Convolutional Neural Network-based Colloidal Self-Assembly State Classification

Journal:	*Soft Matter*
Manuscript ID	SM-ART-02-2023-000139.R1
Article Type:	Paper
Date Submitted by the Author:	24-Mar-2023
Complete List of Authors:	Lizano-Villalobos, Andrés; Louisiana State University, Cain department of Chemical engineering Tang, Xun; Louisiana State University, Cain Department of Chemical Engineering

Journal Name

ARTICLE TYPE

Cite this: DOI: 00.0000/xxxxxxxxxx

Convolutional Neural Network-based Colloidal Self-Assembly State Classification [†]

Andres Lizano,[a] and Xun Tang[a]

Received Date
Accepted Date

DOI: 00.0000/xxxxxxxxxx

Colloidal self-assembly is a viable solution to making advanced metamaterials. While the physicochemical properties of the particles affect the property of the assembled structures, particle configuration is also a critical determinant factor. Colloidal self-assembly state classification is typically achieved with order parameters, which are aggregate variables normally defined with nontrivial exploration and validation. Here, we present an image-based framework to classify the state of a 2-D colloidal self-assembly system. The framework leverages deep learning algorithms with unsupervised learning for state classification and a supervised learning-based convolutional neural network for state prediction. The neural network models are developed using data from an experimentally validated Brownian dynamics simulation. Our results demonstrate the proposed approach gives a satisfying performance, comparable and even outperforms the commonly used order parameters in distinguishing void defective states from ordered states. Given the data-based nature of the approach, we anticipate its general applicability and potential automatability to different and complex systems where image or particle coordination acquisition is feasible.

1 Introduction

Self-assembly of colloidal nano- and micron-sized particles into secondary structure has been a long-standing interest to both academia and industry, as it holds promise for advanced materials with novel physicochemical properties that would permit applications in photonic devices, health care, advanced computing, bio- and nanotechnology[1–3]. While the properties of the assemblies are affected by many factors such as the physiochemical property of the individual particle, including its shape, size, and surface chemistry properties, the configuration of the assembly, for example, the regularity of the arrangement of the particles also plays a key role in determining the functionality of the assemblies. For example, a perfectly ordered structure of colloidal assembly could demonstrate iridescence mimicking the natural butterfly and oval properties[2,4,5]. Current approaches for colloidal self-assembly state quantification typically deploy dimensionality reduction to project the high dimensional system into a low-dimension coordinates, using either mathematical approaches such as Principal Component Analysis, UMAP, etc., or physics reasonings-based aggregate variables such as order parameters[6,7].

Order parameters are mathematical functions, defined from physics reasoning and have been one of the dominating colloidal self-assembly state descriptors, due to their capability of relating their numerical values to physics meanings[8]. Popular examples include the radius of gyration, R_g, which quantifies the root-mean square distance between particles within an ensemble, and can be used to capture the compactness of the particles[9–13]; $C6$, the average number of nearest neighbors of all particles in the system, which can be used to quantify the local ordering of the structure[14–16]; and Ψ_6, which captures the particle-particle bond orientation, and can be used to quantify the global ordering of the ensemble[15,17–19]. Other widely-used order parameters include pair-correlation functions, volume fraction, and dihedral angles.

While effective, order parameters suffer from several limitations. First, identifying an order parameter is nontrivial, and it could involve extensive theoretical and experimental exploration and validation. Second, as order parameters are application-specific, when a new application is needed, a new or a new set of order parameters would need to be identified. Third, given the complexity of the system and the requirement of the application, a single order parameter might not suffice where multiple order parameters would be needed for a specific problem, and chances exist that even the combination of multiple order parameters might not be able to capture the structure of interest. Figure 1 shows such an example, where two order parameters $C6$ and Ψ_6 are used to distinguish the different state of an electric field-mediated colloidal self-assembly system. While they provide an easy classification among obviously different structures (Figure 1a-e), to capture the subtle difference, such as void defect

---

[a] *Cain Department of Chemical Engineering, Louisiana State University, Baton Rouge, LA 700803. E-mail: alizan2@lsu.edu*

[b] *Cain Department of Chemical Engineering, Louisiana State University, Baton Rouge, LA 700803. E-mail: xuntang@lsu.edu*

[†] Electronic Supplementary Information (ESI) available: See DOI: 00.0000/00000000.

(Figure 1f) vs. crystalline structure (Figure 1h), a delicate resolution or cutoff of the parameter values for state classification is needed. However, a finer resolution would lead to increased number of states, which can result in intractable computational issues, while a coarser resolution would result in state misclassification, which will further undermine the operating design of the system thus the quality of the assembly.

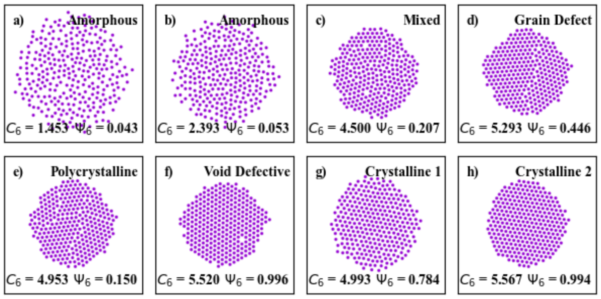


Fig. 1 Typical colloidal self-assembly configurations in a quadrupole electric batch system, labeled with order parameters to indicate different states. a-b) Amorphous structures where the particles are dispersed and far away from each other. c) Mixed defects including void defect and grain defect. d) Grain boundary defect structure consisting of one main grain boundary inside the assembly. e) Multi-crystalline state with several distinct internal grain boundaries. f) Two-void defective state in a closely packed and highly ordered structure. g) Highly ordered structure with low compactness. h) Packed and ordered crystalline state.

