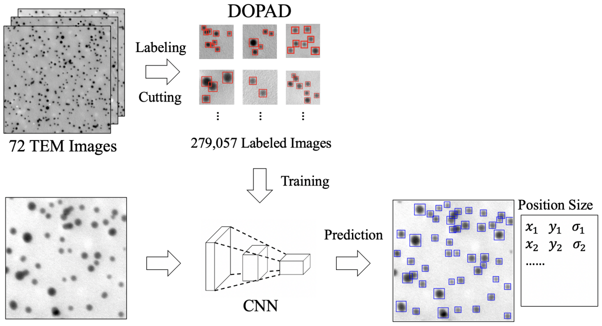
Quantifying Nanoparticle Assembly States in Polymer Matrix Through Deep Learning

Eric Zhonghang Qu,[†] Andrew Matthew Jimenez,[‡] Sanat K. Kumar,[‡] and Kai Zhang[*,†]

[†]*Division of Natural and Applied Sciences, Duke Kunshan University, Kunshan, Jiangsu, 215300, China*

[‡]*Department of Chemical Engineering, Columbia University, New York, New York, 10027, USA*

E-mail: kai.zhang@dukekunshan.edu.cn



Abstract

There is great interest in controlling the spatial dispersion of inorganic nanoparticles (NPs) in an organic polymer matrix, because this centrally underpins the property enhancements obtained from these hybrid materials. Currently, qualitative information on NP spatial distribution is obtained by visual inspection of transmission electron microscopy (TEM) images. Quantitative information is only indirectly obtained through the use of scattering probes such as small angle X-ray/neutron scattering. While the main challenge, that scattering probes operate in reciprocal space, can be remedied by Fourier inverting the data into real space, a much harder issue is deconvolving the contribution of the particle form factor (which is affected by the details of the NP size and shape) from the structure factor which contains information on the NP spatial distribution. These problems become acute when we deal with the popular topic of NPs grafted with polymer chains, because the polymeric corona, and hence the particle form factor, becomes context dependent and hard to quantify. To make progress, we develop and apply a deep-learning based image analysis method to quantify the distribution of spherical NPs in a polymer matrix directly from their real-space TEM images. A dataset of NP detection (DOPAD) is built by manually labeling particle positions on experimental TEM images of diverse polymer composite systems. A convolutional neural network (CNN) object detection model is then trained on DOPAD. Together with sliding-window and merging algorithms, an automated pipeline is established, which takes a large TEM image as input and extracts NP locations and sizes. We validate the structural information resulting from this method against SAXS derived structural information for NPs ordered by polymer crystallization, and then use it to distinguish between different states of the assembly of polymer grafted NPs in a polymer matrix achieved by using their surfactancy. We show that this data-rich protocol allows us to draw critical facets of experimental behavior which have previously not been accessible. The DOPAD dataset, Python source code and trained model are shared on GitHub.

1 Introduction

Organic polymers mixed with (inorganic) nanoparticles (NPs), such as silica nanospheres,[1] carbon nanotubes [2] and graphene nanolayers,[3] are capable of exhibiting significantly enhanced mechanical, electrical, thermal, and barrier properties in comparison to pure polymers.[4,5] These property improvements are strongly affected by the shape, loading, and in particular, the spatial dispersion of the NPs in the polymer matrix. Due to the natural tendency of NPs to agglomerate and phase separate from polymers,[6] a great deal of effort has been applied to control their dispersion;[7,8] the quantitative characterization of the NP dispersion state is then the prerequisite for rational design of composite materials.

Unlike other characterization techniques, transmission electron microscopy (TEM) of polymer nanocomposites directly reveals the morphology of nanofillers. However, further analysis of TEM images usually rely on manual processing and human-centered experience, or at best, with the help of hand-engineered computer algorithms,[9] which can be inaccurate and inefficient.[10] These problems are particularly exacerbated for high filler loadings where the image of NPs in different layers in a TEM slice (typically 100 nm in thickness) can overlap in the resulting two-dimensional image. This makes it difficult to properly resolve individual NPs and hence characterize their dispersion state. Given the ever-increasing amount of experimental data, a fast and reliable way to quantitatively extract positions and sizes of NPs from TEM images becomes crucial for high-throughput materials development.

Deep learning methods based on convolutional neural networks (CNN) can address these issues and provide suitable solutions.[11] Recently, this type of approach has been used to locate magnetite particles,[12] recognize local atomic structure,[13] reconstruct holography of microparticles,[14] identify atomic defects,[15] recognize metal nanoparticles deposited on graphite[16] and resolve liquid-phase TEM videos.[17] Attempts have even been made to directly map two-dimensional (2D) image pixels to high-level predictions, such as material properties[18] and structural types.[19] Despite the wide variety of reseach efforts, the problem of detecting NPs in polymer composites still remains an open challenge. The challenges here are two-fold: First, one has to establish a suitable dataset, namely, representative TEM images of polymer nanocomposites. Using datasets of other systems or styles to solve this problem will give biased predictions. Second, TEM images of polymer nanocomposites can be complicated by the presence of a vast number of (sometimes overlapping) NPs and their exotic morphologies.

In this article, we create a large and representative dataset of TEM images of polymer nanocomposites filled with spherical NPs (Section 2.1-2.2). Compared with fibers, tubes or rods, spheres are a major type of nanofillers that can be synthesized with various chemical compositions.[20,21] A state-of-the-art object detection method in deep learning, YOLOv3 (Section 2.3), is then used to locate particles with rectangular bounding boxes[22] (Section 2.4). An efficient pipeline using a sliding-window method is established to process large TEM images and make accurate predictions about tens of thousands fillers within minutes (Section 2.5). The same task would normally cost a human labeler an hour. Particle locations and sizes are automatically extracted, which allows for further quantitaive analysis[23] (Section 3). The end-to-end predictor in this work is deployed on GitHub[1] to facilitate the task of NP characterization. The dataset and Python source code are also shared, which can be used to systematically improve the performance of our method in the future.

