Machine Learning for Phase Behavior in Active Matter Systems

Austin R. Dulaney and John F. Brady

Division of Chemistry and Chemical Engineering, California Institute of Technology

(Dated: 19 November 2020)

We demonstrate that deep learning techniques can be used to predict motility induced phase separation (MIPS) in suspensions of active Brownian particles (ABPs) by creating a notion of phase at the particle level. Using a fully connected network in conjunction with a graph neural network we use individual particle features to predict to which phase a particle belongs. From this, we are able to compute the fraction of dilute particles to determine if the system is in the homogeneous dilute, dense, or coexistence region. Our predictions are compared against the MIPS binodal computed from simulation. The strong agreement between the two suggests that machine learning provides an effective way to determine the phase behavior of ABPs and could prove useful for determining more complex phase diagrams.

I. INTRODUCTION

Since its inception, the field of active matter has been dominated by studies of motility-induced phase separation (MIPS). The majority of these studies focus on developing a theoretical framework to describe clustering behavior and the accumulation of active particles at boundaries. Due to the striking similarities between classical and active phase behavior, the creation of thermodynamic-like frameworks has been of particular interest but continues to be a source of debate. A key difficulty surrounding this approach is the lack of a well-defined notion of temperature and free energy—as these systems are far from equilibrium—which results from the intrinsic swimming motion of active particles.

Adhering to the structure of traditional thermodynamic frameworks has resulted in several definitions of a non-equilibrium chemical potential, each of which predicts an active binodal but fails to predict the correct coexistence pressure measured inside the phase envelope from simulation. The shortcomings with the current chemical potential definitions do not preclude its existence but necessitate alternative measures for determining the phase boundaries. Large scale computer simulations provide a means to computing system pressure, which can provide insights into the phase behavior through the mechanical instability criterion. While this method is robust and has shown great success, it inherently has a steep trade-off between accuracy and computational cost. Determination of the phase boundary requires the change in system pressure with volume fraction to be zero. To make such a judgment one either needs to finely sweep volume fraction space or rely on fitting functions to smoothly fit the pressure data. Both methods are highly dependent on the quality of the pressure data obtained at each point in phase space, and large fluctuations in active pressure make this a difficult task, especially deep in the coexistence region.

To overcome these limitations we turn towards methods used to characterize other inherently complex materials. Recently, there has been a surge of interest in using machine learning algorithms to aid in material characterization. While early studies were predominantly interested in materials containing explicit symmetries or those confined to two-dimensional lattices, there has been some development in classifying amorphous materials.

In this study, we leverage the developments in machine learning to aid in characterizing the observed phase behavior in suspensions of active Brownian particles (ABPs). ABPs are an important minimal model system for determining the behavior of self-propelled colloids, bacteria, and other living organisms. The key feature that distinguishes an active colloid from a passive one is the driven and persistent nature of its motion. This distinct characteristic of its dynamics gives rise to a wealth of interesting behaviors including self-assembly, clustering, and motility-induced phase separation. As such, active materials have garnered interest from the chemical and material science communities for novel drug delivery methods, remediation strategies, and material design methods at the microscale.

Due to the nonequilibrium nature of these systems, it is difficult to develop analytic theories that can accurately predict the more complex collective behaviors. Thus, we look towards machine learning to aid in this endeavor. Machine learning algorithms are capable of discerning difficult—and potentially nonintuitive—nonlinear relationships among system variables, which would otherwise go unnoticed. These algorithms also have the benefit of readily handling multi-body correlations, which are exceptionally taxing or intractable through traditional analytic means.

Using a combination of deep learning and large-scale simulation, we focus on characterizing the phase behavior of particles in a suspension of active Brownian disks. We use machine learning to predict particle phase at a per particle level for simulations conducted at different regions in phase space. We then use these phase labels to get an estimate for the fraction of particles in each phase present at a given point in phase space. This fraction is then used to determine the system phase behavior. The manuscript is outlined as follows. In section II A we


a)Electronic mail: jfbrady@caltech.edu
define the implementation of the active Brownian particle model used in our simulations. We then outline the datasets generated for use in our machine learning model in section II B. Here we also discuss the feature selection used for our machine learning model. In section II C, we describe the machine learning model architecture used in this work and provide details on the training procedures. In section II D, we discuss the input features used for our model. In section II E, we discuss the representation of our simulation snapshots as graphs. In section II F, we present our results in the form of predictions of the phase behavior for suspensions of ABPs at different regions in the phase diagram. Finally, in section III we discuss the implications of this work and future directions.

