J. Fluid Mech. (2025), *vol.* 1018, R1, doi:10.1017/jfm.2025.10502



Inferring activity from the flow field around active colloidal particles using deep learning

Aditya Mohapatra[1], **Aditya Kumar**[1], **Mayurakshi Deb**[1], **Siddharth Dhomkar**[1] and **Rajesh Singh**[1]

[1]Department of Physics, IIT Madras, Chennai 600036, India

Corresponding authors: Rajesh Singh, rsingh@physics.iitm.ac.in; Aditya Mohapatra, adityamohapatra217@gmail.com

(Received 15 May 2025; revised 21 July 2025; accepted 25 July 2025)

Active colloidal particles create flow around them due to non-equilibrium processes on their surfaces. In this paper, we infer the activity of such colloidal particles from the flow field created by them via deep learning. We first explain our method for one active particle, inferring the $2s$ mode (or the stresslet) and the $3t$ mode (or the source dipole) from the flow field data, along with the position and orientation of the particle. We then apply the method to a system of many active particles. We find excellent agreements between the predictions and the true values of activity. Our method presents a principled way to predict arbitrary activity from the flow field created by active particles.

Key words: active matter, boundary integral methods, machine learning

1. Introduction

Active colloidal particles – such as micro-organisms (Brennen & Winet 1977; Goldstein 2015) and synthetic microswimmers (Ebbens & Howse 2010; Thutupalli *et al.* 2018) – are known to organise themselves in emergent structures via fluid-mediated interactions. An active particle can create flow around it even if it is not moving due to non-equilibrium processes on its surface, which could be of biological origin (Brennen & Winet 1977) or of phoretic origin (Ebbens & Howse 2010). Classifying the irreducible components of flow produced by active particles provides a handle through which we can tune inter-agent interactions to design higher order (hierarchical) assemblies of active particles which display a desired self-organisation and tuneable emergent dynamics.

A topic of recent interest is the application of tools from machine learning to bear upon questions of inference and design principles in active matter systems (Cichos *et al.* 2020; Tsang *et al.* 2020; Supekar *et al.* 2023; Boffi & Vanden-Eijnden 2024). Convolutional neural networks (CNNs) have been used for performing particle image velocimetry

© The Author(s), 2025. Published by Cambridge University Press.

https://doi.org/10.1017/jfm.2025.10502 Published online by Cambridge University Press

(Rabault *et al.* 2017), particle tracking (Newby *et al.* 2018) and feature extraction (Hannel *et al.* 2018). To the best of our knowledge, a work on estimating activity from the flow field around active particles has not been reported. To this end, we use deep-learning techniques given fluid flow around particles to infer the irreducible components of activity.

The remainder of the paper is organised as follows. In § 2, we describe a method to compute irreducible components of flow around active particles. The methodology utilising deep learning to learn activity from fluid flow around active particles is given in § 3. We apply the method to construct irreducible components of flow around an active particle in § 4. We extend this method to infer flow around many particles in § 5. Finally, we conclude in § 6 by summarising our main results and suggesting directions for future investigations.

2. Irreducible components of flow around active particles

In this section, we consider the problem of computing irreducible components of flow around active colloidal particles in a Stokesian fluid of viscosity η. At low Reynolds number, as appropriate to active colloidal particles, the fluid flow $\boldsymbol{v}$ satisfies the Stokes equation (Lauga 2020)

$$-\boldsymbol{\nabla} p + \eta \nabla^2 \boldsymbol{v} = -\boldsymbol{f}, \qquad \boldsymbol{\nabla} \cdot \boldsymbol{v} = 0. \tag{2.1}$$

Here p is the fluid pressure and $\boldsymbol{f}$ is the force density. In Appendix A, we show that the flow around active particles can be written in terms of irreducible components $\boldsymbol{v}^{l\sigma}(\boldsymbol{k})$. The sum of the irreducible components in the Fourier space is

$$\boldsymbol{v}^{M}(\boldsymbol{k}) = \sum_{l=1}^{\infty} \sum_{\sigma \in \{s,a,t\}} \boldsymbol{v}^{l\sigma}(\boldsymbol{k}). \tag{2.2}$$

As described in the Appendix A, the flow $\boldsymbol{v}^{l\sigma}(\boldsymbol{r})$ decays as r^{-l} in an unbounded three-dimensional flow. Here σ can take three values for each l: $\sigma = s$, $\sigma = a$ and $\sigma = t$, which correspond to the three irreducible modes allowed for each l. See Appendix A for details. The terms $\boldsymbol{v}^{1s}$ is the flow due to a net external force, while $\boldsymbol{v}^{2a}$ is the flow due to a net external torque. Since an active particle has no external force or torque on it, the two leading irreducible components of exterior flow created by active particles are

$$\boldsymbol{v}^{2s}(\boldsymbol{k}) = \boldsymbol{G}(\boldsymbol{k}) \cdot \boldsymbol{f}^{2s}, \qquad \boldsymbol{f}^{2s} = i\boldsymbol{k} \cdot \left[M \boldsymbol{Q}^{(2s)}\right], \tag{2.3}$$

$$\boldsymbol{v}^{3t}(\boldsymbol{k}) = \boldsymbol{G}(\boldsymbol{k}) \cdot \boldsymbol{f}^{3t}, \qquad \boldsymbol{f}^{3t} = -k^2 M \ \boldsymbol{Q}^{(3t)}. \tag{2.4}$$

