LEARNING EFFECTIVE SDEs FROM BROWNIAN DYNAMICS SIMULATIONS OF COLLOIDAL PARTICLES

A PREPRINT

Nikolaos Evangelou
Johns Hopkins University
Baltimore, MD 21218, USA

Felix Dietrich
Technical University ofMunich
Munich, 80333, Germany

Juan M. Bello-Rivas
Johns Hopkins University
Baltimore, MD 21218, USA

Alex Yeh
Johns Hopkins University
Baltimore, MD 21218, USA

Rachel Stein
Johns Hopkins University
Baltimore, MD 21218, USA

Michael A. Bevan
Johns Hopkins University
Baltimore, MD 21218, USA

Ioannis G. Kevrekidis*
Johns Hopkins University
Baltimore, MD 21218, USA
yannisk@jhu.edu

ABSTRACT

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 [25, 37, 36]. 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 [14, 37] (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 [29, 5].

In this paper, we identify coarse-grained, effective stochastic differential equations (eSDE) for colloidal particle self-assembly based onfine-grained, Brownian dynamics simulations under the influence of electric fields [51, 11]. 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 [11, 18, 20]. Figure 1 shows a sample path of a latent space trajectory $\{t, \varphi(t)\}_{t\geq 0}$ 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 [7]. 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. [15, 38], but rather (in the spirit of the early work mentioned above) on numerical stochastic integration algorithms [9].

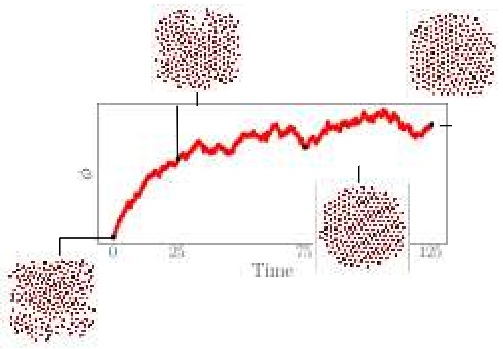

Figure 1: A trajectory of an effective, reduced eSDE in the data driven collective coordinate φ for electric-field mediated colloidal crystallization.

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 basis for emerging technologies (e.g. photonic crystals, meta-materials, cloaking devices, solar cells, etc.), [1] but also impact traditional applications (e.g. ceramics, coatings, minerals, foods, drugs [40],[53]). Despite the range of applications employing microscopic colloidal particles, current state-of-the-art [17] 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. By learning effective collective models, our approach in this work paves the way towards data-driven model-based control of the kinetics of colloidal assembly processes using parametric driving (e.g. via an electric field) [19]. 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 [46, 45, 52]. 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 [23] identified an effective, coarse grained Fokker-Planck using Kramers-Moyal with an application to micelle-formation of surfactant molecules. The identified equation in [23] was constructed in terms of the physical coarse variable, size of the cluster of the surfactant molecules. The authors in [3] 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 [8] 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 [42]. A framework to advance the simulation time by learning the effective dynamics (LED) of molecular systems was proposed by the authors in [48]. 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 [13] 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 [34], Isomap [47], Local Linear Embedding [39], Laplacian Eigenmaps [2], Autoencoders [24] and our method of choice: Diffusion Maps [7]. The Diffusion Maps algorithms enables the discovery of reduced coordinates when data are sampled from signal processing [44], from networks [35], from (stochastic) differential equations [31, 6] but also from Molecular [5] and Brownian simulations [51].

A traditional approach for learning eSDEs has been the Kramers-Moyal expansion [15, 38, 29] 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 [30, 28]. In [33] 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 [4] proposed an extension of the framework called Sparse Identification of Nonlinear Dynamics (SINDy) that can be used for stochastic dynamical systems. The authors in [50] 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 [27] 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 [16]. 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 [9] to allow for additional parameter dependence. Our latent space is defined through Laplace-Beltrami operator eigenfunctions, so, different from [16], (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.

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

In Equation (1) r_{ij} denotes the center-to-center distance between the patricles, a is the radius of each particle and $B^{p\ P}$ 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

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

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 $\lambda = \frac{\pi E_m a^3 (f_{cm}E_0)^2}{kT}$, E_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

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

$P_2(cos\theta_{ij})$ 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

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

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 $\langle \mathbf{F}^B \rangle = 0$, $\langle \mathbf{F}^B(t_1)(\mathbf{F}^B(t_2))^{\mathrm{T}} \rangle = 2(kT)^2(\mathbf{D}^P)^{-1}\delta(t_1 - t_2)$, $\mathbf{r}(t)$ 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

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

$\mathbf{D}^P$ 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.

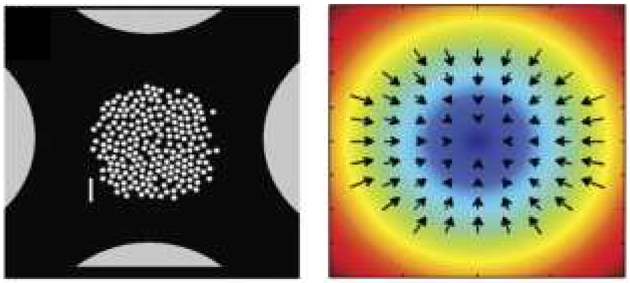

Figure 2: [Left] Top view of simulated experiments of quasi 2D configurations of $N = 210$ colloidal particles compressed with a quadrupole electrode. [Right] Electric field magnitude contour plot in the vicinity of the quadrupole electrode center. The arrows indicate the relative magnitude and direction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

2.2 Diffusion Maps

Introduced by [7], Diffusion Maps offer a parametrization of a data set of points $\mathbf{X} = \{\boldsymbol{x}_i\}_i^N$ sampled from a manifold $\mathcal{M}$, where $\boldsymbol{x}_i \in \mathbb{R}^m$ 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

$$A_{ij} = \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\varepsilon}\right), \tag{10}$$

where $\|\cdot\|$ denotes a norm of choice. In this work we choose the l^2 norm; E 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

$$W(\boldsymbol{x}_i, \boldsymbol{x}_j) = \frac{\tilde{A}(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\sum_{j=1}^{N} \tilde{A}(\boldsymbol{x}_i, \boldsymbol{x}_j)} \tag{13}$$

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

$$\mathbf{W}\boldsymbol{\varphi}_i = \lambda_i \boldsymbol{\varphi}_i. \tag{14}$$

To check if dimensionality reduction of $\mathbf{X}$ is possible, selection of the eigenvectors $\boldsymbol{\varphi}$ that parameterize independent directions (non-harmonic eignvectors) is needed. In our work this selection was made by implementing the algorithm presented in [10]. 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", $\boldsymbol{x}_i$, consist of planar configurations of 210 particle locations obtained either from evolving computations or from experimental movies; our data set is $\mathbf{X} = \{\boldsymbol{x}_i\}_{i=1}^{N}$ where $\boldsymbol{x}_i \in \mathbb{R}^{210\times 2}$. 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 [21, 22]. 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 [41]. 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 [10]. For our Diffusion Maps computations the *datafold* package was used [26].