2 Methods

2.1 Experimental Preparation of TEM Images

We use data from several experimental situations that have been studied previously by our groups. Three types of NPs were used: small iron oxide NP ($\approx$ 11 nm diameter), a large silica NP ($\approx$ 50 nm diameter) and a smaller NP ($\approx$ 14 nm) mixed with a variety of matrices. These are typical NPs used in polymer nanocomposites, which can represent a large number of systems. However, one should be cautious when applying the model developed in this work to study images of extremely different styles.

We first consider a mixture of 50 nm diameter NPs with poly(2-vinyl pyridine) (P2VP) – more details of this system are presented in Ref.,[24] but for the purposes of this work it is sufficient to note that these NPs are well dispersed. In a second class of samples we mix (magnetic) iron oxide NPs in a poly(methylmethacrylate) (PMMA) matrix - in these cases the magnetic moment of the NPs to locally align head-to-tail into clusters. A third class of systems we examine are composites comprised of the semicrystalline polymer -

[1] https://dopad.github.io

polyethyleneoxide (PEO) mixed with 14 nm diameter NPs. The silica NPs are grafted with PMMA chains to ensure that the they are compatible with the PEO in the melt state. In this case, the process of polymer crystallization moves the 14 nm diameter silica NPs into the interlamellar zone. Previous SAXS data and TEM picture confirm this ordering, but the analysis is complicated by three factors: (i) The convolution of the NP form factor with the structure factor. Here it is critical to note that the silica NPs are polydisperse, making the determination of structure complicated. (ii) The presence of the PMMA brush which affects the structure factor, but which essentially has no contrast with the PMMA, complicates the structural assignment. (iii) There is (small) contrast between the crystal and amorphous regions of the polymer which contributes to the scattering, complicating the analysis of the scattering data. A proper analysis of TEM images could be used to mimic the NP structure factor by only using the positions of the NP centers (i.e., treating them as point scatterers) and from there obtain detailed information of the structure. Finally, we examine the case of 14 nm silica NPs grafted with polyisoprene chains placed in a polyisoprene matrix. The grafting density and chain length of the grafted chains control the surfactancy of the NPs; when combined with the chain length of the matrix then we can obtain a variety of NP assemblies. In this work we examine a particular case where the NPs phase separate from the matrix and form large agglomerates.[25]

These polymer nanocomposite samples were typically formed made by codissolving the NPs and the matrix polymer into a common solvent followed by solvent casting and drying.[25,26] The nanocomposites were annealed and subsequently microtomed using a Leica Ultracut UCT microtome to sections 60-200 nm thick. Samples were then imaged on a Philips CM12 TEM with a Gatan $4k \times 2.7k$ digital camera. For model training, we prepare 72 large TEM images of different polymer nanocomposites, each containing about 3500 NPs. The resolution of these images ranges from 2000 × 2000 to 6000 × 6000 pixels. More high resolution TEM images are held for final testing of the model's performance.

2.2 Dataset and Preprocessing

In each of the 72 large TEM images, we first manually labeled the positions of all NPs, distinguishable by the naked eye, with rectangular bounding boxes using the online tool Colabeler.[27] To reduce the dimension of the problem, each large TEM image is cut into many small square images of the same size, each containing a few particles (less than 30). In each cutting, the dimension of small images, $N \times N$ pixels, are set by randomly chosing an integer N from $[75, 375]$. Because the neural network architecture used in this work requires the linear dimension of input images to be a multiple of 32, all the small images are then rescaled to 416×416 pixels such that particles appear to have different sizes across different samples. The moderate number $416 = 32 \times 13$ is often used because it gives a good balance between resolution and computational efficiency. With this protocol, we arrive at a dataset of 279,057 labeled square images, DOPAD (Dataset Of nanoPArticle Detection), as shown by examples in Figure 1.

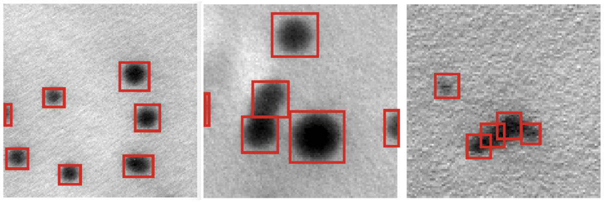


Figure 1: Examples of training images in DOPAD, each of 416×416 pixels, containing 3-9 particles labeled by red rectangles.

After randomly shuffling, the total dataset is then split into a training set of 223,176 (80%) and a validation set of 55,881 (20%) images. For data augmentation, photometric distortions are applied to the training set. Each small square image is subjected to randomized hue (±20%), randomized saturation (50-150%), randomized brightness (50-150%), randomized contrast (50-150%) and a 50% chance of being flipped horizontally.

2.3 Neural Network Model

We adopt the classical deep convolutional neural network (CNN) model, YOLOv3,[22,28] which has 251 layers based on the Darknet-53 backbone,[29] to fulfill this object detection task. The model is built with Keras[30] and tensorflow 2.0[31] and modified from the implementation by Zang.[32] Each input image i with 416 × 416 pixels and three color channels is formatted as a $(416, 416, 3)$ dimensional array $\mathbf{X}_i$, whose entries are integer values between 0 and 255. These integers are rescaled to floats between 0 and 1 before being fed to the neural network.

