Title
Learning-Based Characterization and Control of Colloidal Self-Assembly Systems

Author
O'Leary, Jared

Publication Date
2022

Peer reviewed|Thesis/dissertation

Learning-Based Characterization and Control of Colloidal Self-Assembly Systems

by

Jared O'Leary

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Chemical Engineering

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Ali Mesbah, Co-chair
Professor Clayton Radke, Co-chair
Professor Joelle Frechette

Summer 2022

Learning-Based Characterization and Control of Colloidal Self-Assembly Systems

Abstract

Learning-Based Characterization and Control of Colloidal Self-Assembly Systems

by

Jared O'Leary

Doctor of Philosophy in Chemical Engineering

University of California, Berkeley

Professor Ali Mesbah, Co-chair

Professor Clayton Radke, Co-chair

Colloidal self-assembly (colloidal SA) is the process by which particles in solution spontaneously organize into an ordered structure. The spontaneous self-organization central to colloidal SA enables "bottom-up" materials synthesis, which would allow for manufacturing advanced, highly ordered crystalline structures in an inherently parallelizable and cost-effective manner. Thus, colloidal SA can create new avenues for highly scalable, economical manufacturing of novel metamaterials with unique optical, electrical, or mechanical properties. Colloidal SA is an inherently stochastic (i.e., random) process prone to kinetic arrest due to particle Brownian motion. This leads to variability in materials synthesis and possibly high defect rates, which can severely compromise the viability of using SA to manufacture advanced materials reproducibly. Successful implementation of colloidal SA thus critically hinges on the ability to avoid defective, kinetically arrested configurations and consistently reach highly-ordered, often defect-free states that tend to exist within global minima on the free energy landscape.

The thermodynamic and kinetic driving forces that govern colloidal SA thus need to be precisely modulated – by actively exploiting intermolecular forces, selective template or surface geometries, and/or external fields such as temperature and pressure – to direct colloidal SA systems consistently and efficiently towards mass-producible structures and materials. Two major strategies for this precise modulation are *particle design*, which involves designing the colloidal SA system such that specific inter-particle interactions ensure the high probability realization of a desired configuration, and *control*, which seeks to modulate external actuators systematically based on real-time measurements in order to induce global colloidal SA configuration changes. The primary objective of this thesis is to enable more effective particle design and control of colloidal SA systems. To this end, this thesis investigates strategies based on machine learning and optimal control for quantifying and classifying

colloidal SA system states, learning tractable stochastic dynamical models of colloidal SA dynamics, and learning control policies that dynamically change external actuators to guide colloidal SA. The insights gained from these methods provide a deeper mechanistic understanding of colloidal SA and contribute to an ever-developing archive of methods that can be used or expanded upon to achieve reproducible colloidal SA.

To the Las Vegas Raiders of Oakland and the Black Hole – the first group of people to show me the power of irrational hope. Throw deep, baby!

Contents

List of Figures

List of Tables

Acknowledgments

I am extremely grateful for my PhD advisor, Prof. Ali Mesbah for his continued support and guidance – including the countless hours of discussions about my writing and presentation skills, critical thinking, and research strategies. He said in the beginning of my time at Berkeley that a PhD could fundamentally transform my thinking and therefore my life. It took me 6 years to believe him, and I could not be happier about the journey that led me to this realization. Most importantly, it is no secret that I struggled heavily during the first few years of my PhD. During this time, Prof. Mesbah continued to believe in and advocate for me when maybe he rationally shouldn't have. And I will never forget that.

I would next like to thank Prof. Joel Paulson. Joel provided tremendous support to me throughout his time at Berkeley and even after he left Berkeley and joined the faculty at The Ohio State University. I very often reached out to Joel about random research questions and he provided extremely in-depth and informative answers to such questions. These discussions were not only critical for me to improve as a researcher but also provided a "gold" standard to which I could compare my own critical thinking abilities.

I would also to thank our collaborators from Texas A&M and Lehigh University, Prof. Jeetain Mittal and Runfang Mao. They provided extremely valuable insights about the fundamentals of colloidal self-assembly and played a critical role in guiding much of the research in this thesis. I hope the collaboration between the Mittal and Mesbah groups continues for a long time after I graduate.

I would like to thank the members of qualifying and thesis committee: Profs. Karanthi Mandadpu, Mark Mueller, David Schaffer, Clayton Radke, and Joelle Frechette. Critically, my qualifying committee made me realize the importance of investigating the "end-to-end" research problem, as opposed to hyper-focusing on aspects of problems that I felt most directly fell under my expertise/responsibility. I additionally thank the members of thesis committee for reading and providing feedback on this lengthy document.

I would like to thank Carlet Altamirano and Clay Batton. Carlet helped me through my many administrative mishaps during my PhD and always cheered me up while doing so. Clay has been a great friend throughout my time at Berkeley and even provided me occasional research/computational tool guidance.

Finally, I was so lucky to have the support of many friends over the last 6 years. I'd like to thank them all – especially Alfonso, Scalise, Benvenue, Otto, Kate, Looch, Janani, Dev, Jay, Aaron, Mez, Tori, Ian, Tim, Goldee with a capital G, Sue, Anita, Nelly, and my partner Erin. Vote Janani Ramachandran for Oakland City Council District 4!

Chapter 1

Introduction

This chapter first introduces colloidal self-assembly as a promising route for cost-effective and scalable manufacturing of advanced, highly-ordered materials. This chapter next explains how the Brownian particle motion that governs colloidal self-assembly renders the process intrinsically stochastic and prone to kinetic arrest at defective configurations. This chapter then introduces two potentially viable routes towards reproducible colloidal self-assembly – particle design and control. The chapter is closed with an outline and discussion on the overarching objective and novel contributions of this dissertation.

1.1 The Promise of and Challenges Towards Reproducible Colloidal Self-Assembly

Colloidal self-assembly (colloidal SA) is the process by which particles in solution spontaneously organize into an ordered structure [1]. Colloidal SA is primarily facilitated by specific interactions among particles, their surrounding medium, and surfaces (whose geometry can selectively induce particle deposition). Some of these interactions can be manipulated during assembly by applying and/or dynamically modifying external fields, such as electric fields, magnetic fields, temperature, and pressure [1–3].

Widespread interest in the practical implementation of colloidal SA first arose from the semiconductor and microelectronics industries, whose desire to scale microelectronic devices past the sub-50 nm node began to challenge the viability of conventional solid-state nanofabrication technologies to mass produce devices and integrated circuits cost-effectively [4–7]. In conventional, "top-down" manufacturing processes, lithography and mechanically-directed processing techniques are used to "carve" a final structure from a larger block of matter. These nanofabrication technologies can be limited by insufficient resolution, serial processing, planar fabrication restrictions, environmental regulations to avoid particulate contamination (e.g., clean room requirements for certain steps), nano-structure distortion, and/or high material, equipment, and operating costs [8–11]. The spontaneous self-organization central to colloidal SA, on the other hand, enables "bottom-up" materials synthesis that in princi-

ple is capable of creating three-dimensional structures with sub-nanometer precision. The inherently parallel nature of colloidal SA can also improve process speed by allowing for the simultaneous production of separate structures and their components. Because colloidal SA is governed by tailorable entropic and enthalpic forces, an industrially viable version of colloidal SA can potentially incur lower operating and equipment costs than those required by conventional "top-down" techniques that tend to depend on complex, environmentally-sensitive machinery. Finally, these tailorable enthalpic and entropic forces can be employed to initiate the assembly and disassembly of structures to remove structure defects [1, 4–7, 12].

The fact that colloidal SA can begin with micro- and/or nanoscale building blocks of varying complexity indicates that this bottom-up engineering approach can be used to synthesize novel metamaterials with unique optical, electrical, or mechanical properties [13, 14]. For perspective, a sample of potential applications of colloidal SA systems includes: ever-shrinking semiconductor dielectrics [5, 7], optical circuits [15], super-lenses (i.e., lenses capable of microscopy beyond the diffraction limit) [16], cloaking materials (i.e., materials that can direct the flow of light smoothly around an object to render the object effectively invisible) [17], photocatalytic materials (e.g., materials that harness sunlight to produce hydrogen fuel from water) [18], various magnetic nano-structures for high-density data storage and sensing applications [12], and reconfigurable devices (e.g., antennas that can dynamically modify frequency and radiation properties in a reversible, controlled manner) [13, 17]. An industrially viable materials synthesis strategy must produce materials with commercially competitive defect rates (e.g., 100 ppm in semiconductors [19]), as defects can severely compromise the utility of the above materials, especially those used for optical and electrical applications [5, 12]. Low (ideally non-existent) defect rates are imperative to the real-world implementation of colloidal SA.

Brownian particle motion (i.e., random particle motion that results from random collisions between solvent and solute molecules) renders colloidal SA an inherently stochastic process – repeated experiments yield different results even in the absence of exogenous noise or disturbances [14, 20, 21]. An additional source of systemic stochasticity is the lack of mechanical isolation of colloidal SA systems (e.g., a system may be in contact with a heat bath, whose random perturbations keep the system in thermal equilibrium at some temperature) [20]. Colloidal SA systems can also become kinetically "trapped" at certain meta-stable configurations with large activation barriers (e.g., glasses, gels, and defect-containing crystals) [14, 22]. Here, the most thermodynamically stable (and often highest-value) system configuration is not observed experimentally, as high energy or diffusive barriers prevent systems from reaching equilibrium in a reasonable amount of time. Brownian particle motion, environmental perturbations, and kinetic arrest thus lead to variability in materials manufacturing and possibly high defect rates, which can severely compromise the viability of using colloidal SA to manufacture reproducibly the advanced materials listed earlier. This lack of reproducibility in turn prevents colloidal SA from achieving cost-effective and scalable manufacturing of such materials [14, 22, 23].

1.2 Colloidal Self-Assembly and the Free Energy and Diffusivity Landscapes

The free energy landscape (FEL) and the diffusivity landscape (DL) can be used to provide insights into the intrinsic system stochasticity, thermodynamic and kinetic driving forces, and tendencies towards kinetic arrest and run-to-run variability within colloidal SA systems. Cartoon diagrams of the FEL and DL for a fictional colloidal SA system are shown in Fig. 1.1. The x-axis is the self-assembly coordinate. It is an abstraction with no relation to time that measures the progress of bond-breaking and bond-forming colloidal SA steps. Each (x,y) position in Fig. 1.1 maps a specific colloidal SA system configuration to its corresponding Gibbs free energy (Fig. 1.1a) or its diffusivity (Fig. 1.1b). The FEL of a colloidal SA system will generally contain one global free energy minimum, which corresponds to the most thermodynamically favored (and often most highly-ordered) configuration of the system. Additional local free energy minima corresponding to defective configurations will also generally exist within the FEL, some of which may be more kinetically favored than the global free energy minimum [24–27].

The heights of activation energy barriers between and diffusivities near different configurations of colloidal SA systems are closely related to the kinetic favorability of each configuration, with smaller energy barriers and larger diffusivities corresponding to more kinetically favored configurations. More specifically, the constituents of colloidal SA systems constantly re-arrange themselves due to their kinetic (thermal) energy (e.g., molecular vibration, rotation, and translation). This large amount of particle motion allows colloidal SA systems to explore many configurations. Such behavior facilitates spontaneous self-organization into structures that are favored by statistical probability [20, 22]. Thus, the transition of a colloidal SA system from one configuration to another (e.g., movement along the self-assembly coordinate) is associated with a transition probability that is governed by both the inherent system stochasticity and the accessibility of a system configuration. Note that the transition probabilities are heavily influenced by the kinetic (especially diffusive) properties of the system. For colloidal SA systems with FEL activation energy barriers significantly larger than kT (e.g., Positions 2 and 4 in Fig. 1.1), inter-configuration transition probabilities can be predicted based on the activation energy barrier height and the diffusion coefficient at the activation energy barrier [28–30]. For systems with activation energy barriers in the range of kT (e.g., Positions 1 and 5 in Fig. 1.1), the diffusivity along the entire self-assembly coordinate determines the inter-configuration transition rates [28–31].

The thermodynamic favorability of different colloidal SA configurations (i.e., the free energy values of these configurations) is determined from the competition between enthalpic and entropic driving forces. For example, double-tail lipids in solution are thermodynamically favored to self-assemble to form a lipid bilayer. When individual double-tail lipid molecules are dissolved in solution, hydrogen bonds from distorted water molecules surround individual lipid molecules to create "clathrate cages" and significantly reduce system entropy (while system enthalpy is changed negligibly) [1, 32, 33]. The hydrophobic heads

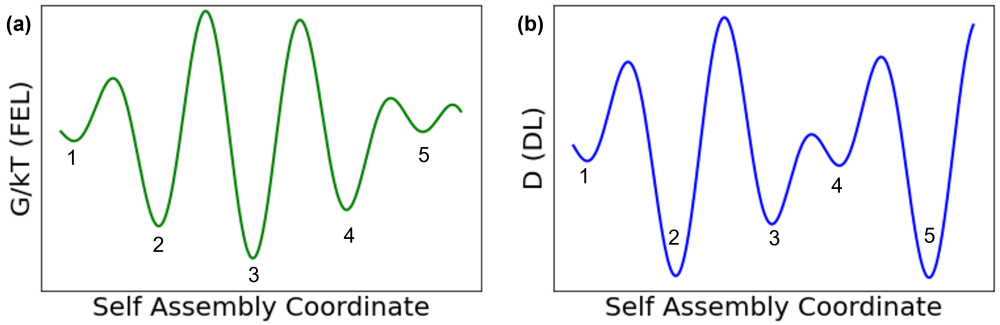


Figure 1.1: Cartoon free energy and diffusivity landscapes (FEL, DL). The x-axis is the self-assembly coordinate. The y-axes are the Gibbs free energy (G/kT) and the diffusion coefficient (D). Each (x,y) position a specific colloidal self-assembly system configuration to its corresponding Gibbs free energy or diffusivity. Numbers 1-5 denote various colloidal self-assembly system configurations that correspond to minima in the FEL. Position 3 refers to the global free energy minimum (e.g., the most thermodynamically favored colloidal state). The local minima at positions 1 and 5 both have relatively small activation barriers, but position 5 is severely diffusion-limited. As a result, the system is more likely to escape position 1 than position 5. Positions 2 and 4 are slightly more thermodynamically favorable than positions 1 and 5. The activation barriers surrounding position 2 are both relatively high and position 2 is severely diffusion-limited. Position 4 on the other hand, has both lower activation barriers and a much higher diffusivity, indicating the system is much more likely to become kinetically trapped in position 2.

of double-tail lipid molecules then interact and amalgamate to reduce their surface exposure to water. This reduced water exposure partially breaks the clathrate cage to increase the overall system entropy (and decrease the system free energy). On the other hand, tris (melamine) derivative molecules dissolved in chloroform are designed to self-assemble preferentially into discrete supramolecular assemblies over hydrogen-bonded oligomers and polymers [34]. In this colloidal SA system, the enthalpic gains from hydrogen bonding, which are specific towards discrete assembly structures, must outweigh the entropic losses from the restricted-bond rotation and enthalpic losses from steric restriction that the discrete assemblies provide. Systems with electrically or magnetically polarizable solute particles can experience induced dipoles from external (sometimes nonuniform) electric or magnetic fields [2, 35, 36]. Here, inter-particle bonding becomes thermodynamically favored if enthalpic gains from dipole-dipole interactions can overcome particle thermal energy and entropically favorable Brownian motion. In other colloidal SA systems, entropic gains from depletant particle positioning [37], enthalpic gains from Van der Waal's bonding, or enthalpic losses from electrostatic repulsion [2, 3] may also contribute to the positions of local and global free energy minima.

The kinetics of colloidal SA not only can influence but can prevent the practical formation of thermodynamically favored configurations [2, 14, 22]. The thermodynamic driving forces that favor colloidal SA may enable the formation of diffusion-limited configurations that are vulnerable to kinetic arrest within locally deep free energy minima [2, 14, 22] (e.g.,

Position 2 in Fig. 1.1). Large free energy barriers and significantly reduced particle diffusivity due to solvent-mediated friction and hydrodynamic interactions (e.g., see [31, 38]) can prevent systems from moving past kinetically arrested configurations on time scales that that are compatible with practical manufacturing needs. In this case, the colloidal SA system effectively becomes "stuck" at one of the local minima, which often correspond to meta-stable configurations with undesirable properties. For example, kinetic arrest at a local minimum of the FEL may lead to the formation of glasses, gels, disordered arrays, or defect-containing crystals instead of the desired, lowest energy structure found at the global minimum. In extreme cases, kinetic bottlenecks trap systems in local minima that are very far from the desired equilibrium phase or in severely diffusion-limited configurations that do not correspond to any local or global minimum (e.g., a configuration near Position 5 in Fig. 1.1).

The shape, concentration, size, and functional group chemistry of micro- and nano-scale particles contribute to the susceptibility of colloidal SA systems to kinetic arrest (e.g., the heights of energy barriers and particle diffusivity). For example, non-spherical dicolloids are more susceptible to kinetic arrest than spherical colloids because, in the former case, solvent cage escape trajectories require a combination of rotation and center-of-mass translation [39]. This extra movement caused by particle shape differences increases diffusive barriers to crystal nucleation, which is facilitated by mutual particle alignment. Another example is colloidal SA systems consisting of only hard nano-particle spheres that experience spontaneous crystal formation in a specific, narrow concentration range [2, 22, 40]. Concentrations below this range show multiple equilibrium phases and concentrations above this range lead to kinetic arrest via glassy suspension formation. DNA-coated micro-scale colloids show a higher likelihood of forming kinetically arrested, amorphous configurations than their nanoscale counterparts do, as the larger particles show stronger inter-molecular interactions and sharper activation barriers [41].

1.3 Particle Design and Control for Reproducible Colloidal Self-Assembly

The FEL and DL provide a unique roadmap to the global free energy minimum of a given colloidal SA system. Competing thermodynamic and kinetic driving forces and Brownian motion affect the relative free energies and inter-configuration transition probabilities of different configurations during colloidal SA. The thermodynamic and kinetic driving forces that govern SA will need to be precisely modulated – by actively exploiting intermolecular forces, selective template or surface geometries, and/or external fields – to direct SA systems consistently and efficiently towards mass-producible structures and materials [2, 3, 22]. Two major strategies for improving the reproducibility of colloidal SA – particle design and control – are introduced below.

Particles in a given colloidal SA system can be designed such that specific inter-particle

interactions ensure the high probability realization of a desired configuration [2, 14, 42]. For example, the SA of "patchy" particles exploits sticky patches coated onto small particles (e.g., complementary DNA strands on gold nano-particles [43]) to confine particle interactions along certain directions (similar to atomic bonding mechanisms). Analogously, Janus particles, whose surfaces combine incompatible elements within the same surface structure (e.g., hydrophilicity/hydrophobicity, anisotropy), have shown the potential to create more persistent macro-scale structures with fewer defects [44]. Patchiness has mostly been implemented on micrometer (and larger) sized objects, however, while SA of patchy particles in the sub-100 nm range remains challenging [2, 45]. Despite improved specificity and directional control, particle design alone does not guarantee a colloidal SA system protection from kinetic trapping or system-dependent, stochastic variations (e.g., delayed nucleation or relaxation times in crystalline systems) – all of which hinder the achievement of the high-quality, defect-free structures required for manufacturing advanced materials.

Control approaches can be designed to actuate external fields systematically to induce global colloidal SA configuration changes (e.g., turning magnetic fields on/off [46] or increasing/decreasing temperature to induce general particle crystallization/relaxation [47]). The resulting "input profiles" serve to modulate the thermodynamic and kinetic driving forces in order to guide SA systems past kinetic traps and towards desired configurations. Input profiles can be determined either heuristically or via stochastic optimization [14, 46]. In the case of open-loop control, input profiles are purely time-dependent and are determined off-line. Various studies report open-loop control (greatly) improving probabilities of achieving and stabilizing desired configurations over uncontrolled assemblies [46–54]. Although open-loop input profiles can account for general system stochasticity (e.g., via stochastic dynamical models), a lack of real-time measurements will always prevent the systems from adjusting to unmeasured and unknown stochasticity. This may prevent open-loop systems from achieving desired lowest-energy structures. Closed-loop control, on the other hand, uses input profiles that change according to real-time measurements of the system configuration. This real-time feedback gives closed-loop control methods the unique ability to account for unknown or unmeasured system uncertainties and disturbances. Closed-loop control is thus the only strategy that has the potential to account for both the inherent stochasticity (via system models or other "off-line" investigations) and unknown stochasticity (via real-time feedback) of SA systems [14, 55]. It is therefore the only strategy that has the potential to reliably guide SA systems towards configurations with defect-free structures. In fact, even simple closed-loop control systems with proportional control algorithms have shown significant improvements in initial nucleation time and desired configuration convergence speed over their open-loop and uncontrolled counterparts [13, 56, 57]. More involved model-based optimal control methods [38, 58, 59] or methods based on reinforcement learning [47, 52, 60] have similarly demonstrated promise for more complex systems.

The real-world implementation of particle design and control approaches for reproducible colloidal SA has been limited largely due to challenges in creating tractable strategies for (i) characterizing colloidal SA structural states and their free energies and diffusivities, and/or (ii) learning control policies that guide colloidal SA towards certain target structural states.

For example, throughout this chapter "defective," kinetically arrested structures and target structures have been described conceptually in the context of local and global minima in the FEL, respectively. Fig. 1.1 even assigns these states"positions" within the SA coordinate. The process of determining which states are in fact "defective" and their relative level of defectiveness is not straightforward, however. Although theoretically perfect lattices (e.g., theoretically perfect FCC, BCC, HCP lattices) have defined topologies, particles' movement due to their thermal energy prevents the realization of such perfectly defined structures. The most rigorous way to quantify the SA system structural state is by recording the 3D spatial coordinates and momenta of every particle – such a description cannot be easily interpreted by humans [20, 61]. The most common methods to characterize the colloidal SA state (e.g., Steinhardt bond-order parameters) create a handful of averaged, "collective" order parameters to which physical meaning can be more easily assigned. However, such methods can be sensitive to thermal and density fluctuations, the coexistence of various phases, and spatial inhomogeneity, all of which can interfere with the ability to resolve various defective, semi-defective, and near-perfect lattice types [62, 63].

Moreover, estimating the free energies and diffusivities of different colloidal SA structural states (which may already be imprecisely quantified by Steinhardt bond-order parameters, for example) is not a trivial process. The state-of-the-art methods for learning FELs and DLs generally require massive amounts of data collection and system-specific sampling techniques that may not be realizable for practically-sized experimental systems [64]. As a result, the majority of these methods have only been demonstrated *in-silico* via molecular dynamics simulations (e.g., weighted histogram analysis [65, 66], adaptively biased molecular dynamics [67], adaptive biasing force [68]). Importantly, the nonlinear and stochastic dynamics of colloidal SA systems can be described by the stochastic Langevin equation [31, 37, 38, 69]. Because the FEL and DL of a given system can alone naturally yield a stochastic Langevin model, the problem of learning free energies and diffusivities of colloidal SA structural states can alternatively be viewed as the problem of learning a stochastic Langevin model. Although many reported methods for learning Langevin models for systems with stochastic and nonlinear dynamics have been reported in the literature, the majority of these methods are extremely sensitive to the frequency at which system measurements can occur, rely on inflexible, system-specific sampling techniques, have been shown to be non-viable when short-time linear regions do not exist in the trajectory data, and/or are not well-suited for systems that contain state-dependent noise [64]. Methods based on Bayesian Inference have been shown to learn FELs and DLs for systems with nonlinear dynamics and state-dependent noise [69, 70] – but these methods can become computationally intractable for practically-sized systems.

Further, limited actuation and real-time sensing pose serious challenges to the practical application of both open- and closed-loop control policies to SA systems. The implementation of control via localized actuators is constrained [14]. Instead, control typically relies on manipulating certain macroscopic variables (e.g., temperature, pressure, external field strength). But, these global actuators are often limited in number and specificity. Difficulties in measuring and quantifying system configurations hamper real-time sensing of SA

systems [14, 55]. Small length scales inhibit conventional probing methods, whereas most other methods tend to be slow and invasive. It is further difficult to translate the information that can be collected (e.g., fluorescent imaging techniques that monitor real-time movements of Janus particles [71]) into quantitative variables.

The challenges described above answer to the questions: how can one design particles such that a colloidal SA system avoids kinetic traps and achieves target structures if one does not even know which configurations are in fact kinetically arrested and which ones are in fact target structures? How can one design particles or control approaches if it is impractical to predict how a system will evolve or even accurately measure how that system evolves over time tractably? How can one effectively control a system with limited actuation? Most importantly, how can one even begin to address the above questions when mechanistic understanding of colloidal SA systems' high-dimensional, highly nonlinear, and intrinsically stochastic dynamics is still fairly limited? The broader purpose of this thesis is to address these challenges.

1.4 Thesis Objectives and Outline

The primary objective of this thesis is to enable more effective particle design and control of colloidal SA systems. To this end, this thesis investigates strategies based on machine learning and optimal control for quantifying and classifying colloidal SA system states, learning tractable stochastic dynamical models of colloidal SA dynamics (and thus FELs and DLs), and learning control policies to guide colloidal SA. Each chapter in this thesis introduces a new learning-based strategy for "characterization" or "control" of colloidal SA and then demonstrates that strategy on *in-silico* colloidal SA systems with experimentally representative pair potentials. In particular, the characterization and control strategies of Chapters 2-4 are demonstrated on an *in-silico* colloidal SA system that has been used to represent the SA of DNA-functionalized particles effectively in two and three dimensions [72, 73]. Each chapter finally analyzes the efficacy of its presented characterization or control strategy in the context of improving particle design or control of colloidal SA.

Chapter 2 introduces a novel framework for colloidal structure characterization that systematically and quantitatively describes the self-assembly of colloidal lattices at the single-particle level. The framework first quantifies local structure via neighborhood graphs that are constructed from a precise methodology that has been shown to be robust to thermal fluctuations and capable of describing complex topologies [63, 74–77]. The high-dimensional, discrete nature of these neighborhood graphs prevents intuitive understanding of how these graphs are related. The framework uses deep neural networks to reduce the dimensionality of the neighborhood graphs and produce a low-dimensional manifold from which relationships among neighborhood graphs can be more easily inferred. The framework next uses agglomerative hierarchical clustering techniques to partition the low-dimensional space and assign physically meaningful classifications to the resulting partitions. The framework is demonstrated on several benchmark three-dimensional *in-silico* systems of colloids, includ-

ing a benchmark problem from reference [78]. The contributions of Chapter 2 have been published in reference [79].

Chapter 3 extends the framework developed in Chapter 2 to account explicitly for particle species. This extended framework quantifies both structural order (i.e., the unit cells within colloidal lattices such as FCC, BCC, and HCP, etc.) and compositional order (i.e., how A- and B-type particles are distributed among these unit cell sites). The framework is then used to investigate the role of particle size ratio and interparticle potential well depth in influencing the three-dimensional self-assembly of *in-silico* binary colloidal mixtures. The framework reveals that binary colloidal mixtures with small interparticle size disparities (i.e., A- and B-type particle radius ratios of $r_A/r_B = 0.8$ to $r_A/r_B = 0.95$) can promote the self-assembly of defect-free binary superlattices much more effectively than systems of identically sized particles at certain interparticle potential well depths, as nearly defect-free BCC-CsCl, FCC-CuAu, and IrV crystals are observed in the former case. The framework additionally reveals that size-disparate colloidal mixtures can undergo non-classical nucleation pathways where binary colloidal lattices evolve from dense amorphous precursors, instead of directly nucleating from dilute solution. The contributions of Chapter 3 have been published in reference [80].

The contributions of Chapters 2 and 3 demonstrate how improving colloidal SA structural characterization can aid in colloidal SA particle design. A key aspect of the presented characterization framework is that the framework can simultaneously and tractably characterize multiple colloidal SA trajectories (as opposed to merely characterizing individual lattices). This tractable characterization strategy allows for an in-depth analysis of two well-recognized colloidal SA particle design parameters, particle size ratio and interparticle potential well depth. The analysis yielded two key findings, which were that (i) small interparticle size disparities (i.e., $r_A/r_B = 0.8$ to $r_A/r_B = 0.95$) at certain interparticle potential well depths can drastically reduce the number of defects in self-assembled binary lattices and (ii) size-disparate systems can undergo non-classical nucleation. Colloidal SA practitioners can in turn use these insights to help design or synthesize particles meant for colloidal SA in the future.

Chapter 4 proposes a neural network-based strategy for model-based feedback control of colloidal SA systems. The framework uses a deep neural network to characterize colloidal self-assembly at the lattice level. The framework next uses a second deep neural network to create a discrete, deterministic model of the colloidal SA dynamics. Finally, the lattice characterization and modeling strategies are integrated into a model-based feedback control framework. The framework is demonstrated by manipulating temperature to guide the two-dimensional SA of an *in-silico* system of colloids. Chapter 4 shows that a model-based feedback control strategy can in principle be used to design a controller that can reliably guide colloidal SA. This work reveals the potential importance of incorporating a stochastic model and information from the FEL and DL into the design of control strategies. The focus of Chapter 5 is thus to develop a tractable framework for learning Langevin models (which naturally yield FELs and DLs) for systems with stochastic and nonlinear dynamics.

Chapter 5 proposes a novel framework to train deep neural networks that represent hidden

physics terms (e.g., FELs and DLs) within stochastic dynamical systems. These hidden physics terms in turn comprise a tractable stochastic dynamical model (e.g., a Langevin equation). The work in particular focuses on systems with multivariable and nonlinear dynamics with multiplicative noise (e.g., colloidal SA systems). The framework addresses several key shortcomings for existing methods for learning hidden physics/stochastic models, with a particular focus on the proposed framework's increased flexiblity and scalability. The framework may be prohibitively data-intensive for colloidal SA experimentalists, however. The contributions of Chapter 5 have been published in reference [64].

Due to the potential challenges in learning stochastic models and FELs and DLs in an experimental context, Chapter 6 proposes two model-free feedback control strategies for colloidal SA based on evolutionary reinforcement learning. These strategies learn control policies by iteratively updating the policies based on progress towards a pre-defined goal (i.e., through "objective search") or by searching for behavior novelty (i.e., "novelty search"), respectively. The strategies yield control policies that manipulate temperature and pressure to guide the three-dimensional SA of a benchmark *in-silico* system of colloids. This work is in its infancy and much of the chapter is devoted to discussing the conceptual advantages and disadvantages of employing objective search and novelty search for controlling colloidal SA.

Chapter 7 concisely summarizes the key findings within this thesis and lays out perspectives on future work that can be performed in the field. I believe the insights gained from this thesis provide a deeper mechanistic understanding of colloidal SA and contribute to an ever-developing archive of methods that can be used or expanded upon to achieve reproducible colloidal SA.

Chapter 2

A Deep Learning Framework for Characterizing Colloidal Self-Assembly

Creating a systematic framework to characterize the structural states of colloidal self-assembly systems is crucial for unraveling the fundamental understanding of these systems' stochastic and non-linear behaviors. The most accurate characterization methods create high-dimensional neighborhood graphs that may not provide useful information about structures unless these are well-defined reference crystalline structures. Dimensionality reduction methods are thus required to translate the neighborhood graphs into a low-dimensional space that can be easily interpreted and used to characterize non-reference structures. This chapter investigates a framework for colloidal system state characterization that employs deep learning methods to reduce the dimensionality of neighborhood graphs. The framework next uses agglomerative hierarchical clustering techniques to partition the low-dimensional space and assign physically meaningful classifications to the resulting partitions. The proposed colloidal self-assembly state characterization framework is first demonstrated on an in-silico system of 500 DNA-functionalized particles that self-assemble in three-dimensions under isothermal conditions. To investigate generalizability, the characterization framework is applied to several independent self-assembly trajectories, including a benchmark in-silico system of 2052 colloidal particles that undergo three-dimensional evaporation-induced self-assembly.

2.1 Introduction

The most common method to characterize the colloidal SA system state is Steinhardt bond-order parameters [81, 82], which provide information about local particle environments by defining certain rotationally invariant combinations of spherical harmonics calculated between particles and their nearest neighbors. Steinhardt bond-order parameters can be extremely sensitive to thermal fluctuations that smear local bond-order into broad overlapping

distributions and interfere with the ability to resolve the character of small domains. Other commonly used methods include Common Neighbor Analysis (CNA) [83, 84], Polyhedral Template Matching (PTM) [85], and Bond Angle Analysis (BAA) [86]. CNA and PTM evaluate the topology of each particle's nearest neighbors to generate neighborhood graphs that describe a given particle's local structure, whereas BAA evaluates the symmetry of each particle's nearest neighbors to create neighborhood graphs. These methods, however, fail to provide quantitative information about particles whose topologies or symmetries do not correspond to well-defined reference crystalline structures [62]. The high-dimensional, discrete nature of these neighborhood graphs prevents intuitive understanding of how these graphs are related; dimensionality reduction methods are thus required to translate the neighborhood graphs into a (continuous) low-dimensional space that can be easily interpreted and used to characterize non-reference structures.

At the time this project began, the state-of-the-art method for colloidal system state characterization accomplished dimensionality reduction by implementing diffusion maps [31, 38, 62, 63, 78, 87–92]. However, diffusion maps are computationally expensive to implement. In fact, the most recent implementations of diffusion maps require the choice of (arbitrarily chosen) "landmark points" to reduce the size of the high-dimensional space before dimensionality reduction takes place [90]. Diffusion maps further do not provide an explicit functional mapping between the high and low-dimensional spaces, thereby, limiting physical interpretation of the low-dimensional space.

Several other researchers have implemented different varieties of "machine learning" for dimensionality reduction and/or classification of colloidal structures [93–98]. For example, refs. [94, 95] implement principal component analysis (PCA) to detect phase transitions in off-lattice systems. PCA, however, is not designed to reduce the dimensionality of variables with highly nonlinear relationships among one another [99]. On the other hand, reference [96] uses a combination of Gaussian Mixture Models and shallow artificial neural networks to identify the overall crystal structures of bulk self-assembled systems. However, this work does not explicitly employ machine learning techniques for dimensionality reduction and instead investigates the learning techniques' ability to create and interpret large neighborhood graphs.

The overarching goal of this chapter is to develop a characterization framework for investigating the stochastic and nonlinear dynamics of entire colloidal SA trajectories (as opposed to merely characterizing individual lattices). This chapter thus proposes an alternative approach to dimensionality reduction based on a deep neural network called an autoencoder [100, 101]. Autoencoders are easy to implement with available tools and cheap to evaluate. This computational efficiency allows autoencoders to reduce simultaneously the dimensionality of the thousands of neighborhood graphs that can appear during SA, an operation which would likely be intractable for diffusion maps. The nonlinear activation functions within deep neural networks also allow the autoencoder to account explicitly for the nonlinear relationships among the diverse neighborhood graphs that may appear during SA. Autoencoders further provide an explicit mapping between the low- and high-dimensional spaces, elucidating which of the high-dimensional inputs are the most "important" for the

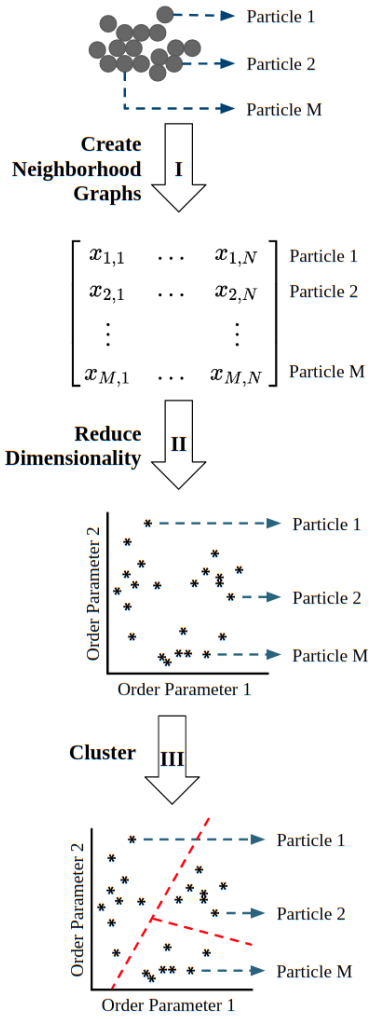


Figure 2.1: Colloidal self-assembly state characterization framework summary. The particle positions are recorded and translated into neighborhood graphs. The dimensionality of the neighborhood graphs is next reduced using deep learning techniques. Agglomerative hierarchical clustering is finally used to partition the low-dimensional space and assign discrete classifications to each particle.

system under analysis.

Note that the authors in reference [102] previously applied autoencoders based on shallow neural networks for dimensionality reduction and subsequent classification of colloidal systems. However, their approach creates neighborhood graphs using a vector of only 8 Steinhardt order parameters. In this project, neighborhood graphs are instead created via a well-established methodology based on Delaunay triangulation and graphlet decomposition [63, 74–77]. This methodology is much less sensitive to thermal fluctuations and has also been shown to quantify detailed colloidal lattice configurations by the authors in reference [63]. Deep neural network-based autoencoders further employ dropout regularization to pre-

vent model overfitting and achieve continuity in the low-dimensional space. Finally note that the authors in reference [102] primarily focused on classifying individual lattices, whereas the focus of the work in this chapter is to study entire colloidal SA trajectories.

This chapter proposes a three-step framework for colloidal SA system state characterization (see Fig. 2.1). The first step establishes neighborhood graphs with a precise methodology that has been shown to be robust to thermal fluctuations and capable of describing complex topologies [63, 74–77]. The second step uses deep learning techniques to reduce the dimensionality of the neighborhood graphs. The third step employs agglomerative hierarchical clustering to partition the low-dimensional space and assign physically meaningful classifications to the resulting partitions.

The proposed three-step colloidal system state classification framework is demonstrated on a three-dimensional *in-silico* system of 500 DNA-functionalized particles (i.e., silica colloids that are coated with blends of complementary single strands of DNA) [72, 73, 103, 104] that self-assemble into a variety of FCC, HCP, and BCC-like lattices. The generalizability of the characterization framework is examined by applying the framework to several independent colloidal SA trajectories (i.e., trajectories that were not used to train the autoencoder), including a system consisting of 2052 *in-silico* colloidal particles that undergo three-dimensional evaporation-induced SA [78]. The characterization framework exists in an easily accessible GitHub format that is explicitly designed for people to use and modify [105]. More in-depth descriptions of the "DNA-functionalized" and "evaporation-induced" *in-silico* self-assembly systems can be found in subsequent sections.

2.2 Neighborhood Graph Construction

The first step in classifying the structure of a given colloidal SA particle is to generate a "neighbor list" that consists of a list of particles that are considered topologically or symmetrically adjacent to the particle of interest. This neighbor list is then used to construct a neighborhood graph that quantifies the local structure of the given particle. Two of the most common local structure classification methods, Common Neighbor Analysis (CNA) [83, 84] and Steinhardt order parameters [81, 82], heavily rely on the concept of particles being "bonded" to establish neighbor lists. These methods thus require a strict definition of a bond, where two particles are considered bonded if they fall within a predefined cutoff radius. However, such a cut-off radius is, by necessity, somewhat arbitrary. In addition, thermal vibrations, the coexistence of various phases, and fluctuations in the local density will introduce noise into the analysis and can even make finding a suitable cut-off radius impossible. This problem is partially mitigated by adaptive CNA [106], where the cutoff radius is determined by the average distance to a heuristically chosen number of particles. Despite the use of averaging, radii for low-density and vapor phase particles can be extremely large and inhibit classification accuracy. In addition, a geometric pre-factor is often required to separate first and second-nearest neighbors in close-packed lattices [62, 63]. The approach further assumes that a given particle's neighborhood is isotropic, which is often not the case

for open lattices [107].

This chapter employs the methodology described in refs. [63, 74] to obtain the neighbor list of topologically adjacent particles and subsequent neighborhood graph. The gist of the method is that the convex hull formed by the set of neighboring atoms describes the local structure around an atom. The convex hull, which is represented as a Voronoi cell, is determined from a Delaunay triangulation of the particle of interest and its 18 nearest neighbors (or half the inner shell atoms in FCC and HCP lattices). Because this method avoids the concept of bonds between particles and instead uses a geometry-based, fixed number of particles to establish the neighborhood, it is less sensitive to thermal fluctuations, density gradients, and anisotropy mentioned above. Finally, this method includes the central particle in the neighborhood graph, which provides greater connectivity between neighbors and therefore greater distinction between structures in comparison to CNA and Steinhardt classification methods. Delaunay triangulation does yield inconsistent results at solid-vapor interfaces, however, as the method tends to connect far-away particles to create three-dimensional convex hulls. The authors in reference [63] use outlier detection techniques to filter these spurious results. As will be discussed in later sections, our proposed dimensionality reduction/classification techniques naturally filter such outliers effectively.

The neighborhood graphs are evaluated using the graphlet decomposition-based methodology of refs. [63, 74–77], which has been successfully implemented for analyzing local structure in a variety of colloidal and biological networks. Graphlets are small, connected, non-isomorphic induced subgraphs of a larger network that contain some number of nodes, k [63, 74–77]. The k nodes in each graphlet are topologically distinguished by their individual automorphism orbits that account for the symmetries among the nodes in said graphlet. Each graphlet thus contains 1 to *k-1* distinct automorphism orbits. The neighborhood graph is evaluated by computing the frequency of these orbits for a given neighborhood. For the purposes of this paper, each node is a particle within the neighborhood graph established by the Delaunay triangulation described above. The neighborhood graph is evaluated using graphlets with 2-5 nodes, as calculations involving larger graphlets quickly become intractable. Graphlets with 2-5 nodes display 73 different automorphism orbits. As a result, the local structure of each particle is quantified by a 73×1 vector (i.e., the neighborhood graph), where each entry in the vector refers to the frequency of an automorphism orbit. Following the procedure of refs. [63, 74] the frequencies are weighed to account for the fact that the appearance of more complex automorphism orbits correlates with the appearance of simpler ones. Finally, each neighborhood graph is normalized such that its sum is unity.

2.3 Dimensionality Reduction via Autoencoders

The previous section described an established methodology for quantifying the local structure of colloidal particles. However, the primary objective of this chapter is not merely to quantify local structure, but to interpret the local structure in a physically-meaningful and intuitive way that helps elucidate understanding of the colloidal SA system of interest. The high

dimensionality of the neighborhood graphs and non-uniformity in the distances among them indicate that dimensionality reduction must be performed to produce a low-dimensional manifold from which relationships among neighborhood graphs can be more easily inferred. The dimensionality of the neighborhood graphs is reduced using a deep artificial neural network called an autoencoder.

An autoencoder is comprised of an *encoder* that constructs a low-dimensional representation of its input (i.e., the neighborhood graph in this case) and a *decoder* that reconstructs the input from the low-dimensional representation [100, 101]. The encoding process is often lossy, meaning that part of the information is lost during the encoding process and cannot be recovered during decoding. Dimensionality reduction is thus accomplished by finding the encoder/decoder pair that keeps the maximum information when encoding and correspondingly has the minimum reconstruction error when decoding. Note that only the encoder is used to reduce dimensionality, whereas the decoder is used to find the encoder model that creates the best low-dimensional representation of the input data.

The encoder and decoder are deep feed-forward artificial neural networks (see Fig. 2.2). These neural networks consist of multiple fully-connected layers that contain various numbers of nodes. Each node multiplies its input by a weight vector and feeds that product into a (generally nonlinear) activation function (e.g., hyperbolic tangent, sigmoid, rectified linear unit). Each neural network has an input layer, some number of middle or hidden layers, and an output layer. In this chapter, the input layer to the encoder is the neighborhood graph while its output is the low-dimensional representation of the neighborhood graph (also called the bottleneck layer). On the other hand, the input to the decoder is the bottleneck layer and the output is the reconstructed neighborhood graph.

