

Soft Matter

rsc.li/soft-matter-journal



ISSN 1744-6848

PAPER

Abdelwahab Kawafi, C. Patrick Royall *et al.*
Colloidoscope: detecting dense colloids in 3D with deep learning



Cite this: *Soft Matter*, 2025, 21, 6338

Colloidoscope: detecting dense colloids in 3D with deep learning†

Abdelwahab Kawafi,^{*a} Lars Kürten,^{id b} Levke Ortlieb,^c Yushi Yang,^c Abraham Mauleon Amieva,^a James Hallett^{id d} and C. Patrick Royall^{id *b}

Colloidoscope is a deep learning pipeline employing a 3D residual U-net architecture, designed to enhance the tracking of dense colloidal suspensions through confocal microscopy. This methodology uses a simulated training dataset that reflects a wide array of real-world imaging conditions, specifically targeting high colloid volume fraction and low-contrast scenarios where traditional detection methods struggle. Central to our approach is the use of experimental signal-to-noise ratio (SNR), contrast-to-noise ratio (CNR), and point-spread-functions (PSFs) to accurately quantify and simulate the experimental data. Our findings reveal that Colloidoscope achieves superior recall in particle detection (it finds more particles) compared to conventional methods. Simultaneously, high precision is maintained (high fraction of true positives). The model demonstrates a notable robustness to photobleached samples, thereby prolonging the imaging time and number of frames that may be acquired. Furthermore, Colloidoscope maintains small scale resolution sufficient to classify local structural motifs. Evaluated across both simulated and experimental datasets, Colloidoscope brings the advancements in computer vision offered by deep learning to particle tracking at high volume fractions. We offer a promising tool for researchers in the soft matter community. This model is deployed and available to use pretrained at <https://github.com/wahabk/colloidoscope>.

Received 5th November 2024,
Accepted 13th May 2025

DOI: 10.1039/d4sm01307g

rsc.li/soft-matter-journal

1 Introduction

In addition to being important materials in their own right,¹ colloids follow the same laws of statistical mechanics as do atoms and molecules,² yet with a timescale amenable to experiments and a length scale appropriate for analysis with optical techniques. Of the latter, light scattering of colloids then provides comparable information to X-ray scattering of atoms and molecules.³ However real space analysis using optical – and particularly 3D confocal – microscopy provides another approach. Tracking the colloidal particles in space and time, so-called particle-resolved studies (PRS), provides insight into phenomena generic to condensed matter, yet are hard to access with scattering methods.^{4–8} For this reason, particle-resolved studies of colloidal systems have long held a fascination well beyond the direct interest in colloids as soft materials. Among the highlights are the first direct observation of the birth of a

crystal nucleus⁹ and dynamical heterogeneity,^{10,11} understood to be important in the long standing challenge of the glass transition. The advent of active colloidal systems lends further insight to be gained from particle-resolved studies.^{12–18}

1.1 Classical colloid detection

At a technical level, the seminal work of Crocker and Grier¹⁹ for imaging of (quasi) 2D colloidal model systems has formed the backbone of much colloid tracking since. It is fair to say that despite ingenious developments using 2D tracking methods,²⁰ 3D systems continue to present great challenges to tracking compared to the case for 2D.^{4–8} For example, not all particles are tracked (low recall), “ghost particles” are erroneously identified (low precision), and all coordinates which are detected are subject to a tracking error (low average precision (AP)). These issues all become rather more significant when smaller particles or suspensions at high volume fraction are analysed.^{8,21} The precise volume fraction at which tracking may become problematic is of course highly sample specific. A suspension at a volume fraction above $\phi \approx 0.4$ would certainly be “high”. However in situations where particles come close together, such as gels,²² this means that even at lower volume fraction, tracking can be problematic.

Recently, the case for using smaller colloids has been made,²³ which provides access to new phenomena, particularly in the context of the glass transition.^{21,24,25} The Crocker and

^a School of Physiology, Pharmacology, and Neuroscience, University of Bristol, Bristol, BS8 1TD, UK

^b Gulliver UMR CNRS 7083, ESPCI Paris, Université PSL, 75005 Paris, France. E-mail: paddy.royall@espci.psl.eu

^c H. H. Wills Physics Laboratory, University of Bristol, Bristol, BS8 1TL, UK

^d Department of Chemistry, School of Chemistry, Food and Pharmacy, University of Reading, Reading, RG6 6AD, UK

† Electronic supplementary information (ESI) available. See DOI: <https://doi.org/10.1039/d4sm01307g>

Grier method has been implemented in the popular Python package Trackpy.²⁶ From here on this will be referred to as TP.

The method of Crocker and Grier¹⁹ constitutes a simple algorithm that takes the “pseudo diameter” (known as w) to be larger than the true radius but smaller than the true diameter of the particles. This is used to preprocess the image with a boxcar filter to remove the background followed by grayscale dilation²⁷ to detect the centroids. Then criteria such as the integrated sphere brightness, the radius of gyration of the particle image or geometric measures such as the eccentricity are used to refine particle positions.

More sophisticated approaches are also available.^{28,29} Among the pioneers of the field, van Blaaderen and Wiltzius³⁰ sought to improve the tracking accuracy in the axial direction by fitting a Gaussian to the integrated intensities in each plane that constituted a given colloid. This early work examined hard-sphere glasses and crystals, *i.e.*, colloidal solids in which diffusion could be neglected.³⁰ Even though this method made tracking easier, the slow scan rates needed to acquire images with low noise left dynamical information beyond reach. More recently, Jenkins *et al.*³¹ pushed the limits of the technique to identify contacts between colloids through ultra-high precision coordinate location. Such precision was achieved by first determining an empirical image of a colloid, which could then be compared to the original image. Further systematic improvements were made by Gao and Kilfoil.³² Despite advances in imaging technologies such as stimulated emission *via* depletion (STED),^{33,34} the software to detect the colloids has lagged behind.

Tracking colloids is often specimen dependant,^{35,36} leading to fragmented tracking methods used by different researchers. A further issue regarding particle tracking, that perhaps has received less attention than it might, is that it can be something of a “dark art”. Analysis can be slow, tedious, expensive, and remains subjective due to the extensive number of user-tuned parameters in tracking software. Many datapoints are discarded because they are of insufficient quality for tracking methodologies. Improved detection algorithms can allow one to use lower laser light, line averaging, resolution, and increasing frame rates. And in this way, we show that detecting smaller particles without needing improved imaging hardware becomes possible. There exists a need for the implementation of state-of-the-art computer vision methodologies for confocal microscopy that can be as specimen agnostic as possible. This desire to produce a method which removes the subjective tuning or parameters in existing approaches is a key aim of the present work.

1.2 Deep learning for dense colloidal suspensions

Before proceeding, we define two key characteristics of how successful the particle tracking is.

- *Recall* is the proportion of particles in the sample which are successfully detected.
- *Precision* is the fraction of those detected particles which correspond to (real) particles in the sample.

As proposed by Bailey *et al.*,³⁷ one might imagine that machine-learning techniques could be applied to the vexing problem of obtaining particle coordinates from microscopy

images. It is natural to enquire as to the opportunities offered by the explosion in machine learning (ML). ML has revolutionised image analysis in many settings from earth observation³⁸ to molecular imaging.³⁹ However, one area in which there is potential for development is in tracking of concentrated colloidal dispersions imaged in 3D.

Deep learning for computer vision classically involves convolutional neural networks (CNNs). Convolutions are used to replace the fully connected layers of the network by constraining neurons to a specific kernel size – *e.g.* 3×3 . This induces a “receptive field” where a hierarchy of neurons analyse each section of an image separately. This reduces the total number of connections and, therefore, parameters in the model, as well as providing translational invariance. We do not consider vision transformers due to their data inefficiency.⁴⁰

A common approach to detecting colloids in microscopy images is using convolutional neural networks (CNNs), particularly in semantic segmentation. In this method, each pixel of the image is classified into specific categories, such as particle (foreground) or solvent (background). Object detection techniques, such as bounding box detection and instance segmentation, are also frequently employed.^{36,41–44} These methods, including a Mask-RCNN, detect individual instances of objects by localizing them in the image. However, using Mask-RCNN for colloid detection typically requires a slice-based approach, where the model is applied to 2D slices one at a time. The individual slice predictions are then combined and used as labels for a Watershed algorithm. Benchmarks have shown that the 3D U-Net outperforms this slice-based Mask-RCNN approach.⁴⁵ This advantage is often attributed to the 3D U-net’s ability to leverage the contextual information across slices, whereas Mask-RCNN’s region proposal methods, which rely on fully connected layers, lose the benefit of translational invariance inherent to fully convolutional models.

In particle tracking, two main strategies are typically used. The first is semantic segmentation, which is commonly employed in applications like cell counting.⁴⁴ After segmentation, a post-processing technique, such as Watershed, can be applied to delineate distinct regions and extract particle positions. The second approach, which our work builds on, is heatmap regression. This method, widely used in dense pose prediction (*e.g.*, detecting body poses in images),⁴⁶ involves defining the problem as detecting a Gaussian distribution in the image and applying L_1 regression to fit the heatmap. Different labeling methods are discussed in the Methods section 2.3.

1.3 Previous deep learning approaches

A deep learning method for PRS has been developed by Newby *et al.*⁴³ This consists of a small fully convolutional and recurrent network trained on simulated training data. This method outperformed all other methods with high signal to noise ratios and at high recall. Moreover, it benefits from a recurrent approach where all frames are detected by the same model. Newby *et al.*’s method also managed to track bacteria, hinting at the possibility to generalise this method to heterogeneous datasets. However, this lacked the ability to predict the positions of particles in suspensions at high volume fraction.

The motion of colloids is an important feature to study. The movement of nearby particles can be correlated.^{47,48} Often deep learning based techniques actually use linking to improve segmentation in a similar fashion to Newby *et al.*⁴³ For instance, DistNet2D⁴⁹ predicts both the Euclidean Distance Map and Geodesic Distance to the Centre Map of the image. These are used as seeds for a Watershed algorithm followed by postprocessing to reduce false positives. Excitingly, DistNet2D uses temporal information by predicting 7 frames at a time (3 before and 3 after). The limitation of 7 2D frames is due to GPU memory, which highlights the difficulty in temporally tracking 3D data. Other methods exist for the detection of colloids using deep learning, but focus on different microscopy approaches such as holographic imaging or stochastic optical reconstruction microscopy (STORM).^{50,51}

This challenge of applying ML methods to PRS has thus received some attention. DeepTrack⁴⁴ has made great headway in tracking a wider variety of colloids with different shapes such as crescents. However, this method is focused towards dilute samples in 2D. As alluded to above, for colloidal systems, 2D analysis in particular is quite well dealt with using conventional methods. We believe that the case of 3D, hampered by the relatively poor resolution in the axial direction of confocal microscopy, is a significant challenge, especially for concentrated systems. Furthermore, DeepTrack does not provide pre-trained hyperparameters and takes the approach of training a unique model for each specimen. When compared to Newby *et al.*,⁴³ this brings up the question of: how much to generalise? Generalising to different datasets can aid in overfitting, but will it sacrifice precision or recall?