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 [32, 49] can be used

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

where $\varphi_i(\boldsymbol{f}_{\text{new}})$ is the estimated value of the i^{th} eigenvector for the new point $\boldsymbol{f}_{\text{new}}$, λ_i is the corresponding eigenvalue, and $\varphi_i(f_j)$ 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 $\boldsymbol{x}(t)$ be a stochastic vector-valued variable whose evolution is governed by the SDE

$$d\boldsymbol{x}(t) = \boldsymbol{v}(\boldsymbol{x}(t))\,dt + \sigma(\boldsymbol{x}(t))\,dB_t, \tag{16}$$

where $\boldsymbol{v} : \mathbb{R}^m \to \mathbb{R}^m$ 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 $\boldsymbol{v}$ and σ. We show how this estimation can be performed, either from the statistical definition of the terms, based on the Kramers- Moyal expansion [38, 15], or via a deep learning architecture inspired by stochastic numerical integrators [9].

2.4.1 Kramers-Moyal expansion

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

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

which is known as the Kramers-Moyal expansion [38]. 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

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

where denotes the average. When $\boldsymbol{x}(t)$ 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

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

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 $\boldsymbol{v}(x) = \mathbf{D}^{(1)}, \boldsymbol{\sigma}^2 = 2\mathbf{D}^{(2)}$. The estimation of the drift and the diffusivity at a point $\boldsymbol{x}^i$ can be performed by multiple local parallel simulations ("bursts").

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

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 [9]. In this approach, the drift and diffusivity are estimated through two networks v_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 $\mathcal{D} = \{\boldsymbol{x}^i(t_{k+h}), \boldsymbol{x}^i(t_k), h^i, p^i\}_{i=1}^{N}$ 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 $\mathcal{D}$ 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 $\mathcal{D}_i$ in this network includes (a) a point at time k in space $\boldsymbol{x}^i(t_k) = (x_1^i(t_k), x_2^i(t_k))$; (b) its coordinates after a short time evolution $\boldsymbol{x}^i(t_{k+h}) = (x_1^i(t_{k+h}), x_2^i(t_{k+h}))$; (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 [9]) is derived from the Euler-Maruyama scheme, a numerical integration method for SDEs. The scheme for the two-dimensional case,

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

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 $\boldsymbol{x}^i(t_{k+h})$ is normally distributed,

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

where the mean $\boldsymbol{\mu}_\theta^i = \boldsymbol{x}^i(t_k) + h^i \boldsymbol{v}_\theta(\boldsymbol{x}^i(t_k), \boldsymbol{p}^i)$ and the covariance matrix $\boldsymbol{\Sigma}_\theta^i = h^i \sigma_\theta(\boldsymbol{x}^i(t_k), \boldsymbol{p}^i)^2$.

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):

$$\mathsf{L}(\theta|\boldsymbol{x}^i(t_{k+h}),\boldsymbol{x}^i(t_k),h^i,\boldsymbol{p}^i) \quad := \quad \log\ \det(\boldsymbol{\Sigma}^i_\theta) \ + \ \frac{1}{2}(\boldsymbol{x}^i(t_{k+h}) \ - \ \boldsymbol{\mu}^i_\theta)^{\mathrm{T}}(\boldsymbol{\Sigma}^i_\theta)^{-1}(\boldsymbol{x}^i(t_{k+h}) \ - \ \boldsymbol{\mu}^i_\theta) \tag{23}$$

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 L over the training set $\mathbf{D}$.

2.5 Order Parameters - Free Energy Landscapes

Order parameters are coarse, collective variables that summarize the physics involved in the colloidal self-assembly process [43]. These quantities often encapsulate features of interest, for example the compactness, or the local or global degree of order of a particle assembly [43, 45]. 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 [45]. Order parameters that are typically used to study colloidal self-assembly are Rg, ψ_6 and C_6 [12]. 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. [12]. 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 $Rg,$ $\psi_6,$ C_6, see Section 3.1.

The effective potential $G(\boldsymbol{x})$ quantifies the free energy landscape. It is obtained from the equilibrium probability distribution, the steady-state solution of the Fokker Planck equation [38]. 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 [38]

$$G(\boldsymbol{x}^i) = -kT \int_0^{\boldsymbol{x}^i} 2(\sigma^2)^{-1}(\boldsymbol{v} - \nabla \cdot \frac{\sigma^2}{2}) \cdot d\boldsymbol{r}. \tag{24}$$

where $\boldsymbol{v}$ 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 [10]. 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 [10]. 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.

3.2 Learning Effective SDEs

For Brownian Dynamics simulations at fixed normalized voltage $V^* = \frac{V}{V_{xtal}} = \frac{1.51\mathrm{V}}{1.89\mathrm{V}} = 0.8$ (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.

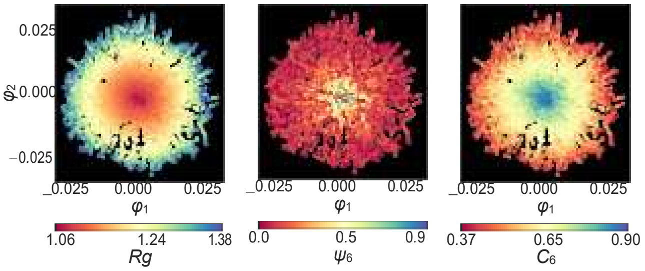

Figure 3: The two leading Diffusion Maps coordinates (φ_1, φ_2)
colored by the physically meaningful order parameters Rg, ψ_6 and, C_6 respectively (Rg clearly correlates with C_6).

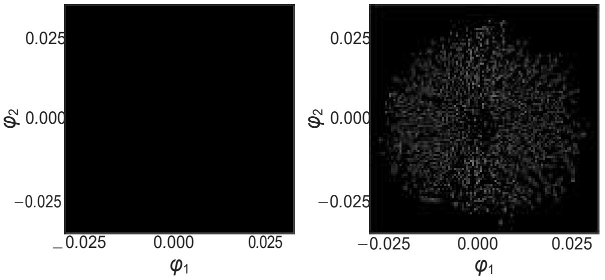

Figure 4: The estimated drift from the neural network and the Kramers-Moyal respectively is plotted as the on average vector field in the two Diffusion Maps coordinates (φ_1, φ_2). The vector field is plotted in subsampled data sets to improve visualization.

