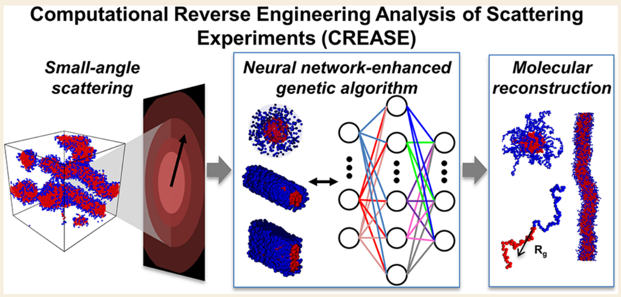
Machine Learning Enhanced Computational Reverse Engineering Analysis for Scattering Experiments (CREASE) to Determine Structures in Amphiphilic Polymer Solutions

Michiel G. Wessels and Arthi Jayaraman*

Cite This: *ACS Polym. Au* 2021, 1, 153−164 Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information


Computational Reverse Engineering Analysis of Scattering Experiments (CREASE)
Small-angle scattering
Neural network-enhanced genetic algorithm
Molecular reconstruction
Rg

ABSTRACT: In this article, we present a machine learning enhancement for our recently developed "Computational Reverse Engineering Analysis for Scattering Experiments" (CREASE) method to accelerate analysis of results from small angle scattering (SAS) experiments on polymer materials. We demonstrate this novel artificial neural network (NN) enhanced CREASE approach for analyzing scattering results from amphiphilic polymer solutions that can be easily extended and applied for scattering experiments on other polymer and soft matter systems. We had originally developed CREASE to analyze SAS results [i.e., intensity profiles, $I(q)$ vs q] of amphiphilic polymer solutions exhibiting unconventional assembled structures and/or novel polymer chemistries for which traditional fits using off-the-shelf analytical models would be too approximate/inapplicable. In this paper, we demonstrate that the NN-enhancement to the genetic algorithm (GA) step in the CREASE approach improves the speed and, in some cases, the accuracy of the GA step in determining the dimensions of the complex assembled structures for a given experimental scattering profile.

KEYWORDS: *Small angle scattering, computational analysis, CREASE, polymers, assembly, structure*

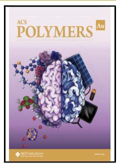
1. INTRODUCTION

Characterization of structures formed in complex solutions containing homopolymers, block copolymers, polymers blended with nanoparticles, or macromolecular conjugates (e.g., protein−polymer conjugates) is key to developing soft materials for applications in healthcare, energy, electronics, and many other fields (see review articles[1−14]). Past studies have shown that one can control the observed nanostructures through tailored polymer design (e.g., choice of chemistries, polymer architecture, and sequence) and processing conditions (e.g., polymer concentration, solvent quality, and temperature).[7,9,15−21] In all these studies, unambiguous characterization of the shape and size of the equilibrium or kinetically trapped structures is a critical step toward building an understanding of the factors that drive and tune these structures.

Methods commonly used for direct visual characterization of nanostructures are scanning electron microscopy (SEM), transmission electron microscopy (TEM), cryogenic TEM (cryo-TEM),[22,23] and atomic force microscopy (AFM).[24,25] These direct imaging methods can have some limitations due to their two-dimensional output for a three-dimensional structure and their inability to resolve small length scales (Å to 10 nm) relevant to polymer materials. Additionally, these microscopy methods require sample preparation prior to imaging (e.g., depositing sample on to an imaging plate) which can alter the environment around the molecules and the solution structure.

Another technique used to analyze structures in polymers and soft materials is small-angle X-ray or neutron scattering (SAXS or SANS, respectively).[26−28] These techniques can analyze polymer solution or melt structures over a broad range of length scales that span Å to μm. Output from SANS or SAXS is usually the scattered intensity, $I(q)$, as a function of the wavevector q, which is a function of the wavelength of incoming beam and the incidence angle. Interpreting these SANS and SAXS intensity profiles (i.e., $I(q)$ versus q), however, requires nontrivial procedures[4,29−31] including fitting the small angle scattering (SAS) profiles with analytical models to extract parameters that describe structural information (e.g., dimensions of the domains, average radius of gyration, etc.). The fitting procedure is facilitated with software packages that have user-friendly interfaces such as ATSAS,[32] McSAS,[33] SASfit,[34] Irena,[35] and SASVIEW.[36] The software packages contain libraries of analytical models that are applicable to many conventional/canonical shapes that have been observed in polymer solutions.

Received: May 28, 2021
Published: July 23, 2021



As an example pertaining to polymer solutions, there are many analytical models developed for spherical and cylindrical micelles with varying features of the core and corona such as sizes and dispersity.[37−40] These models were developed based on assumptions of the solvophilic block chain configurations in the micelle corona (e.g., semiflexible[40] or Gaussian[37,40]). However, with the increasing number of new polymer chemistries with previously unseen chain conformations and/or unconventional/novel assembled structures, these available analytical models and their assumptions become too approximate or even invalid. For example, in recent work[41] we found that the analytical spherical core−corona micelle model failed to capture the correct micelle size and aggregation number which we attributed to the model's assumption of Gaussian chain conformations in the micelle corona that was likely inapplicable for the new polymer chemistry in that study. There are also studies (e.g., refs 42 and 43) where the assembled structures are either kinetically trapped using solvent processing or are dynamic and observed during structural transition from one state to another (e.g., refs 44 and 45); in such cases these structures may not have a corresponding appropriate analytical model.

To avoid these limitations of fitting $I(q)$ versus q with analytical models, one could consider computational methods like reverse Monte Carlo (RMC)[46] and Empirical Potential Structure Refinement (EPSR)[47] that have been used to determine the structure in simple liquids and glasses. These methods manipulate the position and/or interactions of the molecules in the system toward improving the match between the calculated and experimental scattering intensities. In the context of polymers forming assembled structures with large domain sizes (in solutions or melts), these methods require significant modification (e.g., coarse-graining, sophisticated chain moves)[48,49] and their application may still be limited by their low computational efficiency.

To overcome the above limitations in analyzing SAXS or SANS profiles from amphiphilic polymer solutions that exhibit unconventional assembled structures and/or novel polymer chemistries, we recently developed Computational Reverse Engineering Analysis for Scattering Experiments (CREASE).[41,50] CREASE comprises two steps: the first step involves a genetic algorithm[51,52] (GA) which is a robust global optimization method to determine the relevant dimensions of the structure whose computed scattering matches the input experimental scattering; the second step involves molecular dynamics (MD) simulations to reconstruct molecular details in the GA's optimized structure and determine the spatial distribution of monomers as well as the dispersity in chain conformations within the structure, which is not always obtained even with fits to good analytical models. CREASE's GA step has been successfully validated and used to analyze both *in silico* and *in vitro* scattering profiles from solutions with spherical and fibrillar micelles assembled using new polymer chemistries and architectures.[41,50,53] CREASE's GA step has advantages over other computational analysis methods like RMC and EPSR in that the macromolecular details and molecular interactions of the system are not needed in the GA step that only uses scatterers and not polymer chains/amphiphilic macromolecules, a representation reserved for the molecular reconstruction step. By only working with scatterers, the computational efficiency of the GA step within CREASE is significantly increased and allows for efficient use of machine learning by using the many structures generated in previous generations to inform future generations. In this paper, we present a new machine learning-based enhancement for the GA step within the CREASE framework to improve both the speed and accuracy of the GA step.

In CREASE, the GA step outputs the dimensions of the "best" structure whose computed scattering intensity profile matches the input experimental scattering intensity profile. The GA optimizes this match, quantified by fitness, (e.g., high fitness indicates a close match between the experimental and computed scattering intensity profiles) by considering generations of structures (i.e., candidates) that either are carried over due to high fitness or are mutated/changed by genetic operators. The speed of the entire GA step within CREASE is dependent on the speed of calculating the computed scattering intensity profiles. Because of this computed scattering calculation being based on distances for all pairs of scatterers in each structure, the computational requirement for this calculation scales with the number of scatterers squared. To speed up this slow computed scattering profile calculation, in this paper we introduce an artificial neural network-based enhancement. This neural network-enhanced GA (in short, NN-en GA) is partly inspired by other (unrelated to CREASE) studies that used similar ideas of NN enhanced GA for soft materials design optimization.[54−60] In our implementation within CREASE, the NN-en GA uses the information from each generation of the GA step to train a neural network (NN) that can predict the computed scattering intensity profile directly from the structure's dimensions without doing the computationally intensive pairwise calculation for the scatterers. This direct prediction of the computed scattering intensity profile significantly lowers the computational time compared to the computed scattering intensity profile determination using pairwise distance calculation for all the pairs of scatterers. Furthermore, this process allows the NN-en GA to learn from all of the many structures that have been considered in previous generations and improve the accuracy of the output structure prediction (i.e., performance) of the GA. With multiple examples of applications of this NN-en GA method to different micelle structures, in this paper we demonstrate that the NN-en GA improves the accuracy and/or computational speed of the original GA.

The machine learning-enhanced CREASE method described in this paper can be easily extended and applied to determine structures in other soft matter systems from their SAS profiles, in a computationally efficient manner. This improved computational efficiency and the demonstrated ability to "learn" the relationship between $I(q)$ versus q and the structural features that give rise to those intensity profiles should serve as a solution for fast analysis of high throughput scattering experiments as well as to characterize scattering profiles from kinetically trapped nonequilibrium structures for which standard fitting to conventional analytical models would be inaccurate/too approximate.

The rest of the paper is organized as follows. In Section 2, we briefly overview the CREASE methodology, describe the GA step of CREASE, and demonstrate the optimization and implementation of the NN in conjunction with the GA step. In Section 3, we present our results and discussion of the performance in analyzing scattering intensities of cylindrical, elliptically cylindrical, and fibrillar micelle *in silico* and *in vitro* structures of the machine learning enhanced method as compared to the original GA without any machine learning. We conclude with an overview of the paper in Section 4 and potential future directions.