There is still not a sufficient neural network approach that accurately locates individual particles in a dense colloidal suspension in 3D, to the best of our knowledge. This is because obtaining labeled data, where the 3D fluorescent images are annotated with the exact 3D locations for each individual colloid, is difficult. The 3D locations from both existing

tracking software and human operators are not accurate enough to train a sufficient model.

1.4 Ground truth deficiency

From a ML perspective, particle resolved studies (PRSs) are a ground truth deficient problem, by which it is meant that there does not exist training data in which the solution (the coordinates of the particles) is precisely known from experimental data. To address the lack of ground truth for applications such as particle tracking, it is possible to use self-supervised learning, for instance³⁶ using geometric symmetry in sphere detection along with self-distillation to learn from each prediction. Another approach is to use generative adversarial neural networks (GANs) to create the training dataset.⁵²

The data deficiency can also be addressed by using simulated data to train the neural network. Applying molecular dynamics (MD) computer simulation, we can generate an ensemble of coordinates for densely packed particles, by repeatedly collecting particle locations from a simulated trajectory. With the simulated coordinates, we can further simulate their corresponding microscopy images, by convolving the pulse functions located at the coordinates with suitable kernels such as the point spread function (PSF).²⁸ A simulation approach facilitates investigating the model's sensitivity to the training data parameters.

1.5 Colloidoscope

Here we present a deep learning model to detect colloidal particles. We provide comparison with current methods of particle tracking from the wide application of confocal microscopy. We aim to: (i) provide a means to track colloids at high volume fraction (ϕ) with both high precision and recall without the need for extensive tuning of analysis parameters. (ii) Simulate a training dataset of spherical colloids of a wide variety but a focus on glasses. (iii) Train a model on the simulated dataset, for a tracking by detection method – *i.e.* by predicting one frame at a time. In this way, the U-net massively

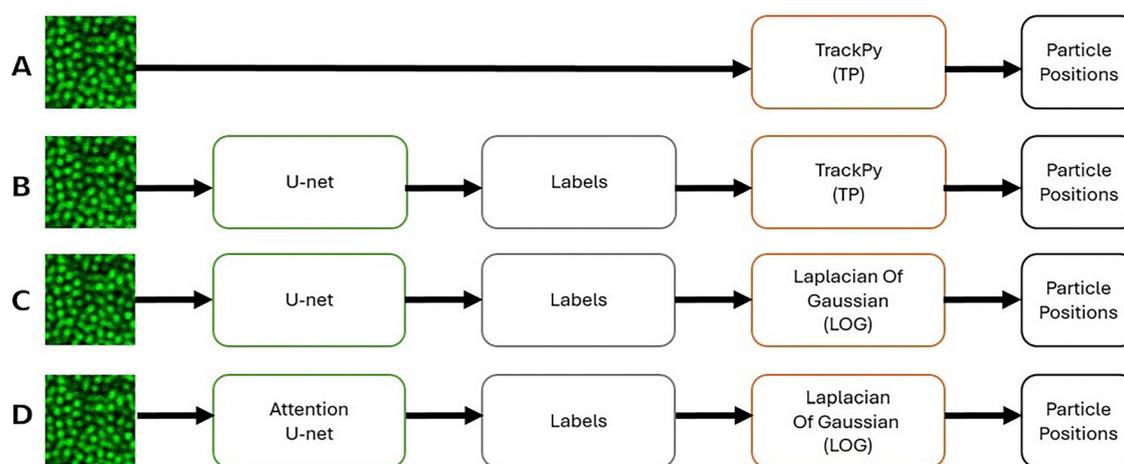


Fig. 1 Colloidoscope Schematic: (A) Trackpy (TP) works end to end, extracting the particle positions from the image. (B) and (C) Colloidoscope uses a U-net to label the image as a Gaussian heatmap, then TP or Laplacian of Gaussian (LOG) can be used as a postprocessing method to extract the particle positions. (B) is the default pipeline of Colloidoscope. (D) An Attention U-net is also tested with LOG postprocessing.

enhances the quality of the image from the point of view of particle detection. The resulting labelled image enables particle tracking to be more effective than the raw (or deconvolved) image. Fig. 1 shows the proposed benchmark and pipelines investigated. We begin by training and testing on the simulation, fine tuning the model on four challenging images of colloids. These colloids have different sizes and volume fractions, including a deconvolved image. Finally, we will test on a relatively simpler image to find how a U-net can improve on TP with or without its parameters being tuned for optimal performance.

2 Methodology

This work is available at Colloidoscope: <https://github.com/wahabk/colloidoscope>.

2.1 Model training, hyperparameter tuning, and data augmentation

This model is provided pretrained and does not require simulation or training by the user. The U-net was trained on an Nvidia A6000 with 48 GB of GPU memory and can be trained in a little over 10 minutes. Training is more demanding than inference, and inference is provided through Google Colaboratory which is equipped with a free GPU.

The model for this project is a simple 3D U-net.⁵³ A U-net comprises a fully convolutional autoencoder with long skip connections from the encoder to decoder that utilizes dense features without losing spatial information. A residual encoder is used⁵⁴ with a block approach to test model depth. Loss functions are crucial for training CNNs. They are particularly crucial for small and or dense particle detection, both of which are applied here. Since this model takes a heatmap approach, a simple L_1 loss function is used.

The volume of a sample which is imaged forms a compromise between resolution, working distance, as well as photobleaching. A confocal image is typically 16 megapixels (256 pixels cubed or larger) which constitutes our input. Such memory demand exceeds the capacities for most personal computing platforms – both the model and the image need to be run on a GPU for computational tractability. This requires storing both the model and the image in GPU memory simultaneously. Tiling – or a patch based pipeline – is crucial for this work, where each image is broken into smaller regions of interest for predictions. Tiling inference saves on GPU memory and allows for a larger extent of the image in the z direction (*i.e.* a greater number of xy image planes) when compared to a 2D model with slice-wise inferencing. This method is also more amenable to a larger batch size during training (for improved normalisation), as well as test-time augmentation.

An Attention U-net⁵⁵ was tested to measure its performance against the standard U-net. The original work introduces attention as a tool to combat class imbalance – defined as an imbalance of foreground and background in the image. This leads to poor model performance to the training biasing towards one class. Attention aims to bias the model towards the important pixels in the image regardless of class. Attention

could still be crucial for dense detection, and would be dependent on the labelling method (binary segmentation *vs.* heatmap). For instance, if each particle prediction is a Gaussian with a small fixed width (*e.g.* 5 pixels), class imbalance could still be an issue where most of the volume is background when ϕ is low. Conversely, in high volume fractions there might be negligible background in the image.

We optimised the parameters for model training with a grid search, to find out the best combination of learning rate, dropout fraction, activation functions and kernel sizes. Augmentation refers to supplementing the training data in a low-cost manner by *e.g.* flipping or zooming. While augmentation is used in sparsely labelled or data-limited applications, it still serves as a useful technique for regularisation (avoiding overfitting). The combination of computationally expensive simulation with cheap training-time augmentation allows for a robust model. Finally, histogram normalisation normalises brightness values which are usually skewed to the lower end to avoid photobleaching. A set of simulated and augmented images can be seen in Fig. A1 in the Appendix (ESI†).

2.2 Simulation of the training dataset

Due to the ground truth deficiency of this problem, the U-net must be trained purely on simulations. Each image can be different due to the system being imaged, the microscope, lens, or imaging parameters. The overall quality of an image is therefore an elusive measure. Brightness, noise, and contrast combine to determine the quality of the image. These are factors of the fluorescent dye, laser power, photobleaching, as well as detector sensitivity.

To begin, we generate particle positions for amorphous colloidal systems by using a hard-sphere Monte-Carlo algorithm.^{56,57} To begin, many random positions are generated at a low volume fraction. Then the system is “crushed”, where the volume is slowly decreased, with the particles being randomly moved until no overlap is measured. This process is then repeated until the desired volume fraction is reached. These coordinate sets were generated using Hoomdblue.^{58,59}

After the positions are generated at different volume fractions, to create the simulation the experimental data parameters must first be measured. The signal to noise ratio (SNR) is a useful measure for assessing signal quality. In signal analysis it is usually defined as the mean divided by the standard deviation. Since σ refers to the particle diameter, we define the *sample* mean brightness as \bar{x} with a standard deviation of s :

$$\text{SNR} = \frac{\bar{x}}{s} \quad (1)$$

While in signal processing this is a pure measure given the signal is constant, in images this is a more qualitative measure. Instead define $f_{\bar{x}}$ and f_s which are the foreground mean brightness and noise for the images.

$$\text{SNR} = \frac{f_{\bar{x}}}{f_s} \quad (2)$$

A more pertinent measure for object detection is the contrast to noise ratio (CNR). The CNR is commonly used in

biomedical imaging to analyse the quality of different imaging modalities such as CT, MRI, PET, or ultrasound. Through various simulated datasets we found that the CNR is a salient measure for image quality, specifically for confocal imaging for colloids where tuning the laser illumination is important for photobleaching. This not only includes the mean of the foreground brightness and foreground noise, but also the background mean brightness $b_{\bar{x}}$. All the particles in the same system are not the same brightness, with the most challenging detections being particles that are extremely close to each other or touching. The separation of these two particles will depend on the noise, but more so on the contrast between the foreground and background.

We define the CNR as:

$$\text{CNR} = \frac{|f_{\bar{x}} - b_{\bar{x}}|}{b_s} \quad (3)$$

with the absolute value expressed to enforce the CNR as a positive ratio. The foreground and background values are based on measurements from experimental data detailed in Section 3.1 and in Table S1 (ESI†).

