Title

Evaluation of particle tracking codes for dispersing particles in porous media.

Permalink
https://escholarship.org/uc/item/14k233ps

Journal
Scientific Reports, 14(1)

Authors
Berghouse, Marc
Miele, Filippo
Perez, Lazaro
et al.

Publication Date
2024-10-15

DOI
10.1038/s41598-024-75581-0

scientific reports

OPEN

Evaluation of particle tracking codes for dispersing particles in porous media

Marc Berghouse[1,2], Filippo Miele[3], Lazaro J. Perez[5], Ankur Deep Bordoloi[4], Verónica L. Morales[3] & Rishi Parashar[1]✉

Particle tracking (PT) is a popular technique in microscopy, microfluidics and colloidal transport studies, where image analysis is used to reconstruct trajectories from bright spots in a video. The performance of many PT algorithms has been rigorously tested for directed and Brownian motion in open media. However, PT is frequently used to track particles in porous media where complex geometries and viscous flows generate particles with high velocity variability over time. Here, we present an evaluation of four PT algorithms for a simulated dispersion of particles in porous media across a range of particle speeds and densities. Of special note, we introduce a new velocity-based PT linking algorithm (V-TrackMat) that achieves high accuracy relative to the other PT algorithms. Our findings underscore that traditional statistics, which revolve around detection and linking proficiency, fall short in providing a holistic comparison of PT codes because they tend to underpenalize aggressive linking techniques. We further elucidate that all codes analyzed show a decrease in performance due to high speeds, particle densities, and trajectory noise. However, linking algorithms designed to harness velocity data show superior performance, especially in the case of high-speed advective motion. Lastly, we emphasize how PT error can influence transport analysis.

Keywords Particle tracking, Porous media, V-TrackMat, Statistical comparison, Dispersion, Image analysis

Particle Tracking (PT) employs detection and linking algorithms to reconstruct the trajectories of objects within time-lapse image data. PT has vast applicability, spanning across any video data with moving entities, and it is one of the principal methods used to decipher microscale transport processes. This includes phenomena such as particle diffusion[1–3], nano and micro particle transport in saturated[4,5] or multiphases flow[6], bacterial dispersion[7–17] , chemotaxis[18], biofilm formation[19,20], viral transport[12], transport in porous media[8–11,21], colloid filtration[22], and computation of DLVO interactions through accurate trajectory analysis and measurements of hindered diffusion[23,24]. For a granular understanding of these processes, the precision and speed of PT are paramount.

The PT landscape boasts a plethora of open source and proprietary codes[25]. This multitude underscores the pressing need for robust comparative analyses between codes. Notably, the seminal comparative study in this domain centered on particles exhibiting Brownian and directed motion within open media[26]. However, a void persists in the exploration of PT methods tailored for particles navigating porous media flows. In these flows, spatial confinement (i.e. obstacles or grains) and dispersion result in complex flow paths, leading to pronounced variability in velocity fields over small temporal and spatial scales[27]. We should note here that particle tracking, in the context used throughout this paper, refers to time lapse image acquisition and subsequent particle identity assignment between consecutive frames[28].

The common strategy to reconstruct single particle trajectories by time lapse image acquisition first requires the particle detection in a single image at each frame and, then, the particle identity assignment (also termed as pairing or linking assignment) between particles detected in two consequent frames. Thus, key challenges in PT revolve around detection, localization, and linking errors[26,29]. Detection errors often stem from overlapping particles, particles out of focus, particles indistinguishable from the background, or varied particle sizes[30,31]. Linking errors can similarly be attributed to a confluence of factors: high particle speeds and densities, algorithmic inaccuracies, and preexisting detection errors[32]. Localization error predominantly emerges from the detection algorithm and the signal-to-noise ratio of the imagery[26]. Linking errors generally represent the most significant

[1]Division of Hydrologic Sciences, Desert Research Institute, Reno 89512, USA. [2]Graduate Program of Hydrologic Sciences, University of Nevada, Reno, Reno 89557, USA. [3]UC Davis, Civil and Environmental Engineering, Davis 95616, USA. [4]University of Lausanne, Geosciences and Environment, Lausanne 1015, Switzerland. [5]Civil and Construction Engineering, Oregon State University, Corvallis 97331, USA. ✉email: Rishi.Parashar@dri.edu

| https://doi.org/10.1038/s41598-024-75581-0

source of error, although large errors can occur for images where detection is especially difficult. Localization errors, which are often sub-pixel, minimally influence PT performance for a large group of particles.

Here, we compare the performance of four PT linking algorithms for a simulated dispersing of particles in porous media. To understand the impacts across a range of particle speed distributions, we simulated tracer particle dispersion in two different porous geometries (further discussed in Methods section). A critical facet of particle tracking is the particle spacing displacement ratio[33], $PSDR = \frac{ipd}{U\triangle t}$, where *ipd* is the average inter-particle distance detected within each frame (averaged over all frames), U is the average particle speed, and $\triangle t$ is the time interval between two consequent frames that corresponds to the inverse of the frame rate. Hence, PSDR is a measure of the mean particle spacing relative to the average jump length of particles between frames. This statistic can be considered a general constraint on the strength of PT algorithms, as it is directly related to the number of probable links each particle can make with other particles. For $PSDR \ll 1$, PT has been shown to be extremely challenging[34], and high particle densities have been show to increase the sensitivity of PT parameters[35]. As particles get very close together or have large displacements between frames, PT algorithms are not able to confidently determine accurate links to respective trajectories between frames. Thus, our analysis covers multiple mean speeds, particle densities, and speed distribution shapes to gauge PT codes across varied PSDRs (Table 1). Our approach not only evaluates PT codes for porous media, but also refines the standard PT comparison benchmark. We show that "classical statistics," which exclusively focus on particle localization, detection, and linking, might not penalize aggressive linking adequately. Specifically, classical statistics aren't effective for understanding the error associated with PT algorithms that "force" links between trajectories under improbable circumstances. To provide a more accurate comparison between PT methods, we use a suite of experimental statistics that offer significant depth in understanding of PT results compared to classical statistics. Moreover, we shed light on the potential for PT error to skew transport analyses of tracer particles in porous media.

Concomitantly, we unveil a novel PT code (V-TrackMat), tailored for microfluidics experiments, crafted by the collaborative efforts of some of the coauthors. The other three algorithms we tested for this paper (Trackpy, TrackMate-LAP, and TrackMate-Kalman) are described in further detail in the methods section. Our findings highlight V-TrackMat's ability to strike a balance between accuracy and judicious tracking at the expense of speed. Through this exploration, we endeavor to amplify the discourse in PT- emphasizing the shortcomings of traditional metrics, unveiling the intricacies of various PT algorithms, illuminating the impacts of PT error on transport analysis, and introducing a robust PT method.

Methods

Simulations

PT codes have been extensively compared for Brownian motion and constant-velocity motion in open media[26]. However, both 3D natural[36] and 2D engineered porous materials[37] are characterized by complex pore structures that result in broadly distributed velocity fields which are known to challenge tracking algorithms, so we chose to simulate dispersing particles in porous media for our comparison. We used simulations to create our ground truth imagery and trajectories, because although the gold standard for PT comparison is experimental data, there are no manually-labeled videos of dispersing particles in porous media that can be used as the ground truth.

For each of the 2D geometries (described in the paragraph below), we used OpenFOAM[38] to solve the flow fields, then calculated the pathlines via the Matlab function "interpstreamspeed" (Fig. 1a). The obtained flow fields exhibited large variations in speed distributions, resulting in simulations that we termed as "bimodal" and "unimodal". The pathlines were seeded with relatively equally spaced particles throughout the whole domain (with minor fluctuations due to grain positions), and their motion along the pathlines over time resulted in the final ground truth imagery and trajectories (Fig. 1b). A small amount of random movement (Gaussian distribution with $\mu = 0$ and $\sigma = 0.5$ pixels) was added on top of the ground truth advective trajectories. This amount of random motion can be considered a reasonable representation of diffusion in advection-dominated conditions ($Pe \approx 500$). Given a characteristic length of about 61.5 pixels (roughly equal to the average pore throat length), and a mean particle speed of between 2.6 and 10.2 px/frame, we can calculate $D = \frac{[2.6,10.2]\cdot 61.5}{500}$, which gives us a D in the range of $[0.32 - 1.25]\ \mathrm{px}^2/\mathrm{frame}$. Because we wanted to primarily focus on the

Bimodal	$S_p = 0.9$	$S_p = 2.6$	$S_p = 9.9$	$S_p = 19.6$
$\rho_p \approx 1.25e^{-4}$	42.8	15.5	4.6	2.5
$\rho_p \approx 2.50e^{-4}$	24.5	10.4	3.0	1.7
$\rho_p \approx 5.00e^{-4}$	14.9	6.6	2.3	1.1
Unimodal	$S_p = 0.8$	$S_p = 1.8$	$S_p = 6.7$	$S_p = 11.3$
$\rho_p \approx 1.25e^{-4}$	34.3	17.4	5.3	3.1
$\rho_p \approx 2.50e^{-4}$	22.8	12.6	4.0	2.4
$\rho_p \approx 5.00e^{-4}$	15.8	8.9	2.6	1.6

Table 1. PSDR values for all simulations in this paper.

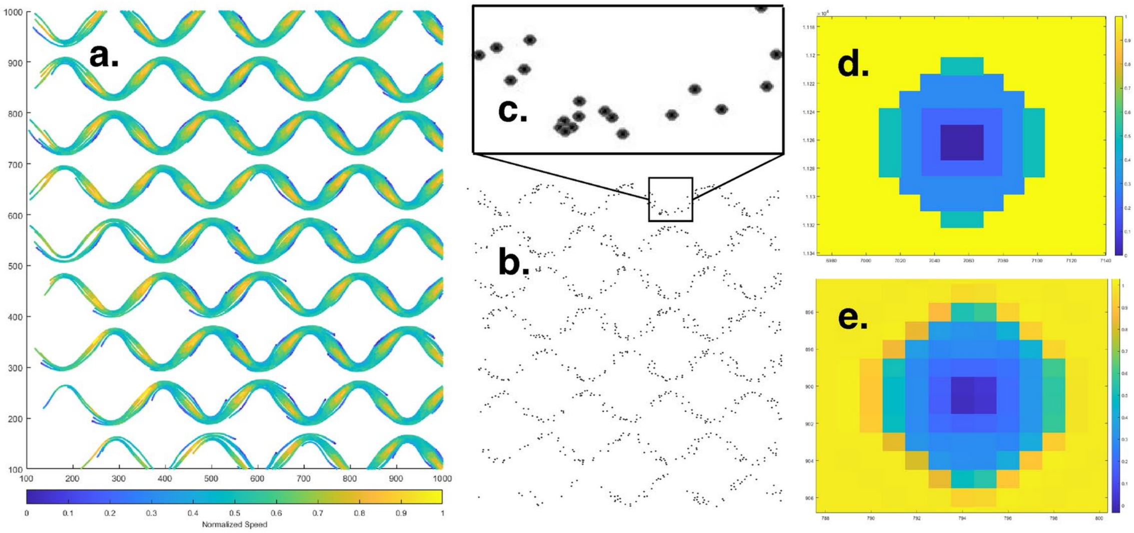

Fig. 1. This figure shows the general workflow for the construction of simulated trajectories. (**a**) Pathlines for the bimodal simulation at $PSDR = 1.1$ colored by normalized speed. (**b**) Simulated imagery analyzed by various particle tracking methods. (**c**) Zoomed in section of simulated particles showing overlapping particles. Most issues with detection occur due to this overlapping, which results in false negatives. (**d**) Particle at full resolution (10000x10000) during simulation creation. (**e**) Particle at final resolution (2000x2000) after interpolation.

linking abilities of different PT algorithms, we removed the background (cylinders of the geometry). However, due to overlapping particles in our simulations (Fig. 1c), we weren't able to completely remove the influence of detection accuracy. The particles with a diameter of 8 pixels (for a 2000x2000 pixel domain) were defined with a 2x2 black dot in the center with slightly increasing brightness for the surrounding pixels (Fig. 1d and e).

