Medical Image Analysis 72 (2021) 102128

Contents lists available at ScienceDirect

Medical Image Analysis

journal homepage: www.elsevier.com/locate/media

Deep probabilistic tracking of particles in fluorescence microscopy images

Roman Spilger[a,*], Ji-Young Lee[b,c], Vadim O. Chagin[d,e], Lothar Schermelleh[f], M. Cristina Cardoso[d], Ralf Bartenschlager[b,c], Karl Rohr[a]

[a] Biomedical Computer Vision Group, Heidelberg University, BioQuant, IPMB, and DKFZ Heidelberg, Heidelberg 69120, Germany
[b] Department of Infectious Diseases, Molecular Virology, Heidelberg University, Heidelberg 69120, Germany
[c] German Center for Infection Research, Heidelberg Partner Site
[d] Cell Biology and Epigenetics, Department of Biology, Technische Universität Darmstadt, Darmstadt 64287, Germany
[e] Institute of Cytology, Russian Academy of Sciences, St. Petersburg, Russia
[f] Micron Advanced Bioimaging Unit, Department of Biochemistry, University of Oxford, Oxford OX1 3QU, United Kingdom

ARTICLE INFO

Article history:
Received 25 August 2020
Revised 14 May 2021
Accepted 26 May 2021
Available online 8 June 2021

Keywords:
Biomedical imaging
Microscopy images
Tracking
Deep learning

ABSTRACT

Tracking of particles in temporal fluorescence microscopy image sequences is of fundamental importance to quantify dynamic processes of intracellular structures as well as virus structures. We introduce a probabilistic deep learning approach for fluorescent particle tracking, which is based on a recurrent neural network that mimics classical Bayesian filtering. Compared to previous deep learning methods for particle tracking, our approach takes into account uncertainty, both aleatoric and epistemic uncertainty. Thus, information about the reliability of the computed trajectories is determined. Manual tuning of tracking parameters is not necessary and prior knowledge about the noise statistics is not required. Short and long-term temporal dependencies of individual object dynamics are exploited for state prediction, and assigned detections are used to update the predicted states. For correspondence finding, we introduce a neural network which computes assignment probabilities jointly across multiple detections as well as determines the probabilities of missing detections. Training requires only simulated data and therefore tedious manual annotation of ground truth is not needed. We performed a quantitative performance evaluation based on synthetic and real 2D as well as 3D fluorescence microscopy images. We used image data of the Particle Tracking Challenge as well as real time-lapse fluorescence microscopy images displaying virus structures and chromatin structures. It turned out that our approach yields state-of-the-art results or improves the tracking results compared to previous methods.

1. Introduction

Accurate particle tracking is of fundamental importance to quantify the dynamic behavior of intracellular structures (e.g., chromatin structures, receptors) and virus structures from time-lapse fluorescence microscopy images. Since the spatial resolution of optical microscopy is limited by diffraction, these small structures tagged with fluorescent probes appear as blurred spots (particles) in microscopy images. Manual tracking of fluorescent particles from large live cell data sets is not feasible, thus fully automatic approaches are indispensable, which have to deal with low signal-to-noise ratio (SNR), complex motion behavior, and high object density. Further challenges are out of focus movement, (almost) indistinguishable object appearance, particle clustering, and photobleaching.

In previous work on tracking particles in fluorescence microscopy images, classical deterministic and probabilistic approaches have been introduced. Deterministic approaches comprise two steps: Particle detection and correspondence finding (e.g., Sbalzarini and Koumoutsakos, 2005; Collinet et al., 2010; Ruhnow et al., 2011; Applegate et al., 2011; Paavolainen et al., 2012). While being computationally efficient, these approaches do not take into account spatial and temporal uncertainties which often leads to difficulties under challenging conditions (e.g., low SNR, high object density). In comparison, probabilistic approaches follow a Bayesian paradigm and define a posterior distribution on the variables describing the object state. The posterior can be resolved via a se-

* Corresponding author.
E-mail addresses: roman.spilger@bioquant.uni-heidelberg.de (R. Spilger), k.rohr@uni-heidelberg.de (K. Rohr).

https://doi.org/10.1016/j.media.2021.102128
1361-8415/

quential Bayesian filter such as the Kalman filter (e.g., Genovesio et al., 2006; Yang et al., 2012; Chenouard et al., 2013; Godinez and Rohr, 2015; Roudot et al., 2017; Ritter et al., 2018) or the particle filter (e.g., Smal et al., 2008; Godinez et al., 2009; Cardinale et al., 2009; Yuan et al., 2012). However, probabilistic approaches typically require selecting a suitable dynamic model and use prior assumptions about the noise statistics (e.g., image and motion noise), which do not necessarily hold. Moreover, classical tracking methods often involve numerous parameters that are difficult to adjust, particularly for non-experts, and do not always have a biophysical interpretation.

Deep learning methods provide state-of-the-art performance in various computer vision tasks including image classification, object detection, and segmentation (e.g., LeCun et al., 2015; Greenspan et al., 2016; Litjens et al., 2017). Approaches for tracking objects in video images of natural scenes (e.g., pedestrians, cars) use deep learning for different purposes (Ciaparrone et al., 2020). Convolutional neural networks (CNNs) are employed for extracting appearance features (e.g., Ullah and Alaya Cheikh, 2018), for generating a discriminative appearance model (e.g., Chen et al., 2016), for object detection (e.g., He et al., 2017; Ma et al., 2020), for computing assignment scores (e.g., Chen et al., 2017), and for motion prediction (e.g., Wang et al., 2017; Hernandez et al., 2019). Recurrent neural networks (RNNs) are often used to compute assignment scores between tracklets and detections (or tracklets and other tracklets) (e.g., Milan et al., 2017), which typically exploit appearance features (e.g., Sadeghian et al., 2017; Zhu et al., 2018) which can hardly be exploited to track indistinguishable particles. In Farrell et al. (2017), CNNs and an RNN are combined for correspondence finding between detector hits in simulated high-energy physics data without using prediction and update steps as in Bayesian filtering.

Recently, deep learning methods for tracking biological objects in microscopy images have been introduced showing promising results. CNNs and RNNs have been used to exploit appearance features for cell tracking (e.g., He et al., 2017; Payer et al., 2019; Hayashida and Bise, 2019; Nishimoto et al., 2019). Since cells (and natural objects) are very different from fluorescent particles both regarding shape and dynamics, these approaches cannot be directly applied for particle tracking. In addition, appearance features are not a reliable cue for correspondence finding for (almost) indistinguishable particles. Few works employed CNNs for particle detection in fluorescence microscopy images (e.g., Gudla et al., 2017; Newby et al., 2018; Wollmann et al., 2019; Dmitrieva et al., 2019). In Zhong et al. (2018), a CNN was used to determine the position of individual polystyrene particles along the z-direction of epifluorescence microscopy images. Sun and Paninski (2018) proposed an RNN which approximates the posterior transition probability densities to track clathrin-coated pits. However, for correspondence finding a classical nearest neighbor strategy is used. RNNs using (past) temporal information for correspondence finding of fluorescent particles were introduced in Yao et al. (2018) and Spilger et al. (2018). Smal et al. (2019) employ a denoising autoencoder and score matching to learn a motion model from data within a classical multiple hypotheses tracking (MHT) framework. Spilger et al. (2020) introduced a bidirectional RNN exploiting past and future information as well as multiple track hypotheses for correspondence finding. Yao et al. (2020) described an RNN that uses handcrafted and learned features. However, none of these deep learning methods takes into account uncertainty, neither in the network model (epistemic uncertainty) nor the inherent noise in the image data (aleatoric uncertainty).

Deep neural networks considering *uncertainty* have been introduced for natural and medical images for different tasks such as segmentation (e.g., street traffic scenes, CT images), disease detection (e.g., fundus images), super-resolution (e.g., diffusion MR images), and image translation (e.g., CT images to MR images) (e.g., Leibig et al., 2017; Esser and Sutter, 2018; Kohl et al., 2018; Tanno et al., 2019). Since neural networks generally consist of a large number of parameters as well as non-linear activations, computing the (multi-modal) posterior distribution of a network output is intractable. Thus, approximation methods have been introduced, which are mainly based on Bayesian inference or Monte-Carlo sampling. *Bayesian neural networks* represent the parameters (weights) by probability distributions instead of using single values (e.g., Kingma and Welling, 2014; Blundell et al., 2015; Hernández-Lobato and Adams, 2015; Wang et al., 2016). Consequently, the network outputs can also be represented by probability distributions and calculated analytically employing graphical models (Su et al., 2016) or non-linear belief networks (Frey and Hinton, 1999). Alternatively, *Monte-Carlo sampling* can be employed. Often, Monte-Carlo samples are obtained using ensembles of neural networks. These ensembles can be generated by differently trained neural networks (e.g., Lakshminarayanan et al., 2017; Lee et al., 2019) or employing dropout during training and testing (Monte-Carlo dropout, Gal and Ghahramani, 2016). However, Monte-Carlo sampling approaches include epistemic uncertainty (model uncertainty) but not aleatoric uncertainty (data uncertainty). In Kendall and Gal (2017), Monte-Carlo dropout was employed to capture epistemic uncertainty and a standard deviation variable is added to each output to include aleatoric uncertainty. None of the above described methods considering uncertainty was employed for object tracking in microscopy images.

In this contribution, we present a novel deep neural network architecture for tracking particles in fluorescence microscopy images which exploits both aleatoric and epistemic uncertainty. Inspired by classical Bayesian filtering, the network learns to predict the next state and to correct the predicted state based on an assigned detection. Gated recurrent units (GRUs) (Cho et al., 2014) are used to exploit both short- and long-term temporal dependencies of individual object dynamics. Epistemic uncertainty is incorporated by variational Bayesian learning using an approximation of the lower bound for efficient learning (Kingma and Welling, 2014; Blundell et al., 2015), which does not require computationally expensive iterative inference schemes such as Markov chain Monte Carlo. Bayesian layers with reparameterization are employed, where parameters are represented by Gaussian distributions. During network training via variational inference the parameters of these probability distributions are learned instead of directly learning the network weights. To capture aleatoric uncertainty due to the particle detector and noise of object motion, the network learns estimating the mean and standard deviation of Gaussian distributions from which the predicted and updated state can be determined. We also introduce a neural network that determines assignment probabilities for correspondence finding based on the Euclidean distance between the predicted states and particle detections obtained by the spot-enhancing filter (Sage et al., 2005) and Gaussian fitting. Assignment probabilities are computed jointly across multiple detections, and probabilities of missing detections are also determined. Network training is based on synthetic data only and manually annotated data is not needed. We propose a novel scheme to generate synthetic training images using automatically extracted information from the images in an application. This enables simulating a large number and spectrum of training images that represent well the images in an application. In contrast, our previous scheme in Spilger et al. (2020) did not use automatically extracted information from the real data to generate training images. The uncertainty information determined by our approach for the computed trajectories is important to assess their reliability and, for example, to exclude unreliable tracks (or track points) to increase the accuracy of subsequent motion analysis (e.g., mean-squared displacement analysis) as we show in our experiments. In

addition, we demonstrate that the uncertainty can be exploited to assess the suitability of the training data and to select the generated training data set with the best-suited motion model so that the training data better represents the real data in an application.

Our approach is the first probabilistic deep learning method for particle tracking in microscopy images and takes into account both aleatoric and epistemic uncertainty. We verified that both types of uncertainty are captured by the network. Besides taking into account and exploiting uncertainty information, we propose a novel neural network architecture which differs from our previous work in Spilger et al. (2020), where prediction and update steps as in Bayesian filtering were not used nor Bayesian layers and GRU layers. In addition, we here use a different loss function, namely a balanced focal loss (Lin et al., 2020) and different activation functions (PReLUs, He et al., 2015). We have conducted a quantitative performance evaluation based on synthetic and real 2D as well as 3D fluorescence microscopy images. We used data from the Particle Tracking Challenge as well as real live cell microscopy image sequences displaying the hepatitis C virus (HCV) protein NS5A, the HCV associated protein ApoE, and chromatin structures (labeled during DNA replication). It turned out that our approach yields state-of-the-art or improved results compared to previous methods.

2. Methods

In this section, we present our novel probabilistic deep learning approach for tracking multiple particles in live cell fluorescence microscopy images, denoted as Deep Probabilistic Particle Tracker (DPPT). First, we give an overview of our approach. Then, we describe the classical Bayesian filtering framework used in previous particle tracking approaches. After that, we introduce our deep learning architecture mimicking classical Bayesian filtering and taking into account both aleatoric and epistemic uncertainty. We also present a neural network architecture to compute assignment probabilities for correspondence finding. Finally, we provide details on the network training.

2.1. Overview of the proposed tracking approach

Fig. 1 provides a schematic overview of the proposed DPPT approach. For particle detection, we employ the spot-enhancing filter (SEF) (Sage et al., 2005) and Gaussian fitting yielding a set of detections represented by image positions. For state prediction and state update, a recurrent neural network (RNN) with gated recurrent units (GRUs) (Cho et al., 2014) is introduced mimicking classical Bayesian filtering. The network takes into account both aleatoric and epistemic uncertainty. Epistemic uncertainty is captured by learning Gaussian distributions for the parameters of the network. To take into account aleatoric uncertainty, the network estimates the mean and standard deviation of Gaussian distributions from which the predicted and updated states are computed. For correspondence finding, we introduce a neural network that determines assignment probabilities as well as probabilities of missing detections between the predicted states and particle detecions. The Jonker-Volgenant shortest augmenting path algorithm (Jonker and Volgenant, 1987) is employed to establish one-to-one correspondences based on the computed assignment probabilities of all objects and the probabilities of missing detections.

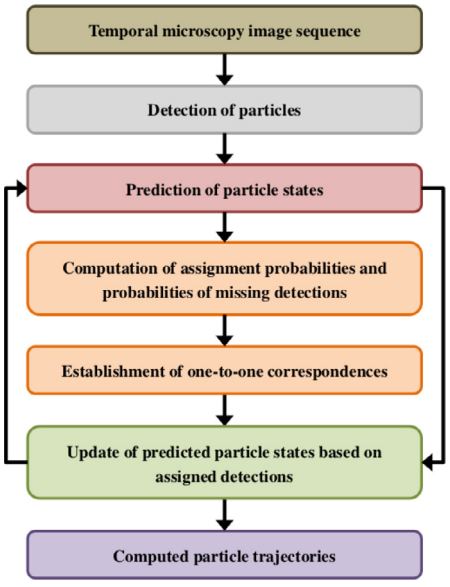


Fig. 1. Overview of the proposed Deep Probabilistic Particle Tracker (DPPT).

2.2. Bayesian filtering

We represent a biological particle i in a temporal microscopy image sequence at time point t by the state vector $\mathbf{x}_t^i \in \mathbb{R}^D$, which is reflected by the noisy measurement (detection) $\mathbf{y}_t^j \in \mathbb{R}^D$. In our settings, both the particle state $\mathbf{x}_t^i$ and the detection $\mathbf{y}_t^j$ are described by the image position at time point t, and therefore D equals the number of image dimensions in an experimental setup.