With the CNR in hand, we proceed to generate simulated images from the coordinates obtained following the method above. We begin by convolving the “fluorescent cores” of the simulation with the point spread function to mimic the optics of the microscope. The PSF depends on many factors, including the excitation and emission wavelength, refractive-index of the particles and solution, numerical-aperture, magnification of the lens, as well as pinhole radius and shape. In practice, however, these factors have minimal effect on the size of the PSF. The ability to resolve particle coordinates ultimately rests on the size of the particle, but the PSF itself does not depend on particle size. The final blur is a function of the size of the PSF when compared to the size of the particle. The PSF of each lens system is of a constant size, but to investigate the effect of apparent particle radius in pixels, the simulated PSFs are resized by calculating the target pixels per nm.

A crude method to approximate a PSF is using a Gaussian blur [Fig. 2(B)], where the filter variance is larger in the z dimension.³⁰ An improvement is to use a least-squares Gaussian approximation method [Fig. 2(C)].⁶⁰ The same settings are used for all PSFs since parameters such as excitation wavelength contribute minimally to the PSF shape. An excitation wavelength of 488 nm was used. However, we are unaware of any methods for a least-squares approximation of a STED PSF. Instead, the PSFs were sourced from Huygens Professional version 22.10 (Scientific Volume Imaging, The Netherlands, <https://svi.nl>). Crucially for this work, the Huygens software provides STED PSFs depending on the power of the xy (doughnut) and z depletion lasers.

Once the SNR and CNR are characterised, the positions are generated, and the PSF is ready, the simulated images can be generated. This section will describe how the positions are used to draw a simulated image of size 64 pixels cubed seen in Fig. 2(G–L). This simulation is simple but allows for easily parameterised datapoints. To generate this kind of simulated image, we proceed as follows:

- We begin with an image twice as large as the target then later coarsen the image by a factor of two for aliasing. Padding is also added to allow the Gaussian blur and point spread function to be convolved in the image and to not lose the signal in the edges.

- Perlin noise⁶¹ is used with 4 octaves to generate the background noise, the brightness of which is derived from the CNR measurements. This corresponds to noise intrinsic to the background data, rather than in the imaging process‡.

- The positions and diameters are used to draw the particles in the image.

- The image is zoomed in or resized for aliasing (zooming is more computationally efficient than adding artificial aliasing).

- The PSF is convolved with the simulated image.

- The image padding is cropped.

- Finally, Gaussian noise is added to simulate the foreground (shot) noise. This simulates noise introduced in the image acquisition.

The parameters of the training dataset are sampled from the distributions described in the ESI† (see Table S1).

To isolate the effect of each simulation parameter, a randomly generated parameter set akin to the training dataset would be insufficient, since there would be too much noise in the data to isolate the effect of each parameter. A test dataset was created that isolates each parameter one-at-a-time. We chose to fix the other parameters at a level that remains at a challenging level for the U-net but not sufficient to limit the performance. These values were chosen to represent the periods with the most dramatic fall in precision/recall of TP against each parameter (see Table S2, ESI†).

2.3 Label simulation

Here the goal is to have high accuracy sub-pixel resolution coordinates of all the particles in the image. Generating the labels for the simulation will depend on the manner we wish to use our model whether it be a semantic map (binary) or a continuous heatmap. Tested are approaches by many heuristics. Previous methods use the different labelling approaches in Fig. 2(N–R). Cao *et al.*⁴⁶ (OpenPose) use an approach identical to Fig. 2N. Nibali *et al.*⁶² take an approach similar to Fig. 2O. On the other hand, Fig. 2Q shows how Newby *et al.*'s⁴³ model labels particles, while Deeptrack^{36,44} creates labels akin to Fig. 2R. These are tested to see which is applicable to 3D dense colloid detection. After testing these methods, Colloidoscope uses a Gaussian heatmap as shown in Fig. 2N.

To visualise examples, we simulate a random image and show multiple methods of drawing training labels in Fig. 2(M–R). The label can either be a smooth Gaussian heatmap or it can be a binary mask. The radius of the label can equal the radius of the particle [Fig. 2(N)], a varying radius can suit object detection problems for polydispersed suspensions. On the other hand, the label can be a smaller fixed radius for every particle - akin to a bullseye [Fig. 2(O) and (P)]. In Fig. 2(Q) and (R) shown are the

‡ Perlin noise corresponds to four greyscale values which are used for the image (Fig. 2(G)).⁶¹

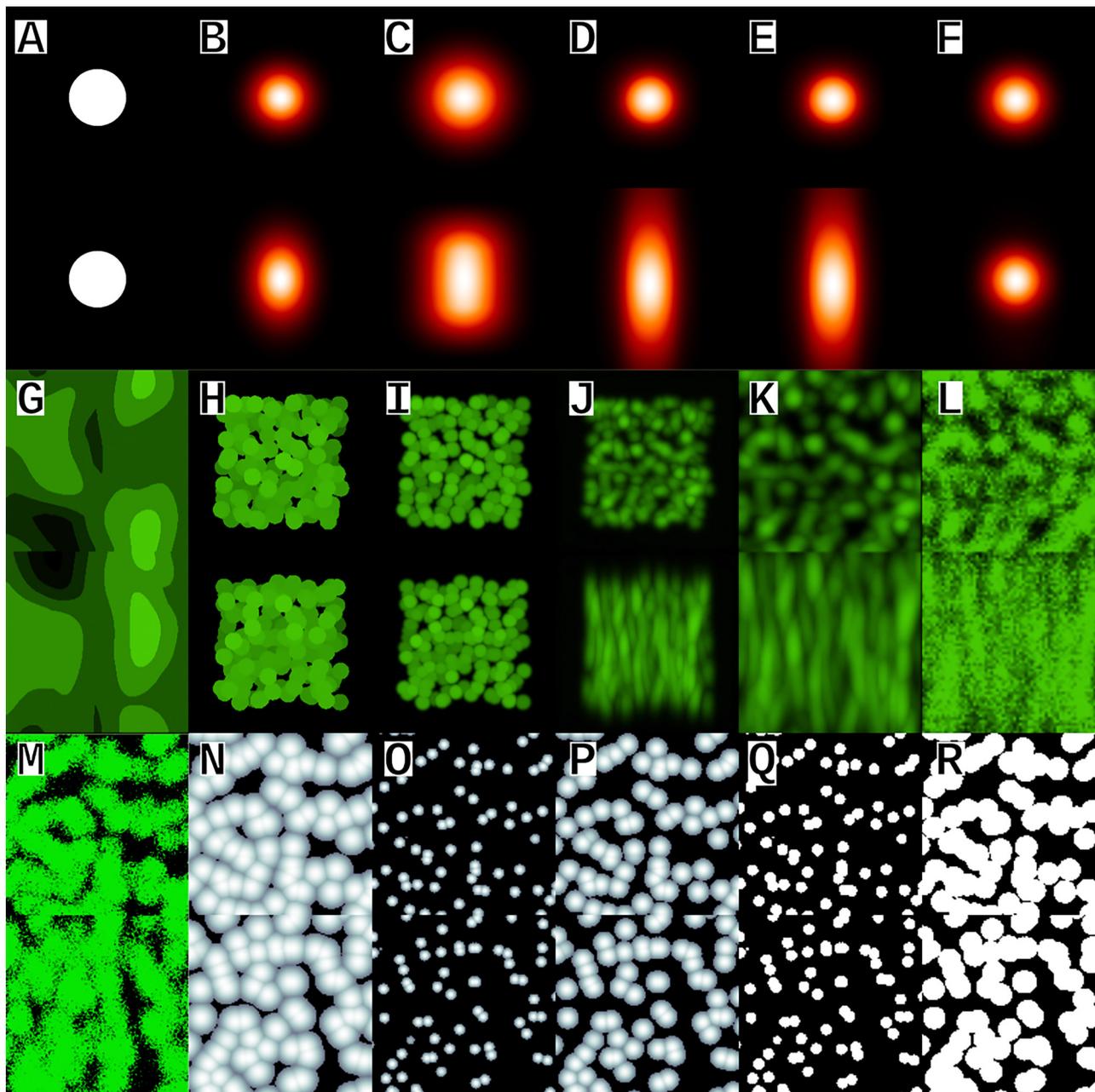


Fig. 2 The process used in Colloidoscope. Red: PSFs. Green: colloids, White: training labels. In each case, the xy plane is shown followed by, lower, an xz cut. (A)–(F) Approximations of confocal and STED PSFs. (A) Simulated particle of diameter = 10 pixels. (B) The particle (A) convolved with a Gaussian kernel of (9,5,5) (z,x,y). (C) A simulated particle of size 200 nm using the least-squares Gaussian method. (D) A simulated particle of size 200 nm using a Huygens STED XY PSF. (E) A simulated particle of size 200 nm using a Huygens STED 50XY 50Z PSF. (F) A simulated particle of size 200 nm using a Huygens STED Z PSF. (G)–(L) Colloid simulation steps. (G) Background. (H) Drawing spheres. (I) Aliasing. (J) Convolved PSF. (K) Crop. (L) Noise. (M)–(R) Gaussian heatmap regression and semantic segmentation for particle detection. (M) A simulated volume of colloids. (N) Gaussian heatmap label of (M) with varying radius $r = \sigma$. (O) Gaussian heatmap with constant radius $r = 4$. (P) Gaussian heatmap with constant radius $r = 7$. (Q) Semantic labels of with varying radius $r = \sigma$. (R) Semantic labels with constant radius $r = 4$.

usual binary semantic segmentation labels, representing foreground and background.

However, semantic labels overlap when it comes to closely packed targets. Therefore, we generate a Gaussian around each particle coordinate in the model. This Heatmap Regression approach (Gaussian labels with an L_1 loss function) is taken to aid in dense detection. Furthermore, a smooth Gaussian heatmap permits the use of an L_1 loss function for the model.

Regression loss functions can be more robust to data imbalance such as in very high or very low volume fractions when compared to binary mask pixel-wise classification.

2.4 Model post processing

The direct output of our model is a heatmap which encodes the information of particle coordinates. We need to apply extra processing to convert this output to particle coordinates. For

this conversion, we tested two approaches. Firstly, we used the Crocker & Grier algorithm implemented in TrackPy (TP) to process the model output. We also used a Laplacian of Gaussian (LOG) algorithm to process the model output. See Fig. 1 for the different approaches taken.

2.5 Accuracy metrics: average precision and $g(r)$

As discussed previously by Nibali *et al.*⁶² it is not possible to have a differentiable loss function derived from the Euclidean distances between the model predictions and the true locations since the number of particles cannot be determined *a priori*. Therefore, it is important to have intuitive and interpretable accuracy metrics that can offer an insight into model performance.