Our bimodal simulations were derived from experimental digital microscopy imagery of a quasi-2D porous media microfluidic device. The geometry of the microfluidic device, and thus our bimodal simulations, consists of a staggered array of equally sized and spaced grains (Figs. 1a and 2c), also referred to as a microfluidic lattice[8]. The unimodal simulation was derived from a geometry with similar average porosity, but with random placement and sizing of grains and pore throats (Fig. 2d). We term the first group of simulations as "bimodal" (Fig. 2a), because there are two clear peaks in the speed distributions for $\overline{S_{sim}} \geq 2.6$. We term the other group of simulations as "unimodal" (Fig. 2b), because there is only one clear peak at all mean speeds.

In addition to examining the impact of particle speed distribution on PT performance, we also vary the number of particles in our simulations. We use the initial number of particles at the beginning of each simulation to calculate the particle density ($\mathrm{particles/pixels^2}$). Particle densities decrease over time as individual particle trajectories disappear or exit the simulation bounds.

The primary goal of this paper is to provide a rigorous comparison of different linking algorithms for simulations of dispersing particles in varying geometry and at different particle density and speed. Although this does not cover the full range of variability expected in videos of dispersing particles, we chose to focus on PSDR and geometry because they highlight differences in the linking capabilities of each PT code. However, performance of PT codes can also be impacted because of limitations of methods and devices to capture and process videos of dispersing particles. These limitations could potentially inject random fluctuations and intermittency (blinking) in particle positions. Thus, we also provide an analysis of simulations where we increase the magnitude of the random displacement on top of our purely dispersing particles, and randomly set 2% of particles to be invisible for each frame. Specifically, we enhance the random motion by using $\sigma = 2$ pixels (instead of 0.5) in the method to represent normally distributed weak diffusion as described above. The displacements were restricted to be less than 6 pixels (three standard deviation). This random motion accounts for the combined effects of diffusion, camera jitter, and oscillations in particle brightness. The intermittency generally accounts for particles moving in and out of the focal plane of the camera, which can be caused by a variety of phenomena such as diffusion, particle-particle interactions, particle-wall interactions, and camera exposure time. The magnitude of the intermittency (2%) was calculated through an analysis of microfluidic experiments of colloids (detailed methods provided in Supplementary Methods). Both of these additions can be thought of as increasing the noise of the trajectories, so in this paper we refer to these experiments as the "noisy simulations". All other simulations in the paper, as described above, contain a minimal amount of noise to ensure tracking capabilities. We thus refer to the main simulations as the "minimal-noise" simulations when comparing their results with those of the noisy simulations.

Particle tracking

In order to focus on the depth of our PT comparisons, we chose to analyze four PT algorithms. For this study, we compared the outputs of TrackMate[39,40], Trackpy[41], and a new PT method developed by our co-authors named

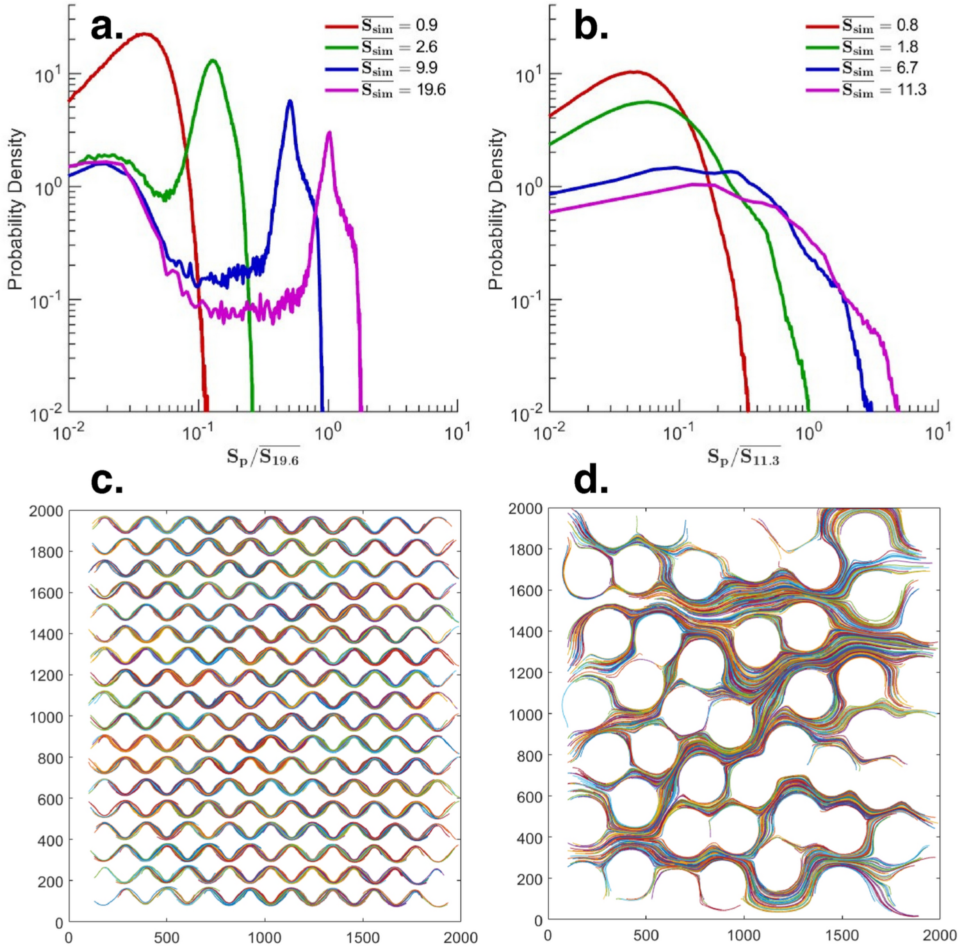

Fig. 2. This figure shows the particle speed distributions for the bimodal (**a**) and unimodal (**b**) simulations, and the simulated pathlines for a bimodal (**c**) and unimodal (**d**) simulation for $PSDR = 2.3$ and $PSDR = 2.6$ respectively. The speed distributions in (**a**) are normalized by the mean speed of the lowest-PSDR bimodal simulation (19.6 pixels/frame). The speed distributions in (**b**) are normalized by the mean speed of the lowest-PSDR unimodal simulation (11.3 pixels/frame). The simulated pathlines and variety of speed distributions illustrate the large range of conditions our PT codes were tested in.

"V-TrackMat". We chose TrackMate (TM) and Trackpy (TP) due to their high popularity in bio-image analysis, and because they use different linking algorithms. Each algorithm tested in our paper uses a nearest-neighbors-based method to link particles. To try and expand the variety of investigated algorithms, we also tested two deep learning (DL) methods[42,43], but found that they either couldn't be run on our hardware (more than 8 GB VRAM or too slow for CPU-based models) or did not perform as well as TM, TP, and V-TrackMat. Likely, the high resolution of our images (2000x2000 pixels), and small size of our particles (≈ 8 pixels) precludes the effective use of convolutional networks and other common DL-based architectures. However, we did not test any standard models for object detection such as Yolov8 or FairMOT[44], so it is possible there are available architectures that outperform the traditional methods. A more rigorous investigation of all available DL models is needed to determine state of the art performance, and thus develop an accurate comparison between traditional and DL-based methods.

TrackMate is one of the most popular methods for particle tracking in the field of biological imaging. TM runs through ImageJ[45], which makes it challenging to script PT analysis. However, TM has shown high levels of accuracy[26], and its use within ImageJ means it is well suited for quick analytical workflows where visual inspection of results is necessary. TM allows the user to pick from a variety of linking methods, but we only chose to analyze the Kalman and Linear Assignment Problem (LAP) methods. The Kalman method[46] uses the autocorrelative tendencies of trajectories to predict the velocities of particles, and therefore their positions in subsequent frames. The LAP method creates a cost matrix that finds the best match for each particle between two frames[47]. The cost matrix can be assigned additional variable-specific penalties that can improve linking, although this feature was not explored in our study. Both methods allow for gap-filling of particles that were missed in one frame and appeared up to a threshold number of frames after they were lost. TM also offers advanced filtering options that allows for easy removal of trajectories appearing to be erroneous. For example, TM allows for filtering based on the number of spots, track length, mean, min, and max speed, directional change rate, and linearity of forward progression.

Trackpy (TP) is another popular PT method written in Python, which makes it generally more scriptable than TM. However, this also means that some programming abilities are needed to effectively use this PT method. Furthermore, many of the trajectory filtering and visualization features that TM has are not part of the TP API, and would need to be manually coded from scratch. One area that TP excels in is its analysis functions. The API has functions to calculate pair correlations, MSDs, particle drift, van Hove correlations, and velocity correlations (amongst other functions). The linking algorithm for TP is based off the Crocker-Grier algorithm[48], one of the fundamental algorithms that many PT codes use in some variation. TP also has a special linking function that incorporates some velocity prediction element ("NearestVelocityPredict"), which was used for all our experiments. This velocity-based linking algorithm differs from the Kalman filter in that the Kalman filter considers the history of a trajectory (accounts for velocity variation in time), whereas the TP linking algorithm considers the velocity of the nearest particle (accounts for velocity variation in space).

V-TrackMat is a new Matlab-based PT method developed by some of our co-authors that has been successfully applied to tracking particles in three-dimensional and bioclogged environment[5,49,50]. The version of the code used in this paper can be found at https://github.com/mberghouse/V-TrackMat. The development of this code was motivated by the need for a customized MATLAB-based code for 2D and 3D PT to both allow a secondary linking phase between anachronistic trajectories and further overcome the current limitations of TM and TP for crowded suspension and long-time image acquisitions. Indeed, accuracy of TP has been reported to suffer for crowded suspension[51] while TM resulted in several crash episodes during the linking step for benchmark experiments of particle tracking in microfluidics-disordered media for a total number of frames above 2000 at $PSDR \simeq 1.4$. Although V-TrackMat has been developed for PT of 1 µm diameter latex particles in microfluidics application under laminar flow, it can be used for a variety of other PT applications. It first uses a *nearest-neighborhood* criteria by calling the *ipdm* routine between coordinates of centers detected in two consecutive frames, named as parents for particles detected at frame n and daughter for frame $n+1$. The pairing has been optimised by assuming that the frame rate is high enough so that the mean particle's jump is lower than the mean inter particle distance, mitigating the effect of intermittent behavior of a single particle's velocity under flow in confined media. Thus, the 2-frames velocity is computed for each pairing and the *ipdm* function is computed between daughters and the projection of the future position for the parents displaced by the quantity $v \cdot dt$ along the tangent direction. After the first loop over the full set of frames is computed, the set of reconstructed trajectories is then processed by gluing anachronistic trajectories in the 2D+2D space. This is reasonable for high Pe and low Re where particles passing through the same position with the same velocity are, in fact, following the same streamline. To glue trajectories, a pairing was first assigned by minimising distances between parent ending points and daughter starting points. The glue is accepted only for pairing whose distance is compatible with a jump allowed by both parents ending velocity and daughter starting velocity. For multiple pairings, the criteria of the minimum in of $|v_{p_f} - v_{d_i}|$ is then applied. This second loop of gluing anachronistic particles can potentially be iterated multiple times until no new pairings are assigned. The algorithm can be applied to track particles from time lapse images acquired over different fields of view. A common challenge for any PT code is the increasing memory cast with both the increasing time and new incoming particles as a new ID must be assigned while keeping track of the already existing ones. This means that the size of the matrix composed by the number of trajectories by the number of frames increases linearly with time and flow rate. To save computing time and avoid memory dredge, V-TrackMat code considers particles as lost if, for 5 consequent frames, no pairing has been assigned. In this case, V-TrackMat saves the trajectories on the fly into binary files and finally removes them from the matrix.

