations under the influence of electric fields[7,8]. We demonstrate that the identified eSDE encodes accurately the physics of the Brownian Dynamic simulations and captures the dynamics of corresponding experimental data. Those experiments have previously been shown to quantitatively match to BD simulations *at equilibrium* in terms of time-averaged distribution functions[8–10]. Figure 1 shows a sample path of a latent space trajectory $\{t, \phi(t)\}_{t\geq 0}$ computed through our learned eSDE. The corresponding instantaneous particle conformations are indicated at representative points along the trajectory. A key feature of our work is the selection of the coarse-grained observables (the variables of our eSDE) in a data-driven manner, using manifold learning techniques like Diffusion Maps[11]. The dependence of the dynamics on physical control parameters (here a driving voltage) is included in the neural architecture and learned during training. A second key feature is that the neural network architecture for eSDE identification is not based on established Kramers-Moyal estimation techniques e.g.[12,13], but rather (in the spirit of the early work mentioned above) on numerical stochastic integration algorithms[14].

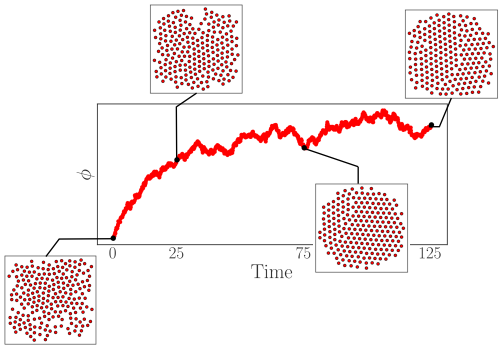

Fig. 1 A trajectory of an effective, reduced eSDE in the data driven collective coordinate ϕ for electric-field mediated colloidal crystallization.

The motivation of the effective SDE is to better understand the ability to assemble nano- and micro- colloidal particles into ordered materials and controllable devices. This could provide the

[a] Department of Chemical and Biomolecular Engineering, Johns Hopkins University, 3400 North Charles Street, Baltimore, 21218, MD *Email*:yannisk@jhu.edu, Tel:(410) 516-2906

[b], *Department of Informatics, Technical University of Munich, 3 Boltzmannstr ,Munich, 85748 Germany.*

† Electronic Supplementary Information (ESI) available: [details of any supplementary information available should be included here]. See DOI: 00.0000/00000000.

basis for emerging technologies (e.g. photonic crystals, metamaterials, cloaking devices, solar cells, etc.),[15] but also impact traditional applications (e.g. ceramics, coatings, minerals, foods, drugs[16,17]). Despite the range of applications employing microscopic colloidal particles, current state-of-the-art[18] capabilities for manipulating microstructures in such systems are limited in two ways: (a) the degree of order that can be obtained, and (b) the time required to generate ordered structures. Both of these limitations are due to fundamental problems with designing, controlling, and optimizing (i.e. engineering) the thermodynamics and kinetics of colloidal assembly processes.

The identified parameter-dependent eSDE models we construct can be used for design applications that involve interpolation for new parameter values (not used in the training set). This allows modeling colloidal assembly by performing simulations with the (simpler) data-driven surrogate, thus sidestepping the computational cost of performing full Brownian Dynamic simulations at these new parameter values.[19]. This work also paves the way toward data-driven model-based control schemes for the kinetics of colloidal assembly processes using parametric driving (e.g. via an external electric field)[19]. Basically, understanding the transient stochastic evolution of colloidal ensembles across different microstructural states provides the information necessary to implement realistic control of such processes. As a direct example, a related modeling approach[7,20] was previously used to perform open and closed-loop control of colloidal crystallization, so to rapidly assemble defect-free perfect crystals with circular morphology. Ths was accomplished by removing grain boundaries and controlling directional stresses on crystals.[21,22] Crucially, we also tackle the issue of interpretability of the learned effective dynamic model by exploring relations between data-driven and candidate physically meaningful observables. Those coarse physical observables are *order parameters* that provide intuition for the colloidal self-assembly process[21–23,23,24]. We combine the data-driven detection of effective latent spaces with the neural network based, numerical analysis inspired, identification of parameter-dependent stochastic eSDEs with state-dependent diffusion. This is based on fine scale data from both Brownian dynamics simulations and from experimental colloidal crystallization movies, and the results are compared.

Developing low-dimensional surrogate models for physical systems has been explored by a number of authors. We report some approaches that utilize machine learning and/or dimensionality reduction here that could be beneficial to the reader. The authors in[25] identified an effective, coarse grained Fokker-Planck using Kramers-Moyal with an application to micelle-formation of surfactant molecules. The identified equation in[25] was constructed in terms of the physical coarse variable, size of the cluster of the surfactant molecules. The authors in[26] constructed an 1D Smoluchowski equation in terms of coarse physical variables (radius of gyration or the average crystallinity) for small colloidal systems of 32-particles. The authors in[27] used simulation and experimental colloidal ensembles with smaller than 14 particles to fit two dimensional Fokker-Planck and Langevin equations. The two coarse variables in which the dynamics are being identified capture the condensation and anisotropy of those small ensembles. A detailed review that summarizes applications of machine learning to discover collective variables and for sampling enhancement was conducted by the authors[28]. A framework to advance the simulation time by learning the effective dynamics (LED) of molecular systems was proposed by the authors in[29]. LED uses mixture density network (MDN) autoencoders to learn a mapping between the molecular systems and latent variables and evolves the dynamics using long short-term memory MDNs. In the context of accelerating molecular simulations, the authors in[30] proposed a framework tested on polymeric systems that utilize graph clustering to obtain coarse observable and allows to model system's evolution for long-time dynamics.

The main machine learning tools of our work involve (a) utilizing a dimensionality reduction scheme that discovers a lower dimensional structure of a given data set and (b) a deep neural network architecture that learns an eSDE.

Regarding the first aspect of dimensionality reduction a wide range of techniques have been proposed for discovering a set of reduced observables. Among others, Principal Component Analysis[31], Isomap[32], Local Linear Embedding[33], Laplacian Eigenmaps[34], Autoencoders[35] and our method of choice: Diffusion Maps[11]. The Diffusion Maps algorithms enables the discovery of reduced coordinates when data are sampled from signal processing[36], from networks[37], from (stochastic) differential equations[38,39] but also from Molecular[6] and Brownian simulations[7].

A traditional approach for learning eSDEs has been the Kramers-Moyal expansion[5,12,13] and a detailed description of this approach is given in Section 2.4.1. For non-Gaussian stochastic differential equations, modifications of the tradiational Kramers-Moyal expansion have also been proposed[40,41]. In[42] the authors proposed a stochastic physics-informed neural network framework (SPINN) that minimizes the distance between the predicted moments of the network (drift and diffusivity) from moments computed with Kramers-Moyal. The authors in[43] proposed an extension of the framework called Sparse Identification of Nonlinear Dynamics (SINDy) that can be used for stochastic dynamical systems. The authors in[44] proposed a physics informed generative model termed *generative ensemble-regression* that learns to generate *fake* sample paths from given densities at several points in time, without point-wise paths correspondence. The authors in[45] extracted an eSDE from long time series data in a memory-efficient way, including learning the eSDE in latent variables. This approach is valuable if the data is available as a few, long time series. In our approach we handle pairs of successive snapshots instead. The most similar approach to learning eSDEs to the one selected for our work is[46]. The authors introduce a Variational Autencoder (VAE) framework for recovering latent dynamics governed by an eSDE. In their method, the latent space and the stochastic differential equation are identified together within the VAE scheme. Their loss function is also based on the Euler-Maruyama scheme.