The two irreducible components of flow – the $2s$ mode (or the stresslet (Lauga 2020)) and the $3t$ mode (or the source dipole (Lauga 2020)) – is known to fit flow around a wide range of active particles (Drescher *et al.* 2010, 2011; Guasto, Johnson & Gollub 2010; Ghose & Adhikari 2014; Delmotte *et al.* 2015; Goldstein 2015; Thutupalli *et al.* 2018; Lauga 2020; Krishnamurthy & Prakash 2023; Marmol, Gachon & Faivre 2024; Baruah *et al.* 2025). The leading fluid flow due to force-free and torque-free active particles is due to the two modes $\boldsymbol{Q}^{(2s)}$ and $\boldsymbol{Q}^{(3t)}$. In addition, we can use uniaxial parameterisation of the single-layer density, which is true for a wide range of micro-organisms having symmetry about a given axis. For such a scenario, with $\boldsymbol{p}$ being the orientation of the particle, we choose

$$\boldsymbol{Q}^{(2s)} = s_0 \left(\boldsymbol{p}\boldsymbol{p} - \frac{1}{n_d} \boldsymbol{I} \right), \qquad \boldsymbol{Q}^{(3t)} = d_0 \, \boldsymbol{p}. \tag{2.5}$$

Here n_d is the space dimension and $\boldsymbol{I}$ is the identity tensor, while s_0 and d_0 are constants. We note that the constants s_0 and d_0 are related (Singh, Ghose & Adhikari 2015) to the

https://doi.org/10.1017/jfm.2025.10502 Published online by Cambridge University Press

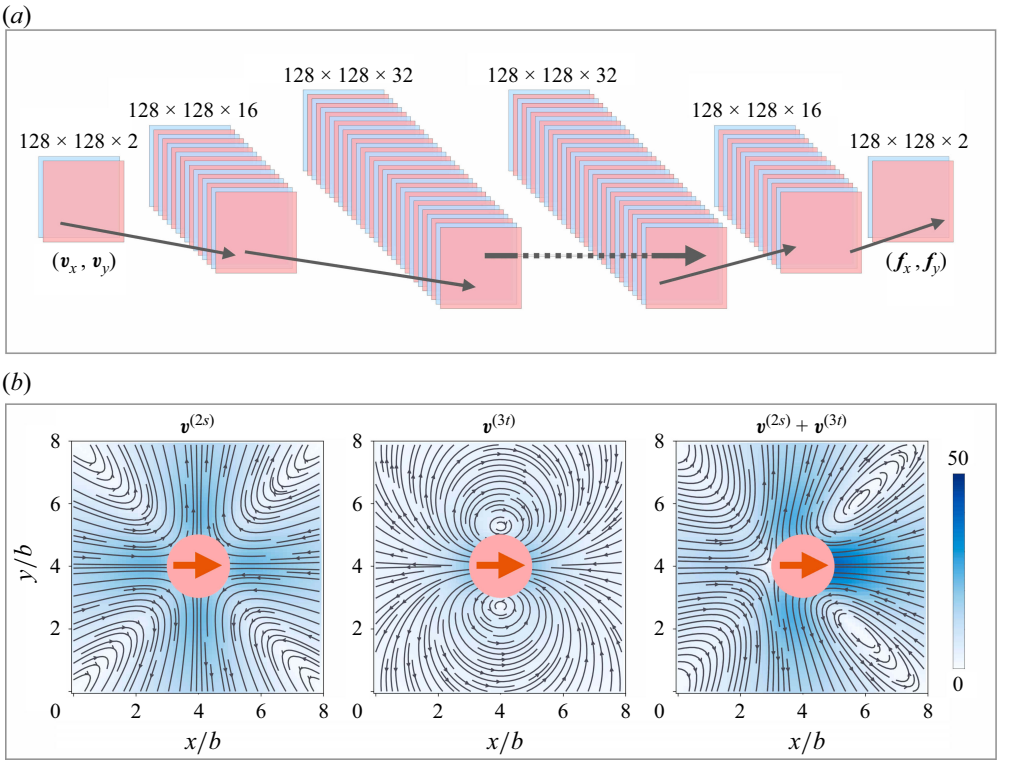


Figure 1. (*a*) The neural network architecture for computing force field from input velocity field. Each layer is a convolution layer with 3×3 kernel size and padding to preserve the dimension. Each block represents the image output of that layer in terms of height $\times$ width $\times$ channels. The network goes till 128 channels (indicated by the dotted arrow). (*b*) Flow due to $2s$ and $3t$ modes of the active slip. A combined flow is also shown. Position $\boldsymbol{c}$ is shown by a circle, while the orientation $\boldsymbol{p}$ is shown by a solid arrow. The flow streamlines have been overlaid on a pseudo colour plot of the speed.