2. METHODS

2.A. Overview of CREASE Method

In this paper, we integrate machine learning into our previously developed CREASE algorithm[41,50] to improve the algorithm's computational speed and efficiency, measured in terms of the computational time needed to achieve the output. In this section, we provide a brief overview of CREASE, Figure 1, motivating the need for the new machine learning enhanced genetic algorithm (GA) step of CREASE. We direct the reader to our previous publications on CREASE[41,50,53] where we presented all of the necessary details of CREASE, its validation, and application to analyze scattering profiles obtained from solutions containing spherical, cylindrical, elliptical cylindrical, and fibrillar micelles.

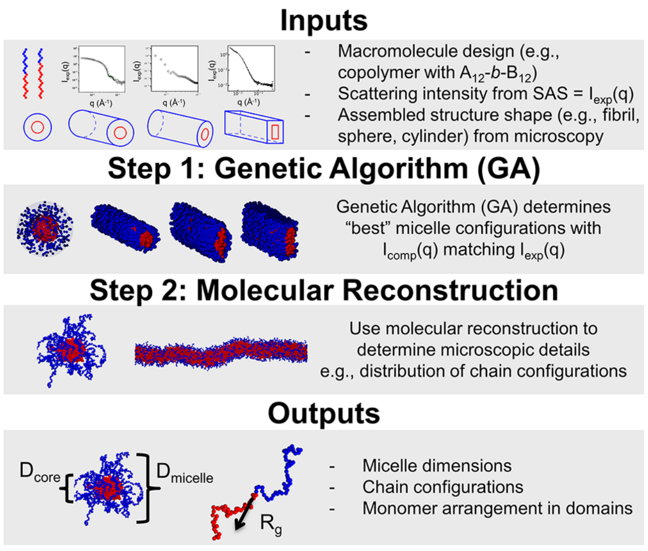

Inputs
q (Å⁻¹)
- Macromolecule design (e.g., copolymer with A12-b-B12)
- Scattering intensity from SAS = Iexp(q)
- Assembled structure shape (e.g., fibril, sphere, cylinder) from microscopy
Step 1: Genetic Algorithm (GA)
Genetic Algorithm (GA) determines "best" micelle configurations with Icomp(q) matching Iexp(q)
Step 2: Molecular Reconstruction
Use molecular reconstruction to determine microscopic details e.g., distribution of chain configurations
Outputs
Dcore
Dmicelle
Rg
- Micelle dimensions
- Chain configurations
- Monomer arrangement in domains

Figure 1. Schematic flow diagram of CREASE.

The inputs to CREASE are the experimental scattering intensity profiles, $I_{\text{exp}}(q)$, as a function of the wavevector q, the design of the amphiphilic polymers (e.g., the chemistry of the solvophilic "A" and solvophobic "B" repeat units and relative ratio of the two chemistries in the polymer), and an estimate of the shape of the assembled structure (e.g., spherical, cylindrical, fibrillar, vesicle, etc.) from another technique. The assembled structure shape can be based on independent measurements using other microscopy techniques or the shape/dimensionality could be estimated from the Porod exponent of the scattering data.[61]

The input information is taken into the Genetic Algorithm (GA) step of the CREASE (Figure 1) that outputs the macroscopic dimensions of the micelle by optimizing to the "best" configuration (examples of which are shown in Figure S1) whose computed scattering intensity, $I_{\text{comp}}(q)$, matches the input $I_{\text{exp}}(q)$. The GA step starts with an initial population containing N random configurations ("individuals"), where the configuration stores "genetic" information about the macroscopic dimensions relevant to the micelle shape specified in the input. The user selects the value of N to balance the computational cost (e.g., real time used by the computer doing the calculations for all individuals in one generation) and population diversity (e.g., individuals varying in relevant dimensions for the given micelle shape). The computational cost associated with the $I_{\text{comp}}(q)$ calculation increases with the number of individuals, N. However, a large value of N allows for increased population diversity which is good for optimization. Thus, the optimal value of N is dependent on the user and their available computational resources and desired accuracy of the GA optimization, as described in Section 2.B.

As stated above, each configuration holds the genetic information about the dimensions relevant to the micelle shape. Supporting Information Figure S1 describes these dimensions for a few assembled structure shapes. For cylindrical micelles, these dimensions include the micelle core diameter, corona width/thickness, and the micelle length. For elliptical cylinders and fibrils, these dimensions include the core cross-sectional width (smallest dimension), the ratio of the width to the height of the core cross-section (P, which is equal to or greater than 1), the corona thickness, and the micelle length. Each of these dimensions are varied among the different configurations ("individuals") while still being within some set bounds, where the bounds are derived by the input polymer chemistry and given micelle shape. For example, the upper bound of the core radius can be based on the contour length of the solvophobic block of the amphiphilic polymer, and similarly the upper bound of the corona width could be based on the contour length of the solvophilic block of the amphiphilic polymer. The variables carrying these dimensions are then encoded in binary within the GA, as discussed in our previous papers.[41,50,53]

For each configuration, we calculate the $I_{\text{comp}}(q)$, by placing scatterers of size, d, within the relevant dimensions of the configuration (e.g., within the core and corona dimensions of a spherical micelle) and then calculate the intramicellar structure factor, $\omega(q)$, based on the Debye equation, as follows

$$\omega(q) = \left\langle 1 + \frac{2}{N_{\text{tot}}} \sum_{i=1}^{N_{\text{tot}}} \sum_{j>i}^{N_{\text{tot}}} \frac{\sin(qr_{ij})}{qr_{ij}} \right\rangle \tag{1}$$

For an amphiphilic polymer, N_{tot} is the sum of N_{A} and N_{B} where N_{A} is the number of solvophilic and N_{B} the number of solvophobic point scatterers placed in the configuration, and r_{ij} is the distance between point scatterers i and j. We note that the ratio of N_{A} to N_{B} is based on the composition of the amphiphilic polymer. These solvophilic and solvophobic point scatterers are randomly placed within the solvophilic and solvophobic domains, respectively, unique to the configuration that is considered. Equation 1 is the key but not the only equation in the $I_{\text{comp}}(q)$ calculation; all equations involved in $I_{\text{comp}}(q)$ calculation are described in our previous papers.[41,50,53] The total number of scatterers and scatterer size, d, are choices the user makes to optimize the increased computational cost that comes with more scatterers and the smaller structural length scales that can be resolved by smaller scatterer size. In our previous papers,[41,50] we demonstrate that the scatter size can be varied to some degree, as long as the scatterer size is small enough to capture the relevant dimensions of the structure; we demonstrate for a select few cases that the scatterer size did not significantly impact the resulting dimensions determined by CREASE.

The fitness of each configuration is then calculated from the log-scaled sum of squared error (sse or fit error) between the $I_{\text{exp}}(q)$ and the $I_{\text{comp}}(q)$ of that configuration, where a high fitness has a low value of sse. The log-scaled form of sse that we have used in this study and past studies,[41,50,53] is chosen because it gives similar importance to the log-scaled features of the $I(q)$ and takes into account the spacing of the q-values, so that the method does not overfit regions where there are more narrowly spaced q-values. After the fitness has been determined for every configuration in the current population, the configurations are selected for the next generation based on a probability that is proportional to the fitness of the configuration. The selected configuration with the highest fitness is preserved as is for the next generation. All selected configurations have a probability to undergo mutation (which increases population diversity) and a different probability to undergo crossover (which improves the convergence of the configurations with high fitness). The probabilities to undergo changes via genetic operators, namely mutation or crossover, are modified in each generation to keep the population from becoming too similar (leading to trapping in local minima of this optimization) or too dissimilar (essentially becoming a random search method that does not consider the fitness of the population). These steps of fitness calculations, selection, and genetic operators are repeated until the fitness of the best configuration and the average fitness of the generation both plateau over multiple generations; in our case the fitness generally converges (reaches a plateau) after 60 generations. At this point, when this plateau is observed we conclude that the GA has converged.

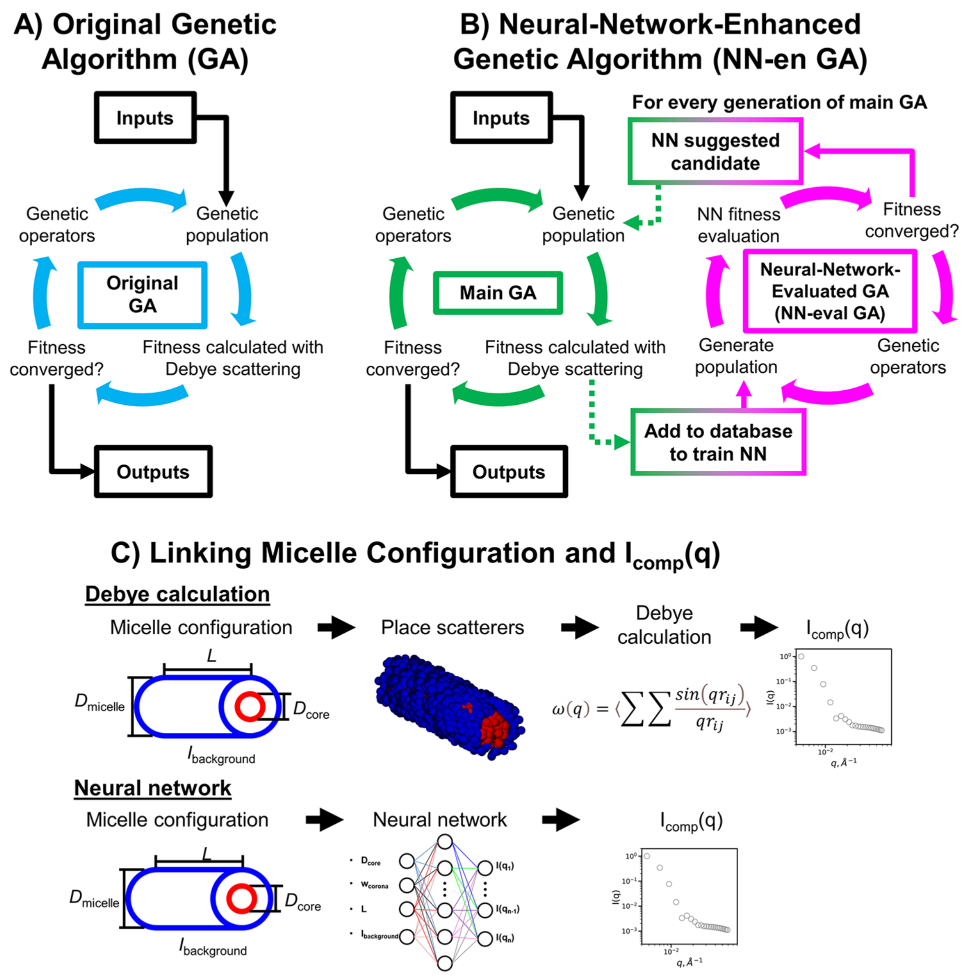


Figure 2. Schematic flow diagrams of (A) genetic algorithm (GA) without machine learning and (B) neural-network-enhanced genetic algorithm (NN-en GA) which contains a neural-network-evaluated genetic algorithm (NN-eval GA). (C) Demonstration of how the micelle configuration is used to determine the $I_{comp}(q)$ with Debye calculations (eq 1) and with an NN.