Although our goal in this paper was to test the ability of the linking algorithms for each code, it worth noting that detection is an important part of the PT workflow. V-TrackMat's detector was specifically designed to find bacteria in particular sets of experimental imagery, and we found that it did not perform well for the simulated imagery used in this paper. Both TP and TM use robust detection methods, and no significant differences were observed from each of the detection methods upon initial inspection. For V-TrackMat, we detected spots with TM, then exported these spots to Matlab to perform linking with V-TrackMat.

Comparative metrics

To understand the true performance of each PT method, we used multiple types of comparative metrics. We used "classical" statistical methods, which are similar to those described in[26]. In addition, we used experimental statistics and visual analysis of trajectories. The classical statistics used in this study are the false link rate (Fig. 3a), mean path length (Fig. 3b), and the Euclidean distance (Fig. 3c). These statistics are aimed at diagnosing basic problems with particle detection, linking frequency, and linking accuracy (the percent of correct links between two frames). Since we simulate the movement of tracer particles in a microfluidic device under constant flow, we also chose to use experimentally relevant statistics as PT comparison metrics.

To compare each PT code for the specific scenario of microbial transport, we used velocity autocorrelations $C_v(\tau) = \langle \mathbf{v}(t+\tau) \cdot \mathbf{v}(t) \rangle$, mean square displacements $MSD(t) = \frac{1}{N}\sum_{i=1}^{N}[r_i(t) - r_i(0)]^2$, and speed-angle joint probability density contours, which are calculated from the particle speed $v_p = \sqrt{(x_{t+1} - x_t)^2 + (y_{t+1} - y_t)^2}/\Delta t$ and turn angle $\alpha_t = \arctan(\frac{y_{t+2} - y_{t+1}}{x_{t+2} - x_{t+1}}) - \arctan(\frac{y_{t+1} - y_t}{x_{t+1} - x_t})$ (Fig. 3d) data, as our experimental metrics. The MSD and C_v for each PT output were calculated via the MSDAnalyzer[52], a companion postprocessing program for TrackMate. All other statistics were calculated via scripts written in-house.

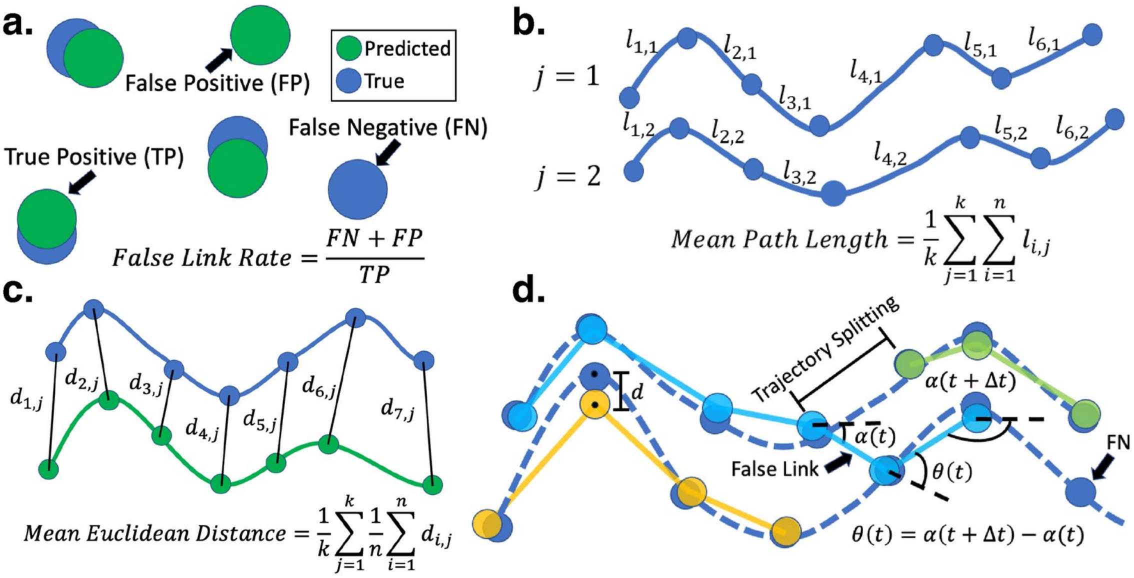
a.
False Positive (FP)
Predicted
True
True Positive (TP)
False Negative (FN)
$False\ Link\ Rate = \frac{FN + FP}{TP}$
b.
$l_{1,1}$ $l_{2,1}$ $l_{3,1}$ $l_{4,1}$ $l_{5,1}$ $l_{6,1}$
$j = 1$
$l_{1,2}$ $l_{2,2}$ $l_{3,2}$ $l_{4,2}$ $l_{5,2}$ $l_{6,2}$
$j = 2$
$Mean\ Path\ Length = \frac{1}{k}\sum_{j=1}^{k}\sum_{i=1}^{n} l_{i,j}$
c.
$d_{1,j}$ $d_{2,j}$ $d_{3,j}$ $d_{4,j}$ $d_{5,j}$ $d_{6,j}$ $d_{7,j}$
$Mean\ Euclidean\ Distance = \frac{1}{k}\sum_{j=1}^{k}\frac{1}{n}\sum_{i=1}^{n} d_{i,j}$
d.
d
Trajectory Splitting
$\alpha(t + \Delta t)$
$\alpha(t)$
False Link
$\theta(t)$
FN
$\theta(t) = \alpha(t + \Delta t) - \alpha(t)$

Fig. 3. This figure shows visual representations of all classical statistics, and one experimental statistic, used to compare particle tracking performance in this paper. In all panels, predicted points and trajectories are green, and ground truth points and trajectories are blue. (**a**) Snapshot of particles in a single frame to illustrate the false link rate (FLR). A false positive implies an erroneous point that was linked to another point in a previous frame. A false negative implies a missing link for a true point. Thus, FLR tests particle detection and missed links on a frame-by-frame basis. (**b**) Mean path length (MPL), shown here for two ground truth trajectories of length 7. Subscript i represents each successive line segment in time, and subscript j represents each trajectory. MPL tests missed links over time, also known as trajectory splitting. (**c**) Mean Euclidean distance (ED), shown here for one ground truth trajectory and one predicted trajectory (each of length 7). ED tests localization, detection, and linking accuracy (but in our case of simple particles and no background, primarily linking accuracy). To compute this statistic, we search for ground truth trajectories that are on average less than two pixels away from the predicted trajectory. From the set of matching trajectories, such as those pictured in the figure, we can calculate the ED. (**d**) Diagram of two ground truth trajectories (shown in dark blue) with potential PT-based trajectories (shown in light blue, yellow and green, with each color signifying a different trajectory predicted by a PT method) on top to illustrate the classical errors (and concept of the turn angle $\theta(t)$) discussed in this paper.

Results

Trajectory patterns illuminate PT method variations

To understand PT performance at a visual level, we plotted a small window of trajectories for the bimodal and unimodal simulations for $PSDR \leq 1.7$ (Fig. 4). Some selected trajectories for $PSDR \geq 3.0$ are also given in Supplementary Fig 1. For the bimodal simulation at $PSDR = 3$, trajectories from all PT codes reasonably mirror the simulations. However, this congruence quickly diminishes at $PSDR = 1.7$ (Fig. 4b), with marked deviations underscoring the nuances of each linking algorithm. It's noteworthy that for $PSDR \leq 1.7$ (Fig. 4a and b), trajectories close to the cylinders (i.e., slow trajectories) are detected more accurately compared to the faster trajectories in the pore throat. In addition to increased speed, particles in the pore throat exhibit spatial convergence, which results in a large local decrease in PSDR. This observation carries over to the unimodal simulations (Fig. 4c), which exhibit consistent patterns across PT codes. A pervasive trend emerges: PT codes tend to underestimate the likelihood of particles moving at high velocities for a variety of simulated speed distributions, especially when spatial convergence further reduces PSDR.

Probing deeper into individual PT code performances for the bimodal simulations, especially at lower simulated PSDRs, TM-Kalman stands out with superior accuracy, although it's not exempt from erroneous links at elevated speeds. In particular, TM-Kalman shows a significant amount of erroneous long links across pore spaces (streamlines don't cross the pore space in our bimodal simulations, so any link across the pore space is a false link). TP shows the greatest amount of false links and split trajectories (further explained in next section) at $PSDR \leq 1.7$ (Fig. 4a and b). TM-Lap similarly exhibits pronounced difficulties in linking high-speed particles, though not as significant as TP. At $PSDR = 1.7$, V-TrackMat trajectories generally resemble those of TM-LAP and TM-Kalman in terms of accuracy, although there are a smaller number of V-TrackMat trajectories. At $PSDR = 1.1$ (Fig. 4a), TM-Kalman and V-TrackMat appear to outperform TM-LAP. Although TM-Kalman and V-TrackMat may have more erroneous links across the pore space, TM-LAP has a much greater number of zig-zagging trajectories (links going back and forth between two or more different particles), and less true trajectories that last a significant distance. Thus, although V-TrackMat's linking algorithm is less aggressive than either TM algorithm, V-TrackMat captures a substantial portion of accurate trajectories.

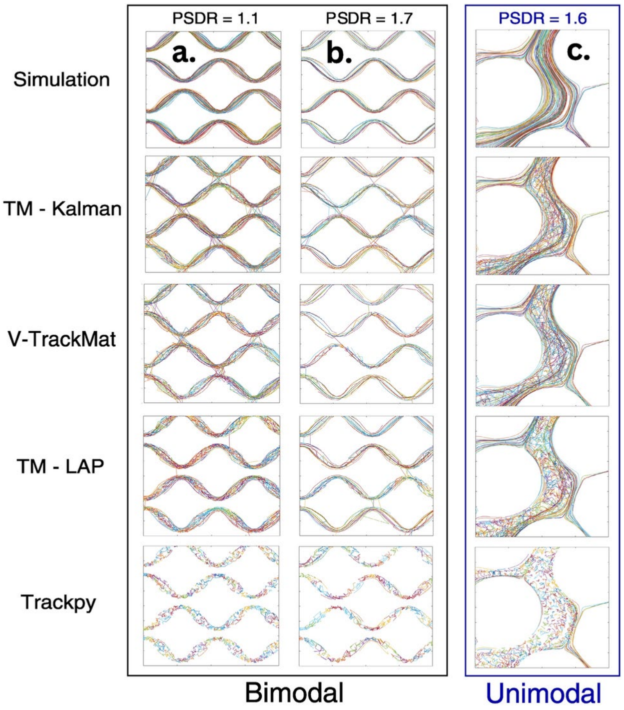

Fig. 4. Sample trajectories for all PT codes for the low-PSDR bimodal and unimodal simulations. Each plot shows a 400x400 pixel section of the whole domain. Within a plot, each line corresponds to a unique trajectory (with random colors used to show the contrast between individual trajectories). (**a**) Bimodal simulations for $PSDR = 1.1$. (**b**) Bimodal simulations for $PSDR = 1.7$. (**c**) Unimodal simulations for $PSDR = 1.6$. All algorithms suffer from trajectory splitting and erroneous linking for these low-PSDR simulations. TM-Kalman, TM-LAP and V-TrackMat clearly outperform TP in all scenarios. Although TM-Kalman and V-TrackMat have more clearly false links that stretch across the pore space (jump from one group of streamlines to another), TM-LAP has a much larger amount of zig-zagging trajectories caused by erroneous links between close particles. The low-PSDR unimodal simulations generally show the same trends as the bimodal simulations; however, the differences between TM-Kalman, TM-LAP, and V-TrackMat are less significan.