B_1 and B_2 modes of the squirmer model (Lighthill 1952; Blake 1971; Pedley 2016; Lauga 2020). The flow field due to these two modes is present in figure 1.

For reasons explained, we truncate the expression of the fluid flow in (2.2) to the leading modes $2s$ (stresslet) and $3t$ (source dipole). For this choice, the expression for the net force density in the fluid, defined in (2.1), is written as the sum

$$\boldsymbol{f}(\boldsymbol{r}) = \boldsymbol{f}^{2s} + \boldsymbol{f}^{3t} = -\nabla M(\boldsymbol{r}) \cdot \boldsymbol{Q}^{(2s)} - \nabla^2 M(\boldsymbol{r})\, \boldsymbol{Q}^{(3t)}. \tag{2.6}$$

Here, M is a Gaussian mollifier whose mean is at the centre of the particle and the variance equals $2b^2/\pi$ (Maxey & Patel 2001), where b is the radius of the particle. In the next section, we present a deep-learning method to infer the individual components of active force density (and also the positions and the orientations of the particles) given fluid-flow data from the flow field around single and many active particles. See Appendix A for further details.

3. Methodology

Finding the velocity field solution to the Stokes equation with a given force density is discussed in the previous section. In addition, the velocity field may also be obtained from experimental systems using methods such as particle image velocimetry (PIV) (Adrian *et al.* 1991; Willert & Gharib 1991). On the other hand, the inverse problem of finding the

https://doi.org/10.1017/jfm.2025.10502 Published online by Cambridge University Press

force density (and its irreducible components) from given flow-field data is non-trivial. This leads us to a neural-network-based learning approach for achieving the force density and corresponding irreducible modes using the universal function approximator capacity of neural networks (Hornik, Stinchcombe & White 1989).

To effectively approximate the inverse function using deep learning, we begin by constructing a sufficiently large and diverse dataset. This dataset is partitioned into training, validation and test subsets. The model is trained on the training set to learn the mapping from velocity to force density. Throughout training, the validation set is used to monitor accuracy. Further, the model is evaluated on the independent test set to verify the model's performance on unseen data. We consider flow data with both a fixed and variable number of particles. For a fixed number of particles, we consider a single particle in a grid and four particles in a grid, which can be generalised to N particles. For a variable number of particles, we choose N to vary from one to four. We generate datasets of 4000 realisations corresponding to systems with different particle counts on a 128×128 domain. The grid is divided into equal subregions, and one particle is randomly placed within each to ensure non-overlapping configurations in the case of more than one particle.

The ith particle, whose centre is located at $\boldsymbol{c}^{(i)} = (x^{(i)}, y^{(i)})$, is assigned an orientation vector $\boldsymbol{p}^{(i)} = (\cos\theta^{(i)}, \sin\theta^{(i)})$, where $\theta^{(i)}$ is an angle made by the orientation vector with the axis of Cartesian coordinates $\hat{x}$-axis. The angle $\theta^{(i)}$ is selected from a discretised set of angles with 1° resolution. Discretisation of orientation space facilitates improved learning by the neural network. Each particle is assigned randomised stresslet strength $s_0 \in [800, 1200]$ and source dipole strength $d_0 \in [12\,000, 16\,000]$. The data are partitioned into training, validation and test sets comprising 2800, 600 and 600 samples, respectively.

The neural network used in this work is a fully convolutional architecture designed to approximate the inverse problem of (2.1). The input is a two-channel (x and y components of the fluid velocity $\boldsymbol{v}$) tensor $\boldsymbol{v} \in \mathbb{R}^{H\times W\times 2}$, representing the force field on a two-dimensional grid of height H and width W. The network output is $\tilde{\boldsymbol{f}} \in \mathbb{R}^{H\times W\times 2}$, which approximates the flow field for the given velocity field. Each convolutional layer (l) applies a localised linear transformation followed by a nonlinear activation function (Goodfellow *et al.* 2016). The network consists of a composition of such layers as shown in figure 1. The operation at layer l is given by

$$z^{(l+1)} = g\big(W^{(l)} * z^{(l)} + b^{(l)}\big) = h^{(l)}\big(z^{(l)}\big), \qquad g(z) = \max(0, z), \tag{3.1}$$

Here, C_l is the depth of lth convolutional kernel. Where $z^{(l)} \in \mathbb{R}^{H\times W\times C_l}$ is the input to the layer, $W^{(l)} \in \mathbb{R}^{C_{l+1}\times C_l\times k\times k}$ is a learnable convolutional kernel of size $k \times k$, and $b^{(l)} \in \mathbb{R}^{C_{l+1}}$ is a bias vector. The symbol $*$ denotes the two-dimensional convolution operation, and g is a nonlinear activation, ReLU, defined in (3.1). Here ReLU is preferred over the sigmoid activation function because it helps prevent the saturation of artificial neurons (Glorot *et al.* 2011). To improve generalisation and reduce overfitting, dropout is applied after each activation with a dropout probability of 0.2. The neural network is trained with mean squared error (MSE) as the loss function and Adam (Kingma & Ba 2014) as the optimiser.