The output contains information on (multiple) predicted bounding boxes. Each bounding box j on image i is characterized by three parts: (coordinates, score, class). The coordinates part has a form of $(x^j_{\text{min}}, y^j_{\text{min}}, x^j_{\text{max}}, y^j_{\text{max}})$, which are coordinates of the lower-left and upper-right corners, or equivalently, (x^j, y^j, w^j, h^j), which are the center coordinates, width and height. The confidence score s_j is a value between 0 and 1, representing the confidence of the bounding box. The class c_j denotes the type that the object belongs to, which is set to 1 because all NPs are considered to be of the same type in our TEM images. The overall output $\mathbf{Y}_i$ of image i with m bounding boxes thus contains a list of arrays $(x^j, y^j, w^j, h^j, s_j, 1)$, $j = 1, 2, \cdots, m$. Our model can detect up to $m = 100$ objects in each image, which is more than enough for current task. Typically, the model returns $m = 4 - 5$ bounding boxes for each small image in DOPAD.

2.4 Training

The model is trained on dual-Nvidia RTX 2080Ti GPUs SLI - the process typically takes 5-6 days for the whole dataset. With transfer learning, the pre-trained weights of YOLOv3 on COCO dataset is loaded[33] for parameter initialization. We use the mini-batch gradient descent with batch size $n = 16$ with the Adam optimizer and a starting learning rate $\eta = 0.0001$.[34] The learning rate is scheduled to reduce by a factor of 10 when performance stagnates. The training is guided by a standard loss function used in object detection, which combines the mean-square error of the locations of bounding boxes (L_b), the cross entropy of the confidence score (L_s) and the cross entropy of classification (L_c).[35]

$$\text{Loss} = \lambda_b L_b + \lambda_s L_s + \lambda_c L_c. \tag{1}$$

We set the relative weight factors to be $\lambda_b = 1, \lambda_s = 10, \lambda_c = 1$ in our training. To reduce overfitting, ℓ^2 regularization is applied. The regularization strength α and initial learning rate η are then tuned for model selection. The final performance of the model is evaluated by the mean average precision (mAP),[36] a numerical score within $[0, 1]$ with 1 being the best. The word "average" refers to a weighted sum of recall over all precisions and "mean" refers to average over all classes of objects (There is only one class in our case). The mAP is numerically computed from the area under the precision-recall (PR) curve, which is obtained by varying the threshold confidence score to assign a true positive detection case.

2.5 Sliding Window and Merging

After the CNN model for DOPAD is optimized, a given large TEM image of arbitrary size can be analyzed by sliding a small square window over the entire image and then merging predictions from all windows, which are similarly used in other problems.[37] In particular, if the original TEM image has $W \times H$ pixels, we apply the following workflow to extract locations of all NPs (Figure 2).

- choose the size p of the small square window with $p \times p$ pixels, such that the diameter of a NP in the image is about $1/10 - 1/4$ of the edge length of the square.

- choose the stride s to slide the window. We set $s = \frac{p}{n_s}$ with a small integer, e.g. $n_s = 2$ or 3.

- choose the margin size d for each window, e.g. $d = \frac{1}{16}p$. Any predicted bounding boxes whose vertex falls in the margin will be discarded. This helps to ignore truncated NPs at the edges of the window.

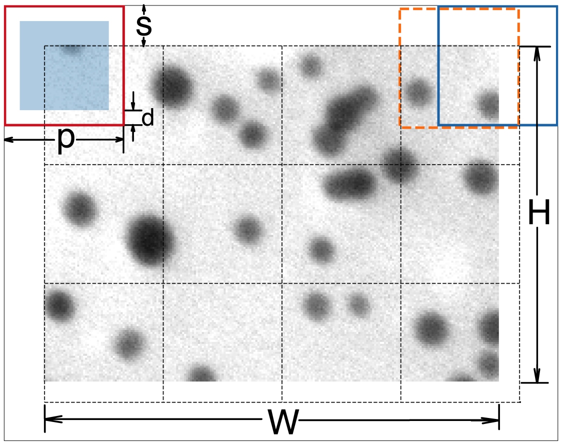


Figure 2: Sliding-window method. The original large TEM image is of dimension $W \times H$. The method slides small square windows of length p with stride s to cover the entire image. Zero paddings are added around the original image to allow an integer number of cuts. In each sliding window, predicted bounding boxes entering margins of width d are discarded. Only boxes within the shaded region are used.

- add zero padding around the original TEM image to make it of size $W' \times H' = (W + W_L + W_R) \times (H + H_U + H_D)$ pixels. The added left W_L, right W_R, up H_U, down H_D pixels should make sliding windows cover the whole image exactly. We set $W_L = s$, $W_R = p(\lfloor W/p \rfloor + 1) - W + s$, $H_U = s$, $H_D = p(\lfloor H/p \rfloor + 1) - H + s$.

- slide the $p \times p$ window over the padded image to take $n = n_W \times n_H$ small square views. With above choice of padding, $n_W = W'/s - (n_s - 1)$ and $n_H = H'/s - (n_s - 1)$.

- for each $p \times p$ square view obtained from the sliding-window method, rescale the image to 416×416 pixels and apply the CNN model to predict bounding boxes. This task can be paralleled in multicore processors since the temporal sequence to analyze different windows does not affect the final result. After discarding boxes that enter the margins of small squares, collect remaining boxes as final candinates into a set $\mathbb{C}$.

- apply non-max suppression algorithm to remove redundant boxes in $\mathbb{C}$ and obtain the set $\mathbb{B}$ of final predictions.

In set $\mathbb{C}$, it is expected that multiple bounding boxes enclosing the same NP are found from different sliding-windows. To remove redundant boxes such that only one box is predicted for each NP, the non-max suppression algorithm[38] is used. Once final bounding boxes are collected in set $\mathbb{B}$, the center of each NP can be readily extracted from box coordinates. The code then outputs coordinates (x_i, y_i) of each located NP in a text file, together with an estimate of its diameter $\sigma_i = \min\{w_i, h_i\}$. If unsatisfactory, the machine-predicted results can be further improved by human researcher, either by manually adding missing particles into the output file or interactively modifying bounding boxes via online labeler.[27] Technical details about our deep-learning predictor are contained on the GitHub site https://dopad.github.io.