On the other hand, machine learning approaches have started to show successes in solving configuration and topology-related problems that are challenging to conventional methods, in protein folding[20–22], network identification[23–25], material discovery[26–29], and colloidal self-assembly[6,30,31]. Machine learning algorithms can be broadly grouped into three general categories: unsupervised, supervised and reinforcement learning[32–34]. Unsupervised learning refers to the process of identifying characteristics from a data set without prior knowledge of the labeling, allowing feature extraction, dimensionality reduction and clustering from raw numerical data[35–37]. Whereas supervised learning learns the underlying patterns of the labeled data to perform mapping of an input to a labeled output, and is normally used for classification and regression models[38–40]. In the work[30], Adorf et al. used machine learning models to represent a three-dimensional colloidal self-assembly process into states, using spherical harmonics descriptors as the inputs, and a two-step dimensionality reduction process for clustering. In their[31], Wang et al. proposed a descriptor-free strategy to classify particle packing properties, with point network cloud and Gaussian mixture models. Other notable works on machine learning-based methods for soft matter system state detection and order parameter identification include Ref. [41-45][41–45]. These studies demonstrated the successful applications of advanced machine learning techniques in exploring new methods of soft matter state detection and classification. However, most of the work still rely on features/signatures defined based on physics reasoning or domain knowledge, such as particle relative distance, particle bond, and particle bond length etc., to pre-process data before training their unsupervised machine learning models[43–45], or to label their dataset for supervised machine learning model development[41].

Here in this work, we present a convolutional neural network (CNN)-based approach for a 2-dimensional colloidal self-assembly system state representation and classification, as shown in Figure 2, which does not require physics-based reasoning to identify features or signatures, as compared to previous work in developing the machine learning models. Specifically, we deploy unsupervised learning to cluster the sampling images generated from Brownian Dynamics simulation, based on features extracted from the images without physics reasoning. With the labeling learned from the unsupervised learning, we then develop a convolutional neural network to predict new images or states. Using a simulated electric field-mediated colloidal self-assembly system as an example, we demonstrate that our approach can distinguish different and important configurations, thus providing a viable alternative to order parameters, as a state representation and classification approach. Given the data-driven nature of our approach, we anticipate it to be applicable to systems with different number of particles, different sized and different shaped particles, and different thermodynamics systems, where image acquisition is feasible.

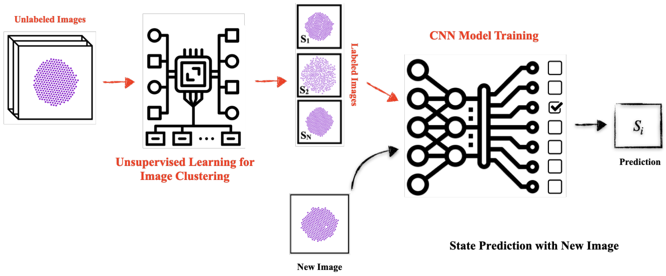


Fig. 2 The proposed approach comprises two steps: the first step is to train an unsupervised neural network to produce labels for unlabeled 2-D images sampled from the system (red pathway); the second step is to develop a convolutional neural network model with the labeled images, to classify new images into their corresponding states (black pathway).

2 System of Interest and Data Generation

In this study we focus on a 2-dimensional colloidal self-assembly process, where 300 $1.5\mu m$ spherical SiO_2 particles are suspended in deionized water in a batch container surrounded by four electrodes, the same as in Tang *et al.*[46]. The electric field is created by applying a $1MHz$ AC field to the system for a radially symmetric force field, that induces particles into dipoles for particle-particle, particle-field, and particle wall interactions. By manipulating the voltage level, thus the field compressing strength, we harness manipulation of the particle movement for assembly. The detailed description of the dynamics and forces in this system is provided in Tang *et al.*[46], and Juárez and Bevan[16].

Instead of performing experiments for data generation, we adopted a detailed force-balance Brownian dynamics simulation model, originally developed in the Bevan Lab[16], for a proof-of-concept demonstration of our approach. This model was validated against experimental measurements and was previously implemented to design optimal control policies for ordered 2-D colloidal self-assembly[16,46,47]. To train our machine learning models, we generated 132 trajectories of the assembly process, with the 300-particle bulk system initialized from randomized initial configurations, under different constant voltages. Each simulation has a duration of 1000s, and the x and y coordinates of each particle along the simulation were periodically stored for image reconstruction. Table S1 summarizes the number of trajectories obtained for each of the four voltage levels, as well as the time interval at which the x,y coordinates were saved. In total, we obtained 13765 images of hard sphere colloidal self-assembly from the x,y coordinates, using GNU Plot and bash scripting, and this constructs the sampling pool for our work

The same four different voltage levels $\lambda = 0.2, 0.9, 2, 19.7$ as in Tang et al.[46] are considered in this work, to generate configurations that include fluid states (at $\lambda = 0.2$), defective and ordered states (at $\lambda = 19.7$), as well as configurations in between (at $\lambda = 0.9, 2$). The dimensionless variable λ indicates the strength of the compression force in the system, where a higher λ value represents a higher voltage magnitude thus a stronger compression force. The lowest voltage level is chosen to produce a dispersed state while still being able to confine the particles from diffusing out of the center of the system. The highest voltage level is chosen to produce perfectly ordered configuration, and the intermediate voltages were chosen to provide mild relaxation for defect correction, while maintaining a certain degree of crystallinity. The detailed conversion between voltage and λ was given in Tang et al.[46], and the detailed BD simulation parameters are summarized in supplementary Table S2.

3 Results

3.1 Unsupervised Learning for State Clustering

Given a set of unlabeled images/states, we first sought to use unsupervised learning to classify the images into different states, which will further be used to train the convolutional neural network for new image-based state prediction. Specifically, we first performed feature extraction to transform an image into a numerical object, which serves as the input to the unsupervised clustering process. Depending on the size of the extracted features, dimensionality reduction could be applied to reduce the computational cost. After clustering, the images were then grouped into clusters representing different states of the colloidal system. This unsupervised clustering process is depicted in Figure 3.

3.1.1 Feature Extraction

For feature extraction, we evaluate three image-based methods in addition to order parameter. Specifically, we investigate the performance of 1) using order parameters as the features, 2) using pixel values as the features, 3) using features extracted from a deep inception model, trained with the ImageNet dataset, and 4) using features extracted from a convolutional autoencoder, trained over our dataset. To use pixel values as the features, grayscale pixel values were first extracted from each image (800x800 px), and then flattened to obtain the feature vector (of size 640000). The ImageNet method consists of a deep inception model, pre-trained over the ImageNet dataset[48], from which we dropped the final layer and obtained 150000 features.