The unimodal simulations at $PSDR = 4$ (Supplementary Fig. 1) further show that all algorithms besides TP have robust performance regardless of geometry. At $PSDR = 1.6$ (Fig. 4c), all algorithms show problems with false links and split trajectories. Similar to the bimodal results, V-TrackMat and TM-Kalman seem to have a larger amount of accurate trajectories than TM-LAP. Thus, general algorithm performance is largely independent of the geometry in which the particles are tracked. However, it should be noted here that the range of possible particle speeds in our simulations, which is largely impacted by geometry and flow conditions, only spans 3-4 magnitudes (Fig. 2a and b). High fidelity simulations of Lagrangian particles in porous media show speed distributions that range up to 8 orders of magnitude[53], so we can't be confident that our findings (relative rankings of PT performance) would remain accurate for transport in any geometry or flow condition. Furthermore, to focus on linking, we didn't include any background. However, in real experiments that image bacteria in microfluidic devices, the geometry has a significant impact on tracking performance due to the presence of light scattering around grains[21].

To further understand differences in our PT codes, we plot both the simulated (ground truth) and PT-generated normalized speed distributions for our lowest PSDR bimodal (Fig. 5a) and unimodal (Fig. 5b) simulations. To quantify these differences, we calculate the 1-Wasserstein distance (W_1) between each ground-truth and tracked speed PDF (Table 2). Visual inspection of the PDFs, as well as the trends in (W_1), indicate that TM-Kalman is able to reproduce the simulated speed distributions the best, followed by V-TrackMat, then TM-LAP, then TP. Interestingly, each PT code besides TP overpredicts the fastest speeds for the bimodal simulation, but underpredicts the fastest speeds for the unimodal simulations. During tracking, an effort was made to use the highest possible linking distance that did not result in a significant number of mislinks. Because the range of speeds for the unimodal simulations is greater than that of the bimodal simulations, we were unable to capture the fastest speeds in the unimodal simulation without causing significant false links.

Relationship between classical statistics and PSDR

To develop a more large-scale understanding of the performance of each PT code, we use a variety of classical and experimental statistics (Fig. 3). Each of these classical statistics target different potential sources of linking error. Because the imagery had a high signal to noise ratio, there were not many errors in the detection stage of PT for each simulation (only occurring due to overlapping particles). Therefore, the false link rate (FLR)

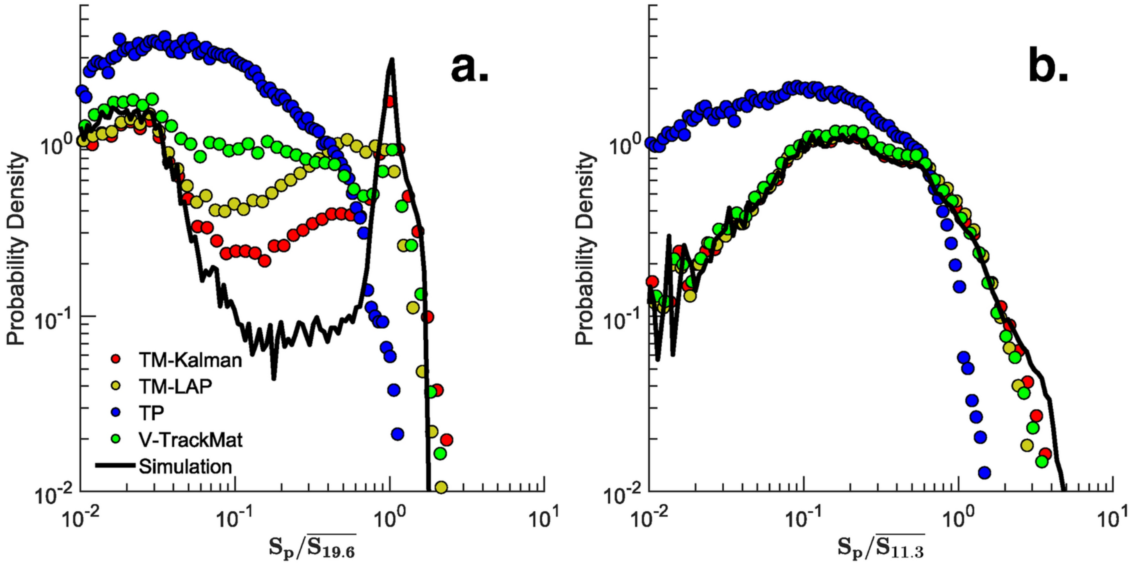

Fig. 5. This figure shows the a comparison between the speed distributions for each PT code for the lowest-PSDR bimodal (a) and lowest-PSDR unimodal (**b**) simulations. The speed distributions in (**a**) are normalized by the mean speed of the lowest-PSDR bimodal simulation (19.6 pixels/frame). The speed distributions in (**b**) are normalized by the mean speed of the lowest-PSDR unimodal simulation (11.3 pixels/frame). This figure shows the ability of each PT code to handle significantly different distributions of particle speeds.

	Bimodal	Unimodal
TM-Kalman	0.1146	0.1363
TM-LAP	0.3400	0.2489
TP	0.7536	0.6349
V-TM	0.3434	0.2158

Table 2. W_1 between ground truth speed distributions and the speed distributions from each PT method for the lowest PSDR bimodal and unimodal simulations.

primarily shows the potential for a particle to be unlinked, meaning there were no probable candidates for linking in nearby frames (Fig. 3a). The mean path length (MPL) shows the propensity for trajectories to be fractured due to lack of linking (Fig. 3b), and the Euclidean distance (ED) indicates the likelihood for links to move back and forth between particles, sampling a large number of particles for a single trajectory (Fig. 3c). A realistic diagram of each of these potential errors is shown in Fig. 3d.

Plotting these statistics over a range of PSDRs reveals that TM-Kalman and TM-LAP consistently eclipse the performance of other PT methods (Fig. 6). In particular, the bimodal simulations reveal several task-relevant patterns. The mean path lengths (Fig. 6a) illuminate the tendency for V-TrackMat and TP to generally have shorter trajectories compared to either TM method. This shortening in TP’s trajectories is significantly accentuated, especially at low PSDR levels. We attribute this phenomenon to ’trajectory splitting’, where a particle is tracked for only a fragment of its presence in the field of view. Intricacies of TP’s linking algorithm, which narrows the search space when inundated with potential particles for the ensuing frame, underpin this observation. While effective for slower-moving particles, especially in terms of memory requirements and algorithm speed, this linking strategy is less adept at tracking high-velocity particles in a directed flow. For V-TrackMat, the trajectory splitting seems to be a result of its more stringent linking algorithm. Although all PT codes try to match all trajectories during linking, V-TrackMat seems to have more extreme criteria that prevent incorrect links, as shown from the sample trajectories (Fig. 4). Thus, many trajectories are lost by V-TrackMat due to the algorithm’s necessity for high-probability links.

Furthermore, the FLRs (Fig. 6b) point towards V-TrackMat’s propensity to either miss or inaccurately record a particle in a frame. However, because this error is likely a result of careful linking, the classical statistics may exaggerate the experimental errors for tracking algorithms such as V-TrackMat’s. The EDs (Fig. 6c) further highlight that TP and V-TrackMat often record the most substantial discrepancies between the actual and tracked positions. This observation, particularly for V-TrackMat, implies that a rigorous linking algorithm doesn’t invariably lead to precise trajectory reconstructions. Although untested, it is theoretically plausible that during velocity-based linking or gluing, particles are incorrectly linked because they have similar velocities.

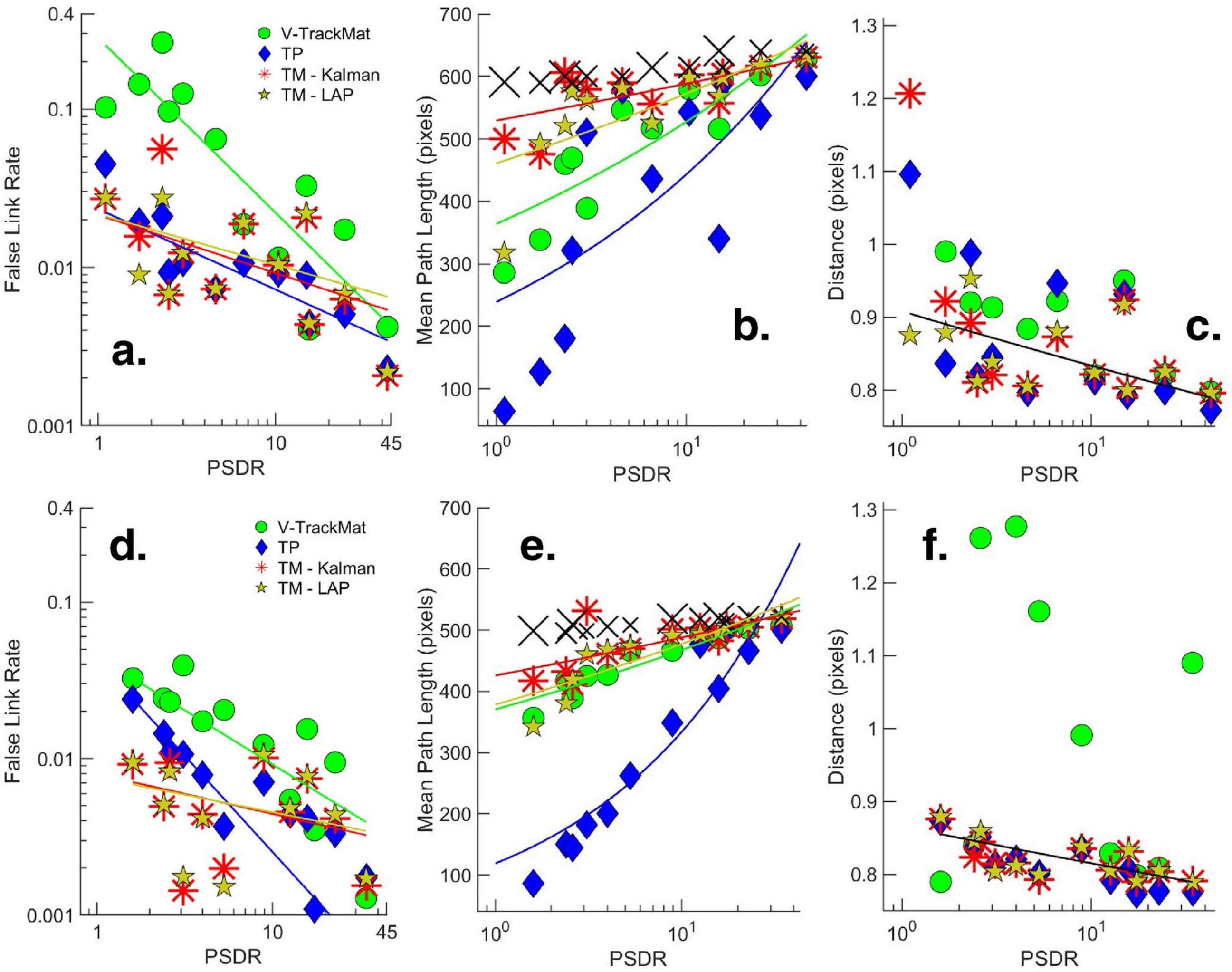

Fig. 6. This figure shows the results of the classical comparative statistics for both the bimodal and unimodal simulations. For all plots, the size of the scatter points represent the particle density of the simulation (larger points means greater particle density). a-c correspond to the bimodal simulations, and d-f correspond to the unimodal simulations. (**a**) Mean false link rate (error due to detection and temporally local missed links). (**b**) Mean path length of all PT-obtained and simulated trajectories. The ground truth is shown as a black X. This statistic describes how often full trajectories are split (linking error over time). (**c**) Mean Euclidean distance between true and predicted trajectories (error due to localization and linking error). (**d–f**) Repeat of a-c but for the unimodal simulations. These figures generally indicate that V-TrackMat and TP have the worst "classical" performance. Furthermore, classical statistics tend to follow a power law trend as a function of PSDR. Power law fit equations and goodness of fit are given in Table 3.