As noted above, to validate the model on simulated data where the ground truth is known, we use precision and recall. Precision describes the proportion of detections that are true, while recall measures the proportion of all true particles detected.

A common metric for object detection in machine learning is average precision (AP). However, here the important prediction is particle positions rather than the bounding box, since the particle size is already known even if the colloids are polydisperse. AP uses intersection over union (IOU) – the IOU takes the intersection of the prediction with the ground truth divided by the union of both. Instead of IOU, we simply use the distance between the prediction and ground truth normalised by the particle diameter in pixels. AP is usually referred to as AP_τ where τ represents the percentage of the diameter as which a prediction is regarded as a match or mismatch. We therefore compute AP as follows:

Distance matrix

$$D_{ij} = |p_i - \hat{p}_j| \quad (4)$$

Mismatch matrix

$$M_{ij} = \begin{cases} 1 & \text{if } D_{ij} > \tau \\ 0 & \text{if } D_{ij} \leq \tau \end{cases} \quad (5)$$

Precision

$$\text{Precision} = \frac{1}{\hat{N}} \sum_{j=1}^{\hat{N}} \left(1 - \prod_{i=1}^N M_{ij} \right) \quad (6)$$

Recall

$$\text{Recall} = \frac{1}{N} \sum_{i=1}^N \left(1 - \prod_{j=1}^{\hat{N}} M_{ij} \right) \quad (7)$$

Here, p is the true particle position out of a total N particles, and \hat{p} is the predicted particle position out of \hat{N} particles.

Precision and recall can be easily measured using the pairwise distances of the predictions and ground truths. AP is defined as the area under the precision and recall curve. The standard threshold is 50% of the particle diameter (AP_{50}). When simply AP is referenced, this is the combined average precision of eleven consecutive thresholds 0%, 10%, 20% ... 100%.

If this is a ground truth deficient problem, how can accuracy be validated for experimental data? For this we leverage the

simplicity of colloidal systems and the long history of structural measures applied to them.^{8,63} The radial distribution function (RDF), $g(r)$, is uniquely determined for an isotropic fluid with spherically symmetric pairwise interactions and can be predicted from theory with high accuracy in many cases.⁶³

We showcase the effect of poor detection on precision, recall, and the $g(r)$ in Fig. 3. In experiments, the $g(r)$ provides great insight into the nature of the prediction and can make biases in detection clear. For instance, if nearby particles are labelled too close, this will show as a peak before $r/\sigma = 1$ meaning a decrease in precision (more false positives, see Fig. 3(C) and (D)). If the $g(r)$ does not level off at unity for large r , this can hint at long length-scale inconsistencies in the detection such as clustering or boundary effects (see Fig. 3(B) and (D)). This can then be used to validate the detections of both simulated and experimental data.

In addition to the radial distribution function ($g(r)$), finer details can be probed with higher-order structural measures, which probe the correlations of multiple particles. A suitable method here is the topological cluster classification (TCC).⁶⁴ This identifies groups of particles whose bond topology (identified by a Voronoi decomposition) is identical to isolated clusters. When processing the experimental results where the ground truth for particle locations is not available, The TCC acts as a more sensitive measure for the PRS results compared to the RDF.

2.6 Colloidal suspension imaging

To validate our methods, it is important to evaluate the model on different experimental image specimens. This is especially important since the training is only done on simulations. In this work five different colloidal suspensions will be analysed. In particle-resolved studies of colloids, for refractive index-matched systems, early work used novel core-shell particles, where the fluorescent dye is located only at the centre of the particle.³⁰ This acts as a means to improve the precision and recall of the detected particle coordinates.

Here, we consider a range of experimental samples of different densities, media, sizes, and fluorescent dyes, and to provide evidence of overfitting (or lack thereof) and support the discussion of generalisability in particle tracking applications. These suspensions are refractive index and density matched to neglect opaqueness and buoyancy effects. Described in Table S3 (ESI[†]) are the composition, size, volume fraction, and polydispersity of the experimental data used to validate the model. A Leica SP8 3D STED II confocal microscope was used to image the specimens.

3 Results

In this Results section, we evaluate the ability of the U-net to detect the coordinates of colloids in confocal microscopy images. We highlight the enhanced capability to distinguish colloids in 4 challenging imaging scenarios when the model is used. We then showcase the model's performance on a relatively less challenging image. This marks a significant

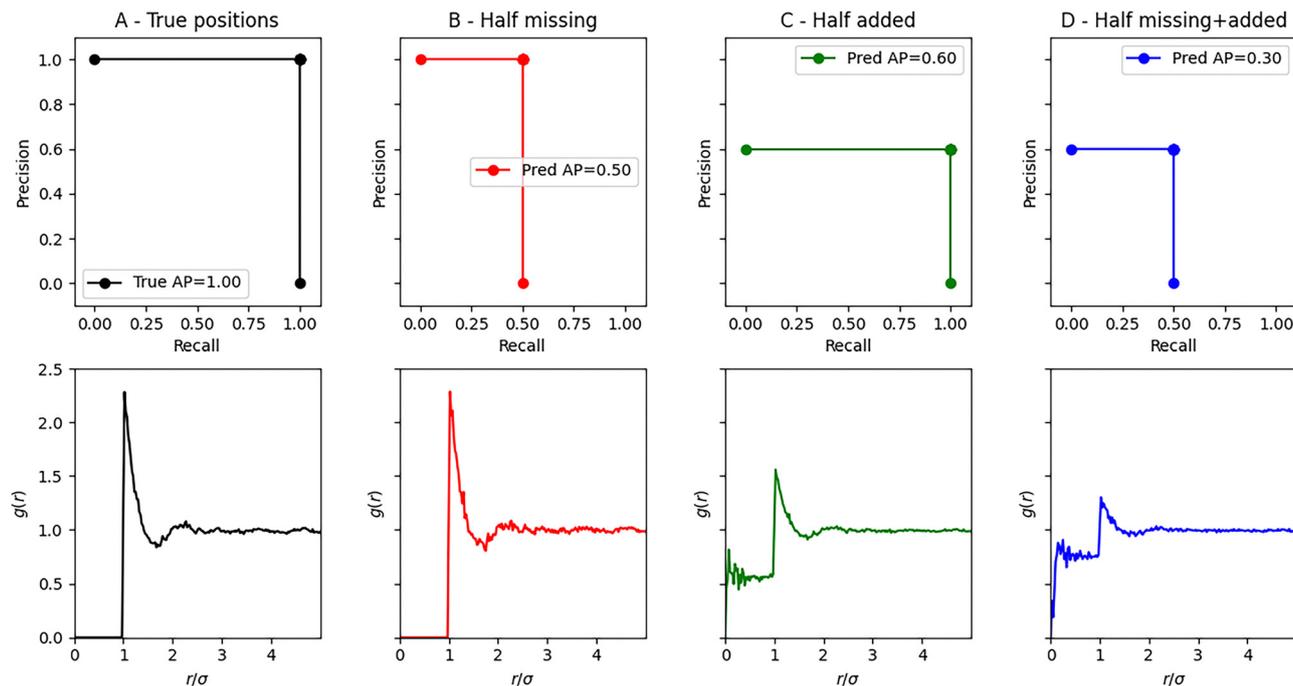


Fig. 3 Metrics: average precision (AP) of a simulated distribution to illustrate the effects of low precision and recall on the radial distribution function $g(r)$. (A) Control simulated detection. (B) Removing half the particles (false negatives) results in a recall of 0.5, this adds noise to the $g(r)$. (C) Adding half of the particles as ghost particles (false positives) results in a lower precision, and a signal in the $g(r)$ below 1. (D) Adding false positives and false negatives is also shown on the right.

advancement in automating the analysis of confocal microscopy data, demonstrating the model's effectiveness in real-world applications.

3.1 Experimental data measurements

To tune the simulation it is important to obtain a benchmark from the experimental data. This is particularly true for the SNR and CNR. To begin the brightness distributions of each of the 4 data points are plotted (see Fig. 4). A crude measurement of signal to noise ratio \bar{x}/s shows values between an SNR of 0.4 and 2. However, the distribution contains the brightness measurement for both background and foreground. This skews the distribution towards lower values and can be seen when there is particular imbalance in the data such as the sheet-like structures in Fig. 4(D).

Labelling the particles and background separately allows the comparison of their probability distributions (see the bottom row of Fig. 4). Since the values are being measured experimentally, false negatives are unavoidable, skewing the distributions closer together. The PSF is anisotropic so drawing perfect spheres of radius r would not capture the correct segmentation mask. Due to these challenges, we use the median of the brightness distribution which is a more robust estimator to outliers than the mean. One can find the CNR to be between 0.2 and 10. The laser illumination power is usually tuned so that particles are relatively dim, to improve the signal to noise ratio and avoid photobleaching. The gain is then adjusted to the maximum range of digitisation.

3.2 Evaluation – U-net

Over 20 datasets were simulated from the contrast and noise measurements to result in the final steps and parameter

distributions described in the methods. The U-net has been trained with an L_1 loss function over 1000 times across multiple grid searches. A 3D residual U-net with a block size of 2 (equivalent to a Resnet18 layer encoder⁵⁴) was trained with a learning rate of 0.002 for 5 epochs. A convolutional kernel⁴³ size of 7 improved accuracy particularly on large particles. Finally the activation was set to a SWISH function,⁶⁵ and instance normalisation⁶⁶ served as the activation function. Finally, the dropout was set to 0.2.

Using a heatmap approach, firstly the model is tested as a “de-noiser” for TP, wherein the model labels are fed to TP for post-processing as shown in Fig. 5. Over a parameter sweep TP maintains almost perfect precision. However, TP suffers from low recall usually only detecting 50% of the particles due to its assumptions and refinement steps. In particular, it fails in low brightness and dense systems where particle separability suffers.

The U-net improves the recall from TP across all simulation parameters while maintaining almost perfect precision. The volume fraction (ϕ) is the main objective and the U-net improves recall at $\phi = 0.55$ from 20% to 30% in simulations (see Section 1.2 for an explanation of recall and precision). The biggest improvements in detection are in detecting colloids with low contrast and low brightness, this is due to TP assuming the particles are in the brightest 30% of the image. While this parameter can be tuned manually and TP can usually perform well at low brightness, the U-net provides a simpler interface during inference and does not require manual tuning. Interestingly, both U-net and TP suffer on particles smaller than 10 pixels in radius.