Our work deviates from the approaches mentioned above in three key aspects: (a) we explicitly separate the latent space construction from learning the eSDE; (b) we extend the loss function informed by numerical integration schemes from[14] to allow for additional parameter dependence. Our latent space is defined

through Laplace-Beltrami operator eigenfunctions, so, different from[46], (c) our latent space coordinates are invariant to isometry and sampling density in the original space by construction.

2 Methodology

2.1 Brownian Dynamics

We model electric field-mediated quasi-2D colloidal assembly in the presence of a quadrupole electrode. An illustration of the set up is shown in Figure 2 In our simulations, each configuration consists of $N = 210$ particles. The interactions between the colloidal particles are electrostatic double layer repulsion $u^{pp}_{e,i,j}$, dipole-field potentials $u^{pf}_{de,i}$ and dipole-dipole interaction potential $u^{pp}_{dd,i,j}$. The electrostatic repulsion, $u^{pp}_{e,i,j}$, between two particles i and j is computed by.

$$u^{pp}_{e,i,j}(r_{i,j}) = B^{pp}\exp\{\left[-\kappa(r_{ij}-2\alpha)\right]\}. \tag{1}$$

In Equation (1) r_{ij} denotes the center-to-center distance between the patricles, α is the radius of each particle and B^{PP} is the electrostatic repulsion pre-factor between colloidal particles.

The dipole field potential $u^{pf}_{de,i}$ in the spatially varying electric field for each particle i is computed by

$$u^{pf}_{de,i}(\mathbf{r}_i) = -2kT\lambda f_{cm}^{-1}[E(\mathbf{r}_i)/E_0]^2, \tag{2}$$

where $\mathbf{r}_i$ is the position of the i^{th} particle, k is the Boltzmann's constant, T is the temperature, f_{cm} is the Clausius-Mossotti factor, λ is a non-dimensional amplitude given by the relation $\lambda = \frac{\pi\varepsilon_m\alpha^3(f_{cm}E_0)^2}{kT}$, ε_m is the medium dielectric constant, the local electric field magnitude is given by $E(\mathbf{r}_i)$. The constant E_0 is given by the expression

$$E_0 = \frac{1}{\sqrt{8}}(V_{pp}/d_g) \tag{3}$$

where V_{pp} denotes the peak-to-peak voltage and d_g the electrode gap. The dipole-dipole interaction potential $u^{pp}_{dd,i,j}$ between two particles i and j is estimated by

$$u^{pp}_{dd,i,j}(\mathbf{r}_{ij}) = -kT\lambda P_2(cos\theta_{ij})(2\alpha/r_{ij})^3[E(\mathbf{r}_i/E_0)]^2. \tag{4}$$

$P_2(cos\theta_{ij})$ is the second Legendre polynomial, θ_{ij} denotes the angle between the particle centers and the electric field direction.

The electric field at the center of the quadrupole can be approximated by the expression

$$\left|\frac{E(\mathbf{r}_i)}{E_0}\right| = \frac{4r}{d_g} \tag{5}$$

where r is the distance from the quadrupole center.

The motion of the Brownian particles is governed by the equation

$$\mathbf{r}(t+\Delta t) = \mathbf{r}(t) + \frac{\mathbf{D}^P}{kT}(\mathbf{F}^P + \mathbf{F}^B)\Delta t + \nabla\cdot\mathbf{D}^P\Delta t \tag{6}$$

where $\langle\mathbf{F}^B\rangle = 0$, $\langle\mathbf{F}^B(t_1)(\mathbf{F}^B(t_2)^{\mathrm{T}} = 2(kT)^2(\mathbf{D}^P)^{-1}\delta(t_1-t_2)$, $\mathbf{r}(t)$ denotes the position vector for all the N particles at time t, $\mathbf{F}^B$ denotes the Brownian force vector and $\mathbf{F}^P$ the total conservative force vector. The conservative force acting on each particle i is given by

$$\mathbf{F}^P_i = \nabla_{\mathbf{r}_i}\left[u^{pf}_{de,i} + \sum_{j\neq i}(u^{pp}_{e,i,j} + u^{pp}_{dd,i,j})\right]. \tag{7}$$

$\mathbf{D}^P$ denotes the diffusivity tensor estimated by the Stoke-Einstein relation

$$\mathbf{D}^P = kT(\mathbf{R}^P)^{-1} \tag{8}$$

where $\mathbf{R}^P$ is the grand resistance tensor $\mathbf{R}^P$ given by

$$\mathbf{R}^P = (\mathbf{M}^\infty)^{-1} + \mathbf{R}_{2B} - \mathbf{R}^\infty_{2B} \tag{9}$$

where $\mathbf{R}_{2B}$ are the pairwise lubrication interactions and $(\mathbf{M}^\infty)^{-1} - \mathbf{R}^\infty_{2B}$ the many-bodied far-field interaction above a no-slip plane. All the parameters used for the BD simulations are included in Table 1 of the SI.

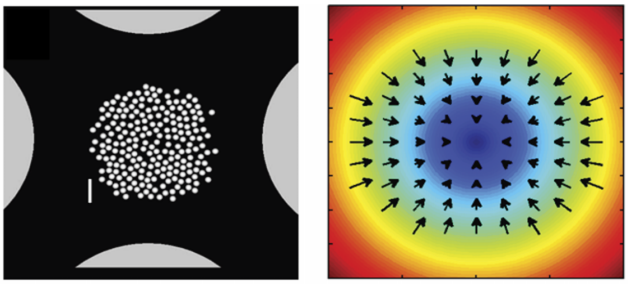

Fig. 2 [Left] Top view of simulated experiments of quasi 2D configurations of $N = 210$ colloidal particles compressed with a quadrupole electrode. [Right] Electric field magnitude contour plot in the vicinity of the quadrupole electrode center. The arrows indicate the relative magnitude and direction of force due to dipole-field interactions. Taken from J.Chem. Phys 144, 204904 (2016) with permission.

2.2 Diffusion Maps

Introduced by[11], Diffusion Maps offer a parametrization of a data set of points $\mathbf{X} = \{x_i\}_i^N$ sampled from a manifold $\mathscr{M}$, where $x_i \in \mathbb{R}^m$ by uncovering its *intrinsic geometry*. This parametrization can then be used to achieve dimensionality reduction of the data set. This is obtained by initially constructing an affinity matrix $\mathbf{A} \in \mathbb{R}^{N\times N}$ through a kernel function, for example the Gaussian Kernel

$$A_{ij} = \exp\left(\frac{-||x_i - x_j||^2}{2\varepsilon}\right), \tag{10}$$

where $||\cdot||$ denotes a norm of choice. In this work we choose the l^2 norm; ε is a hyperparameter regulating the rate of decay of the kernel. To achieve a parametrization of $\mathbf{X}$ regardless of the sampling density, a normalization of $\mathbf{A}$ is performed as follows

$$P_{ii} = \sum_{j=1}^{N} A_{ij}, \tag{11}$$

$$\tilde{\mathbf{A}} = \mathbf{P}^{-\alpha}\mathbf{A}\mathbf{P}^{-\alpha} \tag{12}$$

where $\alpha = 1$ is set to factor out the effect of sampling density. The kernel $\tilde{\mathbf{A}}$ is further normalized

$$W(x_i, x_j) = \frac{\tilde{A}(x_i, x_j)}{\sum_{j=1}^{N}\tilde{A}(x_i, x_j)} \tag{13}$$

so that the matrix $\mathbf{W}$ becomes a row stochastic matrix. The eigendecomposition of $\mathbf{W}$ results in a set of eigenvectors ϕ and eigenvalues λ

$$\mathbf{W}\phi_i = \lambda_i \phi_i. \tag{14}$$

To check if dimensionality reduction of $\mathbf{X}$ is possible, selection of the eigenvectors ϕ that parameterize independent directions (non-harmonic eignvectors) is needed. In our work this selection was made by implementing the algorithm presented in[47]. If the number of the non-harmonic eigenvectors is smaller than the original dimensions of $\mathbf{X}$ then Diffusion Maps achieves dimensionality reduction.