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 approaches leads to similar conclu-

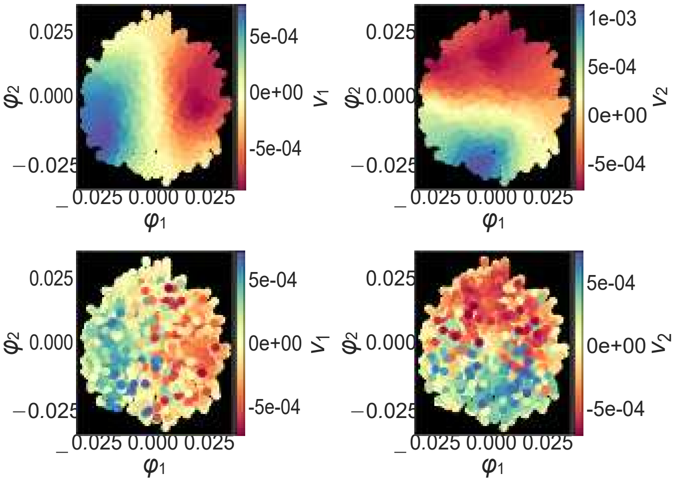

Figure 5: The estimated drift, υ_1, υ_2, from the neural network (first row) and the Kramers-Moyal, $\upsilon_1^*, \upsilon_2^*$, (second row)

is plotted as a function of the Diffusion Maps coordinates (φ_1, φ_2)

.

sions, 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.

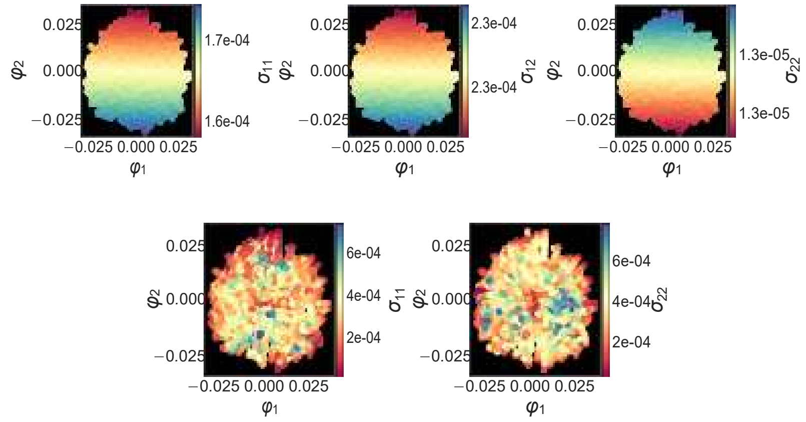

Figure 6: The estimated diffusivity from the neural network, $\sigma_{11}, \sigma_{22}, \sigma_{12} = \sigma_{21}$ (first row) and the one from Kramers-Moyal, $\sigma^*_{11}, \sigma^*_{22}$ (second row) is plotted as a function of the Diffusion Maps coordinates (φ_1, φ_2).

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 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 [9]).

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, $\mathbf{V}^* = \{0.5, 0.6, 0.7, 0.8\}$. 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.

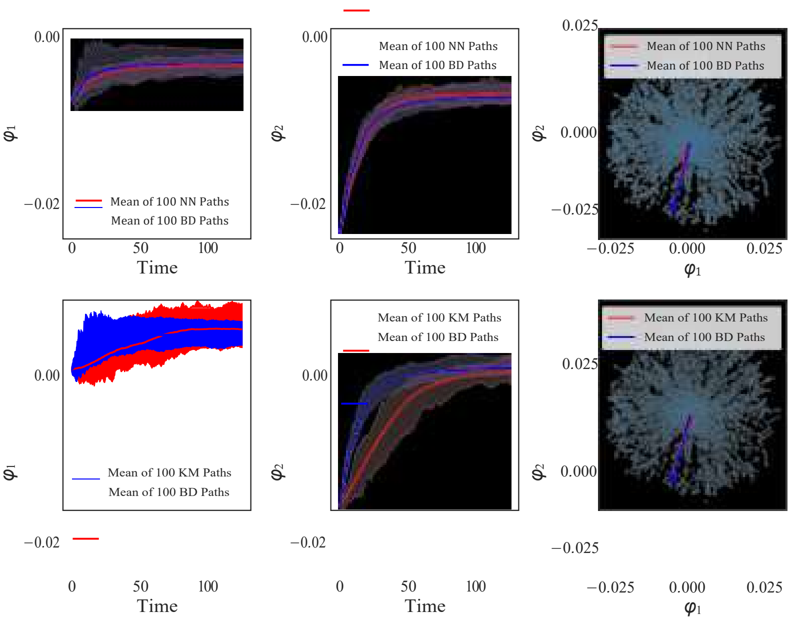

Figure 7: The estimated dynamics from the neural network (first row) and the Kramers-Moyal (second row) is shown compared to restricted (with Nyström) trajectories of the Brownian Dynamics in the two Diffusion Maps coordinates (φ_1, φ_2). The mean of 100 trajectories (starting from the same initial condition) 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.

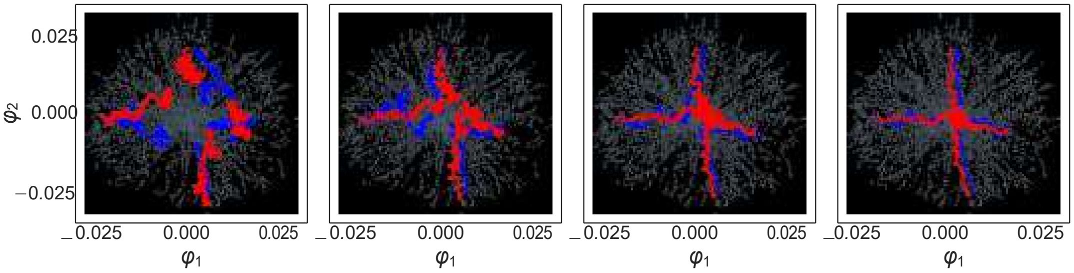

Figure 8: In the Diffusion Maps coordinates (φ_1, φ_2) we illustrate trajectories computed through the neural network trained for different values of the voltage, $V^* = 0.5, \{0.6, 0.7, 0.8\}$ plotted from left to right. Trajectories of the estimated eSDE from the neural network (red paths) are contrasted 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^*, for which the drift is also smaller, larger time steps could be accurately employed.