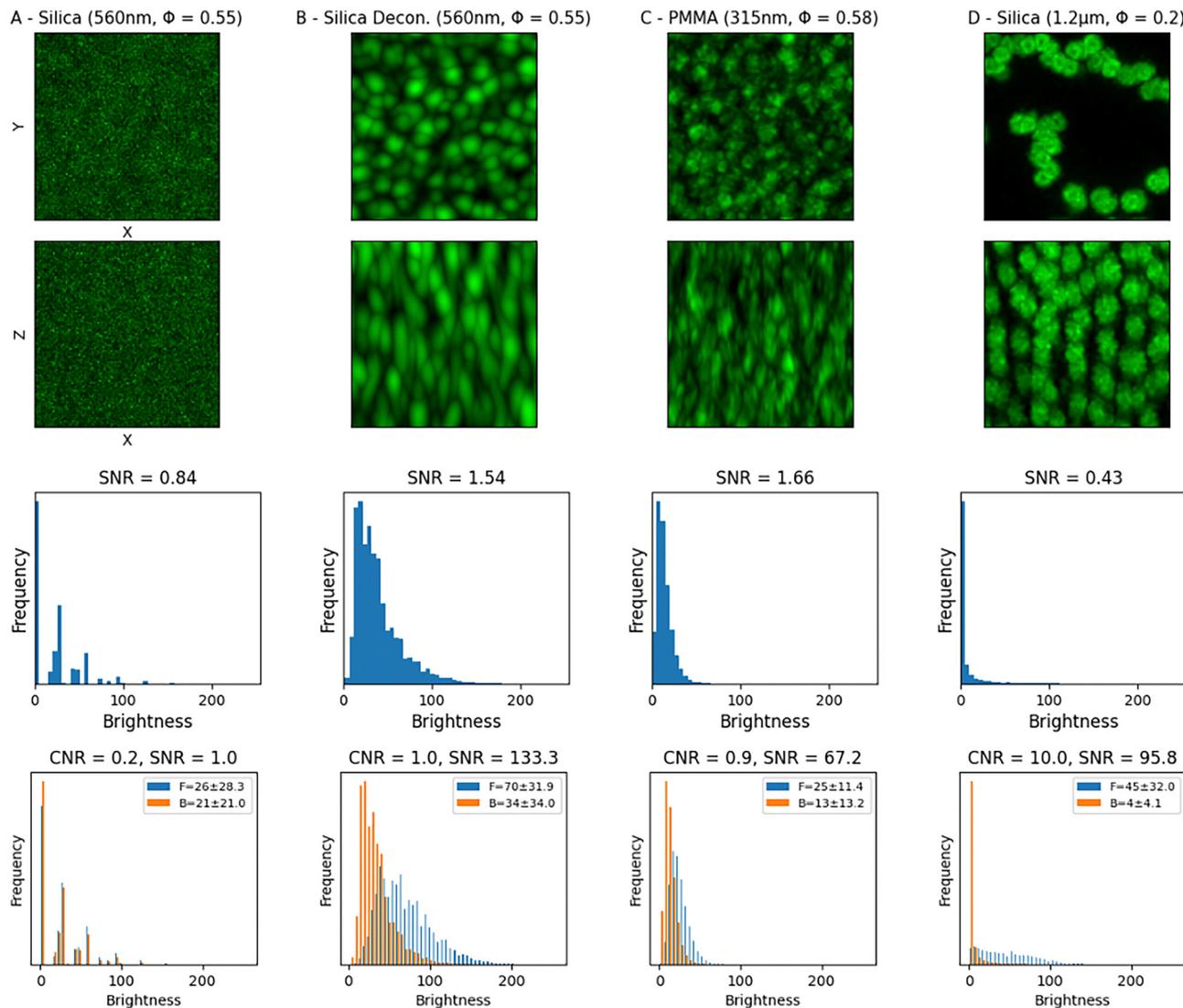


Fig. 4 Signal and contrast to noise ratio of the experimental data. A, B, C, D are different experimental systems as described in the figure. The first and second row indicate xy and xz views of the colloid images. The raw brightness distributions and SNR (\bar{x}/s) of each image are shown in the third row. Finally, the experimental images are separated into foreground (F) and background (B), whose brightness distributions are plotted independently in the bottom row. The legends indicate the mean \pm standard deviation.

The limited recall of TP motivates the use of a different postprocessing method. We used a Laplacian of Gaussian (LOG) blob detection algorithm which shows a tradeoff of slightly lower precision for a pronounced higher recall across simulated image parameters (see the bottom row of Fig. 5). Most notably LOG rescues the low recall on particles smaller than 10 pixels. LOG is also less consistent than TP, with a larger spread of recall. This hints at a lower quality of segmentation labels. The LOG algorithm does not contain any refinement steps, meaning the detections are more reliant on the quality of the model's heatmap labels.

3.3 Evaluation on experimental data

How do these improvements carry forwards with experimental data? While the actual precision and recall cannot be measured due to the ground truth deficiency, the total number of

particles detected and $g(r)$ can be used to analyse the quality of the prediction. Fig. 6 shows that there are more particles detected by the U-net, suggesting that the recall measured using Trackpy is again improved in every example of experimental data. More importantly, we find the $g(r)$ shows no signs of false-positive detections, since there is no signal in the radial distribution lower than the peak $r/\sigma < 1$.

The overall results, increased number of detected particles and the physically meaningful radial distribution functions, indicate that the Image \rightarrow U-net \rightarrow TP pipeline improved the PRS result across different experimental conditions.

In experimental data postprocessing with the Laplacian of Gaussian we find improved precision, but struggles on images of polydisperse systems [Fig. A2(C) and (D), ESI †] and dilute suspensions (Fig. A2E, ESI †). The radial distribution functions obtained from experimental data have higher first peaks in

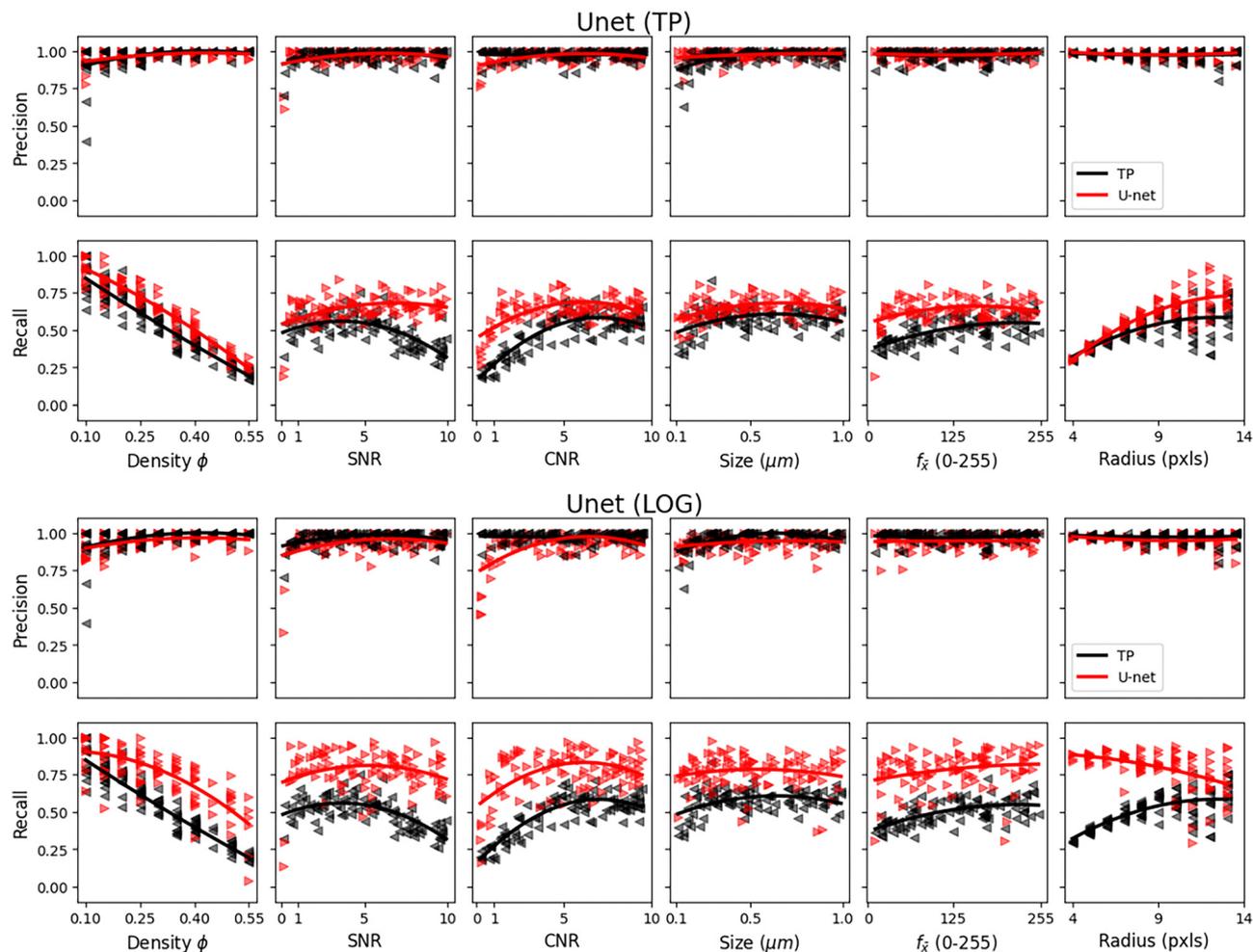


Fig. 5 U-net precision and recall (AP_{50}) against simulation parameters. The top and bottom figures show the results when using Trackpy (TP) and Laplacian of Gaussian (LOG) respectively to extract particle positions from the model labels.

Fig. A2(A–C) (ESI^{\dagger}), which suggests the model is detecting more of the particles closely packed together, furthermore, the $g(r)$ stabilises around 1, indicating there are no long range inhomogeneities in particle detection.