In this paper, our focus is on improving the speed and efficacy of the GA step. We note that the most computationally intensive step within the GA is the computed scattering intensity, $I_{comp}(q)$, calculation based on eq 1. The computational cost of eq 1 scales with square of the number of scatterers and is affected by the choice of the number of scatterers, the densities of the different chemistries, and the complexity of the structure. Our goal is to use machine learning to either eliminate this computationally intensive step of the GA or improve the efficacy of the GA step by reducing the number of generations (which results in fewer $I_{comp}(q)$ calculations) needed for convergence.

As the focus of this paper is on how we have incorporated machine learning to improve the GA algorithm of CREASE, we only briefly overview the second step of CREASE, the molecular reconstruction step (Figure 1). We refer the reader to our previous papers[41,50] for details about the implementation and application of this molecular reconstruction step. Briefly, using the dimensions for the assembled structure output from the GA in step 1, we perform molecular reconstruction using molecular dynamics (MD) simulations. The polymer chains are modeled using an atomistic or appropriate coarse-grained models (e.g., see perspective and review articles[62,63]) and placed into an initial configuration with the shape and dimensions from the GA output. The chains are then relaxed and equilibrated in that assembled structure using standard MD simulation protocol. After equilibration is completed, one can calculate the distribution of chain configurations, spatial distribution of different monomers, and characterization of interfaces within the assembled structure. For example, in our previous paper[50] we presented the results from the molecular reconstruction part of CREASE and compared the output (e.g., distribution of chain conformations, radial distribution functions) from our molecular reconstruction directly against the ones from the simulation that produced the input *in silico* $I_{exp}(q)$. We validated the protocol of the molecular reconstruction by demonstrating excellent agreement between the results from the molecular reconstruction and that from the simulations that produced the input *in silico* $I_{exp}(q)$. In summary, through this molecular reconstruction step, all of the information about the macromolecular packing within the assembled structures in the amphiphilic polymer solutions is obtained, all of which is not easily accessible using standard approaches involving fits of $I_{exp}(q)$ with analytical models.

2.B. Overview of the Neural-Network-Enhanced Genetic Algorithm (NN-en GA)

To reduce the computational intensity of the $I_{comp}(q)$ calculation (based on eq 1) in the GA, we present an artificial neural-network-enhanced genetic algorithm (in short, NN-en GA). NN-en GA makes use of a neural network to link the relevant structural dimensions specific to a shape (e.g., core and corona dimensions) of the configuration to its $I_{comp}(q)$ in a computationally efficient manner. Based on past studies that used a similar NN-en GA architecture in a different context for soft material optimization,[54−60] we anticipate arriving at the "best" configuration in fewer generations in the NN-en GA than the original GA. In Figure 2A, we show the original GA as described in our previous studies[41,50,53] and in Figure 2B we describe the key steps in this new NN-en GA.

The NN-en GA (Figure 2B) differs from the original GA (blue cycle, Figure 2A) as it integrates a GA (the main GA shown as the green cycle, Figure 2B) with a neural-network-evaluated genetic algorithm (NN-eval GA, pink cycle, Figure 2B). The NN uses the micelle configuration (variables used in the genetic algorithm) and predicts its corresponding $I_{comp}(q)$ (Figure 2C); we show an example of potential setups in Figure S2. For each generation of the main GA, the best NN prediction from a full run of a NN-eval GA serves as an NN suggested candidate into the population of the next generation of the main GA. We note that the combination of the NN-eval GA cycle and the main GA cycle is together referred to as the NN-en GA. The NN-eval GA differs from the main GA in the $I_{comp}(q)$ calculation; the NN-eval GA determines the $I_{comp}(q)$ with the NN trained model that uses the data generated up to the current generation, while the main GA determines the $I_{comp}(q)$ for every configuration in the current generation using eq 1. The $I_{comp}(q)$ calculation in the NN-eval GA is (significantly) more computationally efficient than the main GA's or original GA's $I_{comp}(q)$ calculation by placing scatterers and using eq 1. The trained NN model predicts the "best" $I_{comp}(q)$ corresponding to the dimensions of the configuration using all of the structural data saved from a full run of the NN-eval GA which considers many different configurations over multiple generations. The NN-eval GA outputs the "best" configuration that has an $I_{comp}(q)$ that is the closest to $I_{exp}(q)$. The best configuration, the configuration with the highest fitness, is then provided from the NN-eval GA as a suggested candidate to the main GA. The NN suggested candidate is then explicitly evaluated by placing scatterers and calculating $I_{comp}(q)$ using eq 1 in the next generation of the main GA. This means that if the NN's suggested configuration is closer to the right structure (i.e., global optimum) corresponding to the input $I_{exp}(q)$, then it would persist through generations of genetic operations, and accelerate the main GA with explicitly calculated $I_{comp}(q)$ toward the "best" structure in fewer generations. On the other hand, if the NN's suggested configuration is a poor match to the $I_{exp}(q)$, then that candidate would exhibit low fitness and be expelled from the main GA through the genetic operators in a few generations. As a result, even in the worst-case scenario the NN-en GA (Figure 2B) would perform similarly to the original GA (Figure 2A).

We show in the results section that the training of the NN model and running the NN-eval GA are both faster than explicitly calculating $I_{comp}(q)$ for the population of one generation in the main GA, demonstrating that there is little additional cost toward implementing the NN-en GA. Furthermore, NN-en GA does not require an existing large database from which to train the NN; instead, it progressively builds the NN model with the data generated in each generation of the main GA. This strategy enables the NN-en GA to "learn" from the progressively built database rather than being limited to genetic information contained in the configurations of the current generation. As the computational cost of the NN training scales mainly with the database size (number of training sets) and not the number of scatterers that are placed for the corresponding $I_{comp}(q)$ calculation, this method provides a significant improvement on the evaluation time in the cases where many small scatterers are required to capture the dimensions of the structure. The computational cost of the NN-eval GA depends mainly on the choice of the NN-eval GA architecture (number of generations and individuals/configurations in each generation) as the NN prediction is orders of magnitude more computationally efficient.

2.B.1. Optimizing the NN Architecture and the NN-eval GA Setup. Implementing the NN-en GA requires optimization of the NN hyperparameters and architecture (e.g., number of nodes and hidden layers in the NN as shown in Figures S3−S7), population size of the main GA, and population size of the NN-eval GA. In this subsection, we describe our choices and procedure to optimize these choices.

For the main GA component of the NN-en GA, we use the same choices as those from the original GA from our recent CREASE studies,[50,53] as we will compare with the results from those studies to test the performance of this paper's newly developed NN-en GA. The original GA and the main GA component of the NN-en GA each have a population of 80 individuals or configurations; this number of 80 was chosen to balance the computational cost of the $I_{comp}(q)$ calculations for every configuration in the population size and the need for a large enough population to achieve true global optimum (confirmed by consistent results between independent GA runs). As the NN-eval GA is significantly more computationally efficient than the main GA, a larger population size of 150 was chosen in NN-eval GA. The main GA is run for as many generations as needed for convergence. In contrast, the NN-eval GA is run for 100 generations during which all cases in this paper exhibit convergence, as marked by the best and average sse values reaching a plateau before 100 generations (see Figure S8).

The NN component of the NN-en GA needs to be optimized to create a surrogate model that connects the input (e.g., dimensions of the configurations) and output, the $I_{comp}(q)$. NNs in general consist of an input layer, output layer, and one or more hidden layers, each with a different number of nodes that connect to the nodes of the previous and following layers.[64] The weights and biases that relate the nodes' inputs to the nodes' outputs through the activation function are then optimized through backpropagation to minimize the loss function, which is the error between the NN prediction and the training data.[64] We use the Adam optimization algorithm to perform the backpropagation calculations to determine the weights and biases of each layer, the Rectified Linear Unit (ReLU) activation function, and mean squared error as our loss function. Our tests show that the NN tends to work well with these choices without showing signs of significant overfitting (e.g., such as the training loss decreasing while the validation loss plateaus as a function of the training epoch). As such, we focus on optimizing the number of layers, the number of nodes in each layer, the number of epochs for which to train, as well as the structure of the NN outputs.

Training Data for the NN. We preprocess our data to improve the stability and performance of the NN. To facilitate the NN with learning data over different scales, such as the logarithmic nature of the $I(q)$, we standardize the NN training data. Each of the different categories (e.g., each of the micelle dimensions and the $I(q)$ data) in the training data are standardized by subtracting the mean of the data in that category and then dividing the data in that category by the standard deviation of that category. If the standard deviation of a category is equal to 0, we would only divide by 1. Additionally, for the $I(q)$ values, we take the logarithm of the data before standardizing, as the values of the $I(q)$ differ by multiple orders of magnitude. These preprocessing steps reduce large differences in the weights and biases which would unduly affect the stability and performance of the NN. We then postprocess the results of the NN predictions by reversing the preprocessing steps. By performing these steps in our initial tests, we reduced the prediction error of the NNs as compared to NNs trained without the preprocessed data. We also split the data into training and validation sets, with two-thirds of the data used to train the NN. This allows us to monitor and compare the performance of the NN on data that it was not trained with (validation set) to tune the hyperparameters and to look for signs of whether the NN was overfitting the training data.