The goal of Bayesian filtering is to estimate $\mathbf{x}_t^i$ recursively over time based on a sequence of noisy detections $\mathbf{y}_{1:t}^j$. For each time point t, this estimation process comprises two consecutive steps: Prediction and update. Based on the posterior distribution $p(\mathbf{x}_{t-1}^i|\mathbf{y}_{1:t-1}^j)$ at time point $t-1$, the prediction step evaluates the particle dynamics using a dynamical model $p(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i)$ to determine the prior distribution $p(\mathbf{x}_t^i|\mathbf{y}_{1:t-1}^j)$ at time point t:

$$p(\mathbf{x}_t^i|\mathbf{y}_{1:t-1}^j) = \int p(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i)\, p(\mathbf{x}_{t-1}^i|\mathbf{y}_{1:t-1}^j)\mathrm{d}\mathbf{x}_{t-1}^i \tag{1}$$

In the update step, Bayes' rule is applied to compute the posterior distribution $p(\mathbf{x}_t^i|\mathbf{y}_{1:t}^j)$ at time point t from the prior distribution $p(\mathbf{x}_t^i|\mathbf{y}_{1:t-1}^j)$ by incorporating the detection $\mathbf{y}_t^j$ via a measurement model $p(\mathbf{y}_t^j|\mathbf{x}_t^i)$:

$$p(\mathbf{x}_t^i|\mathbf{y}_{1:t}^j) \propto p(\mathbf{y}_t^j|\mathbf{x}_t^i)\, p(\mathbf{x}_t^i|\mathbf{y}_{1:t-1}^j) \tag{2}$$

The state $\mathbf{x}_t^i$ can be determined from the posterior distribution $p(\mathbf{x}_t^i|\mathbf{y}_{1:t}^j)$. The two most common approaches for solving (1) and (2) are the Kalman filter and the particle filter. In contrast, we propose a deep learning approach to mimic classical Bayesian filtering.

2.3. Bayesian neural network for prediction and update

The proposed probabilistic neural network mimics classical Bayesian filtering. The network architecture can be subdivided into a state prediction and update block (see Fig. 2). The prediction block employs a GRU-based RNN (Cho et al., 2014) exploiting both short- and long-term temporal dependencies in the dynamics of an individual particle to estimate its next state. The update block corrects the predicted state based on the assigned detection. Bayesian layers (Kingma and Welling, 2014; Blundell et al., 2015) are used within the network to take into account *epistemic uncertainty*. In addition, the standard deviation for the predicted and updated state is computed to provide information about the *aleatoric uncertainty*. In contrast, previous work on object tracking did not con-

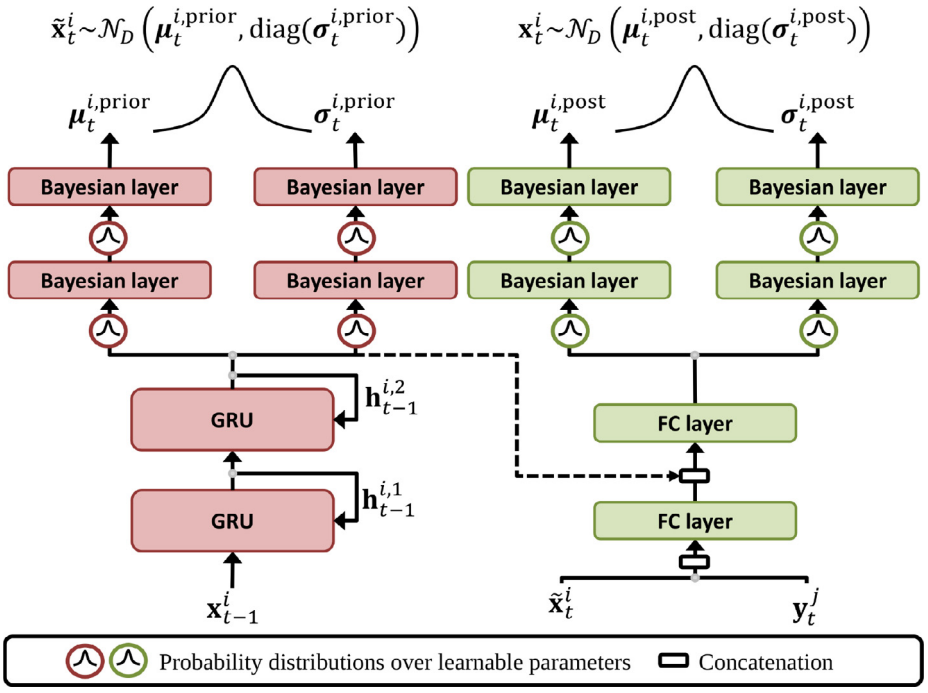


Fig. 2. Architecture of the proposed probabilistic neural network. Red indicates the network block for state prediction, and green the network block for state update. The dashed line indicates how the hidden state of the last GRU layer is exploited by the update block.

sider uncertainty (e.g., Milan et al., 2017; Sadeghian et al., 2017; Yao et al., 2018; Spilger et al., 2018; Yao et al., 2020; Spilger et al., 2020). For each time point $t-1$, the network computes two output vectors for particle i for the next time point t: The predicted state $\tilde{\mathbf{x}}_t^i \in \mathbb{R}^D$ and the updated state $\mathbf{x}_t^i \in \mathbb{R}^D$.

The prediction block of our network comprises L GRU layers (we used $L=2$) each consisting of K units. The structure of GRU allows capturing both short- and long-term temporal dependencies. To solve the vanishing and exploding gradient problems that occur in standard RNNs, GRU uses a hidden state and two gates regulating the information to be kept or discarded at each time point. The gating is designed similarly to that in the long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997), however, GRU is less complex than LSTM (e.g., it has only two gates instead of three in the LSTM) and is faster to compute. The reset gate of GRU determines which part of the previous hidden state is combined with the current input to compute a candidate state. The update gate of GRU determines which portion of the previous hidden state is preserved and which portion of the candidate state (derived from the reset gate) is added to the final hidden state. In more detail, the hidden state of layer l at time point t is represented by $\mathbf{h}_t^{i,l} \in \mathbb{R}^K$ (for $l=1,\dots,L$), while $\mathbf{h}_t^{i,0} \in \mathbb{R}^D$ denotes the network input vector. The output of the last GRU layer is denoted by $\mathbf{h}_t^{i,L}$. To predict the state of object i for the next time point t, the state vector $\mathbf{x}_{t-1}^i$ of the object at time point $t-1$ is used as input vector, i.e. $\mathbf{h}_t^{i,0} = \mathbf{x}_{t-1}^i$. For a particular GRU layer l and time point t, the update gate $\mathbf{z}_t^{i,l}$ and reset gate $\mathbf{r}_t^{i,l}$ are computed based on the previous hidden state $\mathbf{h}_{t-1}^{i,l}$ at time point $t-1$ and the hidden state $\mathbf{h}_t^{i,l-1}$ of the previous GRU layer:

$$\mathbf{z}_t^{i,l} = \sigma\left(\mathbf{W}_z^l \mathbf{h}_t^{i,l-1} + \mathbf{U}_z^l \mathbf{h}_{t-1}^{i,l} + \mathbf{b}_z^l\right) \tag{3}$$

$$\mathbf{r}_t^{i,l} = \sigma\left(\mathbf{W}_r^l \mathbf{h}_t^{i,l-1} + \mathbf{U}_r^l \mathbf{h}_{t-1}^{i,l} + \mathbf{b}_r^l\right) \tag{4}$$

where $\mathbf{W}_z^l$, $\mathbf{U}_z^l$, $\mathbf{b}_z^l$, $\mathbf{W}_r^l$, $\mathbf{U}_r^l$, and $\mathbf{b}_r^l$ represent the learnable parameters of the two gates. σ is the logistic sigmoid activation. Then, the new candidate state $\tilde{\mathbf{h}}_t^{i,l}$ is computed as follows:

$$\tilde{\mathbf{h}}_t^{i,l} = \tanh\left(\mathbf{W}_h^l \mathbf{h}_t^{i,l-1} + \mathbf{U}_h^l (\mathbf{r}_t^{i,l} \odot \mathbf{h}_{t-1}^{i,l}) + \mathbf{b}_h^l\right) \tag{5}$$

where $\mathbf{W}_h^l$, $\mathbf{U}_h^l$, and $\mathbf{b}_h^l$ are the learnable parameters. $\odot$ denotes the element-wise (Hadamard) multiplication and tanh is the hyperbolic tangent activation function. The previous hidden state $\mathbf{h}_{t-1}^{i,l}$ and the candidate state $\tilde{\mathbf{h}}_t^{i,l}$ are weighted by the update gate $\mathbf{z}_t^{i,l}$ to determine the new hidden state $\mathbf{h}_t^{i,l}$:

$$\mathbf{h}_t^{i,l} = \mathbf{z}_t^{i,l} \odot \mathbf{h}_{t-1}^{i,l} + (1 - \mathbf{z}_t^{i,l}) \odot \tilde{\mathbf{h}}_t^{i,l} \tag{6}$$

Finally, the hidden state $\mathbf{h}_t^{i,L}$ of the last GRU layer L is fed into two separate heads, each comprising two consecutive Bayesian layers with K and D Parametric Rectified Linear Units (PReLUs, He et al., 2015), respectively. We used PReLU since this activation function includes a learnable parameter (slope parameter for negative input values) to overcome shortcomings of the dying ReLU problem (inactive ReLU which outputs zero for any input value) and the inconsistent predictions of LeakyReLU for negative input values. The output vectors of the two heads represent the mean $\boldsymbol{\mu}_t^{i,\text{prior}} \in \mathbb{R}^D$ and standard deviation $\boldsymbol{\sigma}_t^{i,\text{prior}} \in \mathbb{R}^D$ of the prior probability distribution modeled as Gaussian distribution $\mathcal{N}_D\left(\boldsymbol{\mu}_t^{i,\text{prior}}, \text{diag}(\boldsymbol{\sigma}_t^{i,\text{prior}})\right)$. From this prior distribution the predicted state $\tilde{\mathbf{x}}_t^i$ is obtained:

$$\tilde{\mathbf{x}}_t^i \sim \mathcal{N}_D\left(\boldsymbol{\mu}_t^{i,\text{prior}}, \text{diag}(\boldsymbol{\sigma}_t^{i,\text{prior}})\right) \tag{7}$$

Therefore, the predicted state depends only on the current state $\mathbf{x}_{t-1}^i$ of the object and the hidden states $\mathbf{h}_{t-1}^{i,1:L}$ of the GRU layers.

Given the assigned detection $\mathbf{y}_t^j \in \mathbb{R}^D$ for the next time point t, the state is updated. First, the vectors $\mathbf{y}_t^j \in \mathbb{R}^D$ and $\tilde{\mathbf{x}}_t^i$ are concatenated, and passed to a FC layer with PReLUs mapping it to a vector of dimension K. Then this vector is concatenated with the hidden state $\mathbf{h}_t^{i,L}$ of the last GRU layer resulting in a vector of dimension $2K$ passed to another FC layer with K PReLUs. Finally, this K-dimensional vector is fed into two separate heads, each comprising two consecutive Bayesian layers with K and D

PReLUs, respectively. The output vectors of the two heads represent the mean $\boldsymbol{\mu}_t^{i,\text{post}} \in \mathbb{R}^D$ and standard deviation $\boldsymbol{\sigma}_t^{i,\text{post}} \in \mathbb{R}^D$ of the posterior probability distribution modeled as Gaussian distribution $\mathcal{N}_D\left(\boldsymbol{\mu}_t^{i,\text{post}}, \text{diag}(\boldsymbol{\sigma}_t^{i,\text{post}})\right)$. From this posterior distribution the updated state $\mathbf{x}_t^i$ is obtained:

$$\mathbf{x}_t^i \sim \mathcal{N}_D\left(\boldsymbol{\mu}_t^{i,\text{post}}, \text{diag}(\boldsymbol{\sigma}_t^{i,\text{post}})\right) \tag{8}$$

The computed standard deviations $\boldsymbol{\sigma}_t^{i,\text{prior}}$ and $\boldsymbol{\sigma}_t^{i,\text{post}}$ reflect the noise in the data (e.g., detector noise and motion noise) and are denoted as *aleatoric uncertainty*.

To take into account not only aleatoric uncertainty but also *epistemic uncertainty* (model uncertainty), we employ Bayesian layers in the two heads of the prediction and update block (four network heads in total), where learnable parameters are represented by probability distributions instead of single values. Since exact Bayesian inference is intractable, we employ a variational approximation. A differentiable estimator of the lower bound that can be optimized straightforwardly using a standard stochastic gradient approach is obtained by a reparameterization of the variational lower bound, also called evidence lower bound (ELBO) (Kingma and Welling, 2014; Blundell et al., 2015). This reparameterization strategy enables efficient learning of the neural network parameters, without requiring computationally expensive iterative inference schemes such as Markov chain Monte Carlo (e.g., Gu et al., 2015; Gong et al., 2019). In more detail, a variational posterior distribution $Q(\mathbf{W};\theta)$ parameterized by θ is used over the learnable parameters $\mathbf{W}$. The parameters θ of the variational posterior distribution $Q(\mathbf{W};\theta)$ representing the uncertainty of $\mathbf{W}$ are learned using variational inference. This is done by maximizing the evidence lower bound objective:

$$\begin{aligned}\text{ELBO}(\theta) = &-\int \mathrm{d}\mathbf{W} Q(\mathbf{W};\theta)\log P(\mathbf{Y}|\mathbf{X},\mathbf{W}) \\ &+\int \mathrm{d}\mathbf{W} Q(\mathbf{W};\theta)\log\frac{Q(\mathbf{W};\theta)}{P(\mathbf{W})}\end{aligned} \tag{9}$$

where $P(\mathbf{W})$ is the prior distribution over the weights that represents the uncertainty in the weights before network training. The Bayesian layer performs a regularization and imposes the constraint that the posterior distribution $Q(\mathbf{W};\theta)$ is close to the prior distribution $P(\mathbf{W})$. For $Q(\mathbf{W};\theta)$ and $P(\mathbf{W})$, we use a multivariate normal distribution. $P(\mathbf{Y}|\mathbf{X},\mathbf{W})$ is the likelihood function that specifies the variation in the labels $\mathbf{Y}$ given the network inputs $\mathbf{X}$ and the learnable parameters $\mathbf{W}$. While the second term (Kullback-Leibler divergence of $Q(\mathbf{W};\theta)$ regarding $P(\mathbf{W})$) is used for regularization and determined analytically, the first term is used to compute labels from the inputs and is approximated by drawing a single random set of weights from $Q(\mathbf{W};\theta)$. The sampling for computing the first term yields an ensemble of different network outputs. Since always a new set of weights is sampled according to $Q(\mathbf{W};\theta)$, we obtain a different prior distribution $\mathcal{N}_D\left(\boldsymbol{\mu}_t^{i,\text{prior}}, \text{diag}(\boldsymbol{\sigma}_t^{i,\text{prior}})\right)$ and posterior distribution $\mathcal{N}_D\left(\boldsymbol{\mu}_t^{i,\text{post}}, \text{diag}(\boldsymbol{\sigma}_t^{i,\text{post}})\right)$ for each time the four network heads are applied. Thus, the diversity in the estimated prior and posterior distributions reflects the uncertainty in the weights (model uncertainty). In our sampling strategy, for each time point t and particle i, the four network heads are applied N times yielding N prior and N posterior distributions. We used $N = 20$, which is a good compromise between computation time and tracking performance. From each of these prior and posterior distributions a predicted state and updated state is obtained, respectively. The final predicted state $\tilde{\mathbf{x}}_t^i$ and updated state $\mathbf{x}_t^i$ are determined by averaging over all predicted and updated states, respectively.