3.4 Evaluation: Attention U-net with Laplacian of Gaussian

To attempt to translate the benefits of Laplacian of Gaussian on simulations to experimental data, a different model is chosen to see whether increased heatmap quality can boost recall and precision. The Attention U-net claims to improve segmentation quality by guiding the model's output towards important pixels. It conserves more spatial information from the encoder to the decoder, and highlights the dense features important for detection. For colloid tracking an Attention U-net with LOG postprocessing shows very similar results on simulated data to a U-net – see Fig. A3 (ESI^{\dagger}). The Attention U-net does not show an improved recall on experimental data over a normal U-net. However, there is higher contrast in the predicted heatmaps which could make it easier for the post processing method. Overall, we find the Attention U-net improves the first peaks of the experimental pair correlation functions [Fig. A3(A)–(C), ESI^{\dagger}].

3.5 Evaluation using larger colloids: photobleaching and higher-order structural measures

Finally, after training and testing the model on the smaller particles, we chose a relatively simpler system of particles of diameter $2.0 \mu\text{m}$ at $\phi = 0.525$ with a polydispersity of 4%. We quantify the output of the U-net and Trackpy by comparing different observables of the model system with computer simulations.

For the Colloidscope approach we use only one parameter, the approximate particle diameter of 5 or 7 pixels for extracting the coordinates using Trackpy, and no image processing before tracking. With the TP approach, on the other hand, the intensity loss in the z -direction is first compensated and then a Gaussian filter is applied to make the image smoother. This increases the number of tracked particles and reduces the localisation uncertainty. In addition, an anisotropic diameter is used. Either 5 pixels in the xy -direction and 7 pixels in the z -direction, or 7 pixels in the xy -direction and 9 pixels in the z -direction. This ensures that the broadened intensity distribution of a single particle in the z -direction is not misinterpreted as two particles sitting on top of each other. Such false-positive identifications would lead to a first peak in the $g(r)$ curves at values $r \ll \sigma$.

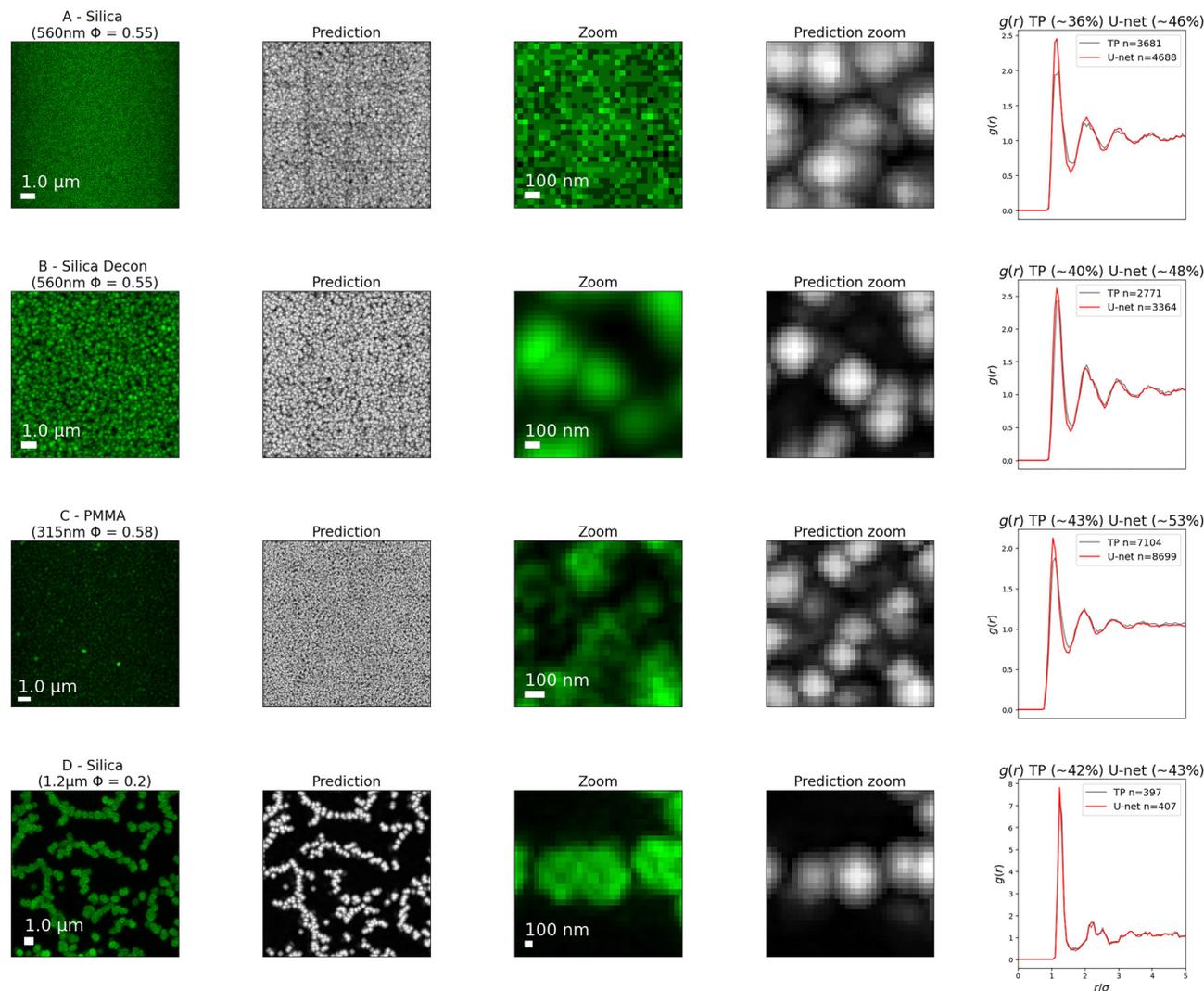


Fig. 6 Experimental predictions using TP as a postprocessing method for a U-net. (A) and (B) Experimental images of colloids. (B) Deconvolved (Decon.) volume of (A). C, D, other experimental systems. The images are shown as maximum projections followed by the model's predicted labels. A zoomed in maximum projection of a 32 cubed volume is taken around a random particle with the corresponding prediction shown on the right-hand side. The approximated percentage detected is derived from the experimentally prepared and predicted volume fractions.

First, we investigate the influence of photobleaching, *i.e.* the loss of signal-to-noise ratio and decrease in average brightness of the experimental image [see Fig. 7(a) and (b)] over the course of a 32 frames measurement (approx. 3 min). Fig. 7(d) shows the volume fraction calculated from the number of particles over the measurement period and the averaged intensity, which falls from approximately 98 to 60 for the 8-bit image. With the smaller diameter of 5 pixels, Colloidoscope overestimates the volume fraction due to some misidentifications. Colloidoscope with 7 pixels on the other hand tracks approximately 100% of the particles and therefore has a higher recall than TP with either of the two chosen diameters. As expected, the recall for TP with the smaller diameter is significantly better ($\phi = 0.51$). Approximately 98% of the particles are tracked. Compared to Colloidoscope with 7 pixels as the particle diameter, this translates to a loss of about 150 particles in each frame.

To quantify the localisation uncertainty of the tracking routines, the shape of the radial distribution functions is

analysed (Fig. 7c). The height of the first peak is significantly influenced by the localisation uncertainty of the particle positions. Colloidoscope and TP with the larger diameter of 7 pixels produce almost identical $g(r)$'s and agree very well with the computer simulations. However, the Trackpy approach with the smaller diameter of 5 pixels has significantly lower precision, which can be seen from the smaller first peak.

So far, we have considered pair correlations of particles through the radial distribution function $g(r)$. However, higher-order structural correlations provide a more detailed probe of the quality of particle tracking when benchmarked against simulation data. Correctly classifying these structures requires high accuracy detection with high recall. For example, in the case of the fcc cluster, a failure to identify one particle out of 13 will result in the cluster not being detected. The same applies to a particle that is localised with greater uncertainty.

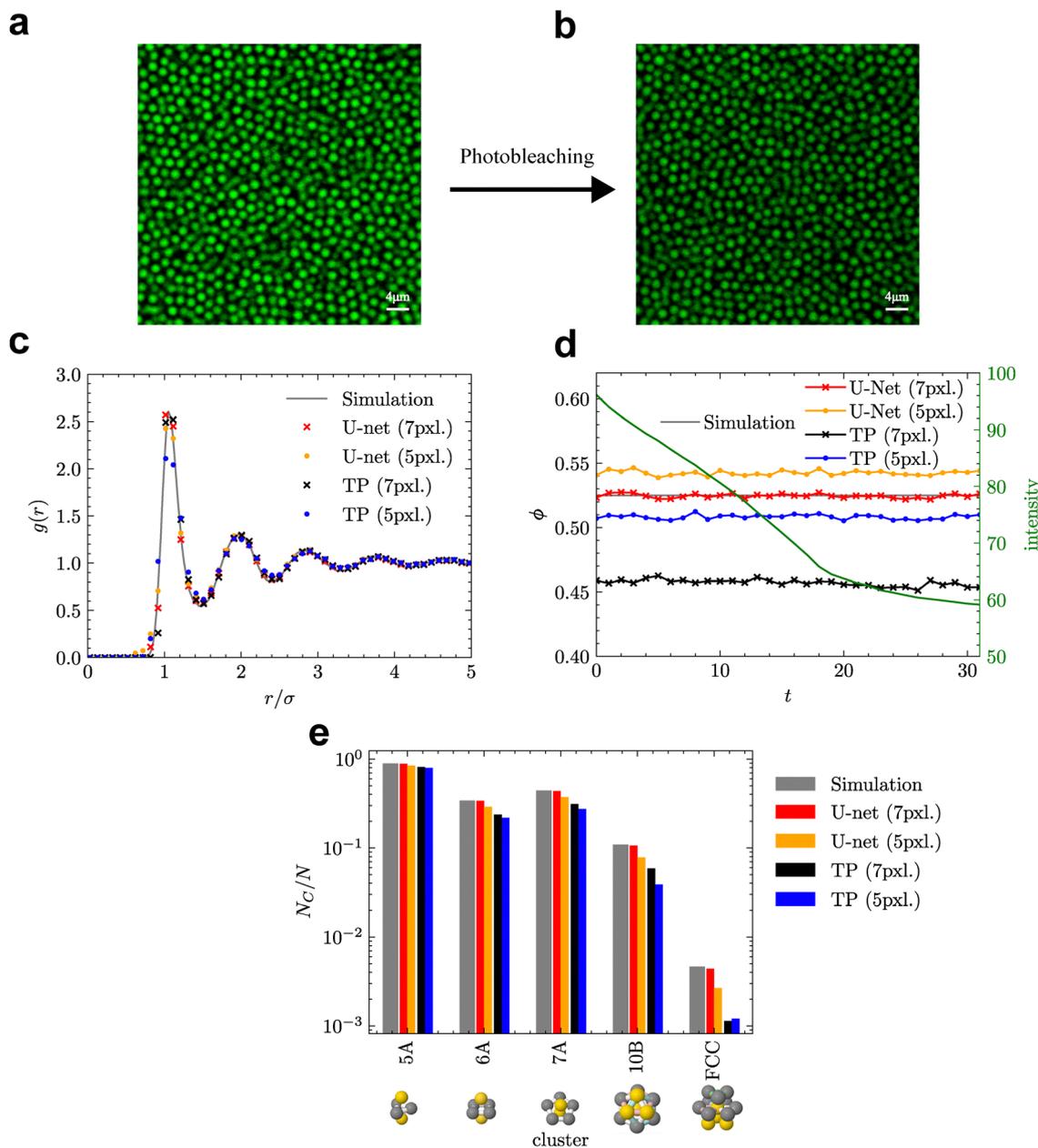


Fig. 7 The U-net applied to larger colloids: Photo-bleaching, radial distribution function and topological cluster classification (TCC) results for different parameters in Colloidoscope and Trackpy. (a) and (b) The effect of consecutively imaging a volume of colloids for 32 frames. (c) Comparison of the radial distribution function from computer simulations, U-net and Trackpy (d) quantified intensity (brightness) of the 32 images, with the predicted volume fractions of the U-net and Trackpy (ground truth $\phi = 0.525$). (e) Averaged cluster population from simulated and experimental data quantified with U-net and Trackpy.

