

Microparticle cloud imaging and tracking for data-driven plasma science

Cite as: Phys. Plasmas **27**, 033703 (2020); <https://doi.org/10.1063/1.5134787>

Submitted: 02 November 2019 . Accepted: 13 February 2020 . Published Online: 05 March 2020

Zhehui Wang , Jiayi Xu , Yao E. Kovach , Bradley T. Wolfe , Edward Thomas, Hanqi Guo , John E. Foster , and Han-Wei Shen

COLLECTIONS

Paper published as part of the special topic on [Invited Papers from the 2nd International Conference on Data-Driven Plasma Science](#)

Note: This paper is part of the Special Collection: Invited Papers from the 2nd International Conference on Data-Driven Plasma Science. Note: Invited talk presented to the 2nd International Conference on Data Driven Plasma Science (ICDDPS) 2019, Marseille, France, May 13–17, 2019.



View Online



Export Citation



CrossMark

ARTICLES YOU MAY BE INTERESTED IN

[Time-dependent modeling of coupled plasma-wall dynamics](#)

Physics of Plasmas **27**, 032503 (2020); <https://doi.org/10.1063/1.5140183>

[Dust-density waves in radio-frequency discharges under magnetic fields](#)

Physics of Plasmas **27**, 033704 (2020); <https://doi.org/10.1063/1.5144591>

[Runaway electron transport in stochastic toroidal magnetic fields](#)

Physics of Plasmas **27**, 032502 (2020); <https://doi.org/10.1063/1.5135588>






AVS Quantum Science

A new interdisciplinary home for impactful quantum science research and reviews

Co-Published by




NOW ONLINE

Microparticle cloud imaging and tracking for data-driven plasma science

Cite as: Phys. Plasmas **27**, 033703 (2020); doi: [10.1063/1.5134787](https://doi.org/10.1063/1.5134787)

Submitted: 2 November 2019 · Accepted: 13 February 2020 ·

Published Online: 5 March 2020



View Online



Export Citation



CrossMark

Zhehui Wang,^{1,a)} Jiayi Xu,² Yao E. Kovach,³ Bradley T. Wolfe,¹ Edward Thomas, Jr.,⁴ Hanqi Guo,⁵ John E. Foster,³ and Han-Wei Shen²

AFFILIATIONS

¹Los Alamos National Laboratory, Los Alamos, New Mexico 87545, USA

²The Ohio State University, Columbus, Ohio 43210, USA

³University of Michigan, Ann Arbor, Michigan 48109, USA

⁴Auburn University, Auburn, Alabama 36849, USA

⁵Argonne National Laboratory, Lemont, Illinois 60439, USA

Note: This paper is part of the Special Collection: Invited Papers from the 2nd International Conference on Data-Driven Plasma Science.

Note: Invited talk presented to the 2nd International Conference on Data Driven Plasma Science (ICDDPS) 2019, Marseille, France, May 13–17, 2019.

^{a)}Author to whom correspondence should be addressed: zwang@lanl.gov

ABSTRACT

Oceans of image and particle track data encountered in plasma interactions with microparticle clouds motivate development and applications of machine-learning (ML) algorithms. A local-constant-velocity tracker, a Kohonen neural network or self-organizing map, the feature tracking kit, and U-Net are described and compared with each other for microparticle cloud datasets generated from exploding wires, dusty plasmas, and atmospheric plasmas. Particle density and the signal-to-noise ratio have been identified as two important factors that affect the tracking accuracy. Fast Fourier transform is used to reveal how U-Net, a deep convolutional neural network developed for non-plasma applications, achieves the improvements for noisy scenes. Viscous effects are revealed in the ballistic motions of the particles from the exploding wires and atmospheric plasmas. Subdiffusion of microparticles satisfying $\Delta r^2 \propto t^k$ ($k = 0.84 \pm 0.02$) is obtained from the dusty plasma datasets. Microparticle cloud imaging and tracking, when enhanced with data and ML models, present new possibilities for plasma physics.

Published under license by AIP Publishing. <https://doi.org/10.1063/1.5134787>

I. INTRODUCTION

Automated knowledge discovery through massive data mining is a recent concept of plasma physics. Fueled by advances in instrumentation and the oceans of archived and new data, it seems unavoidable to use computer hardware and machine-learning (ML) algorithms to replace or heavily supplement manual data mining for knowledge or real-time controls of experiments. How much data are produced in today's plasma physics experiments? Many experiments, if not already, have the capability to produce 1 terabyte (TB) of data per day. The data rate is expected to grow due to the advances in instrumentation, which parallel Moore's law for transistors. One scenario to obtain the TB/day rate is through imaging of plasmas using video cameras. For image sizes of 1 megabyte (MB) and a video recording rate of 10^2 frames per second (fps), the data can be generated at 10^8 bytes/s. 1 TB

of data can be collected in less than 3 h per camera from a continuous plasma experiment. The state-of-the-art smart phone cameras, with image sizes exceeding 10 MB and a constant streaming rate above 10 fps, can exceed such a data rate. In practice, this may not have been done often due to other factors such as the bandwidth of the data transmission or the storage space available. Several commercial off-the-shelf high-speed cameras can deliver a data rate at 10^4 MB/s. In the exploding wire experiment shown in Sec. V, 800×1280 8-bit images were generated at 25 kfps. When such a high-speed camera is used in a pulsed 10-ms-long plasma experiment, an experimental duty cycle at 1 Hz can also produce 1 TB of image data within 3 h. Computer simulations and especially large-scale simulations carried out on supercomputers or large computer clusters, not a focus here, can readily exceed 10 TB/day data rate through parallelization.

How to couple the existing frameworks of physics and statistics to ML algorithms and *vice versa* presents another opportunity toward automated plasma knowledge discovery. On the one hand, a ML process can be symbolically described by a functional dependence $\mathbf{Y} = f(\mathbf{X})$, where $\mathbf{X}$ stands for the inputs such as the one-dimensional time series of a probe measurement or the two-dimensional raw images from a plasma and $\mathbf{Y}$ is the output such as microparticle trajectories or a track classification or the growth rate of an instability. On the other hand, the existing formulas to derive f are usually decoupled from the physics context. If each step such as denoising, image convolution, multi-pixel averaging, or “pooling” is presented by a “daughter” function, f_i , then $f(\cdot) = f_1(f_2(f_3(\dots(f_n(\cdot))\dots)))$, while n could be a large number and thus requires automation or ML algorithms to generate. Is there a systematic way to derive f and f_i based on plasma physics and statistics? Such and similar questions not only stimulate theoretical interest but may also have bearing on ML algorithm optimization for efficient data mining and knowledge discovery.

We shall limit our discussions to the settings of plasma interactions with microparticle clouds. Datasets in this work came from three types of microparticle clouds: exploding wires, dusty plasmas, and atmospheric plasmas. The rich physics and phenomena of microparticle-plasma interactions and the availability of TB datasets motivate new ML approaches. Imaging of microparticle interaction with plasmas has also been ongoing under micro-gravity,^{1,2} where about 20 TB of data is now available, and can be extended to other plasmas including high-temperature magnetic fusion plasmas.^{3–6} When the core electron temperature can reach 10 keV and the edge temperature can be at least tens of eV, there are additional complications of time-dependent microparticle size, shape, and mass due to evaporation and sublimation. Particle nucleation and agglomeration can meanwhile lead to the growth of micrometer and larger particles in an environment such as semiconductor and other material processing plasmas. In the simple scenarios when the particle size can be treated as a constant, outstanding physical questions related to the microparticle-plasma interactions include the microparticle charging and the mechanisms behind the sophisticated motion patterns.

The rest of this paper is partitioned as follows. Section II presents an overview of microparticle tracking models motivated by physics and statistics, followed by Sec. III on data models. Three tracking algorithms: physics-constrained motion tracking, self-organizing map (SOM), and the feature tracking kit (FTK), are also explained in Sec. III. The three types of microparticle clouds are described in Sec. IV. Section V compares the results using different algorithms to process the image sets. Particle density and noise are found to be important factors that affect the effectiveness of the algorithms. An Appendix is also included to account for the fact that particle tracking and imaging are a rapidly expanding interdisciplinary field, and many tools are available.

II. MICROPARTICLE TRACKING FRAMEWORKS

A. Physics models

Digital cameras for particle imaging motivate discretized motion models,

$$\mathbf{r}_{n+1} = \mathbf{r}_n + \mathbf{v}_n \Delta t. \quad (1)$$

Here, the subscripts $n+1$ and n are for the consecutive time steps or video frame numbers with a time step Δt . $1/\Delta t$ is the constant frame

rate of the video camera. The bold face symbol $\mathbf{r}$ is for the instantaneous position of a particle in three-dimensional (3D) physical space in general. $\mathbf{v}_n$ is the instantaneous velocity of the particle at the time step n , which in general also varies as a function of time or frame number n ,

$$\mathbf{v}_{n+1} = \mathbf{v}_n + \mathbf{a}_n \Delta t. \quad (2)$$

For non-relativistic motion, $\mathbf{a}_n$ is given by Newton's equation,

$$M_n \mathbf{a}_n = \sum_i \mathbf{f}_n^i. \quad (3)$$

Here, M_n is the particle mass at the time step n and the summation on the RHS is over different forces. In a laboratory plasma, the sum may include gravity, neutral gas drag, ion drag force, electrostatic or electromagnetic force for a charged particle, etc. In the strongly coupled regime, electrostatic Coulomb interactions among neighboring particles also have to be included.⁷ In cases when the particle mass M_n varies with n , the RHS can also include a “rocket force” given by $\mathbf{v}_n dM_n/dt = \mathbf{v}_n (M_{n+1} - M_n)/\Delta t$. We mention without further elaboration that Eqs. (1)–(3) can also be extended to include additional degrees of freedom such as particle rotation or spin since a particle consists of millions or more atoms.

