

Evaluation of Particle Tracking Codes for Dispersing Particles in Porous Media

Marc Berghouse^{1,2}, Filippo Miele³, Lazaro J. Perez¹, Ankur Deep Bordoloi⁴, Veronica Morales³, and Rishi Parashar^{1,*}

¹Desert Research Institute, Division of Hydrologic Sciences, Reno, 89512, USA

²University of Nevada, Reno, Graduate Program of Hydrologic Sciences, Reno, 89557, USA

³US Davis, Civil and Environmental Engineering, Davis, 95616, USA

⁴University of Lausanne, Geosciences and Environment, Lausanne, 1015, Switzerland

*Rishi.Parashar@dri.edu

ABSTRACT

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 at elevated speeds and particle densities. 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.

Introduction

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 flow⁶, bacterial dispersion^{7–17}, chemotaxis¹⁸, biofilm formation^{19,20}, viral transport¹², transport in porous media^{8–11,21}, and colloid filtration²². 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 codes²³. 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 media²⁴. 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 scales²⁵. 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 frames²⁶.

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^{24,27}. Detection errors often stem from overlapping particles, particles out of focus, particles indistinguishable from the background, or varied particle sizes^{28,29}. Linking errors can similarly be attributed to a confluence of factors: high particle speeds and densities, algorithmic inaccuracies, and preexisting detection errors³⁰. Localization error predominantly emerges from the detection algorithm and the signal-to-noise ratio of the imagery²⁴. Linking errors generally represent the most significant 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

ratio³¹, $PSDR = \frac{ipd}{U\Delta t}$, where ipd is the average inter-particle distance detected within each frame, U is the average particle speed, and Δt 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 challenging³², and high particle densities have been shown to increase the sensitivity of PT parameters³³. 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 media²⁴. However, both 3D natural³⁴ and 2D engineered porous materials³⁵ 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 geometry (described in the paragraph below), we used OpenFOAM³⁶ 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, or zero-velocity, positions), and their positions along the pathlines over time resulted in the final ground truth imagery and trajectories (Fig. 1b). Because we wanted to primarily focus on the linking abilities of different PT algorithms, we removed the background (the 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 (Figs. 1d and 1e).

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 lattice⁸. 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 $S_{sim} \geq 2.6$. 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 (particles/pixels²). Particle densities decrease over time as individual particle trajectories disappear or exit the simulation bounds.

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,^{37,38} Trackpy,³⁹ and a new PT method developed by our co-authors named "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. We also tested two deep learning (DL) methods^{40,41}, 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 (\approx 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 FairMOT⁴², 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 ImageJ⁴³, which makes it challenging to script PT analysis. However, TM has shown high levels of accuracy²⁴, 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 method⁴⁴ 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 frames⁴⁵. 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.

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 algorithm⁴⁶, 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 that was developed by some of our co-authors for tracking particles in of their own experiments. 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 suspension⁴⁷ 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 $|v_{pf} - v_{di}|$ 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

146 initial inspection. For V-TrackMat, we detected spots with TM, then exported these spots to Matlab to perform linking with
147 V-TrackMat.

148 Comparative Metrics

149 To understand the true performance of each PT method, we used multiple types of comparative metrics. We used “classical”
150 statistical methods, which are similar to those described in²⁴. In addition, we used experimental statistics and visual analysis
151 of trajectories. The classical statistics used in this study are the false link rate (Fig. 3a), mean path length (Fig. 3b), and the
152 Euclidean distance (Fig. 3c). These statistics are aimed at diagnosing basic problems with particle detection, linking frequency,
153 and linking accuracy (the percent of correct links between two frames). Since we simulate the movement of tracer particles in a
154 microfluidic device under constant flow, we also chose to use experimentally relevant statistics as PT comparison metrics.

155 To compare each PT code for the specific scenario of microbial transport, we used velocity autocorrelations $C_v(\tau) =$
156 $\langle \mathbf{v}(t + \tau) \cdot \mathbf{v}(t) \rangle$, 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,
157 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 $\alpha_t = \arctan\left(\frac{y_{t+2} - y_{t+1}}{x_{t+2} - x_{t+1}}\right) -$
158 $\arctan\left(\frac{y_{t+1} - y_t}{x_{t+1} - x_t}\right)$ (Fig. 3d) data, as our experimental metrics.

159 Results

160 Trajectory Patterns Illuminate PT Method Variations

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

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

183 The unimodal simulations at $PSDR = 4$ (Supplementary Figure 1) further show that all algorithms besides TP have robust
184 performance regardless of geometry. At $PSDR = 1.6$ (Fig. 4c), all algorithms show problems with false links and split
185 trajectories. Similar to the bimodal results, V-TrackMat and TM-Kalman seem to have a larger amount of accurate trajectories
186 than TM-LAP. Thus, general algorithm performance is largely independent of the geometry in which the particles are tracked.
187 However, it should be noted here that the range of possible particle speeds in our simulations, which is largely impacted by
188 geometry and flow conditions, only spans 3-4 magnitudes (Figs. 2a and 2b). High fidelity simulations of Lagrangian particles
189 in porous media show speed distributions that range up to 8 orders of magnitude⁴⁸, so we can’t be confident that our findings
190 (relative rankings of PT performance) would remain accurate for transport in any geometry or flow condition. Furthermore, to
191 focus on linking, we didn’t include any background. However, in real experiments that image bacteria in microfluidic devices,
192 the geometry has a significant impact on tracking performance due to the presence of light scattering around grains²¹.

193 To further understand differences in our PT codes, we plot both the simulated (ground truth) and PT-generated normalized
194 speed distributions for our lowest PSDR bimodal (Fig. 5a) and unimodal (Fig. 5b) simulations. To quantify these differences,
195 we calculate the 1-Wasserstein distance (W_1) between each ground-truth and tracked speed PDF (Table 2). Visual inspection of
196 the PDFs, as well as the trends in (W_1), indicate that TM-Kalman is able to reproduce the simulated speed distributions the best,
197 followed by V-TrackMat, then TM-LAP, then TP. Interestingly, each PT code besides TP overpredicts the fastest speeds for the

198 bimodal simulation, but underpredicts the fastest speeds for the unimodal simulations. During tracking, an effort was made to
199 use the highest possible linking distance that did not result in a significant number of mislinks. Because the range of speeds for
200 the unimodal simulations is greater than that of the bimodal simulations, we were unable to capture the fastest speeds in the
201 unimodal simulation without causing significant false links.