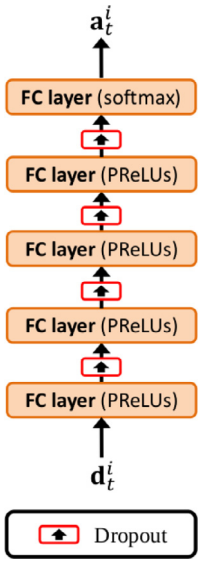


Fig. 3. Architecture of the proposed neural network for correspondence finding.

The parameters of our probabilistic network are learned by minimizing the loss function:

$$\mathcal{L}_{\mathbf{x},t}^i = -\left(\underbrace{\log P\left(\hat{\mathbf{x}}_t^i|\boldsymbol{\mu}_t^{i,\text{prior}},\boldsymbol{\sigma}_t^{i,\text{prior}}\right)}_{\text{state prediction}} + \underbrace{\lambda\log P\left(\hat{\mathbf{x}}_t^i|\boldsymbol{\mu}_t^{i,\text{post}},\boldsymbol{\sigma}_t^{i,\text{post}}\right)}_{\text{state update}}\right) \tag{10}$$

where $\hat{\mathbf{x}}_t^i$ is the true state of particle i at time point t. The loss function consists of two terms: The negative log-likelihood loss for state prediction and state update. The loss function represents the loss for one training sample corresponding to a single time point t of an individual particle i. We used $\lambda = 1$ in all our experiments. For training, the prediction and update block are employed as a unified network. For performing tracking, the two blocks are used sequentially.

2.4. Correspondence finding

For correspondence finding, we propose a deep neural network which computes assignment probabilities and probabilities of missing detections. The network consists of four consecutive FC layers, each with R PReLUs (we used $R = 512$), followed by a fully connected linear output layer with softmax normalization. For each FC layer except the output layer, we employed dropout with a rate of 0.4 during training to avoid overfitting (Srivastava et al., 2014). The network architecture is sketched in Fig. 3.

For each particle i and time point $t-1$, the network takes as input the vector $\mathbf{d}_t^i \in \mathbb{R}^M$, whose components are defined by $d_t^{i,j} = \|\tilde{\mathbf{x}}_t^i - \mathbf{y}_t^j\|_2$ as the Euclidean distance between the predicted state of particle i and detection j at time point t. M is the number of detections within a gate of radius r_g (we used $r_g = 20$ pixel) around the predicted position of particle i at time point t. Since the number of detections within the gate varies and our network requires a fixed input size, we consider at most the M-nearest detections (we used $M = 4$) within the gate. If there are less than M detections within the gate, we pad the vector $d_t^{i,j}$ with a placeholder (we used '−1'). The final output vector $\mathbf{a}_t^i \in [0,1]^{M+1}$ of the network contains the normalized assignment probabilities for time point t, i.e. $\sum_{j=0}^{M} a_t^{i,j} = 1$, where $a_t^{i,0}$ denotes the probability of a missing detection and $a_t^{i,j}$ represents the assignment probability between particle i and detection j (for $j = 1,\ldots,M$). The computed

probabilities adapt to the local neighborhood, for example, in regions with high object density the probability of a missing detection is lower. We employ the Jonker-Volgenant shortest augmenting path algorithm (Jonker and Volgenant, 1987) using the computed assignment probabilities and probabilities of missing detections as input to establish one-to-one correspondences between all tracked particles and the set of detections obtained at time point t. Since the network computes probabilities, a threshold to determine missing detections is not needed.

To measure the deviation between the computed assignment probabilities $\mathbf{a}_t^i$ and ground truth $\tilde{\mathbf{a}}_t^i$, we use a multi-class variant of the α-balanced focal loss described in Lin et al. (2020). The loss $\mathcal{L}_{\mathbf{a},t}^i$ for one training sample (one time point t of an individual particle i) is given as:

$$\mathcal{L}_{\mathbf{a},t}^i = -\sum_{j=0}^{M} \alpha_j (1 - a_t^{i,j})^\gamma \cdot \tilde{a}_t^{i,j} \log(a_t^{i,j}) \tag{11}$$

where γ is a focusing parameter down-weighting easy samples and emphasizing hard samples. In all our experiments we used $\gamma = 2$. $\boldsymbol{\alpha} \in \mathbb{R}^{M+1}$ represent weighting factors that address class imbalance in the training data and are calculated based on the class distribution in the training data.

2.5. Network training

Training neural networks typically requires a vast amount of data to achieve convergence without overfitting. Since ground truth of particles in real fluorescence microscopy images is hardly available and accurate manual annotation is very tedious, our DPPT network is trained using simulated data only. In our data simulator, particle trajectories were simulated based on different motion models and the image data was generated using a Poisson noise model and automatically extracted information from the real images. For the particle dynamics we used four different motion types, namely directed motion, Brownian motion, random switching between directed motion and Brownian motion, and accelerated motion. Motion model parameters (e.g., diffusion coefficient, velocity, acceleration) of individual particles were drawn from uniform distributions. For the diffusion coefficient, we used a uniform distribution in the interval $[1, 6]$ both for the Particle Tracking Challenge data and the live cell fluorescence microscopy images. For the velocity we used an interval of $[1, 6]$ (directed motion) and $[1, 4]$ (accelerated motion and switching motion), and for the acceleration we employed $[0.2, 0.8]$. We used relatively large intervals to increase network generalization. Initial particle positions and particle appearance as well as disappearance are governed by random processes. Movement out of the field of view and out of focus was also simulated. The image noise is reflected by the SNR $= (I_{max} - I_{bg})/\sqrt{I_{max}}$ (Sbalzarini and Koumoutsakos, 2005), where I_{max} is the maximum intensity of the particle and I_{bg} denotes the background intensity. Different to our previous work (Spilger et al., 2018; 2020), we use automatically extracted information from the real images to generate training images that well represent the real data. In our scheme, we detect particles in the real images by the spot-enhancing filter (Sage et al., 2005), and automatically determine the SNR, the particle size, and the particle density based on the detected positions and local neighborhood information. We use Gaussian fitting at the detected positions to determine I_{max} and I_{bg} to compute the SNR as well as to determine the particle size σ. For I_{max}, I_{bg}, and σ, we computed the mean and standard deviation over all detected particles. The SNR of the training data was set according to the determined SNR in the real data. The appearance parameters of individual particles (I_{max}, σ) in the training data were sampled from Gaussian distributions with mean values and standard deviations determined in the real data. The particle density in the training data was set according to the determined average number of particle detections per frame in the real images. Instead, in our previous work, these parameters were drawn from uniform distributions with fixed manually defined intervals. Image size, stack size, and bit depth were set according to the meta-information of the real data. Main differences to the Particle Tracking Challenge simulator (Chenouard et al., 2014; de Chaumont et al., 2012) are that our simulator includes out-of-focus movement and accelerated motion as well as uses automatically extracted information from the real images to generate training images that well represent the real data.

In addition, we exploit the computed uncertainty of our network to select the generated training data set with the best-suited motion model so that the training data well represents the real images. This was done by training our network with different motion models and then using the tracking results with the lowest epistemic uncertainty (see Section 3.2.2 below). Thus, we take advantage of the fact that the epistemic uncertainty of the network is lower when the particle motion in the training data agrees well with the motion in the real data. This strategy for automated motion model selection is novel and has not been used in previous work.

Moreover, to generate training data we use automatic detections in the synthetically generated images to enable the network to learn the detector errors. Particle detection was performed using the spot-enhancing filter (SEF) (Sage et al., 2005) and Gaussian fitting. Then, the resulting detections were mapped to the ground truth trajectories using a nearest neighbor search with a validation gate of 5 pixel. The trajectories and the detections mapped on them were used for network training. Thus, our approach does not require prior knowledge about detection errors but learns this information from the image data. This training strategy allows generating synthetic training data that well represents the real data.

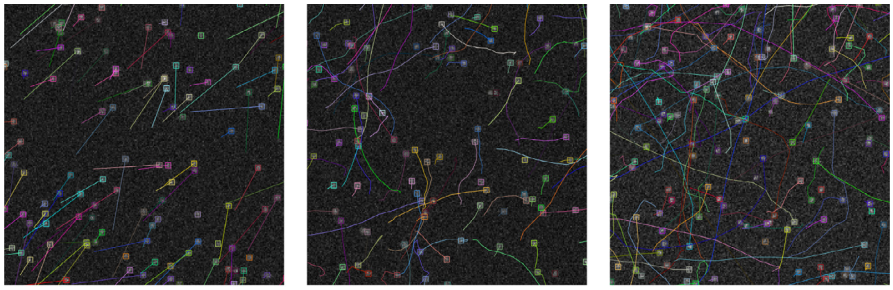
For network training, we employed the AMSGrad optimizer (Reddi et al., 2018) with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We used a mini-batch size of 64 and an initial learning rate of $l_{init} = 0.02$ for the Bayesian neural network and $l_{init} = 0.001$ for the network computing assignment probabilities. To avoid overfitting, early stopping was performed after convergence was reached. For training and validation we used about 102.400 training samples from synthetic image sequences. A training sample includes one time point for an individual particle. We split the data set into 80% for training and 20% for validation. The image dimensions are normalized to the range $[0, 1]$. Our model is implemented in Python 3.7 using Tensorflow 2.1.0 (Abadi et al., 2016) and TensorFlow Probability 0.9.0 (Dillon et al., 2017). We used a laptop with Intel(R) Core(TM) i7-7700HQ CPU, NVIDIA GeForce GTX 1050 Ti GPU, and a Linux operating system.

3. Experimental results

In this section, we present experimental results of our Deep Probabilistic Particle Tracker (DPPT). First, we describe the metrics for quantitative performance evaluation. Then, we verify that DPPT captures both aleatoric and epistemic uncertainty. A performance evaluation is carried out using 2D as well 3D data of the Particle Tracking Challenge. In addition, we study the impact of using the uncertainty information for subsequent motion analysis. We also evaluate DPPT based on 2D and 3D real live cell fluorescence microscopy images.

3.1. Performance metrics

To quantitatively assess and objectively compare the tracking performance, we used the metrics α, β, *JSC*, JSC_θ, and *RMSE* as



(a) $\sigma_m = 0$ pixel (b) $\sigma_m = 2$ pixel (c) $\sigma_m = 4$ pixel

Fig. 4. Example image sections (225×225 pixels) from generated synthetic image sequences displaying particles performing directed motion with different levels of motion noise σ_m (time point $t = 12$). The image contrast was enhanced for better visibility.

described in Chenouard et al. (2014). The metric $\alpha \in [0, 1]$ indicates the overall degree of matching of ground truth and computed tracks excluding spurious tracks. $\beta \in [0, \alpha]$ includes an additional penalization for spurious tracks compared to α ($\alpha = \beta$ when there are no spurious tracks). The Jaccard similarity coefficient $JSC \in [0, 1]$ represents the rate of correctly determined track points, and $JSC_\theta \in [0, 1]$ is the Jaccard similarity coefficient for entire tracks instead of single track points. The overall localization accuracy of correctly determined track points is indicated by the root mean square error (*RMSE*). For all metrics except *RMSE*, higher values indicate a better tracking performance.

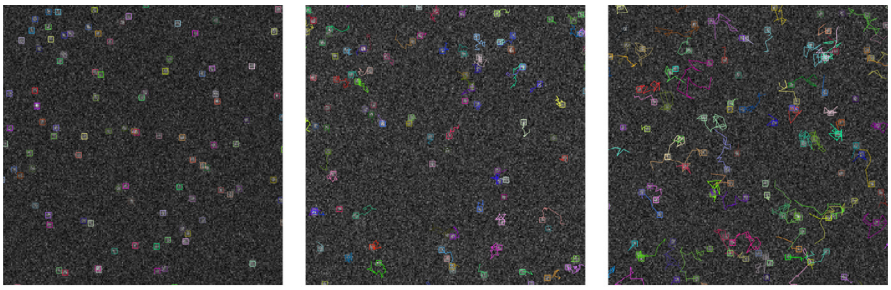
3.2. Evaluation and analysis of aleatoric and epistemic uncertainty

First, we evaluated uncertainty estimation by our network and verified whether both aleatoric and epistemic uncertainty are captured. To this end, we trained and tested DPPT based on generated 2D synthetic image sequences with varying conditions. The synthetic images (512 × 512, 8-bit) simulate real fluorescence microscopy images displaying multiple particles. The particles have an isotropic Gaussian intensity structure with Gaussian distributed standard deviation (mean $\sigma_{x,y} = 1.8$ pixel) and the images are distorted by additive Poisson noise (we used SNR = 4). For directed motion, the particle position at the next time point was determined by the current position plus the velocity vector (given by the velocity from the previous to the current time point) and an additional random change (Gaussian distribution) of both the position and the velocity vector. The changes in the position and the velocity vector were determined from a multivariate normal distribution with covariance matrix $Q = q((\sigma_{11}^2, \sigma_{12}^2), (\sigma_{12}^2, \sigma_{22}^2))$ for each image dimension as in Chenouard et al. (2014). σ_{11}^2 denotes the variance of the position noise, σ_{12}^2 the covariance of position and velocity noise, σ_{22}^2 the variance of the velocity noise, and q the motion noise influence factor. For all data we used $\sigma_{11}^2 = (1/3)$ frame^3, $\sigma_{12}^2 = (1/2)$ frame^2, and $\sigma_{22}^2 = 1$ frame. The motion noise of directed motion can be defined by the standard deviation $\sigma_m = \sqrt{q(\sigma_{11}^2 + 2\sigma_{12}^2 + \sigma_{22}^2)}$, which represents all elements of the covariance matrix Q and follows from the general formula for the variance of the sum of two random variables describing directed motion ($Var(aX + bY) = a^2Var(X) + 2abCov(X, Y) + b^2Var(Y)$). For Brownian motion (random walk), the next particle position was determined by sampling from a Gaussian distribution centered at the current position and with standard deviation σ_m. For each studied condition we generated three synthetic image sequences, two of which were used for network training (training data) and one for testing (test data). Each temporal image sequence comprises 100 time points.