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

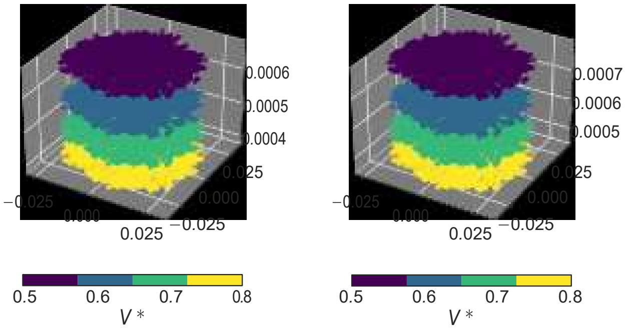

Figure 9: The estimated diagonal diffusivity is plotted against φ_1, φ_2. Different colors correspond to different values of the parameter ($V*$). The trend in the estimated diffusivity is in agreement with the physics of the problem. The higher voltage forces the system to condense faster. On the contrary,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

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.

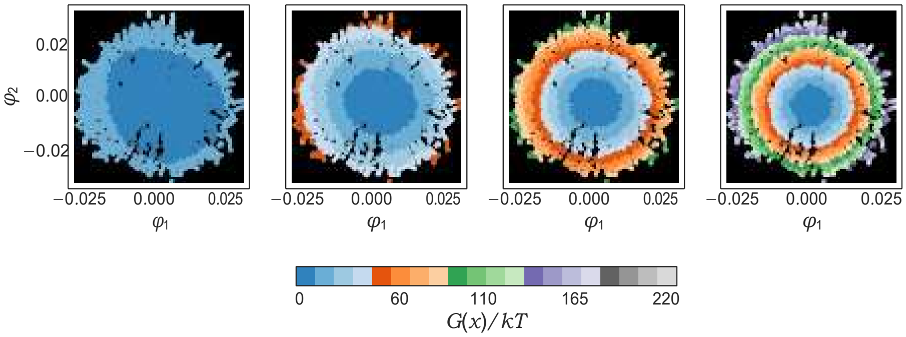

Figure 10: The Free Energy Landscape, $G(x)/kT$, estimated by Equation (24) for different voltages $V^* = \{0.5, 0.6, 0.7, 0.8\}$ (from left to right) are plotted as functions in the Diffusion Maps coordinates (φ_1, φ_2) respectively.

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 [11]. 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 ($V^* = 0.74$) 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 [11]). 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.

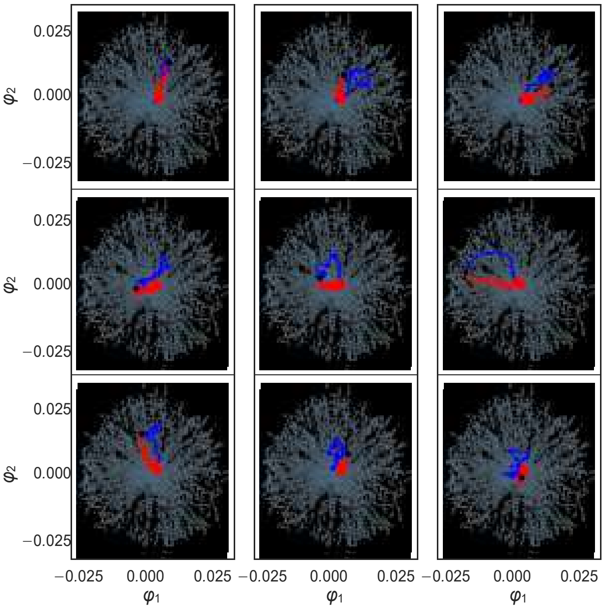

Figure 11: Restricted with Nyström Extension experimental trajectories compared with paths generated from the neural network eSDE trained on the computational data. The red trajectories correspond to the paths generated by the neural network and the blue trajectories to the trajectories of the experiments, 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 [9]. Both estimated reduced eSDEs 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[11, 18, 20]. 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. Merg- ing the parameter dependent eSDE with feedback control policies could guide the evolution of configurations from polycrystalline states to target single-domain crystals [52, 46].

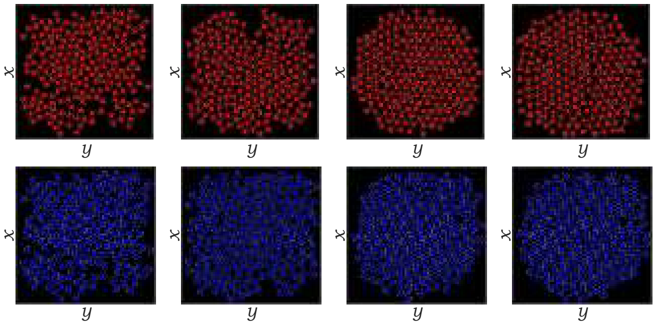

Figure 12: The first row illustrates snapshots of the colloidal particles at different time instances (t = 0s, 12.5s, 62.5s, 125s). Those snapshots generated by mapping a trajectory integrated by the eSDE to the original physical coordinates with the nearest neighbor algorithm (lifting).
The second row 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.

References

[1] Kevin A Arpin, Agustin Mihi, Harley T Johnson, Alfred J Baca, John A Rogers, Jennifer A Lewis, and Paul V Braun. Multidimensional architectures for functional optical devices. *Advanced Materials*, 22(10):1084–1101,

2010.

[2] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.

[3] Daniel J Beltran-Villegas, Ray M Sehgal, Dimitrios Maroudas, David M Ford, and Michael A Bevan. A smoluchowski model of crystallization dynamics of small colloidal clusters. *The Journal of chemical physics*, 135(15):154506, 2011.

[4] Lorenzo Boninsegna, Feliks Nüske, and Cecilia Clementi. Sparse learning of stochastic dynamical equations. *The Journal of chemical physics*, 148(24):241723, 2018.

[5] Eliodoro Chiavazzo, Roberto Covino, Ronald R Coifman, C William Gear, Anastasia S Georgiou, Gerhard Hummer, and Ioannis G Kevrekidis. Intrinsic map dynamics exploration for uncharted effective free-energy landscapes. *Proceedings of the National Academy of Sciences*, 114(28):E5494–E5503, 2017.