We calculated the averaged population of different cluster types identified from coordinates produced by Colloidoscope and TP and compared them to hard sphere simulation data. Here we consider five clusters each consisting of m particles relevant to the system in question. These are the triangular bipyramid ($m = 5$), the octahedron ($m = 6$), the 7-membered defective icosahedron ($m = 10$) and the face-centred cubic crystal ($m = 13$). We see in Fig. 7(e) that Colloidoscope with a particle diameter of 7 pixels matches the cluster population of the simulation, maintaining a closer match than TP. Both TP

approaches deliver comparable results, which can be explained by similar influence of recall and precision on the cluster population. Unidentified particles and particles tracked with high uncertainty influence the population of higher-order structures in a similar way.

Comparing the results for recall and precision derived from the quantities shown in Fig. 7(c) and (d), one can see that for Trackpy there is a trade-off between the number of particles and the localisation uncertainty with which they are tracked. It opens an optimisation space for every parameter that can be

changed in the TP routine. This ranges from the experimental settings during image acquisition, to image preparation, to the selected particle diameters in TP.

To summarise, it can be said that TP can probably reproduce the particle positions produced by Colloidoscope. However, this requires careful fine-tuning of all parameters and probably post-processing in the form of manual filtering. In the test case we have considered, hard spheres, it is straightforward to compare against simulation data. In general, this is not always possible. This brings us to the question of how to actually know which parameters give “more correct” coordinates in the case of TP. An obvious quantity to tune against is the volume fraction. That is to say, to adjust the parameters until the desired volume fraction is obtained. One drawback of this approach is that the volume fraction itself is typically not known to high accuracy, with relative errors of 6% being routine^{67,68} (although very recently, improvements have been made⁶⁹). As discussed above, the radial distribution function $g(r)$ has limitations in assessing tracking quality, although higher-order correlations can shed more insight.⁷⁰ Overall then, such parameter tuning can lead to subjective results.

4 Discussion

The U-net consistently outperforms heuristics on both simulated and experimental data in different configurations. Particle resolved studies of confocal microscopy data for applications of soft matter physics requires the highest precision achievable. This work shows that deep learning approaches can achieve the robustness required for this analysis. Increases in precision and recall using the algorithm described directly lead to improvements in the $g(r)$ of experimental data.

The findings from simulated data generally translated to experimental data, images usually contain a bad mix of all image parameters (e.g. high density, high noise, and low contrast). This shows the challenging nature of designing simulation datasets. The simulated training dataset included volume fractions up to $\phi = 0.55$. Nevertheless the model still generalised well to experimental data up to $\phi = 0.58$ [see Fig. 6(C)].

Image quality is an elusive measure and combines all the parameters discussed here. The diffraction limit has been broken for single fluorophores, which may be detected to an accuracy of 20 nm in STORM imaging.⁵⁰ However, tracking in dense suspensions of colloids remains more challenging. Deconvolution can introduce artefacts which are challenging to simulate, with the model often failing to improve TP results on the deconvolved experimental data [see Fig. 6(A) vs. Fig. 6(B)] where, for example, Colloidoscope detects more particles (3767) from a raw image than Trackpy on a deconvolved image (2771).

Trackpy struggles with polydispersity since it needs the apparent diameter as an input parameter, as its underlying algorithm assumes all particles to be detected have the same size. The LOG can detect particles within a range which shows its superior performance on large and small simulated particles, as well as polydispersed volumes (see the right-most plots

in Fig. 5). The processing methods TP and LOG can give further insight into the performance of the model heatmap. In this case the U-net improved the performance of both postprocessing methods. For dense inference, the Attention U-net further matched the high precision of TP, with the benefits of LOG recall. While attention acts to increase model performance in situations with data imbalance (dilute suspensions), it also boosts performance on polydispersed data (Fig. A3, ESI†).

This model is deployed and available to use pretrained at <https://github.com/wahabk/colloidoscope>. We chose a standard U-net over an Attention U-net since we find it gives better results on data taken from suspensions at high volume fraction. The only parameter that is required is the diameter of the particles. The users are given the option of using TP or LOG, with the choice dependent on whether high precision or high recall is required, often LOG aids in detecting polydispersed colloids. This shows how Deep Learning can also be more accessible in deployment than previous methods. While DeepTrack³⁶ can detect many shapes of colloids (such as crescents), it requires the user to simulate their own data and train their own model, significantly hampering user-friendliness and adoption. This model is deployed pretrained and confidently generalises to all colloidal spheres tested. We set out to address the challenge of tracking colloids in 3d imaging at high volume fraction ($\phi \leq 0.4$). However, an interesting future direction would be to train a version of Colloidoscope for 2D data. A further possibility for future work with a 2D version of Colloidoscope could be to test errors using mean-squared displacement data, in that the error leads to a baseline in the MSD. 2D would be ideal here, due to the enhanced speed of imaging.

Limitations

Due to the inability to determine the number of particles *a priori* this model was not trained end-to-end (Fig. 1). Furthermore, another limitation of this approach is tracking by detection – *i.e.* every frame is analysed separately. Integrating the information across frames can improve model performance if a particle is out of view for only one frame.^{47,48,71} This would also aid in training, since the model is also learning the temporal information and could lead to a better inductive bias. A recurrent approach would also aid in deployment, removing the requirement for further postprocessing steps – mainly trajectory linking. However, the practical training of a recurrent network requires back propagation through time (BPTT) which, prohibitively, requires a much higher training compute budget.

Further network architectures are being developed to counter the difficulty in instance segmentation using embedding. The issue of determining the number of particles before prediction can be phrased in a second way. It is challenging to know which segmentations (foreground) belong to which particle. This is commonly named the data association problem, and is countered using output embedding. DSNT (Differentiable Spatial to Numerical Transform⁶²) embeds the output by predetermining which particle class goes in which output channel during training. Further approaches counter this by

semantically labelling the input, then discriminating different instances using clustering approaches.^{72,73}

While an Attention U-net was used, this model does not benefit from full self attention such as traditional transformers. Transformers were created to address limitations in maintaining context in recurrent neural networks. Furthermore, transformer architectures benefit from positional encoding of patches, instead of analysing patches without the context of surrounding ones. Embedding the input and output can act to alleviate many of the limitations encountered here.⁷⁴ However, transformers still require large datasets ranging from hundreds of thousands to hundreds of millions, which is beyond our computing budget.⁴⁰

5 Conclusion

In this study, we proposed a neural network-based particle resolved study pipeline, Colloidoscope. We compared Colloidoscope with an existing method TrackPy (TP) for their applicability in imaging dense colloidal suspensions.

- Colloidoscope outperformed TP by maintaining high precision and improving recall on simulated and experimental data.
- Colloidoscope demonstrated the ability to improve predictions in high colloid volume fraction and low contrast, thus expanding the scope of imaging and effectively cured the effect of photo-bleaching when images are collected over a long period of time.
- Furthermore, Colloidoscope offers a more user friendly interface with one single parameter, compared to TP. Colloidoscope easily generalized to multiple suspensions (both silica and PMMA colloids), and simply using raw images Colloidoscope outperforms Trackpy for deconvolved images (see Fig. 6A vs. Fig. 6B).
- Overall, the findings suggest that Colloidoscope is a promising alternative to TP for imaging colloidal suspensions with enhanced performance and user-friendliness.

Author contributions

This project was conceived by AK, YY and CPR. AK implemented the U-net and analysed the data. YY devised the position simulations for the training of the U-net. LK, LO, JEL, and AMA contributed data. All authors wrote the manuscript.

Data availability

The code for Colloidoscope is available for use at <https://github.com/wahabk/colloidoscope>.

Conflicts of interest

There are no conflicts to declare.

Acknowledgements

The authors would like to thank Mike Allen for helpful discussions during the course of this project. Jingwen Li is grateful to

Ioatzin Rios de Anda and Yushi Yang for assistance with experiments and theory, and is also thankful for the funding support from the China Scholarship Council. LK and CPR acknowledge the Agence Nationale de Recherche for grant DiViNew.