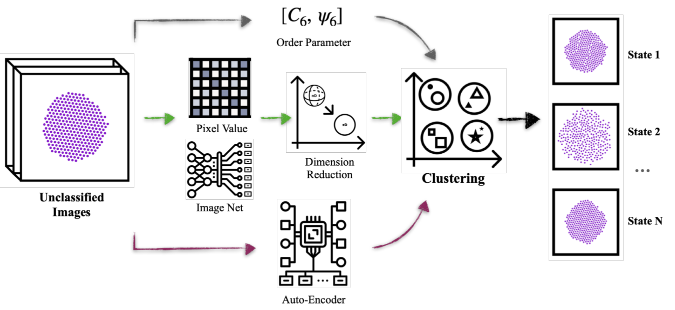


Fig. 3 Unsupervised learning framework. A set of unclassified images is processed with a feature extraction method (order parameter, pixel values, ImageNet or an autoencoder) for numerical representation (feature space) of the images. Dimensionality reduction is conducted when the features have a high dimension (e.g. 500), before being clustered with HDBSCAN into labeled clusters/states.

Finally, we developed a convolutional autoencoder by stacking a 2-D convolutional layer using 9 filters with a 20x20 kernel size and ReLu activation. The output from the convolutional layer was flattened and passed into a dense layer with 500 neurons using ReLu activation.The features are obtained by taking the output of this middle dense layer. The decoder was constructed by using a single dense layer with its size defined by the resolution of the initial image (HxW neurons) which then were reshaped to reconstruct the input image. To train the autoencoder, a loss was constructed by weighting the mean squared error loss with the binary cross entropy with 0.7 and 0.3 as their respective weights. The selection of this 500 feature dimension was based on the maximization of the model accuracy over the testing set, which reached a test accuracy of 81%.

Before clustering, we sought to understand which features would have the strongest distinguishing power among different images/configurations. We adopted a weighted Euclidean distance, $d_{i,j}$, on the feature space as described by Equation 1, to quantify the similarities between two different configurations in the feature space. $d_{i,j} = 0$ indicates non-distinguishable, and $d_{i,j} = 1$ indicates a significant difference between two samples, thus the features that give higher $d_{i,j}$ values are considered with a stronger distinguishing power. $D_{i,j}$ refers to the standard Euclidean distance calculated as the norm of the vector difference between two feature vectors, and $\vec{r}_i$ is the feature vector or the position vector of sample i on the feature space.

$$d_{i,j} = \frac{D_{i,j}}{\max_{i<j} D_{i,j}} \tag{1}$$

$$D_{i,j} = \vec{r}_i - \vec{r}_j$$

Figure 4 summarizes the similarity score $d_{i,j}$ using order parameters, pixel values, features extracted from ImageNet, and features extracted from the autoencoder for 10 different configurations as shown in Figure 4a, which include crystalline states and void defects that are previously found challenging to distinguish. The pairwise similarity heat-map in Figure 4b demonstrates that order parameters managed to distinguish amorphous states from ordered states, but struggled in distinguishing states with minor defects such as void defects (C_6 and C_7) from crystal states (C_5 and C_2). In contrast, features with all the other three approaches showed improved distinguishing power, across all the tested configurations, indicated by the higher $d_{i,j}$ values. In particular, images represented in pixel values seemed to have an equally well distinguishability across all the configurations, while features extracted from ImageNet and the autoencoder showed a stronger distinguishing power between amorphous and ordered states, as compared to distinguishing void defective states and crystal states, as the metric values for the first case are higher. Figure 4c shows the histogram distribution of the $d_{i,j}$ values for all the 13765 images in our data set for each of the methods, for a statistically significant analysis. Consistent observations were found comparing Figure 4c to the heat-map in Figure 4b, that order parameters gave a similarly low score across the entire dataset, and pixel value gave the highest metric score, while ImageNet and autoencoder showed improved distinguishing power with noticeable sensitivity to configurations. All these observations confirmed the promise of distinguishing different configurations with the three image-based feature extraction approaches.

3.1.2 Dimensionality Reduction

The number of features extracted from pixel values and ImageNet are 640000 and 15000, respectively, creating a challenge for training machine learning models due to their high dimension. To address this, we employed UMAP (Uniform Manifold Approximation and Projection) to reduce the dimension to 500, matching the dimension of features obtained from the autoencoder, prior to clustering. UMAP maps high-dimensional data into a low-dimensional space while preserving the topological structure and global relationships between data points, represented by the distance between the points[49]. To ensure the quality of the projection, we optimized the UMAP's hyperparameters using the trustworthiness metric and found optimal performance at 0.987 with 500 components and 10 neighbors. This balance conserves both global and local characteristics of the projection when approximating the manifold.

After reducing the dimension of the features, we then deployed HDBSCAN (Hierarchical Density-base Spatial Clustering of Applications with Noise) to cluster the images into different clusters/states. HDBSCAN is a clustering algorithms that starts by identifying densities based on the distance between the points, and then identifies the clusters hierarchically based on the found densities[50]. The performance of HDBSCAN is affected by hyperparameters such as minimum cluster size, minimum samples, and epsilon. To optimize the performance, we used genetic algorithm with silhouette, Calinsky-Harabasz and Davies-Bouldin scores as the fitness functions, and found that an optimal performance was obtained by setting the hyperparameters as: minimum cluster size of 140 images, minimum samples of 85, and epsilon of 0.24.

To evaluate the clustering results with respect to physics, we defined a metric, purity $\in (0,1]$, which quantifies the degree of misclassification ($1 - purity$) in each cluster, and is calculated by dividing the highest number of images that should be grouped as the same state, by the total number of images in that cluster. For example, if there are, with visual inspection, 8 out of the total 10 images in cluster n should be in the same state, the purity for cluster n is then equal to $8/10 = 80\%$. Therefore, the higher the purity, the more desirable the clustering. Results in Table 1 summarize the purity values for the clusters obtained using order parameters, pixel values, features from ImageNet, and the autoencoder. For a fair comparison, we report the comparison with results for the same number of clusters, and we also found that the number of clusters had limited impact on the purity in general, in this study.