To learn about the plasma conditions or unknown physical properties such as the electric charge or a force on a particle through Eq. (3), it will require the information of $\mathbf{v}_n$ and $\mathbf{a}_n$ first through Eqs. (1) and (2). Therefore, it appears that a non-physical approach such as a data-driven method would be a prerequisite to derive $\mathbf{r}_n$ and $\mathbf{r}_{n+1}$. Meanwhile, even an initial estimate of particle velocities based on physics arguments such as particle kinetic energy, momentum conservation, or energy conservation would be useful to correctly pair up $\mathbf{r}_n$ with $\mathbf{r}_{n+1}$. One example will be included in Sec. III and Fig. 1. Another reason why a pure data-based method may not be the best option is related to the so-called NP-hard problems in computing. When an image contains many microparticles, correctly pairing of the microparticles from one image to another is not obvious, giving rise to the “particle linking” problem in tracking algorithms. For example, when processing a video with 100 particles per frame and 100 frames long, a random pairing algorithm would give rise to $\sim 100^{100} = 10^{200}$ possible tracks. Data association and track-to-track association, two fundamental problems in multi-target tracking, are instances of an NP-hard combinatorial optimization problem known as the multidimensional assignment problem (MDAP).⁸ Physics constraints will allow substantial reduction of computing time to correctly identify ~ 100 tracks. Physics consideration also motivates machine learning of “physical features” such as the particle size, particle brightness, particle velocity, angular momentum, etc. Another application of physics models would be to provide “ground truths” for data model training and validation.

In raw image data, $\mathbf{r}_n$ and $\mathbf{r}_{n+1}$ are not measured directly. An optical camera image is a 2D projection of a 3D physical scene. Such a projection is usually described by epigeometry,^{9–11} which relates 3D coordinates $\mathbf{r}$ (suppressing the subscript for now) in the physical space to at least two independent 2D projections or two pairs of camera coordinates $\mathbf{q}_+$ (u_+ , v_+) and $\mathbf{q}_-$ (u_- , v_-). A triangulation algorithm can recover $\mathbf{r}$ from $\mathbf{q}_\pm$. Therefore, in addition to linking of particles from different image frames in a single camera, another type of linking algorithm is required to link particles from a pair of cameras, which project the same physical scene from different positions and angles.

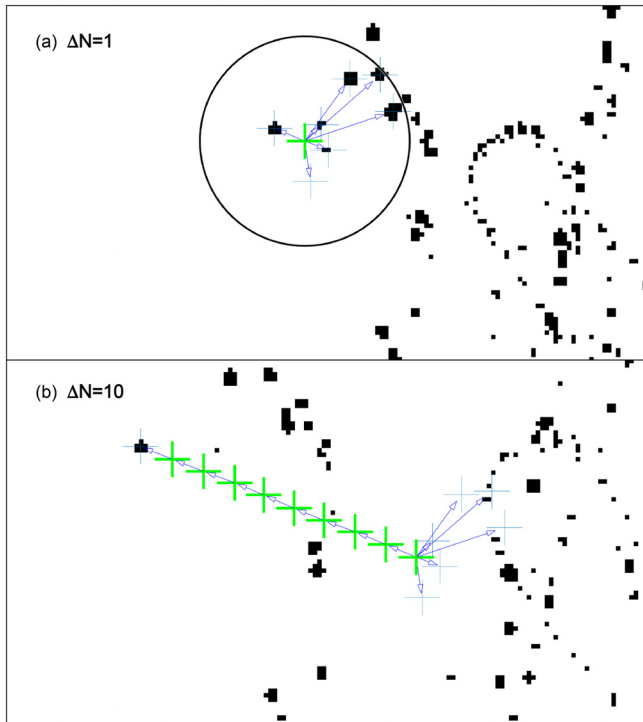


FIG. 1. Illustration of the LCV algorithm. (a) A particle from the initial time step $\Delta N = 0$ (colored “+” in the center of the circle) is looking for its counter part in $\Delta N = 1$. Within the search radius ($R_s = 20$ pixels), seven candidates are found, each gives new information about possible velocity vectors, shown in colored arrows. (b) At $\Delta N=10$, only one of the seven initial velocity vectors survives the consistency check. The particle trajectory is the longest chain of colored + linked by the colored arrows. The error in search is two pixels. The time unit Δt in Eq. (1) is normalized to 1.

Derivation of the positions $\mathbf{q}$ or (u, v) from an image is called *particle localization*. Subscripts $\pm$ are suppressed to avoid clutter. A particle image is typically spread over a cluster of neighboring pixels with an intensity distribution $I_j(\tilde{\mathbf{q}}_j)$. Isolation of the pixel cluster from the rest of the image is called *instance segmentation*. Once segmented, one common algorithm to locate $\mathbf{q}$ is through the centroid of the intensity cluster, $\mathbf{q} = \sum_j I_j \tilde{\mathbf{q}}_j / \sum_i I_i$. Due to the finite pixel size of a camera, the signal-to-noise ratio (SNR) of the particle image intensity, motion blur due to the finite camera exposure time, and particle image overlap in high particle densities, $\mathbf{q}$ can only be determined within a certain accuracy. The errors of $\mathbf{q}$ measurement can propagate down through the whole data processing chain that essentially limits the accuracy about a derived physical quantity such as the spatial coordinates and velocities.

B. Statistical models

In either a data-based approach or physics-driven approach, a central problem is about effectively dealing with noise or uncertainties,¹² which naturally motivates statistical models for positions $\mathbf{r}_n$ and other physical quantities as defined in Sec. II A. A statistical tracking model predicts the motion of an object such as its next position $\mathbf{r}_{n+1}$

as a probabilistic distribution function, $P(\mathbf{x}_{n+1} | \mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n)$. Here, $\mathbf{q}_i$ with $i = 1, \dots, n$ are the measured quantities as a function of time up to the n th step. One example of $\mathbf{q}_i$ is the image coordinates $\mathbf{q}_{\pm}$ as given above. Each $\mathbf{q}_i$ can be further framed as a function of $\mathbf{x}_i$ as explained in Eq. (5) below. $\mathbf{x}_{n+1}$ is a generalized state vector from the three-dimensional position $\mathbf{r}_{n+1}$ that may also include for example instantaneous velocity, acceleration, electric charge (Q); i.e., $\mathbf{x}_{n+1} = (\mathbf{r}_{n+1}, \mathbf{v}_{n+1}, \mathbf{a}_{n+1}, Q_{n+1}, \dots)$. Based on the probabilistic distribution function, and the new measurement $\mathbf{q}_{n+1}$, the optimal estimate for $\mathbf{x}_{n+1}$, or $\hat{\mathbf{x}}_{n+1}$ can be made. The new information through $\mathbf{q}_{n+1}$ also updates the probabilistic function to $P(\mathbf{x}_{n+2} | \mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n, \mathbf{q}_{n+1})$, which allows optimal prediction of the future state, $\hat{\mathbf{x}}_{n+2}$ with additional measurements $\mathbf{q}_{n+2}$ and so on.

One well-known statistical model with applications to particle tracking is the Kalman filter.^{13–16} Originated in the sixties for object tracking using radar, sonar, and electromagnetic techniques, recent surges in automated object recognition and tracking are motivated by computer vision and image processing applications such as robotics and self-driving cars.¹⁷ Kalman filter is an optimal recursive Bayesian filter for linear functions subjected to white Gaussian noise. Constant gain Kalman filter is computationally fast for single-object detection, tracking, and localization. The original Kalman filter has been extended in various ways. The extended Kalman filter (EKF) and unscented Kalman filter are nonlinear versions of the Kalman filter.^{18–20} Additional nonlinear, non-Gaussian filters and application examples can be found in the books^{16,21} and citations therein. Example applications of Kalman filters and variants related to microparticles in plasmas can be found in Refs. 7, 17, and 22.

A Kalman-filter statistical model modifies Eqs. (1)–(3) by a state equation,²³

$$\mathbf{x}_{n+1} = \mathbf{A}_n \mathbf{x}_n + \mathbf{f}_n + \epsilon_{x_n}, \quad (4)$$

and a measurement equation,

$$\mathbf{q}_n = \mathbf{B}_n \mathbf{x}_n + \epsilon_{q_n}, \quad (5)$$

at the time step n . The state equation describes the state evolution to the next step. The measurement equation relates the state $\mathbf{x}_n$ to the measurement $\mathbf{q}_n$. The term $\mathbf{f}_n$ stands for external control including an external force. ϵ_{x_n} and ϵ_{q_n} symbolize uncertainties or noise in the state evolution and errors in measurements, respectively. Both ϵ_{x_n} and ϵ_{q_n} have a zero mean and non-vanishing covariance. A common noise and error model is the white Gaussian noise model,^{14–16} when ϵ_{x_n} and ϵ_{q_n} have a Gaussian distribution with a zero mean.

Both physics models and statistical models rely on assumptions to make predictions. Good assumptions play multiple roles such as (a) simplifying the calculations or the reasoning process; (b) supplementing the incomplete knowledge about the physical systems; and (c) fitting the experimental data well. In practice, good and simplifying assumptions can be difficult to come by in complex plasmas and as a result, the predictive power of a physics or statistical model is limited. For example, in most physics or statistical frameworks for microparticle interaction with plasmas, microparticles are almost always assumed to be a perfect sphere, which reduces the number of geometrical parameters to one (particle radius). In another example, models for material properties rarely consider the surface morphology that could significantly affect the electron emissivity, electron scattering and trapping, and therefore the amount of electric charge on a microparticle

immersed in a plasma. Kinetic effects, non-equilibrium states of a plasma and especially near the sheath of a microparticle, are another example of when good physics and a statistical model can be difficult to develop. Simplifying assumptions is sometimes essential to avoid computational penalty associated with sophisticated physics or statistical models. The continuous decrease in computing cost opens door to more sophisticated physics and statistical models.

III. DATA MODELS

A data model may be characterized by how the data flows within the model (“model architecture”) and how the data adjust the values by the parameters chosen for the model (“model parameters”). A simple data model such as a linear regression only has a single channel that connects the input and output, and the data adjust their values by two parameters. A modern deep convolutional neural network (NN) may have billions or more parameters, corresponding to complex channel connectivity. The data that are used to find out or tune the parameters are called “training data.” New data can be used to validate or further tune the model parameters. Independent of the model complexity, the parameters may start with random initial values. Least squares fitting is a widely used procedure to tune the model with two or a few parameters. Backpropagation is a procedure developed for parameter tuning in feedforward neural networks.