Our data "points", x_i, consist of planar configurations of 210 particle locations obtained either from evolving computations or from experimental movies; our data set is $\mathbf{X} = \{x_i\}_{i=1}^N$ where $x_i \in \mathbb{R}^{210\times 2}$. A number of preprocessing steps are performed before Diffusion Maps can be computed. All configurations are centered and aligned to a reference configuration by using Procrustes analysis, in particular the Kabsch algorithm[48,49]. The reference configuration was selected as the configuration that has the smallest value of the order parameter Rg (see Section 2.5). Centering the data and applying the Kabsch algorithm removes the translational and rotational degrees of freedom. We then compute the density function, $\mathbf{f}_i$, for each configuration at the nodes of a grid and we normalize its integral to one. The density was estimated by a kernel density estimation using Gaussian Kernels in Python. More precisely, the *gaussian_kde* module from *scipy* was used for this computation. The bandwidth for the kernel estimation was selected based on Scott's Rule[50]. The Diffusion Maps algorithm then is applied to the data set $\mathbf{F} = \{\mathbf{f}_i\}_i^N$ of the collected normalized density function discretizations. The density formulation f_i eliminates the problem of permutational invariance of the particles in defining pairwise distances. As we mentioned also earlier the selection of the leading non-harmonic Diffusion Maps coordinates was made by the local linear algorithm proposed by the authors in[47]. For our Diffusion Maps computations the *datafold* package was used[51].

2.3 Nyström Extension

Given a new out-of-sample data point, $x_{\text{new}} \notin \mathbf{X}$ (and subsequently $f_{new} \notin \mathbf{F}$), in order to embed it in the Diffusion Maps coordinates one might add it to the data set and recompute Diffusion Maps. However, this is computationally inefficient and will lead to a new Diffusion Maps coordinate system for every new point added in the data set. To avoid these issues the Nyström Extension formula[52,53] can be used

$$\phi_i(f_{\text{new}}) = \frac{1}{\lambda_i}\sum_{j=1}^{N} \tilde{W}(f_{new}, f_j)\phi_i(f_j), \tag{15}$$

where $\phi_i(f_{\text{new}})$ is the estimated value of the i^{th} eigenvector for the new point f_{new}, λ_i is the corresponding eigenvalue, and $\phi_i(f_j)$ is the j^{th} component of the i^{th} eigenvector.

This formula is extremely useful in mapping trajectories either from the Brownian Dynamics simulations or from experimental snapshots to the Diffusion Maps coordinates (an operation called "restriction"). Restricted long trajectories are used as a *test* set to validate our estimated eSDEs.

2.4 Learning SDEs from data

In this section we describe two approaches to *estimate* SDEs from data. Let $x(t)$ be a stochastic vector-valued variable whose evolution is governed by the SDE

$$dx(t) = \nu(x(t))dt + \sigma(x(t))dB_t, \tag{16}$$

where $\nu : \mathbb{R}^m \to \mathbb{R}^m$ is the drift, $\sigma : \mathbb{R}^m \to \mathbb{R}^{m\times m}$ is the diffusivity matrix, and B a collection of m one-dimensional Wiener processes. The dynamics of such process can be approximated by *estimating* the two functions ν and σ. We show how this estimation can be performed, either from the statistical definition of the terms, based on the Kramers-Moyal expansion[12,13], or via a deep learning architecture inspired by stochastic numerical integrators[14].

2.4.1 Kramers-Moyal expansion

For a stochastic process $x(t)$, the differential change in time of its probability density $P(x,t)$ is given by

$$\frac{\partial P(x,t)}{\partial t} = \sum_{n=1}^{\infty}\left(-\frac{\partial}{\partial x}\right)^n \mathbf{D}^{(n)}(x,t)P(x,t), \tag{17}$$

which is known as the Kramers-Moyal expansion[13]. The moments of a transition probability, jumping from a position $x(t_k)$ to a nearby position $x(t_{k+h})$ in the next time step, are given by

$$D^{(n)}(x,t) = \frac{1}{n!}\lim_{h\to 0}\frac{\langle [x(t_{k+h}) - x(t_k)]^n\rangle}{h}. \tag{18}$$

where $\langle\cdot\rangle$ denotes the average. When $x(t)$ is a Gauss-Markov process; only the first two moments of Equation 17 are non-zero, and the Kramers-Moyal expansion reduces to the forward Fokker-Planck equation. The Fokker-Planck equation provides an alternative description of the dynamics expressed by Equation 16. For N variables, the Fokker-Planck equation is given by

$$\frac{\partial P}{\partial t} = -\sum_{i=1}^{N}\frac{\partial}{\partial x_i}\left(D_i^{(1)}(x)P\right) + \sum_{i,j=1}^{N}\frac{\partial^2}{\partial x_i \partial x_j}\left(D_{ij}^{(2)}(x)P\right), \tag{19}$$

where $\mathbf{D}^{(1)}$ and $\mathbf{D}^{(2)}$ are also the drift and diffusion coefficients and the connection with the coefficients of Equation (16) is given by the expressions $\nu(x) = \mathbf{D}^{(1)}, \sigma^2 = 2\mathbf{D}^{(2)}$. The estimation of the drift and the diffusivity at a point x^i can be performed by multiple local parallel simulations ("bursts").

$$\begin{aligned} \nu_i(x(t_k)) &\approx \frac{1}{h}\langle x_i(t_{k+h}) - x_i(t_k)\rangle, \\ \sigma_{ij}^2(x(t_k)) &\approx \frac{1}{h}\langle (x_i(t_{k+h}) - x_i(t_k))(x_j(t_{k+h}) - x_j(t_k))\rangle. \end{aligned} \tag{20}$$

2.4.2 Deep Learning - Numerical Integrators

The deep learning approach that we have followed for the identification of eSDEs is based on the work of[14]. In this approach, the drift and diffusivity are estimated through two networks ν_θ and σ_θ, where θ are the weights of the networks. In our work

we also introduce a small but meaningful modification of their method by also including a "parameter neuron" along with the snapshot of inputs $\widetilde{\mathbf{D}} = \{x^i(t_{k+h}), x^i(t_k), h^i, p^i\}_{i=1}^N$. This modification allowed us to learn *parameter dependent eSDEs*. The parameter p in our case is the applied voltage to the particles (V^*). The collected data required for this approach do not necessarily need to be sampled from long trajectories. Snapshots $\widetilde{\mathbf{D}}$ are sufficient as long as the region of interest is sampled densely enough. Here we introduce the scheme for the two-dimensional case since our identified eSDE is also two-dimensional. Each snapshot $\widetilde{D}^i$ in this network includes (a) a point at time k in space $x^i(t_k) = (x_1^i(t_k), x_2^i(t_k))$; (b) its coordinates after a short time evolution $x^i(t_{k+h}) = (x_1^i(t_{k+h}), x_2^i(t_{k+h}))$; (c) the time interval between the two points, h^i; and (d) a parameter p^i for the parameter dependent eSDE. Between different sampled snapshots the time step h does not need to be uniform; in our case this property will prove to be quite useful as discussed in the results.