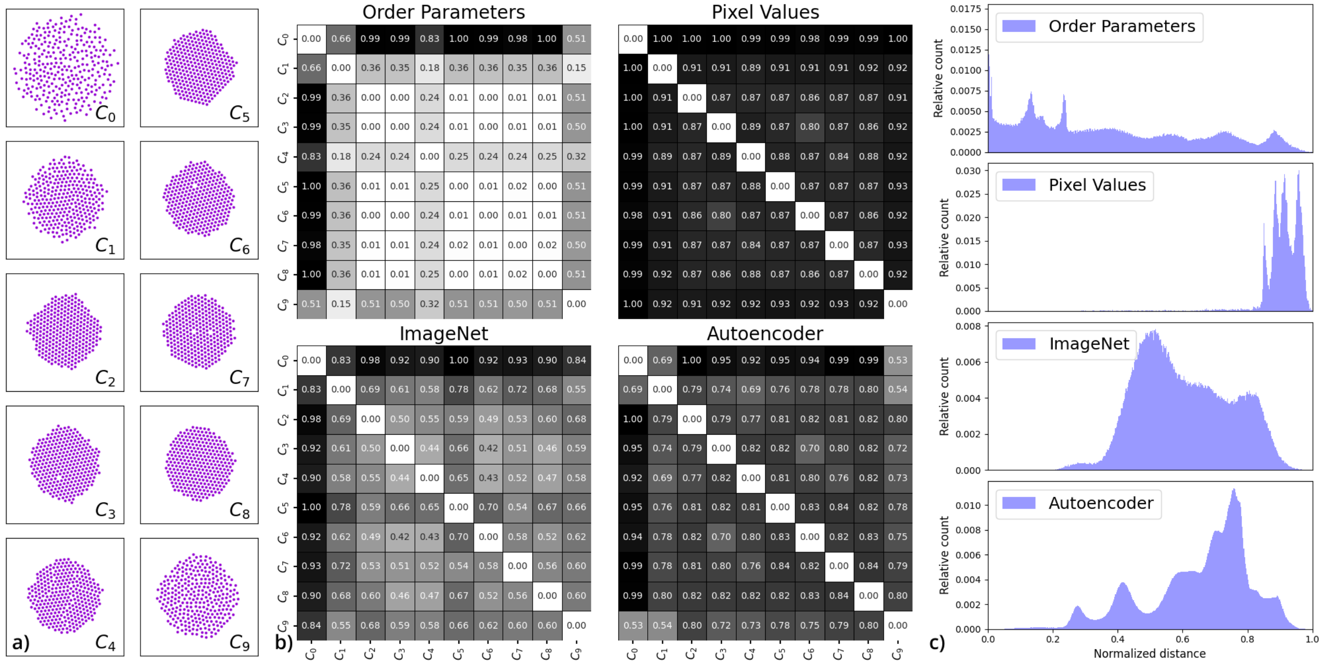
Contrary to our expectation, although pixel value gave the highest distance metric score, clustering with pixel values had the lowest on average purity of 66%, whereas autoencoder scored the highest at 92%, order parameter and ImageNet gave the similarly good performance of 89%. All of the resulting clusters with pixel values had misclassified images, with a purity ranging from 40% to 95%. The majority of the clusters using order parameters, ImageNet and autoencoder have 100% purity, with autoencoder-based approach having the highest lowest purity of 90%. A closer inspection to the results revealed that the poor performance of the pixel value-based approach is primarily due to a prevalent misclassification of amorphous states with highly ordered states (e.g. Figure 5b), while the accuracy of the order parameter and ImageNet-based approaches are jeopardized by misclassifying void defects with perfect crystalline states (e.g. Figure 5a and b). The performance of the autoencoder-based approach is primarily undermined by the misclassification of polycrystalline states with defective states with grain boundary (e.g. Figure 5d).

Figure 6 gives a 2-D representation of the clustering results with the autoencoder-based approach for visualization, where the colors indicate different clusters from the UMAP. In Figure 6, seven clusters are shown with a group of points marked as noise, which contains images that were not able to be fitted into any of their neighboring groups. Note that the number of noise clusters can be adjusted by tuning the hyperparameters in UMAP, however, this comes with a tradeoff between the total number of clusters and the purity of each cluster.

3.2 Convolutional Neural Network for State Prediction

As the autoencoder-based approach provided the highest clustering purity, we proceeded to develop a convolutional neural network (CNN) with the labels learned from the autoencoder-based clustering approach. The CNN model learns the inherent pattern in the image to predict its corresponding label. In our CNN model, we stacked four convolutional 2-D layers with 32, 64, 128, and 256 filters respectively. All these layers used a 3x3 kernel, zero padding and ReLu as the activation function. After flattening the output of the convolutional layers, a fully connected layer with 512 neurons plus bias and ReLu as activation function was added. Finally, a dense layer with the same number of neurons as clusters was added, using the Softmax activation function to obtain the likelihood of the input image to belong to each of the clusters.

To train the model, we split the labeled 11594 images into three subsets: a training set, which contains 80% of the total images, a validation set with 10% of the total images, to reduce the possibility of overfitting, and a testing set with 10% of the total images for validation outside of the training process. The data labelled



Order Parameters

	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
C_0	0.00	0.66	0.99	0.99	0.83	1.00	0.99	0.98	1.00	0.51
C_1	0.66	0.00	0.36	0.35	0.18	0.36	0.36	0.35	0.36	0.15
C_2	0.99	0.36	0.00	0.00	0.24	0.01	0.00	0.01	0.01	0.51
C_3	0.99	0.35	0.00	0.00	0.24	0.01	0.00	0.01	0.01	0.50
C_4	0.83	0.18	0.24	0.24	0.00	0.25	0.24	0.24	0.25	0.32
C_5	1.00	0.36	0.01	0.01	0.25	0.00	0.01	0.02	0.00	0.51
C_6	0.99	0.36	0.00	0.00	0.24	0.01	0.00	0.01	0.01	0.51
C_7	0.98	0.35	0.01	0.01	0.24	0.02	0.01	0.00	0.02	0.50
C_8	1.00	0.36	0.01	0.01	0.25	0.00	0.01	0.02	0.00	0.51
C_9	0.51	0.15	0.51	0.50	0.32	0.51	0.51	0.50	0.51	0.00

Pixel Values

	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
C_0	0.00	1.00	1.00	1.00	0.99	0.99	0.98	0.99	0.99	1.00
C_1	1.00	0.00	0.91	0.91	0.89	0.91	0.91	0.91	0.92	0.92
C_2	1.00	0.91	0.00	0.87	0.87	0.87	0.86	0.87	0.87	0.91
C_3	1.00	0.91	0.87	0.00	0.89	0.87	0.80	0.87	0.86	0.92
C_4	0.99	0.89	0.87	0.89	0.00	0.88	0.87	0.84	0.88	0.92
C_5	0.99	0.91	0.87	0.87	0.88	0.00	0.87	0.87	0.87	0.93
C_6	0.98	0.91	0.86	0.80	0.87	0.87	0.00	0.87	0.86	0.92
C_7	0.99	0.91	0.87	0.87	0.84	0.87	0.87	0.00	0.87	0.93
C_8	0.99	0.92	0.87	0.86	0.88	0.87	0.86	0.87	0.00	0.92
C_9	1.00	0.92	0.91	0.92	0.92	0.93	0.92	0.93	0.92	0.00