Neural networks (NN) have shown to be effective for processing large image data sets,²⁴ solving differential equations, and simulation of physical systems.²⁵ NN are remarkable in recovering “empirical truth” when the physical framework for the data is unknown, and a generic approach with a large number of tunable parameters is used. In classification, the input data such as images need to be sorted into different categories (“dogs,” “cats,” “fish,” “cars”) as accurately as possible. Through parallel processing of different parts of an input image and passing the outputs in sync, it is possible to process a large-size image in real time quickly.²⁶ It has been shown that continuous functions can be approximated by feed-forward NN with a single hidden layer, which is the *universal approximation theorem*.^{27,28} Eldan and Shamir (2015) showed that, to approximate a specific function, a two-layer network requires an exponential number of neurons in the input dimension, while a three-layer network requires a polynomial number of neurons.²⁹ In using NN for stereo image analysis, for example, the state-of-the-art NN can recover millions of parameters out of a dataset. Learning polynomial functions with neural networks was described in Ref. 30. Another example is to learn about probability density $p(\mathbf{X})$ of a turbulent field for a given set of images. For a locally regular or continuous function $p(\mathbf{X})$ and error ϵ , an estimate $\hat{p}(\mathbf{X})$ that satisfies $\mathbb{E}(\|p - \hat{p}\|) \leq \epsilon$, the number of samples required is given by³¹ $n \geq k\epsilon^{-d}$. Since the dimension of a 1 mega-pixel image $d \sim 10^6$, the huge number of samples n required is also known as the curse of dimensionality. A school of thought is to find regularity properties (such as symmetry and scale separation), which can break the curse of dimensionality.

NN methods for particle tracking have been investigated by a growing number of authors.^{32,33} Different types of artificial NN have been used for tracking: feedforward NN (such as autoencoder and convolutional),³⁴ feedback NN, recurrent NN (such as Hopfield, long short-term memory, competitive, or self-organizing), radial basis function NN, and modular NN. Additional examples are given in Table I. We examine data models that determine the individual particle

coordinates, velocities and acceleration, or particle tracking velocimetry (PTV). A closely related class of approaches, particle image velocimetry (PIV), generally does not need to determine the particle coordinates explicitly and is usually used when the particle density is high and separation of individual particles is difficult. The algorithm needs to perform two basic functions: (1) particle localization; (2) particle matching or linking. Particle localization is to determine the coordinates of a particle. Particle matching or linking is to recognize the same particle at different times or from different camera views. A “particle” can be a macroscopic object such as a star,^{35,36} a car or a human, or a microscopic object such as a biological cell or an organelle inside the cell. Despite different physical origins, the images captured have similar features through the uses of telescopes, microscopes, and other hardware. The similarities in the images potentially allow data models developed for one type of context to be used for a different context.

In particle tracking, the input vectors $\mathbf{I}$ are the intensity maps or images. The output vectors $\mathbf{y}$ can vary. For particle localization, $\mathbf{y}$ are the coordinates of individual particles. Correspondingly, NN algorithm workflow could be as simple as $\mathbf{I} \rightarrow \mathbf{q} = \mathbf{y}$. Here, the output $\mathbf{y}$ is the raw camera coordinates $\mathbf{q}$, using the symbols defined in Sec. II. Additional steps or NN layers can be added, for example, $\mathbf{I} \rightarrow \mathbf{I}' \rightarrow \mathbf{q} \rightarrow \mathbf{r} = \mathbf{y}$. Here, $\mathbf{I}'$ stands for transformed raw images after denoising, smoothing, convolution, etc. In a recent example,⁴⁴ output is the probability of a particle centered at a pixel. For particle linking, $\mathbf{y}$ corresponds to linkage between the coordinates from different images. For example, if two pairs of coordinates $\mathbf{q}_1$ and $\mathbf{q}_2$ are linked, then $\mathbf{y}(\mathbf{q}_1, \mathbf{q}_2) = 1$; otherwise $\mathbf{y}(\mathbf{q}_1, \mathbf{q}_2) = 0$.

In general, the input and output can be described by an unknown function $f: \mathbf{I} \rightarrow \mathbf{y}$ and modeled by a neural network for a set of given data pairs $\{(\mathbf{I}, \mathbf{y})\}$. The building elements of a neural network are neurons, which are connected to other neurons.⁴⁵ Each neuron calculates a weighted sum of the outputs of neurons, which are fed to it (and usually adds a bias term, b), $z = \theta(\sum_i w_i x_i + b)$, here, θ is called the activation function. x_i and z are the individual neuron inputs and output. Some of the popular activation functions include rectified linear unit (ReLU), tanH, softmax. In a feed-forward neural network, the neurons are arranged in layers, and the outputs of each layer form the inputs to the next layer. Each layer may be interpreted as a transformation of its inputs to outputs. As the input data propagate through the layers, higher-level concepts or features emerge. The depth of a network is the number of layers and the size of the network is the total number of neurons. In a convolutional neural network (CNN), the weighted input sum to a neuron through multiplication $\sum_i w_i x_i = \mathbf{w} \cdot \mathbf{x}$ is replaced by a convolution operation $\otimes: \mathbf{w} \otimes \mathbf{x}$. The weight parameter $\mathbf{w}$ in a CNN is also called a kernel. Further discussions about algorithms are given in the Appendix.

Once the neuron linkage and activation functions within NN are chosen by design, the next step is to determine the values of the weights $\{\mathbf{w}\}$ and biases $\{b\}$ using a training dataset. Training is iterative starting with random values (within certain bounds) for $\{\mathbf{w}\}$ and $\{b\}$. A cost function such as L_2 norm $\|y_i - f_i\|_2 \equiv (y_i - f_i)^2$ can be obtained,⁴⁵ where y_i is the expected output and f_i is the corresponding value from NN. Iteration continues by adjusting $\{\mathbf{w}\}$ and biases $\{b\}$ to minimize the cost function until the desired value or error is reached. Gradient descent and backpropagation algorithms have been developed for the iteration.^{46–48} It has been shown that sophisticated

TABLE I. Examples of data-driven tracking algorithms and their applications. Illumination is assumed to be optical by default and otherwise specified. CC: cascade classifier. CNN: convolutional neural network. DM: diffusion maps. HF: Haar features. KNN: Kohonen neural network. LSTM: long short term memory. RNN: recurrent neural network. SOM: self-organizing map.

Instrument (image set)	Application	Parameter (feature)	Data model
(Simulations) ³³	Fluids	Position	KNN (SOM)
Microscope ³⁷	Cellular dynamics	Position	HF
Microscope ³⁸	Self-assembly	Cluster (distance)	DM
Video ³⁹	Surveillance	Object	CNN + RNN
Cryo-EM ⁴⁰	Macromolecules	Object	Deep CNN
Particle Detector ⁴¹	High-energy Physics	Track	LSTM + CNN
Holograms ⁴²	Colloidal science	3D position	CC, CNN
MNIST ⁴³	Computer science	Position cloud	SO-Net

NN such as deep CNN involving millions or more parameters $\{\mathbf{w}\}$ and $\{b\}$ are computationally hard to train.⁴⁹ In practice, recent NN are commonly trained using stochastic gradient descent (SGD) for expediency and a variety of tools are used that include a proper selection of activation functions (e.g., ReLU), over-specification (i.e., train networks which are larger than needed), and regularization.^{24,45,50}

A. Physics-constrained motion tracking

We first describe a motion-based linking algorithm using a local-constant-velocity (LCV) assumption. Modifying LCV to, for example, local constant acceleration (LCA) or periodic oscillatory motion (POM), is also possible and depends on the physics context. The essential function of the algorithm is to link particles from two different images, corresponding to particle positions at two different times. For stationary particles that do not move, the track reduces to a point. For a given particle, estimate of the particle velocity and the time lapse between the two images give the estimate of the search radius (R_s). Since the direction of the motion is unknown *a priori*, the candidate particles within the search radius now give the possible velocity vectors (both the magnitude and direction), as illustrated in Fig. 1(a). Third and additional images can be used to down-select the particles. In Fig. 1(a), seven candidate particles are found in the initial search circle. A third image is sufficient to settle down on the correct search, which is further confirmed by a sequence of six additional frames, Fig. 1(b).

Only a rough estimate of the search radius R_s is needed as illustrated in Fig. 2, when the search radius is doubled from Fig. 1. There are now 25 possible matches within R_s at $\Delta N = 1$. Only one particle is left at $\Delta N = 10$. In both examples shown here in Figs. 1 and 2, a nearest-neighbor algorithm would give the incorrect linking. The nearest-neighbor approach fails here because of large particle density (n_p), complicated further by the fact that the velocities of the particle motion (v_p) are sufficiently large so that

$$l_p \equiv v_p \Delta t > n_p^{-1/3}, \quad (6)$$

where Δt is the time step as defined in Eq. (1). A nearest-neighbor algorithm would work if $l_p < n_p^{-1/3}$.

B. KNN/SOM tracking

Similar to Refs. 32 and 33, we describe a parallel tracking method using the Kohonen neural network (KNN), also known as the self-organizing map (SOM).^{51,52} Here, the KNN consists of three layers: the input layer, the neuron layer, and the output layer. The N inputs are the individual particle 2D positions, $\{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_N\}$ from one of the two frames. The output layer is the possible matched particles from the other frame, $\{\mathbf{q}'_1, \mathbf{q}'_2, \dots, \mathbf{q}'_M\}$. $M \neq N$ in general. For a particle that occupies multiple pixels, the average position or the centroid that averages the pixel coordinates is used, similar to Sec. III A.

Each neuron ($\mathbf{z}_i$, $i = 1, 2, \dots, N$) assumes one of the N initiation positions $\{\mathbf{z}_i^{(1)} = \mathbf{q}_i, i = 1, 2, \dots, N\}$ to start. The subsequent positions at the $(k+1)$ th step evolve from the k th step through $\mathbf{z}_i^{(k+1)} = \mathbf{z}_i^{(k)} + \sum_{j=1}^M \mathbf{w}_{ij}^{(k)}$ with the weight $\mathbf{w}_{ij}^{(k)}$ given by³³

$$\mathbf{w}_{ij}^{(k)} = \alpha (\mathbf{q}'_j - \mathbf{z}_{ci}^{(k)}) H(R_s^{(k)} - |\mathbf{z}_i^{(k)} - \mathbf{z}_{ci}^{(k)}|), \quad (7)$$

where the summation is over the possible matches M in the other image frame. $0 < \alpha < 1$ is a constant and we use $\alpha = 0.1$ here. $\mathbf{z}_{ci}^{(k)}$ is

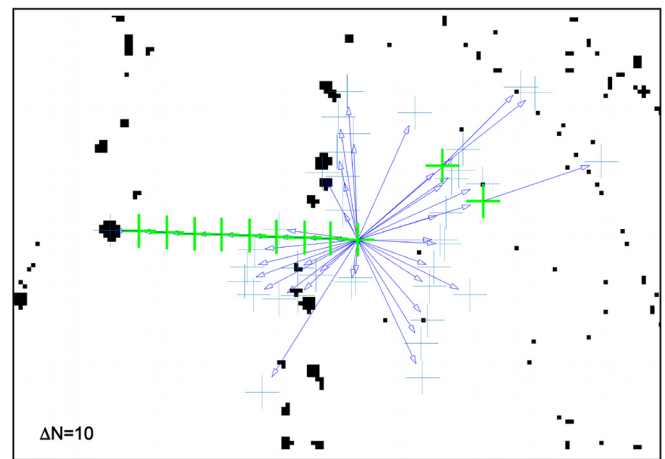


FIG. 2. Illustration of the LCV algorithm with twice the initial search radius ($R_s = 40$ pixels) as in Fig. 1, and with a higher particle density. Only one particle is left at $\Delta N = 10$.

the “winning” neuron position that is the closest to $\mathbf{q}_j^k$ at the step k . $H(x)$ is the Heaviside step function that satisfies $H(x) = 1$ for $x > 0$ and $H(x) = 0$ otherwise. $R_s^{(k)}$ is the search radius at the k th step. In the low density particle regime, the KNN reduces to the nearest neighbor algorithm.