3 Results and Discussion

3.1 Training CNN Model for Small Square Images in DOPAD

The CNN model is trained using 20%, 40%, 60%, 80% and 100% of the training set. Validation loss is typically minimized within ten epochs, as shown by the training curves in Figure 3. While significant overfitting exists for the smallest dataset (20%), the learning curve suggests that validation loss is reduced as the size of dataset increases. We expect collecting more images of similar type for this problem can further improve the performance of the model on unseen data.

Since overfitting is still obvious even with 100% training images, we increase the stength α of ℓ^2 regularization from default value of 0.0005 to 0.001 and 0.005. Starting learning rate η is also adjusted accordingly. The corresponding optimized validation mAP for each model with an intersection over union (IoU) threshold 0.5 is calculated from the precision-recall curve. Here, IoU is a standard metric within $[0, 1]$ used in object detection to quantify the similarity between the gound truth and predicted bounding boxes. The threshold 0.5 is a standard value chosen in object detection research to make benchmark comparisons. As compared in Table 1, the best-performed model with 96.2% mAP is trained at $\alpha = 0.001$ and $\eta = 0.0001$, which is used for further prediction and analysis.

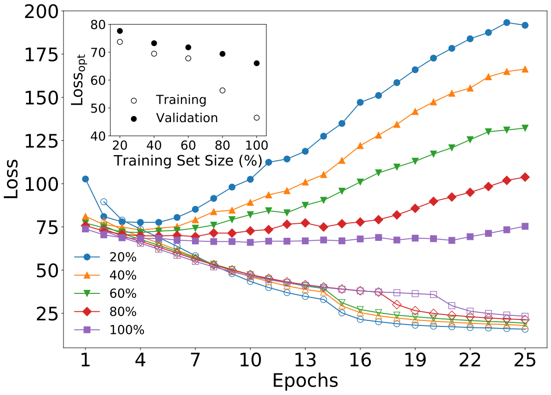


Figure 3: Training (solid symbols) and validation (empty symbols) loss as a function of learning epoch for different data size. Inset: optimized validation loss and corresponding training loss value as a function data size. $\alpha = 0.0005$ and $\eta = 0.0001$.

Table 1: Mean-average precision (mAP) with IoU threshold 0.5 on validation set for different regularization hyperparameter α and starting learning rate η.

α	η	mAP
0.0005	0.0001	91.8 %
0.001	0.0001	96.2%
0.001	0.00001	95.3 %
0.005	0.0001	92.9%
0.005	0.00001	95.1%

3.2 Performance on Large TEM Images from the Dataset

After the CNN model for DOPAD is optimized, we apply it on the original large TEM images in our dataset. Together with the sliding-window and merging protocol mentioned above, NP locations can be determined. An example TEM image with predicted bounding boxes is shown in Figure 4. Results from a standard tool in this field, ImageJ,[9] are also included for comparison. The NP loading in this image is moderate so the performace of ImageJ is acceptable. However, in areas with clustered NPs, our method is clear more accurate. ImageJ also fails to detect NPs when they are much darker than others. In addition, our method can better locate particle centers from the centers of bounding boxes, while ImageJ sometimes assigns particle coordinates at arbitrary locations in a dark area. To quantitative compare, after loading the optimized weight, our end-to-end prediction takes less than one minute on a GPU to locate 2745 NPs on this 6000 × 4800 pixel image, which is known to contain 2715 gound truth particles, giving a F1 score of 0.9597. ImageJ, by contrast, detects 1626 particles with a F1 score of 0.7473.

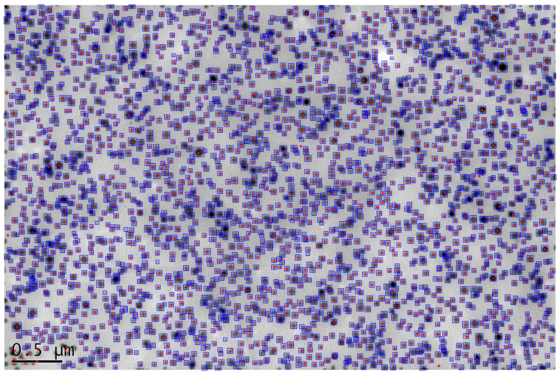


Figure 4: Deep learning predicted bounding boxes (blue) for NPs on a large TEM image, whose small cuts are used in training. Particle locations predicted by ImageJ (red dots) are also shown. The sample is 10wt% 50nm silica in 100kDa P2VP.

The average mAP of the prediction on all the 72 TEM images is 95.2%. More prediction examples are provided in Supporting Information and shared in the GitHub site of this work. Empirically, we find that the method performs less satisfactory in images where NPs are surrounded by white color air bubbles.