3.2.1. Evaluation of aleatoric and epistemic uncertainty

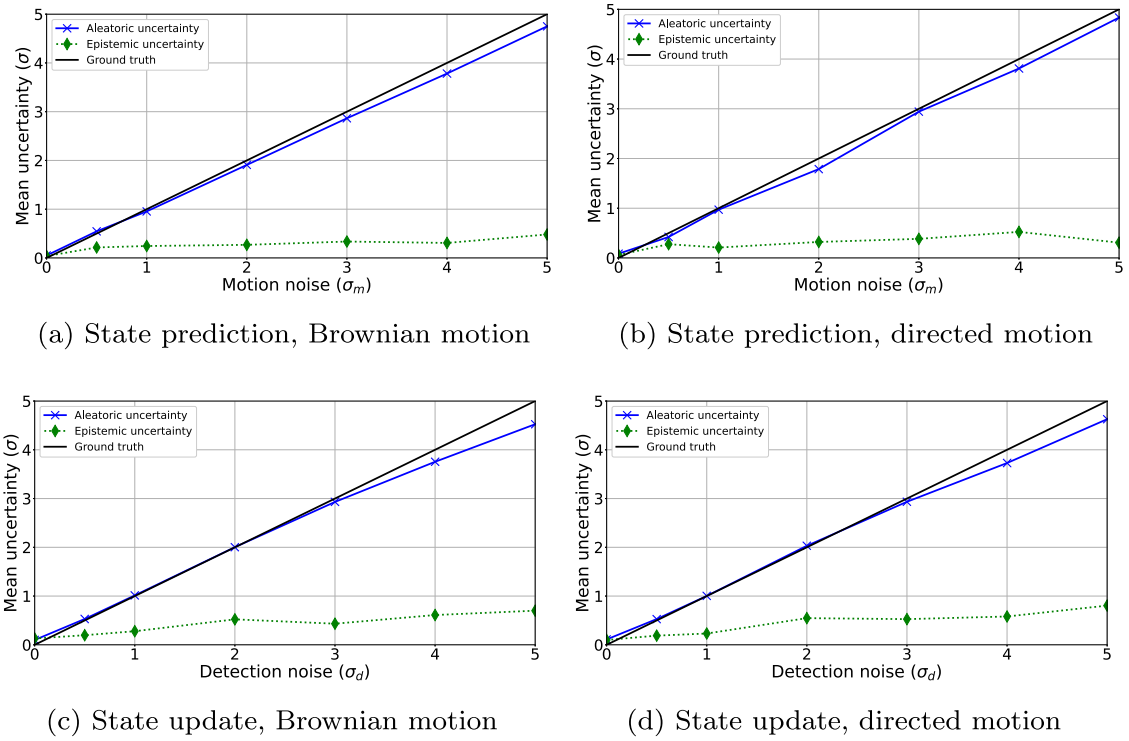
To evaluate uncertainty estimation by DPPT, we established ground truth image data which is used as benchmark. We first considered uncertainty estimation of the *state prediction* which captures the *motion noise*. We generated synthetic image sequences with seven different levels of motion noise σ_m ($\sigma_m = 0$, 0.5, 1, 2, 3, 4, and 5 pixel) for both Brownian motion and directed motion, and evaluated how well the motion noise is estimated by our network. In this experiment we did not consider detection noise (but in the following experiment). Example image sections with ground truth particle trajectories are shown in Fig. 4 for directed motion and in Fig. 5 for Brownian motion. It can be seen that the random fluctuation in the particle position (Brownian motion) and the deviation from a straight path (directed motion) increase with the strength of the motion noise σ_m. We applied DPPT to the generated image sequences and evaluated uncertainty estimation. We considered aleatoric uncertainty (σ_{alea}) and epistemic uncertainty (σ_{epi}). In Fig. 6 (a) and (b), ground truth and mean values of computed aleatoric and epistemic uncertainty of state prediction (over all trajectories) by DPPT are shown as a function of σ_m. It can be seen that the epistemic uncertainty is much smaller than the aleatoric uncertainty, which is expected for a well-trained network. For the image sequences of both motion models the computed aleatoric uncertainty agrees already relatively well with the ground truth. The *RMSE* between the aleatoric uncertainty and the ground truth is 0.143 pixel for Brownian motion and 0.134 pixel for directed motion. The result is further improved when considering the computed *combined* uncertainty comprising aleatoric and epistemic uncertainty (defined as the square root of the sum of the variances σ_{alea}^2 and σ_{epi}^2), which yields a lower *RMSE* of 0.133 pixel for Brownian motion and 0.116 pixel for directed motion. This shows that both types of uncertainty (aleatoric and epistemic uncertainty) should be taken into account for motion noise estimation. The experiment demonstrates that the motion noise is well captured by our network.

In addition, we evaluated uncertainty estimation of the *state update* by DPPT. We generated synthetic image sequences using two motion models (Brownian motion, directed motion), simulated detections with different levels of additive white Gaussian noise ($\sigma_d = 0$, 0.5, 1, 2, 3, 4, and 5 pixel), and evaluated how well the *detection noise* is estimated by our network. For the motion noise we used $\sigma_m = 10$ pixel. Since the detection noise is smaller than the motion noise (i.e. the detection noise has a higher reliability) and since we considered only track points with an assigned detection, the state update mainly takes into account the detection information and the computed uncertainty represents the detection noise. In Fig. 6 (c) and (d), ground truth and computed mean aleatoric and epistemic uncertainty of state update (over all trajectories) by our network are shown as a function of the detection noise level. Also in this experiment the epistemic uncertainty is much smaller than the aleatoric uncertainty as expected for a well-trained network. For both motion models the computed aleatoric uncertainty agrees already relatively well with the ground truth. The *RMSE* between the aleatoric uncertainty and the ground truth is 0.208 pixel



(a) $\sigma_m = 0$ pixel (b) $\sigma_m = 2$ pixel (c) $\sigma_m = 4$ pixel

Fig. 5. Example image sections (225×225 pixels) from generated synthetic image sequences displaying particles performing Brownian motion with different levels of motion noise σ_m (time point $t = 20$). The image contrast was enhanced for better visibility.



(a) State prediction, Brownian motion (b) State prediction, directed motion

(c) State update, Brownian motion (d) State update, directed motion

Fig. 6. Computed mean aleatoric and epistemic uncertainty of state prediction as a function of the motion noise level for (a) Brownian motion and (b) directed motion, and of state update as a function of the detection noise level for (c) Brownian motion and (d) directed motion. Black lines indicate the ground truth.

for Brownian motion and 0.181 pixel for directed motion. The result is further improved when considering the computed combined uncertainty yielding a lower *RMSE* of 0.191 pixel for Brownian motion and 0.161 pixel for directed motion. This shows that both types of uncertainty (aleatoric and epistemic uncertainty) should be taken into account for detection noise estimation. The experiment demonstrates that the detection noise is well captured by our network.

3.2.2. Further analysis of epistemic uncertainty and exploitation to assess the suitability of the training data

To further analyze the *epistemic* uncertainty, we trained DPPT with different numbers of training samples (ranging from 6.4×10^3 to 102.4×10^3) using synthetic image data with directed motion and SNR = 4. One training sample represents one time point of a particle. The mean epistemic uncertainty over all particles and image dimensions as a function of the number of training samples is displayed in Fig. 7. As expected, the mean epistemic uncertainty decreases with the number of training samples, which demonstrates that the epistemic uncertainty is captured by our network.

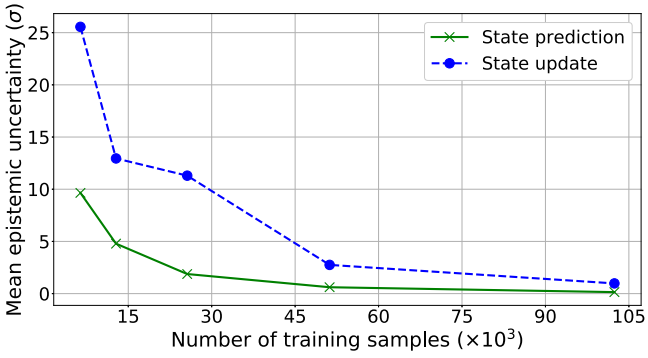
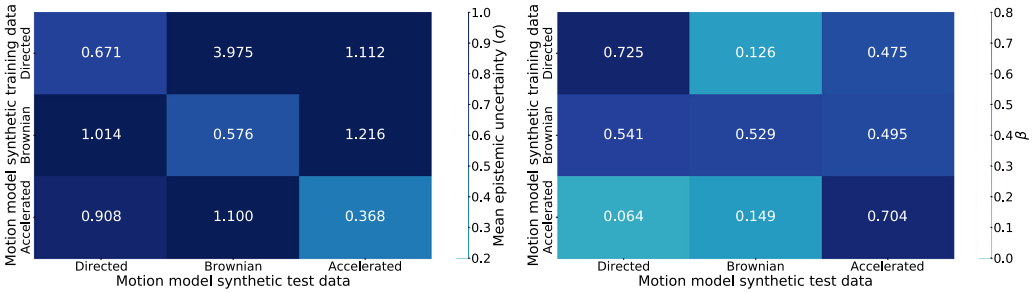


Fig. 7. Mean epistemic uncertainty of state prediction and state update of DPPT as a function of the number of training samples.

We also generated image data with different particle dynamics (SNR = 4). We used three different motion models, namely directed motion, Brownian motion, and accelerated motion. We



(a) Epistemic uncertainty (b) Tracking performance

Fig. 8. (a) Mean epistemic uncertainty and (b) tracking performance of DPPT for different motion models.

trained DPPT on the training data of one of the three motion models and then applied it to the test data of all three motion models. We did this for all three models resulting in nine training and test data combinations. Fig. 8(a) shows the mean epistemic uncertainty as heatmap for the nine combinations. It can be seen that the mean epistemic uncertainty is lowest when the same motion model is used for the training and test data, and otherwise the mean epistemic uncertainty is higher. This verifies that the epistemic uncertainty is captured by our network, and that the epistemic uncertainty can be exploited to assess the suitability of the training data for a considered test data. In addition, Fig. 8(b) shows the tracking performance in terms of β as heatmap for the different motion models. It can be seen that selecting the training data set with the correct motion model improves the tracking performance.

3.3. Particle Tracking Challenge Data

We evaluated our DPPT approach based on both 2D as well as 3D image data from the Particle Tracking Challenge (Chenouard et al., 2014) and compared the tracking performance with the overall top-three methods of the Challenge (Methods 5, 1, and 2). Method 5 localizes particles based on the spot-enhancing filter (SEF) (Sage et al., 2005) and establishes correspondences by probabilistic data association (Godinez and Rohr, 2015). Method 1 employs iterative intensity-weighted centroid calculation for localization and combinatorial optimization via greedy hill-climbing for correspondence finding (Sbalzarini and Koumoutsakos, 2005). Method 2 localizes particles using adaptive local maxima selection and relies on multiple hypothesis tracking for correspondence finding (Coraluppi and Carthel, 2011). We also performed a comparison with our previous deep learning approaches Deep Particle Tracker (DPT) (Spilger et al., 2018) and Deep Particle Hypotheses Tracker (DPHT) (Spilger et al., 2020). DPT employs an LSTM-based RNN for state prediction and correspondence finding. This approach was developed for 2D data and was not applied to the 3D data. DPHT is based on a bidirectional LSTM-based RNN that exploits multiple track hypotheses for correspondence finding. Note that DPT and DPHT use deep learning for correspondence finding, while DPPT uses deep learning also for state prediction and update (as in classical Bayesian filtering) as well as for taking into account aleatoric and epistemic uncertainty. For DPPT, DPT, and DPHT, we used the same set of detections. The networks were trained based on own generated synthetic data, and we did not use the data of the Particle Tracking Challenge for network training.

We assessed the performance for different object dynamics using temporal image sequences from the 2D vesicle and 3D virus scenario of the Particle Tracking Challenge with medium (~500 particles/frame) and high (~1000 particles/frame) particle densities, and all SNR levels (SNR = 1, 2, 4, and 7). The data set (comprising 65.462 trajectories) is challenging due to complex motion in dense environments as well as image noise causing clutter and numerous detection errors. The 2D images of the vesicle scenario (512×512 pixels, 8-bit) display round particles performing Brownian (random walk) motion. The 3D image data of the virus scenario ($512 \times 512 \times 10$ voxels, 8-bit) shows spherical particles switching between Brownian and directed motion. For both scenarios, each image sequence consists of 100 images. Random processes define appearance and disappearance of individual particles compensating each other on average. The computation time for one image sequence of the vesicle scenario with high object density and SNR = 2 was about 203 seconds, using a laptop as specified above (end of Section 2.5).

Table 1
Quantitative tracking performance of different approaches for data of the vesicle scenario with SNR = 1 from the Particle Tracking Challenge. The best performance values are highlighted bold and underlined, and the second best performance values are bold.

Metric	α	β	*JSC*	JSC_θ	*RMSE*
Medium density					
Method 5	0.162	**0.142**	**0.225**	**0.458**	2.172
Method 1	0.027	0.026	0.034	0.300	**1.533**
Method 2	**0.198**	0.111	0.192	0.335	2.386
DPT	0.122	0.071	0.106	0.198	**1.701**
DPHT	0.128	0.100	0.155	0.380	1.858
DPPT	**0.172**	**0.139**	**0.223**	**0.407**	2.164
High density					
Method 5	0.136	**0.120**	**0.198**	**0.460**	2.296
Method 1	0.091	0.064	0.089	0.231	**1.859**
Method 2	**0.163**	0.080	0.147	0.324	2.531
DPT	0.059	0.056	0.076	0.443	**1.654**
DPHT	0.121	0.104	0.158	**0.444**	1.984
DPPT	**0.158**	**0.123**	**0.189**	0.391	2.055

The quantitative results are presented in Tables 1–8. The best performance values are highlighted in bold and underlined, and bold indicates the second best performances. It can be seen that DPPT yields best or second best tracking results for almost all cases and SNR levels of the vesicle scenario. For the virus scenario, DPPT performs best or second best for the lowest (SNR = 1) and highest SNR level (SNR = 7). Overall for both scenarios, DPPT outperforms for β the other methods in eight out of 16 cases, and is second best in three cases. DPPT performs best in terms of α, *JSC*, JSC_θ, and *RMSE* in six, seven, five, and two cases, respectively, and yields the second best result for α, *JSC*, and *RMSE* in four cases, and for JSC_θ in two cases. For the vesicle scenario, DPPT is somewhat better for SNR = 2 compared to SNR = 1, while for the virus scenario it is vice versa. However, the performance values of our method compared to the best method in these cases are partially relatively similar. On the other hand, the results for the vesicle and virus data are not directly comparable since the data characteris-

Table 2
Quantitative tracking performance of different approaches for data of the vesicle scenario with SNR = 2 from the Particle Tracking Challenge. The best performance values are highlighted bold and underlined, and the second best performance values are bold.