When the particle density increases, the KNN is different from the nearest neighbor algorithm as illustrated in Fig. 3, where we select a case when two neighbors in the other frame are found within the initial search radius R_s^1 . The neuron starts to evolve toward the centroid of the two match candidates (P1 and P2) until the search radius R_s shrinks to the value when only one nearest neighbor (P1) is found. Then, it evolves along the straight line determined by the neuron position and the matched particle position. KNN/SOM differs from the nearest neighbor algorithm and the physics-constrained tracker by design. It is attractive to use KNN/SOM for parallel tracking of multiple particles. Meanwhile, its effectiveness varies significantly with the particle density for the datasets presented here.

C. The feature tracking kit (FTK)

The feature tracking kit (FTK)⁵³ is a general purpose library to track features in both simulation and experimental data in a scalable manner. The motivation of the FTK is to ease the burden of developing domain-specific algorithms to track features such as local extrema and superlevel sets. Basically, the FTK incorporates an ensemble of techniques including machine learning, statistical, and topological feature tracking algorithms to help scientists define, localize, and associate features over space and time. The FTK has a unique design to generalize existing 2D/3D feature descriptors to trace the trajectories of features in 3D/4D spacetime directly. The FTK also supports feature tracking in distributed and parallel machines.

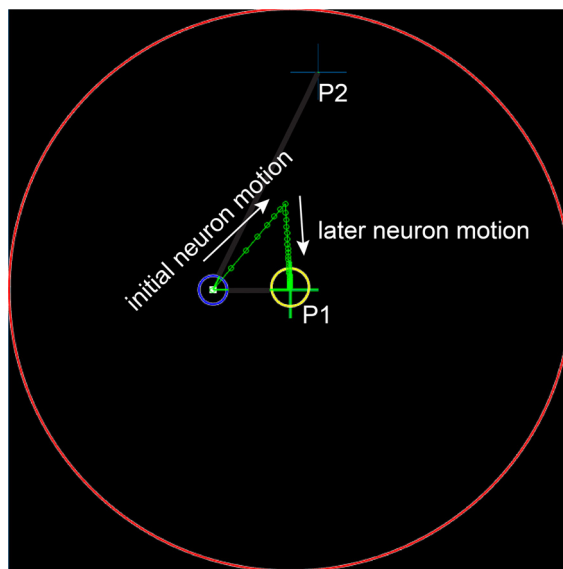


FIG. 3. Illustration of the KNN algorithm described by Eq. (7) when there are at least two particles (P1 and P2) falling within the initially search radius $R_s^{(1)}$. The neuron initially moves toward the weighted centroid between the two particles, until the search radius $R_s^{(k)}$ shrinks and only one particle is nearby. Then, the neuron is “attracted” to the object (P1) in a straight line.

Specific to the microparticle tracking application, we use the FTK to track local maxima in the time-varying imaging data. We first derive the image gradients and then localize maxima—locations where the gradient vanishes and the Jacobian is negative definite in the 3D spacetime mesh. Based on a scalable union-find data structure,⁵⁴ we associate these space-time maxima and construct their trajectories based on the connectivities of spacetime mesh elements. Examples of trajectories obtained from FTK are given in Fig. 9.

IV. EXPERIMENTAL IMAGE SETS

Here, we summarize three experimental video sets of microparticle clouds: in an exploding wire experiment, in a dusty plasma, and in an atmospheric plasma. The exploding wire dataset has the highest signal-to-noise ratio among the three. The dusty and atmospheric plasma datasets provide examples of rich motion patterns at an individual particle level as well as particle group level when particles are immersed in plasmas.

The exploding wire experimental setup and some analyses are reported in Ref. 11 and subsequent publications.^{55–57} Two new examples of microparticle clouds are shown in Fig. 4 at different particle densities.

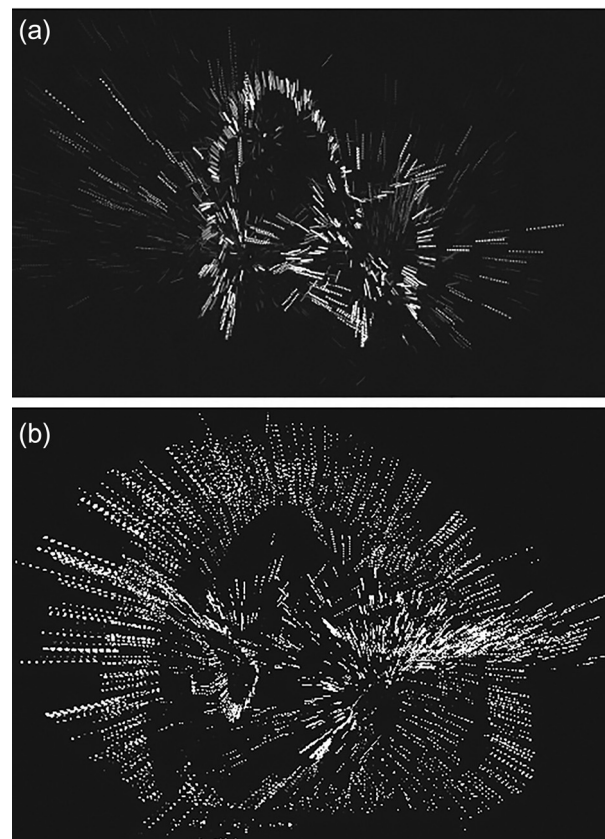


FIG. 4. Two examples of microparticle pattern created from exploding wires. Each panel is a superposition of 11 consecutive video frames taken at the rate of 25k fps. Frame (a) has about 1/10 particle density as in the dense regions in (b).

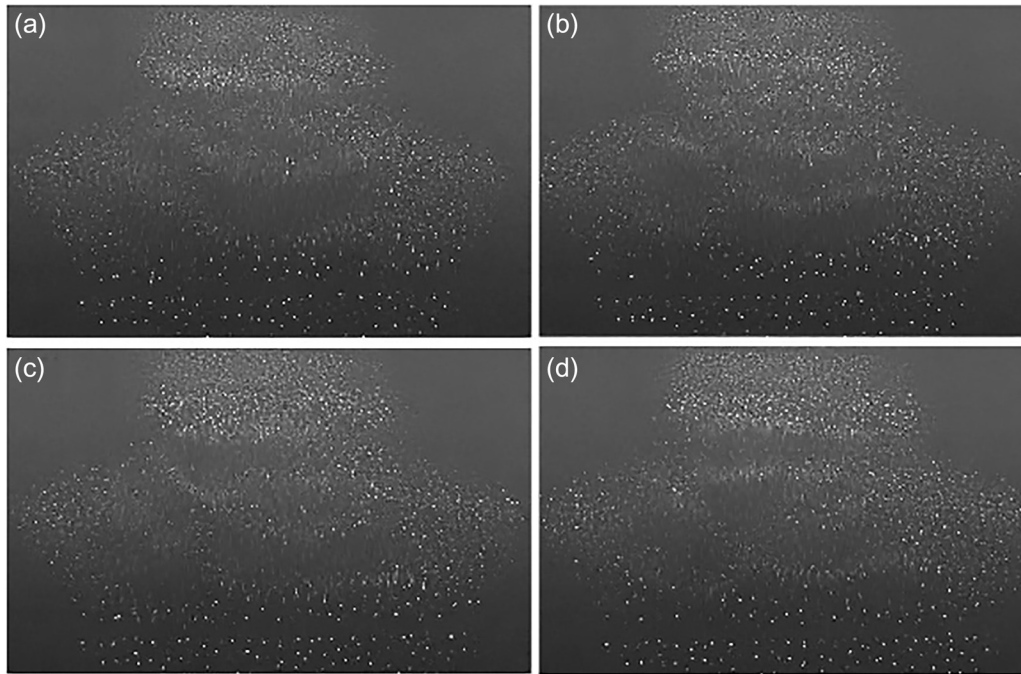


FIG. 5. A movie of dust particles in the MDPX device recorded at 12.5 frames per second. The experiment is performed using $2\ \mu\text{m}$ diameter silica particles in an argon plasma at a magnetic field of $B = 2\ \text{T}$. Motions of individual particles give rise to collective dust acoustic mode.

A. Dusty plasmas in a strong magnetic field

One example of a dusty plasma experiment in the Magnetized Dusty Plasma Experiment (MDPX) device at Auburn University is shown in Fig. 5. MDPX is a superconducting, multi-configuration, high magnetic field ($B_{\text{max}} \sim 4\ \text{Tesla}$) research instrument.^{58,59} A key feature of the MDPX design is that the cryostat has a central, cylindrical warm bore that has overall dimensions of: 50 cm inner diameter, 122 cm outer diameter, and an overall length of 157 cm. The cryostat is “split” into upper and lower halves that are 69 cm long with a gap of 19 cm between the two halves. Each half of the cryostat contains two superconducting coils. In combination, the four coils can be operated independently so that a variety of magnetic field configurations—from uniform to cusp-like—can be formed. For the experiments discussed in this paper, the uniform configuration is used.

B. Atmospheric plasmas

Atmospheric pressure plasmas (APPs) are non-equilibrium plasmas produced in the ambient atmosphere. These plasmas are currently being investigated for a number of applications ranging from wound healing to material processing to water purification.^{60–62} A particular class of these discharges is the 1 ATM DC glow with a liquid anode.⁶³ Here, the anode electrode is an electrolytic solution and ions complete the electric current flow. In such discharges, the anode attachment can self-organize into complex shapes observable at the liquid surface. These discharges produce copious amounts of nanoparticles in the liquid phase. Under certain conditions, particles and droplets derived from the liquid phase are injected into the plasma above. Upon

injection, metal nanoparticles are rapidly oxidized, appearing as luminous streaks in photographs, as shown in Fig. 6(a).