II. METHODS

A. Simulation Details

Suspensions of monodispersed, purely active particles are modeled using the active Brownian particle (ABP) model. The active motion is characterized by an intrinsic swim velocity $\mathbf{U}_0 = U_0 \mathbf{q}$—where $\mathbf{q}$ is the particle orientation—which reorients on a timescale $\tau_R$. Particles of radius $a$ interact through a Weeks-Chandler-Anderson (WCA) potential with cutoff radius $r_{cut} = (2a)^{1/6}$ and depth $\epsilon = 200 F_{swim} a$, where $F_{swim} = U_0$ is the magnitude of the force resulting from the product of the translational drag $\zeta$ and swim velocity. Here we assume particles reorient via a stochastic torque $\mathbf{L}_R$ governed by zero-mean white noise statistics with variance $2\zeta_R \delta(t)/\tau_R$, where $\zeta_R$ is the rotational drag coefficient. Particles positions and orientations can be evolved in time using overdamped Langevin dynamics

$$0 = -\zeta \mathbf{U}_i + F_{swim}^i + \sum_{i\neq j} F_{ij},$$
$$0 = -\zeta_R \mathbf{L}_{R} + L_R^i,$$

where $F_{ij} = \zeta U_0 \mathbf{q}_i$ is the swim force of particle $i$, $F_{ij}^P$ is the interparticle force between pair $i,j$, $\mathbf{U}_i$ is the velocity, $\mathbf{L}_R$ is the angular velocity, and $\zeta$ and $\zeta_R$ are the translational and rotational drags, respectively. The angular velocity is related to the particle orientation by $\partial \mathbf{q}_i / \partial t = \mathbf{L}_R \times \mathbf{q}_i$. Normalizing position and time by $a$ and $\tau_R$, results in the dimensionless reorientation Péclet number $Pe_R \equiv a/l$, which is the ratio of a particle’s size to its persistence length $l = U_0 \tau_R$—the distance traveled between reorientation events.

We performed independent simulations of 40,000 particles for 10,000$\tau_R$ for various combinations of the two governing nondimensional parameters: the packing fraction $\phi$ and $Pe_R$. To avoid introducing an additional force scale $Pe_R$ was varied by changing $\tau_R$ at a fixed value of $U_0$. All simulations were conducted using the HOOMD-Blue software package.23 Hydrodynamic interactions have been neglected.

B. Datasets

Our machine learning model is structured to predict phase identity at a per particle level, similar to what was done by Ha et al.22 This results in a binary classification task in which particles can be members of the gas phase or the dense phase. For simplicity, we ignore the second-order hexatic transition present in two-dimensional hard disk systems and treat the hexatic phase as part of the dense phase.

We use the simulations outlined in section II A to produce datasets for each point in phase space represented by a $(\phi, Pe_R)$ pair. For each of these phase points, we look at 6 snapshots spaced roughly 1,000$\tau_R$ apart. From these snapshots, we construct a feature set for each phase point which consists of 240,000 entries. Predictions of the phase behavior at each phase point are averaged across each of the 6 time points to reduce bias from a single configuration.

C. Learning Framework

Here we give a brief overview of neural networks and describe the architecture and training routine used in this work.

Neural networks have shown great potential for predicting particle phase for both two-state and amorphous phase-separated systems.13,14 Most common neural network employed is the fully connected feedforward network. Feedforward networks are composed of layers of transformations modified by nonlinear functions. These layers can be stacked resulting in the output of one layer acting as the input of the following layer. The basic form for a layer $f$ is $f(x) = g(Wx + b)$, where $g$ is the nonlinearity activation function, $x$ is the vector input data, $W$ is the weight matrix, and $b$ is a vector of biases. When constructing a fully connected network the activation functions $g$ for each layer need not be the same, and additional regularization terms can be added to prevent overfitting to training data. Some common activation functions are the sigmoid, hyperbolic tangent, and rectified linear unit (ReLU), defined as $g(x) = max(0, x)$. Once constructed, a network is given an objective, or loss, function to minimize and updates the weight and bias terms through either gradient descent13,22 or a more sophisticated algorithm like Adam.24 Here we are interested in a binary classification and thus use binary cross entropy to compute loss

$$L = -(y \log(p) + (1 - y) \log(1 - p)),$$
Particle labels for a simulation of interest. These predictions are made using a GNN to provide information on the local structure to the network.