The loss function used in our case (based on [14]) is derived from the Euler-Maruyama scheme, a numerical integration method for SDEs. The scheme for the two-dimensional case,

$$\begin{bmatrix} x_1^i(t_{k+h}) \\ x_2^i(t_{k+h}) \end{bmatrix} = \begin{bmatrix} x_1^i(t_k) \\ x_2^i(t_k) \end{bmatrix} + h^i \begin{bmatrix} \nu_\theta(x_1^i(t_k), p^i) \\ \nu_\theta(x_2^i(t_k), p^i) \end{bmatrix} + \begin{bmatrix} \sigma_\theta(x_1^i(t_k), x_1^i(t_k), p^i) & \sigma_\theta(x_1^i(t_k), x_2^i(t_k), p^i) \\ \sigma_\theta(x_2^i(t_k), x_1^i(t_k), p^i) & \sigma_\theta(x_2^i(t_k), x_2^i(t_k), p^i) \end{bmatrix} \begin{bmatrix} dB_{t_1} \\ dB_{t_2} \end{bmatrix} \quad (21)$$

where dB_{t_1}, dB_{t_2} are normally distributed around zero with variance h^i. This scheme has a similar form for higher dimensions. This scheme implies that each $x^i(t_{k+h})$ is normally distributed,

$$x^i(t_{k+h}) \sim \mathcal{N}\left(x^i(t_k) + h^i \nu_\theta(x^i(t_k), p^i), h^i \sigma_\theta(x^i(t_k), p^i)^2\right) \quad (22)$$

where the mean $\mu_\theta^i = x^i(t_k) + h^i \nu_\theta(x^i(t_k), p^i)$ and the covariance matrix $\Sigma_\theta^i = h^i \sigma_\theta(x^i(t_k), p^i)^2$.

Under this assumption, we formalize a loss function that will lead to a maximization of the probability of Equation (22). This is achieved by combining the logarithm of the probability density of the multivariate normal distribution with the assumed mean and variance from Equation (22):

$$\mathscr{L}(\theta | x^i(t_{k+h}), x^i(t_k), h^i, p^i) := \log\left|\det(\Sigma_\theta^i)\right| + \frac{1}{2}(x^i(t_{k+h}) - \mu_\theta^i)^{\mathrm{T}} (\Sigma_\theta^i)^{-1} (x^i(t_{k+h}) - \mu_\theta^i) \quad (23)$$

where the constant term is dropped since it does not affect the minimization. The training of the network is performed by minimizing the loss function $\mathscr{L}$ over the training set $\widetilde{\mathbf{D}}$.

2.5 Order Parameters - Free Energy Landscapes

Order parameters are coarse, collective variables that summarize the physics involved in the colloidal self-assembly process [54]. These quantities often encapsulate features of interest, for example the compactness, or the local or global degree of order of a particle assembly [22,54]. Such variables can then be used to formulate (ideally analytical, but practically, here, data-driven) models to study the collective dynamics of complex systems. Domain scientists often have prior knowledge of good candidate order parameters based on experience, intuition, or mathematical derivations, and validate a good variable choice among such candidates [22]. Order parameters that are typically used to study colloidal self-assembly are Rg, ψ_6 and C_6 [20]. Rg is the radius of gyration, which quantifies whether the particle ensemble is expanded in a fluid state or condensed in a crystalline state; ψ_6 is the degree of global six-fold bond orientation order (near 0 for ideal gas, 1 for perfect single domain crystal); C_6 is the ensemble average of a local order parameter based on the number of neighbors each particle has with 6-fold order (0 for no neighbors with 6 fold-order, 1 for 6 neighbors with 6-fold order). Expressions for each of these are included in our prior publications. [20]. In our case, we do not use these theoretical "usual candidate" colloidal order parameters in our model construction. We allowed the data to determine which and how many collective variables are needed by using the Diffusion Maps scheme. We then attempt to establish explainability of our data-driven variables in terms of the theoretical ones Rg, ψ_6, C_6, see Section 3.1.

The effective potential $G(x)$ quantifies the free energy landscape. It is obtained from the equilibrium probability distribution, the steady-state solution of the Fokker Planck equation [13]. The integral equation for this effective potential (alternatively, potential of mean force or effective free energy), is given (up to a constant) by the equation [13]

$$G(x^i) = -kT \int_0^{x^i} \left[2(\sigma^2)^{-1}(\nu - \nabla \cdot \frac{\sigma^2}{2})\right] \cdot dr. \quad (24)$$

where ν is drift and σ is the diffusivity. For our computations we chose the origin as the reference state.

3 Results

3.1 Latent Observables

We start by coarse-graining Brownian Dynamics simulation (details about the simulations and the sampling are in the Appendix). Diffusion Maps discovers two latent non-harmonic coordinates denoted as ϕ_1, ϕ_2 [47]. This suggests that two Diffusion Maps coordinates are enough to provide a more parsimonious representation of the original data set. The selection of those two Diffusion Maps coordinates was made by applying the local linear regression algorithm suggested by the authors in [47]. We first check the interpretabilty of these data-driven observables by coloring the Diffusion Maps coordinates as functions of the three order parameters Rg, ψ_6 and, C_6. Those order parameters are physically meaningful coarse variables that measure the degree of condensation of the material (Section 2.5). It is worth highlighting that no pair of the three order parameters is exactly one-to-one with the Diffusion Maps coordinates; however a clear trend appears: condensed configurations arise at the center of our manifold embedding, while further out from the center disordered/fluid like structures are observed. This implies that our latent coordinates encode the physics of the Brownian Dynamic Simulations.

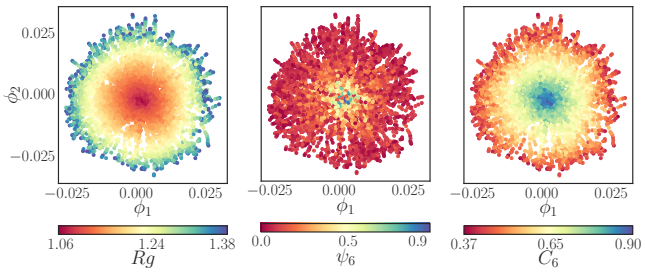

Fig. 3 The two leading Diffusion Maps coordinates (ϕ_1, ϕ_2) colored by the physically meaningful order parameters Rg, ψ_6 and, C_6 respectively (Rg clearly correlates with C_6).

In Figure 3, coloring the Diffusion Maps coordinates with the three order parameters visually indicates a macroscopic rotational symmetry. This symmetry is probably coming from the particle density fields used for our Diffusion Maps computations, see Section 2.2 for more details). In addition to the symmetries factored out by the Kabsch algorithm, testing for an additional azimuthal symmetry requires taking data in the form of several radial slices from every picture, in order to test/confirm that they all appear approximately the same. We do not further pursue this, but would like to point to relevant work on *modding out* symmetries that might be useful to the reader in the context of dynamical systems[55,56] and vector Diffusion Maps[57].