202 Relationship Between Classical Statistics and PSDR

203 To develop a more large-scale understanding of the performance of each PT code, we use a variety of classical and experimental
204 statistics (Fig. 3). Each of these classical statistics target different potential sources of linking error. Because the imagery had a
205 high signal to noise ratio, there were not many errors in the detection stage of PT for each simulation (only occurring due to
206 overlapping particles). Therefore, the false link rate (FLR) primarily shows the potential for a particle to be unlinked, meaning
207 there were no probable candidates for linking in nearby frames (Fig. 3a). The mean path length (MPL) shows the propensity
208 for trajectories to be fractured due to lack of linking (Fig. 3b), and the Euclidean distance (ED) indicates the likelihood for
209 links to move back and forth between particles, sampling a large number of particles for a single trajectory (Fig. 3c). A realistic
210 diagram of each of these potential errors is shown in Figure 3d.

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

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

229 For unimodal simulations, classical statistics (Figs. 6d-6f) generally perform better than their bimodal counterparts. The
230 bimodal simulations have higher mean speeds than the unimodal solutions (Table 1). Furthermore, the bimodal simulations
231 (Fig. 1a) have a larger number of particles at high speeds, which causes more difficulty in particle tracking. In addition, the
232 unimodal simulations show a greater range of speeds and are generally more reminiscent of speed distributions of particles in
233 porous media⁴⁹. Thus, the unimodal simulations likely offer a more comprehensive representation of generic PT code efficacy
234 in porous media. While the general trends mirror those in the bimodal findings, V-TrackMat performs comparatively better
235 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
236 aforementioned challenges with fast-moving particles mean its performance slightly improves in unimodal settings, which
237 aren't dominated by high speed trajectories. Still, TP's mean path lengths (Fig. 6e) depict a sharp decline as PSDR decreases,
238 implying the persistent issue of trajectory splitting in both bimodal and unimodal settings.

239 The results of the classical statistics imply that TM-Kalman and TM-LAP outperform V-TrackMat in all cases, but from the
240 sample trajectories (Fig. 4), we have shown this to not be true. Also, the trajectories show TP performs much worse than the
241 other algorithms at low PSDR, but this is not reflected by the FLR and ED metrics. We posit that the primary reason for the
242 disconnect between the classical statistics and the sample trajectories is that the FLR and ED metrics underpenalize aggressive
243 linking. The FLR will always be lower when more links are forced, since the probability of false positive detection is very
244 low. The ED metric will always be higher when more links occur between different trajectories, but if the trajectories are
245 nearest neighbors, then the error will be relatively small. Thus, long trajectories and links across the pore space (such as those
246 of V-TrackMat and TM-Kalman) will result in more ED error than zig-zagging trajectories between close neighbors (such as
247 TM-LAP) will. Ultimately, the FLR and ED underpredict PT error for nearest-neighbor based algorithms with little constraint
248 for linking. As a result, these statistics fail to grasp the nuanced differences between PT codes.

249 Beyond comparing PT codes' performances, we also demonstrate that all classical statistics have a power law relationship
250 with PSDR, although some relationships are more significant than others (Table 3). As PSDR is reduced, all PT codes generally
251 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 (Figs. 6a and 6d). 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 (Figs. 6d-6f) present slightly different power law relationships compared to those from bimodal simulations (Figs. 6a-6c). 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 — comparative statistics or visual trajectory inspection — 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 Figure 2), both the V-TrackMat and TM algorithms predict speed and angle statistics with near perfection. V-TrackMat and TM-Kalman perform slightly better 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 Figures 3 and 4. 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 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 Figure 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 Figure 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.

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 Figures 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 8b), 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 in achieving reliable PT outcomes. As a benchmark, 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.

PT Algorithm Speed Comparison

In addition to performance analysis, we also report how long each PT code takes to link trajectories (Fig. 9). Generally, TM – LAP is the fastest linking algorithm, then TP, then TM – Kalman, and V-TrackMat is the slowest. Thus, we observe a significant tradeoff 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, Matlab 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 algorithms. In addition, we show that TM-Kalman, and V-TrackMat, which both use particle

358 velocities to enhance predictions, exhibit marked improvements at $PSDR = 1.1$ relative to TM-LAP. Although TP also uses
359 particle velocity information to make linking predictions, the SubnetOversizeException issue was much more significant than
360 any potential gain due to velocity-based predictions. Thus, the leveraging of velocity data, especially for particles dispersing in
361 porous media which exhibit a wide range of potential speeds, emerges as a critical factor in bolstering PT predictions.

362 Beyond algorithmic evaluations, our analysis discusses some of the common errors in transport analysis that emerge as a
363 result of PT error. We find that all PT codes underestimate particle speeds and overestimate turn angles, and that poor tracking
364 causes a significant loss in the accuracy of reproducing cyclical autocorrelations. Furthermore, we advocate for the recording of
365 video data at a minimum threshold of $PSDR \geq 3$, a benchmark that promotes reliable tracking irrespective of the algorithm
366 employed.

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

372 Looking forward, we envision our research catalyzing advancements in PT theory in three pivotal aspects. Firstly, we
373 urge future studies to transcend the boundaries of classical statistics in PT code comparisons, emphasizing the integration of
374 experimental outcomes pertinent to specific applications. Secondly, our findings echo the effectiveness of velocity-centric PT
375 methods for dispersive particles in porous media, extending their proven efficacy from constant velocity scenarios²⁴ to contexts
376 characterized by large velocity fluctuations in time and space. Lastly, we highlight the importance of maintaining a reasonably
377 high PSDR to achieve precise particle transport analysis.

378 Data Availability

379 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
380 the corresponding author (Rishi Parashar).

381 References

- 382 1. Saxton, M. Single-particle tracking: the distribution of diffusion coefficients. *Biophys. journal* **72**(4), 1744–1753 (1997).
- 383 2. Hong, Q., Sheetz, M. & Elson, E. Single particle tracking. analysis of diffusion and flow in two-dimensional systems. *Biophys. journal* **60**(4), 910–921 (1991).
- 384 3. Daumas, F. *et al.* Confined diffusion without fences of a g-protein-coupled receptor as revealed by single particle tracking. *Biophys. journal* **84**(1), 356–366 (2003).
- 385 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).
- 386 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).
- 387 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-Science Rev.* **155**, 93–128 (2016).
- 388 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. The Royal Soc. Interface* **17**, 20200046, DOI: [10.1098/rsif.2020.0046](https://doi.org/10.1098/rsif.2020.0046).
- 389 8. Dehkarghani, 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).
- 390 9. Bhattacharjee, T. & Datta, S. Bacterial hopping and trapping in porous media. *Nat. communications* **10**(1), 2075 (2019).
- 391 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).
- 392 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).
- 393 12. Vallotton, P. *et al.* Diatrack particle tracking software: Review of applications and performance evaluation. *Traffic* **18**(12), 840–852 (2017).
- 394 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).