For unimodal simulations, classical statistics (Fig. 6d–f) generally perform better than their bimodal counterparts. The bimodal simulations have higher mean speeds than the unimodal solutions (Table 1). Furthermore, the bimodal simulations (Fig. 1a) have a larger number of particles at high speeds, which causes more difficulty in particle tracking. In addition, the unimodal simulations show a greater range of speeds and are generally more reminiscent of speed distributions of particles in porous media[54]. Thus, the unimodal simulations likely offer a more comprehensive representation of generic PT code efficacy in porous media. While the general trends mirror those in the bimodal findings, V-TrackMat performs comparatively better in the FLR metric (Fig. 6d) and worse in the ED metric (Fig. 6f), and TP performs better in the ED metric (Fig. 6f). TP's aforementioned challenges with fast-moving particles mean its performance slightly improves in unimodal settings, which aren't dominated by high speed trajectories. Still, TP's mean path lengths (Fig. 6e) depict a sharp decline as PSDR decreases, implying the persistent issue of trajectory splitting in both bimodal and unimodal settings.

The results of the classical statistics imply that TM-Kalman and TM-LAP outperform V-TrackMat in all cases, but from the sample trajectories (Fig. 4), we have shown this to not be true. Also, the trajectories show TP performs much worse than the other algorithms at low PSDR, but this is not reflected by the FLR and ED metrics. We posit that the primary reason for the disconnect between the classical statistics and the sample trajectories is that the FLR and ED metrics underpenalize aggressive linking. The FLR will always be lower when more links

Bimodal	*a*	*b*	R^2	*RMSE*
ED	0.922	-0.039	0.941	0.0210
MPL (TP)	238.9	0.268	0.510	141.00
MPL (TM-Kalman)	529.3	0.046	0.434	36.890
MPL (TM-LAP)	510.0	0.056	0.797	39.500
MPL (V-TrackMat)	363.9	0.161	0.767	55.250
FLR (TP)	0.023	-0.508	0.884	0.0041
FLR (TM-Kalman)	0.019	-0.342	0.813	0.0067
FLR (TM-LAP)	0.022	-0.319	0.261	0.0080
FLR (V-TrackMat)	0.212	-0.783	0.875	0.0288
Unimodal	*a*	*b*	R^2	*RMSE*
ED	0.849	-0.019	0.923	0.0323
MPL (TP)	118.9	0.448	0.885	55.620
MPL (TM-Kalman)	425.9	0.059	0.503	29.770
MPL (TM-LAP)	432.0	0.052	0.940	14.160
MPL (V-TrackMat)	370.4	0.101	0.881	17.960
FLR (TP)	0.043	-1.226	0.901	0.0022
FLR (TM-Kalman)	0.008	-0.255	0.142	0.0032
FLR (TM-LAP)	0.007	-0.223	0.110	0.0033
FLR (V-TrackMat)	0.045	-0.690	0.931	0.0032

Table 3. Power law fit equations and goodness of fit for ED, MPL and FLR.

are forced, since the probability of false positive detection is very low. The ED metric will always be higher when more links occur between different trajectories, but if the trajectories are nearest neighbors, then the error will be relatively small. Thus, long trajectories and links across the pore space (such as those of V-TrackMat and TM-Kalman) will result in more ED error than zig-zagging trajectories between close neighbors (such as TM-LAP) will. Ultimately, the FLR and ED underpredict PT error for nearest-neighbor based algorithms with little constraint for linking. As a result, these statistics fail to grasp the nuanced differences between PT codes.

Beyond comparing PT codes' performances, we also demonstrate that all classical statistics have a power law relationship with PSDR, although some relationships are more significant than others (Table 3). As PSDR is reduced, all PT codes generally exhibit increased ED and FDR, and decreased MPL. V-TrackMat and TP show a steeper relationship between FDR and PSDR than either TM algorithm, which generally indicates that the TM algorithms are more robust with respect to FLR performance over a range of PSDRs (Fig. 6a and d). V-TrackMat and TP also show steeper relationships between MPL and PSDR, further demonstrating the resilience of the TM algorithms when considering classical linking failures. V-TrackMat and TP also generally show more significant (lower RMSE) power-law relationships than the TM algorithms, indicating that classical PT error for V-TrackMat and TP is more predictable. Furthermore, classical statistics from unimodal simulations (Fig. 6d–f) present slightly different power law relationships compared to those from bimodal simulations (Fig. 6a–c). Thus, the choice of PT algorithm, and variations in ground truth particle speed distributions, can influence the specifics of these power law relationships.

Experimental statistics highlight task-specific PT performance

The classical statistics from bimodal simulations (Fig. 6) echo many patterns observed in the sample trajectories (Fig. 4). However, there are notable deviations. The sample trajectories, for instance, present V-TrackMat as clearly superior to TP and comparable or superior to TM-LAP. To discern which mode of analysis â€” comparative statistics or visual trajectory inspection â€” offers a more accurate picture of PT performance, we used a variety of experimental statistics.

In the bimodal simulations, the normalized speed-angle joint probability density difference heatmaps rank TM-Kalman as the top performer, with V-TrackMat and TM-LAP occupying intermediate positions and TP trailing (Fig. 7). All codes demonstrate strong tracking performance at $PSDR \geq 2.5$, but V-TrackMat and TP's limitations become evident at $PSDR \leq 2.3$. TM-LAP and TM-Kalman significantly outperform V-TrackMat for $PSDR \geq 1.7$. However, at $PSDR = 1.1$, V-TrackMat performs better than TM-LAP, as shown by the large amount of overprediction for the probability of low speed and high turn angle particles (Fig. 7). This disparity is likely rooted in the LAP algorithm's propensity for aggressive linking that doesn't take particle velocities into account, in contrast to V-TrackMat's more conservative velocity-based approach. Consequently, at $PSDR = 1.1$, while LAP is prone to errant predictions for high speed particles and forces links with large turn angles, V-TrackMat is more likely to keep particles unlinked, and only significantly overpredicts low turn angles. In other words, V-TrackMat often refrains from making connections altogether, and when V-TrackMat does have false links, its reliance on expected particle velocities, akin to TM-Kalman, ensures that the errors are relatively benign (with respect to velocity and angle distributions) compared to TM-LAP.

In the context of unimodal simulations (Supplementary Fig. 2), both the V-TrackMat and TM algorithms predict speed and angle statistics with near perfection. V-TrackMat and TM-Kalman perform slightly better

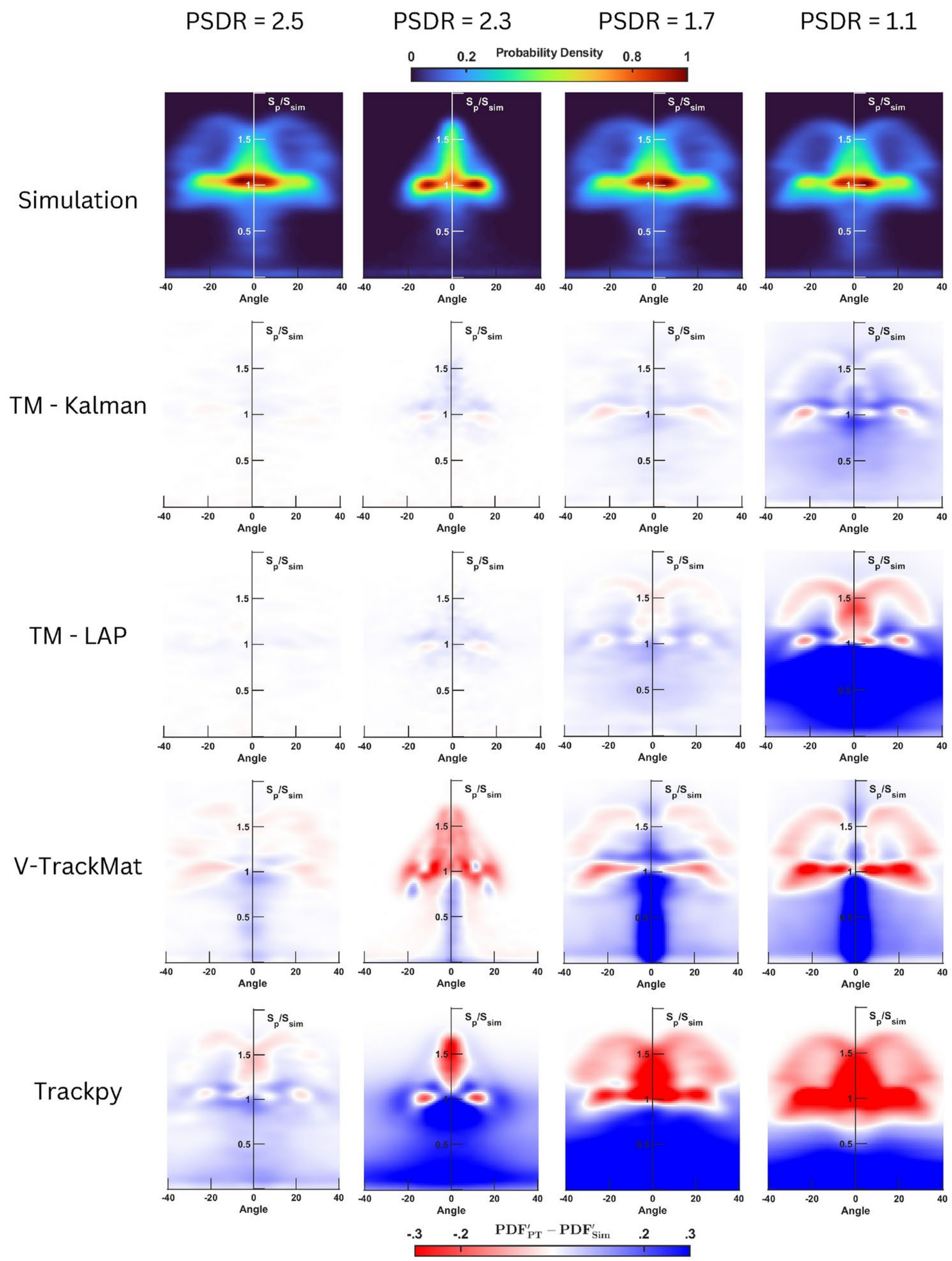

Fig. 7. Speed-angle joint probability density difference heatmaps for the bimodal simulation. Speeds determined from particle tracking (S_p) are normalized by the mean speed of the respective simulation (S_{sim}). Red corresponds to an underprediction of probability density, blue corresponds to an overprediction of probability density, and white corresponds to an accurate probability density prediction within the speed-angle feature space. These results generally show the same trends as the sample trajectories (Fig. 1). At $PSDR = 2.5$, all algorithms show strong performance as indicated by the lack of strong color. All PT methods besides Trackpy and V-TrackMat show good replication of the simulation for $PSDR \geq 1.7$. At $PSDR = 1.1$, TM-Kalman still performs best and TP performs worst, but V-TrackMat surprisingly outperforms TM-LAP. Thus, at very low PSDR, velocity-based algorithms result in significant improvements to PT performance.

than TM-LAP, which can be seen from the slightly greater underprediction of high speed and low turn angle particles for TM-LAP at $PSDR \leq 2.6$. TP shows relatively poor performance for all $PSDR \leq 3.1$. These observations further reinforce the general trends seen in the sample trajectories (Fig. 4). They confirm the case presented by the classical statistics that TM-Kalman has superior performance, but they significantly contrast the relative classical results of V-TrackMat and TM-LAP. Specifically, the speed-angle distributions (both bimodal and unimodal) show that TM-LAP may be favorable for $PSDR \geq 1.7$, but that V-TrackMat is superior for $PSDR \leq 1.6$.

Velocity autocorrelation function (C_v) and mean squared displacement (MSD) analysis (Fig. 8) further corroborates the trends evident in the speed-angle heatmaps. It should be noted here that we only present the first 20 frames of the lowest and highest-PSDR simulations in the main text of this paper, although the full C_v and MSDs for all simulations can be observed in Supplementary Figs. 3 and 3. Because our simulations don't use reinjection to keep the number of particles in the field of view relatively constant, the C_v and MSDs for our simulated particles are unrealistic past 20-30 frames. Since the focus of our analysis is on the relatively accurate simulation of dispersing particles in porous media, we chose to focus on the subset of our results that are the most realistic.