3.3 Performance on Unseen Images

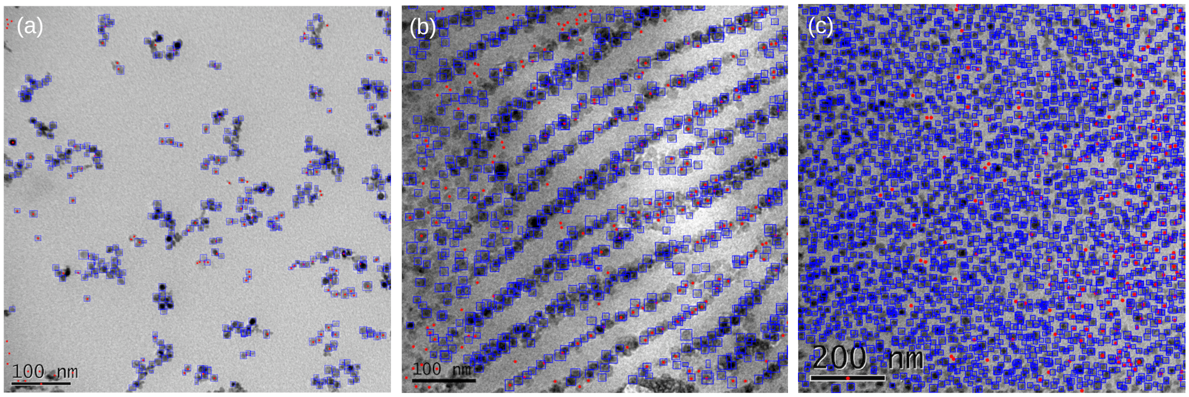


Figure 5: Predicted bounding boxes (blue) for NPs on unseen TEM images. Only a small portion of each full image is shown here. Particle locations predicted by ImageJ (red dots) are also shown. The samples are respecively (a) clustered 5wt% 11nm γ-Fe_2O_3 NPs in 115kDa PMMA, (b) striped 10wt% 14nm silica (PMMA-g-silica, 0.1ch/nm^2, 40kDa) in 100kDa PEO (*cryomicrotome, cryoTEM), and (c) high-density disordered 4wt% 14nm silica (PI-g-silica, 0.25 ch/nm^2, 38kDa) in 35kDa PI (*cryomicrotome).

The ultimate test of a supervised learning model is to examine its ability to generalize to new data, possibly from a different distribution. We thus apply our model to new TEM images that are not used in DOPAD to test its performance. To challenge the model, we select extreme cases with exotic structures, low contrast, or low resolutions. Examples are shown in Figure 5, which have NPs in cluster, stripe structures and extremely high loading. More prediction results can be found in Supporting Information and GitHub. When NPs agglomerate into clusters or stripes, ImageJ is unable to distinguish between them, while our method successfully detects most NPs. The performance of ImageJ also suffers from high loading, which leads to the overlap of NP images from other layers in the image. The variation of contrast in different regimes of the image presents another obstacle to apply ImageJ. In comparison, our model excels at NP detection in these extreme cases, even if they were not used in training and some of them cannot be clearly resolved by human eye.

3.4 Nanoparticle Size Distribution

The quantity that is easiest to obtain is the diameter σ of NPs, which is estimated by the size of the bounding boxes. This information can be used to draw the size distribution in each case. The histograms of NP sizes in the low-density (nominally 50 nm diameter NPs, Figure 4) and high-density disordered samples (nominally 14 nm diameter, Figure 5c) are fitted with log-normal distributions. It is often believed that the impact of Ultracut microtome would cause the measured particle sizes on 2D images to deviate from the true values in 3D samples. Here we find that, depite the interference of particle projections from different slices, the mean value predicted from deep learning analysis, 48.9 nm and 14.4 nm, are close to manufacturer specifications, 50±5 nm and 14±4 nm, respectively. This validates our method and gives us confidence in its predictions.

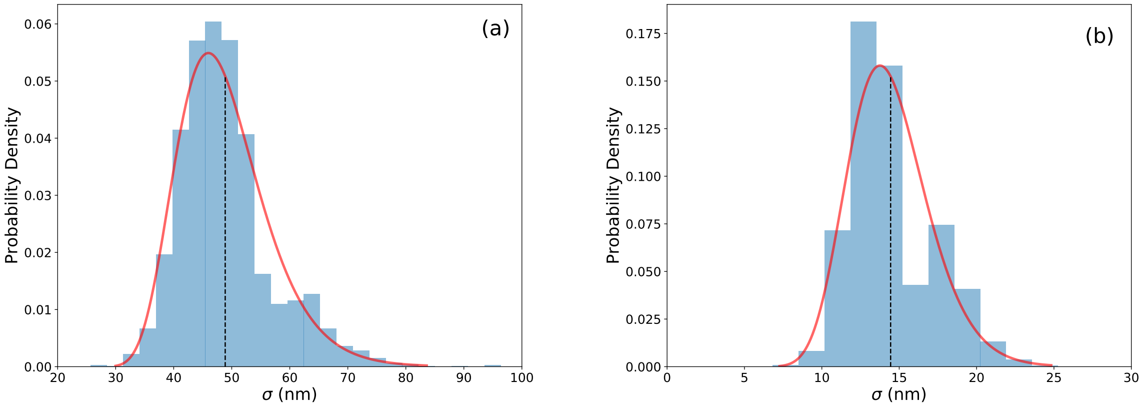


Figure 6: NP size distribution in the low-density (a) and high-density (b) disordered sample (histograms) fitted with a log-normal distribution (red solid line). The mean (vertical black dashed line) ± standard deviation of the fitting is 48.9 ± 7.9 nm and 14.4 ± 2.6 nm, respectively.