- 406 14. Ewers, H. & Schelhaas, M. Effect of motility on the transport of bacteria populations through a porous medium. *Methods enzymology* **506**, 63–80 (2012).
- 407 15. Dehkharghani, A., Waisbord, N. & Guasto, J. S. Self-transport of swimming bacteria is impaired by porous microstructure. *Commun. Phys.* **6**, 18, DOI: [10.1038/s42005-023-01136-w](https://doi.org/10.1038/s42005-023-01136-w) (2023).
- 410 16. Rusconi, R., Guasto, J. S. & Stocker, R. Bacterial transport suppressed by fluid shear. *Nat. Phys.* **10**, 212–217, DOI: [10.1038/nphys2883](https://doi.org/10.1038/nphys2883) (2014).
- 411 17. Bhattacharjee, T. & Datta, S. S. Bacterial hopping and trapping in porous media. *Nat. Commun.* **10**, 2075, DOI: [10.1038/s41467-019-10115-1](https://doi.org/10.1038/s41467-019-10115-1) (2019).
- 412 18. Jeon, H. *et al.* Quantitative analysis of single bacterial chemotaxis using a linear concentration gradient microchannel. *Biomed. microdevices* **11**, 1135–1143 (2009).
- 413 19. Birjiniuk, A. *et al.* Single particle tracking reveals spatial and dynamic organization of the escherichia coli biofilm matrix. *New journal physics* **16**(8), 085014 (2014).
- 414 20. Secchi, E. *et al.* The effect of flow on swimming bacteria controls the initial colonization of curved surfaces. *Nat. Commun.* **11**, 2851, DOI: [10.1038/s41467-020-16620-y](https://doi.org/10.1038/s41467-020-16620-y) (2020).
- 415 21. Wu, H. & Schwartz, D. Nanoparticle tracking to probe transport in porous media. *Accounts Chem. Res.* **53**(10), 2130–2139 (2020).
- 416 22. Linkhorst, J., Beckmann, T., Go, D., Kuehne, A. J. & Wessling, M. Microfluidic colloid filtration. *Sci. reports* **6**(1), 22376 (2016).
- 417 23. Cheng, H., Hsu, C., Hung, C. & Lin, C. A review for cell and particle tracking on microscopy images using algorithms and deep learning technologies. *biomedical journal* **45**(3), 465–471 (2022).
- 418 24. Chenouard, N. *et al.* Objective comparison of particle tracking methods. *Nat. methods* **11**(3), 281–289 (2014).
- 419 25. 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).
- 420 26. Meijering, E., Dzyubachyk, O. & Smal, I. Methods for cell and particle tracking. *Methods enzymology* **504**, 183–200 (2012).
- 421 27. Clarke, D. & Martin-Fernandez, M. A brief history of single-particle tracking of the epidermal growth factor receptor. *Methods protocols* **2**(1), 12 (2019).
- 422 28. Wiggins, C., Santos, R. & Ruggles, A. A feature point identification method for positron emission particle tracking with multiple tracers. *Nucl. Instruments Methods Phys. Res. Sect. A: Accel. Spectrometers, Detect. Assoc. Equip.* **843**, 22–28 (2017).
- 423 29. Travers, T., Colin, V., Loumagine, M., Barillé, R. & Gindre, D. Single-particle tracking with scanning non-linear microscopy. *Nanomaterials* **10**(8), 1519 (2020).
- 424 30. Maska, M. & Matula, P. Particle tracking accuracy measurement based on comparison of linear oriented forests. In *Proc. IEEE Int. Conf. on Comput. Vis. Work.* 11–17 (2017).
- 425 31. Malik, N., Dracos, T. & Papantoniou, D. Particle tracking velocimetry in three-dimensional flows: Part ii: Particle tracking. *Exp. fluids* **15**, 279–294 (1993).
- 426 32. Baek, S. & Lee, S. A new two-frame particle tracking algorithm using match probability. *Exp. Fluids* **22**, 23–32 (1996).
- 427 33. 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).
- 428 34. Bijeljic, B., Mostaghimi, P. & Blunt, M. J. Signature of Non-Fickian Solute Transport in Complex Heterogeneous Porous Media. *Phys. Rev. Lett.* **107**, 204502, DOI: [10.1103/PhysRevLett.107.204502](https://doi.org/10.1103/PhysRevLett.107.204502) (2011).
- 429 35. 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, DOI: [10.1103/PhysRevFluids.2.124103](https://doi.org/10.1103/PhysRevFluids.2.124103) (2017).
- 430 36. Weller, H., Tabor, G., Jasak, H. & Fureby, C. A tensorial approach to computational continuum mechanics using object-oriented techniques. *Comput. Phys.* **12**, 620 (1998).
- 431 37. Tinevez, J. *et al.* Trackmate: An open and extensible platform for single-particle tracking. *Methods* **115**, 80–90 (2017).
- 432 38. Ershov, D. *et al.* Trackmate 7: integrating state-of-the-art segmentation algorithms into tracking pipelines. *Nat. Methods* **19**(7), 829–832 (2022).

- 454 39. Allan, D., Caswell, T., Keim, N., van der Wel, C. & Verweij, R. soft-matter/trackpy: Trackpy v0. 5.0., DOI: <https://doi.org/10.5281/zenodo.4682814> (2021).
- 455 40. Midtvedt, B. *et al.* Quantitative digital microscopy with deep learning. *Appl. Phys. Rev.* **8**(1) (2021).
- 456 41. Fatemi, B., Halcrow, J. & Jaqaman, K. Geometric deep learning of particle motion by magik. *Nat. Mach. Intell.* **5**(5), 457 483–484 (2023).
- 458 42. Zhang, Y., Wang, C., Wang, X., Zeng, W. & Liu, W. Fairmot: On the fairness of detection and re-identification in multiple 459 object tracking. *Int. J. Comput. Vis.* **129**, 3069–3087 (2021).
- 460 43. Abràmoff, M., Magalhães, P. & Ram, S. Image processing with imagej. *Biophotonics international* **11**(7), 36–42 (2004).
- 461 44. Kalman, R. A new approach to linear filtering and prediction problems. *Basic Eng.* **82**, 35–45 (1960).
- 462 45. Jaqaman, K. *et al.* Robust single-particle tracking in live-cell time-lapse sequences. *Nat. methods* **5**(8), 695–702 (2008).
- 463 46. Crocker, J. & Grier, D. Methods of digital video microscopy for colloidal studies. *J. Colloid Interface Sci.* **179**(1), 298–310 464 (1996).
- 465 47. A. Alam, E. *Crowd of Microswimmers*. Phd thesis, Condensed Matter [cond-mat]: Université Grenoble Alpes, Université 466 Grenoble Alpes (2022). Available at <https://theses.hal.science/tel-03894880/document>.
- 467 48. Puyguiraud, A., Gouze, P. & Dentz, M. Upscaling of anomalous pore-scale dispersion. *Transp. Porous Media* **128**, 837–855 468 (2019).
- 469 49. Nguyen, V. & Papavassiliou, D. Velocity magnitude distribution for flow in porous media. *Ind. & Eng. Chem. Res.* **60**(38), 470 13979–13990 (2021).