ImageNet

	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
C_0	0.00	0.83	0.98	0.92	0.90	1.00	0.92	0.93	0.90	0.84
C_1	0.83	0.00	0.69	0.61	0.58	0.78	0.62	0.72	0.68	0.55
C_2	0.98	0.69	0.00	0.50	0.55	0.59	0.49	0.53	0.60	0.68
C_3	0.92	0.61	0.50	0.00	0.44	0.66	0.42	0.51	0.46	0.59
C_4	0.90	0.58	0.55	0.44	0.00	0.65	0.43	0.52	0.47	0.58
C_5	1.00	0.78	0.59	0.66	0.65	0.00	0.70	0.54	0.67	0.66
C_6	0.92	0.62	0.49	0.42	0.43	0.70	0.00	0.58	0.52	0.62
C_7	0.93	0.72	0.53	0.51	0.52	0.54	0.58	0.00	0.56	0.60
C_8	0.90	0.68	0.60	0.46	0.47	0.67	0.52	0.56	0.00	0.60
C_9	0.84	0.55	0.68	0.59	0.58	0.66	0.62	0.60	0.60	0.00

Autoencoder

	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
C_0	0.00	0.69	1.00	0.95	0.92	0.95	0.94	0.99	0.99	0.53
C_1	0.69	0.00	0.79	0.74	0.69	0.76	0.78	0.78	0.80	0.54
C_2	1.00	0.79	0.00	0.79	0.77	0.81	0.82	0.81	0.82	0.80
C_3	0.95	0.74	0.79	0.00	0.82	0.82	0.70	0.80	0.82	0.72
C_4	0.92	0.69	0.77	0.82	0.00	0.81	0.80	0.76	0.82	0.73
C_5	0.95	0.76	0.81	0.82	0.81	0.00	0.83	0.84	0.82	0.78
C_6	0.94	0.78	0.82	0.70	0.80	0.83	0.00	0.82	0.83	0.75
C_7	0.99	0.78	0.81	0.80	0.76	0.84	0.82	0.00	0.84	0.79
C_8	0.99	0.80	0.82	0.82	0.82	0.82	0.83	0.84	0.00	0.80
C_9	0.53	0.54	0.80	0.72	0.73	0.78	0.75	0.79	0.80	0.00

Fig. 4 a) Configurations used to calculate the similarity matrix, where C_n indicates configuration n, to be different from order parameter $C6$. b) Similarity matrices for features with the four different feature extraction methods. c) Similarity distributions for the whole data set, using features extracted with the four different methods.

Table 1 Clustering results with different features extraction methods.

Method	Order Parameters	Pixel Values	ImageNet	Autoencoder
Number of clusters	7	6	7	7
Non-pure clusters	2	6	3	3
Highest Purity	100%	95%	100%	100%
Lowest Purity	57%	40%	45%	90%
Average Purity	89%	66%	89%	92%

as noise was discarded and the categorical cross-entropy function was selected as the loss function to be minimized.

Figure 7 shows the confusion matrices of the CNN, on the testing set. From the results, we can see that the model achieved an overall classification accuracy of 98.8% (i.e. average of the diagonal values in Figure 7a). The lowest classification accuracy was 92%, obtained for cluster 0. Inspecting the configurations in each cluster reveals that both cluster 0 and 3 contained single void defect configurations that differ in location, as shown in Figure 6b. The redundant clusters might be due to the sensitivity of the autoencoder-based approach to the perimetric shape. Once we merge together clusters 0 and 3, an overall validation accuracy of 99.8% was then achieved (Figure 6c).

To further validate and illustrate the performance of the autoencoder-based state classification approach, we applied the CNN model to classify images in Figure 1, as a direct comparison to order parameters. Figure 8 summarizes the prediction results of the images using the 6-cluster CNN model obtained from figure 7c. The model successfully distinguished the void defective state in Figure 8f as state 3 ($S = 3$) from the crystalline states in Figure 8g and 8h, while grouping amorphous conditions with different compactness into a single state (Figure 8a,b, $S = 5$). The model is also able to discern different defective configurations (Figure 8c vs. Figure 8d,e,f), but seems to be less sensitive to polycrystalline (Figure 8e) and states with major grain boundary (Figure 8d), as compared to using order parameters $C6$, and Ψ_6. Overall, the CNN model performs well in classifying different configurations into different states. Furthermore, the CNN model was trained to have in total 6 different states, such a small number of states significantly reduced the total amount of discretized states normally needed for order parameters to accurately distinguish colloidal self-assembly configurations, thus improving the convenience for practical implementation and analysis. For example, in[47], a total of 6000 discretized states were used to capture the fine details of the system. As the number of clusters can also be optimized by tuning the hyperparameters in the HDBSCAN, with a larger amount of training images, we expect further improvement of the clustering accuracy, as well as easy adaptation to systems with more complicated dynamics.

4 Conclusions and Discussions

In this study, using a simulated 2-D electric field-mediated self-assembly system, we demonstrated that the autoencoder-based CNN state classification approach can give a comparable performance to order parameters in distinguishing different configurations of the assembly, especially in detecting void defective states

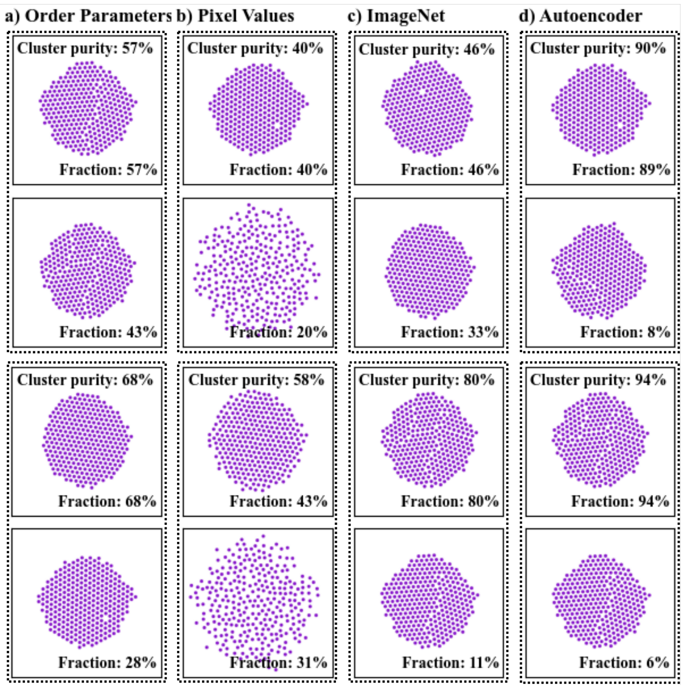


Fig. 5 Misclassifications observed from the results produced by each feature extraction method. Samples from two least pure clusters obtained from features extracted using order parameters (panel a), pixel values (panel b), ImageNet (panel c) and the autoencoder (panel d). The dotted lines associate samples from the same cluster, with the purity for each cluster shown as percentage.