[6] Eliodoro Chiavazzo, Charles W. Gear, Carmeline J. Dsilva, Neta Rabin, and Ioannis G. Kevrekidis. Reduced models in chemical kinetics via nonlinear data-mining. *Processes*, 2(1):112–140, 2014.

[7] Ronald R. Coifman and Stéphane Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5 – 30, 2006. Special Issue: Diffusion Maps and Wavelets.

[8] Anna CH Coughlan, Isaac Torres-D´ıaz, Jianli Zhang, and Michael A Bevan. Non-equilibrium steady-state colloidal assembly dynamics. *The Journal of Chemical Physics*, 150(20):204902, 2019.

[9] Felix Dietrich, Alexei Makeev, George Kevrekidis, Nikolaos Evangelou, Tom Bertalan, Sebastian Reich, and Ioannis G. Kevrekidis. Learning effective stochastic differential equations from microscopic simulations: combining stochastic numerics and deep learning, 2021.

[10] Carmeline J. Dsilva, Ronen Talmon, Ronald R. Coifman, and Ioannis G. Kevrekidis. Parsimonious representation of nonlinear dynamical systems through manifold learning: A chemotaxis case study, 2015.

[11] Tara D Edwards, Daniel J Beltran-Villegas, and Michael A Bevan. Size dependent thermodynamics and kinetics in electric field mediated colloidal crystal assembly. *Soft Matter*, 9(38):9208–9218, 2013.

[12] Tara D Edwards, Yuguang Yang, Daniel J Beltran-Villegas, and Michael A Bevan. Colloidal crystal grain boundary formation and motion. *Scientific reports*, 4(1):1–8, 2014.

[13] Xiang Fu, Tian Xie, Nathan J Rebello, Bradley D Olsen, and Tommi Jaakkola. Simulate time-integrated coarse-grained molecular dynamics with geometric machine learning. *arXiv preprint arXiv:2204.10348*, 2022.

[14] R. González-García, R. Rico-Martínez, and I.G. Kevrekidis. Identification of distributed parameter systems: A neural net based approach. *Computers & Chemical Engineering*, 22:S965–S968, 1998. European Symposium on Computer Aided Process Engineering-8.

[15] Janez Gradisˇek, Silke Siegert, Rudolf Friedrich, and Igor Grabec. Analysis of time series from stochastic processes. *Physical Review E*, 62(3):3146, 2000.

[16] Ali Hasan, João M. Pereira, Sina Farsiu, and Vahid Tarokh. Identifying latent stochastic differential equations, 2021.

[17] Rachel S Hendley, Isaac Torres-D´ıaz, and Michael A Bevan. Anisotropic colloidal interactions & assembly in ac electric fields. *Soft Matter*, 17(40):9066–9077, 2021.

[18] Jaime J Juárez and Michael A Bevan. Interactions and microstructures in electric field mediated colloidal assembly. *The Journal of chemical physics*, 131(13):134704, 2009.

[19] Jaime J Juárez and Michael A Bevan. Feedback controlled colloidal self-assembly. *Advanced Functional Materials*, 22(18):3833–3839, 2012.

[20] Jaime J Juarez, Jing-Qin Cui, Brian G Liu, and Michael A Bevan. kt-scale colloidal interactions in high frequency inhomogeneous ac electric fields. i. single particles. *Langmuir*, 27(15):9211–9218, 2011.

[21] Wolfgang Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 32(5):922–923, 1976.

[22] Wolfgang Kabsch. A discussion of the solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 34(5):827–828, 1978.

[23] Dmitry I Kopelevich, Athanassios Z Panagiotopoulos, and Ioannis G Kevrekidis. Coarse-grained kinetic computations for rare events: Application to micelle formation. *The Journal of chemical physics*, 122(4):044908, 2005.

[24] Mark A Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE journal*, 37(2):233–243, 1991.

[25] K Krischer, R Rico-Mart´ınez, IG Kevrekidis, HH Rotermund, G Ertl, and JL Hudson. Model identification of a spatiotemporally varying catalytic reaction. *AIChE Journal*, 39(1):89–98, 1993.

[26] Daniel Lehmberg, Felix Dietrich, Gerta Köster, and Hans-Joachim Bungartz. datafold: data-driven models for point clouds and time series on manifolds. *Journal of Open Source Software*, 5(51):2283, 2020.

[27] Xuechen Li, Ting-Kam Leonard Wong, Ricky T. Q. Chen, and David Duvenaud. Scalable gradients for stochastic differential equations. In *International Conference on Artificial Intelligence and Statistics, 2020*, 2020.

[28] Yang Li and Jinqiao Duan. A data-driven approach for discovering stochastic dynamical systems with non-gaussian lévy noise. *Physica D: Nonlinear Phenomena*, 417:132830, 2021.

[29] Ping Liu, CI Siettos, C William Gear, and IG Kevrekidis. Equation-free model reduction in agent-based computations: Coarse-grained bifurcation and variable-free rare event analysis. *Mathematical Modelling of Natural Phenomena*, 10(3):71–90, 2015.

[30] Yubin Lu, Yang Li, and Jinqiao Duan. Extracting stochastic governing laws by nonlocal kramers-moyal formulas, 2021.

[31] Boaz Nadler, Stéphane Lafon, Ronald R. Coifman, and Ioannis G. Kevrekidis. Diffusion maps, spectral clustering and reaction coordinates of dynamical systems. *Applied and Computational Harmonic Analysis*, 21(1):113–127, 2006. Special Issue: Diffusion Maps and Wavelets.

[32] E.J Nyström. ¨ uber die praktische aufl oesung von linearen integralgleichungen mit anwendugnen ¨ auf randwertaufgaben der potentialtheorie. *Commentationes Physico Mathematicae*, (4):1–52, 1928.

[33] Jared O'Leary, Joel A. Paulson, and Ali Mesbah. Stochastic physics-informed neural networks (spinn): A moment-matching framework for learning hidden physics within stochastic differential equations, 2021.

[34] Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 2(11):559–572, 1901.

[35] Karthikeyan Rajendran, Assimakis Kattis, Alexander Holiday, Risi Kondor, and Ioannis G Kevrekidis. Data mining when each data point is a network. In *International Conference on Patterns of Dynamics*, pages 289–317. Springer, 2016.

[36] R Rico-Martinez, IG Kevrekidis, MC Kube, and JL Hudson. Discrete-vs. continuous-time nonlinear signal processing: Attractors, transitions and parallel implementation issues. In *1993 American Control Conference*, pages 1475–1479. IEEE, 1993.