472 Acknowledgements

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

474 Author contributions statement

475 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
476 and directed the research. F.M., V.M., and A.B. developed V-TrackMat. M.B. and F.M. wrote the manuscript. All authors
477 reviewed the manuscript.

478 Competing Interests

479 The author(s) declare no competing interests.

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.

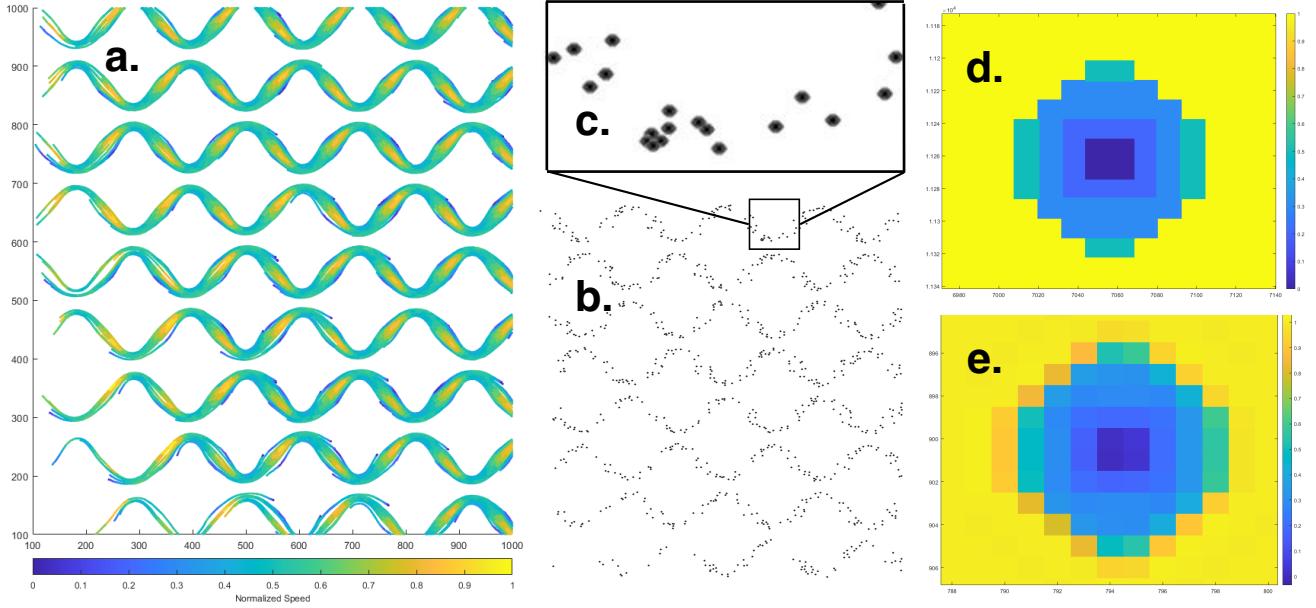


Figure 1. This figure shows the general workflow for the construction of simulated trajectories. (a) Pathlines for the bimodal simulation at $PSDR = 1.1$ colored by normalized speed. (b) Simulated imagery analyzed by various particle tracking methods. (c) Zoomed in section of simulated particles showing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

	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.