We avoid this when looking at MIPS in active disks by using a GNN to provide information on the local structure to the network.

Our training and model architecture is as follows. We first train a supervised deep neural network (DNN) on data in the single-phase region above the critical point. After the supervised network is trained, we predict particle labels for a simulation of interest. These predictions are then taken and those that predict the phase with a >90% confidence are used as seed labels in a semi-supervised GNN. We then take the simulation snapshot and represent it as a graph, which we will discuss in more detail in section I. We then train a GNN for each graph. The training in this step is structured as a transductive, or semi-supervised, learning problem. For each graph we use the seeded particle (node) labels to propagate labels to the remainder of the graph. We use the same features from the DNN, but instead of learning a very general problem, we are using confidently labeled particles to influence the labels given to their neighbors. In this work, we use the graph attention network (GAT) architecture for our GNNs implemented using the DGL software package. The resulting prediction from the GNN is then weighted against the prediction provided by the DNN in the first step.

A flowchart of our learning process is outlined in Fig. 1. The purple, orange, and teal lines of the GAT convolution represent the different attention heads for the layer. Each attention head serves as a means to create feature abstractions. The coefficients $\alpha_{i,j}$ are learned weight parameters which determine the weighted importance of neighbor $j$ on particle $i$. The attention heads from each node are then concatenated or averaged to produce the layer output, which may be a label probability or feature abstraction. Details of the GAT implementation can be found in reference 26. Further details of the model architecture used in this work are presented in appendix B.

FIG. 1. The learning architecture used in this work to predict particle phase labels. First, a particle feature matrix is fed into a fully connected DNN. Simultaneously particles are connected to form a graph structure. The graph is partially labeled using the most confident (>90%) labels from the DNN and is then used with the feature matrix as inputs into a GNN consisting of three GAT convolution layers with a final softmax activation function. The resulting label probabilities from the GNN are then averaged with the label probabilities output from the DNN to achieve the final label probabilities. Each particle is then given the most probable label.

D. Feature Selection

In order to label individual particles, our feature space is limited to quantities that can be computed on a per-particle basis. This includes Voronoi volume, the number of first shell Voronoi neighbors, and the average of first shell Voronoi volumes. We repeat this averaging process for the second and third shell neighbors as well to incorporate information about the local environment. We also include the hexatic and translational order parameters defined as $\psi_6(i) = 1/n \sum_j e^{i6\theta_{ij}}$ and $G_6(r_{ij}) = \sum_j \psi_6(i) \cdot \psi_6(j)$, respectively, where $n$ is the number of Voronoi neighbors, $r_{ij}$ is the vector connecting pair $ij$, and $\theta_{ij}$ is the angle between $r_{ij}$ and the reference vector $(0, 1)$, and $\psi_6(i)$ is the complex conjugate of the hexatic order parameter. The Voronoi volumes and the hexatic and translational order parameters were computed using the Freud analysis software. In order to account for some of the dynamics we include the force-
fluctuations, which results in longer-range morphologi-
face, but they are not as sharp. The increased system
that the graph representation still has bumps on the sur-
look at a primarily dense system [Fig. 2(b),(f)], we see
ing and any structure present is short-range. If we next
by a uniform graph with a rough surface. The disor-
acteristics. The gas phase [see Fig. 2(a),(e)] is marked
phases with their corresponding
dimensional, toroidal shape as depicted in Fig. 2(e)–(h).

to be the distances between particles we obtain a three-
anted graph. If the edges are then constrained
neighbors. With periodic boundaries, this results in a
representing the system as a graph.

For each simulation snapshot, we represent the system
as a graph where each particle is a node in the graph
and connections are made between first shell Voronoi
neighbors. With periodic boundaries, this results in a
fully-connected graph. If the edges are then constrained
to be the distances between particles we obtain a three-
dimensional, toroidal shape as depicted in Fig. 2(e)–(h).