3.5 Assembly State of Nanoparticles

Extracting the locations of the NPs from TEM images allows us to efficiently and quantatively characterize their dispersion.[23] Besides microscopy methods, X-ray and neutron scattering are standard tools to resolve polymer composites structures.[39] One can compare experimentally measured scattering patterns with theoretical predictions, calculated from particle coordinates, to better understand the structural properties of the composites.[40] For example, the NP stripes aligned by polymer crystallization in Figure 5b give rise to two major peaks in the two-dimensional structure factor, $S(q_x, q_y)$, with these $\mathbf{q}$'s being perpendicular to the parallel stripes (Figure 7). At the two peaks, the magnitude of the scattering vector $q^* \approx 1.47\sigma^{-1}$, which corresponds to an interstripe spacing of $\lambda = 2\pi/q^* \approx 4.3\sigma$. Given that $\sigma = 14 \pm 4$ nm, this yields that the separation of the centers of NP layers is ≈ 60 nm. Previous, SAXS results had yielded similar numbers, validating this approach.[41]

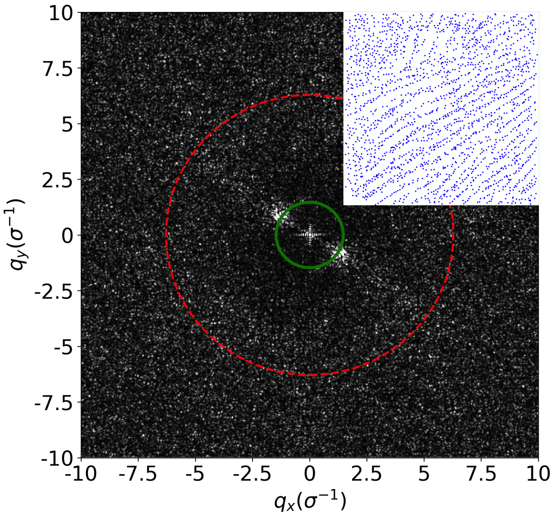


Figure 7: Two-dimensional structure factor $S(q_x, q_y)$ of NP stripes, whose coordinates are extracted by our deep learning method and shown in the inset. Two major peaks are observed in the direction perpendicular to the parallel striples at a radius $q^* \approx 1.47\sigma^{-1}$ (green solid circle). The scattering signal of amorphous packing around a distance of particle diameter σ is at a radius $2\pi/\sigma$ (red dashed circle).

To distinguish different disordered packings of NPs in Figures 4 and 5c, we focus on the particle number fluctuations $\sigma_N^2(R) = \langle N(R)^2\rangle - \langle N(R)\rangle^2$ within a circular window of radius R. This information allows us to understand the different spatial dispersion states of the NPs and in particular to distinguish between states that appear to have a uniform distribtion of NPs across the entire image.[42] In ordinary non-hyperuniform disordered samples, like liquids or glasses, particle number fluctuation grows as fast as window size, i.e. $\sigma_N^2(R) \sim R^2$. In hyperuniform disordered samples, fluctuations are suppressed at large length scales, which leads to the exotic consequence that the system becomes thermodynamically incompressible.

We calculate $\sigma_N^2(R)$ by randomly throwing 1000 circles of radius R on the image and counting the number of particles $N(R)$ in each circle. Because the finite image size imposes strong correlations between different sampling windows as their size approaches the image size, we only measure R up to a quarter of the minimum dimension of each image.

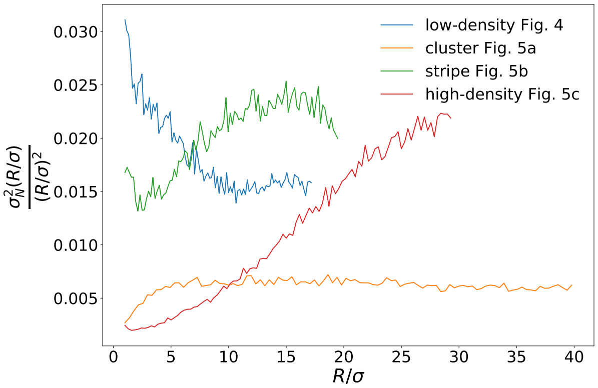


Figure 8: Particle number fluctuation $\sigma_N^2(R)$ as a function of the radius R of sampling circular windows for the low-density disordered packing in Figure 4 and cluster, stripe, high-density disordered packing in Figure 5. The curve for stripe is magnified by 10 times and that for high-density packing is magnified by 50 times to enhance visualization. Horizontal axis is rescaled by particle diameter σ in each image.

Among the four samples in Figures 4 and 5, the most interesing are the two systems which appear to have a relatively uniform distribution of the NPs. The sample in Figure 4, which corresponds to a well-dispersed mixture of NPs and polymer, shows that $\sigma_N^2(R)$ decays faster than R^2 at least at small R, implying that the system is apparently hyperuniform locally. This system thus appears to have suppressed concentration fluctuations at short scales, consistent with it being miscible. In contrast, the high-density disordered sample in Figure 5c is an ordinary liquid (not hyperuniform), i.e., $\sigma_N^2(R)$ grows faster than R^2. (It eventually decreases at the largest R, not shown, but that is because of the finite size of the TEM images.) Since we know that the latter sample corresponds to NP agglomeration due to phase separation from the polymer matrix, as deduced from larger field of views, it is apparent that the small R behavior of $\sigma_N^2(R)$ is an efficient means of distinguishing between the different states of NP/polymer miscibility.

We now consider the two other images. The clusters obtained for the iron oxide NPs (Figure 5a) show that $\sigma_N^2(R)$ scales as R^2 for all except smallest distances – thus, this system appears to have a liquid-like ordering of NP clusters. In the case with the string-like NP morphologies formed by PEO crystallization we find that $\sigma_N^2(R)$ increases as R^2 for small R but then start to decrease for larger R. While the initial increase is likely due to the anistropic ordering of the NPs, at larger scale fluctuations are suppressed. This likely speaks to the long-ranged order imposed on the NPs by the underlying semicrystalline polymer morphology. (This regularity is likely only relevant locally inside a spherulite.)