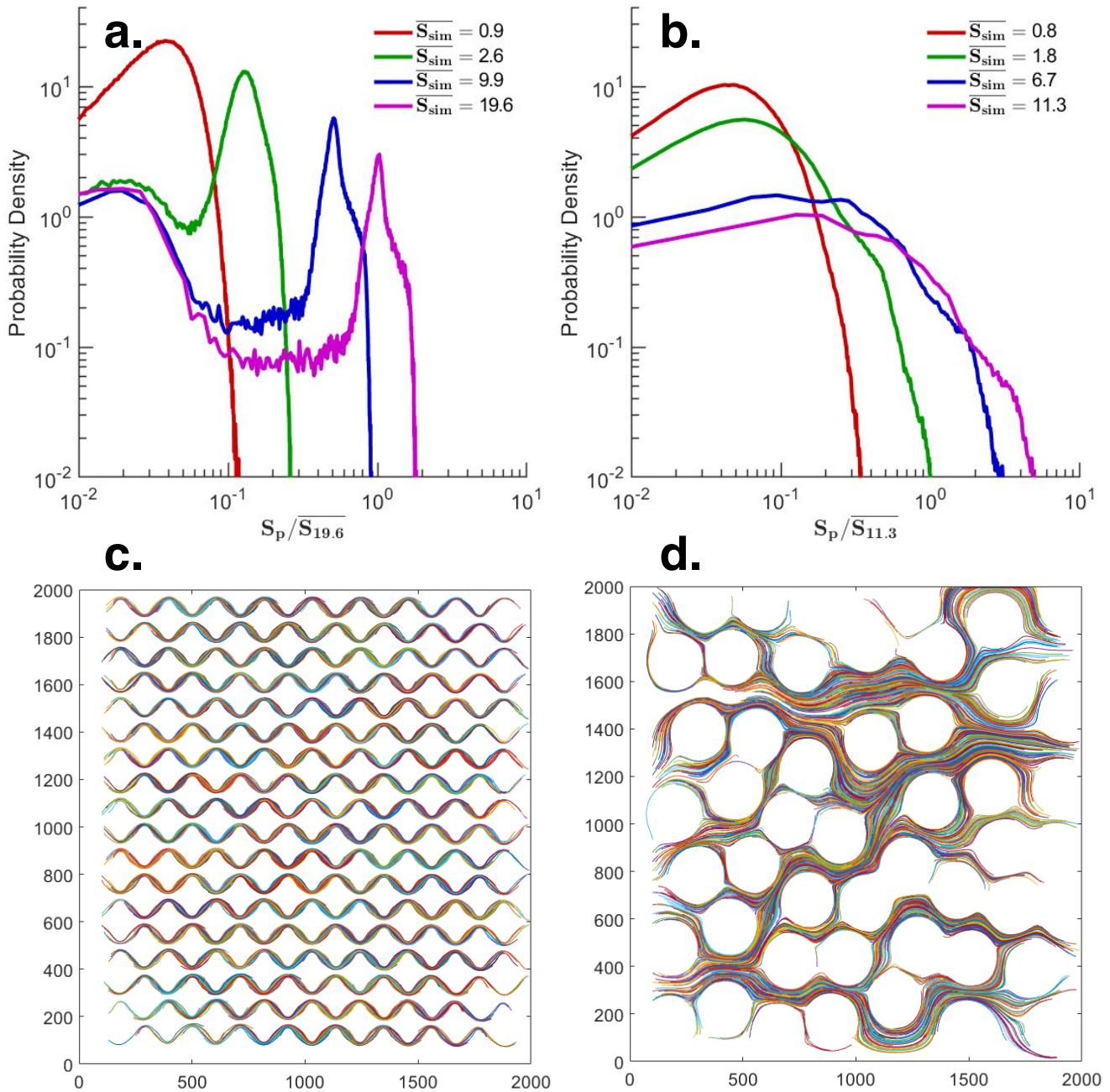


Figure 2. This figure shows the particle speed distributions for the bimodal (a) and unimodal (b) simulations, and the simulated pathlines for a bimodal (c) and unimodal (d) simulation for $PSDR = 2.3$ and $PSDR = 2.6$ respectively. The speed distributions in (a) are normalized by the mean speed of the 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.

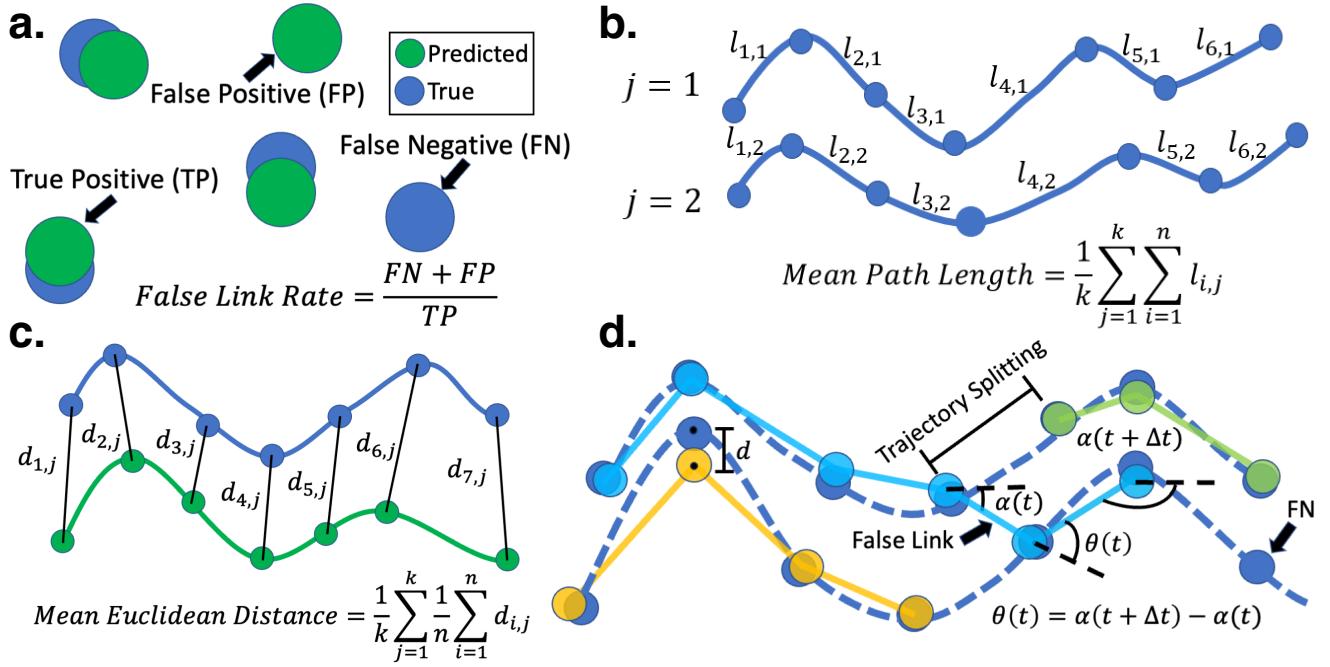


Figure 3. This figure shows visual representations of all classical statistics, and one experimental statistic, used to compare particle tracking performance in this paper. In all panels, predicted points and trajectories are green, and ground truth points and trajectories are blue. (a) Snapshot of particles 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.