Dynamic motion of particle swarms can also be seen in Fig. 6 when the anode attaches to the plasma region above. A high-speed camera at 2k fps was used to video-record the emission process. As can be seen from the video, the particle emission coincides with the plasma attachment center at the liquid surface. Although the discharge is steady DC on average, the emission of particles appears as cyclic bursts. The bursts release copious amounts of fast moving particles, which appear as streaks due to the insufficient camera temporal resolution at 2k fps. Additional excitation is apparent as particles reach the core of the discharge as inferred from the over exposed glow region shown in frames Figs. 6(b) and 6(c). Following the burst release, the particles travel ballistically as inferred from the observed parabolic trajectories. Note that the streaks are longer during the emission phase (frames a and b) in comparison to later stages where the particles follow what appears to be simple rectilinear motion. This disparity in streak length suggests that the particles emitted are moving considerably faster upon launch. For non-viscous ballistic motion however, the particle velocity at the surface upon return should equal the launch speed, which suggests that at later times, the streaks should have lengths similar to the initially emitted particles. The absence of the long streaks at later times implies one or more of the following possible scenarios: (1) a distribution of particles of differing sizes and velocities is launched; (2) the smaller faster moving particles may move out of the camera focus and do not come near the emission zone; and (3) viscosity is not negligible as in the exploding wire experiment.^{55,56}

The mechanism for the particle injection into the gas phase and even the excitation processes along the trajectories are still not well

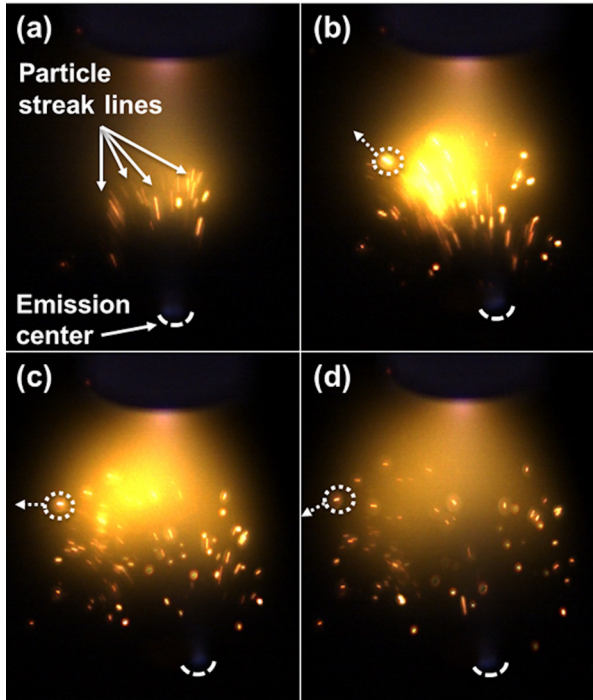


FIG. 6. Time-resolved particle emission with a high-speed camera at 2k fps: (a)–(d) $t = 2, 4, 6$, and 8 ms, respectively.

understood. By capturing the particle swarms with even higher speed cameras as in the exploding wire experiment, and through stereoscopic imaging from different vantage points, we plan to examine the kinematics of the particle generation and motion in further details, including applications of the physics, statistical, and data models described here.

V. ANALYSES AND DISCUSSION

Four algorithms for particle tracking are compared by using the exploding wire set: Trackpy (TP),⁶⁴ self-organizing map (SOM), FTK, and LCV, which is motivated by physics. All four algorithms worked well for low particle densities as in Fig. 4(a), achieving a tracking accuracy above 80%. At the particle density increased by about a factor of 10 higher, however, only LCV achieved the tracking accuracy above 75%. In implementation of each of the algorithms, the image analysis workflow can be divided into the following modules: image preprocessing (denoising), particle localization and characterization, and particle linking. The tracking accuracy is defined as the percentage of the particles being assigned to the different tracks with a track length longer than 30. The number of the expected tracks can be estimated with above 95% accuracy by tallying the number of particles in the video frame that contains the largest number of particles.

A. U-Net for particle localization in noisy images

Besides the particle density, another important factor that affects the tracking accuracy is the signal-to-noise ratio (SNR) of the images. The dusty plasma image set has a SNR less than 3, defined as the average particle intensity to the mean background pixel intensity, which is

much smaller than that of the exploding wire set ($\text{SNR} > 6$). We adopted U-Net for particle localization,^{65,66} a deep CNN (23 convolutional layer total) that can be trained end-to-end with very few images through data augmentation. Here, particle localization is equivalent to pixel classification. The U-Net architecture extends the fully convolutional network⁶⁷ and has a symmetric “U” shape. It consists of a contracting first half and an expanding second half.⁶⁵ The contracting half has the typical CNN architecture to capture context. The symmetric expanding half enables precise localization.

The U-Net algorithm here was trained using eighty images from the dusty plasma dataset. The training inputs were produced by extracting 256×256 patches from the frames and upsampling the patches to 512×512 . Ground truth segmentations were generated by thresholding the images based upon statistical significance above background pixels. In total, this procedure produced 1600 pairs of patches and segmentation maps. The parameters of U-Net were then optimized by using Adam,⁶⁸ which is an extension of stochastic gradient descent that has an adaptive learning rate. Figure 7 shows an example processed by the trained U-Net where the segmentation is used to mask the input image.

Another way to look at the workflow of U-Net algorithm is in the Fourier space. Here, discrete Fourier transform converts an image into a spatial frequency representation. This representation is complex valued and the modulus (amplitude) can be interpreted as the intensity of a particular frequency in the image. The transform is done efficiently using a fast Fourier Transform (FFT) algorithm.^{69,70} U-Net-generated masks allow an improved SNR for the dust acoustic mode (the lowest frequency band in the dusty plasma images), Figs. 8(a) and 8(b), as well as the particle localization (intermediate frequency band), Figs. 8(c) and 8(d). Wavelet analysis of U-Net outputs gives similar confirmation on SNR improvement.

B. Trajectory classification

Examples of reconstructed tracks are shown in Fig. 9 for the set of exploding wire data shown in Fig. 10. The combination of gravity and viscous force motivates a second-order polynomial fitting as discussed below.

The dusty plasma particle tracking based on U-Net is shown in Fig. 11. U-Net provides a labeling as to whether a pixel belongs to a particle or the background. Particle instances are then produced by finding connected components in the segmentation mask. In images, a connected component⁷² is the largest set of pixels such that they are the same value and there is a path between any two pixels. The components were found using 1-connectivity, meaning that paths do not include diagonals. Each connected component is treated as an individual particle. The particles are localized by averaging the pixel locations for pixels in each particle. Figure 11 shows tracks produced from this localization scheme and a nearest neighbor matching between frames.

The particle tracks from the dusty plasma can be checked against the following model $\Delta r^2 = 2ND\Delta t^k$; here, $N = 2$ is for two dimensions. Δr is the distance between the initial time and end time, which differ by Δt . $k = 1$ corresponds to the normal diffusion with D being the diffusion coefficient. However, most of the particle tracks cannot be described by normal diffusion. Some particles are observed to hop their position for a short period of time, similar to ballistic motion, followed by oscillatory motion that cannot be reduced to a simple mode. The result from analyzing the whole population of the particle

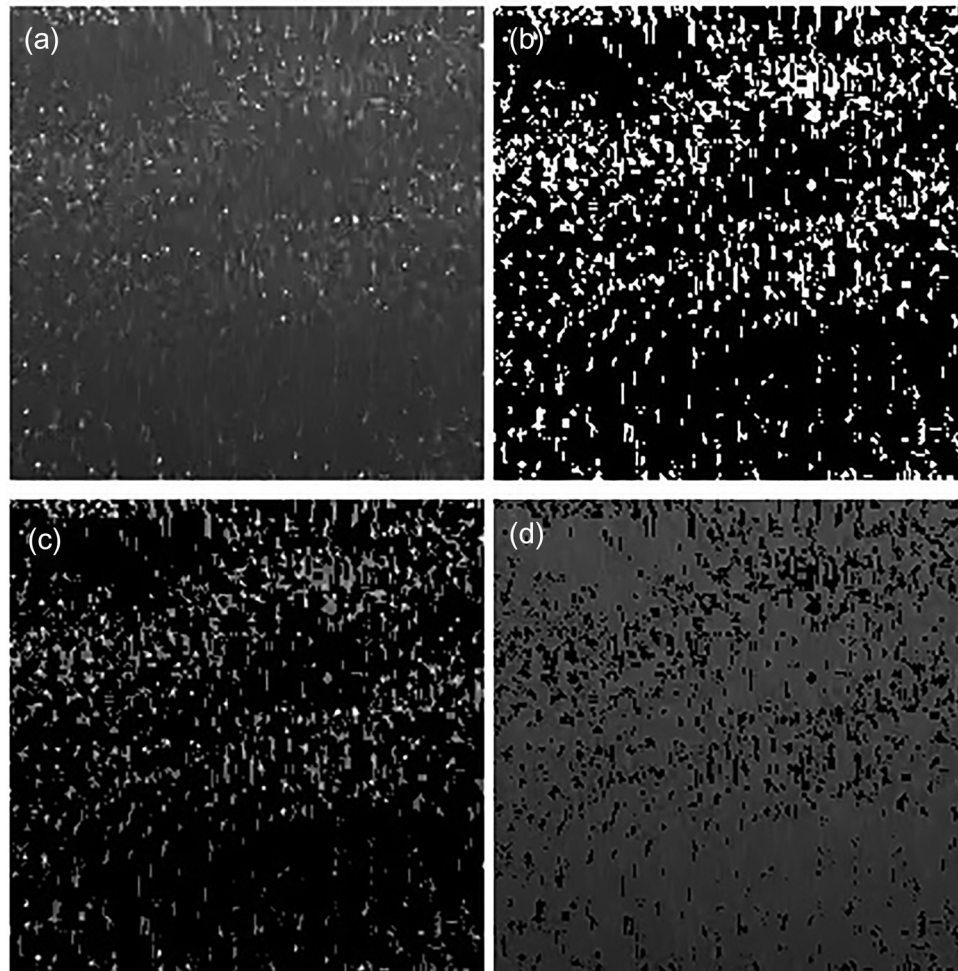


FIG. 7. Supervised noise reduction of the dusty plasma image set using U-Net. (a) An original image; (b). U-Net generated binary mask; (c) masked original image; (d) background after the subtraction of the masked image (c) from the original image (a).

trajectories is shown in Fig. 12, indicating that the “sub-diffusion” model with an exponent $k = 0.84 \pm 0.02$ gives a smaller root-mean square error (RMSE) than the normal diffusion model with $k = 1$. The observed subdiffusion is consistent with the strong coupling due to Coulombic interactions in dusty plasmas.⁷³

VI. SUMMARY AND PERSPECTIVE

We present new datasets of microparticle clouds and their interactions with laboratory plasmas: from exploding wires, in dusty plasmas in a strong magnetic field (MDPX), and in atmospheric plasmas. A physics-motivated local-constant-velocity (LCV) tracker, a Kohonen neural network (KNN) or self-organizing map (SOM), the feature tracking kit (FTK), and U-Net are described and compared with each other for particle tracking using the datasets. Particle density and the signal-to-noise ratio have been identified as two important factors that affect the tracking accuracy. Fast Fourier transform (FFT) is used to reveal how U-Net, a deep convolutional neural network (CNN) developed for non-plasma applications, improves the signal-

to-noise ratio. Viscous effects are revealed in the ballistic motions of the particles from the exploding wires and atmospheric plasmas. Subdiffusion of microparticles is obtained from the dusty plasma datasets, consistent with earlier work.