At $PSDR = 34.3 - 42.8$, all PT methods align closely with the simulated autocorrelations and MSD ratios. There is some slight deviation for the MSD ratio at late times for the bimodal simulation for TP and V-TrackMat (Fig. 8c), but generally, all results are highly accurate. However, at $PSDR = 1.1 - 1.6$, all PT methods show large deviations in autocorrelation and MSD ratio. The autocorrelation for the low PSDR bimodal simulation (Fig. 8b) shows decent performance for TM-Kalman, but poor performance for all other PT methods. The repetitive motion of the C_v is indicative of the wave-like periodic movement of the particles dispersing through the lattice-like geometry of the bimodal simulations. TM-Kalman is slightly able to capture this feature of the autocorrelation, but the other PT codes are not. The most likely explanation for this lies in the false links and splitting of fast trajectories. As previously discussed, as particles travel through the pore throat, they get closer together and speed up, which causes a decrease in the local PSDR. Thus, the C_v reveals that TM-Kalman is more likely to capture these fast/dense particles in the pore throats than the other PT codes are. The unimodal

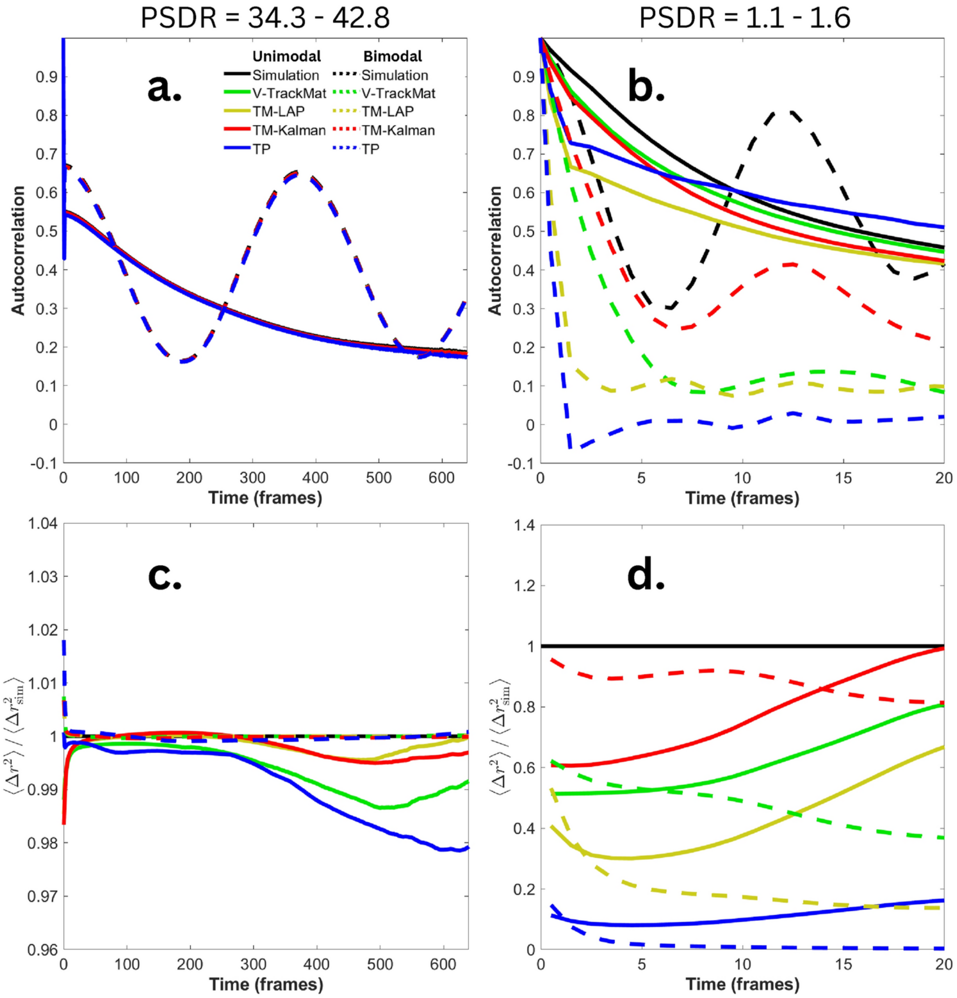

Fig. 8. MSD ratios and VACFs for unimodal and bimodal simulations for high (**a** and **c**) and low (**b** and **d**) PSDRs. The MSD ratio is caluculated as the MSD obtained from particle tracking divided by the simulated MSD. An MSD ratio of 1 implies perfect accuracy. The bimodal MSD ratios are shown by dashed lines, and the unimodal MSD ratios are shown by solid lines. The simulation, or ground truth, is black, and the results from each PT method are different colors. For the unimodal simulations, $PSDR = 34.3$ (a and c) or $PSDR = 1.6$ (**b** and **d**). For the bimodal simulations, $PSDR = 42.8$ (**a** and **c**) or $PSDR = 1.1$ (b and d). These figures generally confirm trends present in the other experimental results. Furthermore, the MSDs and VACFs generally show the same trends, implying that a good prediction of MSD allows for a good prediction of C_v. However, unlike the other experimental statistics, the C_v is not a reliable proxy for general PT performance.

results for the C_v at low PSDR (Fig. 8b) surprisingly show that V-TrackMat outperforms TM-Kalman, and TP outperforms TM-LAP. However, the full C_v (Supplementary Fig. 4) indicates the TM-LAP outperforms TP at $t \geq 30$. Likely, the C_v for TP is relatively accurate at early times because TP can only track very slow particles, so there are no significant false links that would cause velocity decorrelation between successive timesteps. TM-LAP, on the other hand, can track much faster particles, but may also erroneously link these fast particles, meaning a greater amount of velocity decorrelation. Thus, although it is important to know how accurate the C_v is for general analysis of particle transport, the C_v accuracy can't be used as a general proxy for total particle tracking accuracy.

The MSD ratios for low PSDR (Fig. 8d) show significant deviations from the simulated MSD for each PT code. Both the bimodal and unimodal results show TM-Kalman is able to most closely follow the true MSD (i.e., have an MSD ratio of 1), then V-TrackMat, then TM-LAP, and finally TP shows a complete disconnection from the true MSD. Interestingly, the unimodal simulations show an improvement in the MSD ratio over time, which indicates that for each PT code, the history of previous particle positions and links can improve the accuracy of tracking. For the bimodal simulations, we see a decrease in the accuracy of the MSD ratio over time (Fig. 8d). However, the full time-series for the lowest PSDR bimodal simulation (Supplementary Fig. 3) shows a significant improvement in the MSD accuracy over time for both V-TrackMat and TM-Kalman. Thus, velocity-based algorithms show a clear advantage in late time prediction of MSDs for low-PSDR scenarios, regardless of geometry.

Generally, our experimental statistics reveal that while rudimentary comparative statistics can offer broad insights into PT code competencies across various tracking scenarios, they might fall short in pinpointing optimal codes for specific particle motions with particular analytical objectives. In our bacterial dispersion simulation within porous media, these statistics fail to elucidate speed, angle, autocorrelation or displacement distribution accuracies - all crucial for comprehending bacterial transport. Furthermore, these comparative statistics tend to underpenalize aggressive linking. Thus, basic comparative statistics might not capture the full spectrum of PT code capabilities. A more complete analysis, which can be done through a variety of statistical and visual methods, is indispensable for discerning the optimal PT code tailored to specific conditions.

PT performance for simulations with noisy trajectories

While the primary analysis in this paper revolves around simulations where trajectories only vary in speed and particle density, we have also provided an analysis of PT performance for simulations that contain more noise depicting experimental errors in video capture and processing. Specifically, we analyzed PT performance for simulations in which the particles had enhanced random displacement (Gaussian distribution with $\mu = 0$ and $\sigma = 2$ pixels) added to the purely advective tracks, and 2% of the particles were dropped in any given frame to account for particle intermittency. The random displacement is a simple representation of a variety of experimental phenomena/positioning errors such as diffusion, camera jitter, and/or oscillations in particle brightness. The intermittency represents particles moving in and out of the focal plane, which can also be impacted by diffusion, particle-particle interactions, particle-wall interactions, and camera exposure time. Both of these changes can be generalized as increasing the noise of the trajectories in the simulations. The sample trajectories of the lowest PSDR simulations with the intermittent and random-motion particles (Fig. 9) show the same general trends as those of the simulations with minimal noise (Fig. 4), but for each PT code the errors are slightly higher in the case of the noisy trajectories. The speed-angle distributions for the noisy unimodal simulations (Fig. 10) show that TP clearly has the worst performance for $PSDR \leq 2.3$. For the highest PSDR noisy unimodal simulation, the performance of all PT codes are comparable. At the lowest PSDR, TM-Kalman once again shows the best performance. Similar to the unimodal results with minimal trajectory noise (Supplementary Fig. 2), V-TrackMat performs better than TM-LAP in all cases besides the highest-PSDR simulation.

Ultimately, these results indicate that trajectory noise such as large random motions and particle intermittency make the tracking process more error-prone, although the rankings of the linking algorithms are not impacted by these potential experimental issues. However, a more robust analysis of potential experimental errors would deal with a number of other factors such as signal to noise ratio and particle shape/size. This would also require a rigorous comparison of detection methods, which was beyond the scope of our work, but we recommend that future researchers compare PT codes in the context of more diverse simulations.

Consequences of particle tracking errors on transport analysis

Building on our comparative exploration of PT codes, the findings from the bivariate speed-angle heatmaps, MSDs, and C_v (Figs. 7 and 8, and Supplementary Figs. 2, 3, and 4) shed light on the dispersion dynamics of tracer particles within porous media. Specifically, they underscore how inaccuracies introduced by PT errors can skew transport analysis. A predominant manifestation of PT error arises from false links (Fig. 4d), leading to a systematic underestimation of high-speed particles (Figs. 4, 5 and 7). This, in turn, results in a conservative estimation of particle speeds (Figs. 5 and 7) and MSDs (Fig. 8). TP, which shows the most significant error due to trajectory splitting, underestimates the particle speeds and MSDs to an extreme degree for low PSDR.

Other consequences of PT error, which can primarily be observed in the TP results, are inflated turn angles (Fig. 7) and diminished or enhanced C_v (Fig. 8a and b), attributable mainly to trajectory splitting and erroneous linking. Furthermore, we find that in the case of the bimodal geometry, where there is a periodic nature to the velocity of particles over time, only TM-Kalman is able to slightly capture the periodicity of this autocorrelation.

Our analysis also emphasizes the paramount importance of experimental conditions (related to particle speed and density) in achieving reliable PT outcomes. In the case of minimal-noise simulations, a PSDR exceeding 3 ensures nearly flawless tracking, regardless of PT algorithm. Conversely, a PSDR near or below 1 presents challenges for all PT codes. In scenarios characterized by low PSDR coupled with directed particle movements, algorithms that harness velocity-based linking emerge as the more prudent choice.