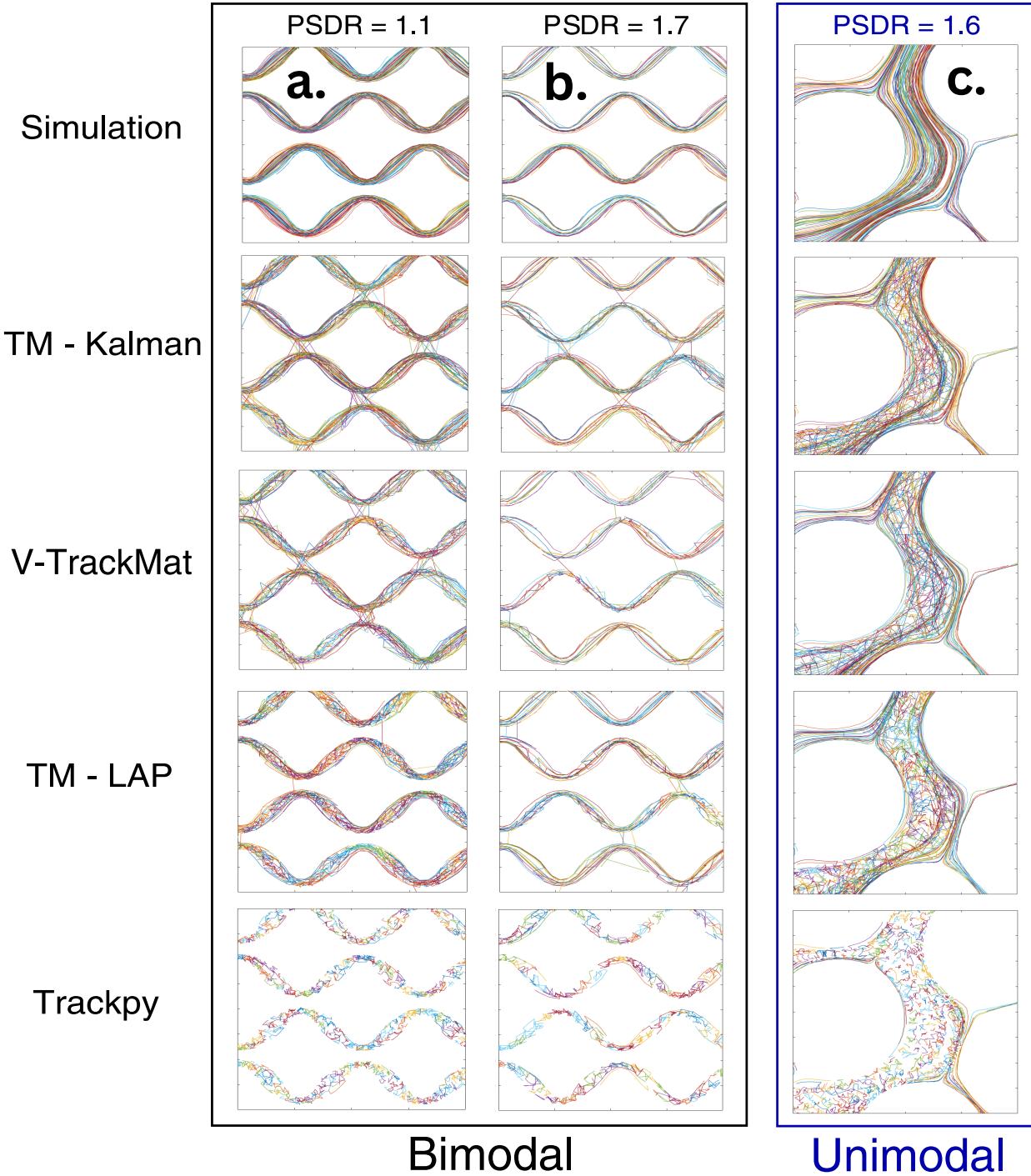


Figure 4. Sample trajectories for all PT codes for the low- $PSDR$ bimodal and unimodal simulations. Each plot shows a 200x200 pixel section of the whole domain. Within a plot, each line corresponds to a unique trajectory (with random colors used to show the contrast between individual trajectories). (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 significant.

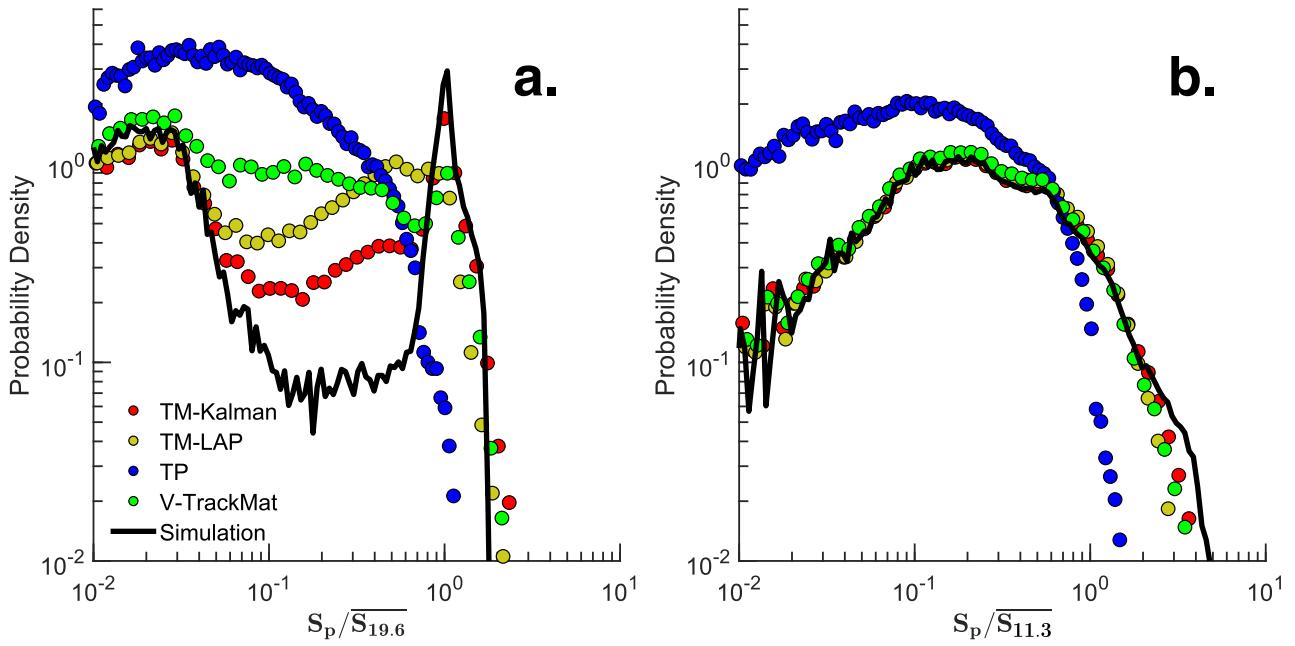


Figure 5. This figure shows the a comparison between the speed distributions for each PT code for the lowest-PSDR bimodal (a) and lowest-PSDR unimodal (b) simulations. The speed distributions in (a) are normalized by the mean speed of the lowest-PSDR bimodal simulation (19.6 pixels/frame). The speed 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	<i>a</i>	<i>b</i>	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	<i>a</i>	<i>b</i>	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.