We note that our inference methodology is only applicable for low-Reynolds-number scenarios when the Stokes equation is a valid description of flow around particles. For active colloidal particles – such as micro-organisms (Goldstein 2015) and synthetic microswimmers (Ebbens & Howse 2010) – the Reynolds number is vanishingly small and fluid flow is indeed Stokesian (Lauga 2020). Crucially, the existence of a valid Stokes flow implies that a one-to-one relation exists between the force density and fluid velocity

https://doi.org/10.1017/jfm.2025.10502 Published online by Cambridge University Press

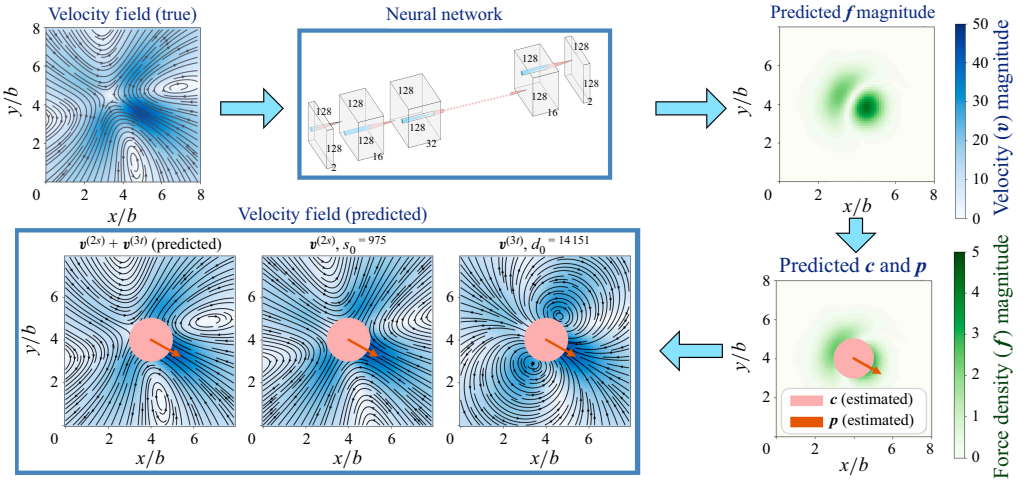


Figure 2. Flowchart of the method for the predictions of the model for a single particle: the velocity field for inference is shown along with a schematic of the neural network. The neural network predicts the force density field (here only magnitude is shown for clarity) from input velocity field. The velocity field obtained from predicted force is reconstructed with good accuracy. See Algorithm 1 for details of the method. The flow streamlines have been overlaid on a pseudo colour plot of the speed.

(Lauga 2020). Thus, the inverse problem of obtaining the force density from a given Stokesian flow is well-posed for active colloidal particles. Indeed, we show that the CNN effectively learns a direct field-to-field mapping to obtain force density from flow field data for both the single-particle and multiple-particle cases accurately. The method also works for data having a variable number of particles. We can also infer about the contribution of stresslet and source dipole in the flow field using a new algorithm, as we describe later, that works for an arbitrary number of particles given a predicted force density from the neural network.

4. Inferring one-particle flow

Active and self-propelled particles in the absence of applied forces or torques, with uniaxial parameterisation, can be characterised by the flow generated. The stresslet $\boldsymbol{Q}^{(2s)}$ generates flows that neither translate nor rotate the particle. The translational effect is generated by the source dipole $\boldsymbol{Q}^{(3t)}$. The sign of stresslet strength s_0 with respect to source dipole strength d_0 determines how the active particle acts. An active particle behaves as 'puller', pulling the fluid inward along the swimming direction and forcing fluid away from their sides, if $s_0/d_0 > 0$. The swimmer is said to be a 'pusher', pushing fluid away along their swimming direction and drawing fluid into the sides of their bodies, if $s_0/d_0 < 0$ Lauga (2020).