The methods for the experimental datasets can be extended to simulation data of corresponding microparticle experiments. As a matter of fact, coupling of simulation and experiment becomes straight-forward if the experimental and simulation data share the same format. A data model such as a neural network (NN) does not differentiate data origins. Some NN reported elsewhere were trained purely on simulations before their applications to experimental data. Therefore, data methods also provide a new approach to unify plasma theory, simulations, and experiments. Such a unifying approach may provide answers to some of the hardest yet elementary problems such as the charging dynamics of microparticles immersed in plasmas.

Here, we emphasize the microparticle cloud imaging and tracking (mCIT or μ CIT) for plasma science when more than a few micrometer-size particles are tracked simultaneously. Rapid recordings

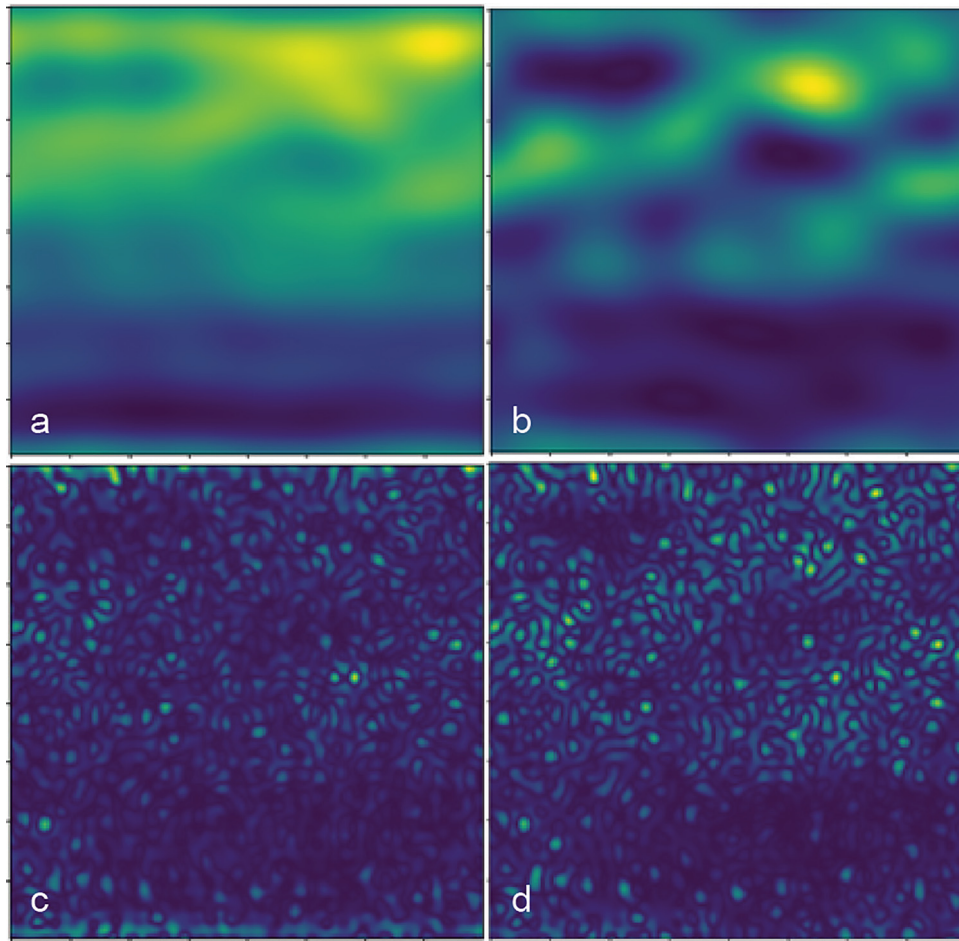


FIG. 8. Enhancement of the SNR using U-Net for different frequency bands: inverse FFT of the lowest frequencies of the original image (a), the lowest frequencies of U-Net masked image (b), intermediate frequency band of the original image (c), and the intermediate frequency band of the U-Net masked image (d).

of individual particle motion lead to temporally resolved local information. The whole particle cloud yields collective and global information. The data size increases with the number of particles, the number of cameras, camera sensor size, and camera frame rate. Terabyte of data per day can be readily achievable in today's plasma experiments. Traditional frameworks of data processing based on physics and statistical principles can now be significantly enhanced by data-driven methods, with the potential toward fully automated image processing and information extraction, doing away with *ad hoc* assumptions commonly employed in physics and statistical models in order to supplement incomplete information or understanding or to accelerate large scale simulations. Meanwhile, physical and statistical methods are indispensable such as by providing ground truths for data model training, simplifying or constraining data models, which could become non-deterministic polynomial-time (NP)-hard otherwise.

Automated knowledge discovery through massive data mining is new to plasma physics, boosted significantly by recent advances in data methods developed outside plasma physics and in high-throughput technologies such as fast cameras. In addition to large

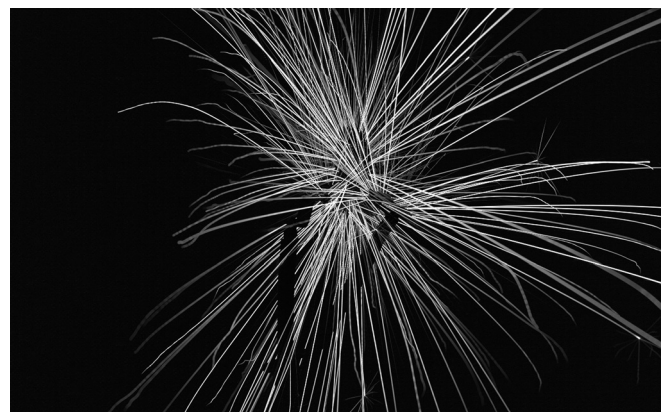


FIG. 9. Particle tracks recovered from the particle cloud scene using the same dataset as in Fig. 10. The tracks can be characterized as ballistic. Bifurcation of some tracks is due to secondary explosions as discussed previously.¹¹

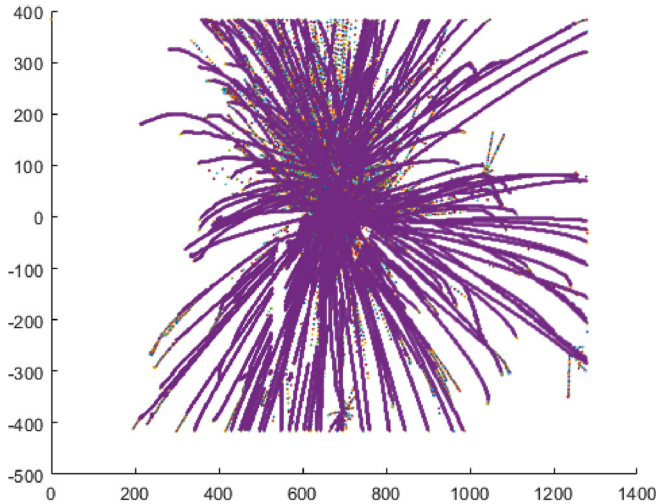


FIG. 10. After the background subtraction, possible track candidates from the raw data are obtained from ImageJ/Fiji⁷¹ by summation of the whole video sequence (~ 1300 frames).

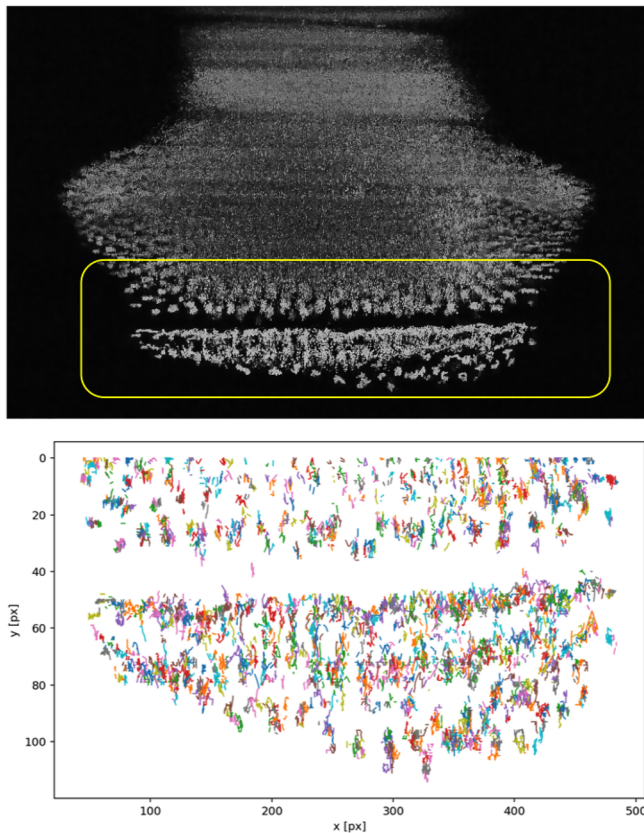


FIG. 11. Top: summation of 100 video frames from the dusty plasma movie shown in Fig. 5 after background subtraction using ImageJ/Fiji. Bottom: the particle tracks recovered using U-Net from the same region as bounded by the box in the Top. The tracks show a combination of ballistic and Brownian motion.

datasets of microparticle cloud interactions with plasmas, other TB plasma datasets also exist or can be readily obtained, giving rise to the “fourth paradigm” of scientific discovery and technology development, extending other approaches based on human intuition, fundamental laws of physics, statistics, and intense computation. Both experiments and simulations can benefit from data-driven discoveries and inventions. For example, data methods are being explored to accelerate high-definition simulations, which can be time-consuming even on the state-of-the-art exascale computers. Data methods are also being developed for real-time control of experiments, when the events occur too swiftly to make decision manually.