NN Architecture. We describe the details of how we optimized the NN architecture in Supporting Information Figures S2−S9. The optimized NN architecture consists of 1 hidden layer with 40 nodes. The NN architecture is kept consistent for each generation and each test case considered in this study. The input layer to the NN consists of a node for each of the variables in the GA configuration (i.e., the dimensions of the micelle, background scattering intensity), and the output layer consists of multiple nodes, one for each $I(q)$ at the corresponding q value of the $I_{exp}(q)$. The NN is trained for 2000 epochs; as a representative example, we show that the choice of 2000 epochs leads to a plateau in the validation loss during the NN training at generations 1, 5, 50, and 100 of the NN-en GA for $I_{exp}(q)$ of *in silico* cylinders in Figure S9. In the early generations of the NN-en GA, we see that the validation error plateaus before the training error, which indicates that if we train for more epochs then the NN would be overfitting the training data, while the choice of 2000 epochs balances optimizing the NN without significantly overfitting to the training data. At later generations, the training and validation loss both plateau before 2000 epochs, indicating that the NN is training well without overfitting to the training data. While this choice could be varied for each generation, we found that the choice of 2000 epochs work well enough in all cases. We use Keras[65] software package for setting up the NN architecture as well as to save and load the trained NNs and use Tensorflow[66] to train the NN models. For benchmarking the code, we performed all calculations on an Intel Core i7-10870H CPU.

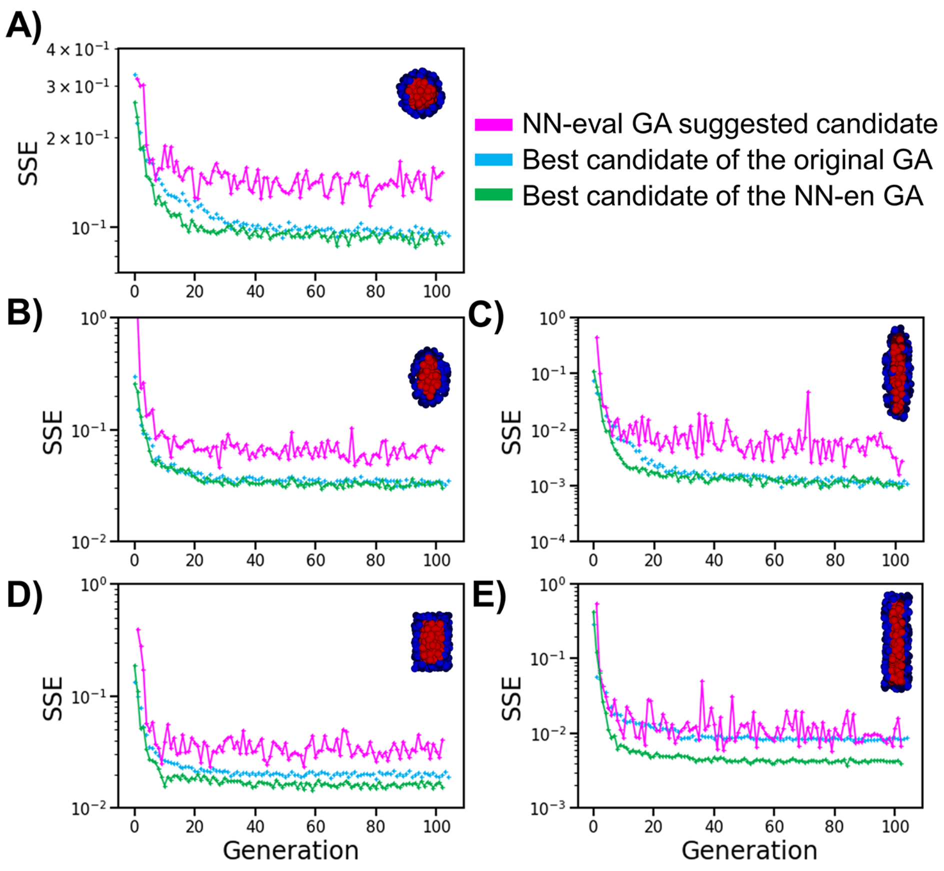


Figure 3. Evolution of the sse (fit error) of the best candidates (highest fitness) in each generation for the original GA (shown in blue) and NN-en GA (shown here in green) as well as the suggested candidate from the NN-eval GA at that generation, for an input $I_{exp}(q)$ from (A) cylinder, (B) ellipse with $P = 2$, (C) ellipse with $P = 8$, (D) fibril with $P = 2$, (E) fibril with $P = 8$. The results shown here are the averages from five independent runs. The % error (95% confidence interval) from these independent runs range from 3 to 40% for the best candidate of the NN-en GA, 8−40% for the best candidate of the original GA, and 100−500% error for the NN-eval GA suggested candidate (this large error is because the suggestion from NN-eval GA is expected to vary significantly from run to run, as shown in Figure S10).

Additionally, we demonstrate that predicting the $I_{comp}(q)$ from a configuration efficiently holds more value in this case than the reverse [i.e., training the NN to predict the configuration from the $I_{exp}(q)$]. This reverse prediction would perform poorly for the case where different configurations could potentially have the same $I(q)$, as it would have conflicting training data. In contrast, the NN-eval GA enables analysis of $I_{exp}(q)$ even in the case where the $I_{comp}(q)$ is nonunique to a single configuration. Efficiently determining the different possible configurations that have the same $I_{comp}(q)$ is valuable as those configurations can then be compared and some possibly eliminated with independent measurements.

2.C. Description of Test Cases

To test the performance of the NN-en GA against the performance of the original GA, we make use of $I_{exp}(q)$ from a selection of different morphologies and scattering profiles described in our previous papers.[50,53] The number of q values can be as low as 24 or as high as 61 for the cases we consider in this paper. We have found that these choices proved to contain enough q values to resolve the key features of the structures for the q range considered. The choice of q values and q range is discussed in more detail in our previous papers.[41,50] The following specific cases of $I_{exp}(q)$ are considered

(a) "cylinder": in this case, the $I_{exp}(q)$ for the cylindrical micelles comes from *in silico* experiments (i.e., molecular simulations) of a solution of linear $A_3-B_{18}-A_3$ chains, where A represents the solvophilic chemistry and B the solvophobic chemistry.

(b) "ellipse with $P = 2$" and "ellipse with $P = 8$": in these cases, the $I_{exp}(q)$ is calculated based on eq 1 for elliptical cylindrical micelles with a typical core−corona type packing with known dimensions without dispersity in the dimensions for $P = 2$ and $P = 8$, where P is the ratio of the largest to smallest dimension of the cross section.

(c) "fibril with $P = 2$" and "fibril with $P = 8$": in these cases, the $I_{exp}(q)$ is calculated based on eq 1 for fibrillar micelles with a typical core−corona type packing with known dimensions that have 15% dispersity in the dimensions for $P = 2$ and $P = 8$.

(d) "compact ellipse" and "compact fibril": in these cases, we analyze the $I_{exp}(q)$ from small angle scattering experiments on solutions that in our past work[53] have been confirmed to have fibrillar micelles with a compact packing (i.e., the whole polymer chain packs into the micelle core) using both the fibril GA code as well as elliptical GA code.

3. RESULTS AND DISCUSSION

In this results section, we evaluate the performance of the NN-en GA against the original GA by comparing the fit error (sse) as well as the resulting dimensions obtained for $I_{exp}(q)$ from different structures between both methods. We test the two methods on $I_{exp}(q)$ for all cases described in Section 2.C.

3.A. Inner Workings of the NN-Enhanced GA Step and NN-Evaluated GA Step

To evaluate the performance of the NN-en GA, we compare the fit error or sse [i.e., log sum of squared errors between $I_{comp}(q)$ and $I_{exp}(q)$] of the best configuration for each generation between the NN-en GA (Figure 2B) and the original GA (Figure 2A).

For cylinder and the ellipse with $P = 8$ (Figure 3A,C), the sse for the best configuration in the NN-en GA (green curve) converges to approximately the same value as the sse of the best configuration in the original GA (blue curve), approximately 20 generations before the original GA. For fibrils with $P = 2$ and $P = 8$ (Figures 3D,E, the sse for the best configurations for both NN-en GA and original GA converge at approximately the same generation, although the NN-en GA converges to a lower value of sse than the original GA. For the ellipse with $P = 2$ (Figure 3B), at all generations the sse of the best configuration from both NN-en GA and the original GA mostly overlap. In this case, the amount of training data required for the NN to predict the best structure for a given $I_{\text{exp}}(q)$ is like the number of configurations needed for the original GA to converge. Overall, the results shown in Figure 3 indicate that in general the NN-en GA decreases the sse value and/or the number of generations required for convergence of the GA or, in the worst case, matches the results from the original GA.

Interestingly, even at later generations the sse of the NN-eval GA suggested candidate in Figure 3 has a higher value than the sse of the best configuration both in the original GA and the NN-en GA. However, if we consider the evolution of sse of the NN-en GA and the NN suggested candidate for each independent run, for the cylinder case in Figure S10, we see that NN suggested candidates becomes the best configuration of the next generation in the main GA of the NN-eval GA, at least once in the first few generations of each of the independent runs. The NN suggested candidate from the NN-eval GA is the best configuration of the next generation in the main GA for the cases where the sse of the NN suggested candidate and the best configuration are equal in Figure S10. While on average the NN suggestion might not reach the converged low sse obtained through the NN-en GA and the original GA, the NN suggested candidate steers the first few generations toward the final sse in fewer generations than without the NN ssuggested candidate.