3.2 Learning Effective SDEs

For Brownian Dynamics simulations at fixed normalized voltage $V^* = \frac{V}{V_{xal}} = \frac{1.51\text{V}}{1.89\text{V}} = 0.8$ (see Section A2 in the Appendix), we estimate the drift and diffusivity in the Diffusion Maps coordinates with (a) a neural network architecture; and with (b) the Kramers-Moyal expansion. The drift estimated by the two approaches is plotted as a vector field on the two Diffusion Maps coordinates, Figure 4. The drift component gives us an estimate of what the trajectories will locally tend to do on average. As can be seen from Figure 4 (on average) the trajectories will evolve towards the center of the manifold, and therefore towards more condensed structures as expected from the detailed Brownian Dynamics simulations.

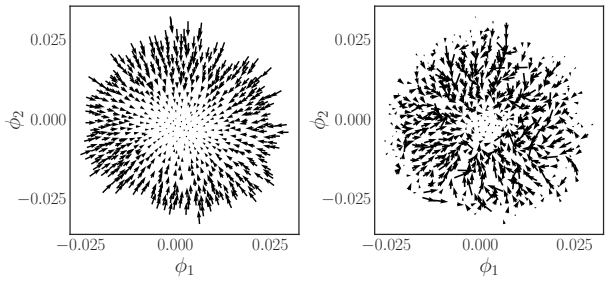

Fig. 4 The estimated drift from the neural network and the Kramers-Moyal respectively is plotted as the on average vector field in the two Diffusion Maps coordinates (ϕ_1, ϕ_2). The vector field is plotted in subsampled data sets to improve visualization.

The estimated vector field from the neural network appears *smoother* compared to the one obtained with the Kramers-Moyal. This could be partially attributed to the fact that the neural network during training for the drift learns simultaneously from many points per iteration through the loss function. On the other hand, Kramers-Moyal uses bursts around each individual data point separately, without information about the nearby points. Figure 5 offers another comparison between the estimated drift from the neural network and the Kramers-Moyal Expansion. The estimated drifts of the two methods are comparable, with the drift estimated from the neural network often slightly larger in magnitude. The comparison for the estimated diffusivity with the two

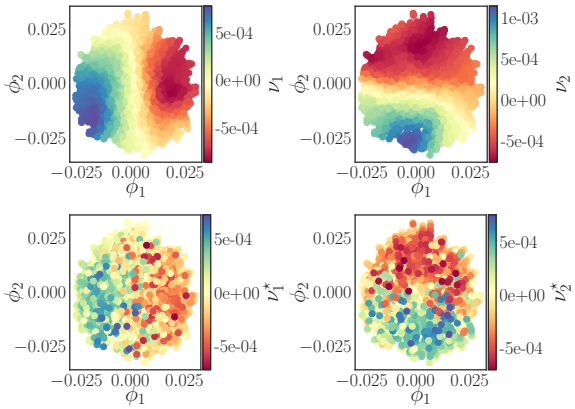

Fig. 5 The estimated drift, v_1, v_2, from the neural network (first row) and the Kramers-Moyal, $v_1^\star, v_2^\star$, (second row) is plotted as a function of the Diffusion Maps coordinates (ϕ_1, ϕ_2).

approaches leads to similar conclusions, Figure 6. The diffusivity estimated from the neural network appears smoother compared to the one estimated from the Kramers-Moyal. Note the neural network approach estimated the diffusivity matrix without assuming it to be diagonal (as opposed to its Kramers-Moyal estimation). Even though a trend appears in the diffusivity computed through the network the computed diffusivity is practically constant along the data and the trend is just an artifact of the fitted diffusivity through the network. This would also become evident for the null models with constant diffusivity used in Dietrich et al[14].

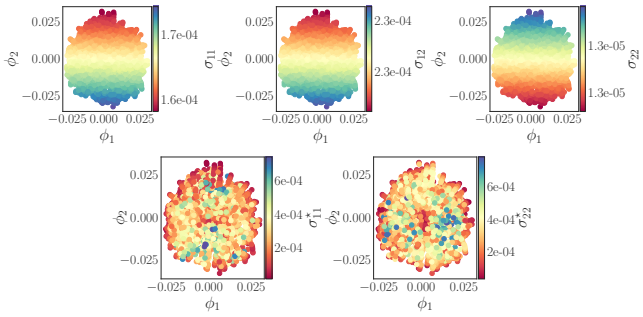

Fig. 6 The estimated diffusivity from the neural network, $\sigma_{11}, \sigma_{22}, \sigma_{12} = \sigma_{21}$ (first row) and the one from Kramers-Moyal, $\sigma_{11}^\star, \sigma_{22}^\star$ (second row) is plotted as a function of the Diffusion Maps coordinates (ϕ_1, ϕ_2).

Given the estimated drift and diffusivity we wish to generate trajectories for the reduced eSDEs in Diffusion Maps coordinates. Evaluating the drift and diffusivity along the integration is trivial for the trained neural network. For the Kramers-Moyal expansion, interpolating from the computed values becomes necessary. Since the functions of the estimated drift and diffusivity are not smooth enough for a global interpolation scheme, a local nearest neighbor interpolation was used during the integration. The numerical integrator used for both cases was the Euler-Maryama scheme.

From the estimated coefficients (drift and diffusivity), and more precisely from the average vector field in Figure 4, it is expected that the trajectories will evolve toward the center of the embedding for both estimated eSDEs.

To evaluate our models' performance against ground-truth data, we sampled Brownian Dynamics trajectories and *restricted* those trajectories with Nyström Extension in the reduced Diffusion Maps coordinates (ϕ_1, ϕ_2). A comparison between the mean of 100 trajectories obtained from the two eSDEs is contrasted to the mean of 100 restricted trajectories computed with Brownian Dynamics simulations in Figure 7. The dynamics from the network on average provide more accurate results compared to the ones obtained from the Kramers-Moyal. To successfully estimate

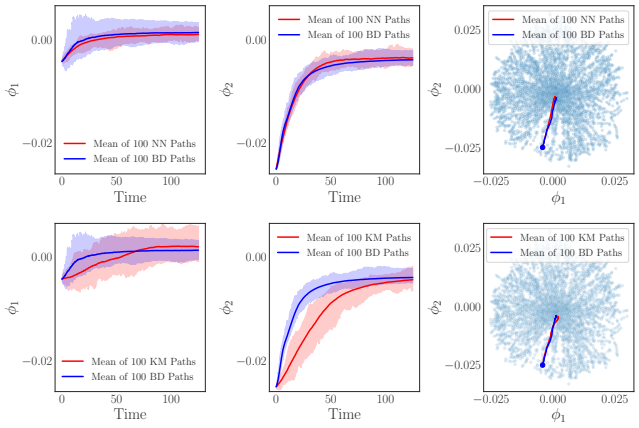

Fig. 7 The estimated dynamics from the neural network (first row) and the Kramers-Moyal (second row) is shown compared to restricted (with Nyström) trajectories of the Brownian Dynamics in the two Diffusion Maps coordinates (ϕ_1, ϕ_2). The mean of 100 trajectories (starting from the same initial condition) is used for all cases. To get a visual inspection of the variance in the estimated 100 trajectories, the area between the maximum and minimum values for those trajectories is being "filled" with solid color. The red paths correspond to the data-driven eSDEs (neural network or Kramers-Moyal) and the blue paths to the restricted Brownian Dynamics Paths.

both the drift and the diffusivity for the neural network, training was performed in two stages. First, we chose a time step h that gave a reasonable estimation of the drift; we then *fixed* the part of the network that estimates the drift, and used snapshots at smaller time steps h' to estimate the diffusivity (see the discussion in[14]).