For a given autoencoder architecture (i.e., number of nodes and layers with chosen activation functions), the "optimal" encoder/decoder scheme is found through an iterative training process. Here, a set of training data is fed to the autoencoder and gradient descent methods are used to update the encoder/decoder weights until the reconstruction loss is sufficiently minimized. Denote E and D as all possible encoder/decoder combinations (i.e., all possible values of the autoencoder weights), x as the neighborhood graph, $e(x;\lambda_e)$ as the encoder where λ_e denotes all encoder weights, $d(e(x;\lambda_e);\lambda_d)$ as the decoder where λ_d denotes all decoder weights, and $J(x,(d(e(x;\lambda_e);\lambda_d))$ as the decoder's reconstruction loss. The reconstruction loss is often formulated as the mean squared error (MSE) of the original and reconstructed neighborhood graphs. The process of finding the optimal encoder/decoder pair is mathematically represented below. Note that training the autoencoder can be thought of as a "self-supervised" learning process, as training determines a (nonlinear) function that maps the neural network's inputs (i.e., the neighborhood graphs) to themselves (i.e., neighborhood graphs that are reconstructed from their low-dimensional representation).

$$(e^*, d^*) = \underset{(e,d)\in E\times D}{\arg\min} \{J(x,(d(e(x;\lambda_e);\lambda_d))\} \tag{2.1}$$

Larger (autoencoder) neural networks (i.e., those with more nodes and/or layers) can

find more complex relationships between their inputs and outputs, leading to a lower reconstruction loss [108]. Larger autoencoders are especially prone to overfitting, however, as the autoencoder is solely trained to encode and decode with as little reconstruction loss as possible, no matter how the low-dimensional space is organized. This can manifest itself in the low-dimensional space lacking continuity (i.e., two close points in the latent space give two completely different decoded contents) and lacking completeness (i.e., certain points within the latent space provide non-physical responses once decoded). One way to overcome this problem is to introduce dropout regularization [109, 110], which omits certain nodes at random gradient descent iterations to reduce the size of neuron weights and prevent co-adaptations of the training data.

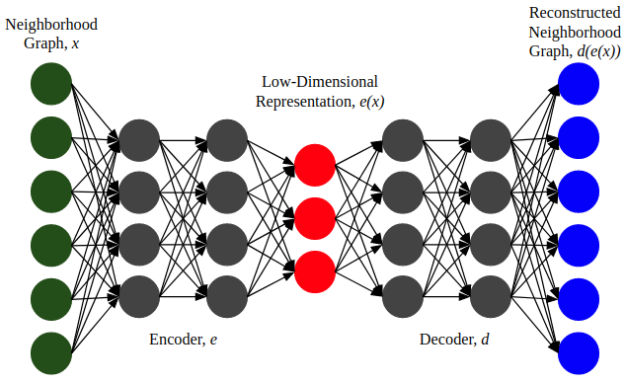


Figure 2.2: Autoencoder architecture. The *encoder*, e, compresses the neighborhood graph of a given particle (a 73×1 vector, x) into a low-dimensional representation $e(x)$. The *decoder*, d, reconstructs the given neighborhood graph from the low-dimensional representation. In this chapter, the encoder and decoder are deep neural networks with nonlinear activation functions that learn the encoding/decoding scheme that minimizes the reconstruction error of the decoder. This "optimal" encoder/decoder pairing is determined through an iterative training process, where the weights and biases within these neural networks are updated through gradient descent methods. Each circle represents a node within the neural network and the arrows represent the connections between these nodes. The autoencoder input layer nodes are green, the autoencoder output layer nodes are blue, the hidden layer nodes are grey, and the bottleneck layer nodes are red.

To train the autoencoder, particle position data for 11 different isothermal trajectories of an *in-silico* three-dimensional system of 500 multi-flavored colloidal particles [72, 73, 103, 104] was first collected. The inter-particle interactions in each trajectory were varied such that a variety of vapor, low-density, defective, and FCC, HCP, and BCC-like lattices appear during assembly. The particle position data was used to evaluate the neighborhood graph for each particle in each simulation frame according to the Delaunay triangulation and graphlet decomposition methodology of Section 2.2 (527,500 total neighborhood graphs). Only the unique neighborhood graphs (4153 unique neighborhood graphs) were used to train the autoencoder.

Neural networks are generally trained using a 5-fold cross-validation methodology in which 60% of the sample data is used to train the model, 20% is used to validate model

accuracy (i.e., test for model over-fitting), and 20% is used to evaluate model performance [108, 111]. This approach to model training implicitly assumes that the sections of sample data chosen to train, validate, and test the model are fairly representative of the remaining sections (i.e., the data is fairly normally or evenly distributed with minimal outliers). However, the natural non-uniformity in the distances among neighborhood graphs and the fact the Delaunay triangulation creates inconsistent neighbor lists at solid-vapor interfaces render such assumptions invalid. As a result, the autoencoder was trained with all of the 4153 unique neighborhood graphs from the sample set. Dropout regularization was then introduced during training to prevent over-fitting. The objective of training the autoencoder is to tune the model weights to achieve the most accurate reconstruction of the neighborhood graphs for a given number of nodes and layers. The performance of the encoder is evaluated by classifying lattice structures in the low-dimensional space and visualizing the resulting classification (see Section 2.7).

Autoencoder performance greatly depends on certain architectural choices, namely network size (i.e., number of nodes in each hidden layer and the bottleneck layer, and the total number of hidden layers), activation function choice (e.g., hyperbolic tangent, linear, rectified linear unit), batch size (i.e., the number of sample data points used for each weight update), and regularization strategy (e.g., norm on the cost function that regulates weight size, dropout regularization) [108, 111, 112]. Certain architectural choices are fairly standard or are informed by the training data. For example, choosing hyperbolic tangent activation functions for the hidden layers is a standard choice given the irregularity of the neighborhood graph data [113]. For the same reason,a batch size equal to the size of the training data set was chosen. Linear activation functions in the encoder/decoder output layers are standard choices regardless of the sample data properties [100–102]. Dropout regularization is a standard technique to prevent the model from over-fitting the training data and MSE is a standard choice of loss function [108–110]. The optimal network size is found by plotting autoencoder training loss as a function of network size and implementing the "elbow method" [102, 114, 115] to choose the network architecture with the best balance of computational cost and performance. Finally, the autoencoder was implemented using the Python library Keras (a TensoFlow API) [116, 117]

One of the main advantages of autoencoders over diffusion maps is that autoencoders provide an exact analytical mapping from the high to low-dimensional spaces. This mapping allows the *relative importance* of each entry in the neighborhood graph to be assessed (e.g., via input perturbation and stepwise methods [102, 118–121]). Relative importance is measured by the variation in neural network loss (which in this case is the MSE) caused by perturbing samples in the training data set. Input perturbation methods add Gaussian white noise to neighborhood graph entries whereas stepwise methods replace all graph entry values with their mean. Graph entries that show the largest MSE variation are deemed the "most important". A mathematical representation of relative importance can be found below. Here, ΔE_k is the variation in MSE caused by the change applied to the kth neighborhood graph entry and the sum in the denominator runs over all entries from a neighborhood

graph of dimension $N \times 1$.

$$RI_k = \frac{\Delta E_k}{\sum_{j=1}^{N} \Delta E_j} \tag{2.2}$$

The relative importance analysis reveals the influence each graph entry has on the quantification of local structure and can even validate (or invalidate) the need for dimensionality reduction.

2.4 Partitioning the Low-Dimensional Space for Classification

The first component of classifying local structure is using the encoder model to compute the low-dimensional coordinates of each unique neighborhood graph. The key challenge of colloidal SA state classification is then partitioning this low-dimensional space into discrete regions to make final decisions regarding structural identity. Although 4153 unique neighborhood graphs were identified, many of those will correspond to either nearly identical structures or structures that are not conceptually relevant for us to distinguish. For example, various particles at solid/vapor interfaces may display very different topologies and thus very distant neighborhood graphs. However, it is not important to distinguish these topologies from a self-assembly point of view. On the other hand, it is extremely important to distinguish topologies with fairly similar neighborhood graphs (e.g., BCC and FCC lattices). As a result, this chapter investigates clustering strategies to assign each neighborhood graph a discrete classification according to the cluster to which its low-dimensional representation belongs.

This chapter implements agglomerative hierarchical clustering with a Ward's minimum variance linkage metric to partition the low-dimensional space [122, 123]. Initially, each cluster is a single point (e.g., 4153 total clusters). In each iteration, two clusters are combined into one by finding the pair of clusters that leads to the minimum increase in total intra-cluster variance after merging. This variance increase is a weighted squared distance between cluster centers and these iterations continue until all data points are grouped into one cluster. The method creates clusters of various shapes, sizes, intra-cluster variances, and membership populations. By iteratively minimizing the increase in total intra-cluster variance, the method can naturally discover both clusters that are adjacent in the low-dimensional manifold with small intra-cluster variances (e.g., the topologically similar BCC and FCC structures) and clusters with high intra-cluster variances that span larger, less-populated sections of the coordinate space (e.g., various weakly crystalline structures). Moreover, the clustering strategy makes no assumptions regarding the distribution of the low-dimensional space as it only assesses similarities between pairs of objects. Agglomerative hierarchical clustering establishes a "cluster tree" that reveals the underlying hierarchical structure of the data. The branches within this tree can be used to determine informed descriptions of the discrete regions of the low-dimensional space and choose a number of clusters that is

appropriate for the application of interest. The justification behind the choice of the number of clusters used in this chapter can be found in Section 2.7.

Once the low-dimensional space has been partitioned into the chosen number of clusters, classification has occurred in the sense that each section of the low-dimensional manifold has been assigned to a class. The results of the classification are next visualized by assigning a unique color to each identified class and using Open Visualization Tool (OVITO) software [124] to create a color-coordinated image of each simulation frame. Qualitative analysis of these simulation frames provides a general idea of what each class physically represents. For example, a simulation frame that contains a perfect FCC lattice reveals which clusters roughly correspond to bulk FCC or surface FCC particles, while a simulation frame that contains primarily disperse particles reveals which classes correspond to vapor particles, etc. The hierarchical structure provided by the cluster tree can then be used to make more refined analyses (e.g., defective HCP vs HCP). The analysis is further enhanced by the fact that perfect crystalline lattices have known topologies, and thus known neighborhood graphs and low-dimensional coordinates.

2.5 Colloidal Self-Assembly System Descriptions

Three separate three-dimensional *in-silico* colloidal SA systems are used to demonstrate the proposed characterization framework. The first two consist of a systems of 500 and 1000 DNA-functionalized particles that self-assemble under isothermal conditions. The third consists of a system 2052 silica colloids that undergo evaporation-induced self-assembly [78]. These systems are described in more detail below.

Self-Assembly of Multi-Flavored Colloids

One way to promote the SA of colloidal particles is through functionalizing their surfaces with DNA. DNA-functionalized particles (DFPs) interact with each other through complementary Watson–Crick base-pairing interactions and have been used to assemble many superlattice structures [72, 125]. As a means of achieving selective binding among DFPs, it has recently been suggested that particles can be functionalized with a blend of two types of DNA strands with complementary concentrations on each particle. These "multi-flavored" particles can exhibit a tunable attraction between the like particles while maintaining interactions between unlike pairs. This approach has been shown to induce the crystallization of equally sized particles into BCC, HCP, and FCC structures [103, 126, 127]. Fig. 2.3 shows a schematic representation of the multi-flavored DFPs.

In this chapter, colloidal SA trajectories are obtained from *in-silico* binary colloidal mixtures that represent multi-flavored DFPs. The pairwise interaction model used in molecular dynamics (MD) simulations for obtaining these trajectories is provided in Eq. (2.3). This pair interaction model is of a Fermi-Jagla type, which has previously been successfully used

to study the self-assembly process of DFPs both in two [72, 73] and three dimensions [103]:

$$\frac{U(r/\sigma)}{\epsilon} = \frac{\epsilon_c}{\epsilon}\Big(\frac{\sigma_c/\sigma}{r/\sigma - R_s/\sigma}\Big)^n + \frac{A_0/\epsilon}{1+exp[A_1(r/\sigma - A_2)]} - \frac{B_0/\epsilon}{1+exp[B_1(r/\sigma - B_2)]} \quad (2.3)$$

The first term in Eq. (2.3) represents the particle-particle core repulsion, where ϵ_c represents the energy scale of the repulsion, σ_c represents the length scale of repulsion and R_s is a shifting factor related to particle size. The second and third terms capture the soft repulsion and attraction from DNA sequences, respectively. A_0 and B_0 control the strength of these interactions, while A_1 and B_1 control the interaction range. A_2 and B_2 control the separation distance.

To tune interparticle potentials, unlike pair interaction E_{AB} are kept fixed and like interactions E_{AA} and E_{BB} are varied independently. The relative like interaction strength $E^*_{AA} = E_{AA}/E_{AB}$ and $E^*_{BB} = E_{BB}/E_{AB}$ can thus be adjusted independently from 0.0 to 1.0. Note that $E^*_{AA} = E^*_{BB}$ throughout all simulations unless otherwise noted. Further note that setting $E^*_{AA} = E^*_{BB}$= 1.0 reduces the multi-flavoring to single flavoring, where all particles are identical. Setting $E^*_{AA} = E^*_{BB} = 0.0$ makes the system a conventional binary mixture, where A-A and B-B interactions are purely repulsive and only A-B interactions are attractive. All particles in all simulations are the same size, so $r_A/r_B = 1.0$ in all simulations. The 11 colloidal SA trajectories used to train the characterization framework had interparticle interaction strength ratios of $E^*_{AA} = E^*_{BB}$= 0.0 to 1.0 at intervals of 0.1.

Molecular dynamic (MD) simulations are performed using LAMMPS [128] in the canonical ensemble. The system used to train the characterization framework contains 500 total particles with a 1:1 mixture ratio of A-type and B-type particles. The interaction strength and size ratio are varied using the pair potential model discussed above. Simulations are performed in a cubic box with periodic conditions applied to all three dimensions, under dilute conditions with number density $\rho = 0.02\sigma^{-3}$, and using a Langevin thermostat with a time constant $\tau = 2\sigma m^{1/2}\epsilon^{-1/2}$. Each simulation involves 1×10^9 total time steps where each time step is $\Delta t = 10^{-3}\sigma m^{1/2}\epsilon^{-1/2}$. Each MD simulation is performed at a constant, pre-determined temperature suitable for crystallization starting from a random dilute liquid phase, where particles are allowed to evolve spontaneously to form crystals. The trajectories generated from these simulations are visualized using OVITO [124]. Note that a version of this system that contains 1000 total particles was used to test the characterization framework (see Section 2.7). Finally, the parameter values used in Eq. 2.3 are: $\epsilon_c = 10\epsilon$, $n = 36$, $A_0 = 11.035\epsilon$, $A_1 = 404.4/\sigma$, $A_2 = 1.0174\sigma$, $\sigma_0 = 0.2\sigma$, $s = 0.8\sigma$, $B_0 = [-1.3219\epsilon, 0$, $B_1 = 1044, 5/\sigma$, and $B_2 = 1.0306\sigma$.

Evaporation-Induced Self-Assembly

One common high-throughput method for fabricating colloidal crystals involves dispersing colloids in a volatile solvent followed by evaporation of the solvent to deposit a crystalline solid onto a substrate (i.e., "evaporation-induced self-assembly") [78]. The authors in reference [78] performed massive-scale non-equilibrium MD simulations with an explicit-solvent

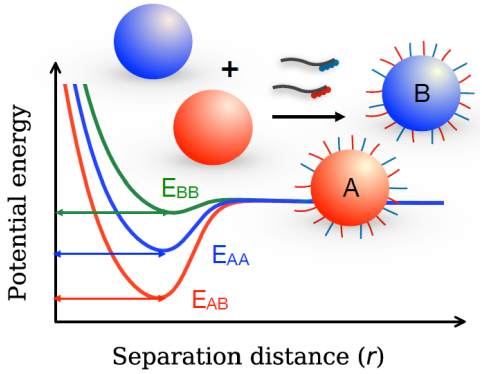


Figure 2.3: Schematic representation of multi-flavored DFPs and their effective pair potential model. Each of the pairwise interaction strengths E_{AA}, E_{BB}, and E_{AB} can be manipulated experimentally by controlling the blending ratio of two different types of DNA sequences. These interaction strengths be adjusted in simulations by changing the parameters of implicit Fermi-Jagla potential in Eq. (2.3).

model to study the evaporation-induced assembly of colloidal crystals from solution onto a horizontal substrate. Six snapshots from an MD simulation consisting of 2052 (initially disperse) silica colloids were used in Section 2.7 to examine the generalizabilty of the characterization framework. Note that this data was provided by the authors of reference [78].

2.6 Autoencoder Architecture and Relative Importance Analysis

The key autoencoder architectural choices are the batch size, activation function, regularization strategy, and network size. Justifications for each of the former three choices are described earlier in Section 2.3, while the latter choice is informed by implementation of the elbow method [102, 114, 115]. The elbow method (which is widely used throughout the self-supervised and unsupervised learning communities [102, 114, 115]) plots some measure of neural network performance (e.g., MSE) against some neural network hyper-parameter (e.g., the number of nodes in a given neural network layer). The method involves visually detecting a "slope change" where the performance of the neural network begins to improve more slowly with the change in the hyper-parameter. The beginning of this slope change is called the "elbow".

Here, several autoencoder models with different network sizes are trained with the sample data (i.e., the 4153 unique neighborhood graphs found from the 11 isothermal *in-silico* trajectories described in Section 2.2 and 2.5). The autoencoder MSE is plotted against the number of nodes in the bottleneck layer (i.e., the size the low-dimensional space found by the encoder) for candidate models that only differ by the number of hidden layers and number of nodes per hidden layer (see Fig.2.4). Elbows in this plot occur between 2 and 4 bottleneck

nodes, indicating that a bottleneck layer size of 3 nodes is likely sufficient to capture the essential information from the neighborhood graphs. Moreover, the corresponding size of the 3×1 low-dimensional representation is convenient from a visualization standpoint. The autoencoders with 2 hidden layers and 500 and 1000 nodes per hidden layer display nearly identical performance, with the latter model showing a marginally lower MSE. Models with larger network sizes do not display any performance improvements. As a result, the chosen autoencoder model contains 2 hidden layers, 1000 nodes per hidden layer, and 3 bottleneck nodes (which creates a low-dimensional space of dimension 3×1).

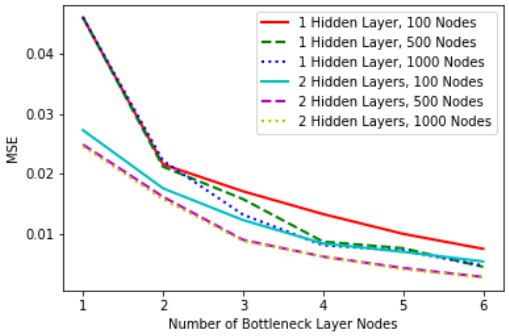


Figure 2.4: Autoencoder architecture optimization. The autoencoder MSE is plotted against the number of nodes in the bottleneck layer (i.e., the length of the low-dimensional representation vector) for various network sizes. "Elbows" in these plots occur between 2 and 4 order parameters, indicating that 3 order parameters are likely sufficient to capture the essential information from the neighborhood graphs. The autoencoder with 2 hidden layers and 1000 nodes per hidden layer displays the (albeit marginally) lowest MSE.

This chapter implements input perturbation (with 10% Gaussian white noise) and stepwise relative importance analyses on the chosen autoencoder [102, 118–121] (see Fig. 2.5). The analysis shows that nearly all neighborhood graph entries (with the exception of graph entries 22 and 23) are equally important and indicates that no single graph entry, or even small group of graph entries can be used to quantify the colloidal SA system state. This validates the need for implementing dimensionality reduction. The 22nd and 23rd neighborhood graph entries do account for nearly 25% of the MSE variation, however. This spike in variation is due to the fact that the neighborhood graph construction methodology can yield extremely large outlier values at solid-vapor interfaces. Translating the neighborhood graphs into a low-dimensional space significantly reduces the effects of these outliers and does not inhibit local structure characterization and classification (see Section 2.7 for more details).

Fig 2.5 further shows spikes in relative importance at neighborhood graph entries 0-1 and 30-36. Entries 0-1 refer to two and three-component linear orbits that are common in newly formed, small crystallites. These spikes indicate that many of the unique signatures used to train the autoencoder correspond to very weakly crystalline particles on the precipice of crystallization. Meanwhile, entries 30-36 refer to square and pentagonal-like shapes that are

common in (defective) FCC, HCP, and BCC structures. Again, these spikes demonstrate the frequency of FCC, HCP, and BCC-like structures in the training data. The above points show that the relative importance analysis can not only point out unexpected behavior in the characterization framework (e.g., the erratic neighborhood graph values for particles at solid/vapor interfaces) but also can demonstrate to which types of data the autoencoder model is more sensitive (e.g., the very weakly crystalline and FCC/BCC/HCP-like particles mentioned a few lines above).

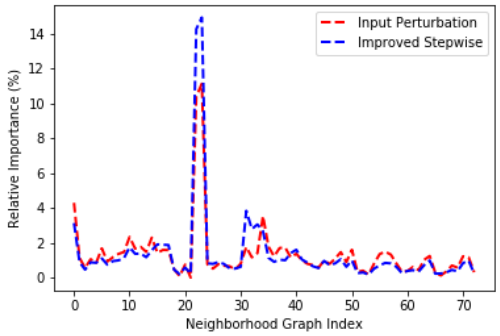


Figure 2.5: Relative importance analysis. Input perturbation and improved stepwise methods are used to assess the relative importance of the 73 entries within the neighborhood graph. Although neighborhood graph entries 22 and 23 account for the largest percentage of MSE variation, these results demonstrate that no single graph entry, or even relatively minor groups of graph entries can be used to quantify the system state. Moreover, the large MSE variation caused by nodes 22 and 23 is a function of certain outliers found at solid-vapor interfaces.

2.7 Classification, Visualization, and Analysis

The chosen encoder model was used to translate the entire training data set (4153 unique neighborhood graphs) into a three-dimensional low-dimensional space. Agglomerative hierarchical clustering (with Ward's linkage) was then implemented to partition the low-dimensional space. Although the strategy produces a cluster tree that shows the hierarchical structure of all 1 to 4153 possible cluster distributions, the process of choosing the "best" number of clusters is somewhat subjective [122, 123]. In fact, a key advantage of agglomerative hierarchical clustering is that the strategy allows the number of clusters for classification to be chosen based on specific application-based needs.

This chapter focuses on the SA of FCC, HCP, and BCC-like structures from a system of multi-flavored DFPs [72, 73, 103, 104]. The topologies of theoretically perfect FCC, HCP, and BCC lattices are known. This information was used to calculate neighborhood graphs and the corresponding low-dimensional points of these three theoretically perfect lattices. However, "perfect" or at least "not meaningfully defective" FCC, HCP, or BCC lattices may

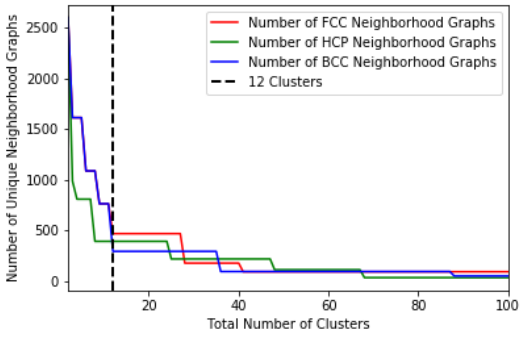


Figure 2.6: Analysis to determine number of clusters. Agglomerative hierarchical clustering (using Ward's linkage) is used to cluster the low-dimensional representations of the 4153 unique neighborhood graphs taken from the 11 isothermal colloidal self-assembly trajectories that were used to train the autoencoder (see Section 2.5). The number of unique neighborhood graphs corresponding to FCC, BCC, and HCP structures is plotted against the number of clusters in each branch of the resulting cluster tree. At 12 total clusters, the low-dimensional representations of FCC, HCP, and BCC neighborhood graphs are separated into different clusters.

have neighborhood graphs that correspond to a number of different low-dimensional points. As a result, any cluster that contains one of these three theoretically perfect lattice points can be analogously labeled.

Fig. 2.6 shows the number of low-dimensional points (that represent neighborhood graphs) corresponding to FCC, BCC, and HCP structures plotted against the number of total clusters in each branch of the cluster tree. At the branch corresponding to 12 total clusters, the FCC, HCP, and BCC perfect lattice points are first separated into different clusters. The number of points assigned to each of the three lattice types decreases with the total number of clusters as the points that are further from the theoretically perfect lattices are placed into other clusters. The choice in the number of clusters is thus a balance between the desired classification precision (i.e., the strictness of the definition of an FCC, HCP, or BCC lattice) and the analytical burden of interpreting potentially hundreds of clusters. In this chapter, the minimum number of clusters required to separate the theoretically perfect FCC, BCC, and HCP lattices (i.e., 12 total clusters) was chosen. Subsequent visual analyses of SA simulation trajectories show this choice to be reasonable (see Section 2.7).

Fig. 2.7a shows the the colored low-dimensional representations of all 4153 unique neighborhood graphs. A distinct color was assigned to each of the 12 clusters. The FCC, HCP, and BCC clusters correspond to clusters C9 (green), C8 (brown), and C12 (purple), respectively, while vapor particles correspond to cluster C1 (red). Note that vapor particles tend to display very small neighborhood graph entries and thus contain predictable neighborhood graphs/low-dimensional coordinates. Particles at solid-vapor interfaces exist in clusters C3 and C5. The neighborhood graphs of these particles tend to contain extremely high neighborhood graph entries (particularly at entries 22 and 23). Although it is not immediately

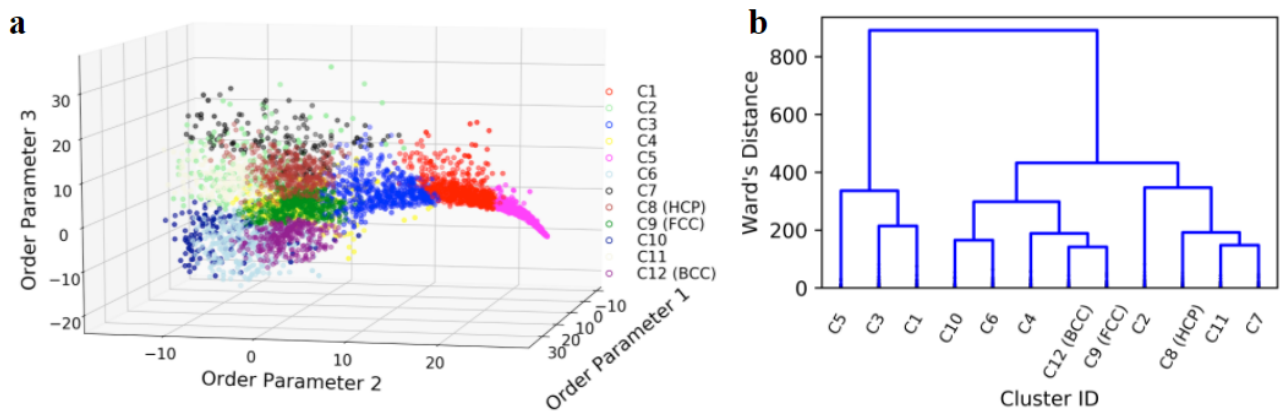


Figure 2.7: Agglomerative hierarchical clustering summary. Agglomerative hierarchical clustering (using Ward's linkage) was used to cluster the low-dimensional representations of 4153 unique neighborhood graphs (from the 11 isothermal colloidal self-assembly trajectories that were used to train the autoencoder described in Section 2.5) into 12 clusters. These clusters are labeled C1-C12. (a) The low-dimensional representation of each unique neighborhood graph is plotted and colored according to its labeled cluster. Points corresponding to bulk FCC, HCP, and BCC lattices exist within clusters C9, C8, and C12, respectively. (b) The Ward's distance between each cluster is plotted against each cluster's placement within the cluster tree.

clear to which types of structures the remaining clusters correspond, their close proximity to one another and distance from the vapor states suggest that they are likely surface or defective crystalline structures.

The structure of the cluster tree (Fig. 2.7b) provides important insights regarding the physical characteristics of the remaining clusters. First, clusters C1, C3, and C5 fall under the same branch while the remaining clusters (which include the FCC, HCP, and BCC clusters) fall under a second branch. This suggests that the first level of the cluster tree likely separates "crystalline" and "vapor/near-vapor" particles. Clusters C4, C6, C9 (FCC), C10, and C12 (BCC) all fall under the second level middle branch, suggesting that C4, C6, and C10 correspond to some types of surface or defective FCC/BCC structures. The fact that C4 belongs to the same parent branch as C9 and C12 indicates that C4 is likely more topologically similar to C9 and C12 than it is to C6 and C10. The right second level branch contains clusters C2, C7, C8 (HCP), and C11. The distances separating the C7, C8 (HCP), and C11 leaves are very small, also indicating that C7 and C11 could correspond to slightly defective HCP structures while C2 could correspond to either highly defective or surface HCP particles.

The low-dimensional space appears to have achieved continuity (i.e., similar structures have similar low-dimensional coordinates). For example, clusters C1, C3, and C5 all correspond to either vapor particles or vapor particles at solid-vapor interfaces. Although these clusters contain both extremely small and large neighborhood graph entries, the clusters have low intra-cluster variances and are adjacent in the low-dimensional space. The presented dimensionality reduction strategy thus effectively handles the massive outliers the

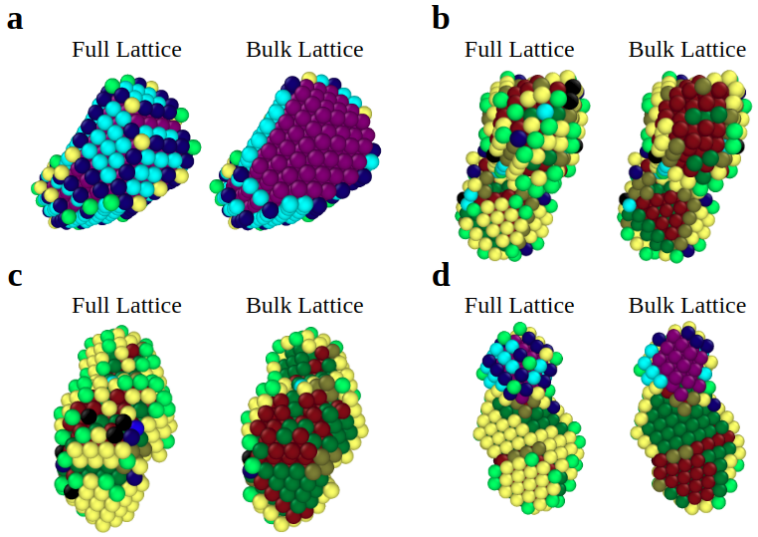


Figure 2.8: Four classified colloidal self-assembly lattices. The figure shows 4 lattices from the final time steps of 4 of the 11 isothermal colloidal self-assembly trajectories used to train the autoencoder. Each particle in each lattice is colored according to its classification in Fig. 2.7. The term "full lattice" indicates that every particle in the snapshot is shown while the term "bulk lattice" indicates that the top layer of particles has been removed. The structure in (a) is primarily BCC, the structures in (b) and (c) are mixed FCC and HCP, and the structure in (d) contains FCC, HCP, and BCC particles.

neighborhood graph construction methodology occasionally produces at solid-vapor interfaces. The FCC, HCP, and BCC clusters are close to one another, yet far apart from the vapor clusters. Meanwhile the remaining clusters (which likely correspond to defective and surface particles) are not only close to one another but also take up a large percentage of the low-dimensional space to reflect their large topological range.

Note that some of the boundaries among clusters appear exceedingly complex, suggesting that some of the data points are misclassified. The boundary complexity is a function of both the (unavoidable) noise in the neighborhood graph construction and the choice of a small number of clusters. This problem could potentially be addressed by increasing the number of clusters. However, the objective of the characterization framework is to elucidate understanding of colloidal SA processes as a whole and not to perfectly characterize each individual particle (otherwise one would avoid dimensionality reduction altogether). Overall, the colloidal SA state characterization framework appears to reduce effectively the dimensionality of neighborhood graphs and sensibly partition the low-dimensional space.

The characterization framework is first demonstrated by using OVITO to visualize 4 different lattices found from 4 of the 11 different isothermal colloidal SA trajectories used to train the autoencoder (see Fig. 2.8). The particles in each lattice are colored according to their classifications in Fig. 2.7. Each of the 4 lattices is shown in full (labeled "Full Lattice") and with its top layer removed (labeled "Bulk Lattice").

The OVITO visualizations were used to assign brief, physically meaningful descriptions

to each cluster (see Table 2.1 for a summary of these descriptions). The bulk particles in 2.8a (purple) almost all belong to cluster C12 and correspond to BCC structures. The surface particles primarily belong to clusters C6 (light blue) and C10 (dark blue), with scattered particles belonging to clusters C4 (yellow) and and C2 (light green). The C6 (light blue) particles clearly correspond to surface BCC (100)-(111) particles. The C10 (dark blue) particles only exist at the interface between two surface planes and likely correspond to BCC surface stacking faults.

Table 2.1: Cluster structural classifications. Each cluster identification (C1-C12) is matched with a brief physical description.

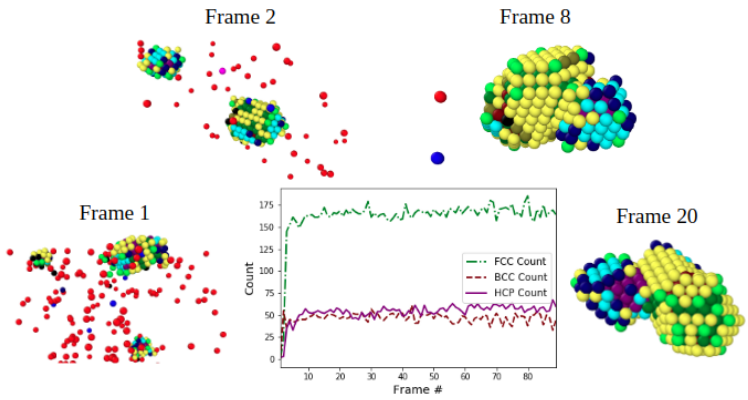
Cluster Label	Structure Description
C1	Vapor
C2	Defective FCC surface particle
C3	Vapor at Solid-Vapor Interface
C4	Surface FCC
C5	Vapor at Solid-Vapor Interface
C6	Surface BCC
C7	Weakly bound HCP-like particle
C8	HCP
C9	FCC
C10	BCC Stacking Fault
C11	Defective Bulk HCP
C12	BCC

Meanwhile, the bulk particles in Fig. 2.8b-7c are primarily from clusters C9 (Green, FCC) and C8 (Brown, HCP). Another bulk particle classification is C11 (beige), which primarily appears on FCC/HCP interfaces. The cluster's placement in the same parent branch as cluster C8 (Brown, HCP) indicates that C11 is likely a defective HCP structure. Structures 2.8b-c show many surface particles belonging to clusters C4 (yellow) and C2 (light green). Note that Fig. 2.8b appears to show C4 (yellow) particles in the bulk, however, these are actually surface particles on an adjacent plane. Based on their placement in the cluster tree and proximity to FCC particles in the Fig. 2.8b-c, cluster C4 corresponds to FCC (100)-(111) surface particles. The C2 (light green) and C7 (black) particles are less commonly observed throughout the SA trajectory data but often appear as defective surface particles on lattices containing HCP and FCC particles. The C2 (light green) particles even occasionally appear as stacking faults (see Fig. 2.9)a, while the C7 (black) particles tend to appear as weakly-bound particles. Despite C2's placement within the cluster branch corresponding to HCP particles, C2 particles often appear above FCC bulk particles. This suggests that C2 refers to defective FCC surface particles that show some HCP-like characteristics. Each of the above classifications remain consistent in Fig. 2.8d, which shows a polymorphic FCC, HCP, and BCC lattice.

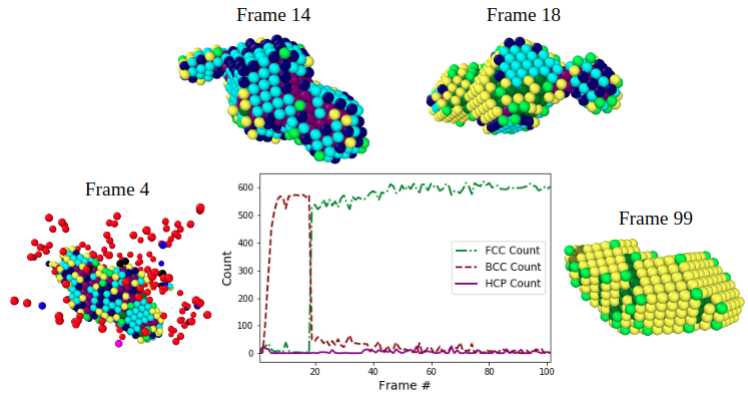
It is important to note that rigorous, direct comparisons to other dimensionality reduction-based characterization frameworks are not necessarily appropriate in this chapter. For example, the most recent implementations of diffusion maps require the choice of "landmark points" to reduce the size of the high-dimensional space before dimensionality reduction takes place [90]. As a result, diffusion maps cannot reduce the same high-dimensional space that the autoencoders can and thus cannot be applied (in the same way) to the colloidal SA trajectories discussed above. This is because analysis of these trajectories requires computing distance matrices between thousands of neighborhood graphs – and diffusion maps can become intractable for such large analyses. This chapter could have directly applied the approach of the authors of [102] to the 11 SA trajectories used to train the characterization framework. However, the elbow plot analysis (Fig. S4.2 in the SI) demonstrates that the single hidden layer autoencoder architecture employed by the authors of [102] does not encode as much information in the low-dimensional space as the proposed multiple hidden layer approach. Moreover, their use of Steinhardt bond-order parameters to create neighborhood graphs (which are much more prone to thermal fluctuations and density gradients than the proposed Delaunay triangulation-based method) indicates that the method of [102] would lead to less general classification. In fact, the work in [102] classifies each particle within a lattice as either FCC, HCP, or "fluid". Meanwhile, the presented approach classifies each particle in one of 12 different categories, which include BCC, FCC, HCP, fluid (which this chapter labels as "vapor"), and several surface and defective states.

The characterization framework is next used to analyze the time evolution of entire colloidal SA trajectories, as opposed to singular colloidal SA system states. Fig. 2.9a shows the time evolution of the colloidal SA trajectory that leads to the lattice in Fig. 2.8d. Here, the number of total particles classified as FCC (cluster C9, green), HCP (cluster C8, brown), and BCC (cluster C12, purple) is plotted against the simulation frame index. Snapshots of four key simulation frames whose particles are classified according to their positions in Fig. 2.7 are also provided. Frame #1 shows several small nuclei beginning to form. Clearly, many particles are still in the vapor phase, as clusters C1 (red) and C3 (blue) are highly prevalent. The bottom right crystallite is forming a BCC structure as evidenced by the C6 (light blue), C10 (dark blue), and C12 (purple) colored particles. Meanwhile the top right cluster primarily contains FCC/HCP particles due to its plethora of C4 (yellow), C8 (HCP), and C9 (FCC) particles. However, this crystallite also contains some BCC-like particles such as C12 (purple) and C6 (light blue). By Frame #2, the two remaining clusters are almost entirely BCC (top left) and almost entirely FCC/HCP (bottom right). The crystals' continued nucleation uncovers a few interesting trends.

First, the characterization framework reveals that a polymorphic lattice is formed by a primarily BCC structure merging with a primarily FCC/HCP structure. This indicates that the assembly conditions are likely favorable to both BCC and HCP/FCC structures. Comparing Frames #2 and #8 shows the merging of the BCC and HCP/FCC structures as part of the growth process. Frame #2 also shows that the FCC/HCP structure is initially covered with surface particles from cluster C6 (light blue), which represent BCC surface particles. However, these light blue particles nearly only exist as FCC particles by the end



(a) Example colloidal self-assembly trajectory with polymorphic lattice.



(b) Example colloidal self-assembly trajectory with phase transition.

Figure 2.9: Example colloidal self-assembly trajectories. Each figure shows the time evolution of the number of particles classified as FCC (cluster C9, green), HCP (cluster C8, brown), and BCC (cluster C12, purple) for a separate *in-silico* colloidal self-assembly trajectory. Note that Frame # refers to the (chronologically ordered) recorded simulation frame. The time evolution plots are accompanied by snapshots of certain chosen simulation frames within these trajectories. In each case, the dimensionality of the neighborhood graphs is reduced with the encoder trained using 11 isothermal trajectories of a system of 500 multi-flavored colloidal particles (see Section 2.5). Each particle in each snapshot is classified according to the proximity of its low-dimensional representation to points in Fig. 2.7a. (a) The figure shows the time evolution of an isothermal trajectory of the self-assembly of 500 multi-flavored colloids that creates the lattice in Fig. 2.8d. The trajectory shows that a polymorphic lattice containing FCC, HCP, and BCC particles forms from a primarily BCC structure merging with a structure that contains FCC and HCP particles (b) The figure shows the time evolution of an isothermal trajectory of the self-assembly of 1000 multi-flavored colloids. The trajectory shows that the system initially self-assembles into a BCC structure before undergoing a phase transition into an FCC structure.

of the trajectory. This could suggest that FCC particles take on a structure similar to that of surface BCC before finding their final state (e.g. Frame #20 and Fig. 2.8d). In fact, the idea that the interfaces of FCC crystallites retain BCC-like ordering during nucleation

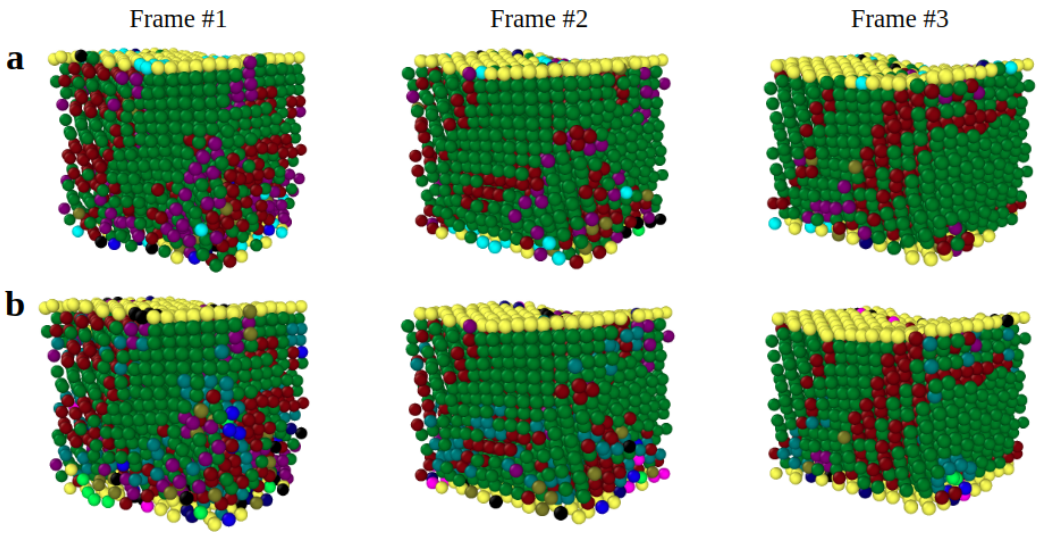


Figure 2.10: Evaporation-induced colloidal self-assembly. The figure shows 3 snapshots of the *in-silico* evaporation-induced self-assembly of 2052 colloidal particles that are classified using two different schemes. Note that the data used to create these snapshots was borrowed from reference [78] and that 6 total snapshots were provided. (a) The dimensionality of the neighborhoood graphs is reduced with an encoder trained using 11 isothermal trajectories of an *in-silico* system of 500 multi-flavored colloidal particles (see Section 2.5) Each particle in each snapshot is classified according to the proximity of its low-dimensional representation to points in Fig. 2.7a (b) The entire characterization framework is performed on the six provided snapshots of the evaporation-induced colloidal self-assembly data. Each unique neighborhood graph is used to train a second autoencoder. The newly-formed encoder is used to reduce the dimensionality of the neighborhood graphs and agglomerative hierarchical clustering (via Ward's linkage) is used to partition the low-dimensional space. In both (a) and (b), FCC particles are green, HCP particles are brown, BCC particles are purple, and surface FCC particles are yellow. The teal particles in (b) correspond to defective FCC structures that were not found by the classification scheme in (a). Overall, the two characterization procedures yielded nearly identical results.

is frequently explored [27, 129, 130]. As an aside, the characterization framework was used to carry out similar analyses for the remaining colloidal SA trajectories used to train the autoencoder, but these analyses were not included in this thesis for brevity.