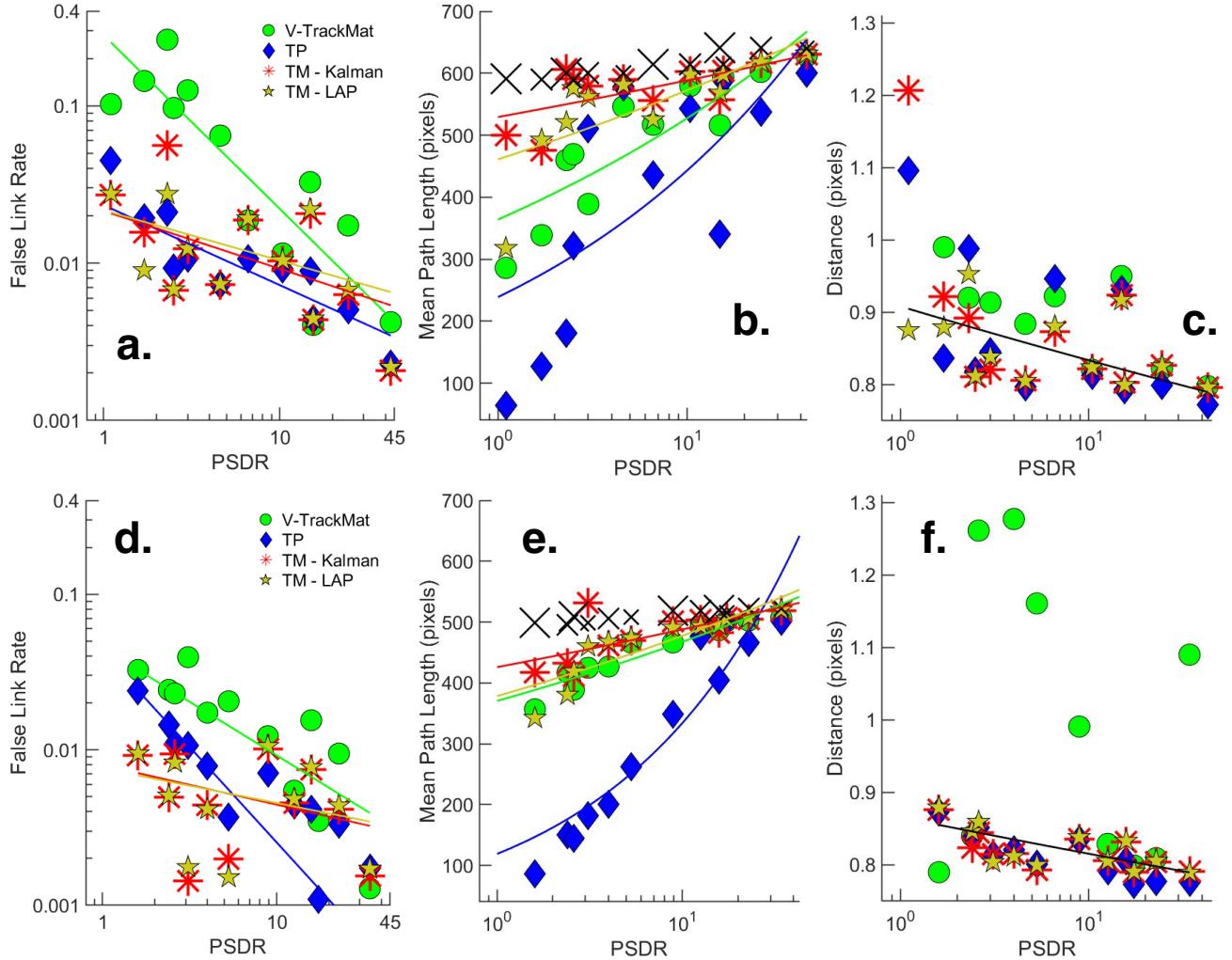


Figure 6. This figure shows the results of the classical comparative statistics for both the bimodal and unimodal simulations. For all plots, the size of the scatter points represent the particle density of the simulation (larger points means greater particle density). a-c correspond to the bimodal simulations, 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.

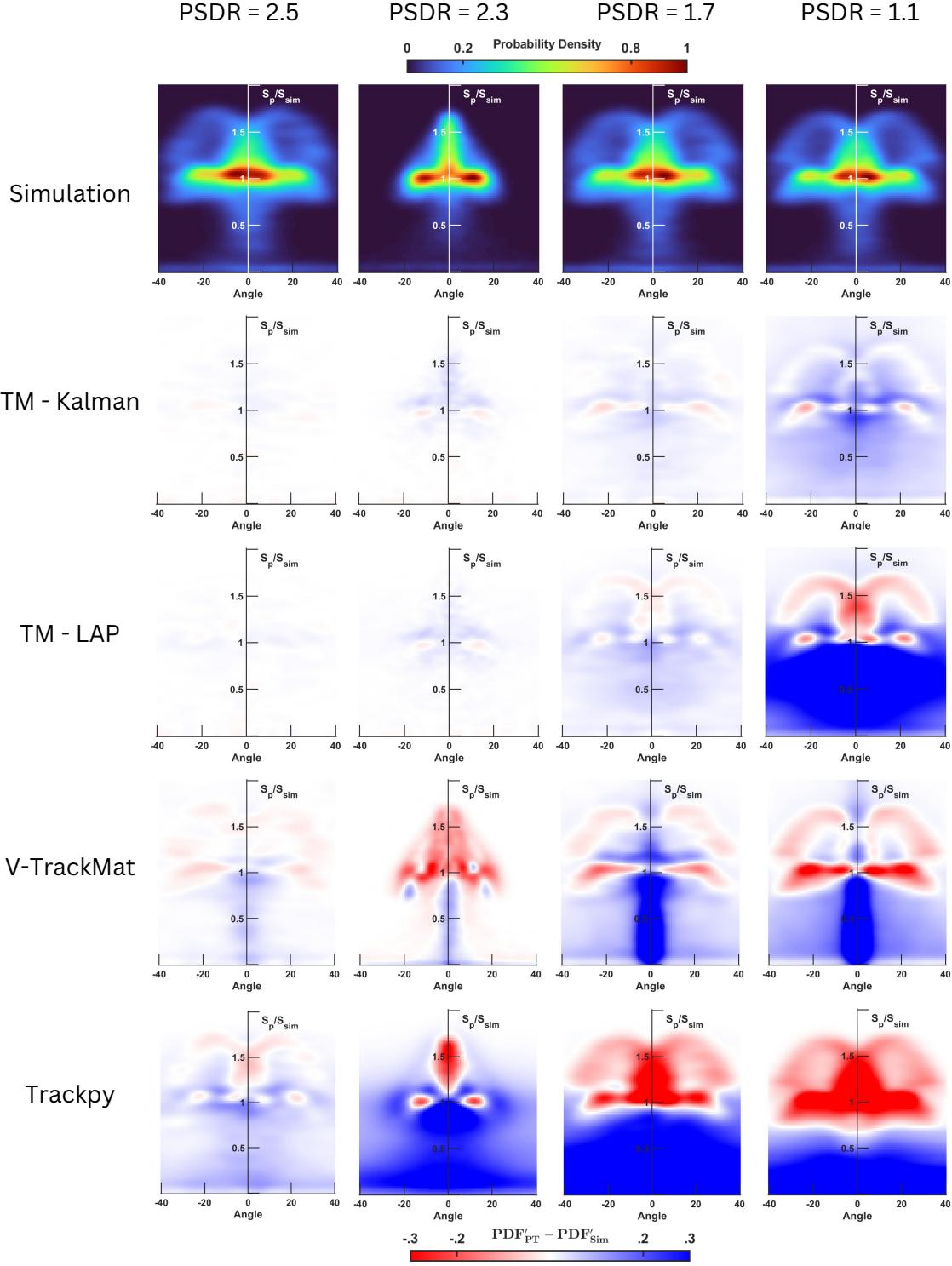
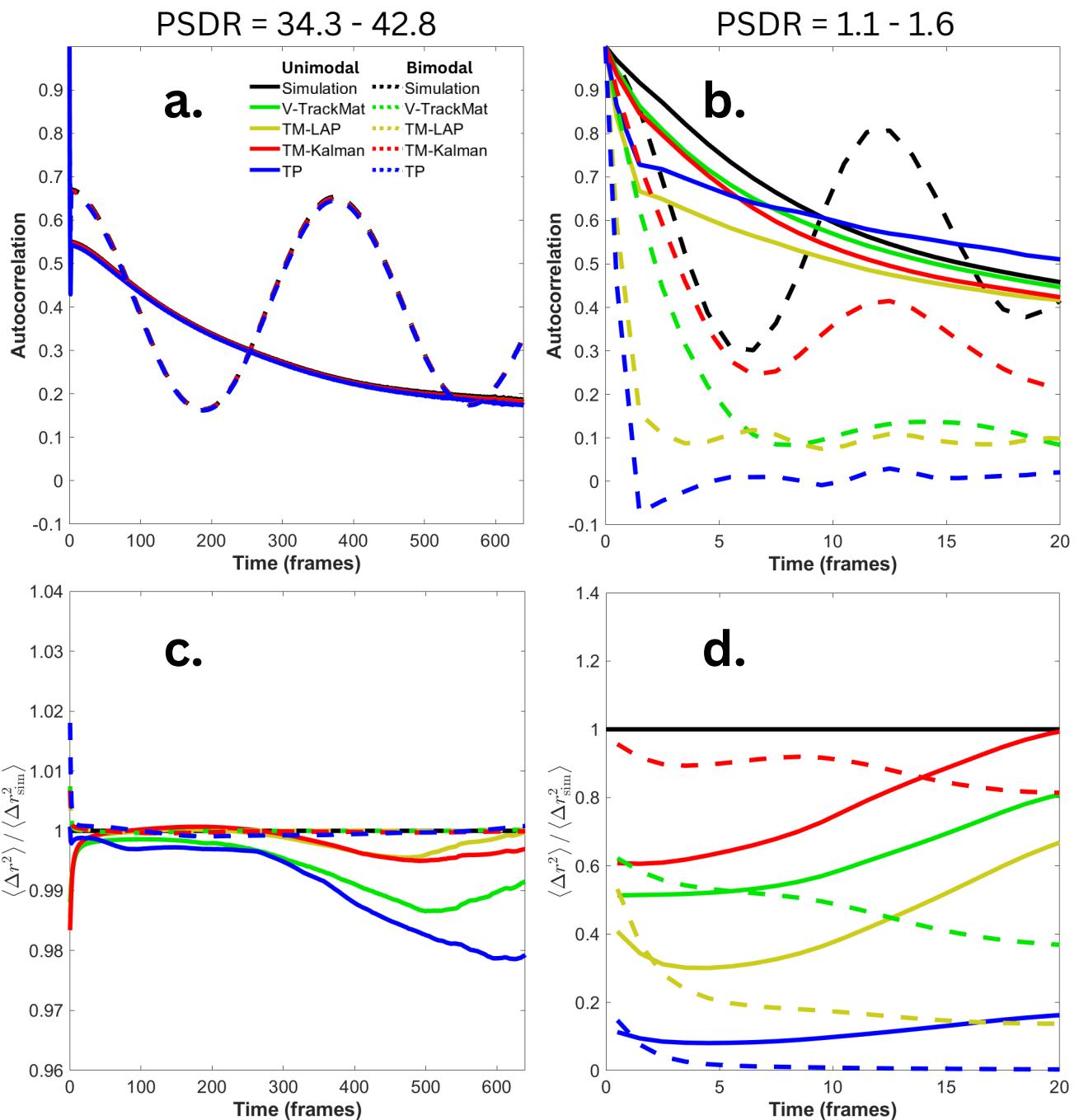