We provide an uncertainty quantification (error analysis) comparison of the neural network model in Section A6 of the SI. This analysis provides some more quantitative measurements on of how certain the reported predictions are. The results suggest the the robustness of the neural-network model. In addition, in Section A7 of the SI we discuss a more quantitative comparison between the two surrogate identified eSDEs (with Kramers-Moyal and the neural-network). The results in this case suggest that the discrepancy between the two surrogate models is in the range of the expected error estimations of the neural-network model.

3.3 Learning a Parameter-Dependent eSDE

In this section we illustrate the ability to learn a parameter dependent eSDE. For this case only the neural network was used. We sampled data (snapshots) for four different voltages, $\mathbf{V}^\star = \{0.5, 0.6, 0.7, 0.8\}$. The larger the voltage becomes, the larger the force that is acting on the particles, and thus the faster they condense. On the contrary, as the voltage becomes lower, the particles can move more freely and they condense slower. Those physical features are expected to be captured in terms of the drift and diffusivity of our eSDE . As the voltage increases the drift (force) is expected to increase and the diffusivity to decrease. In Figure 8 the obtained results from the neural network appear to conform to those features of the simulations.

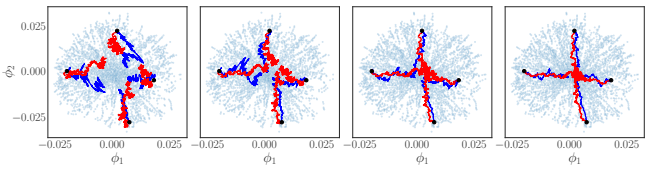

Fig. 8 In the Diffusion Maps coordinates (ϕ_1, ϕ_2) we illustrate trajectories computed through the neural network trained for different values of the voltage, $V^\star = \{0.5, 0.6, 0.7, 0.8\}$ plotted from left to right. Trajectories of the estimated eSDE from the neural network (red paths) are contrasted to restricted (with Nyström Extension) trajectories computed with Brownian Dynamics (blue paths) for different values of the voltage.

For four different initial conditions, and for the same time length, trajectories were integrated with Euler-Maryama; the same integration step was used for all parameter values. As the parameter value increases, from left to right in Figure 8, the trajectories appear to travel faster towards the center of the embedding (towards more condense configurations). This can be attributed to fact that the drift increases in magnitude. In addition, as the voltage decreases, the trajectories appear more noisy, since the diffusivity increases.

In Figure 9 the estimated diffusivity is plotted against the Diffusion Maps coordinates and is colored with the voltage value. Figure 9 supports the observation that as the voltage increases the diffusivity decreases.

For the estimation of the parameter dependent eSDE, the flexibility of having different step sizes h^i proved quite useful. For smaller values of the voltage $V^\star$, for which the drift is also smaller, larger time steps could be accurately employed.

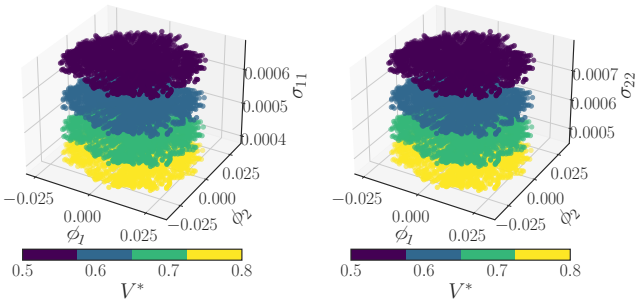

Fig. 9 The estimated diagonal diffusivity is plotted against ϕ_1, ϕ_2. Different colors correspond to different values of the parameter (V^*). The trend in the estimated diffusivity is in agreement with the physics of the problem. The higher voltage forces the system to condense faster. On the contrary, the smaller the voltage becomes, the easier it is for the particles to move freely, and thus the effective diffusivity increases.

3.4 Free Energy Landscapes

We illustrate the ability to estimate free energy landscapes (potential functions) from the coefficients of the reduced eSDE. In Figure 10 the Free Energy in kT units is plotted as a function of the Diffusion Maps coordinates ϕ_1, ϕ_2 for the four different voltages in an increasing order. From Figure 10 the larger the voltage, the larger the range of effective potential values becomes. From our computations it appears that the term $\nabla \cdot \sigma^2$ is negligible compared to the other terms, and that the state dependence of the diffusivity can be practically ignored.

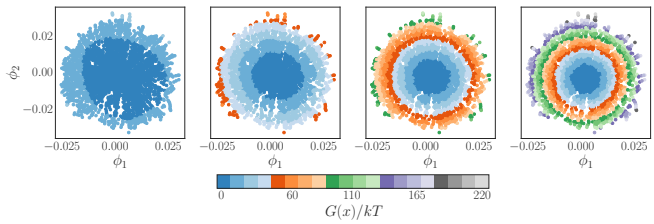

Fig. 10 The Free Energy Landscape, $G(x)/kT$, estimated by Equation (24) for different voltages $V^\star = \{0.5, 0.6, 0.7, 0.8\}$ (from left to right) are plotted as functions in the Diffusion Maps coordinates (ϕ_1, ϕ_2) respectively.

3.5 Experimental Data

In this section we provide a qualitative comparison between our reduced model and experimental dynamic data. The experimental set up from which the data were collected is described in[8]. Note that each configuration used for the experimental data has 204 particles and not 210 as in our simulations. The radii of the particles is the same as the one used for the simulation and the voltage ($V^\star = 0.74$) for those experiments is in the range of the voltages used to train the parameter dependent eSDE. Given the experimental trajectories, we used the same preprocessing as for the computational data, and then Nyström Extension was used to restrict the experimental configurations in the Diffusion Maps coordinates. The experimental data were rescaled in the same range as the simulations based on the ratio of the radii of the two reference configurations used for the Kabsch algorithm. Please note that to restrict the configurations with 204 particles in the Diffusion Maps coordinates obtained from the simulations, a different reference configuration was used for the Kabsch algorithm. The reference configuration was selected also here as the configuration with the smallest value of Rg from the experimental trajectories. Then the same density estimation described in Section 2.2 is applied. These steps allow us to project the experimental particles to the Diffusion Maps coordinates despite their different number of particles. We then use our trained neural network to generate trajectories given the estimated initial conditions in the Diffusion Maps coordinates. The integration of the eSDE was performed for 125 seconds with time step $h = 0.125$. The time step used to integrate the eSDE corresponds to the same frame rate that the experimental measurements were sampled at (8 frames per second[8]). The behavior of the restricted experimental trajectories has the same qualitative behavior as the reduced model, and as the restricted trajectories of the Brownian Dynamics.