In Fig. 2 we present simulation snapshots at different

t of the phase diagram with their corresponding

II. GRAPH REPRESENTATION

The MIPS transition is markedly similar to the liquid-
vapor transition seen in traditional thermodynamic fluids
with the two coexisting phases both being disordered. In
thermodynamic fluids one could measure local density and
use spatial density discontinuities to distinguish be-
tween the coexisting phases. However, this is difficult
to do in practice as we are constrained to finite systems
in simulations. Regions close to the critical point are
subject to large density fluctuations which make it dif-

to observe persistent macroscopic phase domains.
Therefore, we need an alternative way to gather this similar
type of local structure in the system. We do this by representing the system as a graph.

For each simulation snapshot, we represent the system
as a graph where each particle is a node in the graph
and connections are made between first shell Voronoi
neighbors. With periodic boundaries, this results in a
fully-connected graph. If the edges are then constrained
to be the distances between particles we obtain a three-
dimensional, toroidal shape as depicted in Fig. 2(e)–(h).

In Fig. 2 we present simulation snapshots at different

t of the phase diagram with their corresponding

III. RESULTS

Here we present the results from our machine learning
model. Our model was trained on very dilute (\( \phi < 0.2 \)) or
very dense (\( \phi > 0.7 \)) phase points above the critical point and deep within the coexistence region. Training points from deep within the coexistence region were labeled by inspection and particles near phase interfaces were not used for training. The model was then used to predict particle phase below the critical point (\( Pe_{\text{crit}} \sim 0.047 \)). Figure 3 presents the snapshots of particles shown in Fig. 2(a)–(d) colored by their predicted phase labels. It can be seen that our model is highly capable of distin-
guishing particle phase in the homogeneous phase and
deep within the coexistence region. The most challeng-
ing region is near the critical point as these particles are more difficult to readily distinguish from inspection.

Therefore, to evaluate performance in this region we take the predicted particle labels for each phase point and compute the fraction of dilute particles \( F_g \) present and average this across all \( T = 6 \) time points for a given (\( Pe_R, \phi \)) pair. This is represented by

\[
F_g = \frac{1}{T} \sum_{t=0}^{T} \frac{1 - \sum_{j}^{N} y_{ij}}{N},
\]

where \( y_{ij} \) is the predicted label of particle \( j \) and \( N \) is the total number of particles. In our model the positive case is the dense phase (\( y_{ij} = 1 \)) and the null case is the dilute phase (\( y_{ij} = 0 \)). To account for small fluctuations in prediction we consider a point to be in the dilute region if \( F_g > 95\% \) and to be in the dense region if \( F_g < 5\% \).

Any other value of \( F_g \) is labeled as coexisting as we are only considering points below the critical point (\( Pe_{\text{crit}} \sim 0.0468 \)).
FIG. 2. Simulation snapshots and respective graph structures for different regions of phase space colored by particle Voronoi volume. We look at the weakly active \( (Pc_R \sim 0.11) \) (a),(e) dilute and (b),(f) dense regions, (c),(g) the region near the critical point \( (P_R \sim 0.047) \), and (d),(h) deep within the coexistence region \( (Pc_R \sim 0.011) \).

FIG. 3. Simulation snapshots for different regions of the phase diagram with particles colored based on their predicted phase.

Figure 4 presents the MIPS phase diagram with points colored based on which phase the system is predicted to be in using our machine learning model. We compare the predicted phase against the binodal predicted by Takatori and Brady (purple dashed line) and the binodal computed from slab simulations (purple points). The solid purple line is a fourth-order polynomial fit of the computed binodal. The shaded region represents the range of this fit \( \pm 1 \) standard deviation for each fitting parameter. The spinodal predicted by Takatori and Brady (black dash-dotted line) is shown for completeness. We find remarkable agreement between the binodal obtained from simulations and our machine learning predictions.