How much data are available from a plasma? The Bekenstein bound⁷⁴ for the information content (I_B) of a plasma sphere with a radius R is estimated to be $I_B \leq 2\pi RE/hc \ln 2 = 2\pi c R m / h \ln 2 \sim \alpha m R$ bits, with $\alpha = 2.577 \times 10^{43}$ bit/kg m. Consider the ion and electron mass alone and a 1-mm hydrogen plasma sphere with an electron density of 10^{15} m^{-3} , $I_B \leq 2.3 \times 10^7$ TB. Since most of the internal mass energy does not change during a low temperature plasma measurement, we may replace the energy mc^2 by $3/2 k_B (T_i + T_e) \sim 3 \text{ eV}$ for three degrees of freedom for each pair of hydrogen ions and electrons. I_B is then reduced to 17 kB, which may be interpreted as the lower bound in the information content as the plasma system reaches the thermal equilibrium. When the system is deviated from the thermal equilibrium, more bits of information are needed to describe the system. Ignoring the internal degrees of freedom such as excited states of an ion, rotation and vibration states of a molecular ion, or nuclear spin

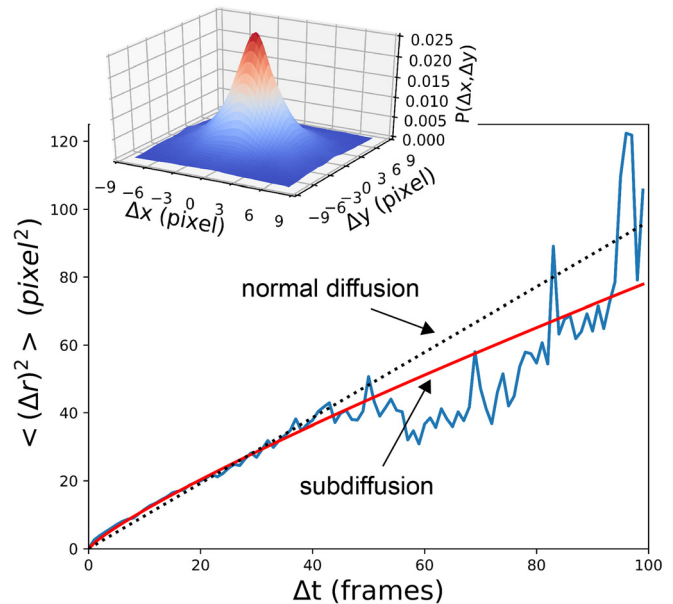


FIG. 12. Trajectory-averaged mean squared displacement with respect to time. The bell-shaped inset in the upper left corner is the probability density function of particle positions obtained from Gaussian kernel density estimation (KDE) of the dataset shown in Fig. 11. The full-width half maximum is (4.2 ± 0.2) pixels. The power-law fit over the range of $\Delta t = 10\text{--}40$ frames is $\langle \Delta r^2 \rangle = (1.6 \pm 0.1) \Delta t^{0.84 \pm 0.02}$ indicating subdiffusion. The linear fit (normal diffusion) over the same range is $\langle \Delta r^2 \rangle = (9.6 \pm 0.1) \times 10^{-1} \Delta t$. The RMSE between the models and the data is 1.12 (power) and 1.36 (linear), respectively. The power-law model produces an improvement around 17.6% in the RMSE over the normal diffusion model.

states for now, only the position and momentum for each ion and electron are needed to fully describe the system. Similar to the “classical” or Shannon information content,⁷⁵ an upper limit in information content is estimated to be on the order of $kN \log_2 N$ for N electron and ion pairs. The multiplier $k=12$ is for three spatial dimensions, three momentum dimensions, and two types of particles (electrons and ions).

In short, powered by data science and technology, further advances in microparticle tracking in plasmas can come in many flavors: from imaging of smaller and smaller microparticles, paving the way toward super-resolution that opens door to nanoparticle and ultimately atom and ion tracking in plasma science, to data algorithms, to automated plasma knowledge discovery, to data unification among experiments, simulations and theory, and to the ultimate information content of a plasma.

ACKNOWLEDGMENTS

The work at the Los Alamos National Laboratory (LANL) was supported through Triad National Security, LLC (“Triad”) Contract No. 89233218CNA000001 by the U.S. Department of Energy (DOE), Office of Science, Office of Fusion Energy Sciences (program manager: Dr. Nirmol Podder). Y.E.K. and J.E.F. were supported by the U.S. DOE under Grant No. DE-SC0018058. E.T. was funded by the National Science Foundation (NSF), the NSF EPSCoR program, the U.S. DOE, and the NASA Jet Propulsion Laboratory. This research used resources of the Magnetized Dusty Plasma Experiment (MDPX), which is a Collaborative Research Facility supported by the Office of Fusion Energy Sciences General Plasma Science program. MPDX was originally funded by the NSF-Major Research Instrumentation program. The feature tracking kit (FTK) development was supported by Laboratory Directed Research and Development (LDRD) funding from the Argonne National Laboratory, provided by the Director, Office of Science, of the U.S. Department of Energy under Contract No. DE-AC02-06CH11357.

APPENDIX: ADDITIONAL PARTICLE TRACKING RESOURCES

Particle or object tracking is widely used in astronomy, space science, biology, materials science, chemistry, and physics such as fluids and plasmas, and more recently in computer vision, autonomous driving, and surveillance. In spite of their contextual and spatial scale differences ranging from galactic sizes to nanometers, similar uses of imaging hardware, similar needs for image processing such as particle/object localization, particle linking from one time to another, particle linking from different camera views, denoising, etc., have led to open software and algorithms that can be used in multiple fields, making particle tracking a powerful and accessible technique for data-driven plasma science.

We include some additional tracking algorithms and software, intended mainly for beginners and users. One open resource is ImageJ/Fiji,⁷¹ which has multiple plug-in particle trackers such as TrackMate, MosaicSuite that includes Particle Tracker tool. OpenPTV is another open source that allows 3D particle velocity field measurement from multiple cameras.⁷⁶ Many research groups have developed application-specific trackers. A large fraction of

these use MATLAB (licensed) or Python (open) platform, and the codes are downloadable from GitHub or individual research group websites. Searchable examples include TracTrac, Lagrangian Particle Tracking. An IDL package⁷⁷ and a MATLAB package⁷⁸ have been developed for dusty plasmas. Reviews and comparative studies exist for cell and biology applications^{79,80} and fluid mechanics.²⁵ The early work on colloidal microscopy by Crocker and Grier led to many developments.⁸¹ For example, Trackpy is a Python implementation of the tracking algorithm originally developed by John Crocker and Eric Weeks in IDL.

REFERENCES

- M. Y. Pustynnik, M. A. Fink, V. Nosenko, T. Antonova, T. Hagl, H. M. Thomas, A. V. Zobnin, A. M. Lipaev, A. D. Usachev, V. I. Molotkov, O. F. Petrov, V. E. Fortov, C. Rau, C. Deysenroth, S. Albrecht, M. Kretschmer, M. H. Thoma, G. E. Morfill, R. Seurig, A. Stettner, V. A. Alyamovskaya, A. Orr, E. Kufner, E. G. Lavrenko, G. I. Padalka, E. O. Serova, A. M. Samokutyayev, and S. Christoforetti, *Rev. Sci. Instrum.* **87**, 093505 (2016).
- S. Jaiswal, M. Y. Pustynnik, S. Zhdanov, H. M. Thomas, A. M. Lipaev, A. D. Usachev, V. I. Molotkov, V. E. Fortov, M. H. Thoma, and O. V. Novitskii, *Phys. Plasmas* **25**, 083705 (2018).
- Z. Wang, C. M. Ticos, L. Dorf, and G. Wurden, *IEEE Trans. Plasma Sci.* **34**(2), 242 (2006).
- Z. Wang, C. M. Ticos, and G. Wurden, *Phys. Plasmas* **14**(10), 103701 (2007).
- Z. Wang, C. H. Skinner, G. L. Delzanno, S. I. Krashenninnikov, G. M. Lapenta, A. Y. Pigarov, P. K. Shukla, R. D. Smirnov, C. M. Ticos, and W. P. West, in *New Aspects of Plasma Physics: Proceedings of the 2007 ICTP Summer College on Plasma Physics*, edited by P. K. Shukla, L. Stanflo, and B. Eliasson (World Scientific, Singapore, 2008), pp. 394–475.
- S. I. Krashenninnikov, R. D. Smirnov, and D. L. Rudakov, *Plasma Phys. Controlled Fusion* **53**, 083001 (2011).
- N. P. Oxtoby, J. F. Ralph, C. Durniak, and D. Samsonov, *Phys. Plasmas* **19**, 013708 (2012).
- P. Emami, P. M. Pardalos, L. Elefteriadou, and S. Ranka, “Machine learning methods for solving assignment problems in multi-target tracking,” *arXiv:1802.06897v1* (2018).
- Z. Zhang, *Int. J. Comput. Vis.* **27**(2), 161 (1998).
- R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. (Cambridge University, Cambridge, USA, 2004).
- Z. Wang, Q. Liu, W. Waganaar, J. Fontanese, D. James, and T. Munsat, *Rev. Sci. Instrum.* **87**, 11D601 (2016).
- Y. Feng, J. Goree, and B. Liu, *Rev. Sci. Instrum.* **82**, 053707 (2011).
- R. Kalman, *J. Basic Eng.* **D82**, 35 (1960).
- A. H. Jazwinski, *Stochastic Processes and Filtering Theory* (Academic Press, New York, 1970).
- B. Anderson and J. Moore, *Optimal Filtering* (Prentice-Hall, Englewood Cliffs, NJ, 1979).
- E. Maggio and A. Cavallaro, *Video Tracking, Theory and Practice* (Wiley, United Kingdom, 2011).
- V. Hadziavdic, F. Melandso, and A. Hanssen, *Phys. Plasmas* **13**, 053504 (2006).
- E. A. Wan and R. van der Merwe, in *Proceedings of the IEEE 2000 AS-SPCC Symposium* (2000), p. 153.
- Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation* (Wiley, New York, 2001).
- J. F. Ralph, “Target tracking,” *Dynamics and control* (Wiley, New York, 2010).
- B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications* (Artech House, Boston, 2004).
- Y. Feng, J. Goree, Z. Haralson, C.-S. Wong, A. Kananovich, and W. Li, *J. Plasma Phys.* **82**, 615820303 (2016).
- D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*, 2nd ed. (Pearson Education, Inc., Upper Saddle River, NJ, USA, 2012).
- I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, 2016).
- S. Brunton, B. Noack, and P. Koumoutsakos, *arXiv:1905.11075* (2019).
- L. O. Chua and L. Yang, *IEEE Trans. Circuits Syst.* **35**, 1273 (1988).