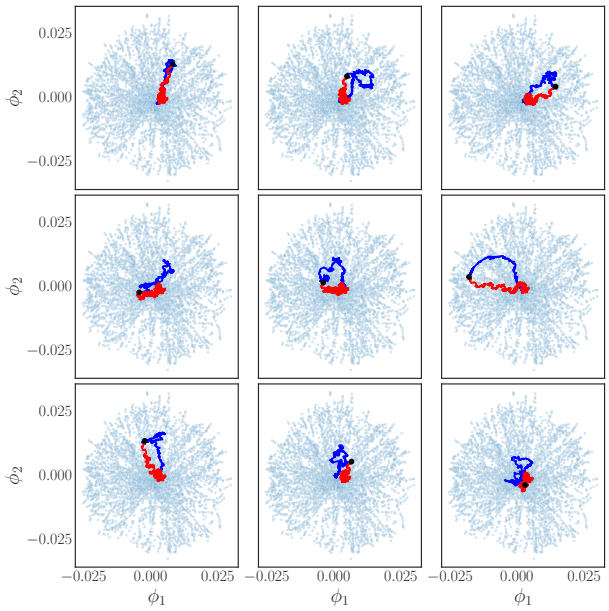

Fig. 11 Restricted with Nyström Extension experimental trajectories compared with paths generated from the neural network eSDE trained on the computational data. The red trajectories correspond to the paths generated by the neural network and the blue trajectories to the trajectories of the experiments, restricted to the latent space using Nyström Extension.

4 Discussion

We demonstrated that the Diffusion Maps algorithm can discover a set of latent observables given a data set of sampled dynamic configurations of crystallizing colloidal particles. We explored the correspondence between our obtained latent observables and established theoretical order parameters (Rg, ψ_6, C_6). We learned an eSDE by using the traditional Kramers-Moyal expansion and compared it with a modern deep learning architecture based on stochastic numerical integrators[14]. Both estimated reduced eS-

DEs qualitatively reproduce the dynamics of the full simulations. We showed that the neural network's dynamics on average appear more accurate, by comparing the data-driven eSDEs with restricted trajectories of the full Brownian Dynamics. It's worth mentioning that the computation cost and the number of data points needed to learn an effective eSDE with the neural network is much smaller than the corresponding Kramers-Moyal effort, we provide a more detailed comparison in the Appendix. We illustrated the ability to learn a parameter dependent eSDE through our neural network architecture. The coefficients (drift and diffusivity) of the parameter dependent eSDE again seem to capture the dynamics of the fine scale simulations. Lastly, we showed that our reduced models qualitatively agree with dynamics of restricted experimental data.

5 Conclusions

The developed eSDEs provides a compressed data-driven model that we believe can help the study of self-assembly. Even though the application was focused on colloidal assembly, this framework can be applied to a range of different applications, from coarse-graining epidemiological models to models of cell motility. Such data-driven models could be useful tools for performing scientific computations (e.g. estimation of mean escape times, construction of bifurcation diagrams) even when analytical expressions are not available.

Our reduced models, while capable of describing the coarse-grained, collective dynamics, do not provide information about the fine-scale conformations themselves. Our assumption that differences in density profiles suffice to determine a similarity measure in configuration space leads to configurations with the same density field being mapped to a single point in our coarse latent space. Therefore, mapping back to the ambient space, i.e. *lifting*, is a nontrivial task since there is a *family* of configurations for each Diffusion Maps point. To support this argument we show a comparison between a *naive* mapping of a generated trajectory from the Diffusion Maps coordinates to the configurations with nearest neighbors (what we call *lifting*, from coarse to fine) and a trajectory generated by the Brownian Dynamic simulation in Figure 12 and in the accompanying video (provided in the SI). Both trajectories start from the same initial condition. In Figure 12 for a reduced trajectory of the eSDE estimated by the neural network we find the nearest point in the Diffusion Maps coordinate that belongs to our data set and *lift* based on that configuration. The *lifted* trajectory exhibits large abrupt changes and appears thus unrealistic. We believe that utilizing conditional Generative Adversarial Networks (cGANs) constitutes a promising direction for reconstructing realistic fine scale configurations conditioned on coarse-grained features.

Learning eSDEs directly from experimental data is a possible extension of our work. The main limitation of learning an eSDE directly is that usually we do not have sufficient experimental data; this is why BD models are matched (as well as possible) to experiments, and then we analyze their simulations [8–10]. Perhaps a transfer learning approach, where the eSDE is initially trained in a large computational data set, and then refined/adapted to experimental data could be an interesting approach for the construction of data-driven models for studying self-assembly.

Another possible extension of our current work deals with using the identified eSDE for control problems. Merging the parameter dependent eSDE with feedback control policies could guide the evolution of configurations from polycrystalline states to target single-domain crystals [23?]

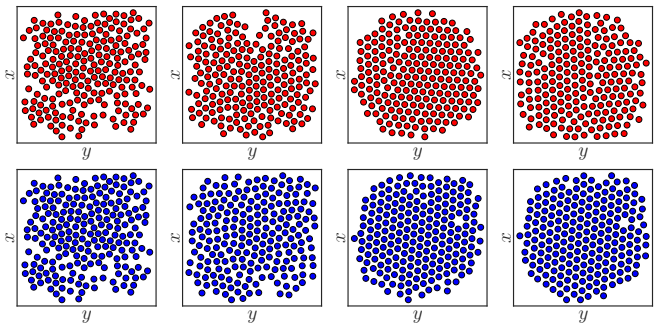

Fig. 12 The first row illustrates snapshots of the colloidal particles at different time instances ($t = 0\text{s}, 12.5\text{s}, 62.5\text{s}, 125\text{s}$). Those snapshots generated by mapping a trajectory integrated by the eSDE to the original physical coordinates with the nearest neighbor algorithm (lifting). The second row illustrates snapshots of colloidal particles for the same timestamps computed with Brownian Dynamics.

Author Contributions

N.E. performed the data curation, the formal analysis of the data and conducted the experiments (simulations and learning the eSDEs). F.D, provided the codes for the neural networks made the modifications for the architecture to include a parameter and provided guidance for the training and evaluation of the neural network models. J.M.B-R, provided theoretical and practical insight for the computations of the effective potentials. A.Y, M.A.B provided the Brownian Dynamic Simulation codes and codes for the computation of the Order Parameters. R.S, M.A.B analyzed and provided data for the experimental paths. I.G.K. planned and coordinated the work and was the PI in funding the effort. N.E, F.D, and I.G.K wrote the manuscript. All authors contributed in manuscript's review and editing.

Conflicts of interest

There are no conflicts to declare.

Acknowledgements

This work was partially supported by the US AFOSR and the US Department of Energy. We acknowledge financial support by the National Science Foundation CBET-1928950.

Notes and references

1 K. Krischer, R. Rico-Martínez, I. Kevrekidis, H. Rotermund, G. Ertl and J. Hudson, *AIChE Journal*, 1993, **39**, 89–98.

2 R. Rico-Martinez, K. Krischer, I. Kevrekidis, M. Kube and J. Hudson, *Chemical Engineering Communications*, 1992, **118**, 25–48.

3 R. Rico-Martinez, I. Kevrekidis, M. Kube and J. Hudson, 1993 American Control Conference, 1993, pp. 1475–1479.

4 R. González-García, R. Rico-Martínez and I. Kevrekidis, *Computers & Chemical Engineering*, 1998, **22**, S965–S968.

5 P. Liu, C. Siettos, C. W. Gear and I. Kevrekidis, *Mathematical Modelling of Natural Phenomena*, 2015, **10**, 71–90.

6 E. Chiavazzo, R. Covino, R. R. Coifman, C. W. Gear, A. S. Georgiou, G. Hummer and I. G. Kevrekidis, *Proceedings of the National Academy of Sciences*, 2017, **114**, E5494–E5503.