From Fig. 4 it is clear that predicting dilute particles is more challenging than predicting dense particles. We suspect this difficulty arises from the large tail in the distribution of Voronoi densities for particles in the dilute phase. Ha et al. observed this type of large overlap for particle density distributions when studying the phase behavior of a Lennard-Jones fluid.\(^{22}\)

V. CONCLUSIONS

We have created a machine learning model to predict the phase identity of individual active Brownian particles. Our results indicate that single-particle parameters are sufficient for learning particle phase when some amount of structure is included in the system. We have also shown that the MIPS phase transition can be predicted using this machine learning model. From our model optimization and feature analysis, we conclude that kinematic features—such as particle speed or the force-orientation correlation—are important for distinguishing the phases present in the MIPS transition (see appendix A and Fig. 6), unlike the traditional liquid-vapor transition present in thermodynamic fluids. The directed motion present in active systems results in a stronger separation for particle speeds and longer correlation lengths than would be seen in traditional systems when considering phase identity near the critical point.
Ha et al. were able to successfully characterize particle phase of a Lennard-Jones fluid with high accuracy using a convolutional neural network and only three structural features, but we find that our model performance steeply drops off near the critical point if we do not include at least one of the kinematic features mentioned above.

We have demonstrated that the local structure plays an important role in determining the phase behavior of active systems. Our graph representations of the system possess unique characteristics specific to their region of the phase diagram—which can be learned using a general graph neural network framework with attention. This matches the results from Swanson et al. and Ha et al. who included structure via a message-passing network and convolutional neural network, respectively, to characterize amorphous materials.

We believe machine learning can be used for more challenging classification problems. It would be straightforward to extend our framework to also distinguish between the hexatic crystalline phase and the disordered dense phase to produce a more complete phase diagram. Our model is already capable of learning the importance of the third shell average Voronoi volumes, which act as a surrogate for the third peak in the radial distribution function. This peak provides a way to distinguish between liquid and solid phases. We also feel a more specific model could be created to directly predict which region of the phase diagram the system is in by performing classification at the graph-level instead of the node-level (as was done in this work). A graph-level classifier can then be readily generalized using an unsupervised learning scheme, where the model is learning distinctions between the present phases.

The learning architecture used here should also readily generalize to active systems with thermal noise, polydispersity, or higher dimensionality. These deviations from the problem focused on in this work would result in different distributions for feature values, but should still maintain similar relationships between features. The graph network can be further extended to include edge features, which would allow for more complicated or varied inter-particle interactions and should prove to be a useful tool in the characterization of other amorphous systems.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ACKNOWLEDGMENTS

A.R.D. would like to thank Yisong Yue for thoughtful discussions pertaining to graph neural networks. J.F.B. acknowledges support by the National Science Foundation under Grant No. CBET-1803662. We gratefully acknowledge the support of the NVIDIA Corporation for the donation of the Titan V GPU used to carry out this work.

Appendix A: Feature Correlation and Importance

The correlation matrix for our full initial feature set is presented at the top of Fig. 5. In the figure we have used a shorthand notation where \( \phi \) is the Voronoi volume fraction, \( \phi_i \) is the Voronoi volume fraction averaged over the \( i^{th} \) shell neighbors, \( N_i \) is the number of neighbors in shell \( i \), \( U \) is the particle speed, \( \mathbf{F} \cdot \mathbf{q} \) is the force-orientation correlation, \( \psi_b \) is the hexatic order parameter, and \( G_6 \) is the translational order parameter. The hexatic and translational order parameters are broken into their real part, imaginary part, magnitude, and angular components represented by \( \Re(\cdot) \), \( \Im(\cdot) \), \( |\cdot| \), and \( \langle \cdot \rangle_{6,6} \), respectively. Features were dropped in order of the strength of the measured collinearity with other features. When considering a pair of collinear features, the feature that contributes the least to the total importance is removed.

We use a simple boosted random forest to compute feature importance. Our random forest classifier is made up of 1000 estimators, with a max decision tree depth of 8, and trained for 30 epochs with early stopping. Our
boosted random forest is implemented in XGBoost. This classifier is then used to compute the SHAP feature importance (see Fig. 6). The color in Fig. 6 indicates the value of the feature in the line, and the actual SHAP value indicates how important a feature value was for predicting the positive (dense) case. As an example, from Fig. 6 (top) we see that $\phi_3$ is the most indicative feature, and large values of this feature strongly indicate that the particle is dense, whereas very low values indicate that the particle in question is likely dilute. The SHAP analysis for the full feature set is not very insightful due to the presence of strong collinearity, but it can still be used to determine which feature to drop from a pair of highly collinear features. After removing a feature the importance is recalculated as this can change as the feature set changes. The final correlation matrix for the features used in this work is shown at the bottom of Fig. 5 and the final feature SHAP values are shown in the bottom of Fig. 6. There is greater diversity in the SHAP values obtained, and now the volume fraction as the most important feature, which is in line with our physical intuition.