To understand how the NN suggested candidate varies with increasing generations of the NN-en GA, for the cylinder case we compare at various generations the NN-model derived $I_{\text{comp}}(q)$ of the NN suggested candidate [in short, the $I_{\text{NN-pred}}(q)$], the eq 1 derived $I_{\text{comp}}(q)$ by placing scatterers for the NN suggested candidate [in short, the $I_{\text{NN-Debye}}(q)$], and the eq 1 derived $I_{\text{comp}}(q)$ of the best candidate at that generation in the main GA component of the NN-en GA [in short, the $I_{\text{main-GA}}(q)$]. In generation 1 of the NN-en GA (Figure 4A), the $I_{\text{NN-pred}}(q)$ has the closest match to the $I_{\text{exp}}(q)$. The $I_{\text{NN-Debye}}(q)$ is different from the $I_{\text{NN-pred}}(q)$ as the NN does not have enough training data to make accurate predictions in this generation. In generation 2, Figure 4B, the $I_{\text{NN-pred}}(q)$ and the $I_{\text{NN-Debye}}(q)$ do not completely match, but both are closer to the $I_{\text{exp}}(q)$ than the best $I_{\text{main-GA}}(q)$, helping guide the NN-en GA to match the $I_{\text{exp}}(q)$. However, by generation 7 the $I_{\text{NN-pred}}(q)$ matches $I_{\text{NN-Debye}}(q)$ (Figure 4C) and are both closer to the $I_{\text{exp}}(q)$ than the $I_{\text{main-GA}}(q)$. In generation 30 (Figure 4D), all of the $I_{\text{comp}}(q)$ curves quantitatively approach the $I_{\text{exp}}(q)$ as the sse has converged (Figure 3A). These results show how the NN "learns" the appropriate $I_{\text{comp}}(q)$ for a given micelle configuration and how the NN-eval GA is able to provide the NN suggested candidate that results in an earlier convergence for the NN-eval GA than the original GA.

Next, we understand how the training of the NN changes over different generations of the NN-en GA. For cylinders, we evaluate the sse between the $I_{\text{NN-pred}}(q)$ and the $I_{\text{NN-Debye}}(q)$ (Figure S11); this sse generally converges within approximately the first 20 generations of the NN-en GA implying that the benefit of the NN in the NN-en GA is primarily within the first

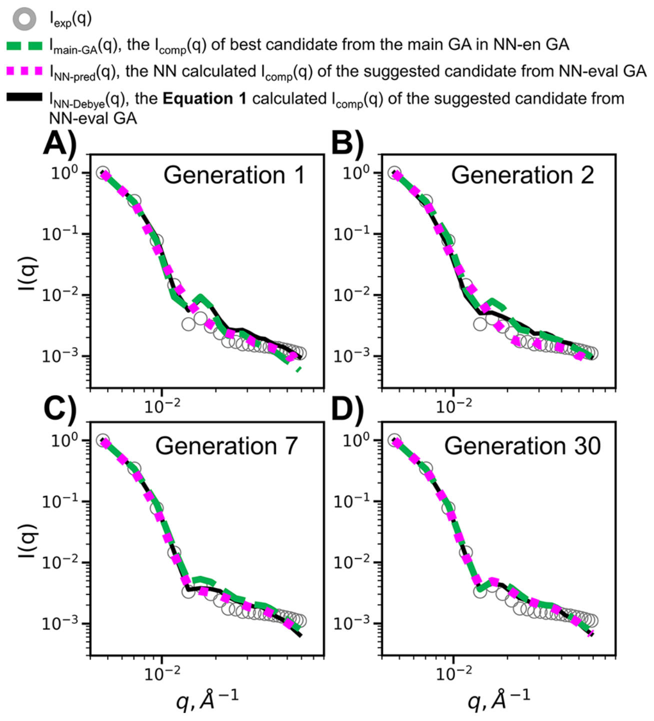


Figure 4. Comparing the input $I_{\text{exp}}(q)$ from *in silico* cylindrical micelles with the $I_{\text{comp}}(q)$ of the "best" (highest fitness) structure of the main GA, the NN determined $I_{\text{comp}}(q)$ of the NN suggested candidate, and the eq 1 determined $I_{\text{comp}}(q)$ of the NN suggested candidate. The results shown are for generations (A) 1, (B) 2, (C) 7, (D) 30 of the NN-en GA.

20 generations of the NN-en GA. After generation 20 of the NN-en GA, the difference between the $I_{\text{NN-pred}}(q)$ and the $I_{\text{NN-Debye}}(q)$ is larger than the final sse obtained by the NN-en GA (Figure 3) and the NN-eval GA does not impact the performance of the NN-en GA. This behavior is inherently linked to how well the NN model is trained; the training loss and validation loss of the NN-model is shown in Figure S12 along with a discussion below that figure.

The sse results for the compact ellipse and compact fibril in Figure S13 are analogous to the ones presented in Figure 3. The sse for the NN-en GA applied to the compact ellipse and compact fibril has both a lower final sse as well as converge at a lower generation as compared to the original GA.

In the next sections, we will look at how the differences in the sse impact the required computational time and the dimensions determined by the NN-en GA and the GA.

3.B. Comparing the Outputs of the NN-en GA and the Original GA

In Figure 5, we compare the dimensions of the final "best" structures as determined from the original GA (without any NN)[50] and this new NN-en GA. We see that the dimensions determined by the original GA, the NN-en GA, and the measured dimensions from the *in silico* cylinders all match within error, specifically the dimensions determined with the original GA and the NN-en GA are similar. In Figure S14, we do a similar comparison for the ellipses with $P = 2$ and $P = 8$ as well as the fibrils with $P = 2$ and $P = 8$. In all cases, the dimensions from the original GA and the NN-en GA match. We see that the original GA and NN-en GA determine the known dimensions of the $I_{\text{exp}}(q)$ within error, except for the total dimensions of the ellipses (Figure S14B,D), for which both GAs obtain dimensions

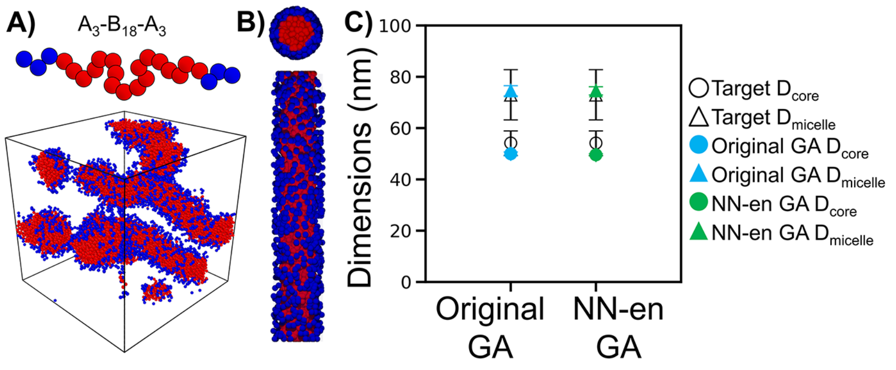


Figure 5. (A) Schematic of ABA amphiphilic polymer and a representative simulation snapshot of *in silico* experiment. (B) Different views of scatterers placed within the micelle configuration whose $I_{comp}(q)$ matches $I_{exp}(q)$. (C) Dimensions of the cylindrical micelles (open symbols) and dimensions obtained from the original GA and NN-en GA. We report the average and standard deviation of the dimensions from five independent runs.

close to the known dimensions but outside of the error. The match between dimensions from original GA and the NN-en GA is surprising in the case of the fibrils with $P = 2$ and $P = 8$ (Figure S14E−H), as the final sse values of the best configuration of the NN-en GA is lower than the sse values of the original GA, indicating that the additional reduction in sse occurred at q values that likely did not significantly impact the dimensions.

In Figure 6, we compare the results from NN-en GA and original GA for an input $I_{exp}(q)$ obtained from scattering experiments on 0.4 wt % of poly(D-glucose carbonate)-based amphiphilic diblock copolymers in a 60:40 vol % mixture of THF and H_2O. The fibril dimensions obtained with TEM, Cryo-TEM, SANS, and the original GA were originally published in ref 53. We see that the dimensions from the NN-en GA are in closer agreement with both the dimensions determined with fits from analytical models to the SANS data and the dimensions from microscopy results than those from the original GA. Additionally, we find that the final sse of best configuration of the NN-en GA is lower than the sse from the original GA (Figure S13). This is an example where NN-en GA improved the accuracy of the CREASE approach.

Overall, all of the results discussed in this section indicate that the NN-en GA reduces the number of generations required to reach a solution that either matches the dimensions determined from the original GA and/or improves the agreement with the target dimensions over those determined by the original GA.

3.C. Computational Cost of the NN-Enhanced GA versus the Original GA

To evaluate the computational cost of the NN-en GA as compared to the original GA, we compare the average computational time that it takes to train the NN for a single generation, run a full NN-eval GA for a trained NN, and do the Debye calculation (eq 1) for all 80 configurations in 1 generation. For all of the results presented in this section, the calculations were performed on the University of Delaware's Farber community cluster compute nodes containing 3800 Intel E5 family processor core and 14TB RAM, a 256TB Lustre filesystem, and an FDR InfiniBand high-speed network.