The characterization framework was next applied to an independent test data set that consists of 1000 *in-silico* multi-flavored colloids undergoing SA in isothermal conditions. Here, the neighborhood graphs of each particle in each simulation frame were calculated. The chosen encoder from Section 2.6 was then used to reduce the dimensionality of the neighborhood graphs. Note that the autoencoder was not retrained and the encoder was used with the same weights and biases as determined in Section 2.6. The points in Fig. 2.7a that were closest to those corresponding to the independent data set were next identified and the particles were classified accordingly. For example, if the low-dimensional representation of a neighborhood graph from the independent data set is $[15.17, 3.50, 18.23]^\top$ and the closest point in Fig. 2.7a is classified as C2, then the particle from the independent data set adopts

this class. Fig. 2.9b shows the time evolution of the total number of particles classified as FCC, HCP, and BCC throughout this trajectory and the classification of 4 example snapshots of simulation frames.

The characterization of this independent trajectory reveals how a primarily disperse colloidal system state initially forms a cluster that almost entirely consists of BCC particles (e.g., purple bulk particles corresponding to cluster C12 and light blue and dark blue surface particles corresponding to clusters C6 and C10 respectively). A sudden, drastic phase transition occurs at Frame #18 as the system state transitions from a BCC structure to an almost entirely FCC structure. Over time, the remaining BCC particles transition slowly to FCC particles. By Frame #99, the system state is an entirely FCC structure. The fact that the characterization framework can identify a BCC/FCC phase transition in an independent data set demonstrates the framework's generalizability.

The characterization framework was finally applied to another independent data set from reference [78] that consists of 6 snapshots of 2052 *in-silico* colloids undergoing evaporation-induced SA (see Fig. 2.10). The particles were first characterized using the same methodology that was used to characterize the previous independent data set (i.e., the isothermal SA of 1000 multiflavored colloids). Fig. 2.10a shows the classification results for the final three snapshots.

The entire characterization framework was next re-performed on the evaporation-induced colloidal SA data alone. In other words, all unique neighborhood graphs from the 6 provided snapshots (4462 total unique neighborhood graphs) were collected. Autoencoders were re-trained using these 4462 unique neighborhood graphs and the elbow method analysis was re-performed. The dimensionality of the neighborhood graphs was reduced using the newly formed encoder and agglomerative hierarchical clustering using Ward's linkage was used to partition the low-dimensional space. Finally, the minimum number of clusters required to separate theoretically perfect FCC, HCP, and BCC lattices into separate clusters (11 total clusters) was chosen.

Distinct colors were assigned to each of the separate clusters and the resulting color-coordinated lattices were visualized using OVITO (see Fig. 2.10b). Note that identical colors were assigned to important classes in both Figs. 2.10a-b. For example, green is FCC, brown is HCP, purple is BCC, and surface FCC is yellow in Figs. 2.10a-b. The evaporation-induced colloidal SA data set shows significantly fewer disperse and weakly crystalline states and more defective crystalline states than the multi-flavored colloidal SA data set does. The evaporation-induced low-dimensional space is thus biased towards such crystalline structures. As a result, the classifications in Figs. 2.10a-b show some important differences. For example, Fig. 2.10b shows teal particles that clearly correspond to defective FCC particles, yet such a class was not recovered from clustering the multi-flavored data. With the exception of these small numbers of particles, however, Figs. 2.10a-b do show almost identical colloidal SA state classifications. These results not only validate the generalizability of the characterization framework (as nearly identical results were seen by training the autoencoder on different systems with vastly different particle numbers) but also highlights how larger, more diverse training data sets can further improve the characterization framework.

2.8 Conclusions and Next Steps

This chapter first demonstrated the proposed colloidal self-assembly state characterization framework on an *in-silico* system of 500 multi-flavored colloids that self-assemble under isothermal conditions. The framework not only characterized the target FCC, BCC, and HCP structures but also "discovered" several relevant defective and surface structures that allowed for greater understanding of example colloidal self-assembly trajectories. The generalizability of the characterization framework was next analyzed by applying the framework to two independent systems, one that consists of 1000 *in-silico* multi-flavored colloidal particles and self-assembles under isothermal conditions and another that consists of 2052 *in-silico* colloidal particles and undergoes evaporation-induced self-assembly. Despite successful characterization of the independent data sets, the framework can be sensitive to the nature of the data on which the autoencoder is trained (e.g., number of crystalline vs. weakly crystalline states).

Although the presented framework successfully characterized "structural" order – the framework identified which particles were apart comprised unit cells such as FCC, BCC, HCP etc. – the framework did not take particle species into account. The next chapter focuses on extending the presented characterization framework to account explicitly for particle species. More importantly, the next chapter uses this extended characterization framework to investigate the role of particle size ratio and interparticle potential well depth in influencing the three-dimensional self-assembly of binary colloidal mixtures. The chapter pays special attention to "compositional' order (i.e., how A- and B-type particles are distributed among FCC, BCC¡ HCP unit cell sites).

Chapter 3

Discovery of Compositional Order and Self-Assembly Pathways in Binary Colloidal Mixtures

Binary colloidal superlattices (BSLs) have demonstrated enormous potential for the design of advanced multi-functional materials that can be synthesized via colloidal self-assembly. However, mechanistic understanding of the three-dimensional self-assembly of BSLs is largely limited due to a lack of tractable strategies for characterizing the many two-component structures that can appear during the self-assembly process. To address this gap, this chapter presents a framework for colloidal crystal structure characterization that uses branched graphlet decomposition with deep learning to systematically and quantitatively describe the self-assembly of BSLs at the single-particle level. Branched graphlet decomposition is used to evaluate local structure via high-dimensional neighborhood graphs that quantify both structural order (e.g., body-centered-cubic vs. face-centered-cubic) and compositional order (e.g., substitutional defects) of each individual particle. Deep autoencoders are then used to efficiently translate these neighborhood graphs into low-dimensional manifolds from which relationships among neighborhood graphs can be more easily inferred. This chapter demonstrates the characterization framework on in-silico systems of DNA-functionalized particles, in which two well-recognized design parameters, particle size ratio and interparticle potential well depth, can be adjusted independently. The framework reveals that binary colloidal mixtures with small interparticle size disparities (i.e., A- and B-type particle radius ratios of $r_A/r_B = 0.8$ to $r_A/r_B = 0.95$) can promote the self-assembly of defect-free BSLs much more effectively than systems of identically sized particles, as nearly defect-free BCC-CsCl, FCC-CuAu, and IrV crystals are observed in the former case. The framework additionally reveals that size-disparate colloidal mixtures can undergo non-classical nucleation pathways where BSLs evolve from dense amorphous precursors, instead of directly nucleating from dilute solution. These findings illustrate that the presented characterization framework can assist in enhancing mechanistic understanding of the self-assembly of binary colloidal mixtures, which in turn can pave the way for engineering the growth of defect-free BSLs.

3.1 Introduction

Binary colloidal superlattices (BSLs) – highly ordered crystalline structures that consist of two sublattices formed by two types of particles – have demonstrated significant potential for the design of multi-functional materials with applications in photonics [131], optical absorption [132], sensing [4, 133], and catalysis [134, 135]. Many studies have demonstrated colloidal SA as a viable synthetic route to achieve BSLs [136–141]. However, colloidal SA is prone to form defective structures that can impact the functional properties of BSLs [142] Some of the most commonly observed defective structures include kinetically trapped amorphous aggregates and crystals that contain substitutional defects (i.e., lattices in which A- and B-type particles occupy inconsistent lattice sites) [136, 143].

Particle size ratio and interparticle potential well depth have been postulated as two of the most important design parameters for influencing the SA of BSLs [136–138, 143]. A natural question is whether these design parameters can promote the SA of defect-free BSLs. The answer to this question has not been systematically explored, however, as creating a tractable framework for accurately characterizing the many complex and possibly defective two-component structures that can appear during colloidal SA remains an open challenge. Although many methods for characterizing self-assembled colloidal structures exist in the literature [62, 63, 79, 81, 82, 85, 86, 88–93, 102, 106], the most common methods either (i) heavily rely on the concept of "cut-off" radii to determine local structure and are thus sensitive to thermal fluctuations, (ii) fail to provide quantitative information about particles whose local structure does not correspond to well-defined reference structures or templates, and/or (iii) rely on diffusion mapping methods that can become intractable for systems with large configurational phase spaces. Most importantly, only three reported characterization methods explicitly account for particle type and, thus, can identify substitutional defects in BSLs [87, 102, 144]. These three methods depend on cut-off radii, diffusion maps, and/or have only been shown to characterize either two-dimensional or very simple three-dimensional lattices.

This chapter presents a framework that employs branched graphlet decomposition with deep learning to address the above challenges for characterizing the SA of three-dimensional BSLs. Branched graphlet decomposition evaluates local structure via "structural" and "compositional" neighborhood graphs that are robust to thermal fluctuations, provide quantitative information about particles whose local structure does not correspond to well-defined reference structures, and explicitly account for particle type. These neighborhood graphs quantify both structural order (i.e., the unit cells within BSLs such as FCC, BCC, and HCP) and compositional order (i.e., how A- and B-type particles are distributed among these unit cell sites). Deep autoencoders [100, 101] are next used to efficiently translate the high-dimensional structural and compositional neighborhood graphs into low-dimensional structural and compositional spaces where it is easier to infer relationships among neighborhood graphs. As such, the presented framework can simultaneously characterize the thousands of unique and defective structures that can appear during SA. The framework can distinguish defective, nearly defective, and ordered lattices and can thus precisely elucidate entire

complex nucleation pathways. The framework is summarized in Fig. 3.1.

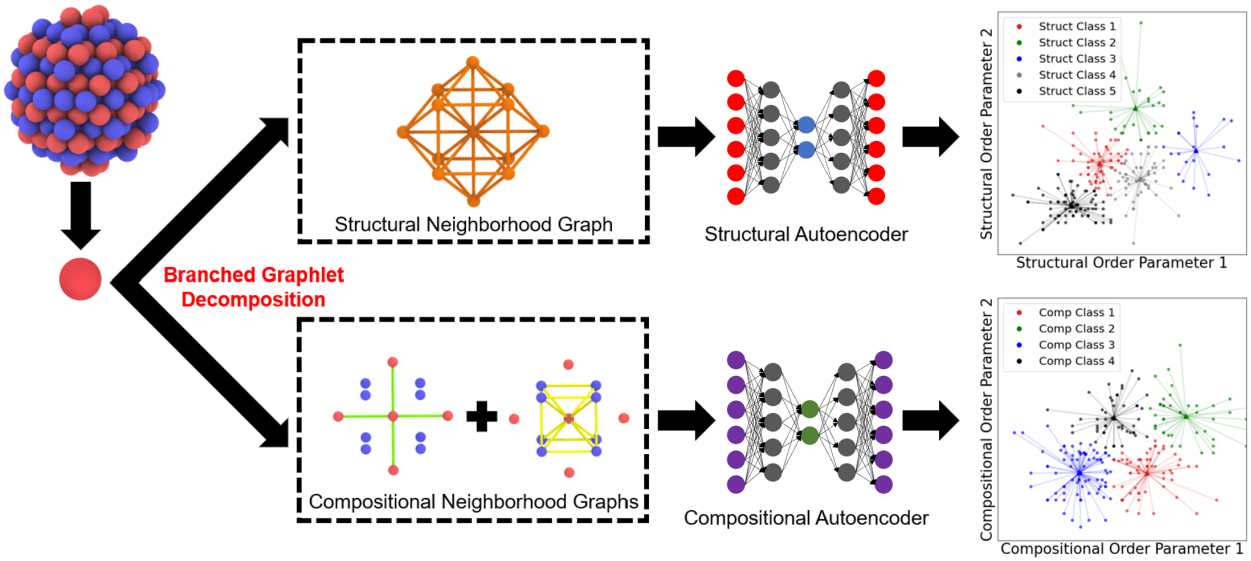


Figure 3.1: Colloidal self-assembly state characterization framework summary. Branched graphlet decomposition translates particle positions into one structural neighborhood graph and two compositional neighborhood graphs for each particle in the two-component lattice. The structural neighborhood graph evaluates the structure to which all particles contribute while the compositional neighborhood graphs evaluate each component's contribution to that structure. The dimensionality of the neighborhood graphs is next reduced using deep neural networks called autoencoders to create structural and compositional low-dimensional spaces. Agglomerative hierarchical clustering is finally used to partition the low-dimensional spaces and assign discrete classifications to each particle.

The characterization framework is demonstrated through the systematic investigation of the SA of an *in-silico* system of 500 DNA-functionalized particles (DFPs) in which particle size ratios and attractive interactions between A-type and B-type particles can be varied independently. The framework reveals that small increases in particle size disparity can drastically reduce the number of substitutional defects in FCC and HCP lattices, while further increasing the size disparity leads to the formation of (nearly) defect-free BCC-CsCl, FCC-CuAu, and IrV crystals. The framework further reveals that mixtures of size-disparate colloids that form nearly defect-free BSLs can undergo non-classical nucleation pathways in which a dense amorphous precursor is formed prior to the final binary crystalline phase. The fine control of colloidal SA using (small) size disparity suggests a promising future research direction for synthesizing defect-free BSLs and the transformation pathway analysis provides a deeper mechanistic understanding of the SA of binary colloidal mixtures.

3.2 Neighborhood Graph Construction via Branched Graphlet Decomposition

The first step in characterizing the structure of a given colloidal particle is to establish that particle's "neighborhood." This neighborhood is established using the exact methodology described in Section 2.2. Importantly, this method avoids the concept of bonds or "cut-off radii" among particles and instead uses a geometry-based, fixed number of particles to establish the neighborhood. This method is thus less sensitive to thermal fluctuations, density gradients, and structural anisotropy than many common structural characterization methods [63, 79] (e.g., Steinhardt bond order parameters [81], common neighbor analysis [106]).

The neighborhood is then used to construct one structural neighborhood graph and two compositional neighborhood graphs that quantify the structural and compositional local topology of the particle of interest. The structural neighborhood graph is composed of all particles in the neighborhood, while the compositional neighborhood graphs are composed of (i) all particles that are of the same species of the particle of interest and (ii) all particles that are of a different species than the particle of interest. Note that the work in this chapter is tailored towards binary lattices in which only two types of particles exist (e.g., A- and B-type particles). The work described in Section 2.2 used Delaunay triangulation to construct structural neighborhood graphs [63, 79] for colloidal characterization. A key contribution here is thus "branching" the structural neighborhood graphs created by Delaunay triangulation to create and evaluate additional compositional neighborhood graphs.

The structural neighborhood graph alone can be used to identify unit cells within a lattice as FCC, BCC, HCP, etc. The compositional neighborhood graphs, however, can be used to quantify how A- and B-type particles are distributed among the FCC, HCP, and BCC unit cell sites. The compositional neighborhood graphs can thus be used to identify substitutional defects within BSLs. To my knowledge, only three reported characterization methods can explicitly account for particle types within lattice sites [87, 102, 144]. In addition to using cut-off radii, these methods did not explicitly identify substitutional defects.

The structural and compositional neighborhood graphs are evaluated using the graphlet decomposition method described in Section 2.2. Remember that this methodology evaluated neighborhood graphs through the use of graphlets with 2-5 nodes that display 73 different automorphism orbits [74]. As a result, each particle's structural and compositional neighborhood graphs is quantified by a 73×1 vector, where each entry in the vector refers to the weighted frequency of an automorphism orbit. The structural neighborhood graph thus becomes a 73×1 vector, while the two compositional neighborhood graphs are concatenated to form one 146×1 vector. From this point forward the term "structural neighborhood graph" will refer to the 73×1 vector and the term "compositional neighborhood graph" will refer to the 146×1 vector.

Note that heterogeneous graphlet decomposition methods have also been reported in the literature.[145, 146] Although these methods explicitly take node type (or in this case, par-

ticle type) into account, they can be difficult to implement and computationally expensive, thus requiring several simplifying assumptions to become tractable. The branched graphlet decomposition strategy outlined above is simple, tractable, and does not require any (extra) simplifying assumptions.

3.3 Dimensionality Reduction via Deep Autoencoders

The high dimensionality of the neighborhood graphs and non-uniformity in the distances among their discrete entries indicate that dimensionality reduction must be performed to produce a continuous, low-dimensional manifold where relationships among neighborhood graphs can be more intuitively analyzed. The dimensionality of the structural and compositional neighborhood graphs is reduced using a self-supervised deep neural network called an autoencoder [79, 100, 101], which is conceptually explained and justified in more detail Section 2.3. Specifically, two separate autoencoders are trained to create one structural and one compositional low-dimensional space that are subsequently used for structural and compositional classification.

This chapter focuses on how the size ratio and interparticle potential well depth between A- and B-type particles affect the SA of FCC, HCP, BCC, IrVA, IrVB, DCsClA, and DCsClB-like BSLs from an *in-silico* system of binary DFPs, with particular attention to the substitutional defects within these target lattices. This chapter seeks to use the presented characterization framework to identify substitutionally defective (i.e., structurally ordered, yet compositionally disordered) and defect-free (i.e., structurally and compositionally ordered) versions of these lattices. That is, this chapter looks to identify particles whose local environments conform to target structures with and without substitutional defects. To this end, particle position data for isothermal *in-silico* SA trajectories of DFPs over a range of size ratios and interaction potential well depths was collected. Structural and compositional neighborhood graphs for each particle in each simulation frame were recorded according to the branched graphlet decomposition method. The unique neighborhood graphs (45,032 unique structural neighborhood graphs and 4,814 unique compositional neighborhood graphs) were then used to train the structural and compositional autoencoders respectively. The structural and compositional autoencoders then translated all unique structural and compositional neighborhood graphs from all recorded trajectories into two separate three-dimensional spaces (see Fig. 3.14 in Section 3.9 for the justification of the size of the low-dimensional spaces). More details describing the pair potential and system simulations can be found later in Section 3.5, while more details describing the autoencoder training can be found in Section 3.9.

3.4 Partitioning the Low-Dimensional Spaces for Structural and Compositional Classification

Similar to Section 2.4, this chapter implements agglomerative hierarchical clustering (with Ward's linkage) on the low-dimensional data in order to partition the structural and compositional low-dimensional spaces. Although the strategy produces a cluster tree (known as a dendrogram) that shows the hierarchical structure of all 1 to N possible cluster distributions, the process of choosing the "best" number of clusters is somewhat subjective [79, 122, 123]; see Section 3.9 and Fig. 3.15 for the justification of the choice of the number of clusters in each low-dimensional space. Since the topologies of the theoretically perfect versions of FCC, HCP, BCC, IrVA, IrVB, DCsClA, and DCsClB lattices are known, this information was used to calculate structural and compositional neighborhood graphs and the corresponding low-dimensional points of these theoretically perfect lattices. However, "defect-free" or "ordered" lattices may have neighborhood graphs that correspond to a number of different low-dimensional points. As a result, any cluster that contains one of these theoretically perfect lattice points can be analogously labeled.

Using this strategy, 14 discrete classifications were identified (see Table 3.1 in Section 3.9), with 7 structurally and compositionally ordered (CO) target lattice groups and 7 structurally ordered, yet compositionally disordered (CD) target lattice groups. For example, if a given particle's structural and compositional low-dimensional representations fall under "FCC" identified clusters in those respective low-dimensional spaces, the particle will be labeled as a "structurally and compositionally ordered FCC particle" (i.e., "CO-FCC"). If only the particle's structural low-dimensional representation falls under an FCC cluster, the particle will be labeled as "structurally ordered, yet compositionally disordered FCC" (i.e., "CD-FCC"). Note that "CO-FCC", "CO-BCC", and "CO-HCP" are referred to as "FCC-CuAu", "BCC-CsCl", and "HCP-straight" throughout the paper. The remaining groups are exclusively described using "CD" and "CO" labels (e.g., CO-IrVA, CD-DCsClB, CD-FCC). If the particle's low-dimensional representation does not fall under any target lattice cluster, the particle is left unlabeled. These "unlabeled" particles can correspond to vapor particles, structurally defective particles, surface particles, non-target lattice groups, etc. Note that in principle labels could be assigned to the clusters that correspond to many of the unlabeled particles, but this was not viewed as necessary for investigating the SA of our binary systems of DFPs. Finally note that the term "structurally ordered" (SO) is used throughout the paper to refer to a particle that belongs to any CD or CO group (i.e., a particle that is not unlabeled).

Fig. 3.2 shows an example of how the characterization framework can classify CD and CO particles within BSLs. Here, A- and B-type particles within a perfect spherical FCC-CuAu lattice are manually swapped over time. Swap attempts are only accepted if the potential energy of the new configuration is higher than that of the current configuration (see Fig. 3.18 for a plot of lattice potential energy versus swap moves/frames). This ensures that progressively more energetically unfavorable lattices are created (i.e., that A-type

particles are only swapped for B-type particles and vice-versa). The (trained) framework then identifies the number of CO and CD FCC particles in each corresponding simulation frame. Figs. 3.2a-b show that both the number of CO particles and A-B nearest neighbors decrease with the number of manual swaps. This agrees with the fact that BSLs become progressively more compositionally disordered as A- and B-type particles swap positions. Fig. 3.2c shows visualizations of classified and unclassified lattices. The open-source codes of the characterization framework are available on GitHub [105].

Note that the characterization framework classifies ≈10 near-surface particles in the defect-free lattice in Frame 0 in Fig. 3.2b as compositionally disordered. This "misclassification" is a reflection of the fact that the 14 discrete classes used in this chapter are defined based on the bulk structures. This choice in turn can lead the framework to occasionally mis-classify particles near the surface (as these particles can have different neighborhood topologies than bulk particles). To reduce this imprecision, more classes (i.e., label more clusters) can be defined that identify surface or near-surface ordered and disordered particles. In fact, such near-surface order was identified in the work in Chapter 2 that only focused on structural order [79]. Identifying near-surface structural and compositional order was not viewed as necessary for the work in this chapter; see Section 3.9 for more details.

3.5 Colloidal Self-Assembly System Descriptions

The purpose of this chapter is to use the presented characterization framework to investigate the SA of an *in-silico* system of binary DFPs under various interparticle size ratios and interaction potential well depths. The pair potential model and resulting molecular dynamics simulations used throughout this chapter are described in more detail below. The exact same pair potential model was actually used in Chapter 2. However, the work in Chapter 2 focused exclusively on systems in which A- and B-type particles were identically sized. The work in this chapter additionally involves systems with different A- and B-type particle size ratios. Most of the writing below is identical to that in Chapter 2 but is repeated here for convenience.

Pair potential model

DFPs interact with each other through complementary Watson–Crick base-pairing interactions. As a means of achieving selective binding among DFPs, particles can be functionalized with a blend of two types of DNA strands with complementary concentrations on each particle. By changing the blending ratio of these two types of DNA strands, these "multi-flavored" particles can exhibit a tunable attraction between the like particles while maintaining interactions between unlike pairs. This approach has been shown to induce the crystallization of equally sized particles into BCC, HCP, and FCC structures [72, 125–127].

The tunable and independent pairwise interactions of DFP colloidal mixtures in this chapter are modeled using the Fermi-Jagla pair potential (see Eq. (3.1)), which previously

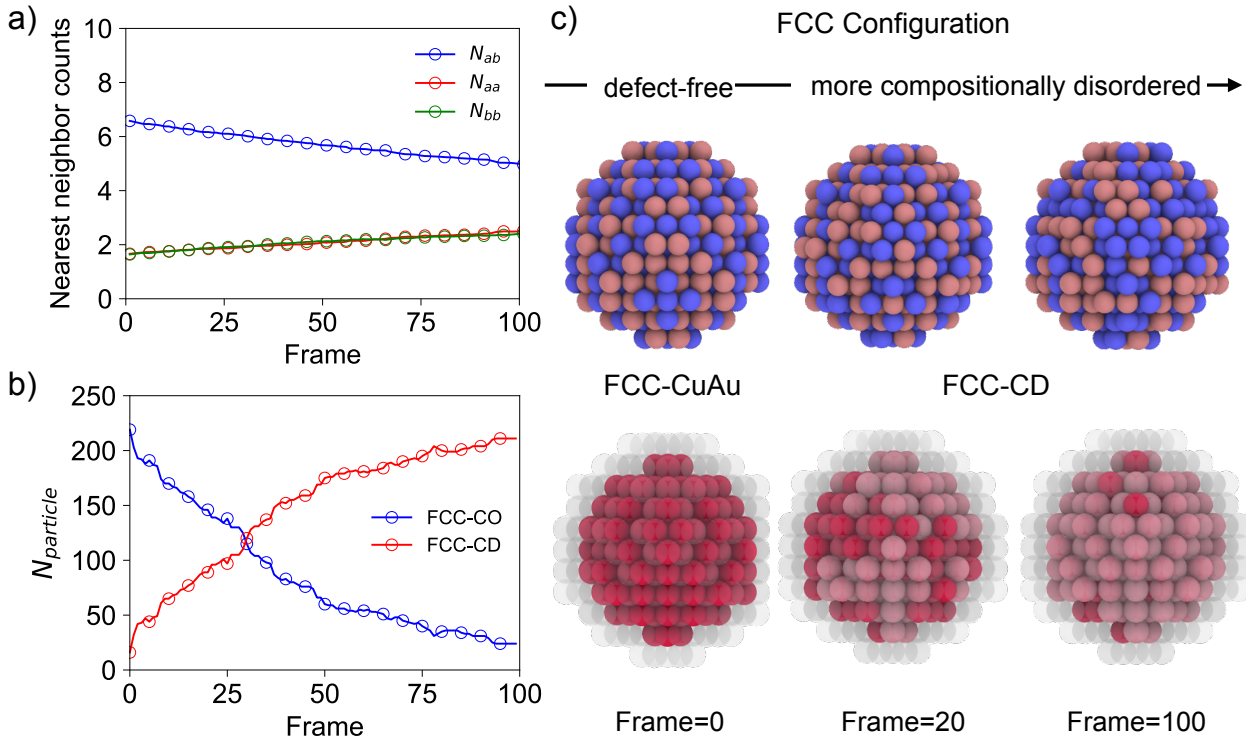


Figure 3.2: Classified colloidal binary superlattices. The A- and B-type particles within a perfect spherical FCC-CuAu lattice are manually swapped over time (i.e., simulation frames). Swap attempts are only accepted if the potential energy of the new configuration is higher than that of the current configuration. (a) The number of like (A-A, B-B) and unlike (A-B) nearest neighbors ($\#N_{ij}$) is plotted against the simulation frame number. (b) The presented characterization framework identifies the number of structurally and compositionally ordered (CO) and structurally ordered, yet compositionally disordered (CD) particles in each frame. Note that both the number of CO particles and A-B nearest neighbors N_{AB} decrease over time. (c) Snapshots of lattices where A-type particles are colored blue and B-type particles are colored orange sit above snapshots of lattices that are classified by the characterization framework. In the latter case, CO particles are colored dark red, CD particles are colored light red, and particles that are not structurally ordered are transparent. Frame 0 contains the perfect FCC-CuAu configuration. Frames 20 and 100 contain lattices that have gone through several swapping attempts.

has been used to represent binary DFPs effectively in both two and three dimensions [72, 73]. The first term in Eq. (3.1) represents the particle-particle core repulsion, where ϵ_c represents the energy scale of the repulsion, σ_c represents the length scale of repulsion and R_s is a shifting factor related to particle size. The second and third terms capture the soft repulsion and attraction from DNA sequences, respectively. A_0 and B_0 control the strength of these interactions, while A_1 and B_1 control the interaction range. A_2 and B_2 control the separation distance.

$$\frac{U(r/\sigma)}{\epsilon} = \frac{\epsilon_c}{\epsilon}\Big(\frac{\sigma_c/\sigma}{r/\sigma - R_s/\sigma}\Big)^n + \frac{A_0/\epsilon}{1 + exp[A_1(r/\sigma - A_2)]} - \frac{B_0/\epsilon}{1 + exp[B_1(r/\sigma - B_2)]}. \tag{3.1}$$

To tune interparticle potentials, unlike pair interaction E_{AB} are kept fixed and like interactions E_{AA} and E_{BB} are varied independently. The relative like interaction strength $E^*_{AA} = E_{AA}/E_{AB}$ and $E^*_{BB} = E_{BB}/E_{AB}$ can thus be adjusted independently from 0.0 to 1.0. E^*_{AA} is set to equal E^*_{BB} throughout all simulations unless otherwise noted. Note that setting E^*_{AA}= E^*_{BB}= 1.0 reduces the multi-flavoring to single flavoring, where all particles are identical. Setting $E^*_{AA} = E^*_{BB} = 0.0$ makes the system a conventional binary mixture, where A-A and B-B interactions are purely repulsive and only A-B interactions are attractive. On the other hand, the particle size ratio r_A/r_B is tuned by varying the size of A-type particles within a range of $\sigma = 0.8$ to 1.0 while maintaining the size of B-type particles at $\sigma = 1.0$. Setting $r_A/r_B = 1.0$ makes all particles the same size. Setting $r_A/r_B = 0.8$ means B particles are 20% larger than A particles. Fig. 3.3 shows the pair potentials of multi-flavored, binary micron-sized DFPs with parameter choices similar to those used in this chapter. These DFPs interact with each other via tunable and independent pairwise interactions E_{AA}, E_{BB}, and E_{AB}.

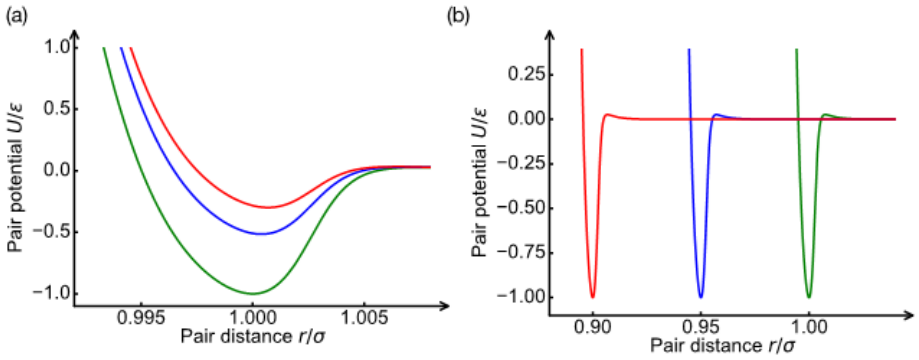


Figure 3.3: (a) Example pair potentials with independent and tunable pairwise interactions E_{AA}, E_{BB}, and E_{AB} for identically sized particles at $\sigma = 1.0$. The red, blue, and green curves represent $E_{AA} = -0.3\epsilon$, $E_{BB} = -0.5\epsilon$, and $E_{AB} = -\epsilon$. These epsilon values are achieved by tuning B_0 to values of 0.56, 0.8, and 1.32 respectively. (b) Example pair potential with different particle sizes. Red, blue, and green curves represent particle sizes of $r_A = 0.9$, $r_{AB} = 0.95$ and $r_B = 1.0$. These sizes are achieved by tuning values of σ to 0.9, 0.95, and 1.0 respectively. All parameter values used to create these plots are provided in SI Table S3.

Simulation details

Molecular dynamic (MD) simulations are performed using LAMMPS [128] in the canonical ensemble. The system contains 500 total particles with a 1:1 mixture ratio of A-type and B-type particles. The interaction strength and size ratio are varied using the pair potential model discussed above. Simulations are performed in a cubic box with periodic conditions applied to all three dimensions, under dilute conditions with number density $\rho = 0.02\sigma^{-3}$, and using a Langevin thermostat with a time constant $\tau = 2\sigma m^{1/2}\epsilon^{-1/2}$. Each simulation involves 1×10^9 total time steps where each time step is $\Delta t = 10^{-3}\sigma m^{1/2}\epsilon^{-1/2}$. Each MD simulation is performed at a constant, pre-determined temperature suitable for crystallization starting from a random dilute liquid phase, where particles are allowed to evolve spontaneously to form crystals. The entire SA process can be tracked and quantified by the characterization framework demonstrated above. The trajectories generated from these simulations are visualized using Open Visualization Tool (OVITO) [124]. The parameter values used in Eq. 3.1 are: $\epsilon_c = 10\epsilon$, $n = 36$, $A_0 = 11.035\epsilon$, $A_1 = 404.4/\sigma$, $A_2 = 1.0174\sigma$, $\sigma_0 = 0.2\sigma$, $s = 0.8\sigma$, $B_0 = [-1.3219\epsilon, 0$, $B_1 = 1044, 5/\sigma$, and $B_2 = 1.0306\sigma$.

3.6 Small Size Disparity Promotes Compositional order of Binary Superlattices

This section illustrates that slightly tuning particle size ratio and attractive interaction strength can change the structural order of self-assembled BSLs and can promote the formation of structurally and compositionally ordered (defect-free) BSLs. The effects of small size disparity and interaction strength on mediating the SA of BSLs are demonstrated with the binary *in-silico* system of 500 DFPs discussed in Section 3.5. In this system, particle size (i.e., r_A and r_B) and inter-particle interaction strengths (i.e., E_{AA}, E_{BB} and E_{AB}) can be tuned independently, which provides additional flexibility and control over the SA process in comparison to standard binary colloidal systems.

The presented characterization framework is used to classify each particle within self-assembled BSLs over a range of relative attractive interaction strengths $E^*_{AA} = E_{AA}/E_{AB}$ ($E^*_{BB} = E^*_{AA}$) and size ratios $r^* = r_A/r_B$. Fig. 3.4a shows how size ratio can change the structural order of self-assembled BSLs. At size ratio $r^* = 1.0$, the formed BSLs are polymorphic, primarily CD, randomly close-packed FCC/HCP structures. Slightly increasing the size disparity results in the formation of FCC-CuAu or polymorphic FCC-CuAu/HCP-straight BSLs. More interestingly, further increasing the size disparity leads to the formation of two more different BSL structures, IrV and distorted CsCl (DCsCl). These two structures can be viewed as deformed versions of BCC-CsCl, in which the increasing size difference and interaction strength of the two species forces some particles to lie either too far apart or too close to one another (see Fig. 3.8). The impact of interaction strength on the formation of IrV or DCsCl structures suggests that the presence of enthalpic driving forces from pairwise interactions adds a degree of freedom (in addition to size ratio) for mediating the SA of BSLs.

Note that the self-assembled BSLs listed here are much richer than systems of hard spheres [147, 148] as well as sticky spheres [126, 149] at similar size ranges, where substitutionally disordered FCC is the most dominant structure.

Fig. 3.4a also shows that an amorphous zone, where particles are trapped in disordered amorphous states, exists for systems of size-disparate particles. Note that poly-dispersity/bi-dispersity are two commonly used parameters for inhibiting crystallization. Crystallization is usually suppressed for systems above 5% poly-dispersity (or 15% bi-dispersity) [150, 151]. As shown in Fig. 3.4a, crystallization suppression is widely observed at extremely high E^*_{AA} and becomes more pronounced for colloidal mixtures with larger size-disparities. However, well-ordered BSLs can still be assembled at relatively weak E^*_{AA}. These observations reinforce the importance of proper selection of E^*_{AA} for adopting different structural ordering of BSLs and demonstrate E^*_{AA} as an important design parameter for promoting or inhibiting crystallization.

More importantly, the characterization framework and simulations reveal that a slight increase in size disparity can change the compositional order of formed BSLs. For example, in the case of identically sized particles, substitutionally defective CD-CP lattices are formed over nearly the entire parameter space ($E^*_{AA} > 0.2$). As E^*_{AA} increases, the fraction of CO particles within BSLs further decreases (see Fig. 3.10). This suggests that the bulk crystals become more and more substitutionally defective, despite the fact that the primary crystals remain structurally ordered. These observations agree well with previous simulation and experimental work that shows that FCC-CuAu crystals change to substitutionally defective FCCs with increasing like-particle interaction strength [126]. Although colloidal mixtures with identically sized particles are prone to form CD BSLs, the presented simulation results reveal size disparity as an exclusive design parameter that can promote the formation of defect-free BSLs. For mixtures with small size disparity, the CO BSL structure FCC-CuAu can be formed within a much larger parameter space of E^*_{AA}, extended from $E^*_{AA} = 0.2$ to $E^*_{AA} = 0.7$. Fig. 3.4c quantifies the fraction of identified CO particles within different types of BSLs as a function of size ratio. Defect-free FCC-CuAu BSLs form at a size ratio of $r^* = 0.95$ and $E^*_{AA} = 0.3$, whereas substitutionally defective CD-CP lattices form at size ratio $r^* = 1.0$. As the size disparity further increases, IrV and DCsCl — two BSLs that are structurally different than the BCC/FCC/HCP-like lattices — are formed. However, the particles within these lattices are usually CO particles, illustrating the universality of the impact of size disparity for reducing the number of CD particles within BSLs. All of these observations indicate the importance of size disparity, and how small changes in this parameter (at certain interaction strengths) can radically change the structural and compositional ordering of self-assembled BSLs.

Anti-site formation penalties provide a plausible explanation for slight size disparity leading to a reduced number of substitutional defects in self-assembled BSLs. It has been previously reported that the anti-site formation penalty decreases as E^*_{AA} increases for systems of identically sized sticky colloidal particles [126]. In these systems, the relative interaction strength is the main parameter that drives the SA of BSLs. As E^*_{AA} increases, less enthalpic penalty is introduced when an A-type particle occupies a site where a B-type particle should

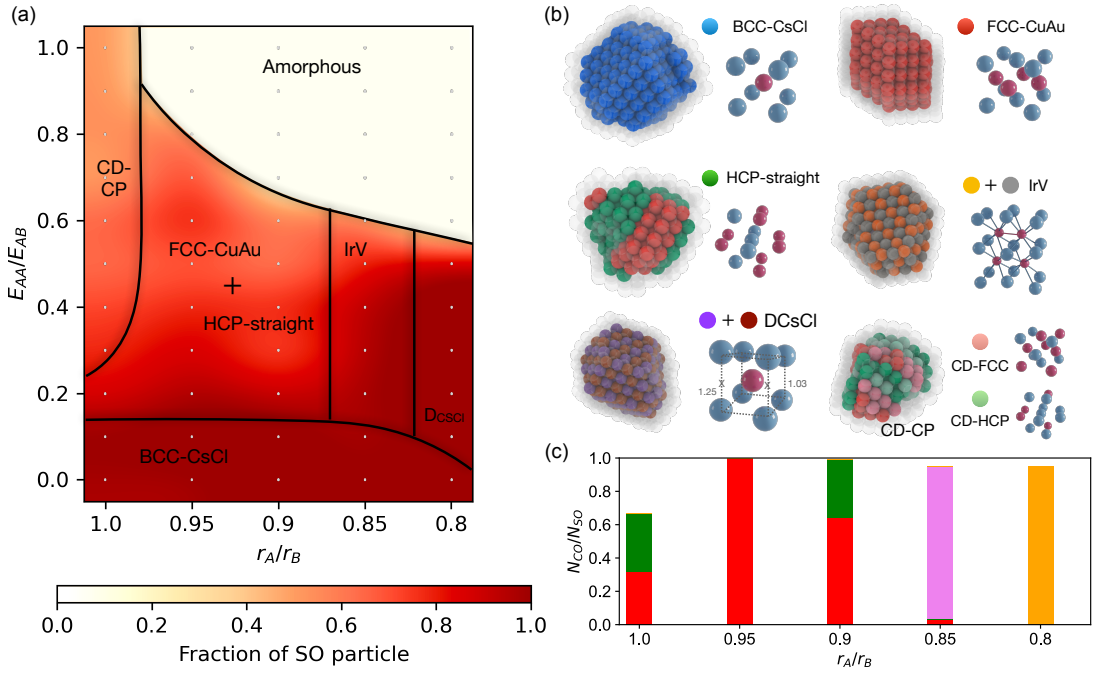


Figure 3.4: (a) Crystallization order diagram as a function of particle size ratio, $r^* = r_A/r_B$, and relative like interaction strength, $E^*_{AA} = E_{AA}/E_{AB}$ ($E^*_{BB} = E^*_{AA}$). MD simulations (see Methods section) are performed at a variety of size ratios and interaction strengths that are indicated by the gray dots. The characterization framework classifies each particle in the final snapshot of each simulation according to SI Table S1. The color bar represents the fraction of structurally ordered (SO) particles in these final snapshots; the fraction calculation is normalized by the number of SO particles in a perfect FCC spherical lattice. Each region within the order diagram is labeled based on the specific classifications of the SO particles. In the compositionally disordered close-packed (CD-CP) region, structurally ordered, yet compositionally disordered (CD) FCC and HCP particles are observed, which form polymorphic and randomly packed lattices. In the FCC-CuAu and HCP-straight region, structurally and compositionally ordered (CO) FCC and HCP particles are observed, which form FCC-CuAu lattices and polymorphic HCP-straight/FCC-CuAu lattices. CO BCC particles are observed in the BCC-CsCl region. In the IrV and DCsCl regions, CD and CO IrVA, IrVB, DCsClA, and DCsClB particles are observed, which form CD/CO and CO IrV and DCsCl lattices. The data for conditions favoring different BSLs is provided in Fig. 3.9. (b) Snapshots of characterized BSLs obtained from the simulations in (a) and their crystal unit cells. Note that IrV and DCsCl classifications are based on two types of SO particles since the structural graphlet for A-type and B-type particles is different for these two crystals. The transparent particles represent surface or amorphous particles that are not explicitly identified by the characterization framework. (c) The ratio of the total number of CO particles (N_{CO}) to the total number of SO particles (N_{SO}) is plotted for different size ratios r_A/r_B at $E_{AA}/E_{AB} = 0.3$. The red, green, pink, and orange bars quantify FCC-CuAu, HCP-straight, and CO IrVA/B and DCsClA/B, respectively. $N_{CO}/N_{SO} = 1.0$ suggests that all particles within BSLs are structurally and compositionally ordered particles (i.e., defect-free).

be present. Fig. 3.5 shows the calculated anti-site formation penalties for both identically sized and size-disparate particles. This figure shows that a slight size disparity significantly raises the anti-site formation penalty. It is thus reasonable to hypothesize that particle size disparity could guide the formation of BSLs at the early stage of nucleation. The next section uses the presented characterization framweork to investigate this hypothesis. This section shows that size disparity can assist in promoting the formation of defect-free BSLs through unique pathways, such as non-classical transformations during colloidal SA.

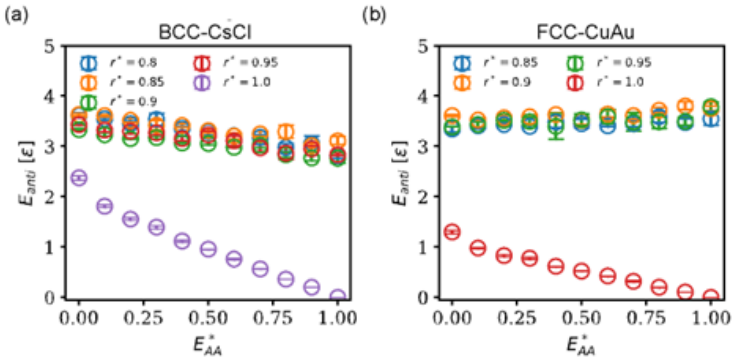


Figure 3.5: Anti-site formation energy is plotted against E*AA at different size ratios starting from (a) BCC-CsCl configuration and b) FCC-CuAu configurations.