Metric	α	β	JSC	JSC_θ	$RMSE$
Medium density					
Method 5	0.448	0.391	0.489	0.664	1.325
Method 1	0.398	0.298	0.340	0.411	**0.840**
Method 2	0.517	0.417	0.510	0.629	1.254
DPT	0.450	0.356	0.413	0.577	**0.795**
DPHT	**0.520**	**0.448**	**0.526**	**0.680**	0.874
DPPT	**0.562**	**0.485**	**0.564**	**0.700**	1.014
High density					
Method 5	0.353	0.295	**0.382**	**0.607**	1.484
Method 1	0.294	0.217	0.256	0.379	1.088
Method 2	0.356	0.249	0.331	0.515	1.582
DPT	0.372	0.293	0.353	0.536	**1.025**
DPHT	**0.383**	**0.311**	0.376	0.580	**1.044**
DPPT	**0.421**	**0.341**	**0.416**	**0.601**	1.232

Table 3
Quantitative tracking performance of different approaches for data of the vesicle scenario with SNR = 4 from the Particle Tracking Challenge. The best performance values are highlighted bold and underlined, and the second best performance values are bold.

Metric	α	β	JSC	JSC_θ	$RMSE$
Medium density					
Method 5	0.658	0.588	0.641	0.776	0.754
Method 1	0.687	0.609	0.652	0.767	0.607
Method 2	0.582	0.514	0.590	0.757	0.970
DPT	**0.695**	0.624	0.658	0.790	**0.545**
DPHT	**0.697**	**0.638**	**0.671**	**0.804**	**0.567**
DPPT	0.686	**0.630**	**0.669**	**0.813**	0.641
High density					
Method 5	0.488	0.408	0.466	0.671	1.004
Method 1	0.531	0.442	0.487	0.641	0.801
Method 2	0.430	0.356	0.429	0.649	1.208
DPT	**0.547**	**0.462**	**0.505**	0.680	**0.746**
DPHT	0.531	0.458	0.500	**0.685**	**0.751**
DPPT	**0.543**	**0.464**	**0.510**	**0.690**	0.867

Table 4
Quantitative tracking performance of different approaches for data of the vesicle scenario with SNR = 7 from the Particle Tracking Challenge. The best performance values are highlighted bold and underlined, and the second best performance values are bold.

Metric	α	β	JSC	JSC_θ	$RMSE$
Medium density					
Method 5	0.677	0.605	0.646	0.783	0.667
Method 1	0.700	0.619	0.650	0.758	0.544
Method 2	0.611	0.547	0.606	0.775	0.828
DPT	**0.711**	0.631	0.651	**0.790**	**0.525**
DPHT	0.705	**0.641**	**0.661**	**0.820**	**0.539**
DPPT	**0.708**	**0.645**	**0.673**	**0.820**	0.638
High density					
Method 5	0.533	0.453	0.503	0.698	0.931
Method 1	**0.582**	0.494	0.526	0.683	**0.683**
Method 2	0.466	0.395	0.458	0.665	1.027
DPT	**0.590**	**0.507**	**0.535**	0.702	**0.677**
DPHT	0.573	**0.506**	0.534	**0.717**	0.703
DPPT	0.577	0.503	**0.541**	**0.722**	0.827

tics are very different (wide-field vs. confocal microscopy, Brownian vs. switching Brownian/directed motion, 2D vs. 3D data). It seems that there is no systematic dependency of the results of DPPT on the SNR level compared to previous methods. Sample trajectories obtained by DPPT are shown in Fig. 9 for an image section (215 × 215 pixels) of the vesicle scenario with medium density

Table 5
Quantitative tracking performance of different approaches for data of the virus scenario with SNR = 1 from the Particle Tracking Challenge. The best performance values are highlighted bold and underlined, and the second best performance values are bold.

Metric	α	β	JSC	JSC_θ	$RMSE$
Medium density					
Method 5	0.086	0.082	0.117	**0.341**	1.788
Method 1	0.088	0.066	0.086	0.221	**1.525**
Method 2	0.057	0.018	0.034	0.155	2.454
DPHT	**0.114**	**0.101**	**0.143**	**0.412**	**1.551**
DPPT	**0.150**	**0.121**	**0.174**	0.314	1.718
High density					
Method 5	0.122	0.115	0.164	**0.438**	1.768
Method 1	0.140	0.099	0.130	0.253	**1.560**
Method 2	0.105	0.046	0.081	0.261	2.305
DPHT	**0.189**	**0.154**	**0.221**	**0.459**	**1.536**
DPPT	**0.223**	**0.173**	**0.245**	0.390	1.651

Table 6
Quantitative tracking performance of different approaches for data of the virus scenario with SNR = 2 from the Particle Tracking Challenge. The best performance values are highlighted bold and underlined, and the second best performance values are bold.

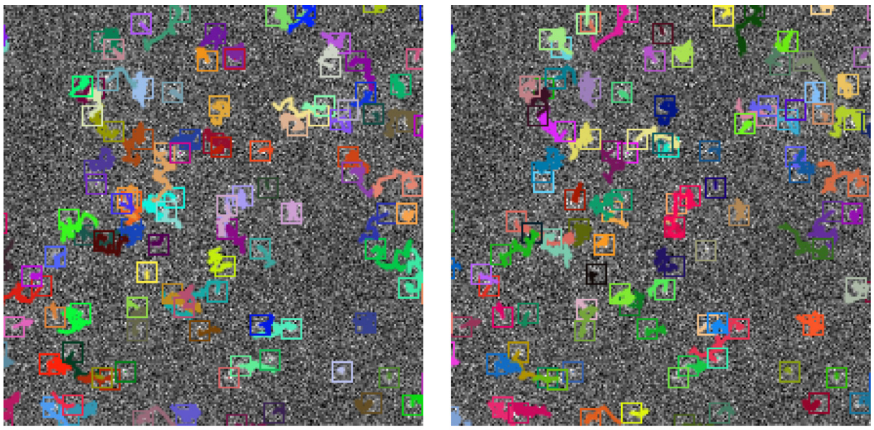
Metric	α	β	JSC	JSC_θ	$RMSE$
Medium density					
Method 5	**0.646**	**0.590**	**0.716**	**0.802**	1.085
Method 1	0.581	0.517	0.595	0.641	**0.776**
Method 2	**0.655**	**0.581**	**0.714**	0.742	1.062
DPHT	0.631	0.567	0.670	**0.777**	0.891
DPPT	0.642	0.546	0.633	0.702	**0.855**
High density					
Method 5	**0.553**	**0.495**	**0.606**	**0.729**	1.101
Method 1	0.528	0.460	0.539	0.610	**0.870**
Method 2	**0.576**	**0.486**	**0.611**	**0.682**	1.140
DPHT	0.542	0.472	0.555	0.680	0.880
DPPT	0.547	0.469	0.548	0.645	**0.877**

Table 7
Quantitative tracking performance of different approaches for data of the virus scenario with SNR = 4 from the Particle Tracking Challenge. The best performance values are highlighted bold and underlined, and the second best performance values are bold.

Metric	α	β	JSC	JSC_θ	$RMSE$
Medium density					
Method 5	**0.776**	**0.748**	**0.854**	**0.891**	0.754
Method 1	0.748	0.712	0.818	0.869	0.875
Method 2	**0.769**	**0.725**	**0.826**	**0.880**	0.724
DPHT	0.739	0.686	0.752	0.829	**0.576**
DPPT	0.767	0.695	0.757	0.814	**0.534**
High density					
Method 5	0.670	**0.619**	**0.712**	**0.805**	0.796
Method 1	0.642	0.582	0.677	0.776	0.936
Method 2	**0.699**	**0.646**	**0.742**	**0.833**	0.792
DPHT	**0.672**	0.607	0.667	0.766	**0.587**
DPPT	0.670	0.582	0.641	0.727	**0.606**

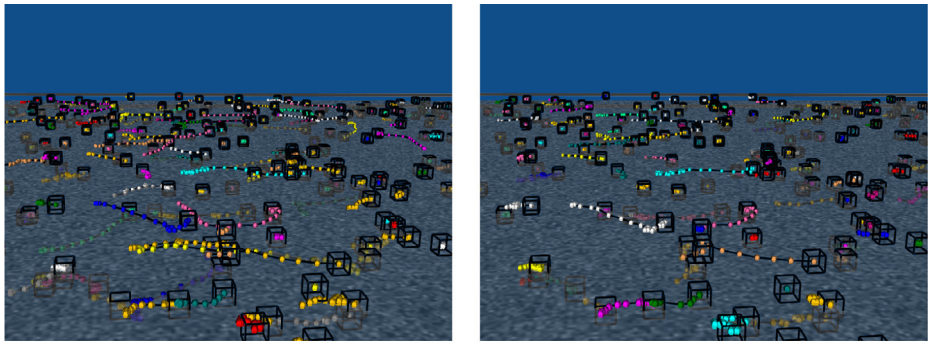
and SNR = 2. Fig. 10 shows sample tracking results of DPPT for the virus scenario with medium density and SNR = 2. In both cases the computed trajectories match well the ground truth despite the relatively low SNR level.

To demonstrate that DPPT copes well with image data that has somewhat different characteristics than the training data, we have applied our approach to data of the vesicle scenario with SNR = 2 and medium object density while it was trained with SNR = 4 and high object density. The obtained performance values ($\alpha = 0.547$, $\beta = 0.474$, $JSC = 0.557$, $JSC_\theta = 0.683$, $RMSE = 1.083$) differ only slightly from the results in Table 2 ($\alpha = 0.562$, $\beta = 0.485$,



(a) Ground truth, vesicle scenario (b) DPPT, vesicle scenario

Fig. 9. Ground truth and tracking results of DPPT for an image section (215×215 pixels) of the vesicle scenario with medium density and SNR = 2. The image contrast was enhanced for better visibility.



(a) Ground truth, virus scenario (b) DPPT, virus scenario

Fig. 10. Ground truth and tracking results of DPPT for the virus scenario with medium density and SNR = 2. A z-slice ($z = 5$) of the original 3D data is shown. The current positions of the individual particles are indicated by cubes, intermediate positions are represented by small spheres along the trajectories. The image contrast was enhanced for better visibility.

Table 8
Quantitative tracking performance of different approaches for data of the virus scenario with SNR = 7 from the Particle Tracking Challenge. The best performance values are highlighted bold and underlined, and the second best performance values are bold.

Metric	α	β	*JSC*	JSC_θ	*RMSE*
Medium density					
Method 5	0.760	0.726	0.825	**0.876**	0.734
Method 1	0.772	0.737	**0.839**	**0.891**	0.806
Method 2	**0.786**	**0.738**	0.827	0.872	0.651
DPHT	0.751	0.691	0.741	0.802	**0.448**
DPPT	**0.827**	**0.775**	**0.833**	0.869	**0.449**
High density					
Method 5	0.665	0.617	**0.718**	**0.806**	0.831
Method 1	0.665	0.612	0.702	0.794	0.881
Method 2	**0.725**	**0.673**	**0.757**	**0.844**	0.706
DPHT	0.706	0.645	0.694	0.781	**0.506**
DPPT	**0.738**	**0.660**	0.710	0.782	**0.504**

$JSC = 0.564$, $JSC_\theta = 0.700$, $RMSE = 1.014$). Thus, DPPT can cope well with somewhat different characteristics of the image data.

3.3.1. Impact of object dynamics

In addition, we studied how changes of the object dynamics affect the tracking performance of DPPT. We simulated image data of the vesicle scenario using the particle tracking benchmark generator of ICY (de Chaumont et al., 2012; Chenouard et al., 2014). We used medium object density, SNR = 2, and different diffusion coefficients ($D_{\text{diff}} = 0, 0.5, 2, 4, 6, 8$ pixel^2/frame) yielding six image sequences with 100 time points each. We trained DPPT only once with training data from our generator, where D_{diff} of individual particles was drawn from a uniform distribution in the interval [1,6] (as in Section 2.5), and applied it to all six image sequences. In Fig. 11 it can be seen that our network is relatively robust and the performance for different metrics is relatively good (e.g., compare with the results of other methods in Table 2 for which there is $D_{\text{diff}} = 1.992$ pixel^2/frame). Also outside the training interval [1,6] reasonable results are obtained. For *RMSE* the performance is rel-

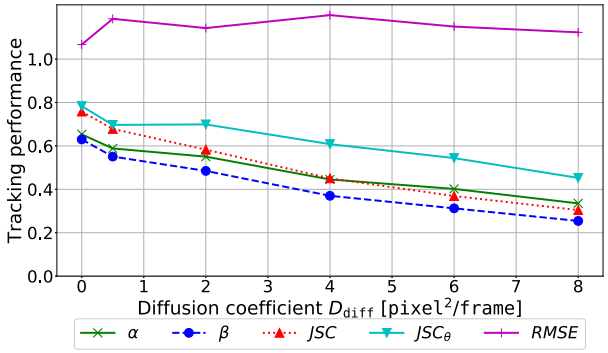


Fig. 11. Tracking performance of DPPT as a function of the diffusion coefficient D_{diff}.

Table 9
Ablation study of our DPPT approach for the vesicle scenario with medium object density and SNR = 2. The best performance values are highlighted bold and underlined.

Experiment	1	2	3	4	5	6	7
Prediction block	✓			✓	✓		✓
Update block		✓		✓		✓	✓
Corresp. finding block			✓		✓	✓	✓
α	0.529	0.517	0.541	0.531	0.561	0.545	**0.562**
β	0.455	0.446	0.465	0.457	0.482	0.471	**0.485**
JSC	0.536	0.521	0.538	0.538	0.562	0.544	**0.564**
JSC_θ	0.686	0.696	0.704	0.685	0.702	**0.709**	0.700
RMSE	1.027	0.856	0.858	1.025	1.014	**0.845**	1.014

atively constant. For the other metrics (α, β, *JSC*, JSC_θ) the performance decreases with increasing D_{diff}. This is expected since the ambiguity and clutter increase with increasing object dynamics (motion strength).

