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 $V^\star = 0.8$ 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 $\sim$ 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 $\sim$ 100 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, α (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

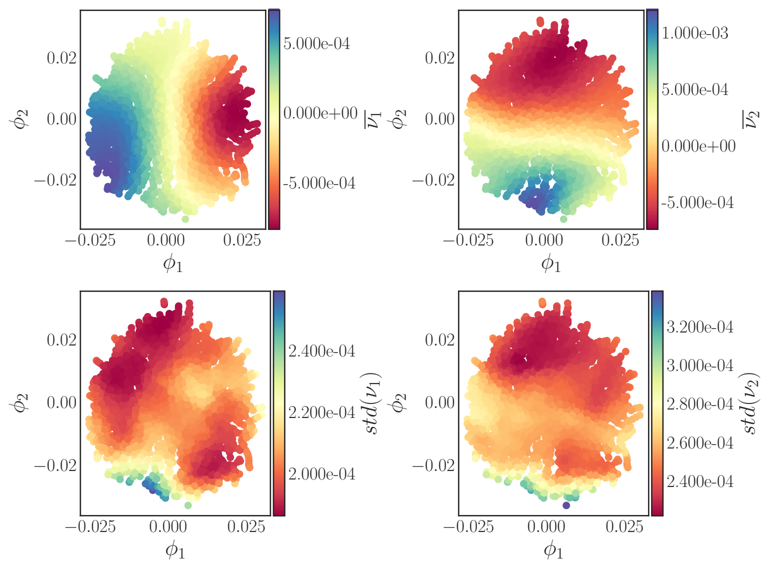

Figure 1: [First row] The average values (over the 200 models) of the drifts $\overline{\nu_1}$,$\overline{\nu_2}$ are shown as functions in the Diffusion Maps coordinates. [Second Row] The standard deviation of the estimated drifts $std(\nu_1)$, $std(\nu_2)$ is shown as functions in the Diffusion Maps 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 1 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 2. 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 3 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.

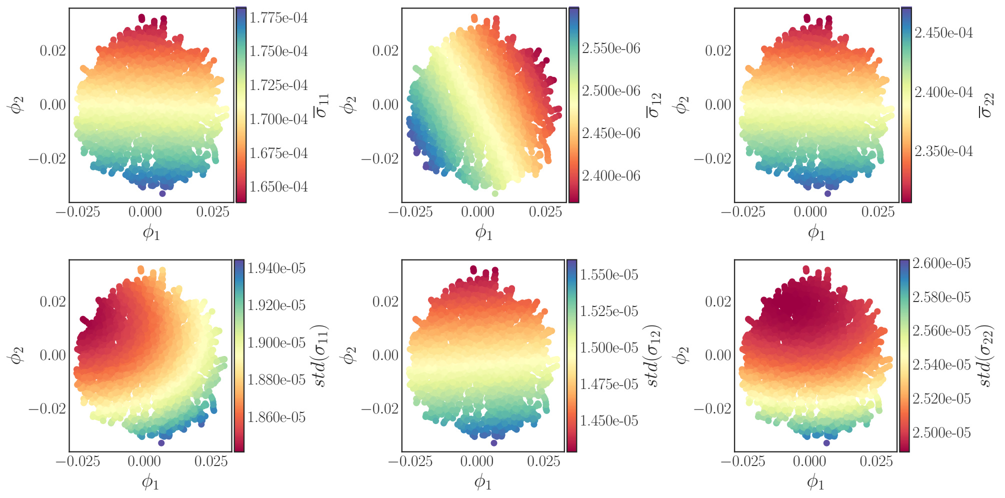

Figure 2: The average values (over the 200 models) of the estimated diffusivities $\overline{\sigma_1}$,$\overline{\sigma_2}$ are shown as functions in the Diffusion Maps coordinates. [Second Row] The standard deviation of the estimated diffusivities $std(\sigma_1)$, $std(\sigma_2)$ is colored as a function 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 4 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.

Mean $\|\|\nu_1 - \nu_1^*\|\|_2$	Mean $\|\|\nu_2 - \nu_2^*\|\|_2$	Mean $\|\|\sigma_{11} - \sigma_{11}^*\|\|_2$	Mean $\|\|\sigma_{22} - \sigma_{22}^*\|\|_2$
0.000234	0.000260	0.000104	0.000083

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 1, 2. 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.