3.7 Probing Self-Assembly and Structural Evolution Processes

The presented characterization framework demonstrates the mechanistic details of how BSLs evolve (or nucleate) from dilute solutions. The characterization framework reveals that BSL nucleation pathways either occur via one-step (classical) or two-step (non-classical) processes (Figs. 3.6a-c). Fig. 3.6b demonstrates how BSLs can self-assemble by classical one-step nucleation. Here, the fraction of identified SO crystalline particles, the fraction of identified CO particles, and the total largest cluster size are plotted over time. First, a small crystalline nucleus with an FCC-CuAu (CO) structure is formed. This small crystal nucleus then grows into a larger size, and the final stabilized crystal is identical in structure with the initially formed nuclei. Fig. 3.6c shows how the nucleation of BSLs can also proceed by non-classical two-step nucleation. Here, instead of forming a small crystal nuclei, the particles rapidly form large disordered amorphous aggregates with very few crystalline particles within these clusters. Subsequently, these disordered amorphous clusters evolve into an ordered BSL, as indicated by the continuous growth of identified SO particles.

Figs. 3.6d-e further demonstrate the differences between the nucleation processes of size-disparate and identically sized particles over a broader parameter space of E^*_{AA}. For colloidal

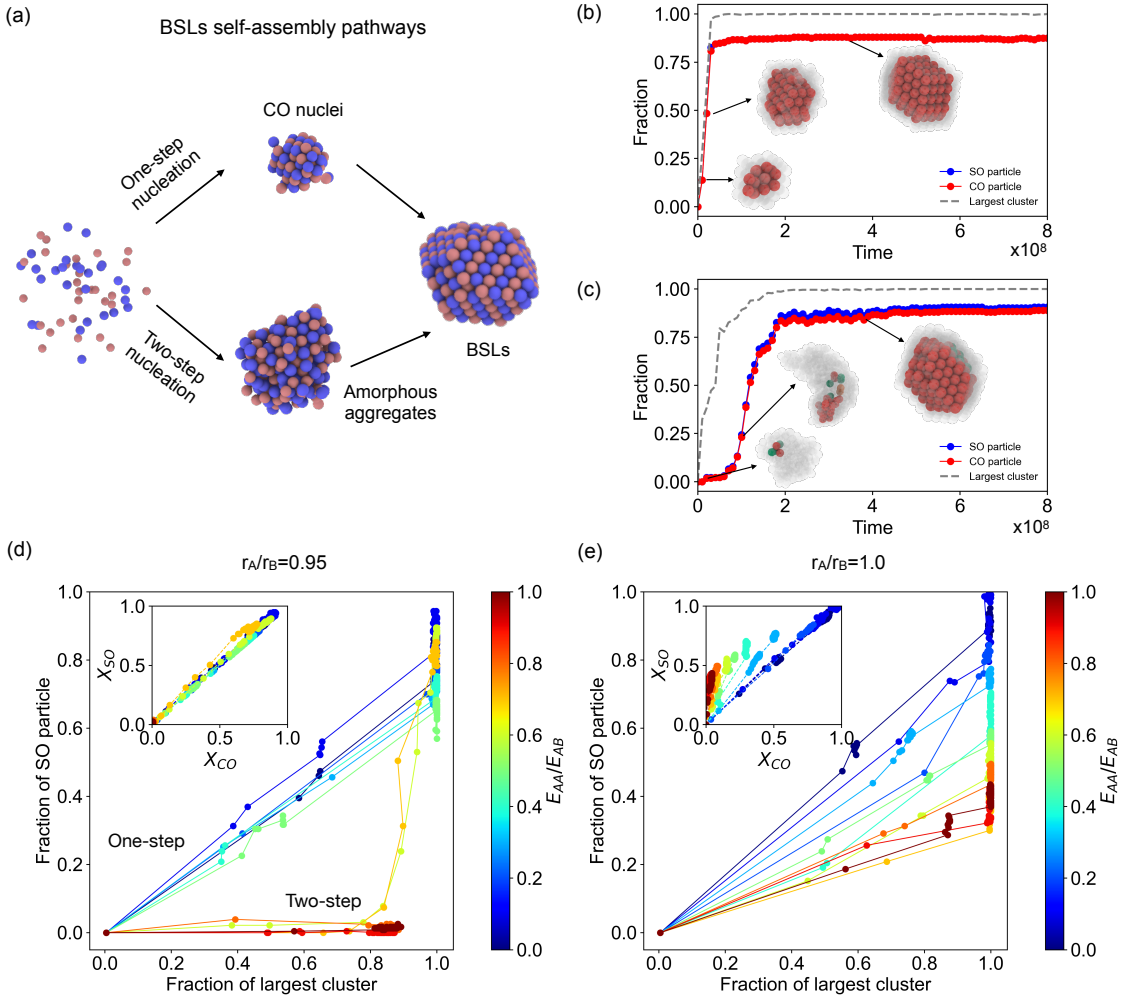


Figure 3.6: (a) Schematic illustration of self-assembly pathways for forming BSLs. The SA of BSLs can either occur via a one-step or two-step nucleation process. (b) Example of a one-step nucleation pathway observed at E^*_{AA}=0.3 and r^*=0.95. (c) Example of a two-step nucleation pathway (amorphous-crystal) observed at E^*_{AA}=0.6 and r^*=0.95. The self-assembly process is quantified by plotting the fraction of identified structurally ordered (SO) particles (blue curves), structurally and compositionally ordered (CO) particles (red curves), and largest cluster size (dashed gray curves) as a function of time. The inset snapshots show identified crystalline particles at the single-particle level at different times. The particle coloring scheme is same as that of Fig. 3.4b. (d) Quantification of self-assembly pathways for size-disparate systems at size ratio r_A/r_B=0.95. (e) Quantification of self-assembly pathways for identically sized systems at size ratio r_A/r_B=1.0. The plots (d) and (e) show the fraction of SO particles within the largest cluster for different E^*_{AA} (color bar), while the insets show the fraction of SO particles (X_{SO}) as a function of the fraction of CO particles (X_{CO}).

mixtures in which one-step nucleation occurs, the SO crystal fraction grows linearly with the size of the largest cluster. In contrast, for colloidal mixtures in which two-step (amorphous-

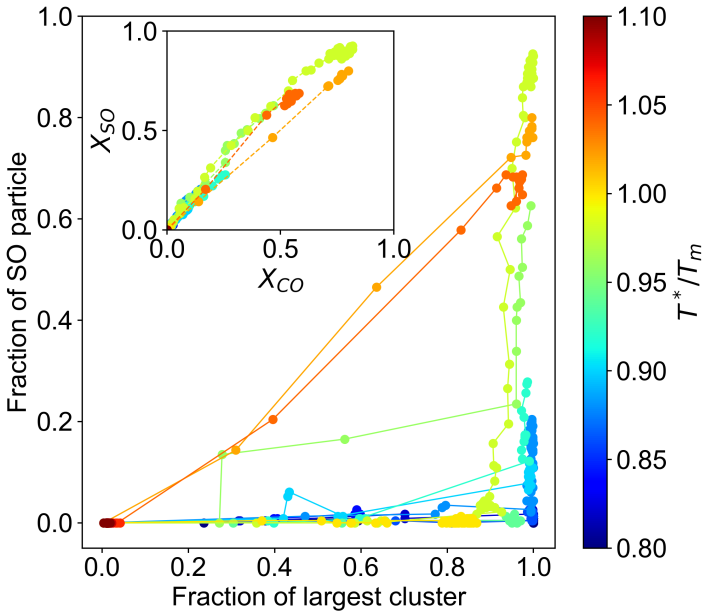


Figure 3.7: Temperature-dependent self-assembly behavior for size-disparate particles at r^* =0.95 and E^*_{AA}=0.7. The figure plots the fraction of structurally ordered (SO) particles within the largest cluster against the largest cluster fraction at different degrees of supercooling T^*/T_m (color bar). T_m is the pre-estimated temperature suitable for crystallization obtained from cooling simulations (see Fig. 3.12 in Section 3.9). The inset shows the fraction of SO particles (X_{SO}) as a function of compositionally and structurally ordered (CO) particles (X_{CO}).

solid) nucleation occurs, no crystal is identified until the largest cluster size reaches about 80% of the total system size. An abrupt increase in crystal fraction is then observed after this initial amorphous state. Interestingly, the two-step nucleation usually occurs for size-disparate particles and relatively high E^*_{AA} close to the boundary of amorphous states (other size ratios are provided in Fig. 3.11). For this relatively high interaction range ($E^*_{AA} = 0.3$ to $E^*_{AA} = 0.7$), however, identically sized particles usually nucleate via a one-step process and form highly CD crystals. Unlike size-disparate particles that can transform from disordered amorphous clusters into BSLs via a diffusionless process, the identically sized particles usually form CD crystalline nuclei quickly at the initial nucleation stage. Such CD nuclei then continuously grow larger in size and remain trapped in substitutionally disordered crystalline phases.

These results illustrate that size-disparate particles can form BSLs via a two-step process: particles first aggregate into disordered amorphous clusters and then rearrange into crystalline BSL structures. Note that the observation of two-step nucleation pathways directly contradicts the well-recognized classical nucleation theory (CNT) [152, 153], suggesting a more complex picture of the transformation mechanism for BSLs. While CNT is a

widely used rule for characterizing nucleation of particles from the solution phase, more and more evidence now supports two-step nucleation's occurrence in nature [154–158]. One commonly believed reason for the emergence of two-step nucleation pathways is the supercooling/supersaturation that can occur within colloidal SA systems. In colloidal SA, two-step nucleation has been reported once the liquid is deeply quenched [159]. In such cases, amorphous aggregates are formed initially before they sluggishly transform into crystals depending on the temperature and cooling rate. Previous work has also suggested that slow particle mobility brought on by supercooling can inhibit crystallization and promote the formation of amorphous aggregates [160]. The successful transformation from amorphous to crystalline phases is assumed to be caused by an interplay between thermodynamics and kinetics. Although the full picture of the emergence of two-step nucleation requires more theoretical calculations, the presented characterization framework provides a way to quantify efficiently the emergence of well-ordered crystalline nuclei from many defective crystalline nuclei or amorphous aggregates at the particle level. This capability allows users to probe into the SA details of BSLs at early nucleation stages under supercooling. Such probing is otherwise not achievable in experiments, as nucleation is usually a rare event that is generally difficult to capture and quantify.

This chapter next investigates further BSL nucleation to determine the role of supercooling in influencing the observed nucleation pathways. Simulation results show that, for size-disparate systems that previously underwent two-step nucleation, raising temperature causes a tendency towards the observation of one-step nucleation pathways that result in well-defined CO crystals (Fig. 3.7). Two-step nucleation usually occurs under moderate supercooling. Under deeper supercooling, more amorphous particles were identified during the SA process. However, even under these highly undercooled conditions, size-disparate particles within dense amorphous aggregates still tend to rearrange into more ordered (but not "well-ordered") BSL structures – despite the process becoming slower with further lowering temperatures. The reduced mobility of the particles in the clusters must significantly impact the kinetics and inhibit the transformation from amorphous to well-defined binary crystalline structures. A similar tendency of suppression of crystallization is observed for identically sized particles (see Fig. 3.13 in Section 3.9). However, neither raising nor lowering temperature improves the formation of defect-free BSLs. Mixtures of identically sized particles rather tend to be kinetically trapped in structures that are formed early on during SA, either in highly compositionally disordered BSLs or more structurally disordered amorphous aggregates at lower temperatures.

The above analysis has shown that particle size disparity can assist in the formation of defect-free BSLs through two unique nucleation pathways: particles can either rearrange from amorphous aggregates into BSLs under moderate supercooling or can directly nucleate and grow into larger BSLs at higher temperatures. Note that raising the temperature can drive crystallization mechanisms from two-step (non-classical) to one-step (classical). The observation of such transitions is similar to those previously observed in an NaCl solution [161] or a Lennard-Jones fluid system [162]. In the NaCl solution, single-step nucleation is observed before the solution reaches the spinodal regime and two-step nucleation is observed

after the solution reaches the spinodal regime. Similarly, in the Lennard-Jones fluid system, a cross-over from a classical nucleation regime to a more collective mechanism of freezing is observed, influenced by the existence of a spinodal singularity at higher supercooling. Experimentally, it is also reported that two-step nucleation is widely observed, especially in DFP systems [158, 163]. Nonetheless, the cooling rate, quench temperature window as well as the details of particles can all impact the SA pathways. For instance, the presence of DNA molecules around particles could result in dramatic sluggishness of rearrangement of amorphous aggregates into crystalline structures due to the presence of hybridization kinetics [163]. Accordingly, it may be valuable to investigate further the relationship between these parameters and crystallization transition mechanisms to build a proper interpretation of the nucleation of BSLs.

3.8 Conclusions and Next Steps

This chapter presented a framework for characterizing the self-assembly of binary colloidal mixtures based on branched graphlet decomposition and deep learning. The characterization framework was demonstrated by investigating the self-assembly of binary mixtures of DNA-functionalized particles while varying two well-recognized design parameters, i.e., particle size ratio and pairwise interaction potential. The investigation revealed that size disparity at certain interaction potentials can improve the structural diversity of self-assembled BSLs, leading to the formation of BCC-CsCl, FCC-CuAu, IrV, DCsCl and CD-CP lattices. As a comparison, systems of hard spheres (without presence of pairwise interactions) assemble a limited range of rFCC-like structures. The presented framework also revealed that small A/B particle size ratios can drastically reduce the number of substitutional defects within BSLs and, thus, promote the formation of defect-free BSLs.

The proposed characterization framework can pave the way for systematic and computationally efficient investigation of the underlying mechanisms of the self-assembly of BSLs. The above analysis showed that size-disparate colloidal mixtures can undergo two-step, non-classical nucleation pathways where BSLs evolve from dense amorphous precursors, instead of directly nucleating from dilute solution in one step. Interestingly, size-disparate mixtures tend to form (nearly) defect-free BSLs, regardless of their adopted nucleation pathway. On the other hand, systems of identically sized particles always follow one-step classical nucleation pathways, but often become kinetically trapped in substitutionally defective structures in the early stage of nucleation. Thus, the fine control of self-assembly of defect-free BSLs using size-disparate particles under given conditions can facilitate potential approaches to engineer defect-free BSLs. The proposed framework can be easily adapted to investigate the underpinning mechanisms of other colloidal self-assembly systems.

The next chapter begins by using autoencoders for full lattice characterization of colloidal SA systems (as opposed to single particle characterization). In other words, autoencoders are used to discover a handful of order parameters that can describe the topology of the entire lattice. Next, deep neural networks are used to create a dynamic model that predicts

the time evolution of the reduced-order system state as a function of previous states and the change in temperature. This dynamic model is then integrated into a model-based feedback control framework in order to guide the two-dimensional SA of an *in-silico* system of colloids. The efficacy of the framework is then analyzed and used to determine next steps.

3.9 Supplementary Information

Autoencoder Training

The structural and compositional autoencoders are trained using the unique structural and compositional neighborhood graphs mentioned in Section 3.3 (45,032 unique structural neighborhood graphs and 4,814 unique compositional neighborhood graphs). For each type of autoencoder, the unique neighborhood graphs are split into training/validation/testing data sets at 60%/20%/20% ratio. For each autoencoder, hyperbolic tangent activation functions are chosen in the hidden layers while linear activation functions are chosen for the output layer. Dropout regularization is used to prevent overfitting [109], a batch size of 32 is used during training, and an MSE loss function is used. The optimal network size (i.e., number of nodes and layers) for reach autoencoder is found by plotting autoencoder training loss as a function of network size and implementing the "elbow method". This method selects the network architecture with the best balance of computational cost and performance (see Fig. 3.14) [114, 115]. Note that the number of nodes in the bottleneck layer corresponds to the size of the low-dimensional space (i.e., the number of "order parameters"). Finally, the autoencoder was implemented using the Python library Keras (a TensorFlow API) [116, 117].

Note that this autoencoder training strategy differs slighlty from the autoencoder training strategy described in Section 2.3. In Section 2.3, the data was not split into training/validation/testing data sets due to the fact that inconsistent neighbor lists at solid-vapor interfaces yielded a large number of "outlier" neighborhood graphs (i.e., the 73×1 entry neighborhood graph vectors). In this chapter, neighborhood graphs were not recorded for particles in which the inter-particle distance exceeded the distance at which the inter-particle potential is zero. This distance is obviously known *a priori* for *in-silico* systems.

Partitioning the low-dimensional spaces for structural and compositional classification

Once the autoencoder models are chosen and the (unique) structural and compositional neighborhood graphs have been translated to their low-dimensional analogs, the next step is to partition these low-dimensional spaces into discrete regions to make final decisions regarding structural identity. The main text mentions that:

1. Agglomerative hierarchical clustering with Ward's linkage is implemented to partition the low-dimensional spaces

2. The agglomerative hierarchical clustering procedure creates a cluster tree that shows the hierarchical structure of all 1 to N possible cluster distributions for the structural and compositional low-dimensional spaces

3. The choice of the "best" number of clusters for analysis is somewhat subjective

4. Clusters are labeled based on the low-dimensional coordinates of theoretically perfect versions of certain structures of interest

5. The structures of interest are FCC, HCP, BCC, IrVA, IrVB, DCsClA, and DCsClB

This section is meant to provide more details about the clustering/labeling process.

For the structural and compositional low-dimensional spaces, Fig. 3.15 plots the number of low-dimensional points (that represent neighborhood graphs) corresponding to FCC, HCP, BCC, IrVA, IrVB, DCsClA, and DCsClB structures against the number of total clusters in each branch of the cluster tree. For the structural low-dimensional space, the cluster size for all 7 relevant groups stabilizes around 410 clusters. The structural low-dimensional space is then partitioned with 410 clusters, 7 of which correspond to the target structures. The compositional dimensionality reduction reveals that the (compositional) low-dimensional representations of several theoretically perfect lattices are nearly identical. As a result, three different clusters represent 7 target structures. These clusters are (i) FCC-HCP-IrVB, (ii) BCC-DCsClB, and (iii) IrVA-DCsClA. Fig. 3.15 shows that the cluster size for these three groups stabilizes around 290 total clusters. The compositional low-dimensional space is then partitioned accordingly.

The cluster trees for the structural and compositional low-dimensional spaces at 410 and 290 clusters are shown in Fig. 3.16 with target structures labeled. Table S1 summarizes the 14 target groups used in the analysis in the main text along with their cluster IDs. Remember that "compositionally ordered" and "compositionally disordered" labels refer to particles whose local environments conform to a target structure without and with substitutional defects, respectively. Fig. 3.17 shows visual examples of some select lattices that are classified according to the descriptors in Table 3.1. These lattices are selected from the 55 isothermal self-assembly trajectories used to train, validate, and test the characterization framework mentioned in the main text. Fig. 3.15 is meant to give the reader a flavor of the types of polymorphic and defective crystals that the framework is capable of classifying.

Supplementary Figures

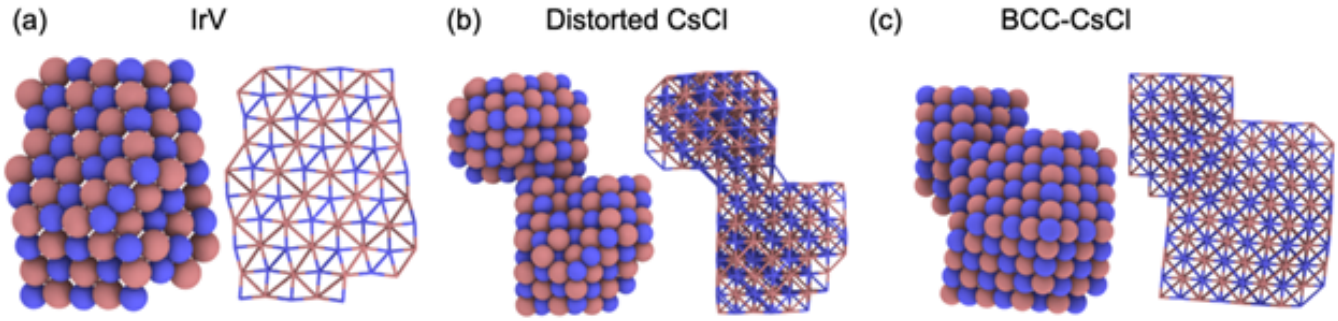


Figure 3.8: The above snapshots show crystal structures obtained from simulation trajectories of (a) IrV at $E^*_{AA} = E^*_{BB} = 0.4$ and $r^* = 0.85$, (b) distorted CsCl at $E^*_{AA} = E^*_{BB} = 0.4$ and $r^* = 0.8$, and (c) BCC-CsCl at $E^*_{AA} = E^*_{BB} = 0.4$ and $r^* = 1.0$. The figures on the left show the snapshots obtained from simulations and the figures on the right show bonds between all particles for better illustration.

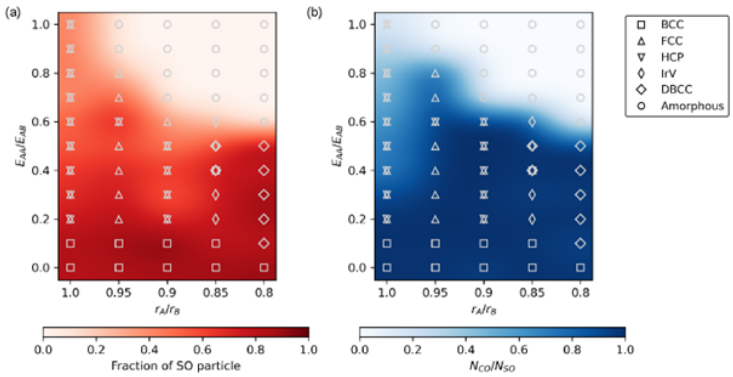


Figure 3.9: (a) Structural crystallization order diagram as a function of particle size ratio $r^* = r_A/r_B$, and relative like interaction strength $E^*_{AA} = E_{AA}/E_{AB}$. The symbols showing the predominant crystal structures obtained from simulation trajectories. The red colormap shows the fraction of structurally ordered particles (i.e., BCC, FCC, HCP, IrV, distorted BCC). (b) Compositional crystallization order diagram as a function of particle size ratio $r^* = r_A/r_B$, and relative like interaction strength $E^*_{AA} = E_{AA}/E_{AB}$. The blue colormap shows the fraction of the number of compositionally ordered particles (i.e., BCC-CsCL, FCC-CuAu, HCP-straight,IrV-CO, DCsCl) over structurally ordered particles (N_{CO}/N_{SO}).

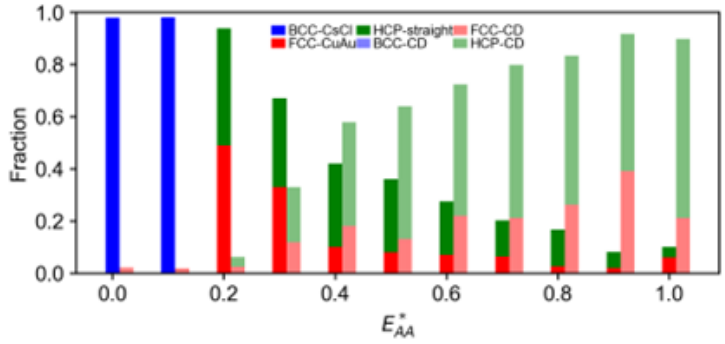


Figure 3.10: The fraction of identified crystals is plotted against E^*_{AA} for systems of identically sized particles. The structurally and compositionally ordered BSLs are plotted as solid color bars while the structurally ordered yet compositionally disordered particles are plotted as lighter color bars.

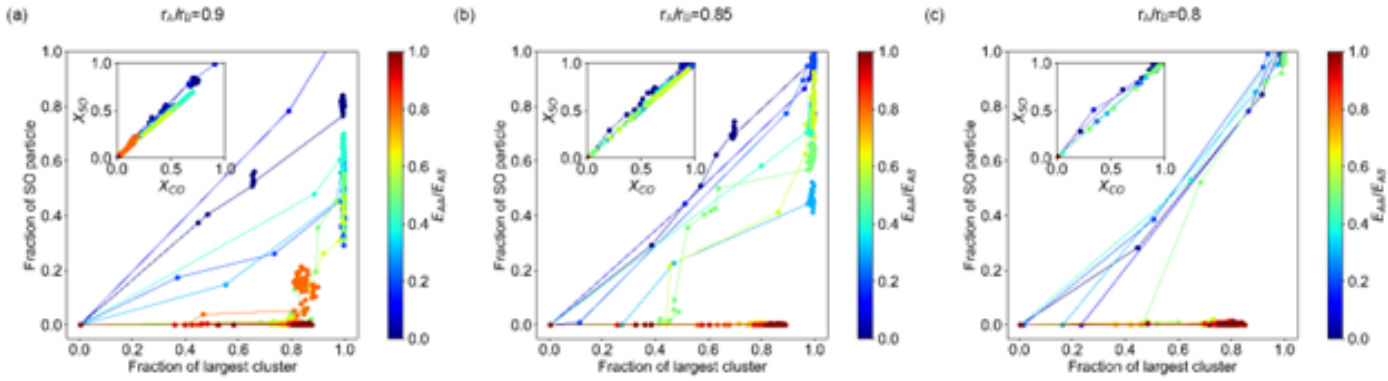


Figure 3.11: Quantification of self-assembly pathways for size ratios (a) $r_A/r_B = 0.9$, (b) $r_A/r_B = 0.85$, and (c) $r_A/r_B = 0.8$. The plot shows the fraction of structurally ordered (SO) particles within the largest cluster for different E^*_{AA} (color bar), while the inset shows the fraction of SO particles (X_{SO}) as a function fraction of compositionally ordered (CO) particles (X_{CO}).

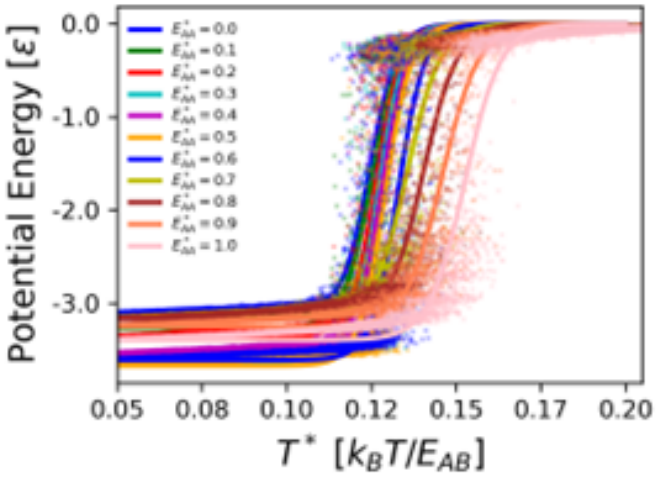


Figure 3.12: Example plot for estimating suitable crystallization temperature T_m at a given size ratio $r^* = 0.95$. The T_m is obtained from fitting curves to a sigmoidal form $PE(T) = PE_{min} + (PE_{max} - PE_{min})/(1 + exp(a(T - T_m))$

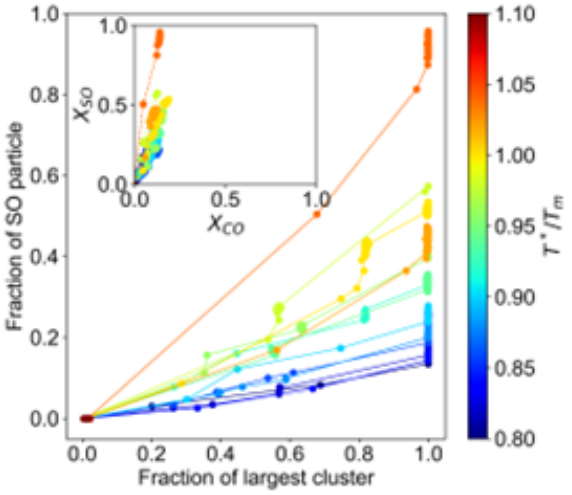


Figure 3.13: Temperature-dependent self-assembly behavior at $r^* = 1.0$ and $E^*_{AA} = 0.7$. (a) The fraction of structurally ordered (SO) particles is plotted against the largest cluster fraction under different degrees of supercooling T*/Tm. The inset shows the fraction of SO particles (X_{SO})as a function fraction of compositionally ordered (CO) particles (X_CO).

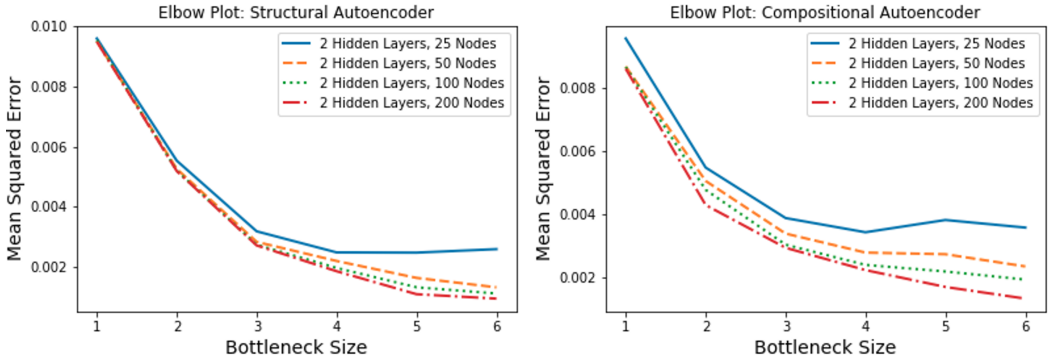


Figure 3.14: Autoencoder architecture optimization. For both the structural and compositional autoencoders, the autoencoder MSE is plotted against the number of nodes in the bottleneck layer (i.e., the length of the low-dimensional representation vector) for various network sizes). "Elbows" in these plots occur around a bottleneck size of 3 nodes, indicating a low-dimensional size of 3 is likely sufficient to capture the essential information from the neighborhood graphs. For both the structural and compositional cases, autoencoders with two hidden layers, 100 hidden nodes, and 3 bottleneck nodes show the lowest MSE. These models are used throughout the main text.

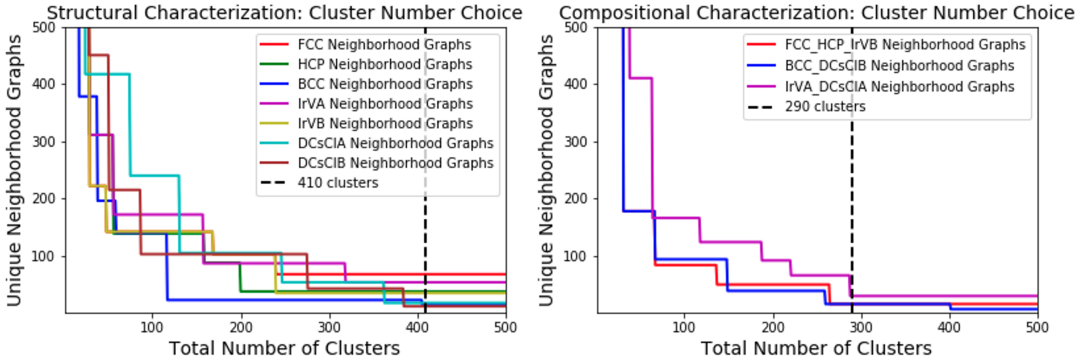


Figure 3.15: Analysis to determine number of clusters. Agglomerative hierarchical clustering (using Ward's linkage) is used to cluster the structural and compositional low-dimensional representations the unique neighborhood graphs identified from the 55 colloidal self-assembly trajectories that were used to train, validate, and test the autoencoder (see main text). The number of unique neighborhood graphs corresponding to FCC, BCC, HCP, IrVA, IrVB, DCsClA, and DCsClB structures is plotted against the number of clusters in each branch of the resulting cluster tree for both low-dimensional spaces. In the structural low-dimensional space, the target structure cluster size stabilizes around 410 total clusters. For the compositional low-dimensional space, the target structure cluster size stabilizes around 290 total clusters. Both low-dimensional spaces are then partitioned accordingly.

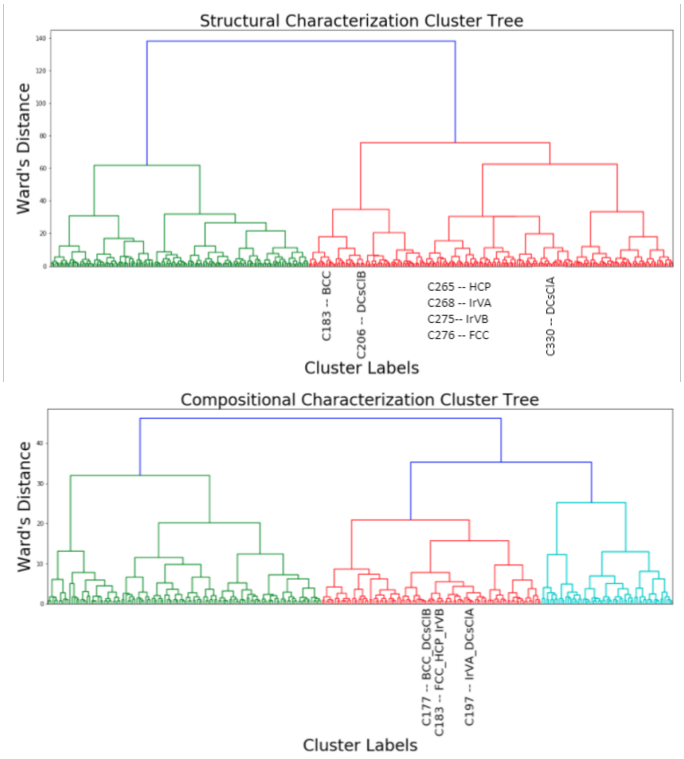


Figure 3.16: .Structural and compositional cluster trees. The structural cluster tree (with 410 clusters, see Fig. 3.15) and the compositional cluster tree (with 290 clusters, see Fig. 3.15) are show above. The branches corresponding to target clusters are labeled accordingly.

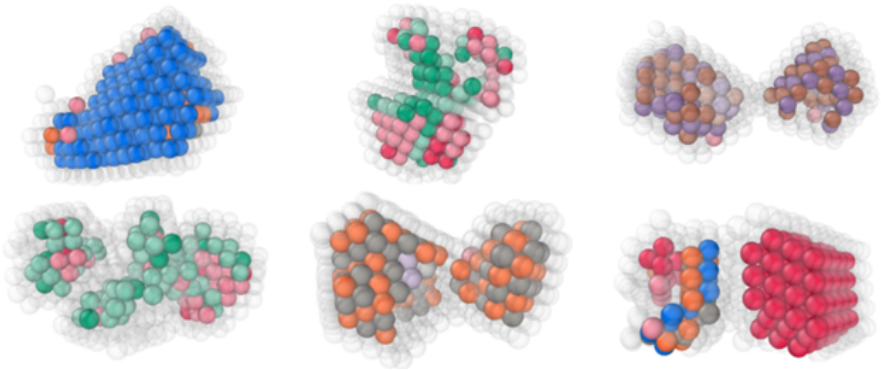


Figure 3.17: Various classified lattices. The figure shows several binary colloidal lattices that are classified by the deep learning-based characterization framework. These lattices are selected from the 55 isothermal self-assembly trajectories used to train, validate, and test the characterization framework (see main text). Table S1 pairs the colors with their physical classifications. Overall, the framework is capable of distinguishing various types of structurally and compositionally ordered, structurally ordered yet compositionally disordered, and fully defective particles within (polymorphic) lattices.

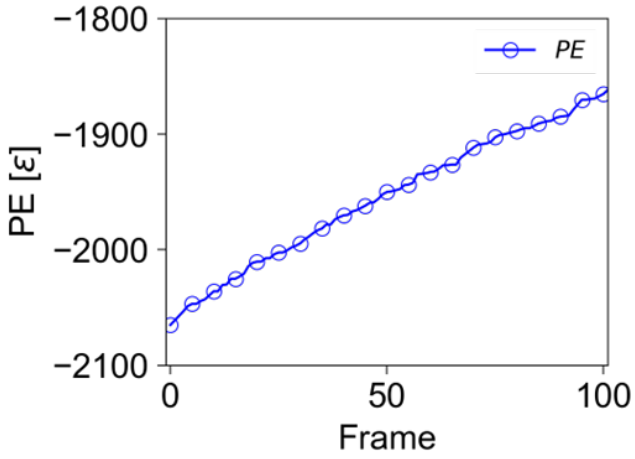


Figure 3.18: Potential energy over time in swapping procedure. In Fig 2 in the main text, the A- and B-type particles within a perfect spherical FCC-CuAu lattice are manually swapped over time (i.e., simulation frames). Swap attempts are only accepted if the potential energy of the new configuration is higher than that of the current configuration. This figure measures the potential energy as more swap moves occur.

Supplementary Tables

Table 3.1: Lattice classifications. This table summarizes the labels and cluster IDs associated with the target structures of interest. "CO" stands for "structurally and compositionally ordered" while "CD" stands for "structurally ordered but compositionally disordered". The colors in the last column are used in Fig. 3.17 and Fig. 3.2-3.4. Note that throughout this chapter, FCC-CO, BCC-CO, and HCP-CO are often referred to as FCC-CuAu, BCC-CsCl, and HCP-straight. All other groups are referred to exactly as they are listed above.

Label	Structural Cluster ID	Compositional Cluster ID	Color
FCC-CO	C276	C183	Red
FCC-CD	C276	Any except C183	Light Red
HCP-CO	C265	C183	Green
HCP-CD	C265	Any except C183	Light Green
BCC-CO	C183	C177	Blue
BCC-CD	C183	Any except C177	Light Blue
IrVA-CO	C268	C197	Brown
IrVA-CO	C268	Any except C197	Light Brown
IrVB-CO	C275	C183	Violet
IrVB-CD	C275	Any except C183	Light Violet
DCsClA-CO	C330	C197	Orange
DCsClA-CD	C330	Any except C197	Light Orange
DCsClB-CO	C206	C177	Gray
DCsClB-CD	C206	Any except C177	Light Gray
Other	None of the Above	Any	Transparent

Chapter 4

Model-Based Feedback Control of Colloidal Self-Assembly Systems

This work proposes a model-based feedback control framework to modulate systematically temperature to guide the two-dimensional self-assembly of an in-silico system of DNA-functionalized particles towards a defect-free crystal. This work first uses deep autoencoders to discover order parameters that can quantify the system state. A deep neural network is then used to model the colloidal self-assembly dynamics in terms of these order parameters. The model-based feedback control framework then manipulates the system temperature based on the current system state and predictions from the dynamic model. The approach can reliably guide the test system towards desired structures.

4.1 Introduction

To drive SA systems towards desired structures more reproducibly, it has been proposed to design a feedback control policy wherein a global actuator is manipulated based on currently available information on the system state [13, 37, 38, 47]. Although such strategies provide an added level of robustness to both intrinsic and extrinsic system disturbances (and thus kinetic arrest), the existing approaches have relied on control strategies that are either not scalable for practically-sized systems (e.g., use dynamic programming) [37, 38], are too simple to handle complex dynamics (e.g., use PID control) [13], cannot quantify the state of complex lattice structures [47], or employ models that rely on Bayesian estimation techniques whose computational cost scales exponentially with the number of states [37, 38, 59]. For these reasons, such techniques have only been applied to small systems whose dynamics are nearly solely governed by repulsive forces.

Systematic modulation of the global manipulated variables of colloidal SA systems to manufacture reproducibly highly-ordered crystalline materials thus remains an open grand challenge. The key objective of this work is to investigate a model-based feedback control strategy for systematically modulating the inherently stochastic and nonlinear dynamics of

colloidal SA systems via a global actuator (e.g., temperature). The control strategy is developed and tested on an *in-silico* system of DNA-functionalized particles (DFPs). The system state is quantified using order parameters that are discovered from a deep neural network called an autoencoder [100, 101]. The system dynamics are modeled by a second deep neural network whose architecture can be easily adapted to multivariate systems. The model-based feedback control strategy avoids the curse of dimensionality and can handle state and input constraints to more effectively guide the colloidal SA. The effects of modeling strategies, state constraints, and controller tuning on the control strategy's efficacy are explicitly analyzed, and suggestions for improved controller performance are provided.

4.2 Colloidal Self-Assembly System Description

The test system, originally presented in [164], is a two-dimensional *in-silico* system that is comprised of 500 "multi-flavored" DFPs. These multi-flavored DFPs are functionalized with identical blends of complementary, single-stranded DNA (ssDNA) and ideally self-assemble into a perfect, two-dimensional hexagonal crystal. The reader can refer to Section 2.5 for more details on multi-flavored DFPs.

Uncontrolled SA of this system has been often observed to lead to defective, semi-crystalline structures [164]. Particle interaction potentials are primarily determined by repulsive interactions between the underlying silica particles, repulsive interactions due to ssDNA chain overlap, and attractive interactions due to ssDNA hybridization [73, 164]. The primary energetic driving force for particle assembly is ssDNA hybridization. Since changes in temperature can be used to turn ssDNA bonds "on" or "off" (i.e., can cause ssDNA annealing or melting), SA for this system can be actuated by changing the temperature (i.e., the "global manipulated variable" or "system input"). The overarching purpose of this work is to investigate a model-based feedback control strategy for systematically guiding SA systems towards desired structures. The purpose of developing this control strategy on a test system is to demonstrate the "proof of concept" of the strategy. The model-based feedback control strategy was developed and tested using molecular dynamics (MD) simulations that represent the key components of the system dynamics. The use of an MD representation of the process allowed for efficient collection of large data sets that are then used to estimate and evaluate the control strategy. The essential physics of the multi-flavored motif are captured by a modified Lennard-Jones (LJ) potential used in previous works that investigated the dynamics of the test system [73, 164]. The MD simulations were performed using the software package LAMMPS [128] with a canonical ensemble at a low packing fraction of 8%. All pairwise interactions are truncated at $r = 3ij$, where ij represents the distance between particles i and j. The simulations were performed in a square, periodic box with an edge length of 71σ. The Verlet algorithm [165] was used to integrate the equations of motion using a time step of $t = 0.01\sigma\sqrt{m/\epsilon_{ij}}$, where m represents particle mass and ϵ the interparticle potential. A Langevin thermostat was used with a damping constant of $\sigma_{ij}\sqrt{m/\epsilon_{ij}}$.

4.3 Colloidal State Characterization

Although the time-scales for certain systems vary, the time-scale for most colloidal SA systems (including the test system) is on the order of seconds [14, 23]. The first step in developing a system model amenable to online control is then to quantify the system state accurately with limited computational cost. Common Neighbor Analysis (CNA) is a computationally cheap yet reasonably accurate technique to classify configurations of complex crystal structures [62, 83, 84, 87, 106]. The technique identifies local structure by constructing a characteristic signature from the connectivity of particles' nearest neighbors (where "nearest" is defined as particles within a pre-determined cut-off radius). This signature is a vector of variable-length discrete integer values that can uniquely identify to which lattice structure an individual particle belongs. A complex system configuration can thus be determined by computing the CNA signature of each individual particle within that system. The method can explicitly identify a number of defective, quasi-crystalline, or perfect unary or binary crystal lattices (as opposed to merely assessing general order). CNA can also adapt the length of the cut-off radius to account for thermal fluctuations. Even observing SA trajectories in terms of CNA signatures allows for a simple visualization of common defects, correlated lattice structures, and common assembly/disassembly pathways. Chapters. 2-3 chose not to employ CNA in favor of methods based on Delaunay Triangulation and graphlet decomposition. This decision was primarily motivated by the fact that CNA depends on cut-off radiuses, which are inherently somewhat arbitrary [79]. However, the characterization methods of Chapters 2-3 are meant for investigating colloidal SA trajectories off-line and are prohibitively expensive for the online control applications on which this chapter focuses.