The results shown in Figure 7 are that for the cylinder. Figure S15 has the corresponding results for ellipses with $P = 2$ and $P = 8$, fibrils with $P = 2$ and $P = 8$, the compact ellipse, and compact fibril. In all cases, the amount of time required to train the NN is about an order of magnitude less than one generation of the original GA run due to its Debye calculation (eq 1), and the time required to run the NN-eval GA is about 2 orders of magnitude less than one generation of the original GA run. For a direct

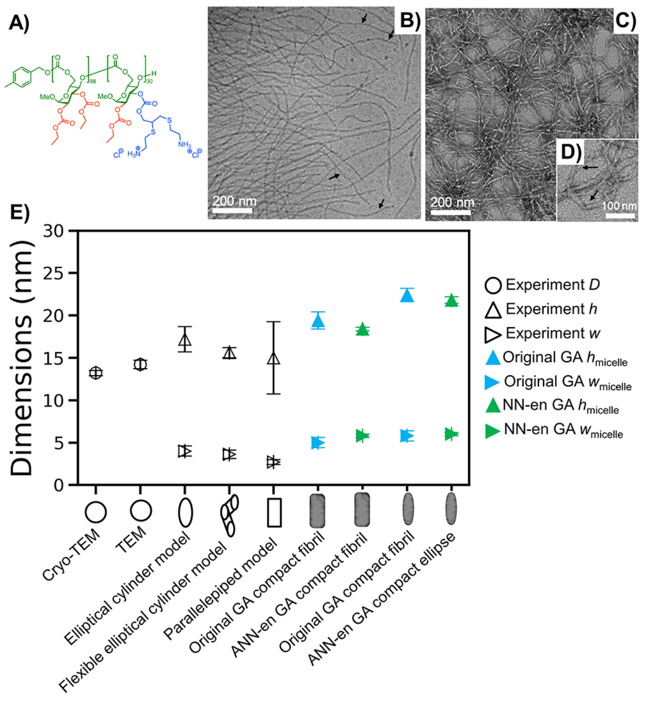


Figure 6. (A) Chemical structure of the poly(D-glucose carbonate) block copolymer with the PGC backbone illustrated in green, the hydrophobic side chains in red, and hydrophilic side chains in blue. (B) Cryogenic-TEM image showing the fibril structures with an average fibril width of 13.2 nm. Arrows in B show twist regions with a different density contrast along the fibril. (C) Negatively stained TEM image is provided to show more detailed characteristics such as (D) folds and twists that are hard to observe from cryo-TEM due to a low contrast from solvent swollen structures. (E) Dimensions obtained with TEM, Cryo-TEM, and SANS presented originally in ref 53 for fibrils formed at 0.4 wt % of poly(D-glucose carbonate)-based amphiphilic diblock copolymers in a 60:40 vol % mixture of THF and H_2O. Also shown in colored symbols are the analyzed results from the original GA and NN-en GA results (the symbols correspond to averages and standard deviations from five independent runs of either original GA or NN-en GA. Images in A, B, C, and D and the data (besides NN-en GA) in part E are reprinted with permission from *Macromolecules* **2020**, *53*, 19, 8581−8591. Copyright (2020) American Chemical Society.

comparison, we take the ratio of the NN training time, Figure S16A, and the run time of the NN-eval GA, Figure S16B, to the total Debye calculation (eq 1) time. On the same computational hardware, the NN training time requires about 20% or less time

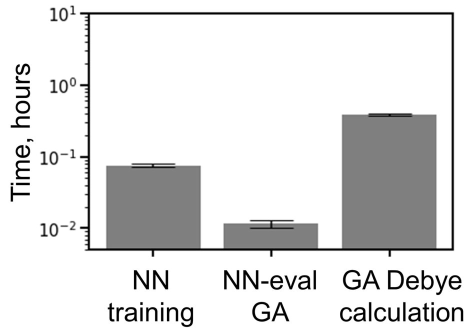


Figure 7. Average computational time required per generation to train the NN, run the NN-eval GA, and perform the GA Debye calculations for inputs from cylinder. The results shown here are the average and 95% confidence interval of five independent runs. The error bars (95% confidence interval) are presented.

than the Debye calculations while requiring nearly half the number of generations to reach a plateau in the sse. This implies an overall speedup of ~30% in the cases that it takes more than a few generations for the sse to converge.

We note that in Figures 7 and S15 the total run time of the NN-eval GA requires <5% of the time that one needs for the Debye calculations of all 80 configurations in one generation. This motivates one to consider using only the NN-eval GA trained with $I_{\mathrm{exp}}(q)$ from high throughput experiments or from a collection of scattering profiles obtained at different times for structures with a similar shape as the case at hand; using only the NN-eval GA could significantly speed up the use of CREASE for systems in which the trained NN is used to calculate $I_{\mathrm{comp}}(q)$, completely replacing the computationally intensive $I_{\mathrm{comp}}(q)$ calculations using eq 1. This is demonstrated next.

3.D. Applying Only the NN-eval GA to Analyze Scattering Profiles

Having demonstrated that the NN-en GA generally improves the performance of the GA step in CREASE as compared to the original GA, in this section we present the use of the NN-eval GA on its own (i.e., only the pink cycle in Figure 2B). In this strategy, one would train the NN to predict the $I_{\mathrm{comp}}(q)$ for a specific structure over the full range of parameters relevant to that structure. After the NN is trained, it should be capable of predicting structures like the structures on which it has been trained. For example, we could train the NN to predict the $I_{\mathrm{comp}}(q)$ of core−shell elliptical cylindrical micelles if one inputs the micelle dimensions. Once trained, the user could solely use the trained NN and the NN-eval GA cycle to analyze the scattering from different experiments (e.g., high throughput scattering data) determining the dimensions of that structure within a significantly shorter computational time as compared to the NN-en GA or original GA of CREASE, which have computationally intensive $I_{\mathrm{comp}}(q)$ calculations in the GA cycle.

We demonstrate the use of NN-eval GA on its own for the case of ellipses. For this demonstration, we trained the NN with data from structures for ellipses with $P = 2$ and $P = 8$. We choose to train the NN-models using data from all 101 generations of the corresponding NN-en GA. These NN models trained on data for $P = 2$ and $P = 8$ core−shell elliptical cylinders are then applied solely in an NN-eval GA cycle to determine the structure for an input $I_{\mathrm{exp}}(q)$ of core−shell elliptical cylinders for a broad range of P-values (= 1, 2, 4, 6, and 8). We use five different NN-models, where each NN-model is trained on the data from an independent NN-en GA run, as shown in Figure S17. The results for the different NN-eval GA runs are shown in Figure S18, with the best performing NN-eval GA results shown in Figure 8. For all P-values, the target core dimensions used to generate the input $I_{\mathrm{exp}}(q)$ (open symbol) match up reasonably well with the core dimensions determined by using solely the NN-eval GA. The qualitative trend in the target micelle dimensions is also captured by the NN-eval GA results. This is remarkable as the NN-model is trained mostly on $P = 2$ and $P = 8$ related configurations sampled in the various generations of the NN-eval GA. In Figure S19, we show the $I_{\mathrm{exp}}(q)$ and $I_{\mathrm{comp}}(q)$ determined by the NN-eval GA for ellipses with target dimensions and $P = 1, 2, 4, 6$, and 8. The results in Figure S19 demonstrate that the $I_{\mathrm{comp}}(q)$ from the NN-eval GA capture the

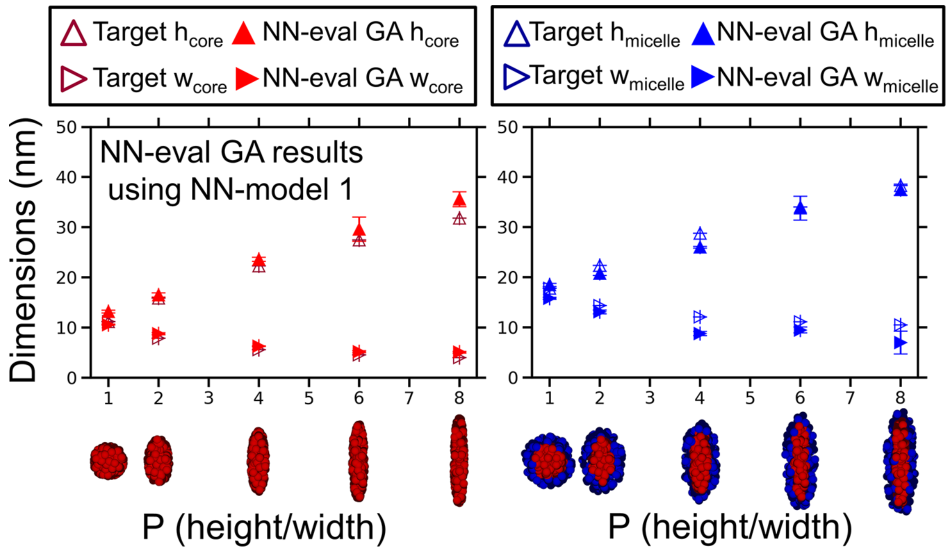


Figure 8. Elliptical cylinder core (red) and micelle (blue) "target" dimensions (open symbols) and dimensions obtained from the NN-eval GA's best configurations (closed symbols). The NNs used (NN1−5) are trained during independent data sets obtained from NN-en GAs for (A) ellipse with $P = 2$ and (B) ellipse with $P = 8$. In each case, we report the average and standard deviations from five independent NN-eval GA runs. The images shown are an example of the scatterers placed within the target dimensions.

changes in the $I_{exp}(q)$ as the P-value changes. We note that these NN-eval GA runs (post NN-training) complete in a matter of minutes on a modest workstation demonstrating the fast and robust predictive power of NN-eval GA.

We expect that the few discrepancies in the NN-eval GA determined dimensions and target dimensions in Figure 8 and Figure S18, result from the data generated during the NN-en GA being skewed toward sampling the dimensions relevant to the target $I_{exp}(q)$ and as such not necessarily sampling a representative set of micelle configurations and their corresponding $I_{comp}(q)$. For example, the $P = 2$ data used to train NN model 1 might have sampled a wider variety of structures and, as such, is trained on a more representative set of data of the full variety of core and total micelle dimensions for elliptical cylinders. Additionally, other considerations made while running the original GA could also be revised to improve the dimensions determined by the NN-en GA. These considerations include choices such as the size and number of the scatterers that are placed to generate the $I(q)$ and the range of q-values that were fit over; the discussion of the size of scatterers and its impact on the q-values fit over is presented our previous papers.[41,50] Indeed, since the computational cost of the NN-eval GA is independent of the choice of the scatterer size, smaller scatterer sizes could be used to generate the training data for the NN to resolve complex structures with features at smaller length scales.