3.3.2. Ablation study

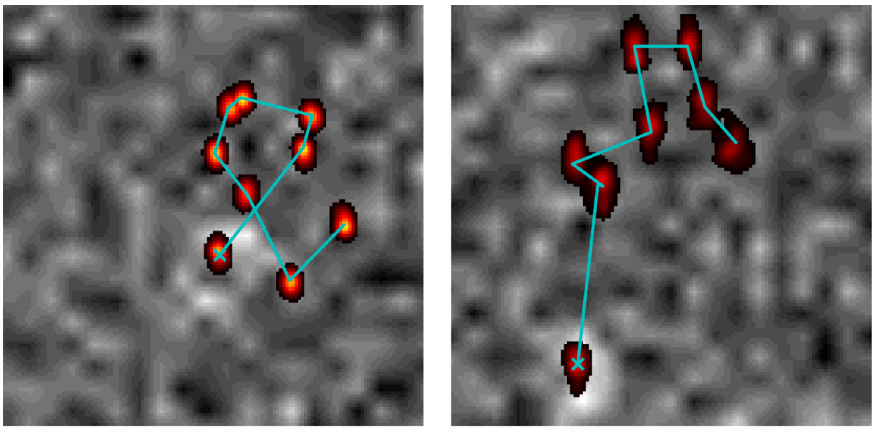
We also conducted an ablation study to investigate the impact of the different components of DPPT on the tracking performance. Therefore, we disabled different components (prediction block, update block, correspondence finding block) and evaluated the performance for the vesicle scenario with medium object density and SNR = 2. In the case of disabling the prediction block, no state prediction was performed and the update step was determined based only on the assigned detections. With the update block disabled, no update step was performed. In this case, the particle states were determined based only on the assigned detections or the predictions (no detection was assigned). When the correspondence finding block was disabled, assignment probabilities as well as probabilities of missing detections were not computed. Table 9 shows the results. It can be seen that each component of DPPT generally improves the tracking result, and the combination yields a further improvement. The main drivers of performance are the prediction and the correspondence finding blocks. In terms of α, β, and *JSC*, the best results are obtained when all components of the approach are enabled. For JSC_θ similar values are obtained for different constellations. When the prediction block is enabled, *RMSE* increases. This is expected, since with a prediction block enabled and using Brownian motion, track points are included which are based only on the prediction in the absence of an assigned detection, and because the *RMSE* of particle detection is generally much smaller than that of the prediction. We also compared our network with and without uncertainty (aleatoric and epistemic uncertainty). Using uncertainty yielded a slight improvement of the tracking performance ($\alpha = 0.562$, $\beta = 0.485$, $JSC = 0.564$, $JSC_\theta = 0.700$, $RMSE = 1.014$) compared to not using uncertainty ($\alpha = 0.560$, $\beta = 0.482$, $JSC = 0.562$, $JSC_\theta = 0.705$, $RMSE = 1.016$). A main advantage of the computed uncertainty is that the suitability of the training data can be assessed to select the data set with the best-suited motion model. If uncertainty is not exploited and a wrong motion model is selected then the tracking performance decreases (Section 3.2.2). For example, if training data with directed motion is used instead of Brownian motion for test data with this type of motion then the tracking performance decreases from $\beta = 0.529$ to $\beta = 0.126$ (see Fig. 8 b). Moreover, the computed uncertainty can be used to exclude low quality tracks to improve the accuracy of subsequent motion analysis (Sections 3.3.3 and 3.4.2). In addition, we compared GRU with LSTM in our network. GRU yielded a slightly better tracking performance ($\alpha = 0.562$, $\beta = 0.485$, $JSC = 0.564$, $JSC_\theta = 0.700$, $RMSE = 1.014$) compared to LSTM ($\alpha = 0.560$, $\beta = 0.481$, $JSC = 0.561$, $JSC_\theta = 0.700$, $RMSE = 1.015$) and the computation time was about 10% lower (due to the lower complexity of GRU compared to LSTM, see Section 2.3).

3.3.3. Exploiting uncertainty information for motion analysis

In addition, we studied the impact of exploiting the computed uncertainty information of DPPT for subsequent *motion analysis*. We considered the vesicle scenario with medium and high object density, and SNR = 1, and determined the diffusion coefficient D_{diff} of the particles. D_{diff} was computed by a mean-squared displacement analysis (de Chaumont et al., 2012). From the computed trajectories we excluded uncertain track points for which the epistemic uncertainty is high (we used a threshold of 1.0 pixel for the standard deviation). The ground truth for the data with high object density is $D_{\text{diff}} = 1.979$ pixel^2/frame, and for medium density we have $D_{\text{diff}} = 2.011$ pixel^2/frame (Chenouard et al., 2014). When considering all track points, the computed diffusion coefficients are $D_{\text{diff}} = 1.571$ pixel^2/frame with a relative error of 20.6% for the high density data, and $D_{\text{diff}} = 2.176$ pixel^2/frame with a relative error of 8.2% for the medium density data. Instead, when excluding uncertain track points, we obtain $D_{\text{diff}} = 1.874$ pixel^2/frame and $D_{\text{diff}} = 1.999$ pixel^2/frame with much lower relative errors of 5.3% and 0.6%, respectively. This demonstrates that the computed uncertainty information of our network can be exploited to improve the accuracy of subsequent motion analysis. Fig. 12 shows example tracking results of DPPT and computed uncertainty (aleatoric and epistemic uncertainty, probability density of state update) for two different trajectories (image sections of 19×19 pixels). One trajectory has a low uncertainty (high probability density values) while the other trajectory has a high uncertainty (low probability density values). Fig. 13 shows tracking results of DPPT and computed uncertainty (aleatoric and epistemic uncertainty, probability density of state update) for a larger image section (36×36 pixels). It can be seen that the uncertainty varies between individual trajectories and track points.

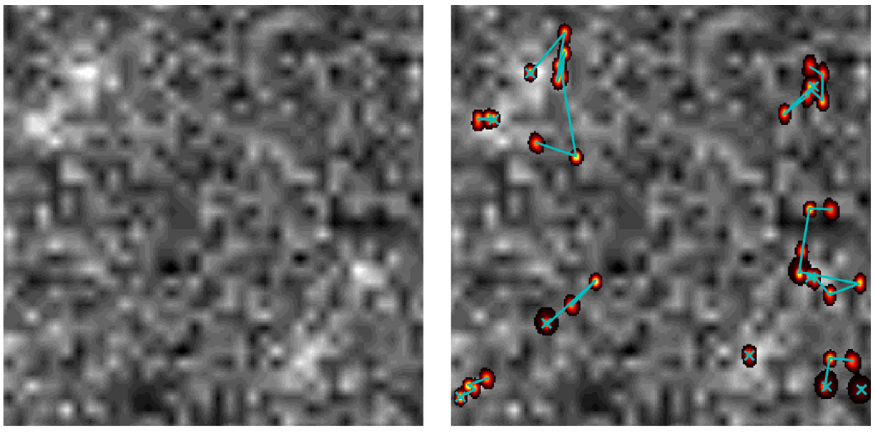
3.3.4. Choice of hyperparameters

We also studied the dependency of the results of DPPT on the hyperparameters. We used image data of the vesicle scenario from the Particle Tracking Challenge with medium object density and SNR = 2. Fig. 14 shows diagrams for the most relevant hyperparameters and for the performance metric β, which is the most comprehensive metric covering all error types (detection, localization, linking). The best parameter settings are marked by red circles, which are the values we used when applying DPPT (both for the Particle Tracking Challenge images and for the real live cell microscopy images). It can be seen that the performance of DPPT is relatively robust within certain ranges of the hyperparameters. The performance decreases strongly when the network gets too complex (e.g., $L > 3$, $R > 70$). Since for some hyperparameters the performance is similar for reduced values, we also tested our network with reduced complexity (less layers and units, e.g., $L = 1$, $K = 10$, Bayesian layers=1, $R = 10$). However, then the performance was reduced, and more importantly, the epistemic uncertainty was too small and not well represented so that motion model selection for the training data did not work well.



(a) Low uncertainty (b) High uncertainty

Fig. 12. Tracking results of DPPT and computed uncertainty (aleatoric and epistemic uncertainty, probability density of state update, yellow: high probability values, red: low probability values) for two different trajectories (image sections of 19×19 pixels) of the vesicle scenario with medium density and SNR = 2. The current position is indicated by a cross. The original images were upscaled using interpolation by a factor 7 and the image contrast was enhanced for better visibility.



(a) Original image (b) Tracking results and uncertainty

Fig. 13. Tracking results of DPPT and computed uncertainty (aleatoric and epistemic uncertainty, probability density of state update, yellow: high probability values, red: low probability values) for an image section (36×36 pixels) of the vesicle scenario with medium density and SNR = 2. The current position is indicated by a cross. The original images were upscaled using interpolation by a factor 7 and the image contrast was enhanced for better visibility.

3.4. Real live cell fluorescence microscopy images

3.4.1. Tracking performance

We also performed a quantitative evaluation of DPPT based on real live cell fluorescence microscopy image data of the hepatitis C virus (HCV) nonstructural protein 5A (NS5A), the HCV associated cellular Apolipoprotein E (ApoE), and chromatin structures (labeled during DNA replication). We considered three 2D image sequences (30 time points, 512×512 pixels, pixel size 0.22×0.22 μm^2, 16-bit) of proteins in HCV proteins expressing Huh7/LunetCD81H cells denoted by Seq. 1 to Seq. 3. Seq. 1 and Seq. 2 display the HCV protein NS5A and Seq. 3 shows the ApoE protein. The images were acquired using a confocal spinning disk microscope and an EMCCD camera (Lee et al., 2019). This data set is challenging due to clustering of particles, clutter, out of focus movement, and relatively low SNR. We also used four 3D image sequences (11 time points, $512 \times 512 \times 5$ voxels, voxel size $0.0410 \times 0.0410 \times 0.125$ μm^3, 16-bit) of chromatin structures (labeled by nucleotide incorporation during DNA replication) in HeLa Kyoto cells denoted by Seq. 4 to Seq. 7. The data was acquired by super-resolution 3D structured illumination microscopy (3D-SIM) using a sCMOS camera (Chagin et al., 2016). Main challenges of this data set are clustering of objects and decreasing SNR over time due to photobleaching. Between 35 and 90 ground truth trajectories for difficult regions were manually annotated in each of the seven image sequences using the ImageJ plugin MTrackJ (Meijering et al., 2012). Table 10 gives an overview of the seven image sequences. Example image sections (200×200 pixels) for Seq. 2 and for a z-slice ($z = 3$) of Seq. 5 are shown in Fig. 15.

We compared the performance of DPPT with the ParticleTracker (PT) (Sbalzarini and Koumoutsakos, 2005), the Kalman filter based approach (KF) implemented in the ImageJ plugin TrackMate (Tinevez et al., 2017), and the multiple hypothesis tracking (MHT) approach (Chenouard et al., 2013) implemented in Icy (de Chaumont et al., 2012). PT (Method 1 of the Particle Tracking Challenge) establishes correspondences by greedy hill-climbing optimization with topological constraints and localizes particles by iterative intensity-weighted centroid calculation. KF employs SEF for particle localization and a linear assignment method for correspondence finding which is very similar to the method used in u-track (Jaqaman et al., 2008). MHT uses multiple motion models and relies on a wavelet-based scheme to localize particles. For PT, KF, and MHT, we tested several parameter settings and used the settings yielding the best tracking results. We also compared DPPT with our previous deep learning approaches DPT (for 2D images) and DPHT (for 2D and 3D images). Note that DPPT, DPT, and DPHT

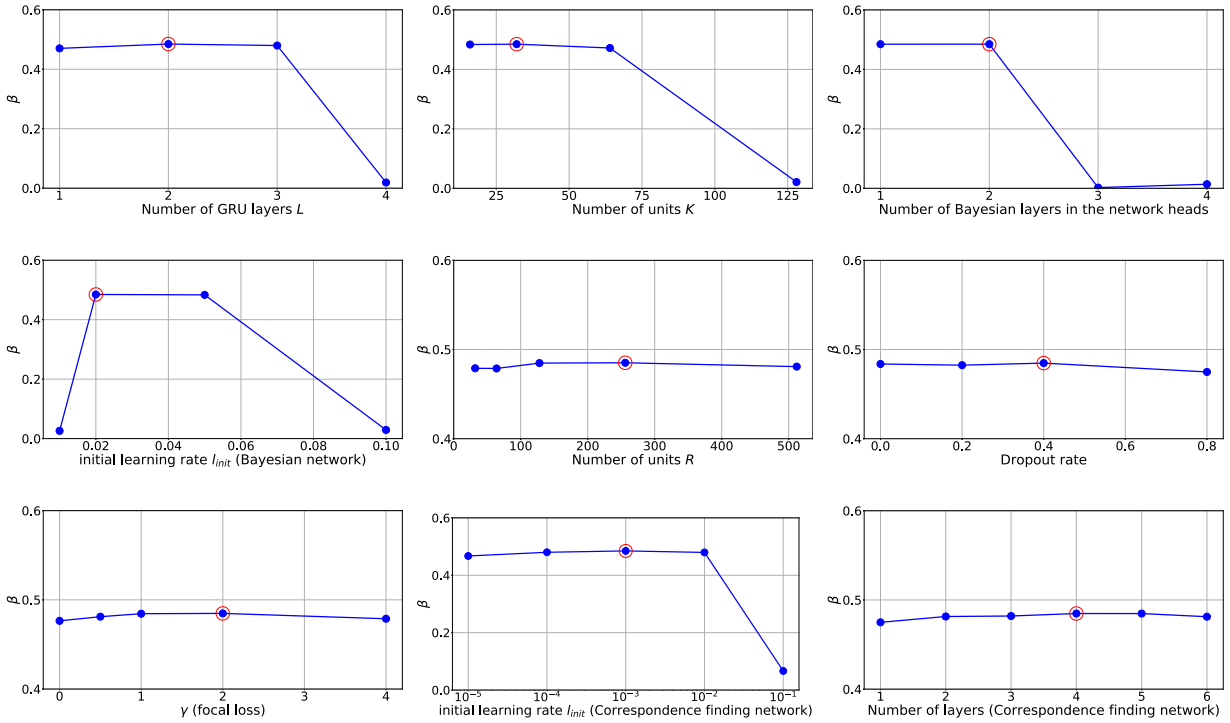
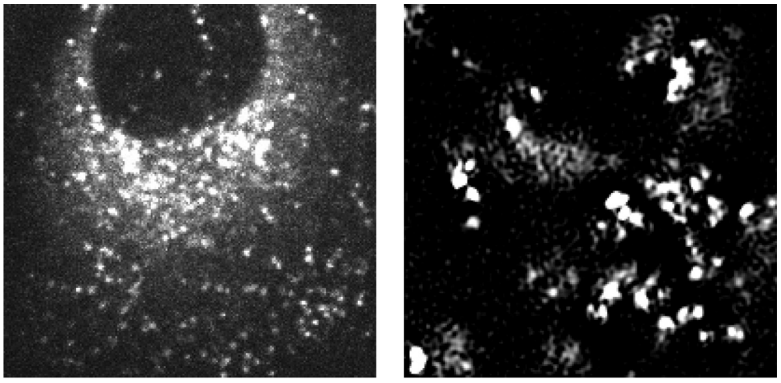


Fig. 14. Tracking performance of DPPT as a function of hyperparameters. Red circles indicate the best performance.

Table 10
Overview of the real fluorescence microscopy sequences used for evalaution.

Sequence	Image dimension	Object type	No. of trajectories
Seq. 1	2D	HCV NS5A protein	75
Seq. 2	2D	HCV NS5A protein	55
Seq. 3	2D	HCV associated ApoE protein	90
Seq. 4	3D	Chromatin structures	66
Seq. 5	3D	Chromatin structures	71
Seq. 6	3D	Chromatin structures	35
Seq. 7	3D	Chromatin structures	60



(a) Seq. 2 (2D, HCV NS5A) (b) Seq. 5 (3D, chromatin structures, z-slice $z = 3$)

Fig. 15. Example image sections (200×200 pixels) of real live cell fluorescence microscopy data. The image contrast was enhanced for better visibility.

were trained based on synthetic data only and then applied to real data. This means that tedious manual annotation of the real data was not required for network training.