CNA yields discrete signatures that represent local particle structure at the single particle level. Online control applications, however, require low-dimensional, continuous descriptions of the entire system state. This work first quantifies the entire system state based on the fraction of particles with certain CNA signatures. Given that hundreds (or even thousands) of unique CNA signatures can appear during SA, this chapter uses a deep autoencoder to reduce these CNA signature fractions to a handful of order parameters.

The autoencoder was trained, validated, and tested with 5 time series, each of which has a length of 4000 sampling times. Each time series began at a separate starting configuration, ranging from dispersed to a perfect crystal. A sampling time of 500 time-steps was used, where a time-step is defined as the dimensionless characteristic time of the system (i.e., $t^*/t = \sqrt{(\epsilon/m\sigma^2)}$, [164]). At each sampling time, an input (i.e., temperature ramp rate) is chosen from a random, uniform distribution. The bounds of this distribution at each sampling time are dictated by the physical limits in maximum and minimum temperature and temperature ramp rates for the system. The temperature ramp rates are assumed to be achievable and the temperature over the system at any given sampling time is assumed to be uniform. These temperature assumptions are used throughout the remainder of this chapter. The time series were divided into three different data sets: 60% for training the model, 20% for validating the model, and 20% for testing model.

Over this dataset, 193 unique CNA signatures were detected and each simulation frame

was thus quantified by a 193×1, where each entry in the vector represents the fraction of a different CNA signature. Following the procedure discussed in Section 3.9, the "elbow" method was employed to choose a final autoencoder architecture and subsequent order parameters (see Fig. 4.1. In all, a bottleneck layer size of three nodes was chosen, yielding a low-dimensional space of size 3×1. the autoencoder was implemented using the Python library Keras (a TensorFlow API) [116, 117].

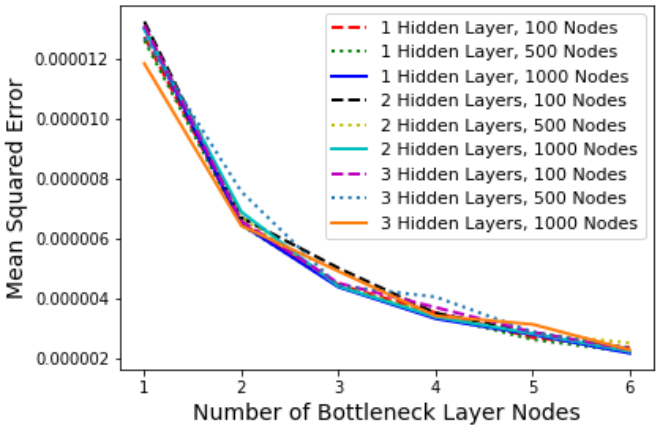


Figure 4.1: Autoencoder architecture optimization. The autoencoder mean-squared error (MSE) is plotted against the number of nodes in the bottleneck layer (i.e., the length of the low-dimensional representation vector) for various network sizes). "Elbows" in these plots occur around a bottleneck size of 3 nodes, indicating a low-dimensional size of 3 is likely sufficient to capture the essential information from the CNA signatures. At a bottleneck size of three nodes, the autoencoder with two hidden layers and 1000 hidden nodes shows the lowest MSE. This model is used throughout the main text. Note that each autoencoder used tanh activation functions in the hidden layers.

4.4 Colloidal Self-Assembly Dynamical Model

Deep neural networks (DNNs) have shown great promise for modeling the complex dynamics of systems with multiple states [166]. Unlike the physics-based modeling approaches in [37, 38, 59], DNNs can tractably model systems with 3 or more states. As a result, this chapter models the system dynamics with a DNN. The time evolution of the order parameters derived from teh autoencoder in the previos section, $y(t+1)$, is a function of the previous autoencoder-based order parameter values, $z(t)$, the previous temperature value, $T(t)$, and the input temperature ramp rate, $u(t)$. Thus the model takes the form:

$$z(t+1) = f(z(t), T(t)u(t)). \tag{4.1}$$

Note that this is a discrete-time dynamic model and the sampling time for this model is 500 time steps, where a time-step is defined as the dimensionless characteristic time of the system (i.e., $t^*/t = \sqrt{(\epsilon/m\sigma^2)}$, [164]).

The model was trained, validated, and tested using the exact same data set that was used to train the autoencoder in the previous section. A hyper-parameter optimization based on the elbow method was implemented. The final DNN contained 2 hidden layers and 100 hidden nodes and tanh activation functions on the hidden layers. The DNN was implemented using the Python library Keras (a TensorFlow API) [116, 117].

The mean validation mean-squared error (MSE) for a 1-step prediction horizon is 0.00020. The test MSE for the same 1-step predictions is 0.00021. The similar performance between the mean validation and test errors suggest that the DNN is neither underfit nor overfit [166]. Fig. 4.2 compares an example test trajectory of the MD simulations under varied temperature ramp rates to 1-step predictions of the DNN. The figure shows excellent agreement between the example realization and the 1-step predictions. Note, however, that the DNN does not account for system stochasticity. As a result, the prediction MSE increases rapidly (in fact, nearly logarithmically) with the number of prediction steps.

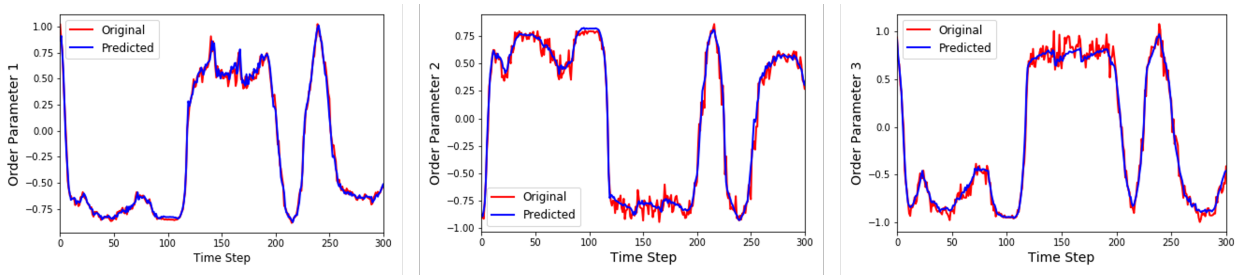


Figure 4.2: Example MD trajectory with DNN predictions. One-step predictions from the DNN (colored in blue) are compared to "true" state trajectories from MD simulations (colored in red). Every unit on the x-axis is one sampling time. The DNN predicts the colloidal SA dynamics with high accuracy.

4.5 Model Predictive Control Formulation

The goal of this chapter is to develop a model-based control framework that uses temperature to guide reliably the colloidal SA of the test system past kinetically arrested states and towards a desired, lowest-energy state at the free energy global minimum (e.g., a "defect-free" 2D hexagonal crystal). Although at least 193 unique CNA signatures can describe various colloidal SA states for the test system, the target, "defect-free" 2D hexagonal crystal is fully described by 4 CNA signatures. In this target state, between 82-84% of particles have one CNA signature, which will be labeled the "crystalline CNA" signature going forward. The remaining three signatures describe particles on the edge of the lattice, as particles on the edge of a perfect lattice will have a different number of neighbors than those in the bulk. Note that these CNA fraction values were verified to indicate a perfect hexagonal crystal from a visual inspection of the lattice geometry.

Interestingly, outside of the target state, larger fractions of these three remaining CNA signatures are commonly observed within defective and kinetically arrested states. For ex-

ample, several clearly visually defective states can be seen with crystalline CNA fractions of 0.6-0.75 and the sum of the 3 CNA fractions total 0.35 to 0.20. These fractions are thus labeled "defective" CNA fractions moving forward. The specific goal of the model-based feedback control strategy is thus to guide the SA test system to a target state of crystalline CNA fraction 0.82 to 0.84 and limit the sum of the defective CNA fractions to below 0.18. The idea of limiting the appearance of certain CNA signature fractions can be viewed as incorporating a "state constraint" within the control strategy. By incorporating state constraints, the control strategy can explicitly seek to avoid colloidal SA states that contain defective CNA fractions above 0.18 while actively seeking to achieve states that contain a crystalline CNA fraction between 0.82 to 0.84.

An SA system may contain competing "control objectives" with respect to product quality and manufacturing time. The ultimate goal of the SA of a colloidal crystallization system is to achieve defect-free crystals, which may involve guiding the colloidal SA system out of essentially unavoidable kinetically arrested states. For example, it may take many more assembly/disassembly cycles (and thus much more manufacturing time) to reach the global free energy minimum at a crystalline CNA fraction of 0.84 than it may take to reach a local free energy minimum at the marginally less crystalline state at a crystalline CNA fraction of 0.76. Control objectives in SA systems may also include economic factors. For example, implementing a series of very fast temperature ramps may guide the system to its global free energy minimum more quickly, but the cost of implementing these quick ramps may make the strategy economically impractical.

The previous paragraphs indicate that an effective model-based control strategy for colloidal SA must be able to handle nonlinear dynamics, state constraints, and competing control objectives. Model predictive control (MPC) methods have previously demonstrated the ability to account effectively for these considerations in other complex systems [167, 168]. MPC involves solving the following optimal control problem (OCP) to determine a future sequence of temperature changes $u(t)$ that guide the system towards the order parameter values that correspond to the target state. The OCP that is solved at each sampling time k is formulated as:

$$
\begin{aligned}
\min_{u} \quad & \sum_{i=0}^{N-1} \|(y(i) - y_{\text{target}})\|_Q^2 + \|(u(i) - u_{\text{target}})\|_R^2 \\
\text{s.t.} \quad & z(k+i+1|k) = f\Big(z(k+i|k)), T(k+i|k), u(k+i|k)\Big) \\
& T(k+i+1|k) = T(k+i|k) + u(k+i|k)\Delta t \\
& Du(k+i|k) \leq d \\
& Fy(k+i|k) \leq f \\
& GT(k+i|k) \leq g \\
& y(k|k) = g(z(k)) \\
& T(k|k) = T(k) \\
& i = 0, ..., N-1.
\end{aligned}
$$

The decision variables are $\mathbf{u} := u(k|k, u(k+1|k, ..., u(k+N-1|k)$. The cost function is designed to follow a set-point for only the crystalline order parameter (i.e., $y_{\text{target}} = 0.84$). The target input value (i.e., temperature ramp rate) is set to $u_{\text{target}} = 0$ temperature units per time in order to penalize the controller for making rapid control moves that can lead to unstable behavior. Q, R, D, F, and G are weight matrices, while d, f, and g represent the upper limits of the state and input constraints. The temperature state constraint and input constraint reflect the physical limitations of the system. The state constraint on the defective CNA fractions is set to prevent the system from reaching highly defective states during assembly. As a result, the constraint attempts to ensure that system exists in a state where $y_{\text{defect}} \leq 0.16$, which is the maximum defect order parameter value in the "most" perfect crystal. The crystalline CNA fraction is not constrained. The dynamic model $f(\cdot)$ takes order parameters derived from the autoencoder, $z(k+i|k)$, the temperature ($T(k+i|k)$, and the temperature ramp rate, $u(k+i|k)$ as input and outputs order parameter values at the next time step $z(k+i+1|k)$. The function g then converts these order parameters to the crystalline and defective CNA fractions that are explicitly placed in the OCP. Using a low-dimensional dynamic model (instead of a high-dimensional dynamic model in terms of all 193 unique CNA signatures) allows the OCP to be solved on the order of seconds and thus allows the proposed control strategy to be computationally tractable.

The OCP is implemented in receding horizon mode in order to provide feedback, where only the first element of the optimal control sequence $u^*(0)$ is applied to the SA system at each sampling instant k. The optimal control sequence is then re-calculated at the next sampling time. The OCP was solved using an IPOPT solver [169] from the Casadi Python library. The state constraint on the defect order parameter was formulated as a "soft" constraint [170].

4.6 Closed-Loop Implementation

The MPC strategy was first tuned by selecting the prediction horizon and the ratio of the cost function weights (i.e., Q/R). A prediction horizon of $N = 10$ was chosen. The ability of the deterministic data-driven model (Eq. (4.1)) to predict accurately the inherently stochastic SA system dynamics degrades with prediction horizon. Although shorter horizons lead to more accurate system dynamics predictions, these horizons are not long enough to predict multiple assembly/disassembly cycles in the future. Ratios smaller than $Q/R = 10$ prevented the controller from choosing input ramp rates that were large enough to break the system out of kinetic traps, while ratios larger than $Q/R = 10$ encouraged the controller to choose overly aggressive input profiles that sometimes led to unstable behavior.

The MPC strategy was applied to the test system under 50 different realizations of MD simulations of the system. The mean and range of variation of the closed-loop simulations are shown in Fig. 4.3, which shows that the system consistently reaches a final state value that corresponds to the system's global free energy minimum (and thus a "defect-free" two-dimensional hexagonal lattice). However, the large range of variation bars in Fig. 4.3 indicate

that the time to reach that target value varies significantly from run to run. Fig. Fig. 4.4 depicts three example closed-loop realizations with vastly different times to the final state. Fig. 4.4 shows that generally the controller chooses a fast cooling rate to drive quickly the system towards its target state, followed by a large heating rate to pull the system out of a kinetic trap (indicated by a large defect order parameter value). Then, the controller implements a series of relatively fast cooling and heating rates to jostle the system out of various kinetic traps until the system reaches a pathway in which slow cooling can allow the target state to be reached.

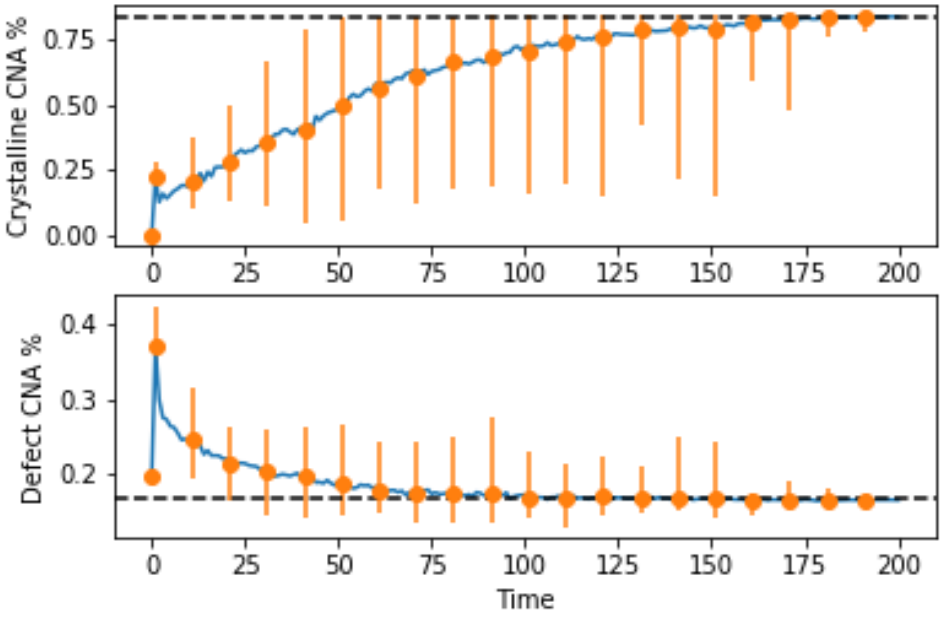


Figure 4.3: Closed-loop profiles of the defective and crystalline CNA fractions. The MPC strategy was applied to 50 different realizations of MD simulations of the test system. The mean order parameter time evolution is shown along with bars indicating the minimum and maximum observed state values (i.e., the range of observed CNA fractions). Every unit on the x-axis is one sampling time.

One reason for the variation in the time to reach the final state is that the system often violates the defect order parameter constraint (i.e., $y_{\text{defect}} > 0.16$) during assembly. This causes the controller to heat the system up to cause crystal disassembly, and thus decrease both the crystalline and defective CNA fractions. This heating is often followed by cooling, which only sometimes increases the crystalline CNA fraction without significantly increasing the defect CNA fraction. The variability in the effects of these heating/cooling sequences (due to intrinsic system stochasticity) leads to the variation in the time to reach the final state seen in Figs. 4.3-4.4.

Fig. 4.5 demonstrates the effects of the identity of the defective CNA constraint on the time it takes for the system to reach its final state (i.e., final CNA fractions) and the average final state values. Here, 50 closed-loop runs are implemented with defect constraint values

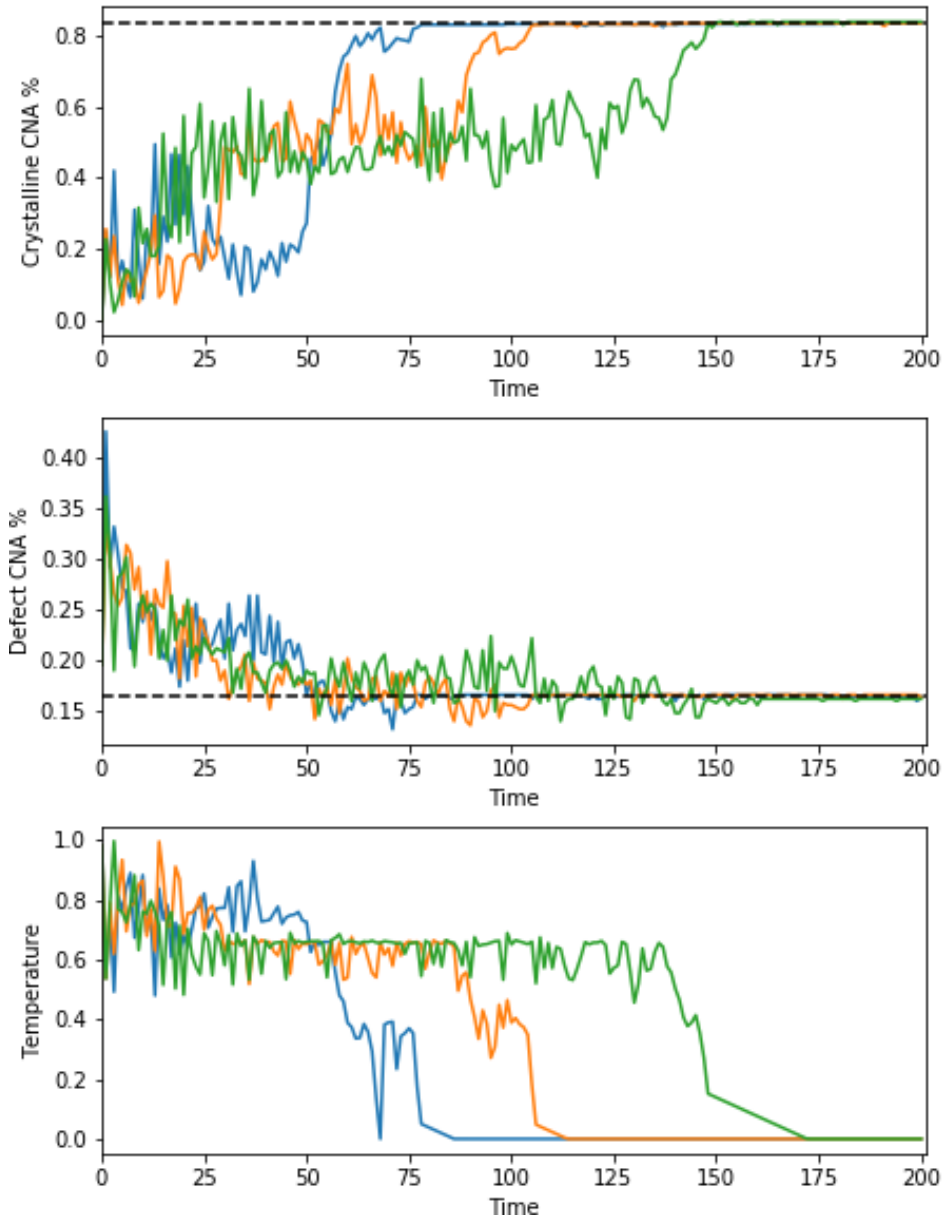


Figure 4.4: Three sample closed-loop runs of the MPC strategy on MD simulations. Every unit on the x-axis is one sampling time and the temperature is scaled from 0 to 1 for visibility purposes. Each color refers to a separate trajectory.

ranging from $y_{\text{defect}} \leq 0.16$ to 0.25. After $y_{\text{defect}} = 0.16$, higher defect CNA fractions allow the system to reach its final state more quickly. However, this final state is a kinetically-favored local free energy minimum, which is less crystalline than the $y_{\text{target}} = 0.82 - 0.84$ perfect crystal. This trade-off clearly demonstrates the previously mentioned competing control objectives of product quality and assembly time. Finally, note that the mean time to the final state scales with the variance in the time it takes the system to reach that final state. This information was not explicitly shown in Fig. 4.5 for brevity purposes.

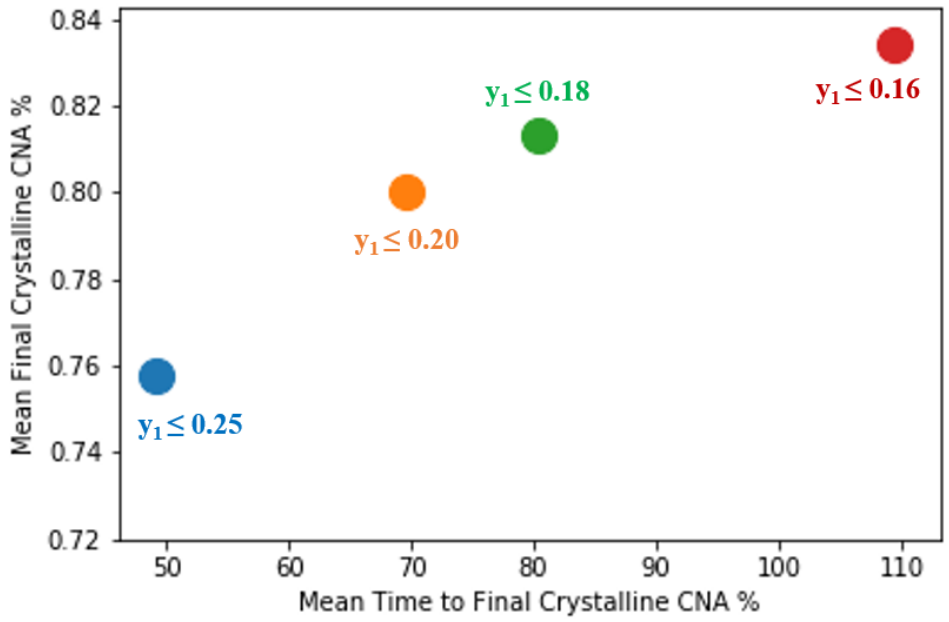


Figure 4.5: Comparison of the effect of the defect constraint on controller performance. The MPC strategy was applied to 50 different realizations of MD simulations of the test system for different values of the defect state constraint. The average time to the final state and average final state value are shown for each defect constraint. Each unit on the x-axis is 1 sampling time.

4.7 Conclusions and Next Steps

This chapter presents a model predictive control approach for controlling colloidal SA systems that involves quantifying the system state with order parameters discovered by an autoencoder and modeling system dynamics with a deep neural network. The model predictive control strategy was able to guide reliably a two-dimensional *in-silico* system of DNA-functionalized silica colloids towards its global free energy minimum (i.e., a perfect 2D hexagonal lattice). Despite consistently reaching the target state, the time to reach the target state was largely variable from run-to-run. This variance could likely be reduced by explicitly accounting for system stochasticity in the formulation of the optimal control problem [14].

The most obvious way to account explicitly for system stochasticity in the optimal control problem is to replace the deterministic, deep neural network-based model with a stochastic dynamical model. Moreover, in practice, the temperature may not be uniformly distributed over the system (i.e., large temperature gradients may exist). The existence of these gradients introduces another source of stochasticity into the system, which further justifies the need for a stochastic dynamical model. As a result, the next section investigates methods for learning low-dimensional stochastic dynamical models of multi-variate, intrinsically stochastic systems with nonlinear dynamics. Although the original motivation for the work in the next chapter was modeling colloidal SA dynamics, the framework in principle be applied to any stochastic dynamical system. As a result, the next chapter deviates from previous ones in its generalizability past colloidal SA systems.

Chapter 5

Stochastic Physics-Informed Neural Ordinary Differential Equations

Stochastic differential equations (SDEs) are used to describe a wide variety of complex stochastic dynamical systems. Learning the hidden physics within SDEs is crucial for unraveling fundamental understanding of these systems' stochastic and nonlinear behavior. This chapter proposes a flexible and scalable framework for training artificial neural networks to learn constitutive equations that represent hidden physics within SDEs. The proposed stochastic physics-informed neural ordinary differential equation framework (SPINODE) propagates stochasticity through the known structure of the SDE (i.e., the known physics) to yield a set of deterministic ODEs that describe the time evolution of statistical moments of the stochastic states. SPINODE then uses ODE solvers to predict moment trajectories. SPINODE learns neural network representations of the hidden physics by matching the predicted moments to those estimated from data. Recent advances in automatic differentiation and mini-batch gradient descent with adjoint sensitivity are leveraged to establish the unknown parameters of the neural networks. This chapter demonstrates SPINODE on three benchmark in-silico case studies and analyzes the framework's numerical robustness and stability. SPINODE provides a promising new direction for systematically unraveling the hidden physics of multivariate stochastic dynamical systems with multiplicative noise.

5.1 Introduction

Stochastic dynamical systems are ubiquitous in a wide range of science and engineering problems, such as dynamical systems governed by Brownian motion or those that experience random perturbations from their surrounding environment [171–175]. Stochastic differential equations (SDEs) are used to describe the complex behavior of a wide variety of stochastic dynamical systems, including those involving electrical and cell signal processing [176–178], colloidal/molecular self-assembly [37, 38], nucleation processes [179, 180], and predator-prey dynamics [181, 182]. An important challenge in constructing and studying SDEs is that

they often contain physics that are either unknown or cannot be directly measured (e.g., free energy and diffusion landscapes [31, 69], transmission functions in models of disease spread [183, 184], etc.). Creating a systematic framework to learn the hidden physics within SDEs is thus crucial for unraveling fundamental understanding of stochastic dynamical systems.

A fairly general representation of SDEs is given by:

$$dx = f(x, g(x))dt + h(x, g(x))dw, \tag{5.1}$$

where x is the system state that is generally vector-valued, t is the time, and w is generally a multivariable Gaussian white noise process. The "modeled" or "known" physics is comprised of $f(\cdot)$, $h(\cdot)$, and the structure of the SDE (i.e., the additive relationship between $f(\cdot)$ and $h(\cdot)$ and the multiplicative relationship between h and w). This chapter considers $g(x)$ to be the "unmodeled" or "unknown" hidden physics. This chapter thus seeks to investigate strategies to create a flexible and scalable framework for systematically learning the hidden physics $g(x)$ within SDEs of form Eq. (5.1) from stochastic trajectory data.

The most commonly reported methods for learning $g(x)$ from stochastic trajectory data involve evaluating the time limits of the first and second conditional moments [185–196]:

$$f(x, g(x)) = \lim_{\tau \to 0} \frac{1}{\tau} \Big\langle (\xi(t+\tau) - \xi(t)) | \xi(t) = x \Big\rangle, \tag{5.2a}$$

$$h(x, g(x)) = \lim_{\tau \to 0} \frac{1}{2\tau} \Big\langle (\xi(t+\tau) - \xi(t))^2 | \xi(t) = x \Big\rangle, \tag{5.2b}$$

where ξ denotes a realization of the stochastic process with a δ-function distribution at the starting point t, $\xi(t) = x$, τ is the sampling time, and the angular brackets denote ensemble averaging. In practice, $\tau \to 0$ must be extrapolated or τ must be chosen to be sufficiently small to represent the limit. As the lower bound of τ is often determined by experimental limitations, the primary challenge facing works [188–196] is how to determine a robust way to extrapolate $\tau \to 0$. Common approaches to address this challenge involve adding correction terms to Eq. (5.2b) [188], using autocorrelation functions to simplify Eq. (5.2b) [197], employing kernel-based regressions over τ [189], and iteratively updating the limit evaluations based on computed probability distributions [191]. However, such methods generally rely on inflexible, data-intensive, and system-specific sampling techniques and/or have been shown to be non-viable when short-time linear regions do not exist in the trajectory data [31, 69, 189].

Alternative approaches for learning $g(x)$ leverage Bayesian inference to estimate transition rates along adjacent intervals of x, e.g., [28, 31, 37, 69, 70, 198–200]. The hidden physics $g(x)$ can then be recovered by exploiting relationships derived from the Fokker-Planck equation [201]. Although Bayesian inference approaches have been shown to be less sensitive to the sampling time than those that depend on extrapolating $\tau \to 0$ [31, 69], these approaches either (i) learn $g(x)$ at discrete values of x and then fit analytic functions to these discrete values [28, 31, 37, 69, 198, 199], or (ii) represent the unknown $g(x)$ using basis functions and learn the coefficients of those basis functions [70, 200]. The former approach can become intractable when the dimension of x is large, or when $g(x)$ is highly nonlinear and thus requires

x to be finely discretized. The latter approach can be highly sensitive to the choice of basis functions and can exhibit other numerical issues. As such, this latter approach often requires *a priori* knowledge about the stochastic system to inform the choice of basis functions [70].

To address the shortcomings described above, this chapter proposes a new framework for learning the hidden physics $g(x)$ in Eq. (5.1), which this chapter refers to as stochastic physics-informed neural ordinary differential equations (SPINODE). SPINODE approximates $g(x)$ as an artificial neural network, where the weights and biases within the neural network represent the SDE hidden physics. Artificial neural networks provide a scalable and flexible way of approximating the potentially highly nonlinear relationship between $g(x)$ and continuous values of x without the need for *a priori* assumptions about the form of that relationship [202–204]. SPINODE then combines the notions of neural ordinary differential equations (neural ODEs) [205, 206] and physics-informed neural networks (PINN) [207–211] to learn the weights and biases within the neural network that approximates $g(x)$ from state trajectory data. If one had access to the true state distribution at particular time points (which is generally non-Gaussian due to the nonlinear terms appearing in Eq. (5.1)), one could attempt to identify the neural network parameters that minimize a distributional loss function (e.g., the sum of the Kullback–Leibler divergence between the true and predicted distribution). However, not only would this loss function be more complicated to evaluate, direct access to exact state distributions is often unavailable since these distributions must be estimated from a finite set of state trajectories collected from simulations or experiments. Therefore, this chapter opts for a more tractable *moment-matching* framework [212–214], which is an established method in statistics for simplifying the distribution-matching problem. There are two key advantages to the moment-matching approach in the context of partially known SDEs:

- The moment-matching approach only requires moments of the state to be measured at discrete time points (with potentially varying sample times) from some known initial state distribution, which are easier to estimate than the full probability distribution or conditional moments.
- The predicted moments of the state based on Eq. (5.1) can be estimated using established uncertainty propagation techniques. As long as the uncertainty propagation method is differentiable, concepts from the neural ODE framework can be used to compute derivatives needed for efficient training while preserving important features from the underlying SDE.

Although SPINODE can be adapted to handle a variety of different uncertainty propagation methods, this chapter mostly focuses on the unscented transform (UT) method [215–217] due to its ability to tradeoff gracefully between accuracy and computational efficiency. The UT method, when applied to Eq. (5.1), yields simple analytic expressions for the mean and covariance of the states in terms of the solution to a relatively small set of ODEs. By defining and evaluating the model in terms of ODE solvers, one immediately gains the well-known benefits of such solvers including: (i) memory efficiency, (ii) adaptive computation

with error control, and (iii) prediction at arbitrary sets of non-uniform time points [205]. All of these benefits are important when developing an efficient training algorithm for the neural network representation of $g(x)$ for which the loss gradient with respect to the neural network parameters can be computed using adjoint sensitivity methods [205, 206, 218].

To highlight the differences between SPINODE and previous methods, let us turn back to Eq. (5.2), which essentially computes the time derivative of the mean and covariance of x at some time t by some limit approximation. Previous works [185–196] have proposed many different strategies for interpolating measured state data from a finite set of discrete time points to estimate this limit; however, these strategies are largely system-specific. SPINODE, on the other hand, uses advanced uncertainty propagation and ODE solvers to predict directly state moment data at any set of time points. Since these underlying methods have been developed to apply to a diverse set of systems, including those that involve high-dimensional, nonlinear, and stiff dynamics, SPINODE can be flexibly applied to systems arising from all different types of applications, which this chapter demonstrates by applying SPINODE to the aforementioned case study systems. Further note that although SPINODE is only described in the context of the first two moments in this paper for simplicity, it can naturally incorporate any number of moments (e.g., skew and kurtosis) when learning $g(x)$. This suggests that SPINODE has the potential to better handle highly non-Gaussian state distributions, which may arise when $f(\cdot)$, $h(\cdot)$, or $g(\cdot)$ are highly nonlinear.

This chapter demonstrates the efficacy, flexibility, and scalability of SPINODE on three benchmark *in-silico* case studies. The dynamics of each system are described by SDEs of form Eq. (5.1) that contain nonlinear and state-dependent hidden physics terms. The first case study is a two-state model for directed colloidal self-assembly with an exogenous input [59], the second is a four-state competitive Lotka-Volterra model with a coexistence equilibrium [219], and the third is a six-state susceptible-infectious-recovered (SIR) epidemic model for disease spread [220]. This chapter shows that SPINODE is able to learn efficiently the hidden physics within these SDEs with high accuracy. This chapter analyzes the numerical robustness and stability of SPINODE and provide suggestions for future research. Finally, a fully open-source version of SPINODE has been released on GitHub with end-to-end examples [221], so that interested readers can easily reproduce and extend the results described in this chapter.

5.2 Method Overview

A schematic overview of the proposed SPINODE method is shown in Fig. 5.1. Repeated stochastic dynamical system trajectories are recorded to estimate the time evolution of statistical moments of the stochastic state, $m_x^{(i)}(t_k)$ for all $i = 1, \ldots, N_m$ where N_m denotes the total number of moments considered (left). The hidden physics $g(x;\theta)$ are represented by a (deep) neural network that is parameterized by unknown weights and biases denoted by θ (center). Established uncertainty propagation methods are used to propagate stochasticity through Eq. (5.1) and ODE solvers within the neural ODE framework are used to predict

the time evolution of the moments for fixed neural network parameters, $\hat{m}_x^{(i)}(t_k;\theta)$ (left to center). A loss function is constructed using the predicted and data-estimated moments (center). Mini-batch gradient descent with adjoint sensitivity is used to update the parameters θ by minimizing the loss function (right). The hidden physics, $g(x;\theta)$, are considered "learned" once the mini-batch gradient descent algorithm converges. The subsequent subsections describe in more detail how data is collected and how SPINODE uses uncertainty propagation, neural ODEs, moment-matching, and mini-batch gradient descent to learn the weights and biases within the neural networks that approximate the unknown hidden physics within SDEs.

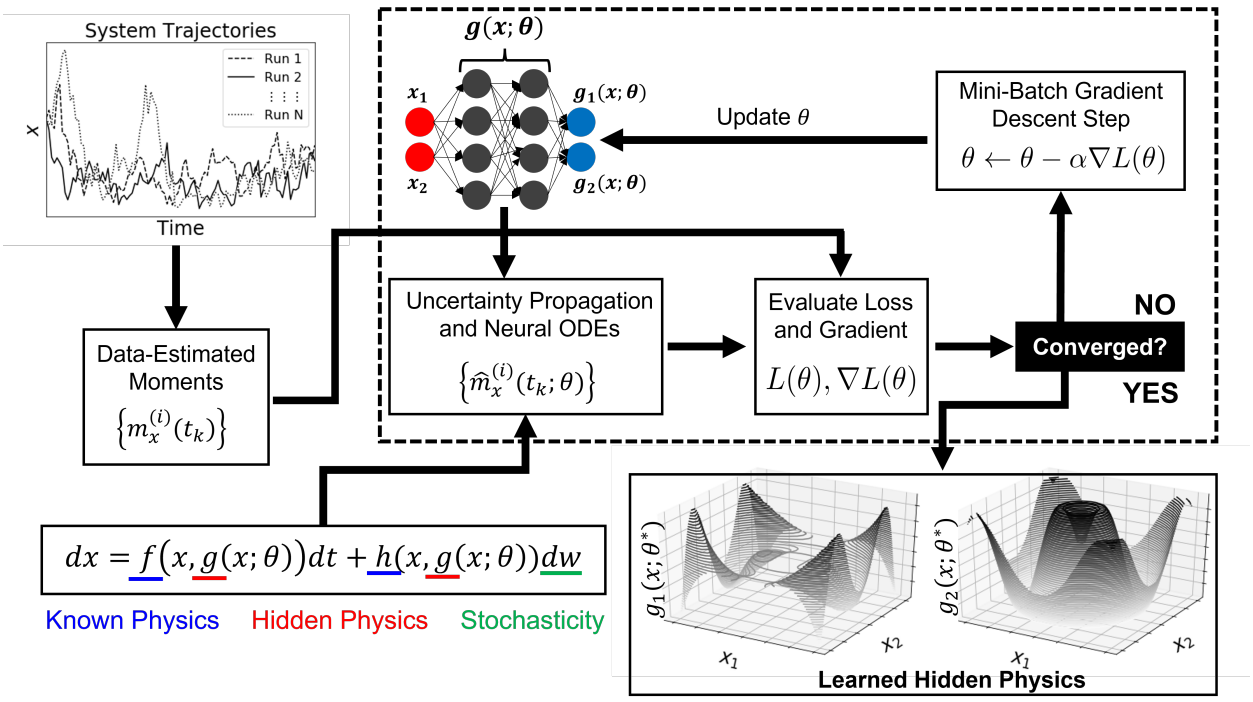


Figure 5.1: Stochastic physics-informed neural ordinary differential equations framework (SPINODE). The key steps include (i) estimating the time evolution of statistical moments $m_x^{(i)}(t_k)$ from repeated stochastic dynamical system trajectories, (ii) approximating the hidden physics as a neural network (e.g., $g(x;\theta) = [g_1(x;\theta), g_2(x;\theta)]^\top$, where the unknown weights and biases are θ), (iii) using uncertainty propagation to propagate stochasticity through the known structure of the stochastic differential equation, (iv) using ODE solvers within the neural ODE framework to predict the time evolution of statistical moments $\hat{m}_x^{(i)}(t_k)$, and (v) using moment-matching and mini-batch gradient descent with adjoint sensitivity to learn "optimal" weights and biases θ^*.

5.3 Data Collection

Data collection is accomplished by repeating stochastic dynamical system trajectories starting from identical initial conditions. Here, N trajectories start from some initial condition x_0. During each trajectory, state values x are recorded at time points t_k for $K+1$ total time steps. The N recorded values of each $x(t_k)$ are used to estimate moments $m_x^{(i)}(t_k)$. For simplicity, this chapter primarily focuses on the first two moments, the state mean and covariance, which are calculated as follows:

$$\begin{aligned}\mu_x(t_k) &= \frac{1}{N}\sum_{n=1}^{N} x_n(t_k),\\ \Sigma_x(t_k) &= \frac{1}{N}\sum_{n=1}^{N}(x_n(t_k)-\mu_x(t_k))(x_n(t_k)-\mu_x(t_k))^\top,\end{aligned} \tag{5.3}$$

where $n \in \{1,\ldots,N\}$ denotes the trajectory index.

Repeated stochastic trajectories from only one initial condition may not explore a large percentage of the state space. To compensate for this, the stochastic trajectories can be collected from multiple unique initial conditions. This chapter chooses initial conditions by performing a grid search within a specified range of state values of interest. Note, however, that more efficient sampling techniques, e.g., [222–226] can also be used, which will be explored in future work.

Although this chapter estimates moments of the stochastic states from repeated stochastic trajectories from identical initial conditions, it is clear that this strategy is not applicable for systems in which one does not have control over initial conditions, number of replica runs, or consistent measurement times. In such cases, probability distributions of state trajectories can be learned using methods that may not necessarily require such fine control over the observed trajectory data. Potentially suitable distribution estimation methods include variational autoencoders [227–229], generative adversarial networks [209, 230–232], and/or energy-based models [233–235]. SPINODE is able to accommodate any data collection method from which the shape of the probability distribution (and thus the moments) can be estimated at discrete time points from observed trajectory data.

5.4 Moment Prediction

As motivated in the introduction, an important advantage of the moment-matching framework is that the framework can use efficient uncertainty propagation methods that do not require access to the full distribution of the states. Unscented transform (UT) [215, 216, 236–238] is one such example of an efficient uncertainty propagation method that estimates moments from a set of well-placed samples (known as *sigma points*) that can be efficiently evaluated using a neural ODE solver.

Before applying UT to SDEs, let us first summarize the UT method for estimating the moments of a random variable $y = F(z)$ that is some static nonlinear transformation $F : \mathbb{R}^n \to \mathbb{R}^m$ of a random input $z \in \mathbb{R}^n$. This chapter assumes knowledge of the mean $m \in \mathbb{R}^n$ and covariance $P \in \mathbb{R}^{n \times n}$ of z. Given this information, UT involves the following 3 steps:

1. Form the set of $2n + 1$ sigma points from the columns of the matrix $A = \sqrt{(n + \lambda P)}$, which denotes the Cholesky decomposition, as follows

$$\begin{aligned} z^{(0)} &= m, && \\ z^{(i)} &= m + [A]_i, && i = 1, \ldots, n, \\ z^{(i)} &= m - [A]_{n-i}, && i = n+1, \ldots, 2n, \end{aligned} \tag{5.4}$$

   where $[A]_i$ denotes the i^{th} column of the matrix A. Then, compute the associated weights of each of these sigma points

$$\begin{aligned} W_0^{(m)} &= \frac{\lambda}{n+\lambda}, && \\ W_0^{(c)} &= \frac{\lambda}{(n+\lambda) - (1 - \alpha^2 + \beta)}, && \\ W_i^{(m)} &= \frac{1}{2(n+\lambda)}, && i = 1, \ldots, 2n, \\ W_i^{(c)} &= \frac{1}{2(n+\lambda)}, && i = 1, \ldots, 2n, \end{aligned} \tag{5.5}$$

   where λ is a scaling factor defined by

$$\lambda = \alpha^2(n + \kappa) - n, \tag{5.6}$$

   and α, β, and κ are positive constants. Typically, one should set α to be small (e.g., 10^{-3}), $\beta = 2$, and $\kappa = 0$ based on observations from [216].