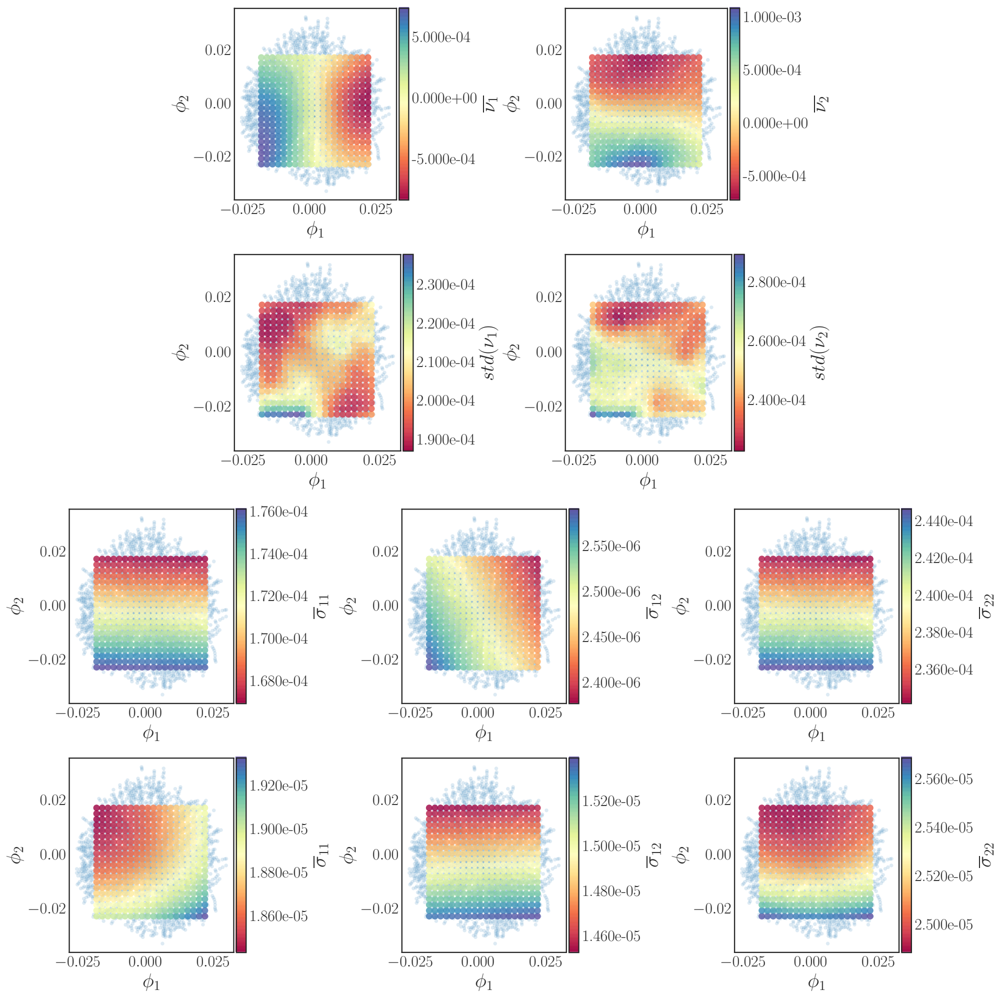

Figure 3: [First-Second Rows] illustrate the average drift and the standard deviation of the drift on a test grid (20x20) based on the trained 200 models. [Third-Fourth Rows] illustrate the average diffusivity and the standard deviation of the diffusivity on a test grid (20x20) based on the trained 200 models.

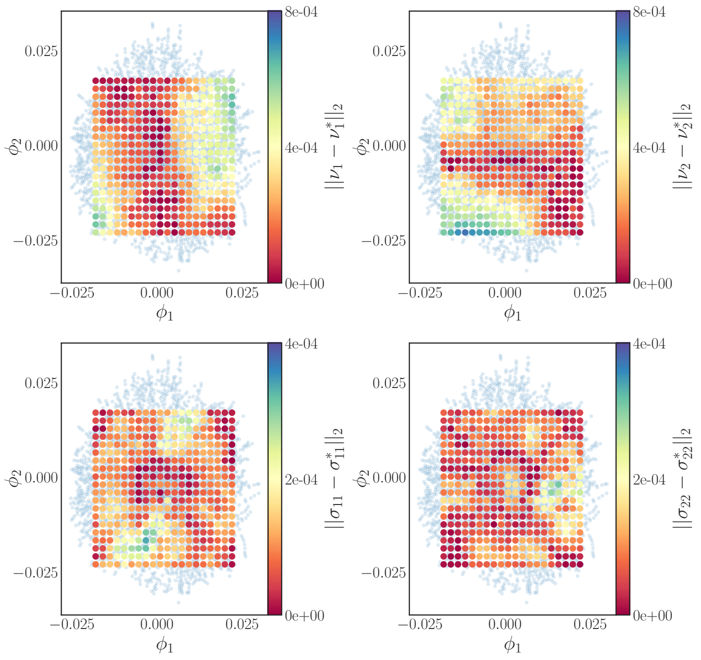

Figure 4: [First row] The l^2 norm of the difference between the evaluated drift of the network, and by the Kramers-Moyal surrogate model, colored as a function at the nodes of the grid. [Second Row]The l^2 norm of the difference between the evaluated diffusivity of the network and the one from the Kramers-Moyal surrogate model, colored as a function at the nodes of the grid.