Tracking results for all approaches for the 2D and 3D image data are presented in Table 11,12, respectively. It can be seen that DPPT performs best for most image sequences. For β, DPPT outperforms the other approaches in five out of seven cases, and is second best in the remaining two cases. In terms of α and *JSC*, DPPT yields the best result in five out of seven cases, and the second best result in one case. In terms of JSC_θ, DPPT performs best in three cases and second best in four cases. For *RMSE*, DPPT yields the second best result in three cases. Fig. 16 shows ground truth and tracking results for a trajectory of image sequence Seq. 1 (HCV) with complex motion (19×19 pixels section). It can be seen that

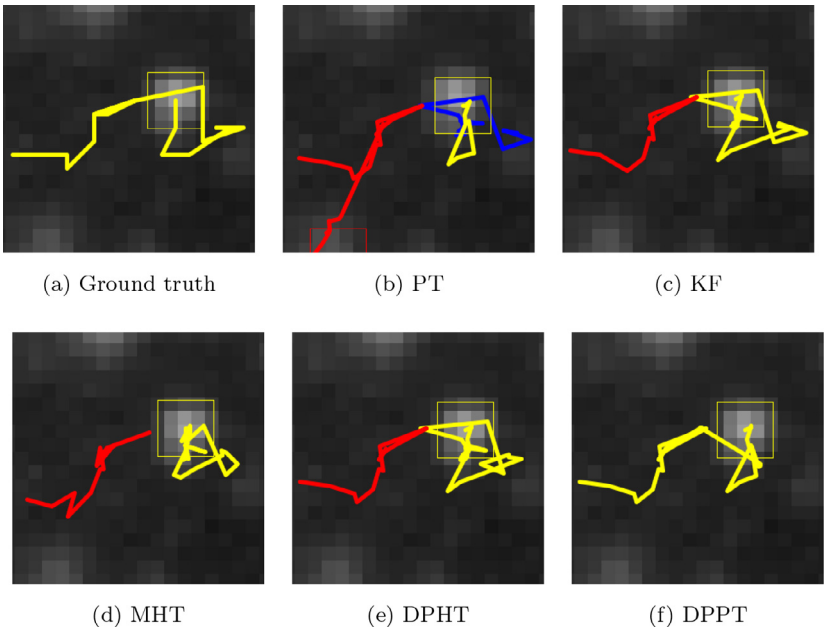


Fig. 16. Ground truth and tracking results of different approaches for a 19 × 19 pixels section of image sequence Seq. 1 (HCV NS5A). The image contrast was enhanced for better visibility.

Table 11
Performance values of different tracking approaches for 2D real fluorescence microscopy image sequences Seq. 1 to 3 displaying HCV NS5A and HCV associated ApoE proteins. The best performance values are highlighted bold and underlined, and the second best performance values are bold. The mean values for all approaches are also shown.

Metric	α	β	JSC	JSC_θ	$RMSE$
Seq. 1 (HCV NS5A)					
PT	0.545	0.506	0.635	0.644	1.253
KF	0.528	0.505	0.586	0.659	0.979
MHT	0.560	0.527	0.648	0.693	1.210
DPT	0.606	0.559	0.649	0.644	**0.904**
DPHT	**0.610**	**0.575**	**0.676**	**0.697**	1.016
DPPT	**0.648**	**0.611**	**0.713**	**0.711**	**0.975**
Seq. 2 (HCV NS5A)					
PT	0.590	0.496	0.629	0.557	**1.064**
KF	0.559	0.481	0.564	0.550	1.088
MHT	0.540	0.480	0.588	0.611	1.237
DPT	**0.633**	0.540	0.632	0.605	**1.008**
DPHT	0.619	**0.556**	**0.652**	**0.622**	1.117
DPPT	**0.639**	**0.567**	**0.658**	**0.630**	1.069
Seq. 3 (HCV associated ApoE)					
PT	0.324	0.306	0.382	0.414	1.255
KF	0.525	0.432	0.499	0.529	1.137
MHT	0.436	0.422	0.507	**0.588**	1.299
DPT	0.614	0.499	0.586	0.527	**1.068**
DPHT	**0.626**	**0.510**	**0.614**	0.538	**1.052**
DPPT	**0.640**	**0.516**	**0.619**	**0.543**	1.192
Mean values					
PT	0.487	0.436	0.549	0.538	1.191
KF	0.537	0.473	0.550	0.579	1.068
MHT	0.512	0.476	0.581	**0.631**	1.249
DPT	0.617	0.533	0.623	0.592	**0.993**
DPHT	**0.618**	**0.547**	**0.647**	0.619	**1.062**
DPPT	**0.642**	**0.565**	**0.663**	**0.628**	1.079

Table 12
Performance values of different tracking approaches for 3D real fluorescence microscopy image sequences Seq. 4 to 7 displaying chromatin structures. The best performance values are highlighted bold and underlined, and the second best performance values are bold. The mean values for all approaches are also shown.

Metric Seq. 4	α	β	JSC	JSC_θ	$RMSE$
PT	0.305	0.285	0.415	**0.500**	**1.823**
KF	0.336	0.282	0.408	0.489	2.048
MHT	**0.398**	0.292	0.420	0.462	2.014
DPHT	**0.353**	**0.337**	**0.502**	**0.551**	1.992
DPPT	0.331	**0.315**	**0.444**	**0.500**	**1.944**
Seq. 5					
PT	0.334	0.255	**0.493**	0.523	2.454
KF	**0.381**	0.290	**0.466**	0.528	2.112
MHT	0.337	0.286	0.443	0.529	2.189
DPHT	0.365	**0.302**	0.436	**0.565**	**1.902**
DPPT	**0.384**	**0.326**	0.463	**0.593**	**1.956**
Seq. 6					
PT	0.443	0.415	**0.657**	0.683	1.962
KF	**0.540**	0.446	0.587	0.673	**1.582**
MHT	0.407	0.390	0.572	0.590	2.044
DPHT	0.493	**0.485**	0.641	**0.806**	**1.675**
DPPT	**0.590**	**0.510**	**0.701**	**0.714**	1.696
Seq. 7					
PT	0.439	0.400	0.587	0.649	1.930
KF	0.415	0.330	0.534	0.553	2.272
MHT	**0.453**	**0.417**	**0.604**	0.647	**1.926**
DPHT	0.424	0.403	0.574	**0.708**	**1.904**
DPPT	**0.447**	**0.415**	**0.624**	**0.701**	2.058
Mean values					
PT	0.380	0.339	**0.538**	0.589	2.042
KF	**0.418**	0.337	0.499	0.561	2.004
MHT	0.399	0.346	0.510	0.557	2.043
DPHT	0.409	**0.382**	**0.538**	**0.657**	**1.869**
DPPT	**0.438**	**0.391**	**0.558**	**0.627**	**1.913**

the previous approaches (PT, KF, MHT, DPHT) yield broken trajectories, whereas DPPT yields a trajectory without gaps which agrees well with the ground truth.

3.4.2. Exploiting uncertainty information for motion analysis

In addition, we studied the impact of exploiting the computed uncertainty of DPPT for subsequent *motion analysis* in real live cell microscopy images. For Seq. 1 and Seq. 2, we excluded uncertain track points for which the epistemic uncertainty is high (we used a threshold of 0.5 pixel for the standard deviation) and determined the diffusion coefficient D_{diff} of the particles (as in Section 3.3.3 for the Particle Tracking

Challenge data). The ground truth for Seq. 1 and Seq. 2 is $D_{\text{diff}} = 0.199$ pixel^2/frame and $D_{\text{diff}} = 0.769$ pixel^2/frame, respectively. When considering all track points, the computed diffusion coefficients are $D_{\text{diff}} = 0.189$ pixel^2/frame with a relative error of 5.0% for Seq. 1, and $D_{\text{diff}} = 0.440$ pixel^2/frame with a relative error of 42.8% for Seq. 2. Instead, when excluding uncertain track points, we obtain $D_{\text{diff}} = 0.190$ pixel^2/frame for Seq. 1, and $D_{\text{diff}} = 0.816$ pixel^2/frame for Seq. 2 with lower relative errors of 4.5% and 6.1%, respectively. This confirms the results for the Particle Tracking Challenge data (Section 3.3.3), and demonstrates for real live cell microscopy images that the computed uncertainty information of our network can be exploited to improve the accuracy of subsequent motion analysis.

4. Conclusion

We have introduced a novel probabilistic deep learning approach for tracking multiple particles in fluorescence microscopy image sequences. The proposed Deep Probabilistic Particle Tracker (DPPT) is based on a recurrent neural network which mimics classical Bayesian filtering. Compared to previous methods for particle tracking, our approach takes into account uncertainty, both aleatoric (intrinsic noise in the data) and epistemic uncertainty (uncertainty in network weights). The network exploits short and long-term temporal dependencies in the object dynamics to predict the state at the next time point, and uses assigned detections to update the predicted state. For correspondence finding, we have introduced a neural network that computes assignment probabilities jointly across multiple detections as well as determines probabilities for missing detections. Network training requires only simulated data and therefore tedious manual annotation of ground truth is not necessary. We proposed a novel scheme to generate synthetic training data using automatically extracted information from the real images. This enables simulating a large amount of training data that represent well the images in an application.

We verified that both types of uncertainty (aleatoric and epistemic uncertainty) are captured by the proposed Bayesian neural network and carried out an evaluation of uncertainty estimation. An advantage of our network is that information about the reliability of the extracted trajectories is determined. We demonstrated that the computed uncertainty can be exploited to increase the accuracy of subsequent motion analysis by excluding unreliable track points. In addition, we demonstrated that the uncertainty can be exploited to assess the suitability of the training data and to select the training data set with the best-suited motion model so that the training data better represents the real data in an application. The uncertainty could also be used to calibrate a microscopy experiment by adjusting acquisition parameters. We conducted a quantitative evaluation of the tracking performance using 2D and 3D image data of the Particle Tracking Challenge as well as 2D and 3D real live cell fluorescence microscopy image sequences. A comparison with previous methods showed that DPPT yields state-of-the-art or improved results. In future work, we will apply DPPT to fluorescence microscopy images from other applications.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Roman Spilger: Conceptualization, Methodology, Software, Investigation, Writing - original draft, Writing - review & editing. **Ji-Young Lee:** Resources, Writing - review & editing. **Vadim O. Chagin:** Resources, Writing - review & editing. **Lothar Schermelleh:** Resources, Writing - review & editing. **M. Cristina Cardoso:** Resources, Writing - review & editing. **Ralf Bartenschlager:** Resources, Writing - review & editing. **Karl Rohr:** Conceptualization, Methodology, Software, Investigation, Writing - original draft, Writing - review & editing.

Acknowledgments

Support of the DFG (German Research Foundation) within the SFB 1129 (projects Z4, P11), project number 240245660, and within the SPP 2202 (RO 2471/10-1, CA 198/15-1) is gratefully acknowledged. LS acknowledges support by the Welcome Trust Strategic Award 107457/Z/15/Z funding advanced imaging at Micron Oxford.

References

Abadi, M., et al., 2016. Tensorflow: a system for large-scale machine learning. In: Proc. USENIX Conference on Operating Systems Design and Implementation (OSDI 2016), Savannah, GA, USA, pp. 265–283.

Applegate, K.T., Besson, S., Matov, A., Bagonis, M.H., Jaqaman, K., Danuser, G., 2011. plusTipTracker: quantitative image analysis software for the measurement of microtubule dynamics. J. Struct. Biol. 176 (2), 168–184.

Blundell, C., Cornebise, J., Kavukcuoglu, K., Wierstra, D., 2015. Weight uncertainty in neural networks. In: Proc. International Conference on Machine Learning (ICML 2015), Lille, France, pp. 1613–1622.

Cardinale, J., Rauch, A., Barral, Y., Szekely, G., Sbalzarini, I.F., 2009. Bayesian image analysis with on-line confidence estimates and its application to microtubule tracking. In: Proc. IEEE International Symposium on Biomedical Imaging (ISBI 2009), Boston, MA, USA, pp. 1091–1094.

Chagin, V.O., Casas-Delucchi, C.S., Reinhart, M., Schermelleh, L., Markaki, Y., Maiser, A., Bolius, J.J., Bensimon, A., Fillies, M., Domaing, P., Rozanov, Y.M., Leonhardt, H., Cardoso, M.C., 2016. 4D Visualization of replication foci in mammalian cells corresponding to individual replicons. Nat. Commun. 7, 11231.

de Chaumont, F., Dallongeville, S., Chenouard, N., Herv, N., Pop, S., Provoost, T., Meas-Yedid, V., Pankajakshan, P., Lecomte, T., Le Montagner, Y., Lagache, T., Dufour, A., Olivo-Marin, J.-C., 2012. Icy: an open bioimage informatics platform for extended reproducible research. Nat. Methods 9 (7), 690–696.

Chen, L., Ai, H., Shang, C., Zhuang, Z., Bai, B., 2017. Online multi-object tracking with convolutional neural networks. In: IEEE International Conference on Image Processing (ICIP 2017), Beijing, China, pp. 645–649.

Chen, Y., Yang, X., Zhong, B., Pan, S., Chen, D., Zhang, H., 2016. CNNTracker: Online discriminative object tracking via deep convolutional neural network. Appl. Soft Comput. 38, 1088–1098.

Chenouard, N., Bloch, I., Olivo-Marin, J.C., 2013. Multiple hypothesis tracking for cluttered biological image sequences. IEEE Trans. Pattern Anal. Mach. Intell. 35 (11), 2736–3750.

Chenouard, N., et al., 2014. Objective comparison of particle tracking methods. Nat. Methods 11 (3), 281–289.

Cho, K., van Merrienboer, B., Gülçehre, Ç., Bougares, F., Schwenk, H., Bengio, Y., 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. CoRR. abs/1406.1078

Ciaparrone, G., Snchez, F.L., Tabik, S., Troiano, L., Tagliaferri, R., Herrera, F., 2020. Deep learning in video multi-object tracking: A survey. Neurocomputing 381, 61–88.

Collinet, C., Stöter, M., Bradshaw, C.R., Samusik, N., Rink, J.C., Kenski, D., Habermann, B., Buchholz, F., Henschel, R., Mueller, M.S., Nagel, W.E., Fava, E., Kalaidzidis, Y., Zerial, M., 2010. Systems survey of endocytosis by multiparametric image analysis. Nature 464 (7286), 243–249.

Coraluppi, S., Carthel, C., 2011. Multi-stage multiple-hypothesis tracking. J. Adv. Inf. Fusion 6 (1), 57–67.

Dillon, J.V., Langmore, I., Tran, D., Brevdo, E., Vasudevan, S., Moore, D., Patton, B., Alemi, A., Hoffman, M.D., Saurous, R.A., 2017. Tensorflow distributions. CoRR. abs/1711.10604

Dmitrieva, M., Zenner, H.L., Richens, J., Johnston, D.S., Rittscher, J., 2019. Protein tracking by CNN-based candidate pruning and two-step linking with Bayesian network. In: Proc. IEEE International Workshop on Machine Learning for Signal Processing (MLSP 2019), pp. 1–6.