2. Transform each of the sigma points as follows

$$y^{(i)} = F(z^{(i)}), \quad i = 0, \ldots, 2n. \tag{5.7}$$

3. Compute estimates for the mean and covariance of y

$$\begin{aligned} \hat{\mu}_y &= \sum_{i=0}^{2n} W_i^{(m)} y^{(i)}, \\ \hat{\Sigma}_y &= \sum_{i=0}^{2n} W_i^{(c)} \left(y^{(i)} - \hat{\mu}_y\right) \left(y^{(i)} - \hat{\mu}_y\right)^\top, \end{aligned} \tag{5.8}$$

As shown in X, UT can be compactly represented in matrix form as follows

$$\begin{aligned} Z &= \begin{bmatrix} m & \cdots & m \end{bmatrix} + \sqrt{\alpha^2(n+\kappa)} \begin{bmatrix} 0 & \sqrt{P} & -\sqrt{P} \end{bmatrix}, \\ Y &= F(Z), \\ \hat{\mu}_y &= Y w_m, \\ \hat{\Sigma}_y &= Y W Y^\top, \end{aligned} \tag{5.9}$$

where Z denotes the matrix of sigma points and $w_m \in \mathbb{R}^{2n+1}$ and $W \in \mathbb{R}^{2n+1 \times 2n+1}$ are a vector and matrix defined in terms of the mean and covariance weight factors

$$\begin{aligned} w_m &= [W_0^{(m)}, \ldots, W_{2n}^{(m)}]^\top, \\ W &= \left(I - \begin{bmatrix} w_m & \cdots & w_m \end{bmatrix}\right) \mathrm{diag}(W_0^{(c)}, \ldots, W_{2n}^{(c)}) \left(I - \begin{bmatrix} w_m & \cdots & w_m \end{bmatrix}\right)^\top \end{aligned} \tag{5.10}$$

and I denotes the identity matrix of appropriate size. This representation will be helpful when applying UT to SDEs of the form Eq. (5.1). Since both x and w are random quantities, it is more convenient to write the SDE in the following form

$$\frac{dz(t)}{dt} = F(z(t); \theta) + De(t), \tag{5.11}$$

where $e(t)$ is a zero-mean white noise process with covariance $Q_c(t)$ and D is a dispersion matrix. Eq. (5.1) can be expressed in this form by defining an augmented state $z(t) = [x(t), w(t)]^\top$ and defining $F(\cdot)$ and D as follows

$$F(z(t); \theta) = \begin{bmatrix} f(x(t), g(x(t); \theta)) + h(x(t), g(x(t); \theta))w(t) \\ 0 \end{bmatrix}, \; D = \begin{bmatrix} 0 & 0 \\ 0 & I \end{bmatrix}.$$

As shown in [217] (Algorithm 4.4), the predicted mean and covariance for any time $t \geq t_k$ can be computed from the initial mean $m(t_k) = [\mu_x(t_k), 0]^\top$ and covariance $P(t_k) = \mathrm{diag}(\Sigma_x(t_k), I)$ (estimated from data as discussed in the previous section) by integrating the following differential equations

$$\begin{aligned} \frac{dm(t)}{dt} &= F(Z(t); \theta) w_m, \\ \frac{dP(t)}{dt} &= Z(t) F^\top(Z(t); \theta) + F(Z(t); \theta) W Z^\top(t) + D Q_c(t) D^\top, \end{aligned} \tag{5.12}$$

where the sigma points $Z(t)$ are defined similarly to that in Eq. (5.9), with $m(t)$ and $P(t)$ now being functions of time. The original state mean and covariance can then be recovered by a simple transformation of the augmented state

$$\hat{\mu}_x(t|t_k; \theta) = \begin{bmatrix} I & 0 \end{bmatrix} m(t; \theta), \quad \hat{\Sigma}_x(t|t_k; \theta) = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} P(t; \theta), \tag{5.13}$$

where the notation $t|t_k$ denotes predicted quantities given initial information at time t_k.

This chapter uses ODE solvers within the neural ODE framework [205, 206] to integrate Eq. (5.12) since $F(\cdot)$ is defined in terms of an embedded neural network used to represent the unknown/hidden physics $g(\cdot)$. The flexible choice of ODE solver provides SPINODE with the ability to handle accurately systems with high-dimensional, stiff, and/or nonlinear dynamics. Another advantage of explicitly integrating the SDE (as opposed to applying a fixed time step discretization) is that the framework can handle potentially sparse, non-uniform time grids $\{t_0, t_1, \ldots, t_K\}$. Although the work in this chapter only exploits information provided by the first two moments of the state distribution, UT can also straightforwardly incorporate higher-order moment data, as described in [237, 238]. As shown in Section 5.10, incorporating higher-order moments into the prediction scheme can lead to improved performance when learning $g(\cdot)$ due to better placement of the sigma points.

5.5 Moment Matching

Since this chapter represents the hidden physics $g(x;\theta)$ with a neural network, a proper loss function needs to be defined in order to estimate θ. In other words, given a loss function $L(\theta)$, the goal of "learning the hidden physics" can be translated into solving the following optimization problem:

$$\theta^\star = \text{argmin}_\theta \; L(\theta). \tag{5.14}$$

A natural loss function for the moment-matching problem is the reconstruction error of the moments, which can be defined as follows

$$L(\theta) = \sum_{k=1}^{K} \sum_{i=1}^{N_m} \|m_x^{(i)}(t_k) - \hat{m}_x^{(i)}(t_k|t_{k-1};\theta)\|^2, \tag{5.15}$$

which simplifies to the following expression when only the first two moments are considered

$$L(\theta) = \sum_{k=1}^{K} \|\mu_x(t_k) - \hat{\mu}_x(t_k|t_{k-1};\theta)\|^2 + \|\Sigma_x(t_k) - \hat{\Sigma}_x(t_k|t_{k-1};\theta)\|^2, \tag{5.16}$$

where $\|\cdot\|^2$ denotes the sum of squared values of all elements in the vector/matrix. Eq. (5.14) is then solved via mini-batch gradient descent, which estimates the gradient of the loss function as follows

$$\nabla_\theta L(\theta) \approx \frac{1}{B} \sum_{k \in \mathcal{B}} \nabla_\theta L_k(\theta), \tag{5.17}$$

where $L_k(\theta) = \|\mu_x(t_k) - \hat{\mu}_x(t_k|t_{k-1};\theta)\|^2 + \|\Sigma_x(t_k) - \hat{\Sigma}_x(t_k|t_{k-1};\theta)\|^2$ is the error in the k^{th} data point, B is the number of "mini-batch" samples, and $\mathcal{B} \subset \{1, \ldots, K\}$ is a set of B

randomly drawn indices. The gradient estimate in Eq. (5.17) can be efficiently evaluated using the adjoint sensitivity method described in [205, 218]. Therefore, SPINODE can be easily implemented using open-source deep learning software such as PyTorch [239] – an implementation for the case studies considered in this chapter has been provided on GitHub [221].

5.6 Simplified Training Procedure with Approximate Unscented Transform

Based on the UT-based ODEs in Eq. (5.12) and the structure of $F(\cdot)$, the evaluation of the mean and covariance are fully coupled, that is, $m(t)$ and $P(t)$ must be simultaneously integrated to evaluate the loss function and its gradient. Since this procedure can be computationally expensive, it is useful to derive alternative approximations that can lead to a simplified training procedure. A particularly important special case of Eq. (5.1) is when the hidden physics $g(\cdot)$ is fully separable, i.e.,

$$dx = f(x, g_1(x;\theta_1)) + h(x, g_2(x;\theta_2))dw, \tag{5.18}$$

where $g_1(x;\theta_1)$ and $g_2(x;\theta_2)$ denote two completely independent neural networks (each with their own set of local parameters). According to Eq. (5.12), $\theta = \{\theta_1, \theta_2\}$ must still be trained simultaneously since the sigma points depend on both the mean and covariance.

To simplify the training process, this chapter presents an approximate UT that formulates independent ODEs that describe the time evolution of the transformed sigma points $Y = F(Z)$, where $Z = [Z^x, Z^w]^\top$:

$$\frac{dY}{dt} = f(Z^x(t), g_1(Z^x(t);\theta)) + h(Z^x(t), g(Z^x(t);\theta))Z^w(t). \tag{5.19}$$

The predictions of Y combined with Eq. (5.9) can be used to predict the mean and covariance. More importantly, since $Z^w(t)$ has a mean of zero and appears in an additive fashion, as long as the weights are chosen in a symmetric fashion, the $h(Z^x(t), g(Z^x(t);\theta))Z^w(t)$ term will cancel when evaluating the mean of the state. Therefore, in this case, the predicted state mean only depends on θ_1, i.e., $\hat{\mu}_x(t|t_k;\theta_1)$. By assuming that the predicted state covariance depends weakly on θ_1, θ_1 and θ_2 can be separately trained. In particular, the following two smaller optimization problems are solved sequentially:

$$\theta_1^\star = \text{argmin}_{\theta_1} \sum_{k=0}^{K} \|\mu_x(t_k) - \hat{\mu}_x(t_k|t_{k-1};\theta_1)\|^2, \tag{5.20}$$

$$\theta_2^\star = \text{argmin}_{\theta_2} \sum_{k=0}^{K} \|\Sigma_x(t_k) - \hat{\Sigma}_x(t_k|t_{k-1};\theta_1^\star,\theta_2)\|^2.$$

Note that the second optimization problem above is solved using a fixed functional form for the drift term $g_1(x;\theta_1^\star)$. Although heuristic in nature, this decomposed training strategy greatly reduces the number of parameters that need to be simultaneously considered when evaluating the loss function gradients. Not only does this significantly reduce computational cost, it also limits the search space such that it is less likely to find solutions that result in overfitting.

5.7 Validation Criteria using Predicted State Distribution

It is important to note that there can be many values for parameters θ that result in small or even zero loss function values since moments only provide limited information about the underlying distributions. In other words, even though two different sets of neural network parameters produce the same loss function value, they may result in substantially different predicted state distributions. A validation test can be developed to determine whether or not a given set of optimal parameter values $\theta^\star$ results in accurate state distributions. In particular, the sum of the Kullback–Leibler (KL) divergence [240] between the measured state distributions p_{x_k} and predicted $\hat{p}_{x_k}(\theta^\star)$ state distributions from a given initial condition over time can be evaluated, i.e.,

$$\text{Validation Error} = \sum_{k=0}^{K_V} \int_{\mathcal{X}} p_{x_k}(x) \log\left(\frac{p_{x_k}(x)}{\hat{p}_{x_k}(x;\theta^\star)}\right) dx, \tag{5.21}$$

where K_V denotes the number of validation time steps. Note that one can easily modify this definition to include multiple initial conditions and other controlled input values. Since neither of these distributions can be evaluated exactly, it is reasonable to employ established sample-based probability density function estimation techniques such as kernel density estimation [241]. This chapter recommends using this validation error criteria to decide if the hidden physics has been learned accurately enough to make reasonable predictions. Whenever the validation error is large, there may be a need to either modify the training strategy, increase the number of moments considered in the loss function, or collect additional data. Due to its simplicity, it is useful to start with the training procedure described in Section 5.6 and, if it does not pass the validation error test described in this section, apply the more detailed coupled training strategy.

5.8 Case Study System Descriptions

This chapter demonstrates SPINODE on three benchmark *in-silico* case studies from the literature: (i) a two-state model for directed colloidal self-assembly with an exogenous input [59], (ii) a four-state competitive Lotka-Volterra model with a coexistence equilibrium [219],

and (iii) a six-state SIR epidemic model for disease spread [220]. Each of these stochastic dynamical systems can be modeled by Eq. (5.1), and, since the hidden physics is fully separable in each case, Eq. (5.18).

State trajectory data is collected by discretizing Eq. (5.1) according to an Euler-Maruyama discretization scheme [242, 243]. These discretized SDEs are meant to represent the "real" system dynamics. Data-estimated moments $m_x^{(i)}(t_k)$ are then collected according to the approach described in Section 5.3 and the SPINODE framework outlined in Sections 2.2–2.4 is used to learn (or reconstruct) the hidden physics $g(x)$ from the collected stochastic trajectory data. SPINODE's performance is evaluated by assessing the accuracy of the reconstructed hidden physics. In each case study, moments $m_x^{(i)}(t_k)$ are calculated from 10^5 replicates of 50 time-step state trajectories from 2000 unique initial conditions (which leads to 10^5 total moments $m_x^{(i)}(t_k)$). As mentioned in Section 5.3, the number of initial conditions could very likely be decreased by employing more advanced sampling strategies, but exploring such strategies is beyond the scope of this work. Section 5.10 examines the relationship between the total number of data points and trajectory replicates and the hidden physics reconstruction accuracy.

Case Study 1: Directed Colloidal Self-Assembly with an Exogenous Input

The first case study is a two-state model for directed colloidal self-assembly with an exogenous input [59]. Here, the voltage of an external electric field is adjusted to mediate the two-dimensional self-assembly of silica micro-particles. The system dynamics are modeled according to Eq. (5.1). Denote x as an order parameter that represents crystal structure (i.e., the system state), u as the electric field voltage (i.e., the exogenous input), K_b as Boltzmann's constant, and T as the temperature:

$$\begin{aligned} dx &= g_1(x,u)dt + \sqrt{2g_2(x,u)}dw, \\ g_1(x,u) &= \frac{d}{dx}\Big(g_2(x,u)\Big) - \frac{d}{dx}\Big(F(x,u)\Big)\frac{g_2(x,u)}{K_bT}, \\ g_2(x,u) &= 4.5\times10^{-3}e^{-(x-2.1-0.75u)^2} + 0.5\times10^{-3}, \\ F(x,u) &= 10K_bT(x-2.1-0.75u)^2. \end{aligned} \tag{5.22}$$

The hidden physics is comprised of the drift coefficient, $g_1(x,u)$, and the diffusion coefficient, $g_2(x,u)$. Note that $g_1(x,u)$ is a function of a $g_2(x,u)$ and the free energy landscape $F(x,u)$. This relationship provides an example of how drift and diffusion coefficients can be used to derive other hidden system physics.

This case study was chosen because the hidden physics terms are highly nonlinear and depend on an exogenous input. To our knowledge, no previously reported approach for learning SDE hidden physics has explicitly learned $g(x,u)$. Instead, existing approaches typically seek to learn $g(x)$ at discrete values of u and interpolate [28, 31, 37, 38, 59]. This

requires repeating the entire hidden physics learning procedure for many discrete values of u and, thus, demands a trade-off between computational cost and accuracy. SPINODE, on the other hand, directly learns $g(x, u)$ over the entire (x, u) state space.

Case Study 2: Competitive Lotka-Volterra with a Coexistence Equilibrium

The second case study is a four-state competitive Lotka-Volterra model with a coexistence equilibrium [219]. The stochastic dynamics are modeled according to Eq. (5.1). Note that $x = [x_1, x_2]^\top$ and x_i^{eq} are the coexistence equilibrium points:

$$
\begin{aligned}
dx_1 &= g_1(x)_1 dt + \sqrt{2g_2(x)_1} dw_1, \\
dx_2 &= g_1(x)_2 dt + \sqrt{2g_2(x)_2} dw_2, \\
g_1(x)_1 &= x_1(1 - x_1 - k_1 x_2), \\
g_1(x)_2 &= x_2(1 - x_2 - k_2 x_1), \\
g_2(x)_1 &= x_1(x_2 - x_2^{\text{eq}}), \\
g_2(x)_2 &= x_2(x_1 - x_1^{\text{eq}}), \\
x_1^{\text{eq}} &= \frac{1 - k_1}{1 - k_1 k_2}, \quad x_2^{\text{eq}} = \frac{1 - k_2}{1 - k_1 k_2}, \\
k_1 &= 0.4, \quad k_2 = 0.5.
\end{aligned}
\tag{5.23}
$$

The hidden physics is comprised of the two-dimensional drift and diffusion coefficients, $g_1(x_1, x_2)$ and $g_2(x_1, x_2)$. As a result, this chapter seeks to train two multi-input, multi-output neural networks that approximate the hidden physics. The drift coefficient neural network takes x_1 and x_2 as input and outputs $g_1(x_1, x_2)_1$ and $g_1(x_1, x_2)_2$. The diffusion coefficient neural network takes x_1 and x_2 as input and outputs $g_2(x_1, x_2)_1$ and $g_2(x_1, x_2)_2$.

This case study was chosen because both the drift and diffusion coefficients are multi-dimensional and nonlinear. SPINODE is applied to this "more complex" SDE to demonstrate the framework's scalability. Note that no aspect of the framework was altered from its implementation for the previous case study.

Case Study 3: Susceptible-Infectious-Recovered Epidemic Model

The third case study is a six-state SIR epidemic model for disease spread [220]. The stochastic dynamics are modeled according to Eq. (5.1):

$$\begin{aligned}
&dS = (b - dS - g(S,I) + \gamma R)dt + \sigma_1 S dw_1,\\
&dI = (g(S,I) - (d+\mu+\delta)I)dt + \sigma_2 I dw_2,\\
&dR = (\mu I - (d+\gamma)R)dt + \sigma_3 R dw_3,\\
&g(S,I) = \frac{kS^h I}{S^h + \alpha I^h},\\
&b = 1,\ d = 0.1,\ k = 0.2,\ \alpha = 0.5,\ \gamma = 0.01,\ \mu = 0.05,\\
&\delta = 0.01,\ h = 2,\ \sigma_1 = 0.2,\ \sigma_2 = 0.2,\ \sigma_3 = 0.1.
\end{aligned} \tag{5.24}$$

The hidden physics is the infection transmission rate, $g(S,I)$, which plays a key role in determining disease spread dynamics in many epidemic models [183, 184, 220, 226, 244–247]. The form of $g(S,I)$ is widely considered to be unknown, and each of the above-listed references propose different versions of this function. SPINODE is applied to Eq. (5.24) to learn $g(S,I)$. This case study was chosen to demonstrate that SPINODE can not only broadly learn drift and diffusion coefficients but also can learn specific unknown physics terms within complex SDEs.

The hidden physics $g(S,I)$ primarily contribute to the deterministic dynamics (i.e., $f(x, g(x)$ in Eq. (5.1)) and appears in the time evolution equations for both S and I in Eq. (5.24). The resulting loss function used to train $g(S,I;\theta)$ is then given by:

$$\min_{\theta} \sum_{k=0}^{K} \left\|\hat{\mu}_S(t_k) - \mu_S(t_k))\right\|^2 + \left\|\hat{\mu}_I(t_k) - \mu_I(t_k))\right\|^2, \tag{5.25}$$

while the loss functions used to train $g_1(x,u)$ and $g_1(x_1,x_2)$ in the previous two case studies were given by Eq. (5.20).

5.9 Learning Hidden Physics

This chapter demonstrates SPINODE on the case studies outlined in Section 5.8. In each case study, moments $m_x^{(i)}(t_k)$ (e.g., means and covariances) are estimated from stochastic trajectory data. The approximate UT method described in Section 5.6 is then used to yield deterministic ODEs that describe the time evolution of the sigma points. An Euler ODE scheme is used to solve these ODEs and thus predict the time evolution of the means and covariances. Mini-batch gradient descent with adjoint sensitivity is then used to (i) match the predicted means and covariances to the data-estimated means and covariances and (ii) train the neural networks $g(x;\theta)$ that approximate the true hidden physics $g(x)$. Note that the time intervals over which the means and covariances are predicted (i.e., the sampling times) are approximately $1/50^{\text{th}}$ of the time it takes each system to reach steady state.

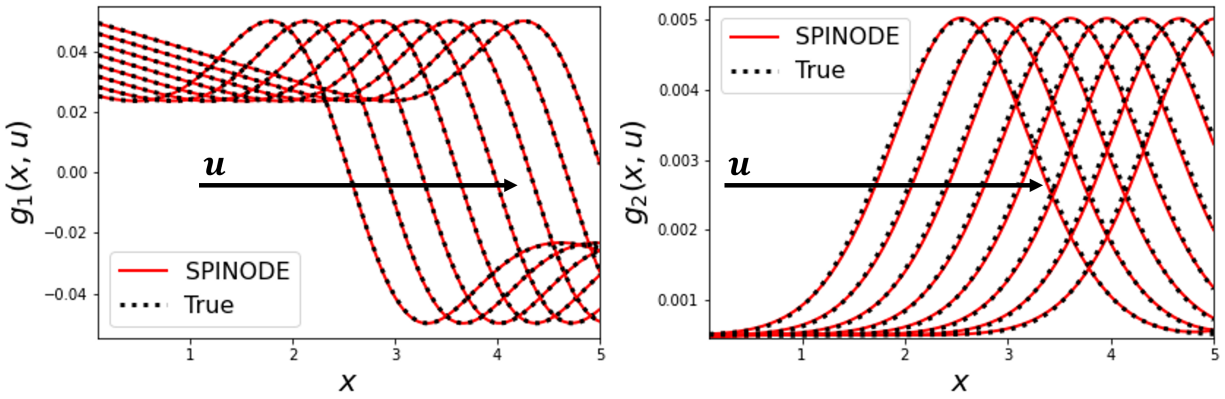


Figure 5.2: Learned hidden physics of directed colloidal self-assembly system with an exogenous input. SPINODE learns the drift and diffusion coefficients $g_1(x, u)$ and $g_2(x, u)$ of the stochastic dynamical system described by Eq. (5.22) with high accuracy.

In principle, SPINODE's performance can vary from run-to-run due to the randomness involved in neural network weight initialization and assigning data-estimated moments $m_x^{(i)}(t_k)$ to training, validation, and test sets. SPINODE's performance is then assessed by calculating the root mean squared errors (RMSE) between the learned hidden physics $g(x; \theta^*)$ and the actual system hidden physics $g(x)$ over 30 SPINODE runs with randomly selected initial weight values and training/validation/test set data assignments. Table 1 shows the mean and standard deviations of these RMSEs while Figs. 5.2 – 5.4 show a visual comparison of $g(x; \theta^*)$ and $g(x)$ for representative runs. In each case, SPINODE learns the hidden physics $g(x; \theta^*)$ with high accuracy and low run-to-run variation. Note that in the real-world, the actual values of the hidden physics $g(x)$ will be unavailable. In these cases, SPINODE's performance should be validated via the methodology described in Section 5.7., i.e., by comparing the data-estimated moments of trajectories from the real dynamics $m_x^{(i)}(t_k)$ to those generated from the learned dynamics involving $f(\cdot)$, $h(\cdot)$, and $g(x, \theta^*)$. Visual representations of the time-evolution of the probability distributions of the states from randomly selected initial conditions and exogenous input values for the colloidal self-assembly and Lotka-Volterra case studies are shown in Figs. 5.5 – 5.6.

Further note that the hidden physics reconstructions shown in Figs. 5.2 – 5.4 and Table 1 essentially occur under "ideal" conditions – as $g(x; \theta)$ is trained using a large number of moments $m_x^{(ij)}(t_k)$ that are estimated from a large number of repeated trajectories from a large number of initial conditions (see Section 5.3 and Section 5.8 for details). In addition, the sampling times are identical to the discretization times used in the Euler-Maruyama simulations that represent the "true" system dynamics. This last point motivated the use of an Euler ODE solver to predict the moment time-evolution. In the next section, The performance of SPINODE is then assessed after decreasing the number of repeated trajectories used to estimate the moments $m_x^{(ij)}(t_k)$, decreasing the total number of moments

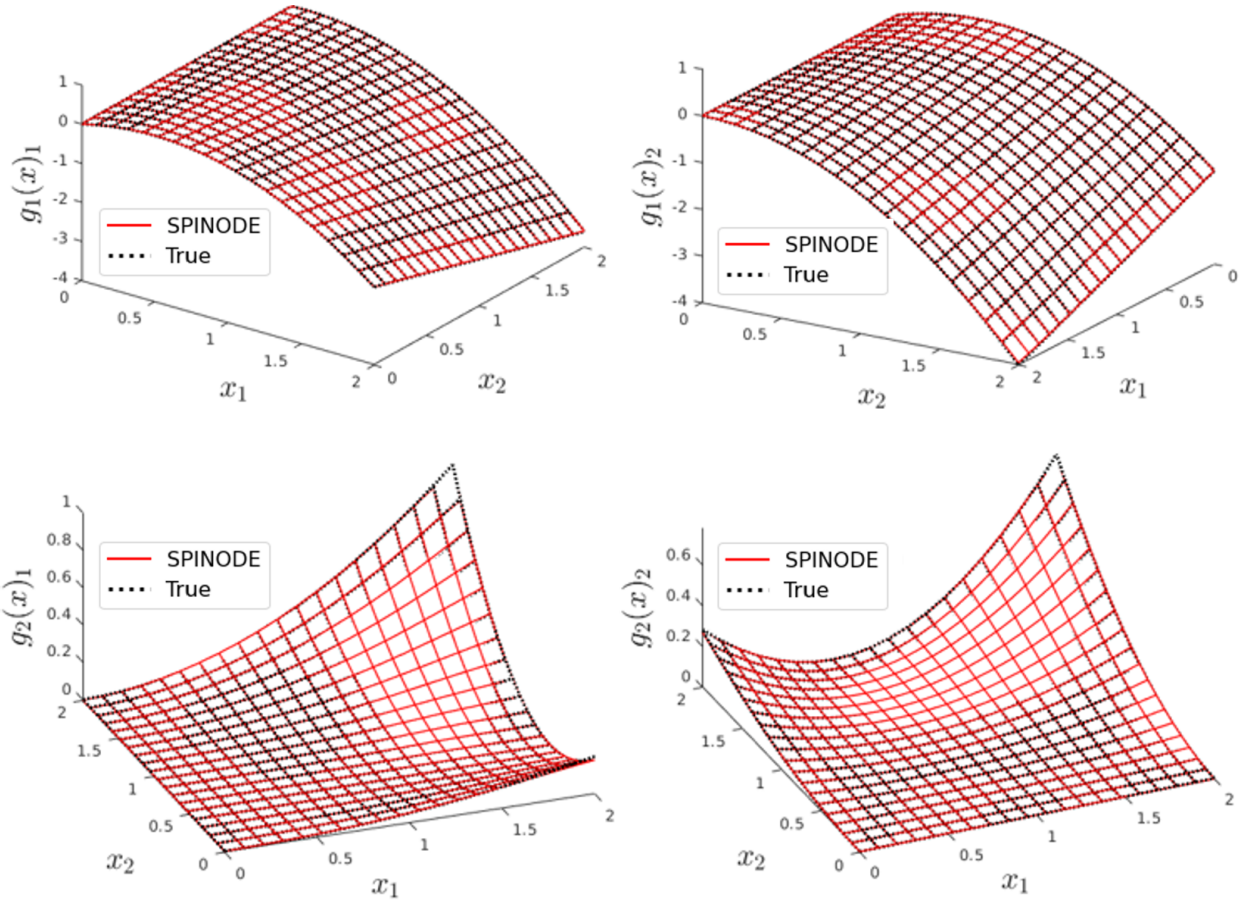


Figure 5.3: Learned hidden physics of competitive Lotka-Volterra with a coexistence equilibrium. SPINODE learns the drift and diffusion coefficients $g_1(x_1, x_2)_{1,2}$ and $g_2(x_1, x_2)_{1,2}$ of the stochastic dynamical system described by Eq. (5.23) with high accuracy.

$m_x^{(ij)}(t_k)$ used to train $g(x;\theta)$, altering the uncertainty propagation strategy, and adjusting the sampling time.

5.10 Numerical Robustness

Figs. 5.7-5.8 show RMSEs between the learned hidden physics $g(x;\theta^*)$ and the actual system hidden physics $g(x)$ for the Lotka-Volterra and SIR epidemic case studies as a function of the total number of repeated trajectories used to calculate the data-estimated moments $m_x^{(i)}(t_k)$. As can be seen, the RMSEs converge between 10^3 and 10^4 total repeats in both case studies and the RMSEs grow very quickly under 10^2 total repeats. Figs. 5.7-5.8 highlight that SPINODE's ability to the learn the hidden physics $g(x)$ critically hinges on how accurately moments $m_x^{(ij)}(t_k)$ can be estimated from data. In this work, moments are estimated from

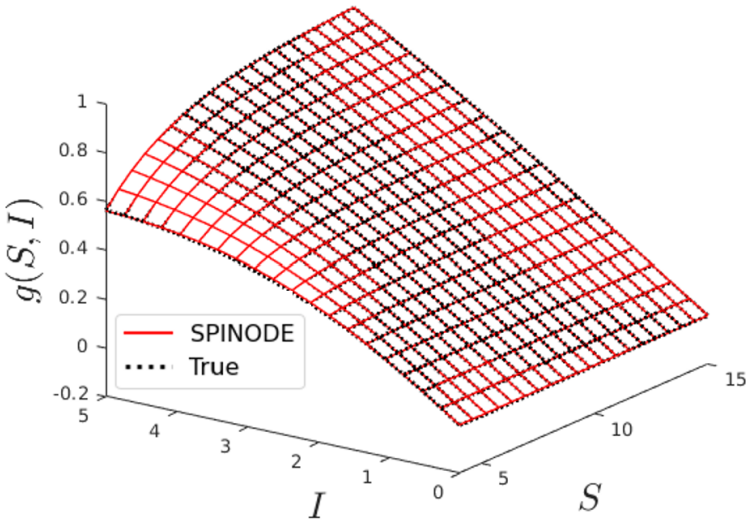


Figure 5.4: Learned hidden physics of susceptible-infectious-recovered (SIR) epidemic model. SPINODE learns $g(S, I)$ from Eq. (5.24) with high accuracy.

Case Study	RMSE Mean	RMSE Std
Colloidal Self-Assembly, $g_1(x, u)$	1.33×10^{-4}	2.77×10^{-5}
Colloidal Self-Assembly, $g_2(x, u)$	4.97×10^{-5}	4.67×10^{-6}
Lotka-Volterra , $g_1(x_1, x_2)_1$	9.20×10^{-4}	1.24×10^{-4}
Lotka-Volterra , $g_1(x_1, x_2)_2$	7.91×10^{-4}	7.59×10^{-5}
Lotka-Volterra , $g_2(x_1, x_2)_1$	3.93×10^{-3}	5.97×10^{-5}
Lotka-Volterra , $g_2(x_1, x_2)_2$	4.88×10^{-3}	7.15×10^{-5}
Susceptible-Infectious-Recovered , $g(S, I)$	2.62×10^{-3}	1.89×10^{-4}

Table 5.1: Reconstruction root mean square errors of learned hidden physics. SPINODE is used to learn the hidden physics of the case studies in Sections 5.8 – directed colloidal self-assembly with an exogenous input, competitive Lotka-Volterra with a coexistence equilibrium, and the susceptible-infectious-recovered epidemic model. The root mean square error (RMSE) between the learned hidden physics $g(x; \theta^*)$ and actual hidden physics $g(x)$ is then calculated. This process is repeated 30 times with randomly selected initial weight values and training/validation/test set data assignments. The means and standard deviations (std) of the calculated RMSEs are shown. For each case study, SPINODE learns the hidden physics $g(x; \theta^*)$ with high accuracy and low run-to-run variation.

data by repeating (many) stochastic trajectories from identical initial conditions. Section 5.3 discusses how this strategy is not appropriate for systems in which one does not have control over initial conditions, number of replica runs, consistent measurement times, etc. Section 5.3 also suggests potential methods for learning data-estimated moments in such cases.

Figs. 5.9-5.10 plot the RMSEs between the learned hidden physics $g(x; \theta^*)$ and the actual system hidden physics $g(x)$ for the Lotka-Volterra and SIR epidemic case studies as a function of the total number of data-estimated moments $m_x^{(i)}(t_k)$ used to train $g(x; \theta)$. In

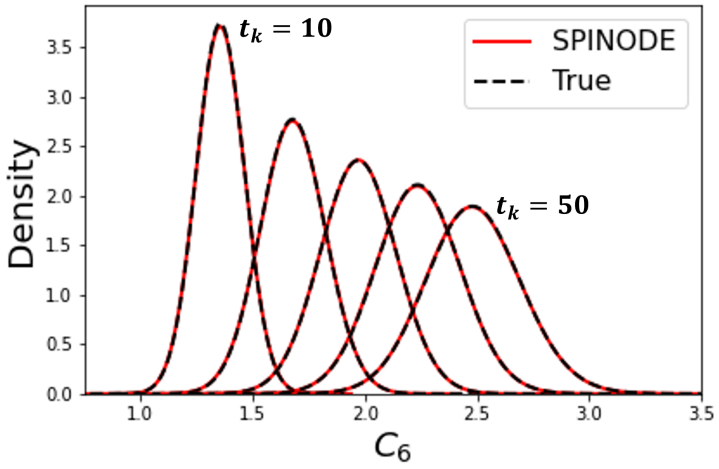


Figure 5.5: Time evolution of kernel density estimates for directed colloidal self-assembly system with an exogenous input. Trained neural networks $g_1(x, u; \theta_1^\star)$ and $g_2(x, u; \theta_2^\star)$ are used to simulate the system dynamics from a randomly selected initial condition with a randomly selected exogenous input. The true dynamics are then simulated using the same initial condition and exogenous input. In each case, the stochastic trajectory is repeated 10^5 times and kernel density functions are calculated at each sampling time. Estimates of the kernel density function for the "true" and "learned" dynamics at select sampling times are plotted against one another. SPINODE reproduces the kernel density function with high accuracy.

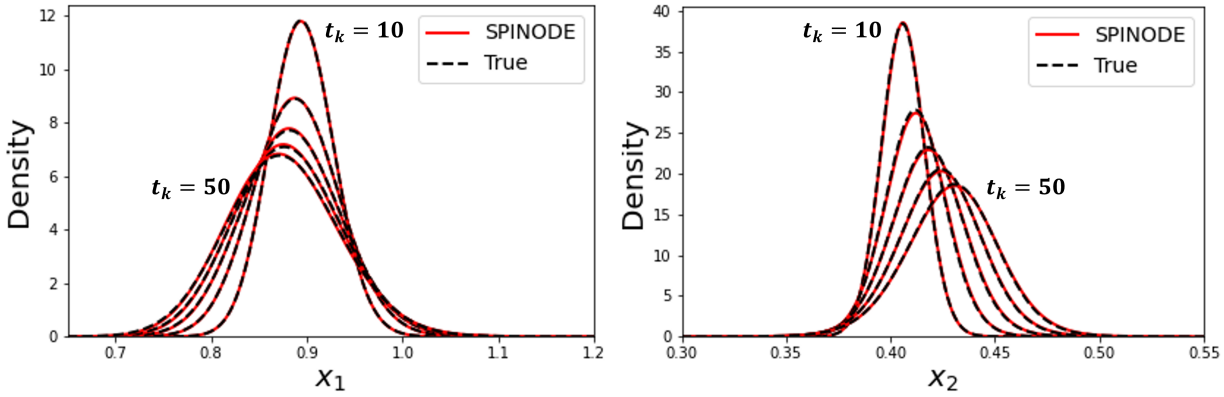


Figure 5.6: Time evolution of kernel density estimates for Lotka-Volterra with a coexistence equilibrium. Trained neural networks $g_1(x_1, x_2; \theta_1^\star)$ and $g_2(x_1, x_2; \theta_1^\star)$ are used to simulate the system dynamics from a randomly selected initial condition. The true dynamics are then simulated using the same initial condition. In each case, the stochastic trajectory is repeated 10^5 times and kernel density functions are calculated at each sampling time. Estimates of the kernel density function for the "true" and "learned" dynamics at select sampling times are plotted against one another. SPINODE reproduces the kernel density function with high accuracy.

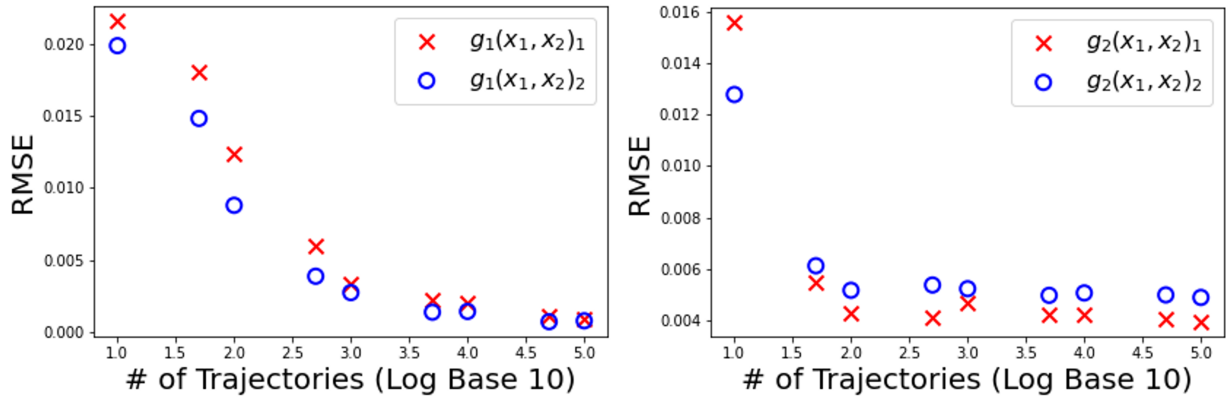


Figure 5.7: Sampling sensitivity analysis: competitive Lotka-Volterra with a coexistence equilibrium. SPINODE trains the neural networks that approximate $g_1(x_1, x_2)_{1,2}$ and $g_2(x_1, x_2)_{1,2}$ after decreasing the total number of repeated stochastic trajectories used to estimate the moments $m_x^{(i)}(t_k)$ from data. The root mean square errors (RMSEs) between the learned and actual hidden physics are then calculated. The RMSEs converge around 10^3 total repeats for $g_1(x_1, x_2)_{1,2}$ and 10^2 total repeats for $g_2(x_1, x_2)_{1,2}$.

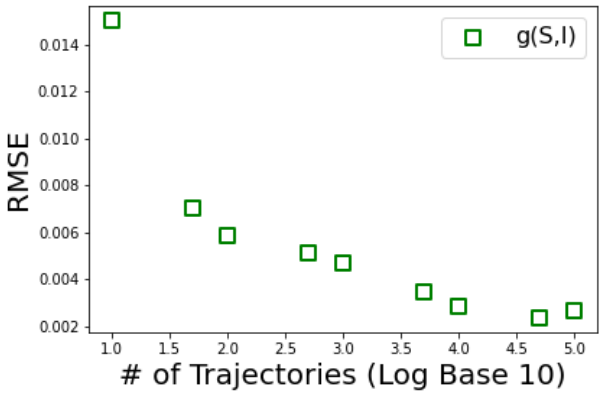


Figure 5.8: Sampling sensitivity analysis: susceptible-infectious-recovered (SIR) epidemic model. SPINODE trains the neural network that approximates $g(S, I)$ after decreasing the total number of repeated stochastic trajectories used to estimate the moments $m_x^{(i)}(t_k)$ from data. The root mean square error (RMSE) between the learned and actual hidden physics is then calculated. The RMSE converges around 10^4 total repeats.

this case, each moment $m_x^{(i)}(t_k)$ is estimated using 10^5 total repeated trajectories – only the total number of moments used to train $g(x; \theta)$ is varied. Both figures show that more training data can lead to a more accurate recovery of the hidden physics. The amount of training data required for the RMSEs to converge depends on a combination of the complexity of $g(x)$ and the "informativeness" of the loss function used to train $g(x; \theta)$. For example, the behavior of $g(S, I)$ in SIR epidemic case study can be considered more nonlinear than that

of $g_1(x_1, x_2)_1$ and $g_1(x_1, x_2)_2$ in the Lotka-Volterra case study, which is more nonlinear still than that of $g_2(x_1, x_2)_1$ and $g_2(x_1, x_2)_2$ in the Lotka-Volterra case study, Correspondingly, the RMSEs of $g_2(x_1, x_2)_1$ and $g_2(x_1, x_2)_2$ converge after fewer total data points than the other hidden physics terms. Despite the more nonlinear behavior of $g(S, I)$, however, its RMSE converges earlier than the RMSEs of $g_1(x_1, x_2)_1$ and $g_1(x_1, x_2)_2$. Note that the cost function used to train $g(S, I)$ is more "informative" than the cost functions used to train $g_1(x_1, x_2)$ and $g_2(x_1, x_2)$ – compare Eq. (5.20) to Eq. (5.25)) – as Eq. (5.25) contains added information from multiple known "physical" terms in Eq. (5.24). Overall, the general notion that more training data can lead to higher-performing neural network models is expected [248]. However, the fact that $g(S, I)$'s RMSE seems to converge at fewer total data points suggests the previously reported observation [207] that incorporating more physics into the cost function can reduce data requirements for training neural networks.

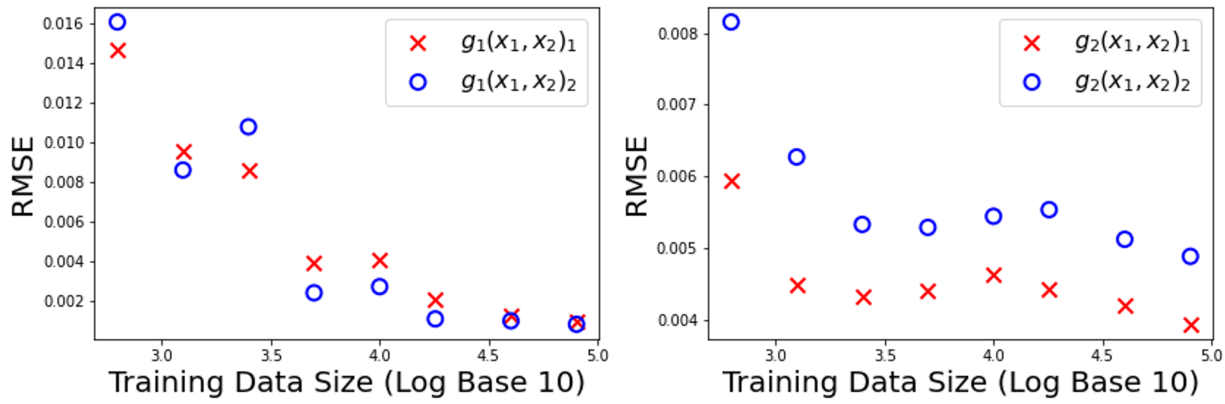


Figure 5.9: Training data size sensitivity analysis: competitive Lotka-Volterra with a coexistence equilibrium. SPINODE trains the neural networks that approximate $g_1(x_1, x_2)_{1,2}$ and $g_2(x_1, x_2)_{1,2}$ after decreasing the size of the training data (i.e., the total number of data-estimated moments $m_x^{(i)}(t_k)$). The root mean square error (RMSE) between the learned and actual hidden physics is then calculated. The RMSEs converge around 2.5×10^4 total moments for $g_1(x_1, x_2)_{1,2}$ and 5×10^3 total moments for $g_2(x_1, x_2)_{1,2}$.

The colloidal self-assembly case study is next used to investigate SPINODE's sensitivity to the chosen uncertainty propagation method. Fig. 5.11 shows SPINODE's reconstruction of the hidden physics $g_1(x, u)$ and $g_2(x, u)$ when propagating stochasticity via linearization [236] and two methods based on unscented transform: UT-2M and UT-4M. UT-2M, which is explained in detail in Sections 5.4 and 5.6, describes the time evolution of the mean and covariance based on the data-estimated mean and covariance at previous time points. UT-4M, which can be viewed as an extension of UT-2M based on the work in reference [237], describes the time evolution of the mean and covariance based on the data-estimated means, covariance, skew, and kurtosis at previous time points. SPINODE with both UT methods significantly outperforms SPINODE with linearization. This performance discrepancy indicates that the UT methods propagate stochasticity through Eq. (5.22) much more ac-

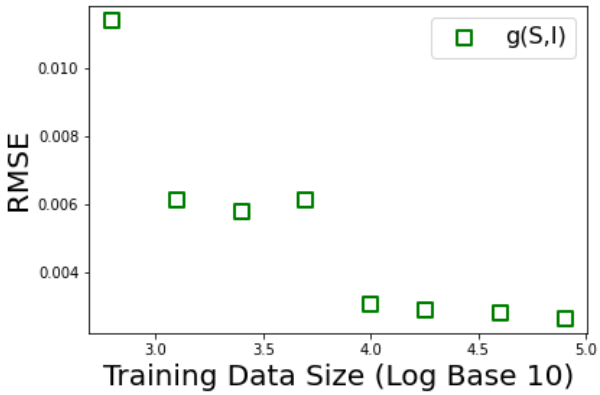


Figure 5.10: Training data size sensitivity analysis: susceptible-infectious-recovered (SIR) epidemic model. SPINODE trains the neural networks that approximate $g(S, I)$ after decreasing the size of the training data (i.e., the total number of data-estimated moments $m_x^{(i)}(t_k)$). The root mean square error (RMSE) between the learned and actual hidden physics is then calculated. The RMSE converges slightly before 10^4 total moments.

curately than does the linearization method. Although SPINODE with UT-2M and UT-4M learn $g_1(x, u)$ with near identical accuracy, the RMSE of SPINODE with UT-4M's recovery of $g_2(x, u)$ is marginally lower than the RMSE of SPINODE with UT-2M's recovery of $g_2(x, u)$ (i.e., 8.64×10^{-5} vs. 4.67×10^{-5}). UT-4M thus leads to a more accurate prediction of the time evolution of the covariance than UT-2M does, as only the covariance is used to train $g_2(x, u)$ (see Eq. (5.20)). The latter point supports our earlier remark that SPINODE's ability to incorporate higher moments can make SPINODE well-suited for learning $g(x)$ when $f(x, (g(x))$ and $h(x, (g(x))$ are highly nonlinear and the distribution of x is non-Gaussian as a result. Note that although kernel density estimations in Fig. 5.5 appear fairly Gaussian, the relatively minor skews and kurtoses of the distributions of $x(t)$ are still large enough to affect the uncertainty propagation.