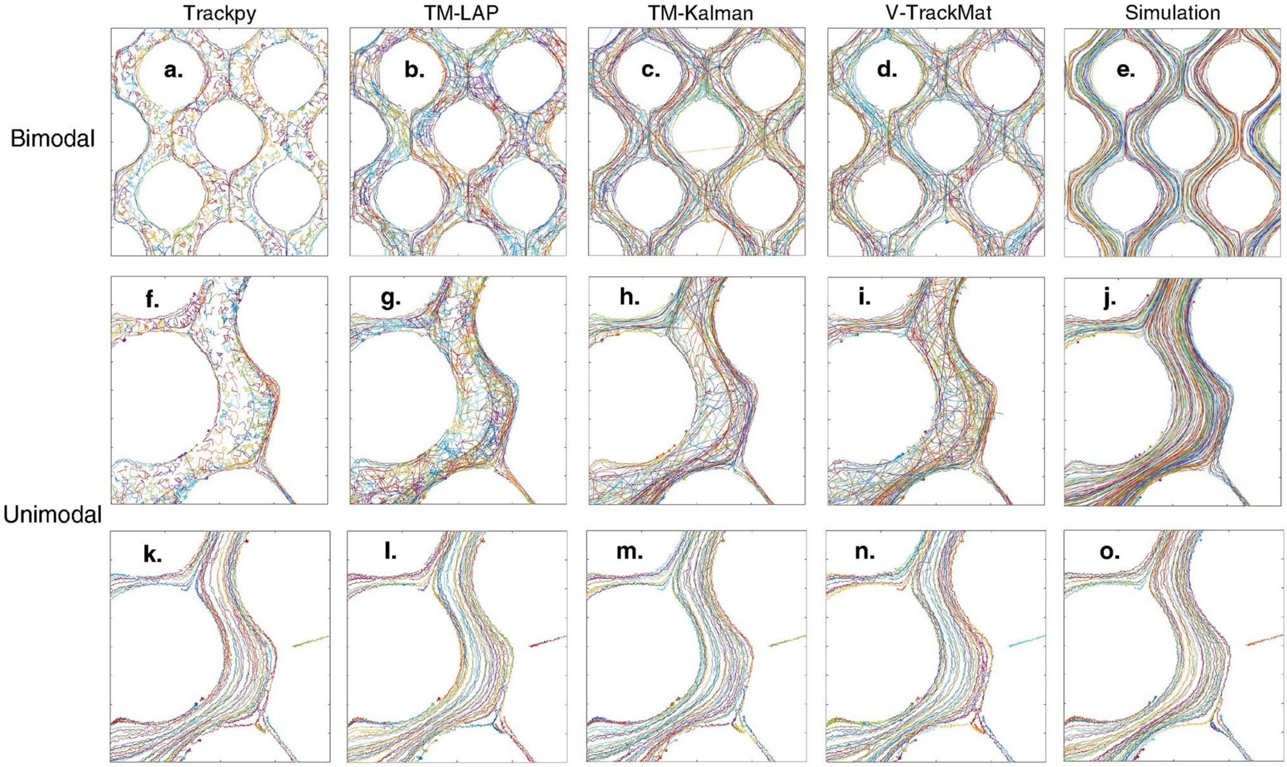

Fig. 9. Sample trajectories for all PT codes for the low-PSDR bimodal and unimodal simulations with random motion and 2% particle intermittency. Each plot shows a 400x400 pixel section of the whole domain. Within a plot, each line corresponds to a unique trajectory (with random colors used to show the contrast between individual trajectories). (**a**–**e**) Bimodal simulations for $PSDR = 1.5$. (**f**–**j**) Unimodal simulations for $PSDR = 1.5$. (**k**–**o**) Unimodal simulations for $PSDR = 8.1$. Compared with Fig. 4, this figure (specifically the top left of **f**–**j**) shows a slight decrease in tracking performance for similar PSDR due to the addition of trajectory noise.

In addition, the results for noisy simulations are worse than for simulations with minimal noise (Figs. 4 and 9), which highlights the need for tight experimental controls to improve the visual quality of the particles. Although some amount of noise is unavoidable, these results show the importance of trying to ensure that all particles remain in the focal plane of the acquisition device.

PT algorithm speed comparison

In addition to performance analysis, we also report how long each PT code takes to link trajectories (Fig. 11). Generally, TM - LAP is the fastest linking algorithm, then TP, then TM - Kalman, and V-TrackMat is the slowest. Thus, we observe a significant trade-off between performance and computation time - the best PT methods at low PSDR also take the longest. However, we must also note that each PT code is developed in a different programming language (Python, Matalb and Javascript), so we are unable to fairly assess the speed of the underlying algorithms.

Discussion

Our comprehensive analysis of PT methods underscores TM-Kalman as the leading PT algorithm in terms of accuracy and robustness. While V-TrackMat emerges as a strong contender at low PSDR, TM-LAP stands out at high PSDR. TP, although impressive at high PSDR, falters notably with disconnection challenges at $PSDR \leq 1.7$. Despite the TP authors suggesting that adjusting the “SubnetOversizeException” variable could rectify the constrained search space at low PSDR, our attempts in this direction were unsuccessful. However, it should be noted that we used TP version 0.5.0, and that a newer version may have more potential to rectify this error. Barring this error, TP would, in all probability, align more closely with the performance of other methods at low PSDR.

It’s evident across the board that PT methods grapple with high FLRs, high ED, and trajectory splitting/fragmentation, especially at $PSDR \leq 1.7$. However, we show that poor performance in classical statistics doesn’t necessarily imply poor performance in experimental statistics. Specifically, classical statistics underpenalize aggressive linking algorithms, and overpenalize careful linking agorithms. In addition, we show that TM-Kalman, and V-TrackMat, which both use particle velocities to enhance predictions, exhibit marked improvements at $PSDR = 1.1$ relative to TM-LAP. Although TP also uses particle velocity information to make

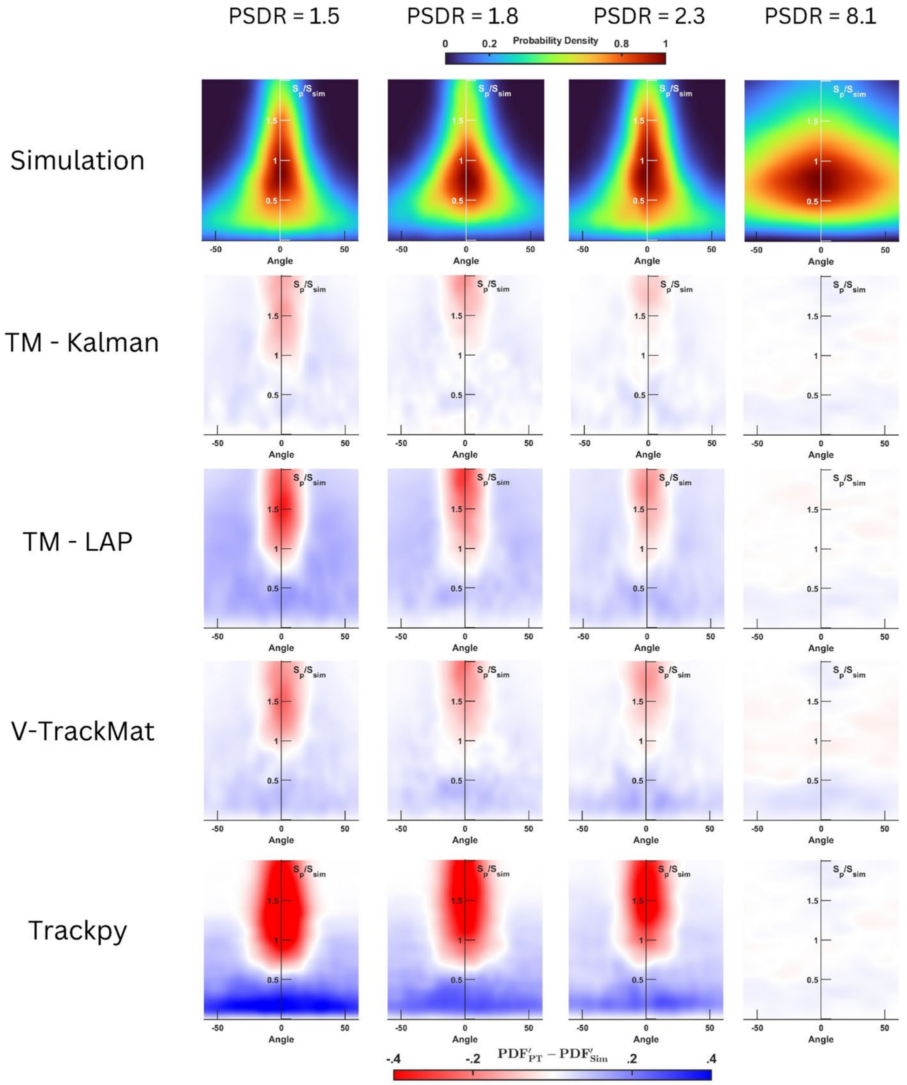

Fig. 10. Speed-angle joint probability density difference heatmaps for the unimodal simulations with random motion and 2% particle intermittency. Speeds determined from particle tracking (S_p) are normalized by the mean speed of the respective simulation (S_{sim}). Red corresponds to an underprediction of probability density, blue corresponds to an overprediction of probability density, and white corresponds to an accurate probability density prediction within the speed-angle feature space. Although these noisy simulations are slightly harder to track, the general trends in PT performance remain the same.

linking predictions, the SubnetOversizeException issue was much more significant than any potential gain due to velocity-based predictions. Thus, the leveraging of velocity data, especially for particles dispersing in porous media which exhibit a wide range of potential speeds, emerges as a critical factor in bolstering PT predictions.

Beyond algorithmic evaluations, this work highlights some of the common errors in transport analysis that emerge as a result of PT error. We find that all PT codes underestimate particle speeds and overestimate turn angles, and that poor tracking causes a significant loss in the accuracy of reproducing cyclical autocorrelations. Furthermore, we advocate for the recording of video data at a minimum threshold of $PSDR \geq 3$, a benchmark that promotes reliable tracking irrespective of the algorithm employed. Finally, we show that experimental noise can reduce the quality of the predicted trajectories from each PT code, but it doesn't significantly impact the relative rankings of each PT code.

While we do provide robust statistical comparisons between some of the best-known open source PT codes, we recognize that our study is somewhat limited in scope. A more robust analysis would use a wider variety of PT codes (including DL-based ones), more task-relevant statistics to test performance across a variety of domains, more variety of particle motions and noise, more variety in imagery type, and would enlist the creators of each code to submit the trajectories to be scored. Such an investigation was far beyond the scope of our paper, and would require a large collaborative effort spanning multiple disciplines.

Looking forward, we envision our research catalyzing advancements in PT theory in three pivotal aspects. First, we urge future studies to transcend the boundaries of classical statistics in PT code comparisons,

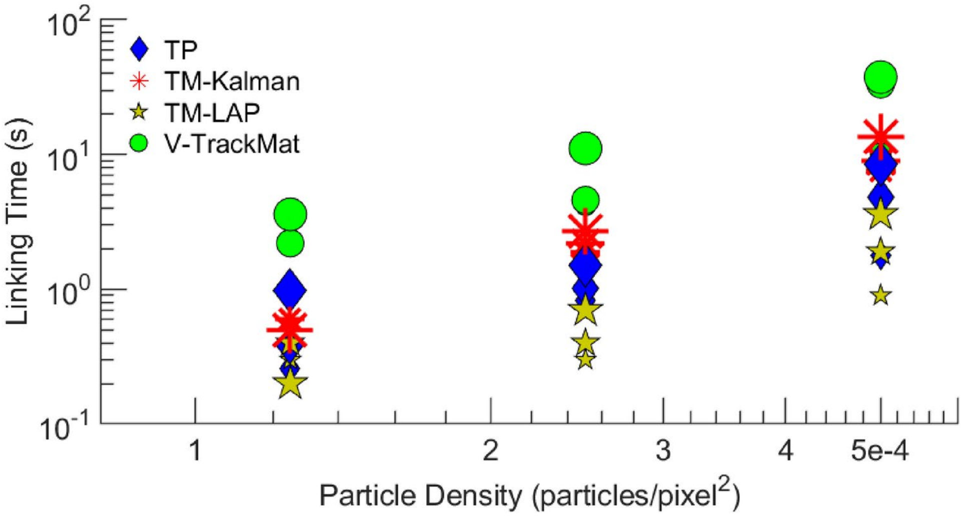

Fig. 11. Amount of time each PT code takes during the linking stage for selected unimodal simulations. Simulation speed is represented by scatter point size (large = 9.9 px/frame, medium = 2.6 px/frame, small = 0.9 px/frame). V-TrackMat consistently has the longest linking times, and TM-LAP has the shortest linking times. All algorithms show a power law relationship between linking time and particle density. High speed simulations generally take longer to link than low speed ones, and this difference increases at higher particle density.

emphasizing the integration of experimental outcomes pertinent to specific applications. Second, our findings echo the effectiveness of velocity-centric PT methods for dispersing particles in porous media, extending their proven efficacy from constant velocity scenarios[26] to contexts characterized by large velocity fluctuations in time and space. Third, we highlight the importance of maintaining a reasonably high PSDR to achieve precise particle transport analysis.