7 Y. Yang, R. Thyagarajan, D. M. Ford and M. A. Bevan, *The Journal of chemical physics*, 2016, **144**, 204904.

8 T. D. Edwards, D. J. Beltran-Villegas and M. A. Bevan, *Soft Matter*, 2013, **9**, 9208–9218.

9 J. J. Juárez and M. A. Bevan, *The Journal of chemical physics*, 2009, **131**, 134704.

10 J. J. Juarez, J.-Q. Cui, B. G. Liu and M. A. Bevan, *Langmuir*, 2011, **27**, 9211–9218.

11 R. R. Coifman and S. Lafon, *Applied and Computational Harmonic Analysis*, 2006, **21**, 5 – 30.

12 J. Gradišek, S. Siegert, R. Friedrich and I. Grabec, *Physical Review E*, 2000, **62**, 3146.

13 H. Risken, *The Fokker-Planck Equation*, Springer, 1996, pp. 63–95.

14 F. Dietrich, A. Makeev, G. Kevrekidis, N. Evangelou, T. Bertalan, S. Reich and I. G. Kevrekidis, *Learning effective stochastic differential equations from microscopic simulations: combining stochastic numerics and deep learning*, 2021.

15 K. A. Arpin, A. Mihi, H. T. Johnson, A. J. Baca, J. A. Rogers, J. A. Lewis and P. V. Braun, *Advanced Materials*, 2010, **22**, 1084–1101.

16 W. B. Russel, *MRS Online Proceedings Library*, 1989, **177**, 281–290.

17 C. Zukoski, *Chemical engineering science*, 1995, **50**, 4073–4079.

18 R. S. Hendley, I. Torres-Díaz and M. A. Bevan, *Soft Matter*, 2021, **17**, 9066–9077.

19 J. J. Juárez and M. A. Bevan, *Advanced Functional Materials*, 2012, **22**, 3833–3839.

20 T. D. Edwards, Y. Yang, D. J. Beltran-Villegas and M. A. Bevan, *Scientific reports*, 2014, **4**, 1–8.

21 X. Tang, B. Rupp, Y. Yang, T. D. Edwards, M. A. Grover and M. A. Bevan, *ACS nano*, 2016, **10**, 6791–6798.

22 X. Tang, M. A. Bevan and M. A. Grover, *Molecular Systems Design & Engineering*, 2017, **2**, 78–88.

23 J. Zhang, J. Yang, Y. Zhang and M. A. Bevan, *Science advances*, 2020, **6**, eabd6716.

24 X. Tang, J. Zhang, M. A. Bevan and M. A. Grover, *Journal of Process Control*, 2017, **60**, 141–151.

25 D. I. Kopelevich, A. Z. Panagiotopoulos and I. G. Kevrekidis, *The Journal of chemical physics*, 2005, **122**, 044908.

26 D. J. Beltran-Villegas, R. M. Sehgal, D. Maroudas, D. M. Ford and M. A. Bevan, *The Journal of chemical physics*, 2011, **135**, 154506.

27 A. C. Coughlan, I. Torres-Díaz, J. Zhang and M. A. Bevan, *The Journal of Chemical Physics*, 2019, **150**, 204902.

28 H. Sidky, W. Chen and A. L. Ferguson, *Molecular Physics*, 2020, **118**, e1737742.

29 P. R. Vlachas, J. Zavadlav, M. Praprotnik and P. Koumoutsakos, *Journal of Chemical Theory and Computation*, 2021, **18**, 538–549.

30 X. Fu, T. Xie, N. J. Rebello, B. D. Olsen and T. Jaakkola, *arXiv preprint arXiv:2204.10348*, 2022.

31 K. Pearson, *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 1901, **2**, 559–572.

32 J. B. Tenenbaum, V. De Silva and J. C. Langford, *science*, 2000, **290**, 2319–2323.

33 S. T. Roweis and L. K. Saul, *science*, 2000, **290**, 2323–2326.

34 M. Belkin and P. Niyogi, *Neural computation*, 2003, **15**, 1373–1396.

35 M. A. Kramer, *AIChE journal*, 1991, **37**, 233–243.

36 R. Talmon, I. Cohen, S. Gannot and R. R. Coifman, *IEEE signal processing magazine*, 2013, **30**, 75–86.

37 K. Rajendran, A. Kattis, A. Holiday, R. Kondor and I. G. Kevrekidis, International Conference on Patterns of Dynamics, 2016, pp. 289–317.

38 B. Nadler, S. Lafon, R. R. Coifman and I. G. Kevrekidis, *Applied and Computational Harmonic Analysis*, 2006, **21**, 113–127.

39 E. Chiavazzo, C. W. Gear, C. J. Dsilva, N. Rabin and I. G. Kevrekidis, *Processes*, 2014, **2**, 112–140.

40 Y. Lu, Y. Li and J. Duan, *Extracting Stochastic Governing Laws by Nonlocal Kramers-Moyal Formulas*, 2021.

41 Y. Li and J. Duan, *Physica D: Nonlinear Phenomena*, 2021, **417**, 132830.

42 J. O'Leary, J. A. Paulson and A. Mesbah, *Stochastic Physics-Informed Neural Networks (SPINN): A Moment-Matching Framework for Learning Hidden Physics within Stochastic Differential Equations*, 2021.

43 L. Boninsegna, F. Nüske and C. Clementi, *The Journal of chemical physics*, 2018, **148**, 241723.

44 L. Yang, C. Daskalakis and G. E. Karniadakis, *arXiv e-prints*, 2020, arXiv–2008.

45 X. Li, T.-K. L. Wong, R. T. Q. Chen and D. Duvenaud, International Conference on Artificial Intelligence and Statistics, 2020, 2020.

46 A. Hasan, J. M. Pereira, S. Farsiu and V. Tarokh, *Identifying Latent Stochastic Differential Equations*, 2021.

47 C. J. Dsilva, R. Talmon, R. R. Coifman and I. G. Kevrekidis, *Parsimonious Representation of Nonlinear Dynamical Systems Through Manifold Learning: A Chemotaxis Case Study*, 2015.

48 W. Kabsch, *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 1976, **32**, 922–923.

49 W. Kabsch, *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 1978, **34**, 827–828.

50 D. W. Scott, *Multivariate density estimation: theory, practice, and visualization*, John Wiley & Sons, 2015.

51 D. Lehmberg, F. Dietrich, G. Köster and H.-J. Bungartz, *Journal of Open Source Software*, 2020, **5**, 2283.

52 E. Nyström, *Commentationes Physico Mathematicae*, 1928,

(4), 1–52.
53 C. Williams and M. Seeger, Advances in Neural Information Processing Systems 13, 2001, pp. 682–688.
54 P. J. Steinhardt, D. R. Nelson and M. Ronchetti, *Phys. Rev. B*, 1983, **28**, 784–805.
55 C. W. Rowley, I. G. Kevrekidis, J. E. Marsden and K. Lust, *Nonlinearity*, 2003, **16**, 1257.
56 B. Sonday, A. Singer and I. G. Kevrekidis, *Computers & Mathematics with Applications*, 2013, **65**, 1535–1557.
57 A. Singer and H.-T. Wu, *Communications on pure and applied mathematics*, 2012, **65**, 1067–1144.