Figure 7. Speed-angle joint probability density difference heatmaps for the bimodal simulation. Speeds determined from particle tracking (S_p) are normalized by the mean speed of the respective simulation (S_{sim}). Red corresponds to an underprediction of probability density, blue corresponds to an 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.



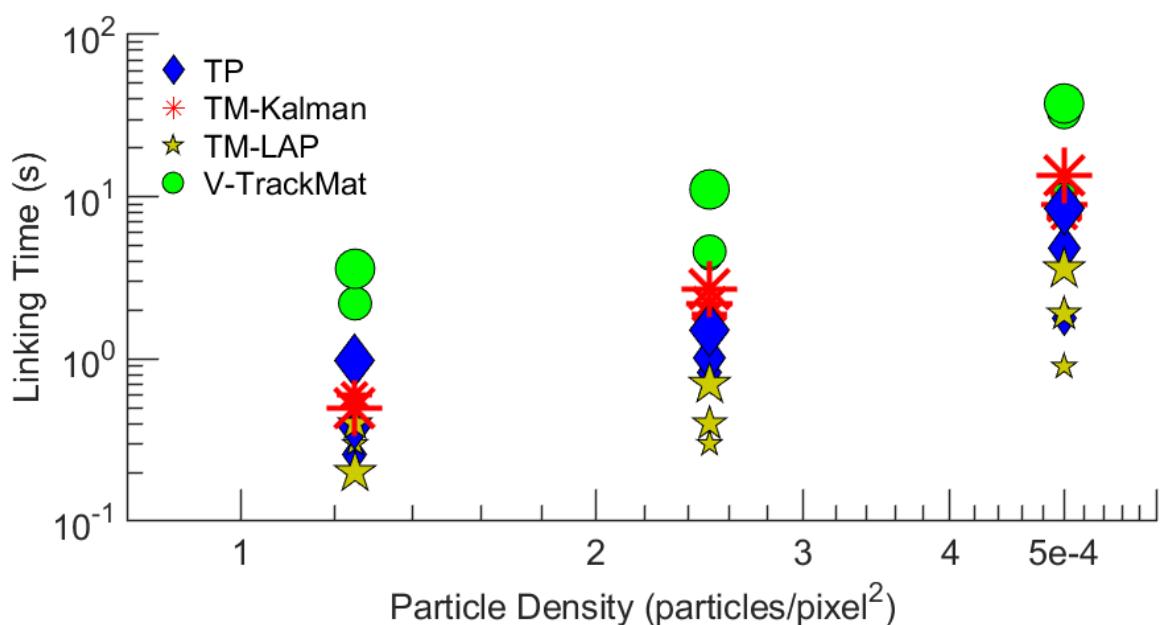


Figure 9. Amount of time each PT code takes during the linking stage for selected unimodal simulations. Simulation speed is represented by scatter point size (large = 9.9 px/frame, medium = 2.6 px/frame, small = 0.9 px/frame). V-TrackMat consistently has the longest linking times, and TM-LAP has the 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.