Esser, P., Sutter, E., 2018. A variational u-net for conditional appearance and shape generation. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2018), Salt Lake City, UT, USA, pp. 8857–8866.

Farrell, S., Anderson, D., Calafiura, P., Cerati, G., Gray, L., Kowalkowski, J., Mudigonda, M., Prabhat, Spentzouris, P., Spiropoulou, M., Tsaris, A., Vlimant, J.-R., Zheng, S., 2017. The HEP.TrkX Project: deep neural networks for HL-LHC online and offline tracking. EPJ Web Conf. 150, 00003.

Frey, B.J., Hinton, G.E., 1999. Variational learning in nonlinear Gaussian belief networks. Neural Comput. 11 (1), 193–213.

Gal, Y., Ghahramani, Z., 2016. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In: Proc. International Conference on Machine Learning (ICML 2016), pp. 1050–1059.

Genovesio, A., Liedl, T., Emiliani, V., Parak, W.J., Coppey-Moisan, M., Olivo-Marin, J., 2006. Multiple particle tracking in 3-D+t microscopy: Method and application to the tracking of endocytosed quantum dots. IEEE Trans. Image Process. 15 (5), 1062–1070.
Godinez, W.J., Lampe, M., Wörz, S., Müller, S., Eils, R., Rohr, K., 2009. Deterministic and probabilistic approaches for tracking virus particles in time-lapse fluorescence microscopy image sequences. Med. Image Anal. 13 (2), 325–342.
Godinez, W.J., Rohr, K., 2015. Tracking multiple particles in fluorescence time-lapse microscopy images via probabilistic data association. IEEE Trans. Med. Imag. 34 (2), 415–432.
Gong, W., Li, Y., Hernández-Lobato, J.M., 2019. Meta-learning for stochastic gradient MCMC. In: Proc. International Conference on Learning Representations (ICLR 2019), New Orleans, LA, USA.
Greenspan, H., van Ginneken, B., Summers, R.M., 2016. Deep learning in medical imaging: Overview and future promise of an exciting new technique. IEEE Trans. Med. Imag. 35 (5), 1153–1159.
Gu, S.S., Ghahramani, Z., Turner, R.E., 2015. Neural adaptive sequential Monte Carlo. In: Proc. Advances in Neural Information Processing Systems (NIPS 2015), Montréal, Canada.
Gudla, P.R., Nakayama, K., Pegoraro, G., Misteli, T., 2017. SpotLearn: convolutional neural network for detection of fluorescence in situ hybridization (fish) signals in high-throughput imaging approaches. Cold Spring Harb. Symp. Quant. Biol. 82, 57–70.
Hayashida, J., Bise, R., 2019. Cell tracking with deep learning for cell detection and motion estimation in low-frame-rate. In: Proc. Medical Image Computing and Computer Assisted Intervention (MICCAI 2019), Shenzhen, China, pp. 397–405.
He, K., Gkioxari, G., Dollr, P., Girshick, R., 2017. Mask R-CNN. In: Proc. IEEE International Conference on Computer Vision (ICCV 2017), Venice, Italy, pp. 2980–2988.
He, K., Zhang, X., Ren, S., Sun, J., 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proc. IEEE International Conference on Computer Vision (ICCV 2015), Santiago, Chile, pp. 1026–1034.
He, T., Mao, H., Guo, J., Yi, Z., 2017. Cell tracking using deep neural networks with multi-task learning. Image Vis. Comput. 60, 142–153.
Hernandez, A., Gall, J., Moreno-Noguer, F., 2019. Human motion prediction via spatio-temporal inpainting. In: Proc. IEEE International Conference on Computer Vision (ICCV 2019), Seoul, Korea, pp. 3823–3832.
Hernández-Lobato, J.M., Adams, R.P., 2015. Probabilistic backpropagation for scalable learning of Bayesian neural networks. In: Proc. International Conference on International Conference on Machine Learning (ICML 2015), pp. 1861–1869. Lille, France
Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9 (8), 1735–1780.
Jaqaman, K., Loerke, D., Mettlen, M., Kuwata, H., Grinstein, S., Schmid, S.L., Danuser, G., 2008. Robust single-particle tracking in live-cell time-lapse sequences. Nat. Methods 5 (8), 695–702.
Jonker, R., Volgenant, A., 1987. A shortest augmenting path algorithm for dense and sparse linear assignment problems. Computing 38 (4), 325–340.
Kendall, A., Gal, Y., 2017. What uncertainties do we need in Bayesian deep learning for computer vision? In: Proc. International Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, pp. 5580–5590.
Kingma, D.P., Welling, M., 2014. Auto-encoding variational Bayes. In: Proc. International Conference on Learning Representations (ICLR 2014), Banff, Canada.
Kohl, S., Romera-Paredes, B., Meyer, C., De Fauw, J., Ledsam, J.R., Maier-Hein, K., Eslami, S.M.A., Jimenez Rezende, D., Ronneberger, O., 2018. A probabilistic U-Net for segmentation of ambiguous images. In: Advances in Neural Information Processing Systems (NIPS 2018), Montréal, Canada, pp. 6965–6975.
Lakshminarayanan, B., Pritzel, A., Blundell, C., 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. In: Proc. International Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, pp. 6405–6416.
LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521, 436–444.
Lee, J.-Y., Cortese, M., Haselmann, U., Tabata, K., Romero-Brey, I., Funaya, C., Schieber, N.L., Qiang, Y., Bartenschlager, M., Kallis, S., Ritter, C., Rohr, K., Schwab, Y., Ruggieri, A., Bartenschlager, R., 2019. Spatiotemporal coupling of the Hepatitis C virus replication cycle by creating a lipid droplet-proximal membranous replication compartment. Cell Rep. 27, 3602–3617.e5.
Lee, K., Wang, Z., Vlahov, B., Brar, H., Theodorou, E.A., 2019. Ensemble Bayesian decision making with redundant deep perceptual control policies. In: Proc. IEEE International Conference On Machine Learning And Applications (ICMLA 2019), Boca Raton, FL, USA, pp. 831–837.
Leibig, C., Allken, V., Ayhan, M.S., Berens, P., Wahl, S., 2017. Leveraging uncertainty information from deep neural networks for disease detection. Sci. Rep. 7, 17816.
Lin, T., Goyal, P., Girshick, R., He, K., Dollár, P., 2020. Focal loss for dense object detection. IEEE Trans. Pattern Anal. Mach. Intell. 42 (2), 318–327.
Litjens, G., Kooi, T., Bejnordi, B.E., Setio, A.A.A., Ciompi, F., Ghafoorian, M., van der Laak, J.A., van Ginneken, B., Sánchez, C.I., 2017. A survey on deep learning in medical image analysis. Med. Image Anal. 42, 60–88.
Ma, H., Smal, I., Daemen, J., van Walsum, T., 2020. Dynamic coronary roadmapping via catheter tip tracking in x-ray fluoroscopy with deep learning based Bayesian filtering. Med. Image Anal. 61, 101634.
Meijering, E., Dzyubachyk, O., Smal, I., 2012. Methods for cell and particle tracking. In: Imaging and Spectroscopic Analysis of Living Cells. In: Methods in Enzymology, Vol. 504. Academic Press, pp. 183–200.
Milan, A., Rezatofighi, S.H., Dick, A., Reid, I., Schindler, K., 2017. Online multi-target tracking using recurrent neural networks. In: Proc. Conference on Artificial Intelligence (AAAI 2017), San Francisco, CA, USA, pp. 4225–4232.
Newby, J.M., Schaefer, A.M., Lee, P.T., Forest, M.G., Lai, S.K., 2018. Convolutional neural networks automate detection for tracking of submicron-scale particles in 2D and 3D. Proc. Natl. Acad. Sci. U.S.A. 115 (36), 9026–9031.
Nishimoto, S., Tokuoka, Y., Yamada, T.G., Hiroi, N.F., Funahashi, A., 2019. Predicting the future direction of cell movement with convolutional neural networks. PLOS ONE 14 (9), 1–14.
Paavolainen, L., Kankaanpää, P., Ruusuvuori, P., McNerney, G., Karjalainen, M., Marjomäki, V., 2012. Application independent greedy particle tracking method for 3D fluorescence microscopy image series. In: Proc. IEEE International Symposium on Biomedical Imaging (ISBI 2012), Barcelona, Spain, pp. 672–675.
Payer, C., Štern, D., Feiner, M., Bischof, H., Urschler, M., 2019. Segmenting and tracking cell instances with cosine embeddings and recurrent hourglass networks. Med. Image Anal. 57, 106–119.
Reddi, S.J., Kale, S., Kumar, S., 2018. On the convergence of Adam and beyond. In: Proc. International Conference on Learning Representations (ICLR 2018), Vancouver, Canada.
Ritter, C., Imle, A., Lee, J.Y., Müller, B., Fackler, O.T., Bartenschlager, R., Rohr, K., 2018. Two-filter probabilistic data association for tracking of virus particles in fluorescence microscopy images. In: Proc. IEEE International Symposium on Biomedical Imaging (ISBI 2018), Washington, DC, USA, pp. 957–960.
Roudot, P., Ding, L., Jaqaman, K., Kervrann, C., Danuser, G., 2017. Piecewise-stationary motion modeling and iterative smoothing to track heterogeneous particle motions in dense environments. IEEE Trans. Image Process. 26 (11), 5395–5410.
Ruhnow, F., Zwicker, D., Diez, S., 2011. Tracking single particles and elongated filaments with nanometer precision. Biophys. J. 100 (11), 2820–2828.
Sadeghian, A., Alahi, A., Savarese, S., 2017. Tracking the untrackable: Learning to track multiple cues with long-term dependencies. In: Proc. IEEE International Conference on Computer Vision (ICCV 2017), Venice, Italy, pp. 300–311.
Sage, D., Neumann, F.R., Hediger, F., Gasser, S.M., Unser, M., 2005. Automatic tracking of individual fluorescence particles: application to the study of chromosome dynamics. IEEE Trans. Image Process. 14 (9), 1372–1383.
Sbalzarini, I., Koumoutsakos, P., 2005. Feature point tracking and trajectory analysis for video imaging in cell biology. J. Struct. Biol. 151 (2), 182–195.
Smal, I., Draegestein, K., Galjart, N., Niessen, W., Meijering, E., 2008. Particle filtering for multiple object tracking in dynamic fluorescence microscopy images: Application to microtubule growth analysis. IEEE Trans. Med. Imag. 27 (6), 789–804.
Smal, I., Yao, Y., Galjart, N., Meijering, E., 2019. Facilitating data association in particle tracking using autoencoding and score matching. In: Proc. IEEE International Symposium on Biomedical Imaging (ISBI 2019), Venice, Italy, pp. 1523–1526.
Spilger, R., Imle, A., Lee, J., Müller, B., Fackler, O.T., Bartenschlager, R., Rohr, K., 2020. A recurrent neural network for particle tracking in microscopy images using future information, track hypotheses, and multiple detections. IEEE Trans. Image Process. 29, 3681–3694.
Spilger, R., Wollmann, T., Qiang, Y., Imle, A., Lee, J.Y., Müller, B., Fackler, O.T., Bartenschlager, R., Rohr, K., 2018. Deep particle tracker: automatic tracking of particles in fluorescence microscopy images using deep learning. In: Proc. Deep Learning in Medical Image Analysis (DLMIA), Vol. 11045, Granada, Spain, pp. 128–136.
Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 15 (56), 1929–1958.
Su, Q., Liao, X., Chen, C., Carin, L., 2016. Nonlinear statistical learning with truncated Gaussian graphical models. In: Proc. International Conference on Machine Learning (ICML 2016), New York City, NY, USA, pp. 1948–1957.
Sun, R., Paninski, L., 2018. Scalable approximate Bayesian inference for particle tracking data. In: Proc. International Conference on Machine Learning (ICML 2018), 80, pp. 4800–4809. Stockholm, Sweden
Tanno, R., Worrall, D., Kaden, E., Ghosh, A., Grussu, F., Bizzi, A., Sotiropoulos, S.N., Criminisi, A., Alexander, D.C., 2019. Uncertainty quantification in deep learning for safer neuroimage enhancement. CoRR abs/1907.13418.
Tinevez, J.-Y., Perry, N., Schindelin, J., Hoopes, G.M., Reynolds, G.D., Laplantine, E., Bednarek, S.Y., Shorte, S.L., Eliceiri, K.W., 2017. Trackmate: An open and extensible platform for single-particle tracking. Methods 115, 80–90.
Ullah, M., Alaya Cheikh, F., 2018. Deep feature based end-to-end transportation network for multi-target tracking. In: Proc. IEEE International Conference on Image Processing (ICIP 2018), Athens, Greece, pp. 3738–3742.
Wang, H., SHI, X., Yeung, D.-Y., 2016. Natural-parameter networks: a class of probabilistic neural networks. In: Proc. International Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, pp. 118–126.
Wang, L., Xu, L., Kim, M.Y., Rigazico, L., Yang, M., 2017. Online multiple object tracking via flow and convolutional features. In: Proc. IEEE International Conference on Image Processing (ICIP 2017), Beijing, China, pp. 3630–3634.
Wollmann, T., Ritter, C., Dohrke, J., Lee, J.-Y., Bartenschlager, R., Rohr, K., 2019. Detnet: deep neural network for particle detection in fluorescence microscopy images. In: Proc. IEEE International Symposium on Biomedical Imaging (ISBI 2019), Venice, Italy. 517–512
Yang, L., Qiu, Z., Greenaway, A.H., Lu, W., 2012. A new framework for particle detection in low-SNRfluorescence live-cell images and its application for improved particle tracking. IEEE Trans. Biomed. Eng. 59 (7), 2040–2050.
Yao, Y., Smal, I., Grigoriev, I., Akhmanova, A., Meijering, E., 2020. Deep-learning method for data association in particle tracking. Bioinformatics. Btaa597

Yao, Y., Smal, I., Meijering, E., 2018. Deep neural networks for data association in particle tracking. In: Proc. IEEE International Symposium on Biomedical Imaging (ISBI 2018), Washington, DC, USA, pp. 458–461.
Yuan, L., Zheng, Y.F., Zhu, J., Wang, L., Brown, A., 2012. Object tracking with particle filtering in fluorescence microscopy images: Application to the motion of neurofilaments in axons. IEEE Trans. Med. Imag. 31 (1), 117–130.
Zhong, Y., Li, C., Zhou, H., Wang, G., 2018. Developing noise-resistant three-dimensional single particle tracking using deep neural networks. Anal. Chem. 90 (18), 10748–10757.
Zhu, J., Yang, H., Liu, N., Kim, M., Zhang, W., Yang, M.-H., 2018. Online multi-object tracking with dual matching attention networks. In: Proc. European Conference on Computer Vision (ECCV 2018), Munich, Germany, pp. 379–396.