from the perfectly ordered state. With unsupervised learning, we successfully clustered different configurations into different states by learning the inherent patterns embedded in images taken from the system, without using state descriptors, such as order parameters. This strategy avoids the arduous human efforts in trial-and-error exploration and validation, associated with identifying order parameters for state classification, thus permitting potential automation of state classification. The flexibility in tuning the hyperparameters in the supervised and unsupervised model development, and in determining the number of clusters provides extra advantages over conventional state classification approaches, for practical implementations.

Understanding how the system phase changes with respect to the design parameters is critical to deciphering and controlling the system dynamics, and the labeling/descriptor developed from our work can also contribute to this effort. For illustration purpose, we present the phase diagram as a function of the four voltage levels in supplementary Figure S1, which shows how the system phase evolves as the voltage level changes. To develop the phase diagram, we analyzed the endpoint (at 1000s) configurations of all the BD simulations performed in this study for each of the four voltage levels. To account for the stochasticity of the system, we calculated the probability for the system to end up in each of the six identified phases/states, and presented the probability distribution as a heatmap. As demonstrated in Figure S1, at low voltage levels (Voltage level 1 and 2), the system would end up in fluid state (S_5), as the voltage level increases, defective states and more compact states would emerge (Voltage level 3),

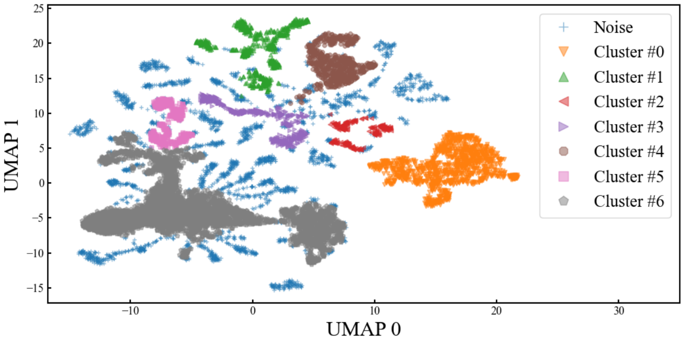
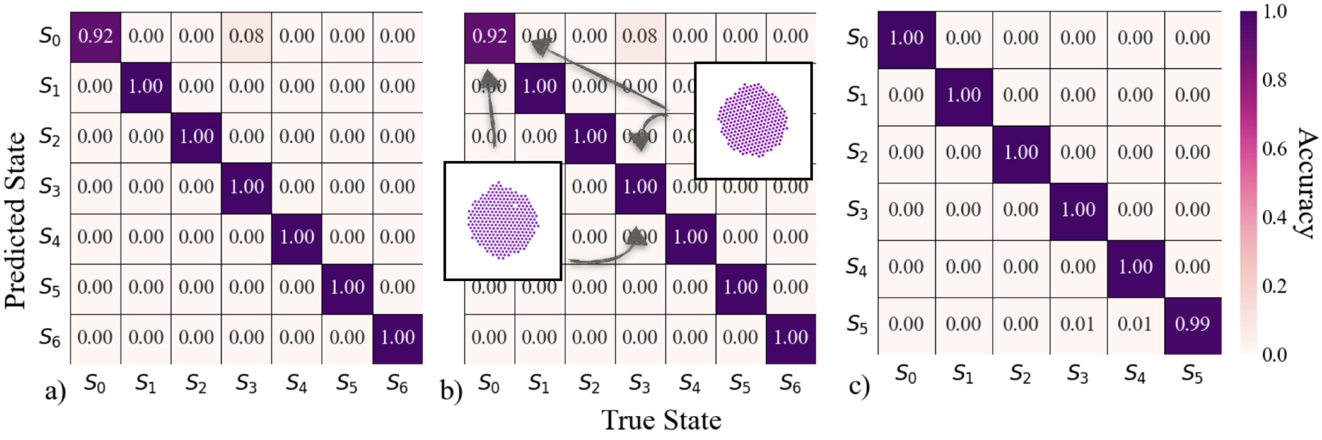


Fig. 6 2-D rendering of the clusters obtained by applying UMAP and HDBSCAN to the features learned from the Autoencoder. Data points marked as noise are images not classified into any of the clusters.

and at the highest voltage (Voltage level 4), ordered crystalline states (S_0) can be achieved, while the system can still get stuck in defective states (S_1,S_2), but no fluid states can be achieved. Note that, as the phase diagram is developed by taking the configuration at 1000s of the simulation, these configurations might not represent the equilibrium phase of the system (especially for higher voltage levels as multiple metastable states might exist), however, the same procedure described here can be applied to equilibrium system condition analysis, as well as to phase diagrams in terms of other controlling parameters, such as salt concentration, AC field frequency etc.

Despite the promising results, we also noticed several challenges and open issues for future studies. First, the lack of a reliable and convenient validation metric for the unsupervised clustering hinders the scalability of the proposed approach. It is critical to ensure the unsupervised learning has correctly grouped images into different clusters. Although there are metrics to test the convergence of the training of the unsupervised model, these metrics do not necessarily relate to the physics of the system. In this work, we have introduced purity to gauge the clustering performance, and relied on visual inspection to ensure the quality of the examination. While such an approach works at small scales (in terms of number of images), with relatively simple dynamics, more automatable approaches will be needed to handle systems with complicated dynamics which exhibit hundreds of thousands of distinct yet important states. Possible solutions can be to use available order parameters to aid the development of the model, or to deploy an iterative model development strategy which dynamically updates the model hyperparameters as new images come in.