Although our method enables quantitative characterization of particle assembly states in 2D slices, it should be kept in mind that the genuine packing and hyperuniformity states in 3D samples need to be further examined, for example, from tomographic techniques.[43]

4 Conclusion

In this work, we propose a dataset of nanoparticle detection (DOPAD) built from experimental TEM images of polymer nanocomposites. A convolutional neural network model is trained on DOPAD to predict NP locations inside each small square image. In combination with sliding-window and merging algorithms, a large TEM image can be fed into the pipepline for end-to-end analysis, which automatically extracts NP locations and sizes. The method established here thus paves the way for efficient and quantitative characterization of NP dispersion. More exploring efforts can potentially make additional improvements to the model, for example, to include particle shape distortion during data augmentation or to fine-tune neural network architectures.

We validate the structural information resulting from this method against SAXS derived structural information for NPs ordered by polymer crystallization, and then use it to distinguish between different states of the assembly of polymer grafted NPs in a polymer matrix achieved by using their surfactancy. We show that this data-rich protocol allows us to draw critical facets of experimental behavior by reducing hours of manual work to minutes of automated processes. It would be interesting in the future to combine the fast 2D detector proposed here with tomographic TEM images to reconstruct 3D dispersion states of NPs in polymer matrix.

The current model, which excels at many NP detection tasks compared with traditional computer tools, is deployed on GitHub. Different from a shallow pixel classifier,[44] the performance of the deep neural network predictor can still be further improved by enriching the dataset to include more diverse cases, such as nonspherical particles, extremely large or small particles, and particles at image boundaries. Detection of particles of other shapes, such as nanotubes and nanolayers, can also be made possible, but the model has to be carefully retrained. To facilitate future development of this line of research, we also release the source code and dataset to the public.

Associated content

Supporting Information

The Supporting Information is available free of charge at

Full images for samples in Figure 4, 5a, 5b and 5c (pdf).

Author Information

Corresponding Author

Kai Zhang

Email: kai.zhang@dukekunshan.edu.cn

ORCID

Eric Z. Qu: 0000-0001-8816-869X

Andrew M. Jimenez: 0000-0001-7696-9705

Sanat K. Kumar: 0000-0002-6690-2221

Kai Zhang: 0000-0001-9700-7943

Author Contributions

E.Q. processed the image dataset and performed deep learning analysis. A.J. prepared experimental TEM images and performed ImageJ analysis. S.K.K. and K.Z. designed the project. All authors contribute to the writing of the manuscript.

Notes

The authors declare no competing financial interest. DOPAD with labeling, more sample images, source code, final weights of trained model are available at https://dopad.github.io.

Acknowledgements

E.Q. and K.Z. thank Duke Kunshan startup funding and Summer Research Scholars (SRS) program for supporting this work. A.J. and S.K.K. thank NYULMC DART Microscopy Lab for assistance in EM work. S.K.K. thanks the National Science Foundation for funding through the grant DMR-1829655.

References

(1) Schaefer, D. W.; Justice, R. S. How nano are nanocomposites? *Macromolecules* **2007**, *40*, 8501–8517.

(2) Moniruzzaman, M.; Winey, K. I. Polymer nanocomposites containing carbon nanotubes. *Macromolecules* **2006**, *39*, 5194–5205.

(3) Kim, H.; Abdala, A. A.; Macosko, C. W. Graphene/polymer nanocomposites. *Macromolecules* **2010**, *43*, 6515–6530.

(4) Tjong, S. C. Structural and mechanical properties of polymer nanocomposites. *Materials Science and Engineering: R: Reports* **2006**, *53*, 73–197.

(5) Kumar, S. K.; Benicewicz, B. C.; Vaia, R. A.; Winey, K. I. 50th anniversary perspective: are polymer nanocomposites practical for applications? *Macromolecules* **2017**, *50*, 714–731.

(6) Mackay, M. E.; Tuteja, A.; Duxbury, P. M.; Hawker, C. J.; Van Horn, B.; Guan, Z.; Chen, G.; Krishnan, R. General strategies for nanoparticle dispersion. *Science* **2006**, *311*, 1740–1743.

(7) Green, P. F. The structure of chain end-grafted nanoparticle/homopolymer nanocomposites. *Soft Matter* **2011**, *7*, 7914–7926.

(8) Ning, X.; Jimenez, A. M.; Pribyl, J.; Li, S.; Benicewicz, B.; Kumar, S. K.; Schadler, L. S. Nanoparticle Organization by Growing Polyethylene Crystal Fronts. *ACS Macro Letters* **2019**, *8*, 1341–1346.

(9) ImageJ. `https://imagej.nih.gov/ij/index.html`.

(10) Shimmin, R. G.; Schoch, A. B.; Braun, P. V. Polymer size and concentration effects on the size of gold nanoparticles capped by polymeric thiols. *Langmuir* **2004**, *20*, 5613–5620.

(11) LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* **1998**, *86*, 2278–2324.

(12) Güven, G.; Oktay, A. B. Nanoparticle detection from TEM images with deep learning. 2018 26th Signal Processing and Communications Applications Conference (SIU). 2018; pp 1–4.

(13) Madsen, J.; Liu, P.; Kling, J.; Wagner, J. B.; Hansen, T. W.; Winther, O.; Schiøtz, J. A deep learning approach to identify local structures in atomic-resolution transmission electron microscopy images. *Advanced Theory and Simulations* **2018**, *1*, 1800037.

(14) Wu, Y.; Ray, A.; Wei, Q.; Feizi, A.; Tong, X.; Chen, E.; Luo, Y.; Ozcan, A. Deep learning enables high-throughput analysis of particle-aggregation-based biosensors imaged using holography. *ACS Photonics* **2018**, *6*, 294–301.

(15) Ziatdinov, M.; Dyck, O.; Li, X.; Sumpter, B. G.; Jesse, S.; Vasudevan, R. K.; Kalinin, S. V. Building and exploring libraries of atomic defects in graphene: Scanning transmission electron and scanning tunneling microscopy study. *Science Advances* **2019**, *5*, eaaw8989.