Overall, these results indicate that while the NN-eval GA could provide accurate results for determining the micelle dimensions in a reasonable time for a user to apply this method in practice, the NNs would require training on a representative data set to be able to quantitatively determine the dimensions of structures from $I_{exp}(q)$. Additionally, if the NN-eval GA is to be used on new systems, uncertainty estimates for the NN predictions would be important to include. We do not include the uncertainty estimates in this work because in this implementation we use the NN in the NN-en GA to improve the speed and efficacy of the GA. If one wishes to use uncertainty estimates, then the error estimates for the NN predictions would have to be determined either by testing the trained NN over the relevant parameter space or by using different training schemes that focus on achieving predictions within a set error tolerance over the parameter space of interest. Further, one can also evaluate the error in the NN predictions by comparing the $I_{NN\text{-}pred}(q)$ for the NN-eval GA's "best" candidate by directly comparing it with the $I_{comp}(q)$ calculated using eq 1.

Finally, we consider the reverse case: training the NN to predict the micelle dimensions directly from the $I_{exp}(q)$ and not through a NN-eval GA; we refer to this NN as the direct NN. As shown in Figures S20 and S21 and the accompanying text in the SI, we first determine an appropriate NN architecture to use for the direct NN. Then, to test the performance of the direct NN, we train direct NN models 10 separate times on the same data generated during the NN-en GA run for an ellipse with $P = 2$. These 10 trained direct NN models are then applied to input $I_{exp}(q)$ of core−shell elliptical cylinders with a broad range of P-values (= 1, 2, 4, 6, and 8). The results in Figure S22 show that the direct NN qualitatively predicts the increase in P-values, although the uncertainties of the dimensions are significantly larger than those obtained using a NN-eval GA (Figure 8). Additionally, as we briefly discussed in Section 2.B, the $I_{exp}(q)$ features are not necessarily unique to a single micelle configuration, further increasing the large uncertainties in the direct NN predictions. In contrast, the NN-eval GA provides multiple candidates whose $I_{comp}(q)$ matches the $I_{exp}(q)$ which can then be considered or carefully eliminated through additional experimentation, rather than using direct NN and providing a single micelle configuration with large uncertainties in the predicted values.

4. CONCLUSIONS

In this paper, we have presented a machine learning enhanced computational approach (CREASE) that can be used to analyze scattering results from amphiphilic polymer solutions and determine structural features of the assembled polymers. The original computational approach, CREASE, consists of two steps: the first step involves the use of a genetic algorithm (GA) to determine the macroscopic dimensions of the assembled structure and the second step involves the use of molecular simulations to elucidate the molecular and chain packing within the domains of the assembled structure. By using machine learning techniques, we have demonstrated significant improvement in the computational speed and efficacy of the GA step of CREASE, making it faster and in some cases, more accurate for scattering analysis than our original CREASE method.

The GA step in CREASE is used to determine the dimensions of the domains in the assembled structure by finding the optimal configuration whose computed scattering profile matches the input experimental scattering profile. This calculation of the computed scattering profile is computationally intensive as it is based on pairwise distances of the scatterers and thus, scales with squared number of scatterers placed within the dimensions of each structure. The number of scatterers is related to scatterer size which is chosen to resolve the dimensions of the structure being investigated; if one chooses a smaller scatterer size, then that leads to the need for more scatterers to define the shape of the structure. Furthermore, this calculation of the computed scattering profile is also repeated for every structure sampled in the various generations of the GA step leading to a large computational cost. By using artificial neural networks (NNs), we have reduced the number of generations required by the GA step in CREASE to converge and quickly identify the "best" structure whose computed scattering matches with the experimental scattering. We tested the application of the NN-enhanced GA on scattering profiles from cylindrical, elliptical cylindrical, and fibrillar micelles. The results demonstrate that the NN-enhanced GA generally improves the speed and efficacy of the GA step in CREASE by reducing the number of generations required for the GA fitness to plateau as well as matching or improving the dimensions determined by the original GA without machine learning.

Furthermore, the trained NN model could also be applied to determine the dimensions for an input $I_{exp}(q)$ for structures with different dimensions but the same shape as the structures used to train the NN model. By using that trained NN model, we could completely skip the costly calculation that scales with the number of scatterers in the main GA, and run the NN-evaluated GA alone, which is much faster than the original GA component of CREASE.

Both the NN-enhanced GA step and NN-evaluated GA step of the CREASE approach are useful for quickly analyzing results from high throughput scattering experiments on amphiphilic polymer solutions and can be easily extended and used for analyzing scattering experiments on other polymer and soft matter systems. These approaches are valuable for characterizing scattering profiles from kinetically trapped, nonequilibrium structures and/or novel polymer chemistries for which standard fitting to conventional analytical models would be inaccurate/too approximate.

ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acspolymersau.1c00015.

Schematics of the relevant dimensions of the shapes considered in this study, our optimization steps for determining the NN architecture, and additional cases that on which we have tested the NN-en GA (PDF)

AUTHOR INFORMATION

Corresponding Author

Arthi Jayaraman − *Colburn Laboratory, Department of Chemical and Biomolecular Engineering, University of Delaware, Newark, Delaware 19716, United States; Department of Materials Science and Engineering, University of Delaware, Newark, Delaware 19716, United States;* orcid.org/0000-0002-5295-4581; Email: arthij@udel.edu

Author

Michiel G. Wessels − *Colburn Laboratory, Department of Chemical and Biomolecular Engineering, University of Delaware, Newark, Delaware 19716, United States*

Complete contact information is available at: https://pubs.acs.org/10.1021/acspolymersau.1c00015

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

We acknowledge and thank the National Science Foundation DMREF program, Grant NSF DMREF-1629156, for their financial support. The GA runs reported in this paper were conducted on the high-performance computing resources of the Farber supercomputing cluster run by the Information Technologies (IT) resources at the University of Delaware.

REFERENCES

(1) Hammouda, B. SANS from polymers—review of the recent literature. *Polym. Rev.* **2010**, *50* (1), 14−39.

(2) Hore, M. J. Polymers on nanoparticles: structure & dynamics. *Soft Matter* **2019**, *15* (6), 1120−1134.

(3) Olsen, B. D.; Segalman, R. A. Structure and thermodynamics of weakly segregated rod− coil block copolymers. *Macromolecules* **2005**, *38* (24), 10127−10137.

(4) Pokorski, J. K.; Hore, M. J. Structural characterization of protein−polymer conjugates for biomedical applications with small-angle scattering. *Curr. Opin. Colloid Interface Sci.* **2019**, *42*, 157−168.

(5) Atanase, L. I.; Riess, G. Self-assembly of block and graft copolymers in organic solvents: An overview of recent advances. *Polymers* **2018**, *10* (1), 62.

(6) Blanazs, A.; Armes, S. P.; Ryan, A. J. Self-assembled block copolymer aggregates: from micelles to vesicles and their biological applications. *Macromol. Rapid Commun.* **2009**, *30* (4−5), 267−277.

(7) Kataoka, K.; Harada, A.; Nagasaki, Y. Block copolymer micelles for drug delivery: design, characterization and biological significance. *Adv. Drug Delivery Rev.* **2012**, *64*, 37−48.

(8) O'Reilly, R. K.; Hawker, C. J.; Wooley, K. L. Cross-linked block copolymer micelles: functional nanostructures of great potential and versatility. *Chem. Soc. Rev.* **2006**, *35* (11), 1068−1083.

(9) Riess, G. Micellization of block copolymers. *Prog. Polym. Sci.* **2003**, *28* (7), 1107−1170.

(10) Rodriguez-Hernandez, J.; Chécot, F.; Gnanou, Y.; Lecommandoux, S. Toward 'smart'nano-objects by self-assembly of block copolymers in solution. *Prog. Polym. Sci.* **2005**, *30* (7), 691−724.

(11) Rösler, A.; Vandermeulen, G. W.; Klok, H.-A. Advanced drug delivery devices via self-assembly of amphiphilic block copolymers. *Adv. Drug Delivery Rev.* **2012**, *64*, 270−279.

(12) Verduzco, R.; Li, X.; Pesek, S. L.; Stein, G. E. Structure, function, self-assembly, and applications of bottlebrush copolymers. *Chem. Soc. Rev.* **2015**, *44* (8), 2405−2420.

(13) Ngo, T. D.; Kashani, A.; Imbalzano, G.; Nguyen, K. T. Q.; Hui, D. Additive manufacturing (3D printing): A review of materials, methods, applications and challenges. *Composites, Part B* **2018**, *143*, 172−196.

(14) Wei, M. L.; Gao, Y. F.; Li, X.; Serpe, M. J. Stimuli-responsive polymers and their applications. *Polym. Chem.* **2017**, *8* (1), 127−143.

(15) Bates, F. S. Polymer-polymer phase behavior. *Science* **1991**, *251* (4996), 898−905.

(16) Bates, F. S.; Fredrickson, G. H. Block copolymer thermodynamics: theory and experiment. *Annu. Rev. Phys. Chem.* **1990**, *41* (1), 525−557.

(17) Mai, Y.; Eisenberg, A. Self-assembly of block copolymers. *Chem. Soc. Rev.* **2012**, *41* (18), 5969−5985.

(18) Tritschler, U.; Pearce, S.; Gwyther, J.; Whittell, G. R.; Manners, I. 50th anniversary perspective: Functional nanoparticles from the solution self-assembly of block copolymers. *Macromolecules* **2017**, *50* (9), 3439−3463.

(19) Cho, H. K.; Cheong, I. W.; Lee, J. M.; Kim, J. H. Polymeric nanoparticles, micelles and polymersomes from amphiphilic block copolymer. *Korean J. Chem. Eng.* **2010**, *27* (3), 731−740.

(20) Choucair, A.; Eisenberg, A. Control of amphiphilic block copolymer morphologies using solution conditions. *Eur. Phys. J. E: Soft Matter Biol. Phys.* **2003**, *10* (1), 37−44.

(21) Sprouse, D.; Jiang, Y.; Laaser, J. E.; Lodge, T. P.; Reineke, T. M. Tuning cationic block copolymer micelle size by pH and ionic strength. *Biomacromolecules* **2016**, *17* (9), 2849−2859.

(22) Franken, L. E.; Boekema, E. J.; Stuart, M. C. A. Transmission Electron Microscopy as a Tool for the Characterization of Soft Materials: Application and Interpretation. *Adv. Sci.* **2017**, *4* (5), 1600476.