Second, the boundary shape of the assembly shows a noticeable effect on the performance of the deep convolutional autoencoder-based CNN model. It was noticed that the model tends to misclassify states with the same single void defect but different perimetric shapes into different states (Figure 7b). Such a heavy dependence on local structure for classification is the foundation of CNN algorithms, but constructs an issue for our application. Potential solutions can include adding rotated images in the training set to reduce the boundary shape and defect location impact on the model performance. However, this would cause increased com-



a)

	S_0	S_1	S_2	S_3	S_4	S_5	S_6
S_0	0.92	0.00	0.00	0.08	0.00	0.00	0.00
S_1	0.00	1.00	0.00	0.00	0.00	0.00	0.00
S_2	0.00	0.00	1.00	0.00	0.00	0.00	0.00
S_3	0.00	0.00	0.00	1.00	0.00	0.00	0.00
S_4	0.00	0.00	0.00	0.00	1.00	0.00	0.00
S_5	0.00	0.00	0.00	0.00	0.00	1.00	0.00
S_6	0.00	0.00	0.00	0.00	0.00	0.00	1.00

b)

S_0	S_1	S_2	S_3	S_4	S_5	S_6
0.92	0.00	0.00	0.08	0.00	0.00	0.00
0.00	1.00	0.00	0.00	0.0		
0.00	0.00	1.00	0.00	0.0		
		0.00	1.00	0.00	0.00	0.00
		0.00	0.00	1.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	1.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	1.00

c)

	S_0	S_1	S_2	S_3	S_4	S_5
S_0	1.00	0.00	0.00	0.00	0.00	0.00
S_1	0.00	1.00	0.00	0.00	0.00	0.00
S_2	0.00	0.00	1.00	0.00	0.00	0.00
S_3	0.00	0.00	0.00	1.00	0.00	0.00
S_4	0.00	0.00	0.00	0.00	1.00	0.00
S_5	0.00	0.00	0.00	0.01	0.01	0.99

Fig. 7 Confusion matrices for CNN model prediction. a) Confusion matrix obtained from CNN-model trained for 7 cluster. b) Samples from clusters 0 and 3 contain similar single void defect states that undermines the prediction accuracy of cluster 0. c) Confusion matrix for CNN model with clusters 0 and 3 merged into a single cluster.

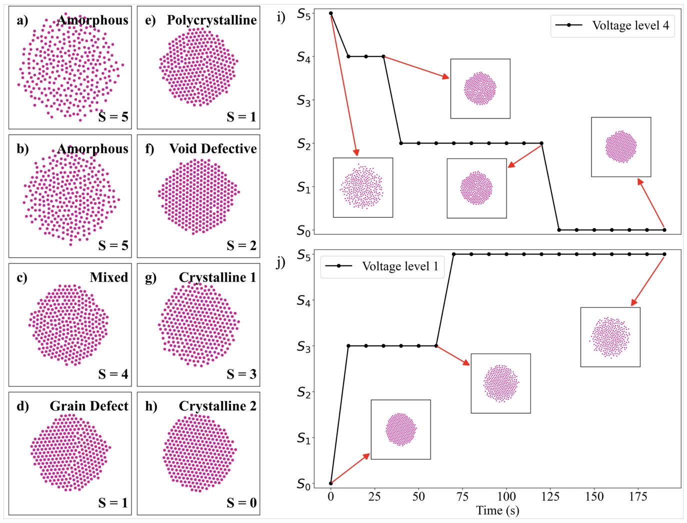


Fig. 8 Classification of different colloidal configurations using the 6-cluster CNN model. a-h) The model is able to distinguish different states, especially void defective states from the crystalline states, but shows less sensitivity to polycrystalline and single grain boundary defective states. S in the figure stands for state. i) System state evolution under the highest voltage (Voltage level 4) showing the system starts from state S_5, which corresponds to a fluid state, and assembles into the ordered state S_0. j) System state evolution under the lowest voltage (Voltage level 1) showing the system starts from an ordered state S_0, and relaxes into fluid state starting from state S_5. Images are sampled every 5 s from the simulation.

putational cost.

Third, the images in this study were reconstructed from x,y-coordinates simulated by a Brownian dynamics simulation, representing a well-defined, noise-free condition. However, real experimental images would normally come with noise, variations in intensity, and possible irregularity in particle size and shape. While we anticipate our method to hold its reliability for such disturbances, given the model to be trained with a sufficient amount of images, experimental validation will be needed, for a better understanding of the applicability, improvement, and limitation of the proposed approach.

Fourth, current machine learning-based 2-D or 3-D phase characterization approaches mainly utilize particle coordinates or neighborhood topological graph-based analysis for feature extraction[42–45,51], which require access to the coordinates of internal particles. The framework presented in this work is designed primarily for 2-D systems and its application to 3-D systems remains to be further examined. However, as image reconstruction techniques have shown the capability of reconstructing 3-D structures from 2-D cross images[52,53], we anticipate potential extension of the proposed 2-D image-based approach to 3-D systems, with modifications on the architecture and augmentation of 3-D image processing.

As imaging, data acquisition, and machine learning techniques advance, the application of image-based machine learning models would continue benefitting the study of configurational features and self-assembly dynamics, and we anticipate our work to contribute to future scalable and automatable strategies for material internal structure identification and classification.

5 Author Contribution

X. T. conceptualized the idea, A. L. conducted data generation and model development. A. L and X. T. analyzed the data and wrote the manuscript. The codes involved in this study are available at https://github.com/alizano94/SoftMatter-CollidalStateRep.git

6 Acknowledgments

X. T. and A. L. are supported by NSF grant #2218077, and the startup fund provided by the Cain Department of Chemical Engineering at Louisiana State University.

7 Competing Interests

The authors declare no competing financial interests.

Notes and references

1 M. Mayer, M. J. Schnepf, T. A. König and A. Fery, *Advanced Optical Materials*, 2019, **7**, 1800564.

2 E. S. Goerlitzer, R. N. Klupp Taylor and N. Vogel, *Advanced Materials*, 2018, **30**, 1706654.

3 X. Tang and M. A. Grover, *Annual Review of Control, Robotics, and Autonomous Systems*, 2022, **5**, 491–514.

4 M. Karg, T. A. König, M. Retsch, C. Stelling, P. M. Reichstein, T. Honold, M. Thelakkat and A. Fery, *Materials Today*, 2015, **18**, 185–205.