This chapter further investigates SPINODE's sensitivity to uncertainty propagation by extending the sampling times at which data-estimated moments are collected. All previous results for the colloidal self-assembly case study used a sampling time of 1 second. Fig. 5.12a plots the RMSEs of $g_1(x, u)$ and $g_2(x, u)$ (when UT-4M is implemented for uncertainty propagation) as a function of sampling time. The RMSE increases nearly linearly with sampling time. Fig 5.12b shows that the prediction errors for the mean and covariance also increase nearly linearly with sampling time. It is thus reasonable to suggest that SPINODE's sensitivity to sampling time in the colloidal self-assembly case study can be attributed to the sensitivity of the uncertainty propagation method to the sampling time.

Above all else, Figs. 5.11–5.12 demonstrate SPINODE's sensitivity to the choice of uncertainty propagation method. Section 5.1 discusses how SPINODE can in principle accommodate any uncertainty propagation method. As a result, the uncertainty propagation method should be viewed as a "hyper-parameter" within SPINODE. The choice of the ODE

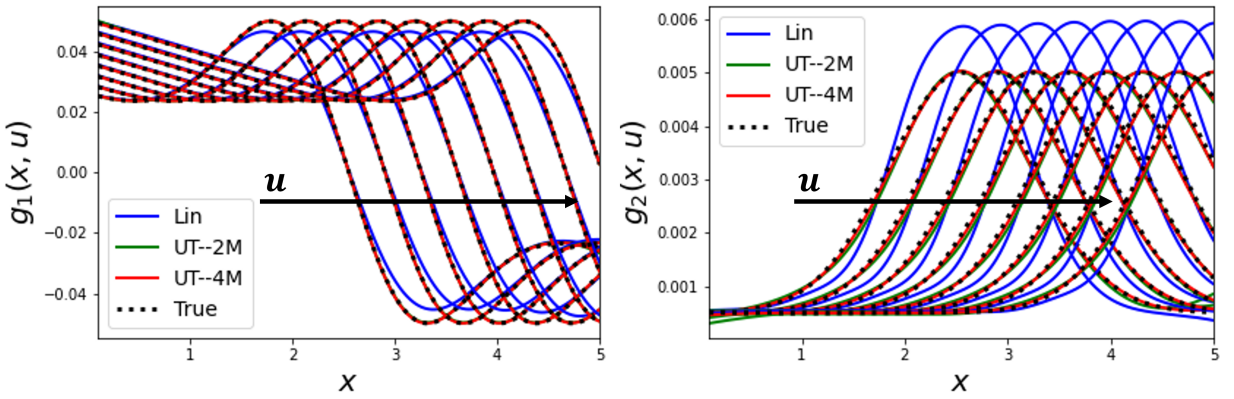


Figure 5.11: Uncertainty propagation sensitivity analysis: directed colloidal self-assembly with an exogenous input. SPINODE trains neural networks that approximate $g_1(x,u)$ and $g_2(x,u)$ using three different uncertainty propagation methods: linearization (Lin), unscented transform with two moments (UT-2M) and unscented transform with four moments (UT-4M). UT-2M describes the time evolution of the mean and covariance based on the data-estimated mean and covariance at previous time points while UT-4M describes the time evolution of the mean and covariance based on the data-estimated means, covariance, skew, and kurtosis at previous time points. SPINODE with UT-2M and UT-4M significantly outperforms SPINODE with linearization, while SPINODE with UT-4M slightly outperforms SPINODE with UT-2M for learning $g_2(x,u)$.

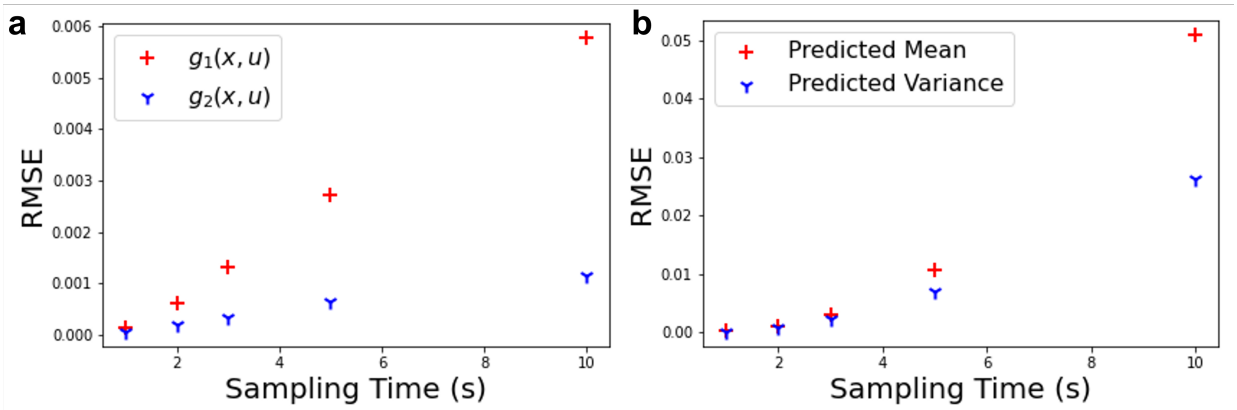


Figure 5.12: Sampling time sensitivity analysis: directed colloidal self-assembly with an exogenous input. (a) SPINODE trains neural networks that approximate $g_1(x,u)$ and $g_2(x,u)$ using the UT-4M uncertainty propagation method with different sampling times. The root mean square error (RMSE) between the learned and actual hidden physics is then calculated. (b) UT-4M is used to propagate stochasticity through the true dynamics (i.e., Eq. (5.22)) for each of the data-estimated moments in the training data set at different sampling times. The RMSEs between the predicted and the data-estimated means and covariances at the next sampling time are then calculated. The errors in reconstructing the hidden physics and predicting the mean and covariance grow nearly linearly with the sampling time.

solver within SPINODE must also be discussed. Because the case study simulation data was generated via an Euler-Maruyama discretization, an Euler ODE solver within SPINODE yielded the most accurate reconstructions of the hidden physics. The current implementation of SPINODE [221], however, includes other advanced, even adaptive time-step solvers that have been shown to integrate accurately high-dimensional, stiff, and nonlinear ODEs [205]. The choice of ODE solver should thus also be viewed as a "hyper-parameter" within SPINODE. In fact, known hyper-parameter optimization strategies such as Bayesian Optimization [249] can be used to determine the "best" ODE solver to use during training.

5.11 Conclusions and Next Steps

This chapter proposed a flexible and scalable framework based on the notions of neural ordinary differential equations, physics-informed neural networks, and moment-matching for training deep neural networks to learn constitutive equations that represent hidden physics within stochastic differential equations. This chapter demonstrated the proposed stochastic physics-informed neural ordinary differential equation framework on three benchmark *in-silico* case studies from the literature. This chapter analyzed the performance of the proposed framework in terms of its repeatability, sensitivity to weight initialization and training/validation/testing set allocation, total number of data points, total number of repeated trajectories, uncertainty propagation method, and sampling time. This chapter showed the framework's scalability by learning highly nonlinear hidden physics within multidimensional stochastic differential equations with multiplicative noise. This chapter illustrated the framework's flexibility by (i) learning both general drift and diffusion coefficients (with or without an exogenous input) and specific unknown functions within stochastic differential equations for different systems and (ii) demonstrating that key aspects of the framework (e.g., the choice of uncertainty propagation method) can be easily and independently adjusted. An open challenge is that a large number of repeated state trajectories are required to learn accurately hidden physics. Future work will focus on learning probability distributions directly from data instead of estimating moments from repeated stochastic trajectories from identical initial conditions. To this end, future work will explore variational autoencoders [227–229], generative adversarial networks [209, 230–232], and energy-based models [233–235]. Other open challenges include optimizing the uncertainty propagation method and choice of ODE solver during neural network training. Future work will also explore optimizing these "hyper-parameter" choices using methods based on Bayesian optimization [249]. Finally, the current implementation of the neural ODE framework based on [205] should be updated with more recent and advanced neural ODE framework implementations (e.g., [206]).

Despite the flexiblity and scalability of SPINODE, SPINODE is fairly data-intensive. Moreover, these data requirements may increase with limited precision in the measurement of the system state – it is reasonable to assume that imprecise system state measurements can require large number of repeated trajectories. It is known that system state measurements of experimental colloidal self-assembly systems can be imprecise [14, 55]. Moreover, the data

collection required to build accurate low-dimensional stochastic models may be unrealistic for such systems.

It is important to note that an initial motivation for this work was to investigate methods for developing accurate, low-dimensional stochastic models for feedback control. Chapter 4 clearly shows, however, that guiding colloidal SA may not require excessively complex input profiles. For example, the temperature (i.e., global manipulated variable) profiles in Fig. 4.4 are fairly simple – they consist of a series of quenches and holds. This begs the question: Is a complex model-based feedback control approach really needed to control colloidal self-assembly effectively? Recent work by the authors of [47, 52] indicates just the opposite. These authors use model-free approaches to feedback control based on evolutionary reinforcement learning [250, 251] to control very effectively the two-dimensional colloidal SA of an *in-silico* system of patchy disks. The next chapter investigates model-free methods for feedback control of the three-dimensional SA of a benchmark *in-silico* system of colloids.

Chapter 6

Model-Free Feedback Control of Colloidal Self-Assembly Systems

This chapter investigates a model-free feedback control framework based on evolutionary reinforcement learning for achieving reproducible colloidal self-assembly. The framework learns a neural network representation of a control policy that maps colloidal self-assembly state information to the values of global self-assembly actuators such as temperature and pressure ramp rates. In particular, this chapter compares the efficacy of "objective search," where progress towards a pre-defined goal is measured and control policies are iteratively updated according to this progress, and "novelty search," where behavioral novelty alone is used to learn control policies. The model-free feedback control framework is demonstrated on a benchmark in-silico colloidal self-assembly system that consists of 2048 colloids that interact via Lennard-Jones interparticle potentials within an NPT ensemble. Closed-loop control policies are learned using objective search and novelty search and their performance is analyzed and compared.

6.1 Introduction

Absent a (stochastic) model for predicting colloidal SA dynamics, recent work [47, 52, 60] has proposed the use of reinforcement learning (RL) to learn control policies that directly map colloidal SA state information (e.g., quantifications of colloidal structure) to values of global SA actuators (e.g., temperature ramp rates, changes in electric field voltage). These control policies in turn have been shown to guide consistently colloidal SA systems towards desired structures. RL is a branch of machine learning concerned with learning to perform actions so as to achieve an objective (i.e., maximize an objective function), and has been used extensively to outperform humans in video games, manipulate robotics, and enhance natural language processing algorithms, among many other applications [252–254]. Broadly, there exist two classes of RL algorithms – those that explicitly evaluate gradients of the objective function and those that do not [255]. Evolutionary RL methods fall under the latter class,

as these methods propose and probabilistically accept changes to candidate control policies [250, 251]. Although evolutionary RL is much less widely applied than gradient-based RL, evolutionary RL is naturally suited to "sparse-reward" problems such as colloidal SA, where the outcome of assembly (i.e., whether or not a desired structure forms) is not apparent until the later stages of the assembly process [47, 250, 256]. The authors in [47] implement evolutionary RL to learn control policies that guide the two-dimensional colloidal SA of an *in-silico* system of patchy discs towards desired structures. The learned control policies avoid kinetically arrested states and can even control the polymorph into which certain sets of patchy discs assemble.

The control policies in [47] are represented as artificial neural networks that contain thousands of parameters. Despite the complexity in model structure, the control policies yield fairly simple (e.g., quadratic) relationships between colloidal SA state information and the system's global SA actuators. It may thus be reasonable to conclude that a simpler model architecture with fewer parameters could have yielded similar control performance. The simpler model architecture may in turn converge with less data and be more robust to overfitting. Given the well-documented relationship between model architecture and performance [252–254], it may be beneficial to explore RL algorithms that learn model architecture and parameters simultaneously.

Most RL strategies (including that of [47]) learn feedback control policies through objective search, where progress towards a pre-defined goal is measured and the control policies are iteratively updated according to this progress. Objective search RL strategies, however, are prone to learning poor-performing control policies for "deceptive" systems that must first be guided further away from their pre-defined goals before ever achieving them [257, 258]. Complex colloidal SA systems are inherently deceptive, as successful control policies must be able to initiate disassembly (and late re-assembly) of unavoidable kinetically arrested structures to smooth out defects and achieve desired structures [38, 55]. Although explored significantly less than objective search, RL methods based on novelty search have learned superior control laws for many deceptive systems [257, 258]. These RL approaches search for behavioral novelty alone and create a cache of the learned control policies that cause the "most novel" system behaviors. The cached policy that best accomplishes a pre-defined goal is then selected.

The purpose of this chapter is not merely to use established evolutionary RL algorithms to learn control policies that guide the SA of a given colloidal system, rather the goal is to perform a proof-of-concept study that investigates the the viability of (i) learning network parameters and architecture simultaneously and (ii) searching for behavioral novelty to learn such policies. More specifically, this chapter uses a class of evolutionary RL algorithms called NEAT, or NeuroEvolution of Augmenting Topologies [259], to learn colloidal SA control policies via objective search and novelty search. NEAT evolves the neural network weights and architecture simultaneously with the goal of discovering the smallest neural network architecture required to solve a given problem. NEAT is initialized with artificial neural networks that only contain input and output nodes and discovers progressively more complex behavior over time. As a result, when paired with novelty search, the order in which new

behaviors are discovered is principled (from less complex to more complex) and not random [258]. In fact, NEAT with objective search and novelty search has been shown to learn effective control policies for a number of complex problems [257, 258, 260].

This chapter uses NEAT with objective search and novelty search to learn closed-loop control policies that manipulate the temperature and pressure of a benchmark *in-silico* system of 2048 colloids with Lennard-Jones interparticle potentials to create high-quality FCC crystals. Section 6.2 describes the colloidal SA benchmark test system in more detail. Section 6.3 describes the chosen colloidal state characterization method. Section 6.4 more formally describes the colloidal SA control objectives. Sections 6.5-6 describe NEAT, objective search, and novelty search in more detail and further justify their application towards control of colloidal SA. Section 6.7 analyzes the results of the learned closed-loop control policies. Section 6.8 discusses potential strategies to use novelty search in the design of future control approaches and suggests future experiments.

6.2 Colloidal Self-Assembly System Description

The test system is a three-dimensional *in-silico* system of 2048 colloids in an NPT ensemble with Lennard-Jones potentials of form:

$$V(r) = 4\epsilon \left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6} \right], \tag{6.1}$$

where $\sigma = \epsilon = 1.0$ and r is the inter-particle distance. All simulations were performed in HOOMD-blue [261] and all parameter values can be found in [261] (e.g., buffer is 0.4, the coupling constant for the thermostat is 1.0, the coupling constant for the barostat is 1.2, the integrator time step is 0.005). All simulations are initialized from a high "temperature" and low "pressure" (i.e., $kT = 3.0$ and $S = 0.25$). Each simulation is run for 10^5 time steps (i.e., the batch time is 10^5 time steps). The control policies in this chapter predict changes in the temperature and pressure (i.e., ΔkT and ΔS) every 500 time steps (i.e., the sampling time is 500 time steps). As a result, each simulation of 10^5 time steps will contain 200 changes to the temperature and pressure.

The goal of this chapter is to implement evolutionary RL algorithms based on NEAT with objective and novelty search to manipulate the temperature and pressure of the test system to assemble high-quality FCC crystals. Although linear temperature and pressure ramps from the given initial condition of $kT = 3.0$ and $S = 0.25$ can create high-quality FCC crystals after 10^6 time steps, this chapter only focuses on batch times of 10^5 time steps, where linear ramps and cools exclusively form extremely defective, low-quality crystals (see Fig. 6.1).

6.3 Colloidal State Characterization

The model-based feedback control framework in Chapter 4 relied on a novel state characterization strategy based on autoencoders and common neighbor analysis (CNA) [83, 84, 106]. A key motivation for using CNA was that CNA is particularly well-suited for identifying hexagonal topologies in two-dimensional lattices. CNA had also been shown to characterize effectively the test system's colloidal SA structures in previous works [62, 87, 164]. A primary motivation for the use of an autoencoder (which translated the CNA-based characterization to a lower-dimensional space) was that the model-based feedback control framework would become intractable with more than a few state variables. For the test system in this work, Steinhardt bond-order parameters q_4, q_6, q_8, q_{10}, and q_{12} have been shown to characterize effectively and distinguish FCC, BCC, SC, and a variety of defective states that can appear during SA at the single particle level [81, 96]. Moreover, at the lattice level, the averages of q_4, q_6, q_8, q_{10}, and q_{12} over all particles have been shown to distinguish lattices that are primarily FCC, BCC, SC, defective, etc. [81, 96]. From now on, the Steinhardt bond-order parameters that refer to this averaged, lattice-level characterization will be denoted as $\bar{q}_i$ Note that Steinhardt bond-order parameters' sensitivity to thermal fluctuations, density gradients, and anisotropy due to the use of a cut-off radius is well documented [79, 80]. The work in this chapter and the work of [96] mitigates some of this sensitivity by avoiding the use of a cut-off radius and instead establishing local structure via Voronoi cells. Moreover, there is no need to reduce the dimensionality of the Steinhardt bond-order parameters as evolutionary RL algorithms can tractably learn control policies with upwards of 5 state variables [258–260]. To summarize, each particle in the colloidal SA lattice can be characterized by a vector of Steinhardt bond-order parameters, each entry of which is between 0 and 1 (i.e., $[q_4, q_6, q_8, q_{10}, q_{12}]^\top$) and the entire lattice can be characterized by the averages of these Steinhardt bond-order parameters over all particles (i.e., $[\bar{q}_4, \bar{q}_6, \bar{q}_8, \bar{q}_{10}, \bar{q}_{12}]^\top$).

6.4 Problem Formulation

The objective of this chapter is to learn a control policy that manipulates temperature and pressure to create high-quality FCC crystals at the end of the given batch time (e.g., 10^5 time steps). Given that the Steinhardt bond-order parameters of the particles within a theoretically perfect FCC lattice are known (i.e., $[q_4^*, q_6^*, q_8^*, q_{10}^*, q_{12}^*]^\top$, the "control objective" is to learn a control policy that minimizes the distance between the Steinhardt bond-order parameters of the individual particles in the colloidal SA system at the end of the batch time $[q_4, q_6, q_8, q_{10}, q_{12}]^\top$ and the theoretically perfect FCC values $[q_4^*, q_6^*, q_8^*, q_{10}^*, q_{12}^*]^\top$. This

goal can be mathematically formulated as follows:

$$\begin{aligned} \max_{x} K - \sum_{i=0}^{N} (q_{j,i} - q^*_{j,i})^2 \\ \text{s.t. } x \in X \\ j = 4, 6, 8, 10, 12 \end{aligned} \tag{6.2}$$

where K is a large constant, N is the number of particles in the colloidal SA system j is the Steinhardt bond-order parameter index, $\sum_{i=0}^{N}(q_{j,i} - q^*_{j,i})^2$ is the objective function, and $x \in X$ represents the space of all possible control policies. To summarize, this chapter uses evolutionary RL algorithms based on NEAT with objective search and novelty search to find the control policy x that maximizes Eq. (6.2). The next section discusses the NEAT algorithm in more detail.

6.5 Neuroevolution of Augmenting Topologies (NEAT)

NEAT, like most evolutionary RL algorithms, is initialized by creating a *population* of *genomes* [259]. At this stage, each genome is a candidate control policy that is represented by an artificial neural network with randomly selected weights and biases and the simplest possible architecture (e.g., no hidden nodes). During each *generation*, each genome's *fitness* is calculated. In this chapter, fitness is calculated by a genome guiding the colloidal SA of the test system. The genomes with the largest fitnesses are then chosen to be the "parents" of the next generation. These parents are probabilistically mutated, and the process repeats itself for a pre-determined number of generations or until a pre-determined control performance is reached. Mutations can add nodes and node-to-node connections, disable node-to-node connections, and change activation functions, and/or change weight and bias values.

NEAT is unique in that it begins evolution with a population of small, simple networks and complexifies the network topology into diverse species over generations, leading to the potential for increasingly sophisticated behavior; although complexifying the structure of an artificial neural network does not always increase the complexity of the behavior of the neural network, it does increase the upper bound of possible behavioral complexity by adding more parameters. Simpler behaviors must thus be encountered before more complex behaviors. Encountering simple behaviors first is significant because the most complex behaviors are often associated with irregularity and chaos. A key feature distinguishing NEAT from prior work in complexification is its unique approach to maintaining a healthy diversity of complexifying structures simultaneously. For example, to keep track of which gene is which while new genes are added, a historical marking is uniquely assigned to each new structural component. During crossover (i.e., two parent genomes "combining" to create a new

genome), genes with the same historical markings are aligned to produce meaningful offspring efficiently. NEAT also employs speciation – where only fitnesses among sufficiently similar structures are compared during each generation. Speciation protects new structural innovations by reducing competition among differing structures, thereby giving newer, more complex structures room to adjust. Networks are assigned to species based on the extent to which they share historical markings. Complexification is thus supported by both historical marking and speciation, allowing NEAT to establish high-level features early in evolution and then elaborate on them later. In effect, then, NEAT searches for a compact, appropriate network topology by incrementally complexifying existing structure.

6.6 Objective Search and Novelty Search

The previous section mentions that genome fitness is evaluated by using a genome (and its corresponding control policy) to guide the colloidal SA of the test system. When NEAT is implemented with objective search, the fitness is calculated by evaluating the objective function in Eq. (6.2). The goal of objective search is thus to find genomes that progressively yield larger objective function values. Objective search evolutionary RL algorithms, however, are prone to local optima and have performed poorly when applied to deceptive systems that must first be guided further away from their pre-defined goals before ever achieving them [257, 258, 260]. One reasonable explanation for this problem is that the objective function does not necessarily reward the stepping stones in the search space that ultimately lead to the control objective (i.e., a high-quality FCC crystal in this case). Novelty search, on the other hand, proposes an alternative fitness criteria based on behavioral novelty alone. That is, instead of searching to maximize continuously the objective function, the evolutionary RL algorithm rewards genomes whose functionality is significantly different than what has been discovered before. Instead of a traditional objective function, the evolution employs a novelty metric that in no way measures overall progress. For example, when trying to learn a control policy that achieves biped locomotion, initial candidate genomes may lead the biped simply to fall down. The novelty metric would reward falling down in a different way, regardless of whether it is closer to the objective behavior or not. In contrast, an objective function may explicitly reward falling the farthest, which likely does not lead to the ultimate objective of walking and thus exemplifies a deceptive local optimum. In contrast, in the search for novelty, a set of instances are maintained that represent the most novel discoveries. Further search then jumps from these representative behaviors. After a few ways to fall are discovered, the only way to be rewarded is to find a behavior that does not fall right away. In this way, behavioral complexity rises from the bottom up. Eventually, to do something new, the biped would have to walk successfully for some distance even though successfully walking is in no way explicitly included in the fitness calculation.

A natural question about novelty search is whether it follows any principle beyond naively enumerating all possible behaviors. The answer is that although it does attempt to find all possible behaviors over time, when combined with a complexifying algorithm like NEAT, the

order in which they are discovered is principled and not random. Recall that NEAT evolves increasingly complex neural networks. That way, the number of nodes and connections and thus the maximal complexity of neural networks discovered by novelty search increases over time, ensuring that simple behaviors must be discovered before more complex behaviors. Regardless of the particular encoding, this ordering from simple to complex is generally beneficial due to the minimum description-length principle in machine learning i.e., the idea that the simplest satisfying description is usually the best [262].

A second natural question is whether novelty search is essentially identical to exhaustive search – of course a search that enumerates all possible solutions will eventually find the solution, but at enormous computational cost. Many environments provide sufficient constraints on the types behaviors that can actually be observed, without the need for further constraint from an objective function. For example, it is known that colloidal SA systems are prone to kinetic traps. Although the control policy search space is effectively infinite, the behavior space into which points in the search space collapse is limited, as systems often tend to become trapped in a handful of relatively similar configurations. In cases such as this, the search space can collapse into a manageable number of novelty points, significantly differentiating novelty search from exhaustive search.

The novelty of a newly generated genome is computed with respect to the behaviors (i.e. not the genomes) of an archive of past genomes whose behaviors were highly novel when they originated. An archive that is smaller than the total number of observed behaviors represents the most recently visited points. The aim is to characterize how far away the new individual is from the rest of the population and its predecessors in behavior space, i.e. the space of unique behaviors. A good metric should thus compute the sparseness at any point in the behavior space. Areas with denser clusters of visited points are less novel and therefore rewarded less. A simple measure of sparseness at a point is the average distance to the k-nearest neighbors of that point, where k is a fixed parameter that is determined experimentally [263]. Intuitively, if the average distance to a given point's nearest neighbors is large then it is in a sparse area; it is in a dense region if the average distance is small. In this chapter, and many others [257, 258], the sparseness $\rho(y)$ at point y is given by:

$$\rho(y) = \frac{1}{k}\sum_{i=0}^{k} \text{dist}(y, \mu_i), \tag{6.3}$$

where μ_i is i-th nearest neighbor of y with respect to the distance metric, dist. In this chapter, dist is the l^2 norm. In this chapter's colloidal SA test system, the "behavior" is the averaged order parameter values at the final state (i.e., $[\bar{q}_4, \bar{q}_6, \bar{q}_8, \bar{q}_{10}, \bar{q}_{12}]^\top$). Note that this quantification is separate from that in the objective function, which looks at distances between observed and theoretically perfect order parameter values at the single particle level (i.e., $[q_4, q_6, q_8, q_{10}, q_{12}]^\top$ and $[q_4^*, q_6^*, q_8^*, q_{10}^*, q_{12}^*]^\top$).

6.7 Closed-Loop Implementation

NEAT with objective and novelty search was implemented on the test system to learn closed-loop control policies that manipulate temperature and pressure to guide the test system towards high-quality FCC crystals. NEAT was implemented using an open-source Python package [264]. The inputs to the control policy are the time step, temperature, pressure, and the 5 averaged order parameter values $[\bar{q}_4, \bar{q}_6, \bar{q}_8, \bar{q}_{10}, \bar{q}_{12}]^\top$. The outputs are temperature and pressure ramp rates. Each evolutionary RL trial used populations of 25 genomes with 50 generations (leading to 1250 total evaluations). Both objective search and novelty search-based protocols learned control policies that guide consistentlyd the test system towards fairly high-quality crystals. Fig. 6.1 uses OVITO [124] to visualize example lattices produced by these control policies.

Although the example lattices in Fig. 6.1 produced by both objective search and novelty search seem visually comparable, the novelty search control policy produced a more highly-ordered crystal. In fact, on average, the novelty search control policy produced more highly-ordered crystals more consistently than did the objective search policy (see Fig. 6.2). Fig. 6.3 shows the average input profiles for each control policy over 50 runs. Both temperature (i.e., kT) profiles are similar in that they contain a a large number of heating and cooling ramps (which correspond to initiating assembly and disassembly). Despite these frequent ramps, the overall shape of the objective search profile is a nonlinear cool. Meanwhile, the novelty search profile involves a fast cool, followed by a slow heating, and an even slower cooling. The objective search pressure (i.e., S) profile essentially only involves a fast pressure increase and hold. The novelty search pressure profile, on the other hand, involves a fast quench and mild oscillations before a linear pressure decrease. This pressure profile demonstrates the benefits of novelty search over objective search – the novelty search profile was able to recognize that oscillating the pressure, and in some form initiating assembly/disassembly and temporarily creating a lower-order crystal structure, was beneficial for achieving higher order in the long term. Objective search, on the other hand, opted for a fast pressure increase and the immediate increase order that comes with it. A reasonable question is why the objective search policy found a complex temperature profile, or at least one with multiple assembly/disassembly cycles. This chapter does not claim that objective search will *never* find control policies that sacrifice short term rewards for long term ones (i.e., overcome deception). Instead, this chapter and previous literature [257, 258, 260] claim that novelty search is potentially more likely to overcome deception. Fittingly, the novelty search neural network contains more node-to-node connections than the objective search network, which possibly allows the novelty search network to display more complex input profiles (see Fig. 6.4). Note that the architecture of these networks is not standard, however, as input nodes are connected to both hidden and output nodes. In adding more connections with fewer nodes, NEAT can encode reasonably complex behavior with extremely small architectures [259].

It is important to remember that this case study is by no means comprehensive. The evolutionary RL trials involved an extremely small number of evaluations and any difference

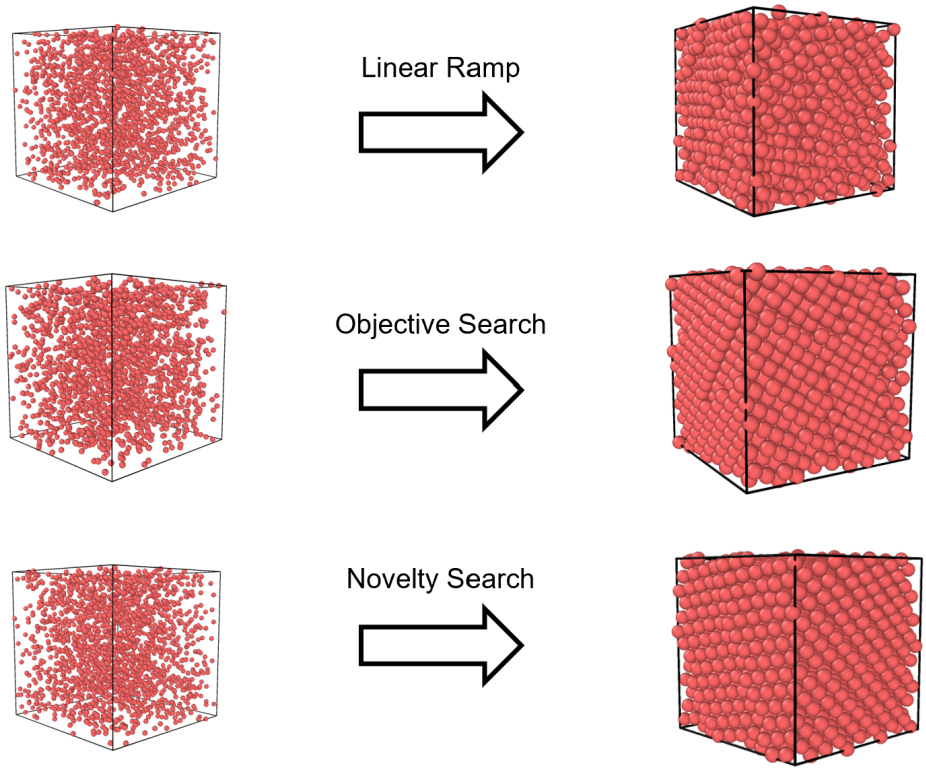


Figure 6.1: Example self-assembled lattices. Each row shows the initial and final assembled state for representative colloidal SA trajectories that are guided towards FCC crystals with (i) linear ramp control policy, (ii) a control policy learned using NEAT with objective search, and (iii) a control policy learned using NEAT with novelty search. Each initial state has a temperature of $kT = 3.0$ and a pressure of $S = 0.25$. To choose the linear ramp, a grid search was performed with final temperatures and pressures set between $kT = 0.1$ and $kT = 3.0$ and $S = 0.25$ and $S = 6.0$. Fifty trajectories were then repeated with the highest performing setting (i.e., $kT = 0.1$ and $S = 6.0$). The final state for the median performance run is shown here. The control policies learned with NEAT with objective search and novelty search were also repeated 50 times and the final states for their median performance runs are shown here. Although not visually obvious, the novelty search final lattice is more highly-ordered than the objective search lattice (see Fig. 6.2) for a more detailed comparison of the two control policies' behaviors.

between the performance of the novelty search and objective search protocols may not be meaningful. The purpose of this chapter was merely to demonstrate a proof-of-concept of NEAT with novelty search. The results shown above provide justification of further extensive testing of NEAT with novelty search on this test system and other similar ones.

6.8 Conclusions and Future Work

The goal of the research presented in this chapter was to investigate the viability of NEAT with objective search and novelty search for evolutionary RL of colloidal SA systems. This

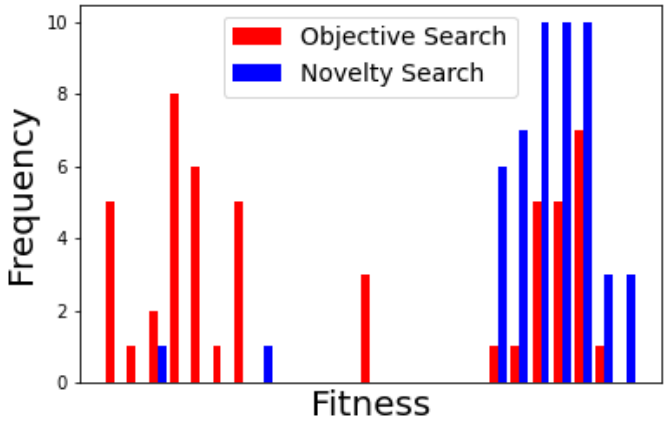


Figure 6.2: Closed-loop control policy comparison. The control policies learned from NEAT with objective search and NEAT with novelty search are each applied to the test system 50 times. Eq. (6.2) is then used to calculate the fitness of the final assembled state (e.g., a higher fitness correlates to a higher-quality crystal. The novelty search control policies yield higher-quality crystals more consistently.

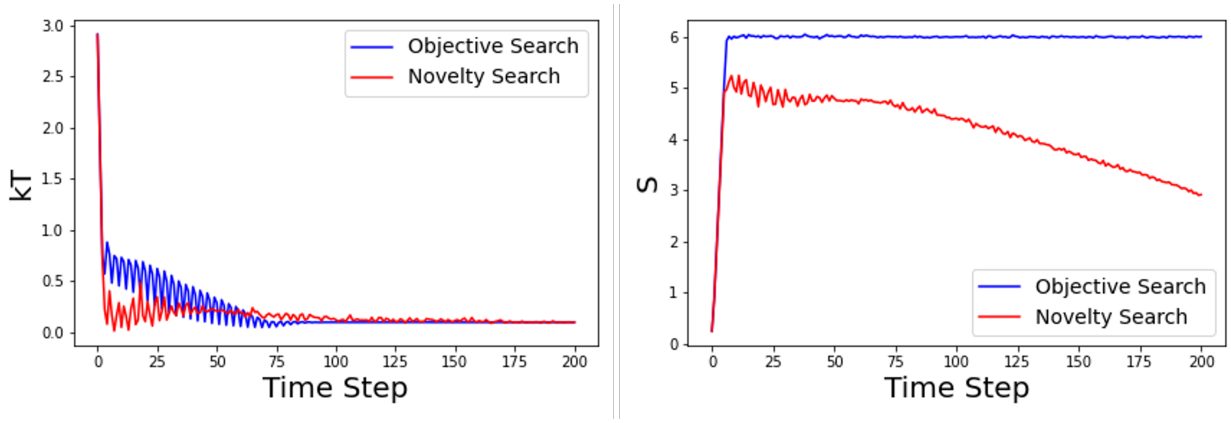


Figure 6.3: Example closed-loop control policy input profiles. The control policies learned from NEAT with objective search and NEAT with novelty search are each applied to the test system 50 times. The average temperature and pressure (e.g., kT and S profiles are plotted versus the time step. Note the fast oscillations in the temperature profiles between time steps 5 and 75 correspond to a series of heating and cooling ramps (which correspond to initiating assembly and disassembly).

chapter used NEAT with objective and novelty search to manipulate the temperature and pressure of the test system to assemble high-quality FCC crystals. Importantly, this chapter provided a proof-of-concept demonstration that learning control policies with very few model parameters can lead to effective control of colloidal SA and that novelty search may in fact lead to improved control performance over objective search.

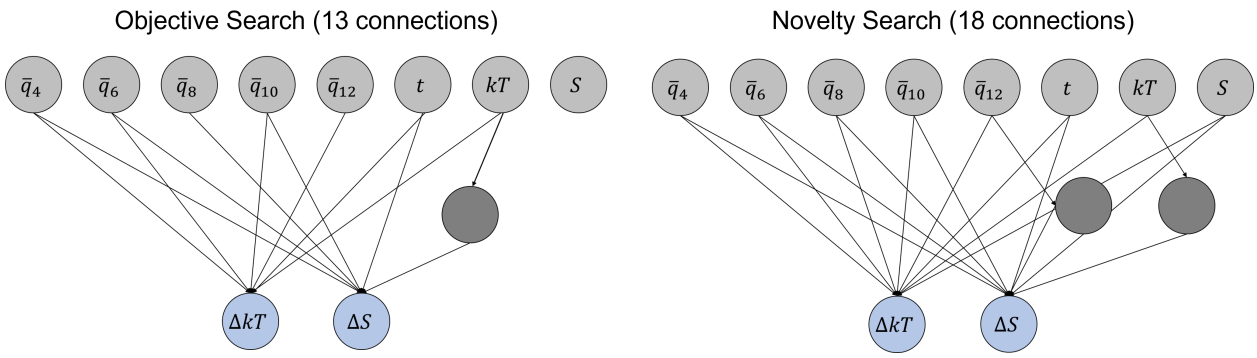


Figure 6.4: Example closed-loop control policy neural network representations. The artificial neural networks that represent the control policies learned from NEAT with objective search and NEAT with novelty search are shown above.The inputs to the control policy are the time step, temperature, pressure, and the 5 averaged, lattice-level order parameter values $[\bar{q}_4, \bar{q}_6, \bar{q}_8, \bar{q}_{10}, \bar{q}_{12}]^\top$. The outputs are temperature and pressure ramp rates, ΔS and ΔkT. The policy learned with novelty search has 5 more node-to-node connections and is thus slightly more complex than the objective search policy. Further note that the objective search policy eliminated the pressure (S) input.

The next immediate step in this project is to run much longer trials to see if (i) more effective open- and closed-loop control policies can be learned and (ii) if novelty search continues to outperform objective search or vice versa. A sensitivity analysis of the various parameters within NEAT (e.g., the genome mutation probability) should also be performed. NEAT with novelty search will then be applied to *in-silico* systems with more complex dynamics (e.g., the systems of DNA-functionalized particles mentioned in Chapters 2-3). It is possible that such complex dynamics may require learning larger model architectures, and that NEAT may need to be abandoned as a result.

The primary objective of future work will be to understand better the role of novelty search in learning control policies for colloidal SA. Future work will investigate topics such as how and whether novelty search and objective search could be combined into one cohesive framework. For example, minimum criteria novelty search [265], which only rewards novel behavior if that novel behavior has some minimum goal-oriented performance, has been shown to learn efficiently control policies of systems with stochastic, nonlinear dynamics. Future work will also explore how novelty search can be used in another evolutionary RL algorithmic frameworks outside of NEAT, as the small architectures that NEAT evolves may in fact be too simple to control many colloidal SA systems. However, applying novelty search outside of NEAT, or outside of a complexifying framework may be challenging, as it has been shown that the complexifying aspect of NEAT principles the novelty search [260]. Frameworks such as HyperNEAT and ES-HyperNEAT claim to evolve more complex architectures, but often require extensive *a-priori* knowledge of the test system [266, 267].

It is further reasonable to consider how physics can be directly incorporated into the learning algorithms. For example, the novelty criteria in Eq. (6.3) is extremely simple and may not be complex enough to reflect the inconsistency between inter-structural distance

and material "value" (i.e., it is important to distinguish perfect and slightly defective lattices while it may not be as critical to distinguish various amorphous and semi-crystalline structures). A more "physics-informed" novelty criteria could reduce the search space and lead to more efficient learning. It is possible that this same physical knowledge could be incorporated into HyperNEAT and ES-HyperNEAT frameworks or could be used to constrain the fitness function, as physics has in fact been used to constrain similar functions in the past [207, 208].

Chapter 7

Conclusions and Perspectives

The primary objective of this thesis was to enable more effective particle design and control of colloidal SA by investigating learning-based strategies for quantifying and classifying colloidal SA system states, learning tractable stochastic dynamical models of colloidal SA dynamics, and learning control policies that dynamically change certain global external actuators. Chapters 2 and 3 focus on a novel framework for quantifying and classifying colloidal SA structure at the single particle level. The key steps of the framework are (i) establishing a high-dimensional, yet precise characterization of local structure, (ii) using a deep autoencoder to translate that high-dimensional representation into a low-dimensional space that is more easily interpretable, and (iii) using unsupervised learning techniques to partition that low-dimensional space to assign physically meaningful classifications of structure. Although previous work followed this same general approach (i.e., dimensionality reduction of a precise characterization followed by partitioning the low-dimensional space), this previous work could not tractably characterize entire colloidal SA trajectories. The methods of Chapters 2-3 allowed us to not only compare and contrast thousands of structures within one continuous low-dimensional space, but also to elucidate entire colloidal SA pathways. This extra information in turn allowed us to glean two key insights from trajectory data that otherwise would have been unattainable: (i) binary colloidal mixtures with small interparticle size disparities (i.e., A- and B-type particle radius ratios of $r_A/r_B = 0.8$ to $r_A/r_B = 0.95$) can promote the SA of defect-free binary colloidal lattices much more effectively than SA systems of identically sized particles and (ii) binary colloidal mixtures of size disparate particles can undergo non-classical nucleation pathways. Although these insights by themselves are useful for designing particles in future colloidal SA systems, the work more broadly serves to show how advanced characterization methods can pave the way for systematic and computationally efficient investigation of the underlying mechanisms of colloidal SA.

Despite the utility of the presented characterization framework, certain aspects of the framework can be viewed as "ad-hoc." For example, a key advantage of the framework is that it avoids the concept of "cut-off radii" among particles and instead uses a geometry-based, fixed number of particles to describe local particle structure in the form of neighborhood graphs. However, this particle number is based on the geometry of FCC and HCP unit

cells and may not be appropriate for more complex structures. Moreover, the framework evaluates these neighborhood graphs using graphlets with 2-5 nodes. Larger graphlets are not used due to computational cost, not due to some physical aspect of colloidal SA. It is possible that a more natural, physics-informed approach to colloidal characterization could increase the precision and generalizability of the characterization.

A promising strategy can be discovering physics-informed strategies for determining particle neighborhoods. For example, it may be reasonable to define a physics-informed cut-off radius (as opposed to the ad-hoc approach described above) based on the estimated distance at which interparticle potentials become zero. Testing the viability of such a framework on *in-silico* systems would in turn be straightforward because the distance at which inter-particle potentials become zero is known. The resulting size of this neighborhood, however, could be too large for the graphlet decomposition methods of chapters 2-3 to evaluate tractably. Graph neural networks, on the other hand, could in principle evaluate such large neighborhoods [268–270] naturally and tractably. Here, colloids themselves could be "nodes," all particles within a neighborhood could be considered "bonded," and their inter-particle distances (i.e., bond lengths) could be considered "edges." In fact, the work in reference [271] successfully uses a similar graph neural network-based strategy to model glassy dynamics. One known challenge with graph neural networks is that they are not necessarily permutation or rotation-invariant [268–270]. As a result, directly incorporating permutation- and rotation-invariant features (such as those in reference [144]) could mitigate such an issue. In summary, an exploration of how physical knowledge, graph-based machine learning, and rotation- and permutation-invariance can be combined provides exciting opportunities for improving colloidal characterization.