Notes and references

- 1 W. Russel, D. Saville and W. Schowalter, *Colloidal Dispersions*, Cambridge University Press, Cambridge, 1989.
- 2 R. Evans, D. Frenkel and M. Dijkstra, *Phys. Today*, 2019, **72**, 38.
- 3 P. N. Pusey, in *Introduction to Scattering Experiments, Neutrons, X-Rays and Light*, ed. P. Lindner and T. Zemb, Elsevier, 2002, ch. 1.
- 4 A. Ivlev, H. Löwen, G. E. Morfill and C. P. Royall, *Complex Plasmas and Colloidal Dispersions: Particle-resolved Studies of Classical Liquids and Solids*, World Scientific Publishing Co, Singapore, 2012.
- 5 U. Gasser, *J. Phys.: Condens. Matter*, 2009, **21**, 203101.
- 6 G. L. Hunter and E. R. Weeks, *Rep. Prog. Phys.*, 2012, **75**, 066501.
- 7 P. J. Yunker, K. Chen, M. D. Gratale, M. A. Lohr, T. Stil and A. G. Yodh, *Rep. Prog. Phys.*, 2014, **77**, 056601.
- 8 C. P. Royall, P. Charbonneau, M. Dijkstra, J. Russo, F. Smalenburg, T. Speck and C. Valeriani, *Rev. Mod. Phys.*, 2024, **96**, 2045003.
- 9 U. Gasser, E. R. Weeks, A. Schofield, P. N. Pusey and D. A. Weitz, *Science*, 2001, **292**, 258–262.
- 10 E. R. Weeks, J. C. Crocker, A. C. Levitt, A. Schofield and D. A. Weitz, *Science*, 2000, **287**, 627–631.
- 11 W. K. Kegel and A. van Blaaderen, *Science*, 2000, **287**, 290–293.
- 12 C. Bechinger, R. Di Leonardo, H. Löwen, C. Reichhardt and G. Volpe, *Rev. Mod. Phys.*, 2016, **88**, 045006.
- 13 S. A. Mallory, C. Valeriani and A. Cacciuto, *Ann. Rev. Phys. Chem.*, 2016, **69**, 59–79.
- 14 J. Palacci, C. Cottin-Bizonne, C. Ybert and L. Bocquet, *Phys. Rev. Lett.*, 2010, **105**, 088304.
- 15 A. Bricard, J.-B. Caussin, N. Desreumaux, D. Dauchot and O. Bartolo, *Nature*, 2013, **503**, 95–98.
- 16 A. Mauleon-Amieva, M. P. Allen, T. B. Liverpool and C. P. Royall, *Sci. Adv.*, 2023, **9**, eadf5144.
- 17 I. Buttinoni, J. Bialké, F. Kümmel, H. Löwen, C. Bechinger and T. Speck, *Phys. Rev. Lett.*, 2013, **110**, 238301.
- 18 N. Sakaï, K. Skipper, F. J. Moore, J. Russo and C. P. Royall, *Soft Matter*, 2025, DOI: [10.1039/d5sm00182j](https://doi.org/10.1039/d5sm00182j).
- 19 J. C. Crocker and D. G. Grier, *J. Colloid Interface Sci.*, 1996, **179**, 298–310.
- 20 E. K. R. Mackay, S. Marbach, B. Sprinkle and A. L. Thorneywork, *Phys. Rev. X*, 2024, **14**, 041016.
- 21 J. E. Hallett, F. Turci and C. P. Royall, *Nat. Commun.*, 2018, **9**, 3272.
- 22 C. P. Royall, M. A. Faers, S. Fussell and J. Hallett, *J. Phys.: Condens. Matter*, 2021, **33**, 453002.
- 23 O. Dauchot, F. Ladieu and C. P. Royall, *C. R. Phys.*, 2023, **24**, 25–56.

- 24 J. E. Hallett, F. Turci and C. P. Royall, *J. Stat. Mech.:Theory Exp.*, 2020, 014001.
- 25 L. Ortlieb, T. Ingebrigtsen, J. E. Hallett, F. Turci and C. P. Royall, *Nat. Commun.*, 2023, **14**, 2621.
- 26 D. B. Allan, T. Caswell, N. C. Keim, C. M. van der Wel and R. W. Verweij, *Zenodo*, 2023.
- 27 A. K. Jain, *Fundamentals of Digital Image Processing*, Pearson, 1989.
- 28 M. Bierbaum, B. D. Leahy, A. A. Alemi, I. Cohen and J. P. Sethna, *Phys. Rev. X*, 2017, **7**, 041007.
- 29 M. Leocmach and H. Tanaka, *Soft Matter*, 2013, **9**, 1447–1457.
- 30 A. van Blaaderen and P. Wiltzius, *Science*, 1995, **270**, 1177–1179.
- 31 M. C. Jenkins, M. D. Haw, G. C. Barker, W. C. K. Poon and S. Egelhaaf, *Phys. Rev. Lett.*, 2011, **107**, 038302.
- 32 Y. Gao and M. Kilfoil, *Opt. Exp.*, 2009, **17**, 4685–4704.
- 33 S. W. Hell, *Nanoscopy with Focused Light*, Nobel Lecture, 2014.
- 34 S. W. Hell, *Science*, 2007, **353**, 1163–1168.
- 35 J. Dong, F. Turci, R. L. Jack, M. A. Faers and C. P. Royall, *J. Chem. Phys.*, 2022, **156**, 214907.
- 36 B. Midtvedt, J. Pineda, F. Skärberg, E. Olsén, H. Bachimanchi, E. Wesén, E. K. Esbjörner, E. Selander, F. Höök, D. Midtvedt and G. Volpe, *Nat. Commun.*, 2022, **13**, 1–13.
- 37 M. R. Bailey, F. Grillo and L. Isa, *Soft Matter*, 2022, **18**, 7291–7300.
- 38 X. Yuan, J. Shi and L. Gu, *Expert Syst. Appl.*, 2021, **169**, 114417.
- 39 M. Ge, F. Su, Z. Zhao and D. Su, *Mater. Today Nano*, 2020, **11**, 100087.
- 40 A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit and N. Houlsby, *arXiv*, 2020, preprint, arXiv:2010.11929, DOI: [10.48550/arXiv.2010.11929](https://doi.org/10.48550/arXiv.2010.11929).
- 41 J. Redmon, S. Divvala, R. Girshick and A. Farhadi, *CVPR*, 2016.
- 42 K. He, G. Gkioxari, P. Dollár and R. Girshick, *IEEE International Conference on Computer Vision (ICCV)*, 2017.
- 43 J. M. Newby, A. M. Schaefer, P. T. Lee, M. G. Forest and S. K. Lai, *Proc. Natl. Acad. Sci. U. S. A.*, 2018, **115**, 9026–9031.
- 44 B. Midtvedt, S. Helgadottir, A. Argun, J. Pineda, D. Midtvedt and G. Volpe, *Appl. Phys. Rev.*, 2020, **8**, 011310.
- 45 A. Kar, M. Petit, Y. Refahi, G. Cerutti, C. Godin and J. Traas, *PLoS Comput. Biol.*, 2022, **18**, e1009879.
- 46 Z. Cao, T. Simon, S.-E. Wei and Y. Sheikh, *CVPR*, 2016.
- 47 C. Rodríguez-Cruz, M. Molaie, A. Thirumalaiswamy, K. Feitosa, V. N. Manoharan, S. Sivarajan, D. H. Reich, R. A. Riggleman and J. C. Crocker, *Soft Matter*, 2023, **19**, 6805–6813.
- 48 J. Clara-Rahola, T. A. Brzinski, D. Semwogerere, K. Feitosa, J. C. Crocker, J. Sato, V. Breedveld and E. R. Weeks, *Phys. Rev. E:Stat., Nonlinear, Soft Matter Phys.*, 2015, **91**, 010301.
- 49 J. Ollion, M. Maliet, C. Giuglaris, E. Vacher and M. Deforet, *PRX Life*, 2024, **2**, 023004.
- 50 E. Nehme, L. E. Weiss, T. Michaeli and Y. Shechtman, *Optica*, 2018, **5**, 458–464.
- 51 L. E. Altman and D. G. Grier, *J. Phys. Chem. B*, 2020, **124**, 1602–1610.
- 52 A. Zargari, B. R. Topacio, N. Mashhadi and S. A. Shariati, *iScience*, 2024, **27**, 109740.
- 53 O. Cicek, A. Abdulkadir, S. S. Lienkamp, S. Brox and O. Ronneberger, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016, 9901 LNCS, 424–432.
- 54 M. Drozdal, E. Vorontsov, G. Chartrand, S. Kadoury and C. Pal, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016, 10008 LNCS, 179–187.
- 55 O. Oktay, J. Schlemper, L. Le Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz, B. Glocker and D. Rueckert, *CVPR*, 2018.
- 56 M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids*, Oxford University Press, 2017.
- 57 D. Frenkel and B. Smit, *Understanding molecular simulation: from algorithms to applications*, Elsevier, 2nd edn, 2002.
- 58 J. A. Anderson, M. E. Irrgang and S. C. Glotzer, *Comput. Phys. Commun.*, 2016, **204**, 21–30.
- 59 J. A. Anderson, J. Glaser and S. C. Glotzer, *Comput. Mater. Sci.*, 2020, **173**, 109363.
- 60 B. Zhang, J. Zerubia and J. C. Olivo-Marin, *Appl. Opt.*, 2007, **46**(10), 1819–1829.
- 61 K. Perlin, *ACM SIGGRAPH Computer Graphics*, 1985, **19**, 22–26.
- 62 A. Nibali, Z. He, S. Morgan and L. Prendergast, *CVPR*, 2018.
- 63 J. P. Hansen and I. R. McDonald, *Theory of simple liquids*, Academic Press, 1976.
- 64 A. Malins, S. R. Williams, J. Eggers and C. P. Royall, *J. Chem. Phys.*, 2013, **139**, 234506.
- 65 P. Ramachandran, B. Zoph and Q. V. L. G. Brain, 6th International Conference on Learning Representations, ICLR 2018 - Workshop Track Proceedings, 2017.
- 66 D. Ulyanov, A. Vedaldi and V. Lempitsky, *CVPR*, 2016.
- 67 W. C. K. Poon, E. R. Weeks and C. P. Royall, *Soft Matter*, 2012, **8**, 21–30.
- 68 C. P. Royall, W. C. K. Poon and E. R. Weeks, *Soft Matter*, 2013, **9**, 17–27.
- 69 L. Kürten, A. Castagnède, F. Smallenburg and C. P. Royall, *arXiv*, 2025, preprint, arXiv:2503.17270, DOI: [10.48550/arXiv.2503.17270](https://doi.org/10.48550/arXiv.2503.17270).
- 70 R. Pinchaipat, M. Campo, F. Turci, J. E. Hallett, T. Speck and C. P. Royall, *Phys. Rev. Lett.*, 2017, **119**, 028004.
- 71 M. Gehrig and D. Scaramuzza, *CVPR*, 2023.
- 72 B. De Brabandere, D. Neven and L. Van Gool, 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2017, 478–480.
- 73 D. Neven, B. de Brabandere, M. Proesmans and L. van Gool, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2019, 8829–8837.
- 74 J. Hu, Y. Lu, S. Zhang and L. Cao, *IEEE Trans. Image Process.*, 2024, **33**, 2895–2907.