We consider the case of a single particle in a 128×128 grid, which is a slice from a large grid with periodic boundaries. The velocity field of the grid is fed into the neural network as input. The network computes the force density $\tilde{\boldsymbol{f}} = (\tilde{f}_x, \tilde{f}_y)$ by approximating the inverse problem accurately as shown in figure 2. The orientation $(\boldsymbol{p})$ of the particle is computed by taking local average of predicted force density in the neighbourhood where the magnitude of predicted force density is maximum. The position $(\boldsymbol{c})$ of the particle, strength of stresslet (s_0) and the source dipole (d_0) are estimated by minimising the squared error of true force density and estimated force density with gradient descent. The algorithm for estimating these parameters is given in Algorithm 1.

https://doi.org/10.1017/jfm.2025.10502 Published online by Cambridge University Press

Algorithm 1: Particle detection and parameter estimation for N particles

Require: Force density $\tilde{f}_x$, $\tilde{f}_y$; threshold ratio τ; particle size b; minimum distance d_{min}

Compute force density magnitude: $\tilde{f}_{mag}(x, y) \leftarrow \sqrt{\tilde{f}_x^2(x, y) + \tilde{f}_y^2(x, y)}$

Obtain global maximum: $\tilde{f}^*_{mag} \leftarrow \max_{x,y} \tilde{f}_{mag}(x, y)$

Detect peaks:

$$\mathcal{P} \leftarrow \{(x, y) : (x, y) \notin \boldsymbol{c}^{(i)} \pm d_{min} \text{ and } \tilde{f}_{mag}(x, y) > \tau \cdot \tilde{f}^*_{mag}\}$$

Detect number of particles $N \leftarrow ||\mathcal{P}||$ and compute centre of mass $\{\boldsymbol{c}^{(i)}\}_{i=1}^N$ for each region of $2d_{min} \times 2d_{min}$ centred around peak.

for each $\boldsymbol{c}^{(i)}$ **do**

  Initialise $s_0^{(i)}$, $d_0^{(i)}$

  Estimate local direction: $\boldsymbol{p}^{(i)} \leftarrow \left(\tilde{\bar{f}}_x, \tilde{\bar{f}}_y\right)$ in neighborhood of $\boldsymbol{c}^{(i)}$

  Initialise best loss: $\mathcal{L}_{min} \leftarrow \infty$

  for $\boldsymbol{c}^{(i)} \leftarrow \boldsymbol{c}^{(i)} + \Delta \boldsymbol{c}$ **do**

    Define estimated force field using Gaussian envelope M:

$$\hat{\boldsymbol{f}} = s_0^{(i)} \nabla M(\boldsymbol{c}^{(i)}) \cdot \left(\boldsymbol{p}^{(i)} \boldsymbol{p}^{(i)} - \frac{\boldsymbol{I}}{2}\right) - d_0^{(i)} \nabla^2 M(\boldsymbol{c}^{(i)}) \boldsymbol{p}^{(i)}$$

    Minimise loss over the domain Ω:

$$\mathcal{L} = \textstyle\sum_\Omega \parallel \tilde{\boldsymbol{f}} - \hat{\boldsymbol{f}} \parallel^2$$

    Update parameters if $\mathcal{L} < \mathcal{L}_{min}$

  end for

  Store $(s_0^{(i)}, d_0^{(i)}, \boldsymbol{p}^{(i)}, \boldsymbol{c}^{(i)})$

end for

return $\{(s_0^{(i)}, d_0^{(i)}, \boldsymbol{p}^{(i)}, \boldsymbol{c}^{(i)})\}_{i=1}^N$

The prediction of the neural network gives a very accurate representation of force density when compared with the ground truth as seen in figure 2. We reconstruct the velocity field using the parameters obtained from the predicted density using Algorithm 1. The algorithm returns position and orientation of the particle along with the strengths of the stresslet ($2s$ mode) and source dipole ($3t$ mode). These parameters can be used to reconstruct velocity field using (2.1)–(2.6). The predicted velocity field represents the flow with a good accuracy and we are able to decompose the flow into $2s$ and $3t$ modes. The predictions of the stresslet and source dipole strength give very close results with respect to the ground truth having 3.26 % and 5.70 % error, respectively, averaged over 600 test cases unseen to the neural network. We have predicted the parameters using the neural networks for particles with arbitrary positions and orientations and reconstructed the velocity field (see figure 3).

5. Inferring many-particle flow

We extend our model further to test with flow due to N particles. Here we consider the case of $N = 4$. In an arbitrary velocity field, we first detect the number of particles in the box, detecting the number of locations where force density magnitude is concentrated over some small local area using scikit-ndimage (Van der Walt *et al.* 2014). We ensure there is sufficient distance between two different regions associated with particles while detecting. The approximate particle position can be guessed as the centre of mass

https://doi.org/10.1017/jfm.2025.10502 Published online by Cambridge University Press