[37] R Rico-Martinez, K Krischer, IG Kevrekidis, MC Kube, and JL Hudson. Discrete-vs. continuous-time nonlinear signal processing of cu electrodissolution data. *Chemical Engineering Communications*, 118(1):25–48, 1992.

[38] Hannes Risken. Fokker-planck equation. In *The Fokker-Planck Equation*, pages 63–95. Springer, 1996.

[39] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326, 2000.

[40] William B Russel. Formulation and processing of colloidal dispersions. *MRS Online Proceedings Library*, 177(1):281–290, 1989.

[41] David W Scott. *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 2015.

[42] Hythem Sidky, Wei Chen, and Andrew L Ferguson. Machine learning for collective variable discovery and enhanced sampling in biomolecular simulation. *Molecular Physics*, 118(5):e1737742, 2020.

[43] Paul J. Steinhardt, David R. Nelson, and Marco Ronchetti. Bond-orientational order in liquids and glasses. *Phys. Rev. B*, 28:784–805, Jul 1983.

[44] Ronen Talmon, Israel Cohen, Sharon Gannot, and Ronald R Coifman. Diffusion maps for signal processing: A deeper look at manifold-learning techniques based on kernels and graphs. *IEEE signal processing magazine*, 30(4):75–86, 2013.

[45] Xun Tang, Michael A Bevan, and Martha A Grover. The construction and application of markov state models for colloidal self-assembly process control. *Molecular Systems Design & Engineering*, 2(1):78–88, 2017.

[46] Xun Tang, Bradley Rupp, Yuguang Yang, Tara D Edwards, Martha A Grover, and Michael A Bevan. Optimal feedback controlled assembly of perfect crystals. *ACS nano*, 10(7):6791–6798, 2016.

[47] Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

[48] Pantelis R Vlachas, Julija Zavadlav, Matej Praprotnik, and Petros Koumoutsakos. Accelerated simulations of molecular systems through learning of effective dynamics. *Journal of Chemical Theory and Computation*, 18(1):538–549, 2021.

[49] Christopher Williams and Matthias Seeger. Using the Nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems 13*, pages 682–688. MIT Press, 2001.

[50] Liu Yang, Constantinos Daskalakis, and George Em Karniadakis. Generative ensemble-regression: learning stochastic dynamics from discrete particle ensemble observations. *arXiv e-prints*, pages arXiv–2008, 2020.

[51] Yuguang Yang, Raghuram Thyagarajan, David M Ford, and Michael A Bevan. Dynamic colloidal assembly pathways via low dimensional models. *The Journal of chemical physics*, 144(20):204904, 2016.

[52] Jianli Zhang, Junyan Yang, Yuanxing Zhang, and Michael A Bevan. Controlling colloidal crystals via morphing energy landscapes and reinforcement learning. *Science advances*, 6(48):eabd6716, 2020.

[53] CF Zukoski. Particles and suspensions in chemical engineering: Accomplishments and prospects. *Chemical engineering science*, 50(24):4073–4079, 1995.

A Appendix

A.1 Sampling

The data set in which Diffusion Maps was computed was sampled as follows. Brownian Dynamic simulations were performed given a fixed voltage $V^* = 0.8$ for a system of 210 particles. For $\sim$1500 *random* initial configurations the system was integrated for 1000s. Particle configurations were stored every 1s. The order parameters Rg and ψ_6 were computed based on these collected snapshots. The configurations with $Rg > 1.39$ were discarded. Since these dilute states arise from the random initialization of each simulation, the remaining data set contains configurations that cover the fluid, polycrystalline and crystalline space. We further subsampled in Rg, ψ_6 space in order to get a more uniform data set. The subsampled data set in which Diffusion Maps was performed contains $\sim$11,000 configurations.

For the Kramers-Moyal expansion (discussed in Section 2.3.1) further subsampling of the data set used for Diffusion Maps was performed. A data set of 2800 configurations (and subsequently Diffusion Maps coordinates) was used. For each point in this data set 300 *short* Brownian Dynamic simulations were performed.

For the deep learning approach (discussed in Section 2.3.2) the same data set as for the Kramers-Moyal expansion was used. In this case for each data point only a short trajectory was needed in order to compute the snapshots. The same data points were also used to sample snapshots for different values of the parameter. Different time steps (h) were necessary in order to sample properly the snapshots for the different values of the parameters. The fact that the dimensionality of the manifold remained the same in this range of parameter values was checked before learning the parameter dependent eSDE.

The computational time needed to sample data points for the Kramers-Moyal compared to the one for the neural network is $\sim$38 times and the number of data used to compute Kramers-Moyal was 100$\sim$times more than the one used for the neural network. There is no significant difference between the training time needed for the neural network and Kramers-Moyal estimation.

A.2 Parameters for simulations

Table 1: Simulation Parameters of Colloidal Particles in the presence of a Quadrupole Electrode.

Variable	Value
Number of particles, N	210
Particle Size, a (nm),	1400
Temperature, T ($^\circ C$)	20
Clausius-Mossotti factor for 1 MHz AC field, f_{cm}	-0.4667
Debye length, κ^{-1}nm	10
Electrostatic potential prefactor, B^{PP} (kT)	3216.5
Lowest voltage to crystalize system, V_{xtal} (V)	1.89
Applied voltage, V(V)	0.95, 1.13, 1.32, 1.51
Normalized voltage, $V*$	0.5, 0.6, 0.7, 0.8

A.3 Video

In the attached video in the upper left a trajectory that evolves by integrating the eSDE is shown. In the lower left, a trajectory integrated with the Brownian Dynamic Simulations and restricted with Nyström Extension in the Diffusion Maps coordinates is shown. In the upper right the *lifted* with nearest neighbors estimation configurations are shown. In the down left the configuration integrated with the Brownian Dynamic Simulations are shown.

A.4 Codes

All the codes that used to produce the Figures and generate the models are provided in the public repository Gitlab repository here.