Appendix B: Model and Training Details

The trained DNN used in this work is 5 layers with batch normalization and dropout on some of the layers for regularization. The number of layers in the
TABLE II. The architecture of the graph network portion of the neural network used for the results presented in this work.

| Layer | Size | Activation | Batch Norm | Dropout |
|-------|------|------------|------------|---------|
| 1     | 128  | ReLU       | –          | –       |
| 2     | 128  | LeakyReLU* | True       | 0.69    |
| 3     | 128  | LeakyReLU* | True       | 0.35    |
| 4     | 64   | LeakyReLU* | –          | 0.75    |
| 5     | 2    | SoftMax    | –          | –       |

*LeakyReLU activation function has negative slope \( \alpha = 0.1 \).

TABLE II. The architecture of the graph network portion of our model.

| Layer | Size | Activation | Attention Heads |
|-------|------|------------|-----------------|
| 1     | 8    | LeakyReLU* | 2               |
| 2     | 8    | LeakyReLU* | 2               |
| 3     | 8    | LeakyReLU* | 2               |
| 4*    | 2    | SoftMax    | –               |

*LeakyReLU activation function has negative slope \( \alpha = 0.2 \).

\( \text{b} \) This is a fully-connected layer used to get the final prediction.

The GNN model architecture used in this work was explored manually. The graph network is intentionally kept small as this was shown by Velicković et al. to be effective at transductive learning.25 The parameters of our GNN are shown in Table II. Each layer in the network is a GAT convolution layer except for the last one, which is a fully-connected layer to give the outputs.

REFERENCES

1. Y. Fily and M. C. Marchetti, “Athermal Phase Separation of Self-Propelled Particles with No Alignment,” Physical Review Letters 108, 235702 (2012).

2. C. Takatori and J. F. Brady, “Towards a thermodynamics of active matter,” Physical Review E 91, 032117 (2015).

3. D. Levis, J. Codina, and I. Pagonabarraga, “Active Brownian equation of state: metastability and phase coexistence,” Soft Matter 13, 8113–8119 (2017).

4. J. U. Klamser, S. C. Kapfer, and W. Krauth, “Thermodynamic phases in two-dimensional active matter,” Nature Communications 9 (2018), 10.1038/s41467-018-07491-5.

5. A. P. Solon, J. Stenhammar, M. E. Cates, Y. Kafri, and J. Tailleur, “Generalized thermodynamics of motility-induced phase separation: phase equilibria, Laplace pressure, and change of ensembles,” New Journal of Physics 20, 075001 (2018).

6. C. Takatori, W. Yan, and J. F. Brady, “Swim Pressure: Stress Generation in Active Matter,” Physical Review Letters 113, 028103 (2014).

7. S. Chakraborti, S. Mishra, and P. Pradhan, “Additivity, density fluctuations, and nonequilibrium thermodynamics for active Brownian particles,” Physical Review E 93, 052606 (2016).

8. S. Palival, J. Rodenburg, R. van Roij, and M. Dijkstra, “Chemical potential in active systems: predicting phase equilibrium from bulk equations of state?” New Journal of Physics 20, 015003 (2018).

9. Y. Fily, Y. Kafri, A. P. Solon, J. Tailleur, and A. Turner, “Mechanical pressure and momentum conservation in dry active matter,” Journal of Physics A: Mathematical and Theoretical 51 (2018), 10.1088/1751-8121/aa9966.

10. A. Patch, D. M. Sussman, D. Yllanes, and M. C. Marchetti, “Curvature-dependent tension and tangential flows at the interface of motility-induced phases,” Soft Matter 14, 7435–7445 (2018).