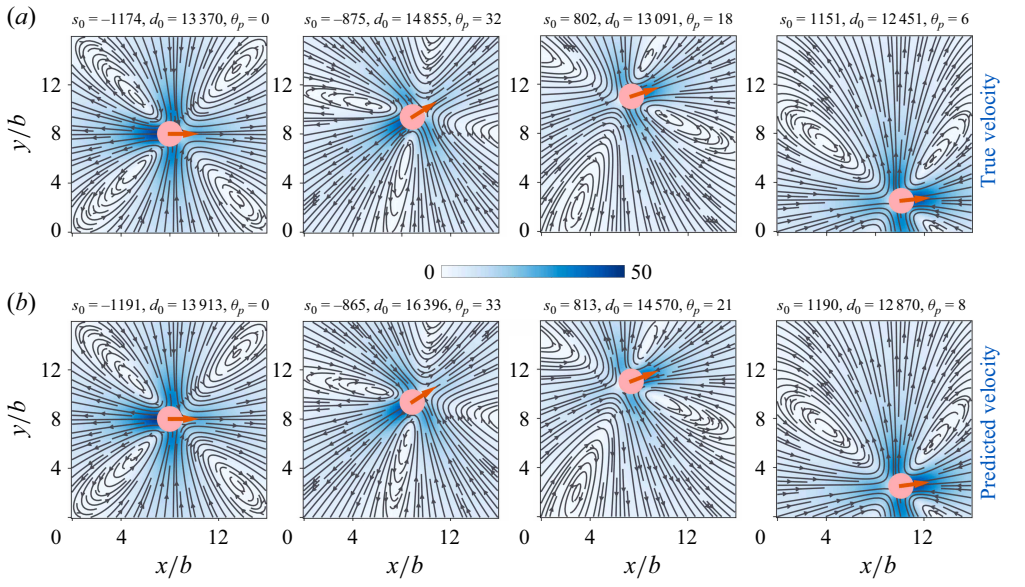


Figure 3. True (*a*) and predicted (*b*) velocity fields due to different values of stresslet strength (s_0), source dipole strength (d_0), orientation ($\boldsymbol{p}$) and particle position ($\boldsymbol{c}$). Values have been rounded to the nearest integer for clarity. We show four distinct examples for illustration. The flow velocity streamlines have been overlaid on a pseudo colour plot of the flow speed. Here, θ_p is the angle made by the particle with the positive x-axis of Cartesian coordinates.

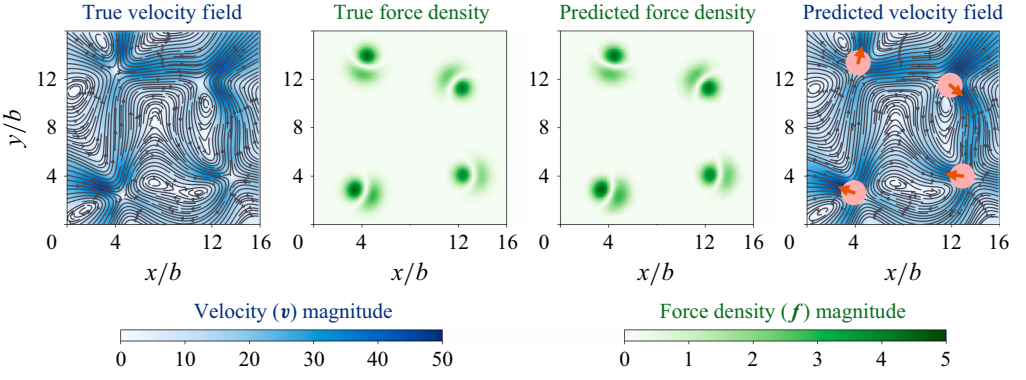


Figure 4. Predictions of the model for multiple particles in a grid. The flow velocity streamlines have been overlaid on a pseudo colour plot of the flow speed. Magnitude of the force density is also shown. Our inference method using deep learning predicts location and orientation of the particles as well as the force density in the fluid due to them.

(force density magnitude) of that region. Now each of these regions can be thought of as a single-particle system and the estimation methods of a single particle can be applied. The process is explained in Algorithm 1.

The neural network is able to predict the force density effectively as seen from the comparison with ground-truth force density in figure 4. From the predicted force density, we get the parameters by Algorithm 1. The final output of this procedure is a set of inferred parameters for all particles: $\{\boldsymbol{p}^{(i)}, \boldsymbol{c}^{(i)}\ s_0^{(i)}$ and $d_0^{(i)}\}_{i=1}^{N}$. These parameters can be used to reconstruct the velocity field using the results of § 2. Thus, our method using deep learning

https://doi.org/10.1017/jfm.2025.10502 Published online by Cambridge University Press

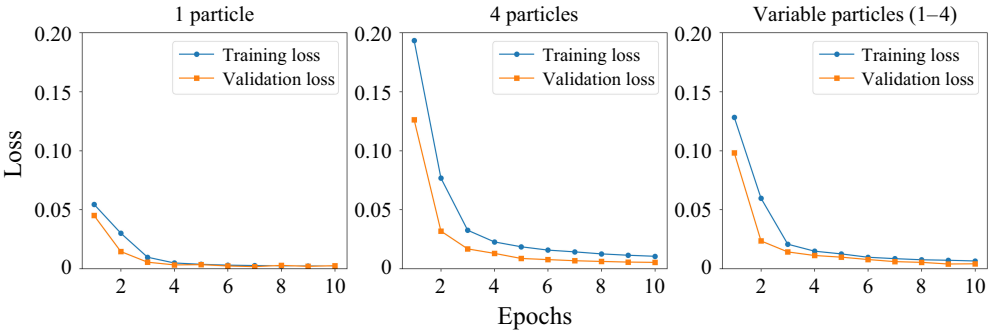


Figure 5. The training loss and validation loss over epochs indicating that the model is learning well without overfitting. We show this for cases with fixed number of particles ($N = 1$ and $N = 4$ are shown) and variable number of particles (N varies from 1 to 4).

to extract force density and then fitting that data to the known model for the fluid flow allows prediction in a system of many particles.