- ²⁷G. Cybenko, *Math. Control Signals Syst.* **2**, 303 (1989).
- ²⁸K. Hornik, M. Stinchcombe, and H. White, *Neural Networks* **2**, 359 (1989).
- ²⁹E. Eldan and O. Shamir, [arXiv:1512.03965](https://arxiv.org/abs/1512.03965) (2015).
- ³⁰A. Andoni, R. Panigrahy, G. Valiant, and L. Zhang, in *ICML* (2014), p. 1.
- ³¹S. Mallat, "Learning physics with deep neural networks," Simons Foundation lecture, Oct. 17 2018.
- ³²I. Grant and X. Pan, *Exp. Fluids* **19**, 159 (1995).
- ³³G. Labonté, *Pattern Anal. Appl.* **3**, 267 (2000).
- ³⁴D. Held, S. Thrun, and S. Savarese, *Mon. Not. R. Astron. Soc.* **464**, 4463 (2017).
- ³⁵B. B. Spratling and D. Mortari, *Algorithms* **2**, 93 (2009).
- ³⁶E. J. Kim and R. J. Brunner, *Mon. Not. R. Astron. Soc.* **000**, 1 (2018).
- ³⁷S. Jiang, X. Zhou, T. Kirchhausen, and S. T. C. Wong, *Cytometry, Part A* **71A**, 563 (2007).
- ³⁸A. W. Long, J. Zhang, S. Granick, and A. L. Ferguson, *Soft Matter* **11**, 8141 (2015).
- ³⁹R. Stewart and M. Andriluka, "End-to-end people detection in crowded scenes," [arXiv:1506.04878v3](https://arxiv.org/abs/1506.04878v3) (2015).
- ⁴⁰Y. Zhu, Q. Ouyang, and Y. Mao, *BMC Bioinf.* **18**, 348 (2017).
- ⁴¹A. Tsaris, D. Anderson, J. Bendavid, P. Calafiura, G. Cerati, J. Esseiva, S. Farrell, L. Gray, K. Kapoor, J. Kowalkowski, M. Mudigonda, Prabhat, P. Spentzouris, M. Spiropoulou, J.-R. Vlimant, S. Zheng, and D. Zurawski, *J. Phys.: Conf. Ser.* **1085**(4), 042023 (2018).
- ⁴²M. D. Hannel, A. Abdulali, M. O'Brien, and D. G. Grier, *Opt. Express* **26**(12), 15221 (2018).
- ⁴³J. Li, B. M. Chen, and G. H. Lee, "So-net: Self-organizing network for point cloud analysis," [arXiv:1803.04249v4](https://arxiv.org/abs/1803.04249v4) (2018).
- ⁴⁴J. M. Newby, A. M. Schaefer, P. T. Lee, M. G. Forest, and S. K. Lai, *Proc. Natl. Acad. Sci.* **115**, 9026 (2018).
- ⁴⁵C. M. Bishop, *Pattern Recognition and Machine Learning* (Springer, New York, 2006).
- ⁴⁶P. J. Werbos, *The Roots of Backpropagation* (John Wiley & Sons, New York, USA, 1994).
- ⁴⁷D. E. Rumhart, G. E. Hinton, and R. J. Williams, in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*, edited by D. E. Rumhart and I. L. McClelland (MIT Press, Cambridge, MA, USA, 1986), Vol. 1, pp. 318–362.
- ⁴⁸D. E. Rumhart, G. E. Hinton, and R. J. Williams, *Nature* **323**, 533 (1986).
- ⁴⁹R. Livni, S. Shalev-Shwartz, and O. Shamir, in *Advance Neural Information Processing Systems* (2014), p. 855.
- ⁵⁰A. Ng, see <https://www.andrewng.org/courses/> for "Machine Learning Courses" (last accessed September 30, 2019).
- ⁵¹T. Kohonen, *Neurocomputing* **21**, 1 (1998).
- ⁵²T. Kohonen, *Self-Organizing Maps*, 3rd ed. (Springer Verlag, Berlin, 2001).
- ⁵³See <http://web.archive.org/web/20080207010024/http://www.808multimedia.com/winnt/kernel.htm> for "MS Windows NT Kernel Description" (last accessed September 30, 2019).
- ⁵⁴J. Xu, H. Guo, H.-W. Shen, M. Raj, X. Xu, X. Wang, Z. Wang, and T. Peterka, in *Proceedings of the 2020 IEEE Pacific Visualization Symposium* (2019).
- ⁵⁵P. Chu, B. T. Wolfe, and Z. Wang, *Rev. Sci. Instrum.* **89**(10), 10K101 (2018).
- ⁵⁶M. Szott, Z. Wang, and D. N. Ruzic, *Rev. Sci. Instrum.* **89**(10), 10K118 (2018).
- ⁵⁷B. T. Wolfe, O. Iaroshenko, P. Chu, G. T. Kenyon, T. J. Schaub, A. Thresher, Y. Watkins, L. Zhao, and Z. Wang, *Rev. Sci. Instrum.* **89**(10), 10K109 (2018).
- ⁵⁸E. Thomas, R. L. Merlino, and M. Rosenberg, *Plasma Phys. Controlled Fusion* **54**, 124034 (2012).
- ⁵⁹E. Thomas, U. Konopka, D. Artis, B. Lynch, S. Leblanc, S. Adams, R. L. Merlino, and M. Rosenberg, *J. Plasma Phys.* **81**, 345810206 (2015).
- ⁶⁰J. P. Trelles, *Plasma Sources Sci. Technol.* **27**, 093001 (2018).
- ⁶¹I. Adamovich, S. D. Baalrud, A. Bogaerts, P. J. Bruggeman, M. Cappelli, V. Colombo, U. Czarnetzki, U. Ebert, J. G. Eden, P. Favia, D. B. Graves, S. Hamaguchi, G. Hieftje, M. Hori, I. D. Kaganovich, U. Kortshagen, M. J. Kushner, N. J. Mason, S. Mazouffre, S. M. T. H.-R. Metelmann, A. Mizuno, E. Moreau, A. B. Murphy, B. A. Niemira, G. S. Oehrlein, Z. L. Petrovic, L. C. Pitchford, Y.-K. Pu, S. Rauf, O. Sakai, S. Samukawa, S. Starikovskaia, J. Tennyson, K. Terashima, M. M. Turner, M. C. M. van de Sanden, and A. Vardelle, *J. Phys. D* **50**, 323001 (2017).
- ⁶²J. E. Foster, *Phys. Plasmas* **24**, 055501 (2017).
- ⁶³Y. E. Kovach, M. C. Garcia, and J. E. Foster, *IEEE Trans. Plasma Sci.* **47**, 3214 (2019).
- ⁶⁴Trackpy Contributors, see <http://soft-matter.github.io/trackpy/v0.4.2/> for "Trackpy: Fast, Flexible Particle-Tracking Toolkit" (last accessed October 12, 2019).
- ⁶⁵O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," [arXiv:1505.04597](https://arxiv.org/abs/1505.04597) (2015).
- ⁶⁶Z. X. Hao, see <https://github.com/zhiyuhao/unet> for "UNET" (last accessed September 27, 2019).
- ⁶⁷J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," [arXiv:1411.4038](https://arxiv.org/abs/1411.4038) (2014).
- ⁶⁸D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014).
- ⁶⁹J. W. Cooley and J. W. Tukey, *Math. Comput.* **19**, 297 (1965).
- ⁷⁰P. N. Swartztrauber, in *Vectorizing the FFTs, in Parallel Computations*, edited by G. Rodrigue (Academic Press, New York, 1982), pp. 51–83.
- ⁷¹See <https://imagej.net/Fiji> for "ImageJ, Fiji" (last accessed October 29, 2019).
- ⁷²C. Fiorio and J. Gustedt, *Theor. Comput. Sci.* **154**, 165 (1996).
- ⁷³O. S. Vaulina and S. V. Vladimirov, *Phys. Plasmas* **9**, 835 (2002).
- ⁷⁴J. D. Bekenstein, *Phys. Rev. D* **23**, 287 (1981).
- ⁷⁵S. V. Krivovichev, *Miner. Mag* **77**(3), 275 (2013).
- ⁷⁶See <https://www.openptv.net/> for "Openptv" (last accessed October 29, 2019).
- ⁷⁷U. Konopka, "Wechselwirkungen geladener Staubteilchen in Hochfrequenzplasmen," Ph.D. thesis (Ruhr-Universität Bochum, 2000).
- ⁷⁸M. Himpel and A. Melzer, *J. Imaging* **5**, 28 (2019).
- ⁷⁹K. Jaqaman, D. Loerke, M. Mettlen, H. Kuwata, S. Grinstein, S. L. Schmid, and G. Danuser, *Nat. Methods* **5**(8), 695 (2008).
- ⁸⁰N. Chenouard, I. Smal, F. de Chaumont, M. Maska, I. F. Sbalzarini, Y. Gong, J. Cardinale, C. Carthel, S. Coraluppi, M. Winter, A. R. Cohen, W. J. Godinez, K. Rohr, Y. Kalaidzidis, L. Liang, J. Duncan, H. Shen, Y. Xu, K. E. G. Magnusson, H. M. B. J. Jaldén, P. Paul-Gilloteaux, P. Roudot, C. Kervrann, F. Waharte, J.-Y. Tinevez, S. L. Shorte, J. Willemsse, K. Celler, G. P. van Wezel, H.-W. Dan, Y.-S. Tsai, C. O. de Solórzano, J.-C. Olivo-Marin, and E. Meijering, *Nat. Methods* **11**(3), 281 (2014).
- ⁸¹J. C. Crocker and D. G. Grier, *J. Colloid Interfaces Sci.* **179**, 298 (1996).