5 S. Sagebiel, L. Stricker, S. Engel and B. J. Ravoo, *Chemical Communications*, 2017, **53**, 9296–9299.

6 J. O'Leary, R. Mao, E. J. Pretti, J. A. Paulson, J. Mittal and A. Mesbah, *Soft Matter*, 2021, **17**, 989–999.

7 M. Dijkstra and E. Luijten, *Nature materials*, 2021, **20**, 762–773.

8 J. P. Sethna, *Statistical mechanics: entropy, order parameters, and complexity*, Oxford University Press, USA, 2021, vol. 14.

9 S. Das, M. Lange and A. Cacciuto, *The Journal of Chemical Physics*, 2022, **156**, 094901.

10 J. G. Coldstream, P. J. Camp, D. J. Phillips and P. J. Dowding, *Soft Matter*, 2022, **18**, 6538–6549.

11 F. Shen, H. Ling, W. Ge, Y. Yang, X. Wang, J. Ren and X. Wang, *Carbohydrate Polymers*, 2021, **261**, 117886.

12 M. Moinuddin and M. Tripathy, *Macromolecules*, 2022, **55**, 9312–9323.

13 R. Verweij, J. Melio, I. Chakraborty and D. J. Kraft, *arXiv preprint arXiv:2209.13427*, 2022.

14 E. Dieuzy, S. Auguste, K. Chougrani, V. Alard, L. Billon and C. Derail, *Colloids and Surfaces A: Physicochemical and Engineering Aspects*, 2021, **613**, 126082.

15 G. E. Fernandes, D. J. Beltran-Villegas and M. A. Bevan, *Langmuir*, 2008, **24**, 10776–10785.

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

17 E. O. Solis, P. I. Barton and G. Stephanopoulos, *Industrial & Engineering Chemistry Research*, 2010, **49**, 7746–7757.

18 G. E. Fernandes, D. J. Beltran-Villegas and M. A. Bevan, *The Journal of Chemical Physics*, 2009, **131**, 134705.

19 D. R. Nelson and B. Halperin, *Physical Review B*, 1979, **19**, 2457.

20 F. Noé, G. De Fabritiis and C. Clementi, *Current Opinion in Structural Biology*, 2020, **60**, 77–84.

21 L. Wei and Q. Zou, *International journal of molecular sciences*, 2016, **17**, 2118.

22 C.-C. Li and B. Liu, *Briefings in Bioinformatics*, 2020, **21**, 2133–2141.

23 M. Shafiq, Z. Tian, A. K. Bashir, X. Du and M. Guizani, *IEEE Internet of Things Journal*, 2020, **8**, 3242–3254.

24 Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah and F. Ahmad, *Transactions on Emerging Telecommunications Technologies*, 2021, **32**, e4150.

25 F. Hussain, S. A. Hassan, R. Hussain and E. Hossain, *IEEE Communications Surveys & Tutorials*, 2020, **22**, 1251–1275.

26 J. E. Saal, A. O. Oliynyk and B. Meredig, *Annual Review of Materials Research*, 2020, **50**,.

27 G. H. Gu, J. Noh, I. Kim and Y. Jung, *Journal of Materials Chemistry A*, 2019, **7**, 17096–17117.

28 Z. Xiong, Y. Cui, Z. Liu, Y. Zhao, M. Hu and J. Hu, *Computational Materials Science*, 2020, **171**, 109203.

29 S. A. Tawfik, M. Rashid, S. Gupta, S. P. Russo, T. R. Walsh and S. Venkatesh, *npj Comput Mater*, 2023, **9**, 5.

30 C. S. Adorf, T. C. Moore, Y. J. Melle and S. C. Glotzer, *The Journal of Physical Chemistry B*, 2019, **124**, 69–78.

31 Y. Wang, W. Deng, Z. Huang and S. Li, *The Journal of Chemical Physics*, 2022, **156**, 154504.

32 Z.-H. Zhou, *Machine learning*, Springer Nature, 2021.

33 M. I. Jordan and T. M. Mitchell, *Science*, 2015, **349**, 255–260.

34 H. Wang, Z. Lei, X. Zhang, B. Zhou and J. Peng, *Deep Learn*, 2016, 98–164.

35 H. B. Barlow, *Neural computation*, 1989, **1**, 295–311.

36 Z. Ghahramani, *Advanced Lectures on Machine Learning: ML Summer Schools 2003, Canberra, Australia, February 2-14, 2003, Tübingen, Germany, August 4-16, 2003, Revised Lectures*, 2004, 72–112.

37 T. Berg-Kirkpatrick, A. Bouchard-Côté, J. DeNero and D. Klein, Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, 2010, pp. 582–590.

38 P. Cunningham, M. Cord and S. J. Delany, *Machine learning techniques for multimedia*, Springer, 2008, pp. 21–49.

39 T. Hastie, R. Tibshirani, J. Friedman, T. Hastie, R. Tibshirani and J. Friedman, *The elements of statistical learning: Data mining, inference, and prediction*, 2009, 9–41.

40 B. Liu and B. Liu, *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*, 2011, 63–132.

41 D. Bhattacharya and T. K. Patra, *Macromolecules*, 2021, **54**, 3065–3074.

42 R. B. Jadrich, B. A. Lindquist, W. D. Piñeros, D. Banerjee and T. M. Truskett, *The Journal of Chemical Physics*, 2018, **149**, 194110.

43 A. W. Long, J. Zhang, S. Granick and A. L. Ferguson, *Soft Matter*, 2015, **11**, 8141.

44 W. F. Reinhart, A. W. Long, M. P. Howard, A. L. Ferguson and A. Z. Panagiotopoulos, *Soft Matter*, 2017, **13**, 4733.

45 A. Statt, D. C. Kleeblatt and W. F. Reinhart, *Soft Matter*, 2021, **17**, 7697.

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

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

48 C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2818–2826.

49 L. McInnes, J. Healy and J. Melville, *arXiv preprint arXiv:1802.03426*, 2018.

50 L. McInnes, J. Healy and S. Astels, *J. Open Source Softw.*, 2017, **2**, 205.

51 R. Mao, J. O'Leary, A. Mesbah and J. Mittal, *JACS Au*, 2022, **2**, 1818–1828.

52 D. Samak, A. Fisher and D. Rittel, *CIRP Annals*, 2007, **56**, 149–152.

53 R. Bostanabad, *Computer-Aided Design*, 2020, **128**, 102906.