(16) Okunev, A. G.; Mashukov, M. Y.; Nartova, A. V.; Matveev, A. V. Nanoparticle Recognition on Scanning Probe Microscopy Images Using Computer Vision and Deep Learning. *Nanomaterials* **2020**, *10*, 1285.

(17) Yao, L.; Ou, Z.; Luo, B.; Xu, C.; Chen, Q. Machine Learning to Reveal Nanoparticle Dynamics from Liquid-Phase TEM Videos. *ACS Central Science* **2020**, *6*, 1421–1430.

(18) Wang, Y.; Zhang, M.; Lin, A.; Iyer, A.; Prasad, A. S.; Li, X.; Zhang, Y.; Schadler, L. S.; Chen, W.; Brinson, L. C. Mining structure–property relationships in polymer nanocomposites using data driven finite element analysis and multi-task convolutional neural networks. *Molecular Systems Design & Engineering* **2020**, *5*, 962.

(19) Bruno, A. E.; Charbonneau, P.; Newman, J.; Snell, E. H.; So, D. R.; Vanhoucke, V.; Watkins, C. J.; Williams, S.; Wilson, J. Classification of crystallization outcomes using deep convolutional neural networks. *PLOS one* **2018**, *13*, e0198883.

(20) Schmidt, G.; Malwitz, M. M. Properties of polymer–nanoparticle composites. *Current Opinion in Colloid & Interface Science* **2003**, *8*, 103–108.

(21) Hanemann, T.; Szabó, D. V. Polymer-nanoparticle composites: from synthesis to modern applications. *Materials* **2010**, *3*, 3468–3517.

(22) Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv preprint arXiv:1804.02767* **2018**,

(23) Khare, H.; Burris, D. A quantitative method for measuring nanocomposite dispersion. *Polymer* **2010**, *51*, 719–729.

(24) Jimenez, A. M.; Zhao, D.; Misquitta, K.; Jestin, J.; Kumar, S. K. Exchange Lifetimes of the Bound Polymer Layer on Silica Nanoparticles. *ACS Macro Letters* **2019**, *8*, 166–171.

(25) Bonnevide, M.; Jimenez, A. M.; Dhara, D.; Phan, T. N.; Malicki, N.; Abbas, Z. M.; Benicewicz, B.; Kumar, S. K.; Couty, M.; Gigmes, D.; Jestin, J. Morphologies of Polyisoprene-Grafted Silica Nanoparticles in Model Elastomers. *Macromolecules* **2019**, *52*, 7638–7645.

(26) Akcora, P.; Liu, H.; Kumar, S. K.; Moll, J.; Li, Y.; Benicewicz, B. C.; Schadler, L. S.; Acehan, D.; Panagiotopoulos, A. Z.; Pryamitsyn, V.; Ganesan, V.; Ilavsky, J.; Thiyagarajan, P.; Colby, R. H.; Douglas, J. F. Anisotropic self-assembly of spherical polymer-grafted nanoparticles. *Nature Materials* **2009**, *8*, 354–359.

(27) Colabeler. `http://www.colabeler.com/`.

(28) Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. *arXiv preprint arXiv:1506.02640* **2015**,

(29) Redmon, J. Darknet: Open source neural networks in c. `https://pjreddie.com/darknet`, 2013.

(30) Keras: the Python deep learning API. https://keras.io/.

(31) Martin, A.; et al., Tensorflow: A system for large-scale machine learning. 12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16). 2016; pp 265–283.

(32) Zang, Z. YoloV3 Implemented in TensorFlow 2.0. https://github.com/zzh8829/yolov3-tf2, 2019.

(33) COCO dataset. https://cocodataset.org/.

(34) Kingma, D. P.; Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* **2014**,

(35) Zhao, Z.-Q.; Zheng, P.; Xu, S.-t.; Wu, X. Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems* **2019**, *30*, 3212–3232.

(36) The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Development Kit. http://host.robots.ox.ac.uk/pascal/VOC/voc2012/htmldoc/devkit_doc.html.

(37) Van Etten, A. You only look twice: Rapid multi-scale object detection in satellite imagery. *arXiv preprint arXiv:1805.09512* **2018**,

(38) Neubeck, A.; Van Gool, L. Efficient non-maximum suppression. 18th International Conference on Pattern Recognition (ICPR'06). 2006; pp 850–855.

(39) Roe, R.-J.; Roe, R. *Methods of X-ray and neutron scattering in polymer science*; Oxford University Press New York, 2000; Vol. 739.

(40) Li, D.; Zhang, K. Unifying the Concepts of Scattering and Structure Factor in Ordered and Disordered Samples. *arXiv preprint arXiv:2010.06126* **2020**,

(41) Jimenez, A. M.; Krauskopf, A. A.; Perez-Camargo, R. A.; Zhao, D.; Pribyl, J.; Jestin, J.; Benicewicz, B. C.; Muller, A. J.; Kumar, S. K. Effects of Hairy Nanoparticles on Polymer Crystallization Kinetics. *Macromolecules* **2019**, *52*, 9186–9198.

(42) Torquato, S. Hyperuniform states of matter. *Physics Reports* **2018**, *745*, 1–95.

(43) Kurita, R.; Weeks, E. R. Incompressibility of polydisperse random-close-packed colloidal particles. *Physical Review E* **2011**, *84*, 030401.

(44) Ilett, M.; Wills, J.; Rees, P.; Sharma, S.; Micklethwaite, S.; Brown, A.; Brydson, R.; Hondow, N. Application of automated electron microscopy imaging and machine learning to characterise and quantify nanoparticle dispersion in aqueous media. *Journal of Microscopy* **2020**, *279*, 177–184.