In addition, our method can be applied to a variable number of particles. We note that our network is capable of predicting the force density accurately even if the number of particles is varied, while the Algorithm 1 is indifferent to the number of particles. We plot the training loss and validation loss in figure 5 for cases of fixed and variable number of particles. For the case of a fixed number, we show data for the system with one and four particles, while in the case of a variable number of particles, we have considered one to four particles. Histogram of error is shown in figure 6 in Appendix B. Thus, robustness of the method is demonstrated by showing that the model works for a variable number of particles. Moreover, we show that the model works even when we change spatial resolution of the data. See figure 7 in Appendix B.

6. Summary and conclusions

In this paper, we have studied the inverse problem of computing activity of colloidal particles from the specified flow field data around them by combining artificial intelligence and physics-based methods. Such a method is particularly attractive since PIV data from experiments can provide us with flow field data. Using deep learning, we predict activity from the two-dimensional flow field data created by active particles. We achieve this by inverting the flow field data to obtain force density in the fluid using CNNs. The reconstructed flow field from the predicted force density is in excellent agreement with the true flow field. Finally, using a new inference algorithm, we infer the position and orientation of active particles along with irreducible components of the velocity (stresslet and source dipole).

Extending our method to infer the arbitrary modes of active slip around many active particles for a given flow data in two- and three-dimensional systems suggests an exciting direction for future work. In this paper, we have used deterministic fluid velocities around active particles in two-dimensions to infer their activity. Another potential future direction is to consider denoising and reconstruction methods in cases where there is noise in the fluid velocity (Weng *et al.* 2001; Wang, Liu & Wang 2022). Comparison of different architectures to find the most optimal one for these applications is another direction for future work. For generality, we have used a simple loss function in our CNN architecture. Using a physics-informed neural network with a more problem-specific choice of loss function suggests another exciting avenue for future work.

https://doi.org/10.1017/jfm.2025.10502 Published online by Cambridge University Press

In this paper, we have used CNNs to solve an inverse problem to obtain force density given fluid-flow data. This inverse problem has inherent non-trivial spatial correlations. The CNNs are particularly well-suited for solving inverse problems involving spatial correlations because they are inherently designed to capture and learn hierarchical spatial features through convolutional filters (Goodfellow *et al.* 2016). In addition, we show in figure 8 and Appendix B that the ability of CNNs to learn nonlinear mapping is crucial for our case by training the convolution layers with and without activation functions. Moreover, CNNs are desirable for large datasets (Goodfellow *et al.* 2016), which is the case for inference from flow-field data, especially when larger numbers of particles are considered. We also note that the methodology using CNNs presented in this paper is only expected to work for Stokesian fluid-flow data. For more complex settings, when the steady Stokes equation, (2.1), is no longer a valid description of fluid flow – such as scenarios where inertia and/or memory effects in the fluid are important – entirely different architectures and strategies of training may be needed.

Declaration of interests. The authors report no conflict of interest.

Data availability. All code used in this work is freely available on GitHub (https://github.com/soft-matter-physics/deep-learning-active-flows).

Appendix A. Solving the Stokes equation to compute flow around active particles

The flow around a finite-sized rigid sphere of radius b in a fluid of viscosity η with single layer density $\boldsymbol{q}$ is given by the boundary integral expression (Pozrikidis 1992):

$$\boldsymbol{v}(\boldsymbol{r}) = -\int \boldsymbol{G}(\boldsymbol{r}-\boldsymbol{r}')\cdot\boldsymbol{q}(\boldsymbol{r}')\,\mathrm{d}S. \tag{A1}$$

Here $\boldsymbol{r}$ is the point at which the fluid velocity is computed. $\boldsymbol{G}$ is a Green's function of Stokes equation and S is the surface of the particle, while $\boldsymbol{r}'$ is an integration variable over the surface. These integrals can be solved by expanding the boundary fields in TSHs (tensorial spherical harmonics) (Hess 2015). The TSHs are defined as

$$Y^{(l)}_{\alpha_1\ldots\alpha_l}(\hat{\boldsymbol{b}}) = (-1)^l\, b^{(l+1)}\,\nabla_{\alpha_1}\ldots\nabla_{\alpha_l}1/b. \tag{A2}$$