(23) Watt, J.; Huber, D. L.; Stewart, P. L. Soft matter and nanomaterials characterization by cryogenic transmission electron microscopy. *MRS Bull.* **2019**, *44* (12), 942−948.

(24) McConney, M. E.; Singamaneni, S.; Tsukruk, V. V. Probing Soft Matter with the Atomic Force Microscopies: Imaging and Force Spectroscopy. *Polym. Rev.* **2010**, *50* (3), 235−286.

(25) Chyasnavichyus, M.; Young, S. L.; Tsukruk, V. V. Recent advances in micromechanical characterization of polymer, biomaterial, and cell surfaces with atomic force microscopy. *Jpn. J. Appl. Phys.* **2015**, *54* (8S2), 08LA02.

(26) Hammouda, B. SANS from Polymers-Review of the Recent Literature. *Polym. Rev.* **2010**, *50* (1), 14−39.

(27) Narayanan, T.; Wacklin, H.; Konovalov, O.; Lund, R. Recent applications of synchrotron radiation and neutrons in the study of soft matter. *Crystallogr. Rev.* **2017**, *23* (3), 160−226.

(28) Bernado, P.; Shimizu, N.; Zaccai, G.; Kamikubo, H.; Sugiyama, M. Solution scattering approaches to dynamical ordering in biomolecular systems. *Biochim. Biophys. Acta, Gen. Subj.* **2018**, *1862* (2), 253−274.

(29) Gräwert, T. W.; Svergun, D. I. Structural modeling using solution small-angle X-ray scattering (SAXS). *J. Mol. Biol.* **2020**, *432* (9), 3078−3092.

(30) Blanchet, C. E.; Svergun, D. I. Small-angle X-ray scattering on biological macromolecules and nanocomposites in solution. *Annu. Rev. Phys. Chem.* **2013**, *64*, 37−54.

(31) Li, T.; Senesi, A. J.; Lee, B. Small angle X-ray scattering for nanoparticle research. *Chem. Rev.* **2016**, *116* (18), 11128−11180.

(32) Petoukhov, M. V.; Franke, D.; Shkumatov, A. V.; Tria, G.; Kikhney, A. G.; Gajda, M.; Gorba, C.; Mertens, H. D.; Konarev, P. V.;

Svergun, D. I. New developments in the ATSAS program package for small-angle scattering data analysis. *J. Appl. Crystallogr.* **2012**, *45* (2), 342−350.

(33) Breßler, I.; Pauw, B. R.; Thünemann, A. F. McSAS: software for the retrieval of model parameter distributions from scattering patterns. *J. Appl. Crystallogr.* **2015**, *48* (3), 962−969.

(34) Breßler, I.; Kohlbrecher, J.; Thünemann, A. F. SASfit: a tool for small-angle scattering data analysis using a library of analytical expressions. *J. Appl. Crystallogr.* **2015**, *48* (5), 1587−1598.

(35) Ilavsky, J.; Jemian, P. R. Irena: tool suite for modeling and analysis of small-angle scattering. *J. Appl. Crystallogr.* **2009**, *42* (2), 347−353.

(36) Doucet, M.; Cho, J. H.; Alina, G.; Bakker, J.; Bouwman, W.; Butler, P.; Campbell, K.; Gonzales, M.; Heenan, R.; Jackson, A. *SasView*, version 4.2.2; 2017 (accessed January 2020).

(37) Pedersen, J. S.; Gerstenberg, M. C. Scattering form factor of block copolymer micelles. *Macromolecules* **1996**, *29* (4), 1363−1365.

(38) Heo, T. Y.; Kim, I.; Chen, L. W.; Lee, E.; Lee, S.; Choi, S. H. Effect of Ionic Group on the Complex Coacervate Core Micelle Structure. *Polymers* **2019**, *11* (3), 455.

(39) Borbely, S. Aggregate structure in aqueous solutions of Brij-35 nonionic surfactant studied by small-angle neutron scattering. *Langmuir* **2000**, *16* (13), 5540−5545.

(40) Svaneborg, C.; Pedersen, J. S. A Monte Carlo study on the effect of excluded volume interactions on the scattering from block copolymer micelles. *J. Chem. Phys.* **2000**, *112* (21), 9661−9670.

(41) Beltran-Villegas, D. J.; Wessels, M. G.; Lee, J. Y.; Song, Y.; Wooley, K. L.; Pochan, D. J.; Jayaraman, A. Computational Reverse-Engineering Analysis for Scattering Experiments on Amphiphilic Block Polymer Solutions. *J. Am. Chem. Soc.* **2019**, *141* (37), 14916−14930.

(42) Cui, H.; Chen, Z.; Zhong, S.; Wooley, K. L.; Pochan, D. J. Block copolymer assembly via kinetic control. *Science* **2007**, *317* (5838), 647−650.

(43) Vena, M. P.; de Moor, D.; Ianiro, A.; Tuinier, R.; Patterson, J. P. Kinetic state diagrams for a highly asymmetric block copolymer assembled in solution. *Soft Matter* **2021**, *17* (4), 1084−1090.

(44) Early, J. T.; Block, A.; Yager, K. G.; Lodge, T. P. Molecular Weight Dependence of Block Copolymer Micelle Fragmentation Kinetics. *J. Am. Chem. Soc.* **2021**, *143*, 7748.

(45) Zhao, D.; Wang, E.; Lodge, T. P. Hybridization of a Bimodal Distribution of Copolymer Micelles. *Macromolecules* **2020**, *53* (18), 7705−7716.

(46) McGreevy, R. L. Reverse monte carlo modelling. *J. Phys.: Condens. Matter* **2001**, *13* (46), R877.

(47) Soper, A. Empirical potential Monte Carlo simulation of fluid structure. *Chem. Phys.* **1996**, *202* (2−3), 295−306.

(48) Hagita, K.; McGreevy, R.; Arai, T.; Inui, M.; Matsuda, K.; Tamura, K. First example of multi-scale reverse Monte Carlo modeling for small-angle scattering experimental data using reverse mapping from coarse-grained particles to atoms. *J. Phys.: Condens. Matter* **2010**, *22* (40), 404215.

(49) Soper, A.; Edler, K. Coarse-grained empirical potential structure refinement: Application to a reverse aqueous micelle. *Biochim. Biophys. Acta, Gen. Subj.* **2017**, *1861* (6), 1652−1660.

(50) Wessels, M. G.; Jayaraman, A. Computational Reverse-Engineering Analysis of Scattering Experiments (CREASE) on Amphiphilic Block Polymer Solutions: Cylindrical and Fibrillar Assembly. *Macromolecules* **2021**, *54* (2), 783−796.

(51) De Jong, K. A. *Analysis of the behavior of a class of genetic adaptive systems*; University of Michigan, 1975.

(52) Eiben, A. E.; Smith, J. E. *Introduction to evolutionary computing*; Springer, 2003; Vol. 53.

(53) Lee, J. Y.; Song, Y.; Wessels, M. G.; Jayaraman, A.; Wooley, K. L.; Pochan, D. J. Hierarchical Self-Assembly of Poly (D-glucose carbonate) Amphiphilic Block Copolymers in Mixed Solvents. *Macromolecules* **2020**, *53* (19), 8581−8591.

(54) Patra, T. K.; Meenakshisundaram, V.; Hung, J. H.; Simmons, D. S. Neural-Network-Biased Genetic Algorithms for Materials Design: Evolutionary Algorithms That Learn. *ACS Comb. Sci.* **2017**, *19* (2), 96−107.

(55) Mannodi-Kanakkithodi, A.; Pilania, G.; Huan, T. D.; Lookman, T.; Ramprasad, R. Machine Learning Strategy for Accelerated Design of Polymer Dielectrics. *Sci. Rep.* **2016**, *6*, 1−10.

(56) Kim, C.; Batra, R.; Chen, L. H.; Tran, H.; Ramprasad, R. Polymer design using genetic algorithm and machine learning. *Comput. Mater. Sci.* **2021**, *186*, 110067.

(57) Kolsbjerg, E. L.; Peterson, A. A.; Hammer, B. Neural-network-enhanced evolutionary algorithm applied to supported metal nanoparticles. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2018**, *97* (19), 195424 DOI: 10.1103/PhysRevB.97.195424.

(58) Wu, S.; Kondo, Y.; Kakimoto, M. A.; Yang, B.; Yamada, H.; Kuwajima, I.; Lambard, G.; Hongo, K.; Xu, Y. B.; Shiomi, J.; Schick, C.; Morikawa, J.; Yoshida, R. Machine-learning-assisted discovery of polymers with high thermal conductivity using a molecular design algorithm. *Npj Computational Materials* **2019**, *5*, 66.

(59) Jennings, P. C.; Lysgaard, S.; Hummelshoj, J. S.; Vegge, T.; Bligaard, T. Genetic algorithms for computational materials discovery accelerated by machine learning. *Npj Computational Materials* **2019**, *5*, 1−6.

(60) Meenakshisundaram, V.; Hung, J. H.; Simmons, D. S. Design rules for glass formation from model molecules designed by a neural-network-biased genetic algorithm. *Soft Matter* **2019**, *15* (39), 7795−7808.

(61) Hammouda, B. A new Guinier−Porod model. *J. Appl. Crystallogr.* **2010**, *43* (4), 716−719.

(62) Zhang, Q.; Lin, J.; Wang, L.; Xu, Z. Theoretical modeling and simulations of self-assembly of copolymers in solution. *Prog. Polym. Sci.* **2017**, *75*, 1−30.

(63) Gartner, T. E., III; Jayaraman, A. Modeling and simulations of polymers: A Roadmap. *Macromolecules* **2019**, *52* (3), 755−786.

(64) Schmidhuber, J. Deep learning in neural networks: An overview. *Neural networks* **2015**, *61*, 85−117.

(65) Chollet, F., Keras: The python deep learning library. *Astrophysics Source Code Library* **2018**, ascl: 1806.022 (accessed November 2020).

(66) Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M. In *Tensorflow: A system for large-scale machine learning*; 12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16), November 2−4, 2016, Savannah, GA, USA; ISBN 978-1-931971-33-1; pp 265−283.