A.5 Neural Network Models Architecture

The architecture and the training protocols for our two neural-network models are reported in this Section. For the surrogate model with fixed voltage, 4 layers with 25 neurons were used. The activation function for the drift network

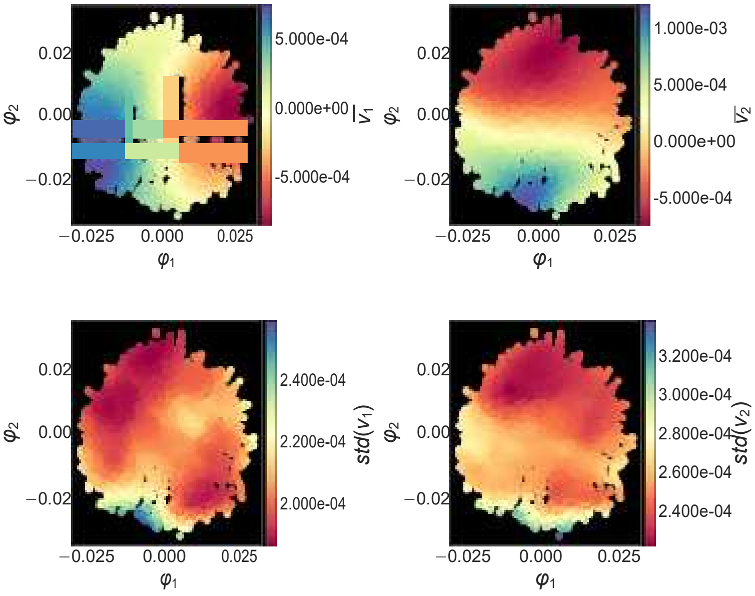

Figure 13: [First row] The average values (over the 200 models) of the drifts $\overline{v_1, v_2}$ are shown as functions in the Diffusion Maps coordinates. [Second Row] The standard deviation of the estimated drifts $std(v_1)$, $std(v_2)$ is shown as functions in the Diffusion Maps coordinates.

was selected as *ReLU* (activation function *ELU* was also tested and gave similar results). The activation function for the diffusivity network was *softplus*. The training of the model was made in two stages: first we use snapshots with h = 1.0 and train for 200 epochs with batch size equal to 32. This first stage approximates reasonably well the drift, but gives a much larger diffusivity. To decrease the diffusivity estimate we fix the weights for the drift network to *non-trainable*. We then use snapshots of $h = 0.125$ and train for 1, 000 more epochs.

For the parameter dependent eSDE a network with 5 layers and 26 with *ELU* was trained. This network was initially trained for 1, 000 epochs with batch size equal to 51. We then set the weights for the diffusivity part of the network to *non-trainable* and train for the drift for 5, 000 epochs. We then set the weights for the drift part of the network to *non-trainable* and train for 1, 000 epochs to improve the diffusivity estimations. For each voltage snapshots at different step size h were used.

For both networks outliers were removed (e.g. with *z-score*). This led to a better approximation of the diffusivity. To train each model, the data was split 90|10 (train/validation). As test set we used the individual trajectories reported in the main text.

A.6 Uncertainty Quantification of our Neural-Network Model

We performed the following computations to obtain an estimate of the accuracy and the sensitivity of the learned parameters of the neural network. We trained the neural network model 200 times, each time using different splits of our original data set to training/validation sets. For each of those models in tensorflow we used *validation_split=0.1*. The data were shuffled before training. To ensure that each time the training/validation sets we get are unique, a different seed was used from the *numpy* random generator for each of the 200 models. To alleviate the computational time needed to train these 200 networks we used *job-arrays*. All the other hyperparameters involved in training the model were kept fixed during this procedure. To obtain a quantitative measurement of the sensitivity of our model in terms of the training set, we then evaluated the drift and diffusivity for the entire data set (training and validation) as well as on a grid of test points.

In Figure 13 the first row illustrated the drift components (averaged over the 200 models) colored as a function of the diffusion maps coordinates. The second row depicts the pointwise standard deviation of the 200 models, colored as a function of the Diffusion Maps coordinates. It appears that the overall trend and the order of magnitude between the estimated average value of the drift here, and the one shown in Figure 4 in the main text are consistent. The estimated standard deviation appears almost everywhere smaller than the estimated mean drift.

The same computation was performed for the diffusivity and is shown in Figure 14. The magnitude of the diagonal terms is comparable to the model described in the main text (Figure 5). The average off-diagonal term seems to be an order of magnitude smaller than the one presented in the main paper. The average off-diagonal terms are two orders of magnitudes smaller than the average diagonal terms. The standard deviations for the diffusivities for the diagonal elements are one order of magnitude smaller than the averages; yet for the off-diagonal elements, they are an order of magnitude larger than the average. This latter observation of the off-diagonal elements can be attributed to the fact that some models are having positive and some negative off-diagonal elements. Therefore their mean is closer to zero but their standard deviation is larger than zero.

In Figure 15 we illustrate the same estimations for the 200 models on a 2D grid: we plot the observed values for the mean drift, mean diffusivity based on the 200 models and their standard deviation. The observations that we can draw based on those models are similar to the ones discussed above.

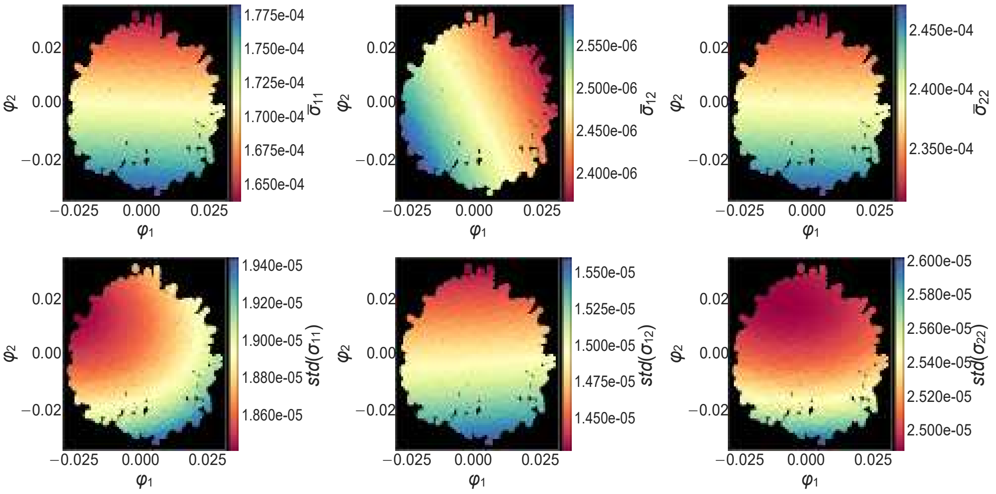

Figure 14: [First row] The average values (over the 200 models) of the estimated diffusivities $\overline{\sigma_1,\sigma_2}$ are shown as functions in the Diffusion Maps coordinates. [Second Row] The standard deviation of the estimated diffusivities $std(\sigma_1)$, $std(\sigma_2)$ is colored as a function in the Diffusion Maps coordinates.