11. I. Carrasquilla and R. G. Melko, “Machine learning phases of matter,” Nature Physics 13, 431–434 (2017).

12. P. L. van Nieuwenburg, Y.-H. Liu, and S. D. Huber, “Learning phase transitions by confusion,” Nature Physics 13, 435–439 (2017).

13. P. Suchsland and S. Wessel, “Parameter diagnostics of phases and phase transition learning by neural networks,” Physical Review E 97, 174435 (2018).

14. K. Swanson, S. Trivedi, J. Lequieu, K. Swanson, and R. Kondor, “Deep learning for automated classification and characterization of amorphous materials,” Soft Matter 16, 435–446 (2020).

15. A. Mallory, C. Valeriani, and A. Cacciuto, “An Active Approach to Colloidal Self-Assembly,” Annual Review of Physical Chemistry 69, 59–79 (2018).

16. J. Palacci, S. Sacanna, A. P. Steinberg, D. J. Pine, and P. M. Chaikin, “Living Crystals of Light-Activated Colloidal Surfers,” Physical Review E 93, 052606 (2016).

17. C. Bechinger, R. Di Leonardo, H. Löwen, C. Reichhardt, G. Volpe, and G. Volpe, “Active Particles in Complex and Crowded Environments,” Reviews of Modern Physics 88, 045006 (2016).

18. W. Gao and J. Wang, “The Environmental Impact of Micro/Nanomachines: A Review,” ACS Nano 8, 3170–3180 (2014).

19. E. Ebbens, “Active colloids: Progress and challenges towards realising autonomous applications,” Current Opinion in Colloid & Interface Science 21, 14–23 (2016).

20. A. Anderson, C. D. Lorenz, and A. Travesset, “General purpose molecular dynamics simulations fully implemented on graphics processing units,” Journal of Computational Physics 227, 5342–5359 (2008).

21. I. Glaser, T. D. Nguyen, J. A. Anderson, P. Lui, F. Spiga, J. A. Millan, D. C. Morse, and S. C. Glotzer, “Strong scaling of general-purpose molecular dynamics simulations on GPUs,” Computer Physics Communications 192, 97–107 (2015).

22. M. Y. Ha, T. J. Yoon, T. Tlusty, Y. Jho, and W. B. Lee, “Widom Delta of Supercritical Gas-Liquid Coexistence,” Journal of Physical Chemistry Letters 9, 1734–1738 (2018).

23. Gradient descent is an iterative method for locating the local minimum of a function by determining the steepest gradient of the function with respect to its independent variables. In this context gradient descent is used with backpropagation, which relates the weights of each layer in the neural network back to the loss function so the entire network can be updated during each iteration.

24. D. P. Kingma and J. L. Ba, “Adam: A method for stochastic optimization,” in 3rd International Conference on Learning Rep-
resentations, ICLR 2015 - Conference Track Proceedings (International Conference on Learning Representations, ICLR, 2015).  

25 T. N. Kipf and M. Welling, “Semi-Supervised Classification with Graph Convolutional Networks,” (2016).  

26 P. Veličković, A. Casanova, P. Liò, G. Cucurull, A. Romero, and Y. Bengio, “Graph attention networks,” in 6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings (International Conference on Learning Representations, ICLR, 2018).  

27 Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, “A Comprehensive Survey on Graph Neural Networks,” (2019).  

28 M. Wang, D. Zheng, Z. Ye, Q. Gan, M. Li, X. Song, J. Zhou, C. Ma, L. Yu, Y. Gai, T. Xiao, T. He, G. Karypis, J. Li, and Z. Zhang, “Deep Graph Library: A Graph-Centric, Highly-Performant Package for Graph Neural Networks,” (2019).  

29 V. Ramasubramani, B. D. Dice, E. S. Harper, M. P. Spellings, J. A. Anderson, and S. C. Glotzer, “freud: A Software Suite for High Throughput Analysis of Particle Simulation Data,” Computer Physics Communications 254 (2019), 10.1016/j.cpc.2020.107275.  

30 S. M. Lundberg, P. G. Allen, and S.-I. Lee. “A Unified Approach to Interpreting Model Predictions,” Tech. Rep. (2017).  

31 J. Bergstra, D. Yamins, and D. D. Cox, “Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures,” Tech. Rep. (2013).