Data availability

Most of the data used in this study can be found at https://doi.org/10.5281/zenodo.10891931. Please send any data requests to the corresponding author (Rishi Parashar).

Received: 2 April 2024; Accepted: 7 October 2024
Published online: 15 October 2024

References

1. Saxton, M. Single-particle tracking: The distribution of diffusion coefficients. *Biophys. J* . **72**(4), 1744–1753 (1997).
2. Hong, Q., Sheetz, M. & Elson, E. Single particle tracking. analysis of diffusion and flow in two-dimensional systems. *Biophys. J.* **60**(4), 910–921 (1991).
3. Daumas, F. et al. Confined diffusion without fences of a g-protein-coupled receptor as revealed by single particle tracking. *Biophys. J.* **84**(1), 356–366 (2003).
4. Forier, K. et al. Transport of nanoparticles in cystic fibrosis sputum and bacterial biofilms by single-particle tracking microscopy. *Nanomedicine* **8**(6), 935–949 (2013).
5. Morales, V. L., Dentz, M., Willmann, M. & Holzner, M. Stochastic dynamics of intermittent pore-scale particle motion in three-dimensional porous media: Experiments and theory. *Geophys. Res. Lett.* **44**, 9361–9371 (2017).
6. Bultreys, T., De Boever, W. & Cnudde, V. Imaging and image-based fluid transport modeling at the pore scale in geological materials: A practical introduction to the current state-of-the-art. *Earth Sci. Rev.* **155**, 93–128 (2016).
7. Scheidweiler, D., Miele, F., Peter, H., Battin, T. J. & de Anna, P. Trait-specific dispersal of bacteria in heterogeneous porous environments: From pore to porous medium scale. *J. R. Soc. Interface* **17**, 20200046. https://doi.org/10.1098/rsif.2020.0046 (2020).
8. Dehkharghani, A., Waisbord, N., Dunkel, J. & Guasto, J. Bacterial scattering in microfluidic crystal flows reveals giant active taylor-aris dispersion. *Proc. Natl. Acad. Sci.* **116**(23), 11119–11124 (2019).
9. Bhattacharjee, T. & Datta, S. Bacterial hopping and trapping in porous media. *Nat. Commun.* **10**(1), 2075 (2019).
10. Dentz, M., Creppy, A., Douarche, C., Clément, E. & Auradou, H. Dispersion of motile bacteria in a porous medium. *J. Fluid Mech.* **946**, A33 (2022).
11. Creppy, A., Clément, E., Douarche, C., D'angelo, M. & Auradou, H. Effect of motility on the transport of bacteria populations through a porous medium. *Phys. Rev. Fluids* **4**(1), 013102 (2019).
12. Vallotton, P. et al. Diatrack particle tracking software: Review of applications and performance evaluation. *Traffic* **18**(12), 840–852 (2017).
13. Germain, D., Leocmach, M. & Gibaud, T. Differential dynamic microscopy to characterize Brownian motion and bacteria motility. *Am. J. Phys.* **84**(3), 202–210 (2016).
14. Ewers, H. & Schelhaas, M. Effect of motility on the transport of bacteria populations through a porous medium. *Methods Enzymol.* **506**, 63–80 (2012).
15. Dehkharghani, A., Waisbord, N. & Guasto, J. S. Self-transport of swimming bacteria is impaired by porous microstructure. *Commun. Phys.* **6**, 18. https://doi.org/10.1038/s42005-023-01136-w (2023).
16. Rusconi, R., Guasto, J. S. & Stocker, R. Bacterial transport suppressed by fluid shear. *Nat. Phys.* **10**, 212–217. https://doi.org/10.1038/nphys2883 (2014).

17. Bhattacharjee, T. & Datta, S. S. Bacterial hopping and trapping in porous media. *Nat. Commun.* **10**, 2075. https://doi.org/10.1038/s41467-019-10115-1 (2019).
18. Jeon, H. et al. Quantitative analysis of single bacterial chemotaxis using a linear concentration gradient microchannel. *Biomed. Microdev.* **11**, 1135–1143 (2009).
19. Birjiniuk, A. et al. Single particle tracking reveals spatial and dynamic organization of the Escherichia coli biofilm matrix. *New J. Phys.* **16**(8), 085014 (2014).
20. Secchi, E. et al. The effect of flow on swimming bacteria controls the initial colonization of curved surfaces. *Nat. Commun.* **11**, 2851. https://doi.org/10.1038/s41467-020-16620-y (2020).
21. Wu, H. & Schwartz, D. Nanoparticle tracking to probe transport in porous media. *Acc. Chem. Res.* **53**(10), 2130–2139 (2020).
22. Linkhorst, J., Beckmann, T., Go, D., Kuehne, A. J. & Wessling, M. Microfluidic colloid filtration. *Sci. Rep.* **6**(1), 22376 (2016).
23. Bevan, M. & Prieve, D. Hindered diffusion of colloidal particles very near to a wall: Revisited. *J. Chem. Phys.* **113**(3), 1228–1236 (2000).
24. Sholl, D., Fenwick, M., Atman, E. & Prieve, D. Brownian dynamics simulation of the motion of a rigid sphere in a viscous fluid very near a wall. *J. Chem. Phys.* **113**(20), 9268–9278 (2000).
25. Cheng, H., Hsu, C., Hung, C. & Lin, C. A review for cell and particle tracking on microscopy images using algorithms and deep learning technologies. *Biomed. J.* **45**(3), 465–471 (2022).
26. Chenouard, N. et al. Objective comparison of particle tracking methods. *Nat. Methods* **11**(3), 281–289 (2014).
27. Residori, M., Praetorius, S., de Anna, P. & Voigt, A. Influence of finite-size particles on fluid velocity and transport through porous media. *Phys. Rev. Fluids* **8**, 7 (2023).
28. Meijering, E., Dzyubachyk, O. & Smal, I. Methods for cell and particle tracking. *Methods Enzymol.* **504**, 183–200 (2012).
29. Clarke, D. & Martin-Fernandez, M. A brief history of single-particle tracking of the epidermal growth factor receptor. *Methods Protocols* **2**(1), 12 (2019).
30. Wiggins, C., Santos, R. & Ruggles, A. A feature point identification method for positron emission particle tracking with multiple tracers. *Nucl. Instrum. Methods Phys. Res. Sect. A* **843**, 22–28 (2017).
31. Travers, T., Colin, V., Loumaigne, M., Barillé, R. & Gindre, D. Single-particle tracking with scanning non-linear microscopy. *Nanomaterials* **10**(8), 1519 (2020).
32. Maska, M. & Matula, P. Particle tracking accuracy measurement based on comparison of linear oriented forests. In: *Proc. IEEE International Conference on Computer Vision Workshops* 11–17 (2017).
33. Malik, N., Dracos, T. & Papantoniou, D. Particle tracking velocimetry in three-dimensional flows: Part II: Particle tracking. *Exp. Fluids* **15**, 279–294 (1993).
34. Baek, S. & Lee, S. A new two-frame particle tracking algorithm using match probability. *Exp. Fluids* **22**, 23–32 (1996).
35. Shuang, B., Chen, J., Kisley, L. & Landes, C. Troika of single particle tracking programing: Snr enhancement, particle identification, and mapping. *Phys. Chem. Chem. Phys.* **16**(2), 624–634 (2014).
36. Bijeljic, B., Mostaghimi, P. & Blunt, M. J. Signature of Non-Fickian Solute Transport in Complex Heterogeneous Porous Media. *Phys. Rev. Lett.* **107**, 204502. https://doi.org/10.1103/PhysRevLett.107.204502 (2011).
37. De Anna, P., Quaife, B., Biros, G. & Juanes, R. Prediction of the low-velocity distribution from the pore structure in simple porous media. *Phys. Rev. Fluids* **2**, 124103. https://doi.org/10.1103/PhysRevFluids.2.124103 (2017).
38. Weller, H., Tabor, G., Jasak, H. & Fureby, C. A tensorial approach to computational continuum mechanics using object-oriented techniques. *Comput. Phys.* **12**, 620 (1998).
39. Tinevez, J. et al. Trackmate: An open and extensible platform for single-particle tracking. *Methods* **115**, 80–90 (2017).
40. Ershov, D. et al. Trackmate 7: Integrating state-of-the-art segmentation algorithms into tracking pipelines. *Nat. Methods* **19**(7), 829–832 (2022).
41. Allan, D., Caswell, T., Keim, N., van der Wel, C. & Verweij, R. soft-matter/trackpy: Trackpy v0. 5.0., https://doi.org/10.5281/zenodo.4682814 (2021).
42. Midtvedt, B. *et al.* Quantitative digital microscopy with deep learning. *Appl. Phys. Rev.* **8**(1) (2021).
43. Fatemi, B., Halcrow, J. & Jaqaman, K. Geometric deep learning of particle motion by magik. *Nat. Mach. Intell.* **5**(5), 483–484 (2023).
44. Zhang, Y., Wang, C., Wang, X., Zeng, W. & Liu, W. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *Int. J. Comput. Vision* **129**, 3069–3087 (2021).
45. AbrÃ moff, M., MagalhÃ£es, P. & Ram, S. Image processing with imagej. *Biophoton. Int.* **11**(7), 36–42 (2004).
46. Kalman, R. A new approach to linear filtering and prediction problems. *Basic Eng.* **82**, 35–45 (1960).
47. Jaqaman, K. et al. Robust single-particle tracking in live-cell time-lapse sequences. *Nat. Methods* **5**(8), 695–702 (2008).
48. Crocker, J. & Grier, D. Methods of digital video microscopy for colloidal studies. *J. Colloid Interface Sci.* **179**(1), 298–310 (1996).
49. Carrel, M. et al. Pore-scale hydrodynamics in a progressively bioclogged three-dimensional porous medium: 3-d particle tracking experiments and stochastic transport modeling. *Water Resour. Res.* **54**(3), 2183–2198 (2018).
50. Carrel, M. et al. Biofilms in 3d porous media: Delineating the influence of the pore network geometry, flow and mass transfer on biofilm development. *Water Res.* **134**, 280–291 (2018).
51. A. Alam, E. *Crowd of Microswimmers*. Phd thesis, Condensed Matter [cond-mat]: Université Grenoble Alpes, Université Grenoble Alpes. https://theses.hal.science/tel-03894880/document (2022).
52. Tarantino, N. et al. Tnf and il-1 exhibit distinct ubiquitin requirements for inducing nemo-ikk supramolecular structures. *J. Cell Biol.* **204**(2), 231 (2014).
53. Puyguiraud, A., Gouze, P. & Dentz, M. Upscaling of anomalous pore-scale dispersion. *Transp. Porous Media* **128**, 837–855 (2019).
54. Nguyen, V. & Papavassiliou, D. Velocity magnitude distribution for flow in porous media. *Ind. Eng. Chem. Res.* **60**(38), 13979–13990 (2021).

Acknowledgements

This research is based upon work supported by the U. S. Department of Energy (DOE) under award number DE-SC0019437.

Author contributions

M.B. conceived the research and designed/performed the experiments. M.B., F.M., R.P., L.J.P., and V.M analyzed the results and directed the research. F.M., V.M., and A.B. developed V-TrackMat. M.B. and F.M. wrote the manuscript. All authors reviewed the manuscript.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-024-75581-0.

Correspondence and requests for materials should be addressed to R.P.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.