Chapter 4 investigates the utility of a model-based feedback control strategy for colloidal SA. The strategy manipulates temperature based on structural state information to create defect-free crystals in an *in-silico* test system of DNA-functionalized particles. Despite the fact that the strategy consistently guides the test system towards a defect-free structure, the time to reach that structure varies significantly run-to-run. One potential way to improve control performance is to account explicitly for the intrinsic stochasticity present in colloidal SA dynamics, which in turn requires learning a tractable stochastic dynamical model. Chapter 5 presents a novel framework for learning tractable stochastic dynamical models along with free energy and diffusion landscapes from stochastic trajectory data. The framework, referred to as stochastic physics-informed neural ordinary differential equations, or SPINODE, accurately recovered stochastic dynamics and energy and diffusion landscapes for a handful of benchmark *in-silico* case studies. SPINODE was shown to be a flexible and scalable framework for learning nonlinear stochastic dynamical models, as SPINODE does not depend on system-specific sampling techniques or extremely small sampling times. SPINODE also tractably learned nonlinear, multivariate, stochastic dynamical models for systems with multiplicative noise [64]. An open challenge with SPINODE is the fact that a large number of repeated state trajectories from identical initial conditions are required to learn these dynamical models accurately. Thus, SPINODE may not be appropriate for experimental colloidal SA, where structural characterization can be imprecise and poten-

tially lead to even larger data requirements. The primary focus of future work should be to learn probability distributions directly from data, especially in scenarios in which one does not have control over initial conditions, number of replica runs, or consistent measurement times. Once these probability distributions are learned, the remaining machinery within SPINODE (e.g., moment matching and gradient descent) can in principle be used to learn stochastic dynamical models efficidently. Variational autoencoders [227–229], generative adversarial networks [209, 230–232], and energy-based models [233–235] can all possibly be used to learn probability distributions under such scenarios and thus should be explored. A successful combination of one of these methods with SPINODE could lead to an efficient "dynamical" characterization of colloidal SA and could enhance analysis from the colloidal state characterization from Chapters 2 and 3. For example, a learned energy landscape could allow for one to not only quantify how structurally defective but also how energetically or kinetically favorable that lattice may be. Moreover, the corresponding dynamical model can predict how a given external actuator (e.g., temperature) could break a system out of that defective state and guide it towards a more highly-ordered state.

Given the challenges with constructing stochastic dynamical models, Chapter 6 explores model-free approaches to feedback control of colloidal SA. The work uses evolutionary RL to learn a neural network representation of a control policy that manipulates temperature and pressure to guide the colloidal SA of a benchmark *in-silico* test system. In particular, this chapter compares the efficacy of "objective search," where progress towards a pre-defined goal is measured and control policies are iteratively updated according to this progress, and "novelty search," where behavioral novelty alone is used to learn control policies. The work demonstrated that evolutionary RL algorithms in principle can be used to guide colloidal SA effectively. However, much more work is required to elucidate properly the roles of objective and novelty search in learning control policies. One promising direction for future work is to incorporate physics into evolutionary RL algorithms to shrink the search space and lead to more efficient learning. For example, novelty search depends on novelty criteria (see Eq. (6.3). The novelty criteria used in Chapter 6 is essentially a k-nearest neighbors algorithm that doesn't account for the fact that it may be more important to distinguish certain structurally similar structures (e.g., a defect-free and nearly defect-free crystal) than to distinguish structures that are fairly structurally dissimilar but not necessarily relevant to colloidal SA (e.g., many amorphous or weakly crystalline structures). Incorporating physical knowledge directly into the novel criteria could thus possibly lead to more efficient learning.

Overall, the research presented in this thesis provides a deeper mechanistic understanding of colloidal SA and contributes to an ever-developing archive of methods that can be used or expanded upon to achieve reproducible colloidal SA. Although each chapter focuses on *in-silico* colloidal SA systems, each of these *in-silico* systems is either a benchmark problem (e.g., a classic Lennard-Jones system from [261] or the well-characterized two-dimensional colloidal SA problem in [59]) or uses experimentally-validated pair potentials (see Chapters 2-4). As such, the insights gained from this research can be useful for informing experiments and future computational studies with high-fidelity models. For example, the insight from Chapter 3 that small size disparity in binary colloidal mixtures can drasti-

cally affect the compositional order of self-assembled lattices could (and should) encourage colloidal SA experimentalists to synthesize such binary colloidal mixtures in their laboratories. More broadly, the findings in thesis are relevant to many systems outside of colloidal SA. For example, Chapter 5 investigated SPINODE for learning stochastic dynamical models of systems with nonlinear dynamics and multiplicative noise. In addition to colloidal SA, Chapter 5 demonstrated SPINODE on systems with Lotka-Volterra and susceptible-infectious-recovered epidemic model dynamics. In fact, a major contribution of Chapter 5 was that SPINODE can be easily generalized to many different system types. Although Chapter 6 could be interpreted as a mere proof-of-concept study for model-free feedback control of colloidal SA, the work can also be considered as an investigation of the viability of objective search and novelty search to learn control policies for deceptive systems with stochastic and nonlinear dynamics. Chapter 4 could similarly be regarded as a proof-of-concept study for model-based feedback control of colloidal SA. However, certain aspects of that work, such as the use of autoencoders to reduce the state dimension in order to create a model that is tractable for online control, can be applicable to a wide range of systems with high-dimensional, nonlinear dynamics. The characterization framework presented in Chapters 2-3 can in principle be applied to any system that can be described with some graph-based structure (e.g., proteins, social media relationships). In fact, the graphlet decomposition method used in Chapters 2-3 was first developed to determine protein function in protein-protein interaction networks [74]. I hope that future members of the Mesbah lab can continue exploring how learning-based strategies for colloidal SA characterization and control can not only lead to the achievement of reproducible colloidal SA but also can lead to insights in a variety of societally impactful systems.

Bibliography

(1) Whitesides, G. M.; Grzybowski, B. *Science* **2002**, *295*, 2418–2421.

(2) Grzelczak, M.; Vermant, J.; Furst, E. M.; Liz-Marzán, L. M. *ACS nano* **2010**, *4*, 3591–3605.

(3) Velev, O. D.; Gupta, S. *Advanced Materials* **2009**, *21*, 1897–1905.

(4) Barnaby, S. N.; Thaner, R. V.; Ross, M. B.; Brown, K. A.; Schatz, G. C.; Mirkin, C. A. *Journal of the American Chemical Society* **2015**, *137*, 13566–13571.

(5) Parviz, B. A.; Ryan, D.; Whitesides, G. M. *IEEE transactions on advanced packaging* **2003**, *26*, 233–241.

(6) Bohr, M. T. *IEEE Transactions on Nanotechnology* **2002**, *1*, 56–62.

(7) Lu, W.; Sastry, A. M. *IEEE transactions on Semiconductor Manufacturing* **2007**, *20*, 421–431.

(8) Mönch, L.; Fowler, J. W.; Dauzère-Pérès, S.; Mason, S. J.; Rose, O. *Journal of scheduling* **2011**, *14*, 583–599.

(9) Pan, C.; Zhou, M.; Qiao, Y.; Wu, N. *IEEE transactions on automation science and engineering* **2017**, *15*, 586–601.

(10) Iwai, H. In *IEDM Technical Digest. IEEE International Electron Devices Meeting, 2004.* 2004, pp 11–16.

(11) Iwai, H.; Kakushima, K.; Wong, H. In *Frontiers In Electronics: (With CD-ROM)*; World Scientific: 2006, pp 43–81.

(12) Whitesides, G. M.; Kriebel, J. K.; Mayers, B. T. In *Nanoscale assembly*; Springer: 2005, pp 217–239.

(13) Juárez, J. J.; Bevan, M. A. *Advanced Functional Materials* **2012**, *22*, 3833–3839.

(14) Paulson, J. A.; Mesbah, A.; Zhu, X.; Molaro, M. C.; Braatz, R. D. *Journal of Process Control* **2015**, *27*, 38–49.

(15) Joannopoulos, J. D.; Villeneuve, P. R.; Fan, S. *Nature* **1997**, *386*, 143–149.

(16) Zhang, X.; Liu, Z. *Nature materials* **2008**, *7*, 435–441.

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

(18) Tong, H.; Ouyang, S.; Bi, Y.; Umezawa, N.; Oshikiri, M.; Ye, J. *Advanced materials* **2012**, *24*, 229–251.

(19) Whitaker, J. C., *Microelectronics*; Crc Press: 2018.

(20) Gillespie, D. T. et al. *Annual review of physical chemistry* **2007**, *58*, 35–55.

(21) Gillespie, D. T.; Hellander, A.; Petzold, L. R. *The Journal of chemical physics* **2013**, *138*, 05B201_1.

(22) Furst, E. M. *Soft Matter* **2013**, *9*, 9039–9045.

(23) Liddle, J. A.; Gallatin, G. M. *ACS nano* **2016**, *10*, 2995–3014.

(24) Beltran-Villegas, D. J.; Bevan, M. A. *Soft Matter* **2011**, *7*, 3280–3285.

(25) Van Duijneveldt, J. S.; Frenkel, D. *The Journal of chemical physics* **1992**, *96*, 4655–4668.

(26) Wolde, P. R. t.; Frenkel, D. *Science* **1997**, *277*, 1975–1978.

(27) Rein ten Wolde, P.; Ruiz-Montero, M. J.; Frenkel, D. *The Journal of chemical physics* **1996**, *104*, 9932–9947.

(28) Beltran-Villegas, D. J.; Sehgal, R. M.; Maroudas, D.; Ford, D. M.; Bevan, M. A. *The Journal of chemical physics* **2011**, *135*, 154506.

(29) Hänggi, P.; Talkner, P.; Borkovec, M. *Reviews of modern physics* **1990**, *62*, 251.

(30) Risken, H. In *The Fokker-Planck Equation*; Springer: 1996, pp 63–95.

(31) Beltran-Villegas, D. J.; Sehgal, R. M.; Maroudas, D.; Ford, D. M.; Bevan, M. A. *The Journal of chemical physics* **2012**, *137*, 134901.

(32) Israelachvili, J. N.; Mitchell, D. J.; Ninham, B. W. *Biochimica et Biophysica Acta (BBA)-Biomembranes* **1977**, *470*, 185–201.

(33) Zhang, S. *Nature biotechnology* **2003**, *21*, 1171–1178.

(34) Seto, C. T.; Whitesides, G. M. *Journal of the American Chemical Society* **1993**, *115*, 905–916.

(35) Edwards, T. D.; Bevan, M. A. *Langmuir* **2014**, *30*, 10793–10803.

(36) Correa-Duarte, M. A.; Grzelczak, M.; Salgueiriño-Maceira, V.; Giersig, M.; Liz-Marzán, L. M.; Farle, M.; Sierazdki, K.; Diaz, R. *The Journal of Physical Chemistry B* **2005**, *109*, 19060–19063.

(37) Bevan, M. A.; Ford, D. M.; Grover, M. A.; Shapiro, B.; Maroudas, D.; Yang, Y.; Thyagarajan, R.; Tang, X.; Sehgal, R. M. *Journal of Process Control* **2015**, *27*, 64–75.

(38) Tang, X.; Rupp, B.; Yang, Y.; Edwards, T. D.; Grover, M. A.; Bevan, M. A. *ACS nano* **2016**, *10*, 6791–6798.

(39) Zhang, R.; Schweizer, K. S. *The Journal of chemical physics* **2010**, *133*, 104902.

(40) Zaccarelli, E.; Valeriani, C.; Sanz, E.; Poon, W.; Cates, M.; Pusey, P. *Physical review letters* **2009**, *103*, 135704.

(41) Di Michele, L.; Varrato, F.; Kotar, J.; Nathan, S. H.; Foffi, G.; Eiser, E. *Nature communications* **2013**, *4*, 1–7.

(42) Klavins, E. *IEEE Control Systems Magazine* **2007**, *27*, 43–56.

(43) Wang, Y.; Wang, Y.; Breed, D. R.; Manoharan, V. N.; Feng, L.; Hollingsworth, A. D.; Weck, M.; Pine, D. J. *Nature* **2012**, *491*, 51–55.

(44) Granick, S.; Jiang, S.; Chen, Q. *Phys. Today* **2009**, *62*, 68–69.

(45) Jiang, S.; Chen, Q.; Tripathy, M.; Luijten, E.; Schweizer, K. S.; Granick, S. *Advanced materials* **2010**, *22*, 1060–1071.

(46) Swan, J. W.; Bauer, J. L.; Liu, Y.; Furst, E. M. *Soft Matter* **2014**, *10*, 1102–1109.

(47) Whitelam, S.; Tamblyn, I. *Physical Review E* **2020**, *101*, 052604.

(48) Solis, E. O.; Barton, P. I.; Stephanopoulos, G. *Industrial & Engineering Chemistry Research* **2010**, *49*, 7728–7745.

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

(50) Ramaswamy, S.; Lakerveld, R.; Barton, P. I.; Stephanopoulos, G. *Industrial & Engineering Chemistry Research* **2015**, *54*, 4371–4384.

(51) Ramaswamy, S.; Barton, P. I.; Stephanopoulos, G. *Industrial & Engineering Chemistry Research* **2015**, *54*, 8520–8532.

(52) Whitelam, S.; Tamblyn, I. *Physical review letters* **2021**, *127*, 018003.

(53) Bakar, M. R. A.; Nagy, Z. K.; Rielly, C. D. *Organic Process Research & Development* **2009**, *13*, 1343–1356.

(54) Jiang, M.; Zhu, X.; Molaro, M. C.; Rasche, M. L.; Zhang, H.; Chadwick, K.; Raimondo, D. M.; Kim, K.-K. K.; Zhou, L.; Zhu, Z., et al. *Industrial & Engineering Chemistry Research* **2014**, *53*, 5325–5336.

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

(56) Klotsa, D.; Jack, R. L. *The Journal of Chemical Physics* **2013**, *138*, 094502.

(57) Lavergne, F. A.; Wendehenne, H.; Bäuerle, T.; Bechinger, C. *Science* **2019**, *364*, 70–74.

(58) Tang, X.; Zhang, J.; Bevan, M. A.; Grover, M. A. *Journal of Process Control* **2017**, *60*, 141–151.

(59) Tang, X.; Xue, Y.; Grover, M. A. In *2013 American Control Conference*, 2013, pp 4228–4233.

(60) Zhang, J.; Yang, J.; Zhang, Y.; Bevan, M. A. *Science advances* **2020**, *6*, eabd6716.

(61) Gillespie, D. T. *Physica A: Statistical Mechanics and its Applications* **1992**, *188*, 404–425.

(62) Reinhart, W. F.; Long, A. W.; Howard, M. P.; Ferguson, A. L.; Panagiotopoulos, A. Z. *Soft Matter* **2017**, *13*, 4733–4745.

(63) Reinhart, W. F.; Panagiotopoulos, A. Z. *Soft matter* **2018**, *14*, 6083–6089.

(64) O'Leary, J.; Paulson, J. A.; Mesbah, A. *Journal of Computational Physics* **2022**, 111466.

(65) Kumar, S.; Rosenberg, J. M.; Bouzida, D.; Swendsen, R. H.; Kollman, P. A. *Journal of computational chemistry* **1992**, *13*, 1011–1021.

(66) Ono, S.; Nakajima, N.; Higo, J.; Nakamura, H. *Chemical physics letters* **1999**, *312*, 247–254.

(67) Babin, V.; Roland, C.; Sagui, C. *The Journal of chemical physics* **2008**, *128*, 134101.

(68) Darve, E.; Pohorille, A. *The Journal of chemical physics* **2001**, *115*, 9169–9183.

(69) Hummer, G. *New Journal of Physics* **2005**, *7*, 34.

(70) Ghysels, A.; Venable, R. M.; Pastor, R. W.; Hummer, G. *Journal of chemical theory and computation* **2017**, *13*, 2962–2976.

(71) Chen, Q.; Whitmer, J. K.; Jiang, S.; Bae, S. C.; Luijten, E.; Granick, S. *Science* **2011**, *331*, 199–202.

(72) Pretti, E.; Zerze, H.; Song, M.; Ding, Y.; Mahynski, N. A.; Hatch, H. W.; Shen, V. K.; Mittal, J. *Soft Matter* **2018**, *14*, 6303–6312.

(73) Song, M.; Ding, Y.; Zerze, H.; Snyder, M. A.; Mittal, J. *Langmuir* **2018**, *34*, 991–998.

(74) Milenković, T.; Pržulj, N. *Cancer informatics* **2008**, *6*, CIN–S680.

(75) Pržulj, N.; Corneil, D. G.; Jurisica, I. *Bioinformatics* **2004**, *20*, 3508–3515.

(76) Pržulj, N. *Bioinformatics* **2007**, *23*, e177–e183.

(77) Hočevar, T.; Demšar, J. *Bioinformatics* **2014**, *30*, 559–565.

(78) Howard, M. P.; Reinhart, W. F.; Sanyal, T.; Shell, M. S.; Nikoubashman, A.; Panagiotopoulos, A. Z. *The Journal of chemical physics* **2018**, *149*, 094901.

(79) O'Leary, J.; Mao, R.; Pretti, E. J.; Paulson, J. A.; Mittal, J.; Mesbah, A. *Soft Matter* **2021**, *17*, 989–999.

(80) Mao, R.; O'Leary, J.; Mesbah, A.; Mittal, J. *JACS Au* **2022**.

(81) Steinhardt, P. J.; Nelson, D. R.; Ronchetti, M. *Physical Review B* **1983**, *28*, 784.

(82) Lechner, W.; Dellago, C. *The Journal of chemical physics* **2008**, *129*, 114707.

(83) Honeycutt, J. D.; Andersen, H. C. *Journal of Physical Chemistry* **1987**, *91*, 4950–4963.

(84) Faken, D.; Jónsson, H. *Comput. Mater. Sci* **1994**, *2*, 279–286.

(85) Larsen, P. M.; Schmidt, S.; Schiøtz, J. *Modelling and Simulation in Materials Science and Engineering* **2016**, *24*, 055007.

(86) Ackland, G.; Jones, A. *Physical Review B* **2006**, *73*, 054104.

(87) Reinhart, W. F.; Panagiotopoulos, A. Z. *Soft matter* **2017**, *13*, 6803–6809.

(88) Long, A. W.; Ferguson, A. L. *The Journal of Physical Chemistry B* **2014**, *118*, 4228–4244.

(89) Long, A. W.; Zhang, J.; Granick, S.; Ferguson, A. L. *Soft Matter* **2015**, *11*, 8141–8153.

(90) Long, A. W.; Ferguson, A. L. *Applied and Computational Harmonic Analysis* **2019**, *47*, 190–211.

(91) Ferguson, A. L.; Panagiotopoulos, A. Z.; Kevrekidis, I. G.; Debenedetti, P. G. *Chemical Physics Letters* **2011**, *509*, 1–11.

(92) Ferguson, A. L. *Journal of Physics: Condensed Matter* **2017**, *30*, 043002.

(93) DeFever, R. S.; Targonski, C.; Hall, S. W.; Smith, M. C.; Sarupria, S. *Chemical science* **2019**, *10*, 7503–7515.

(94) Jadrich, R.; Lindquist, B.; Truskett, T. *The Journal of chemical physics* **2018**, *149*, 194109.

(95) Jadrich, R.; Lindquist, B.; Piñeros, W.; Banerjee, D.; Truskett, T. *The Journal of chemical physics* **2018**, *149*, 194110.

(96) Spellings, M.; Glotzer, S. C. *AIChE Journal* **2018**, *64*, 2198–2206.

(97) Jadrich, R. B.; Lindquist, B. A.; Truskett, T. M. *arXiv preprint arXiv:1706.05405* **2017**.

(98) Kadulkar, S.; Howard, M. P.; Truskett, T. M.; Ganesan, V. *The Journal of Physical Chemistry B* **2021**, *125*, 4838–4849.

(99) Jolliffe, I. T.; Cadima, J. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* **2016**, *374*, 20150202.

(100) Baldi, P. In *Proceedings of ICML workshop on unsupervised and transfer learning*, 2012, pp 37–49.

(101) Wang, Y.; Yao, H.; Zhao, S. *Neurocomputing* **2016**, *184*, 232–242.

(102) Boattini, E.; Dijkstra, M.; Filion, L. *The Journal of chemical physics* **2019**, *151*, 154901.

(103) Pretti, E.; Mao, R.; Mittal, J. *Molecular Simulation* **2019**, *45*, 1203–1210.

(104) Pretti, E.; Zerze, H.; Song, M.; Ding, Y.; Mao, R.; Mittal, J. *Science advances* **2019**, *5*, eaaw5912.

(105) O'Leary, J. Framework for characterizing three-dimensional, binary, colloidal self-assembly system states, `https://github.com/jtoleary/colloid_char`, Online; accessed 1 August 2022.

(106) Stukowski, A. *Modelling and Simulation in Materials Science and Engineering* **2012**, *20*, 045021.

(107) Mahynski, N. A.; Mao, R.; Pretti, E.; Shen, V. K.; Mittal, J. *Soft Matter* **2020**, *16*, 3187–3194.

(108) Parisi, G. I.; Kemker, R.; Part, J. L.; Kanan, C.; Wermter, S. *Neural Networks* **2019**, *113*, 54–71.

(109) Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. *The journal of machine learning research* **2014**, *15*, 1929–1958.

(110) Wager, S.; Wang, S.; Liang, P. S. In *Advances in neural information processing systems*, 2013, pp 351–359.

(111) Krogh, A.; Vedelsby, J. In *Advances in neural information processing systems*, 1995, pp 231–238.

(112) Feurer, M.; Hutter, F. In *Automated Machine Learning*; Springer, Cham: 2019, pp 3–33.

(113) Nwankpa, C.; Ijomah, W.; Gachagan, A.; Marshall, S. *arXiv preprint arXiv:1811.03378* **2018**.

(114) Farina, M.; Nakai, Y.; Shih, D. *Physical Review D* **2020**, *101*, 075021.

(115) Salvador, S.; Chan, P. In *16th IEEE international conference on tools with artificial intelligence*, 2004, pp 576–584.

(116) Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M., et al. In *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, 2016, pp 265–283.

(117) Chollet, F. et al. *ascl* **2018**, ascl–1806.

(118) Yao, J.; Teng, N.; Poh, H.-L.; Tan, C. L. *J. Inf. Sci. Eng.* **1998**, *14*, 843–862.

(119) Scardi, M.; Harding Jr, L. W. *Ecological modelling* **1999**, *120*, 213–223.

(120) Gevrey, M.; Dimopoulos, I.; Lek, S. *Ecological modelling* **2003**, *160*, 249–264.

(121) Olden, J. D.; Joy, M. K.; Death, R. G. *Ecological modelling* **2004**, *178*, 389–397.

(122) Sasirekha, K.; Baby, P. *International Journal of Scientific and Research Publications* **2013**, *83*, 83.

(123) Jain, A. K.; Murty, M. N.; Flynn, P. J. *ACM computing surveys (CSUR)* **1999**, *31*, 264–323.

(124) Stukowski, A. *Modelling and Simulation in Materials Science and Engineering* **2009**, *18*, 015012.

(125) Macfarlane, R. J.; Lee, B.; Jones, M. R.; Harris, N.; Schatz, G. C.; Mirkin, C. A. *science* **2011**, *334*, 204–208.

(126) Casey, M. T.; Scarlett, R. T.; Rogers, W. B.; Jenkins, I.; Sinno, T.; Crocker, J. C. *Nature communications* **2012**, *3*, 1–8.

(127) Scarlett, R. T.; Ung, M. T.; Crocker, J. C.; Sinno, T. *Soft Matter* **2011**, *7*, 1912–1925.

(128) Plimpton, S. *Journal of computational physics* **1995**, *117*, 1–19.

(129) Kawasaki, T.; Tanaka, H. *Proceedings of the National Academy of Sciences* **2010**, *107*, 14036–14041.

(130) Ten Wolde, P. R.; Ruiz-Montero, M. J.; Frenkel, D. *Physical review letters* **1995**, *75*, 2714.

(131) Sun, L.; Lin, H.; Park, D. J.; Bourgeois, M. R.; Ross, M. B.; Ku, J. C.; Schatz, G. C.; Mirkin, C. A. *Nano letters* **2017**, *17*, 2313–2318.

(132) Redl, F. X.; Cho, K.-S.; Murray, C. B.; O'Brien, S. *Nature* **2003**, *423*, 968–971.

(133) Lee, J.-S.; Han, M. S.; Mirkin, C. A. *Angewandte Chemie International Edition* **2007**, *46*, 4093–4096.

(134) Kang, Y.; Ye, X.; Chen, J.; Cai, Y.; Diaz, R. E.; Adzic, R. R.; Stach, E. A.; Murray, C. B. *Journal of the American Chemical Society* **2013**, *135*, 42–45.

(135) Kang, Y.; Ye, X.; Chen, J.; Qi, L.; Diaz, R. E.; Doan-Nguyen, V.; Xing, G.; Kagan, C. R.; Li, J.; Gorte, R. J., et al. *Journal of the American Chemical Society* **2013**, *135*, 1499–1505.

(136) Knorowski, C.; Travesset, A. *Soft Matter* **2012**, *8*, 12053–12059.

(137) Shevchenko, E. V.; Talapin, D. V.; Kotov, N. A.; O'Brien, S.; Murray, C. B. *Nature* **2006**, *439*, 55–59.

(138) Leunissen, M. E.; Christova, C. G.; Hynninen, A.-P.; Royall, C. P.; Campbell, A. I.; Imhof, A.; Dijkstra, M.; Van Roij, R.; Van Blaaderen, A. *Nature* **2005**, *437*, 235–240.

(139) Rogach, A. L. *Angewandte Chemie International Edition* **2004**, *43*, 148–149.

(140) Bodnarchuk, M. I.; Kovalenko, M. V.; Heiss, W.; Talapin, D. V. *Journal of the American Chemical Society* **2010**, *132*, 11967–11977.

(141) Yu, Y.; Bosoy, C. A.; Smilgies, D.-M.; Korgel, B. A. *The journal of physical chemistry letters* **2013**, *4*, 3677–3682.

(142) Ross, M. B.; Ku, J. C.; Blaber, M. G.; Mirkin, C. A.; Schatz, G. C. *Proceedings of the National Academy of Sciences* **2015**, *112*, 10292–10297.

(143) Bartlett, P.; Campbell, A. I. *Physical review letters* **2005**, *95*, 128302.

(144) Reinhart, W. F. *Computational Materials Science* **2021**, *196*, 110511.

(145) Gu, S.; Johnson, J.; Faisal, F. E.; Milenković, T. *Scientific reports* **2018**, *8*, 1–16.

(146) Rossi, R. A.; Ahmed, N. K.; Carranza, A.; Arbour, D.; Rao, A.; Kim, S.; Koh, E. *ACM Transactions on Knowledge Discovery from Data (TKDD)* **2020**, *15*, 1–43.

(147) Bommineni, P. K.; Klement, M.; Engel, M. *Physical Review Letters* **2020**, *124*, 218003.

(148) Eldridge, M.; Madden, P.; Frenkel, D. *Nature* **1993**, *365*, 35–37.

(149) Tkachenko, A. V. *Proceedings of the National Academy of Sciences* **2016**, *113*, 10269–10274.

(150) Lee, G. H.; Kim, J. B.; Choi, T. M.; Lee, J. M.; Kim, S.-H. *Small* **2019**, *15*, 1804548.

(151) Auer, S.; Frenkel, D. *Nature* **2001**, *413*, 711–713.

(152) Debenedetti, P. G., *Metastable liquids*; Princeton university press: 2021.

(153) Pruppacher, H. R.; Klett, J. D., *Microphysics of Clouds and Precipitation: Reprinted 1980*; Springer Science & Business Media: 2012.

(154) Vekilov, P. G. *Nanoscale* **2010**, *2*, 2346–2357.

(155) De Yoreo, J. *Nature materials* **2013**, *12*, 284–285.

(156) Myerson, A. S.; Trout, B. L. *Science* **2013**, *341*, 855–856.

(157) Pan, W.; Kolomeisky, A. B.; Vekilov, P. G. *The Journal of chemical physics* **2005**, *122*, 174905.

(158) Dai, W.; Kumar, S. K.; Starr, F. W. *Soft Matter* **2010**, *6*, 6130–6135.

(159) Wang, Y.; Wang, Y.; Zheng, X.; Ducrot, E.; Yodh, J. S.; Weck, M.; Pine, D. J. *Nature communications* **2015**, *6*, 1–8.

(160) Debenedetti, P. G.; Truskett, T. M.; Lewis, C. P.; Stillinger, F. H. **2001**.

(161) Jiang, H.; Debenedetti, P. G.; Panagiotopoulos, A. Z. *The Journal of chemical physics* **2019**, *150*, 124502.

(162) Trudu, F.; Donadio, D.; Parrinello, M. *Physical review letters* **2006**, *97*, 105701.

(163) Macfarlane, R. J.; Lee, B.; Hill, H. D.; Senesi, A. J.; Seifert, S.; Mirkin, C. A. *Proceedings of the National Academy of Sciences* **2009**, *106*, 10493–10498.

(164) Mahynski, N. A.; Zerze, H.; Hatch, H. W.; Shen, V. K.; Mittal, J. *Soft Matter* **2017**, *13*, 5397–5408.

(165) Grubmüller, H.; Heller, H.; Windemuth, A.; Schulten, K. *Molecular Simulation* **1991**, *6*, 121–142.

(166) Schmidhuber, J. *Neural networks* **2015**, *61*, 85–117.

(167) Garcia, C. E.; Prett, D. M.; Morari, M. *Automatica* **1989**, *25*, 335–348.

(168) Rawlings, J. B.; Mayne, D. Q.; Diehl, M., *Model predictive control: theory, computation, and design*; Nob Hill Publishing Madison, WI: 2017; Vol. 2.

(169) Biegler, L. T.; Zavala, V. M. *Computers & Chemical Engineering* **2009**, *33*, 575–582.

(170) Rawlings, J. B. *IEEE control systems magazine* **2000**, *20*, 38–52.

(171) Honerkamp, J., *Stochastic dynamical systems: concepts, numerical methods, data analysis*; John Wiley & Sons: 1993.

(172) Van Kampen, N. G. *Physics reports* **1976**, *24*, 171–228.

(173) Van Kampen, N. G., *Stochastic processes in physics and chemistry*; Elsevier: 1992; Vol. 1.

(174) Volpe, G.; Wehr, J. *Reports on Progress in Physics* **2016**, *79*, 053901.

(175) Arnold, L. *New York* **1974**.

(176) Schuster, S.; Marhl, M.; Höfer, T. *European Journal of Biochemistry* **2002**, *269*, 1333–1355.

(177) Pereyra, M.; Schniter, P.; Chouzenoux, E.; Pesquet, J.-C.; Tourneret, J.-Y.; Hero, A. O.; McLaughlin, S. *IEEE Journal of Selected Topics in Signal Processing* **2015**, *10*, 224–241.

(178) Young, P. C. *Nonlinear and nonstationary signal processing* **2000**, 74–114.

(179) Singer, K. *Journal of the Royal Statistical Society: Series B (Methodological)* **1953**, *15*, 92–106.

(180) Penrose, O. *Analysis and Stochastics of Growth Processes and Interface Models* **2008**, 265.

(181) Zhu, C.; Yin, G. *Journal of Mathematical Analysis and Applications* **2009**, *357*, 154–170.

(182) Nguyen, D. H.; Yin, G. *Journal of differential equations* **2017**, *262*, 1192–1225.

(183) Bain, C. *Journal of Epidemiology and Community Health* **1990**, *44*, 254.

(184) Korobeinikov, A.; Maini, P. K. *Mathematical medicine and biology: a journal of the IMA* **2005**, *22*, 113–128.

(185) Friedrich, R.; Siegert, S.; Peinke, J.; Siefert, M.; Lindemann, M.; Raethjen, J.; Deuschl, G.; Pfister, G., et al. *Physics Letters A* **2000**, *271*, 217–222.

(186) Siegert, S.; Friedrich, R.; Peinke, J. *Physics Letters A* **1998**, *243*, 275–280.

(187) Friedrich, R.; Renner, C.; Siefert, M.; Peinke, J. *Physical review letters* **2002**, *89*, 149401.

(188) Ragwitz, M.; Kantz, H. *Physical Review Letters* **2001**, *87*, 254501.

(189) Lamouroux, D.; Lehnertz, K. *Physics Letters A* **2009**, *373*, 3507–3512.

(190) Gottschall, J.; Peinke, J. *New Journal of Physics* **2008**, *10*, 083034.

(191) Kleinhans, D.; Friedrich, R.; Nawroth, A.; Peinke, J. *Physics Letters A* **2005**, *346*, 42–46.

(192) Gradišek, J.; Siegert, S.; Friedrich, R.; Grabec, I. *Physical Review E* **2000**, *62*, 3146.

(193) Hegger, R.; Stock, G. *The Journal of chemical physics* **2009**, *130*, 034106.

(194) Prusseit, J.; Lehnertz, K. *Physical Review E* **2008**, *77*, 041914.

(195) Van Mourik, A. M.; Daffertshofer, A.; Beek, P. J. *Physics Letters A* **2006**, *351*, 13–17.

(196) Kopelevich, D. I.; Panagiotopoulos, A. Z.; Kevrekidis, I. G. *The Journal of chemical physics* **2005**, *122*, 044908.

(197) Woolf, T. B.; Roux, B. *Proceedings of the National Academy of Sciences* **1994**, *91*, 11631–11635.

(198) Mittal, J.; Truskett, T. M.; Errington, J. R.; Hummer, G. *Physical review letters* **2008**, *100*, 145901.

(199) Mittal, J.; Hummer, G. *The Journal of Chemical Physics* **2012**, *137*, 034110.

(200) Karimi, H.; McAuley, K. B. *Industrial & Engineering Chemistry Research* **2018**, *57*, 8946–8961.

(201) Bicout, D.; Szabo, A. *The Journal of chemical physics* **1998**, *109*, 2325–2338.

(202) Shrestha, A.; Mahmood, A. *IEEE Access* **2019**, *7*, 53040–53065.

(203) Zhang, R.; Li, W.; Mo, T. *arXiv preprint arXiv:1804.01653* **2018**.

(204) Fawaz, H. I.; Forestier, G.; Weber, J.; Idoumghar, L.; Muller, P.-A. *Data mining and knowledge discovery* **2019**, *33*, 917–963.

(205) Chen, R. T.; Rubanova, Y.; Bettencourt, J.; Duvenaud, D. K. *Advances in neural information processing systems* **2018**, *31*.

(206) Rackauckas, C.; Ma, Y.; Martensen, J.; Warner, C.; Zubov, K.; Supekar, R.; Skinner, D.; Ramadhan, A.; Edelman, A. *arXiv preprint arXiv:2001.04385* **2020**.

(207) Raissi, M.; Perdikaris, P.; Karniadakis, G. E. *Journal of Computational Physics* **2019**, *378*, 686–707.

(208) Raissi, M.; Yazdani, A.; Karniadakis, G. E. *Science* **2020**, *367*, 1026–1030.

(209) Yang, Y.; Perdikaris, P. *Journal of Computational Physics* **2019**, *394*, 136–152.

(210) Yang, L.; Meng, X.; Karniadakis, G. E. *Journal of Computational Physics* **2021**, *425*, 109913.

(211) Zhang, D.; Lu, L.; Guo, L.; Karniadakis, G. E. *Journal of Computational Physics* **2019**, *397*, 108850.

(212) Hall, A. R. et al., *Generalized method of moments*; Oxford university press: 2005.

(213) Mátyás, L.; Gourieroux, C.; Phillips, P. C., et al., *Generalized method of moments estimation*; Cambridge University Press: 1999; Vol. 5.

(214) Wooldridge, J. M. *Journal of Economic perspectives* **2001**, *15*, 87–100.

(215) Wan, E. A.; Van Der Merwe, R. In *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373)*, 2000, pp 153–158.

(216) Julier, S. J.; Uhlmann, J. K. In *Signal processing, sensor fusion, and target recognition VI*, 1997; Vol. 3068, pp 182–193.

(217) Sarkka, S. *IEEE Transactions on automatic control* **2007**, *52*, 1631–1641.

(218) Pontryagin, L. S., *Mathematical theory of optimal processes*; CRC press: 1987.

(219) Xiong, J.; Li, X.; Wang, H. *Mathematical Biosciences* **2019**, *1*, 1–k1k2.

(220) Cai, Y.; Kang, Y.; Wang, W. *Applied Mathematics and Computation* **2017**, *305*, 221–240.

(221) O'Leary, J. Stochastic Physics-Informed Neural Ordinary Differential Equations, `https://github.com/jtoleary/SPINODE`, Online; accessed 1 May 2022, 2022.

(222) Peeters, B.; De Roeck, G. *J. Dyn. Sys., Meas., Control* **2001**, *123*, 659–667.

(223) Reynders, E. *Archives of Computational Methods in Engineering* **2012**, *19*, 51–124.

(224) Hong, X.; Mitchell, R. J.; Chen, S.; Harris, C. J.; Li, K.; Irwin, G. W. *International journal of systems science* **2008**, *39*, 925–946.

(225) Schoukens, J.; Ljung, L. *IEEE Control Systems Magazine* **2019**, *39*, 28–99.

(226) Bauer, I.; Bock, H. G.; Körkel, S.; Schlöder, J. P. *Journal of Computational and Applied mathematics* **2000**, *120*, 1–25.

(227) Kingma, D. P.; Welling, M. *arXiv preprint arXiv:1312.6114* **2013**.

(228) An, J.; Cho, S. *Special Lecture on IE* **2015**, *2*, 1–18.

(229) Doersch, C. *arXiv preprint arXiv:1606.05908* **2016**.

(230) Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. *Advances in neural information processing systems* **2014**, *27*.

(231) Zoufal, C.; Lucchi, A.; Woerner, S. *npj Quantum Information* **2019**, *5*, 1–9.

(232) Mescheder, L.; Nowozin, S.; Geiger, A. In *International Conference on Machine Learning*, 2017, pp 2391–2400.

(233) LeCun, Y.; Chopra, S.; Hadsell, R.; Ranzato, M.; Huang, F. *Predicting structured data* **2006**, *1*.

(234) Kim, T.; Bengio, Y. *arXiv preprint arXiv:1606.03439* **2016**.

(235) Gustafsson, F. K.; Danelljan, M.; Bhat, G.; Schön, T. B. In *European Conference on Computer Vision*, 2020, pp 325–343.

(236) Paulson, J. A.; Martin-Casas, M.; Mesbah, A. *Industrial & Engineering Chemistry Research* **2017**, *56*, 9593–9605.

(237) Ponomareva, K.; Date, P.; Wang, Z. *Proceedings of Mathematical Theory of Networks and Systems (MTNS 2010), Budapest* **2010**.

(238) Ebeigbe, D.; Berry, T.; Norton, M. M.; Whalen, A. J.; Simon, D.; Sauer, T.; Schiff, S. J. *arXiv preprint arXiv:2104.01958* **2021**.

(239) Paszke, A. et al. In *Advances in Neural Information Processing Systems 32*; Curran Associates, Inc.: 2019, pp 8024–8035.

(240) Joyce, J. M. In *International encyclopedia of statistical science*; Springer: 2011, pp 720–722.

(241) Chen, Y.-C. *Biostatistics & Epidemiology* **2017**, *1*, 161–187.

(242) Sancho, J. M.; San Miguel, M.; Katz, S.; Gunton, J. *Physical Review A* **1982**, *26*, 1589.

(243) Kloeden, P. E.; Platen, E. In *Numerical Solution of Stochastic Differential Equations*; Springer: 1992, pp 103–160.

(244) Hethcote, H. W. *SIAM review* **2000**, *42*, 599–653.

(245) Capasso, V.; Serio, G. *Mathematical biosciences* **1978**, *42*, 43–61.

(246) Liu, W.-m.; Levin, S. A.; Iwasa, Y. *Journal of mathematical biology* **1986**, *23*, 187–204.

(247) Yuan, S.; Li, B. *Discrete Dynamics in Nature and Society* **2009**, *2009*.

(248) Hinton, G.; Srivastava, N.; Swersky, K. *Cited on* **2012**, *14*, 2.

(249) Snoek, J.; Larochelle, H.; Adams, R. P. *Advances in neural information processing systems* **2012**, *25*.

(250) Whiteson, S. *Reinforcement learning* **2012**, 325–355.

(251) Moriarty, D. E.; Schultz, A. C.; Grefenstette, J. J. *Journal of Artificial Intelligence Research* **1999**, *11*, 241–276.

(252) Sutton, R. S.; Barto, A. G., *Reinforcement learning: An introduction*; MIT press: 2018.

(253) Li, Y. *arXiv preprint arXiv:1701.07274* **2017**.

(254) Dayan, P.; Niv, Y. *Current opinion in neurobiology* **2008**, *18*, 185–196.

(255) Whitelam, S.; Selin, V.; Park, S.-W.; Tamblyn, I. *Nature communications* **2021**, *12*, 1–10.

(256) Khadka, S.; Tumer, K. *Advances in Neural Information Processing Systems* **2018**, *31*.

(257) Lehman, J.; Stanley, K. O., et al. In *ALIFE*, 2008, pp 329–336.

(258) Lehman, J.; Stanley, K. O. *Evolutionary computation* **2011**, *19*, 189–223.

(259) Stanley, K. O.; Miikkulainen, R. *Evolutionary computation* **2002**, *10*, 99–127.

(260) Lehman, J. **2012**.

(261) Anderson, J. A.; Glaser, J.; Glotzer, S. C. *Computational Materials Science* **2020**, *173*, 109363.

(262) Grünwald, P. D.; Myung, I. J.; Pitt, M. A., *Advances in minimum description length: Theory and applications*; MIT press: 2005.

(263) Kramer, O. In *Dimensionality reduction with unsupervised nearest neighbors*; Springer: 2013, pp 13–23.

(264) McIntyre, A.; Kallada, M.; Miguel, C. G.; Feher de Silva, C.; Netto, M. L. neat-python.

(265) Lehman, J.; Stanley, K. O. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, 2010, pp 103–110.

(266) D'Ambrosio, D. B.; Gauci, J.; Stanley, K. O. *Growing adaptive machines* **2014**, 159–185.

(267) Risi, S.; Stanley, K. O. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, 2011, pp 1539–1546.

(268) Gupta, A.; Matta, P.; Pant, B. *Materials Today: Proceedings* **2021**, *46*, 10927–10932.

(269) Zhou, J.; Cui, G.; Hu, S.; Zhang, Z.; Yang, C.; Liu, Z.; Wang, L.; Li, C.; Sun, M. *AI Open* **2020**, *1*, 57–81.

(270) Asif, N. A.; Sarker, Y.; Chakrabortty, R. K.; Ryan, M. J.; Ahamed, M. H.; Saha, D. K.; Badal, F. R.; Das, S. K.; Ali, M. F.; Moyeen, S. I., et al. *IEEE Access* **2021**, *9*, 60588–60606.

(271) Bapst, V.; Keck, T.; Grabska-Barwińska, A.; Donner, C.; Cubuk, E. D.; Schoenholz, S. S.; Obika, A.; Nelson, A. W.; Back, T.; Hassabis, D., et al. *Nature Physics* **2020**, *16*, 448–454.