A.7 Kramers-Moyal vs Neural Network Comparison

At the nodes of a Cartesian grid (20x20) we evaluate the drift and diffusivity of the two surrogate models (the Kramers-Moyal and the Neural Network). We then compute the l^2 norm between the estimated values of the coefficients with the two models. In Figure 16 we illustrate the grid points (in Diffusion Maps coordinates) colored with the l^2 norm for each coefficient. In Table 1 we also provide the mean values of the l^2 norms for the drift and the diffusivity.

Table 2: Estimated mean l^2 differences between the estimated drift and diffusivity by the neural network and the Kramers-Moyal surrogate models on a (20 x 20) grid.

[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]

The mean values of the l^2 norm for the drift and diffusivity reported in the table above are of the same order of magnitude as the standard deviation in Figures 13, 14. This suggests that the discrepancy between the two approaches Kramers-Moyal and neural network falls into the range of uncertainty estimations of the neural network model discussed in Section A.6 of the SI.

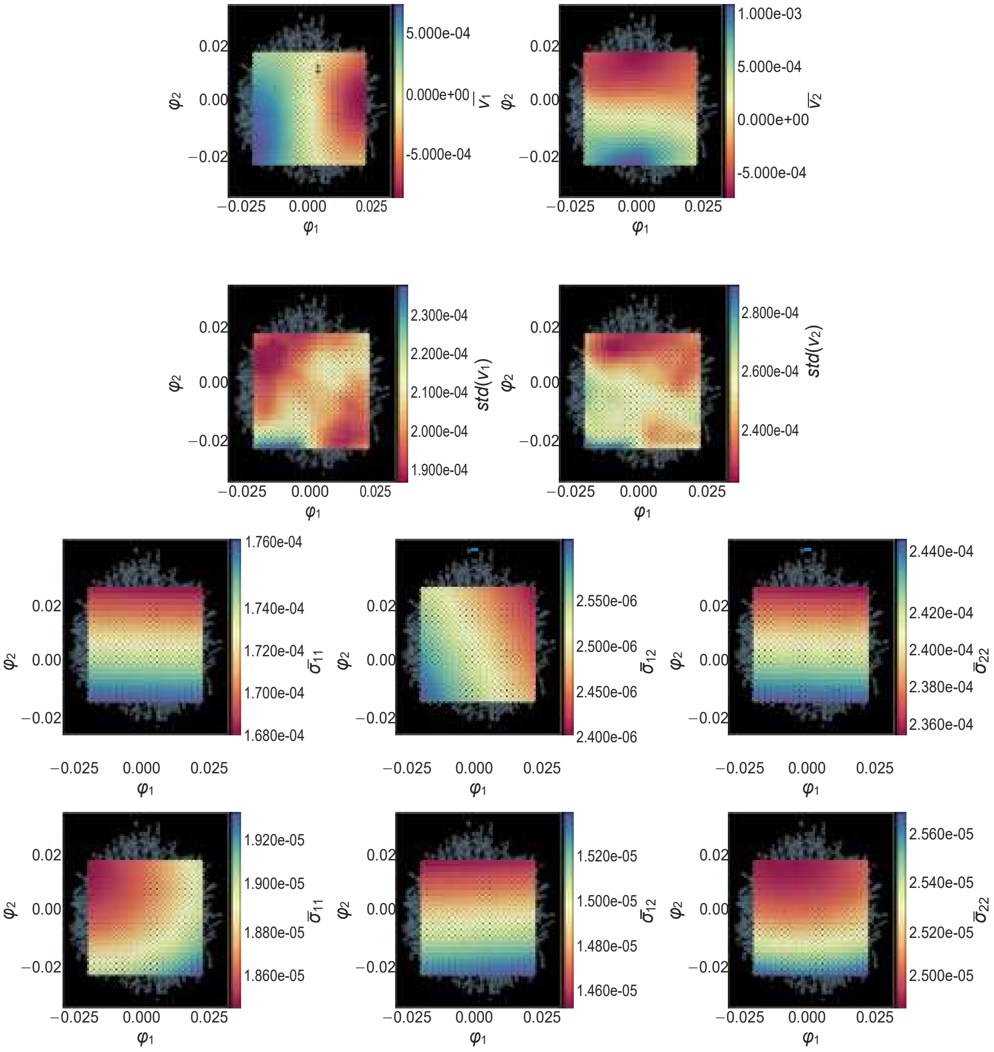

Figure 15: [First-Second Rows] illustrate the average drift and the standard deviation of the drift on a test grid (20x20) based on the trained 200 models. [Third-Fourth Rows] illustrate the average diffusivity and the standard deviation of the diffusivity on a test grid (20x20) based on the trained 200 models.

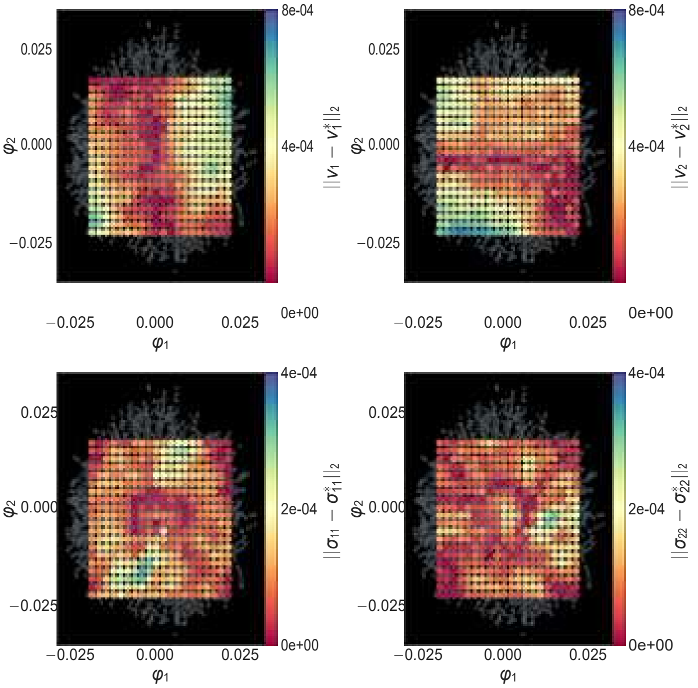

Figure 16: [First row] The l^2 norm of the difference between the evaluated drift of the network, and by the Kramers-Moyal surrogate model, colored as a function at the nodes of the grid. [Second Row]The l^2 norm of the difference between the evaluated diffusivity of the network and the one from the Kramers-Moyal surrogate model, colored as a function at the nodes of the grid.