Here $\hat{\boldsymbol{b}} = \boldsymbol{b}/b$, while $b = ||\boldsymbol{b}||$ and $\boldsymbol{b}$ is the radius vector of the active colloidal sphere. The expansion of the single layer density $\boldsymbol{q}$ in TSHs is

$$\boldsymbol{q}(\boldsymbol{b}) = \sum_{l=1}^{\infty} w_l\, \boldsymbol{Q}^{(l)}\cdot\boldsymbol{Y}^{(l-1)}(\hat{\boldsymbol{b}}), \qquad \boldsymbol{Q}^{(l)} = \tilde{w}_l\int \boldsymbol{q}\,\boldsymbol{Y}^{(l-1)}(\hat{\boldsymbol{b}})\,\mathrm{d}S. \tag{A3}$$

Here, the l-dependent constants are $w_l = (2l-1)/(4\pi b^2)$ and $\tilde{w}_l = 1/((l-1)!(2l-3)!!)$ (Hess 2015). Here, and throughout the paper, a maximal contraction of two tensors is denoted by a dot product. The boundary integral of (A1) can be solved exactly as follows: (*a*) expanding the boundary fields in TSHs, (*b*) performing a Taylor expansion of kernels (the Green's function) about the centre of the particles and (*c*) using the orthogonality of TSHs (Singh *et al.* 2015; Singh 2018). The final result is the following summation:

$$\boldsymbol{v}(\boldsymbol{r}) = -\sum_{l=1}^{\infty}\left(1+\frac{b^2}{4l+2}\nabla^2\right)\nabla^{(l-1)}\boldsymbol{G}(\boldsymbol{r})\cdot\boldsymbol{Q}^{(l)}. \tag{A4}$$

Thus, the flow due to lth mode decays as $\nabla^{l-1}\boldsymbol{G}(\boldsymbol{r})$. In three-dimensions $\boldsymbol{G}(\boldsymbol{r}) \propto r^{-1}$, and thus, the flow due to any of the l modes as r^{-l} (Singh *et al.* 2015; Deb & Singh 2025),

https://doi.org/10.1017/jfm.2025.10502 Published online by Cambridge University Press

where r is the distance from the source. Equation (2.1) in the Fourier space is

$$\boldsymbol{v}(\boldsymbol{k}) = \boldsymbol{G}(\boldsymbol{k}) \cdot \boldsymbol{f}(\boldsymbol{k}), \qquad \boldsymbol{G}(\boldsymbol{k}) = \frac{1}{\eta k^2}\left(\boldsymbol{I} - \frac{\boldsymbol{k}\boldsymbol{k}}{k^2}\right). \tag{A5}$$

Here $\boldsymbol{I}$ is the identity tensor and $\boldsymbol{k}$ is Fourier wave vector. We define the Fourier transform of a function $\varphi(\boldsymbol{r})$ and Dirac delta function $\delta(\boldsymbol{r})$ as

$$\varphi(\boldsymbol{k}) = \int \varphi(\boldsymbol{r})\, \mathrm{e}^{-i\boldsymbol{k}\cdot\boldsymbol{r}}\, \mathrm{d}\boldsymbol{r}, \qquad \delta(\boldsymbol{r}) = \frac{1}{(2\pi)^3}\int \mathrm{e}^{i\boldsymbol{k}\cdot\boldsymbol{r}}\, \mathrm{d}\boldsymbol{k}. \tag{A6a}$$

Using (A5) and (A4), we can identify an effective force density:

$$\boldsymbol{f}(\boldsymbol{k}) = -\sum_{l=0}^{\infty}\left(1 - \frac{b^2}{4l+2}k^2\right)(i\boldsymbol{k})^{(l-1)} \cdot \boldsymbol{Q}^{(l)} \tag{A7}$$

that produces the flow in (A4). Using the definition of the Dirac delta function, the force density can be written in the coordinate space as

$$\boldsymbol{f}(\boldsymbol{r}) = -\sum_{l=1}^{\infty}\left(1 + \frac{b^2}{4l+2}\nabla^2\right)\nabla^{l-1}\delta(\boldsymbol{r}) \cdot \boldsymbol{Q}^{(l)} \tag{A8}$$

that satisfies the Stokes equation. Since our interest is in the external flow, we are free for this purpose to replace the δ function in (A8) with a mollifier M which produces the same exterior flow (Maxey & Patel 2001; Delmotte *et al.* 2015; Singh 2018). We choose a Gaussian mollifier whose mean is at the centre of the particle and the variance equals $2b^2/\pi$ (Maxey & Patel 2001). Using (A5), the expression of the flow in Fourier space is

$$\boldsymbol{v}^M(\boldsymbol{k}) = -\sum_{l=1}^{\infty}\left(1 - \frac{b^2}{4l+2}k^2\right)(-i\boldsymbol{k})^{l-1}\, M(\boldsymbol{k})\boldsymbol{G}(\boldsymbol{k}) \cdot \boldsymbol{Q}^{(l)}. \tag{A9}$$

The coefficients of single layer density $\boldsymbol{Q}^{(l)}$ are lth rank tensors, symmetric irreducible in their last $l-1$ indices. The reducible tensors $\boldsymbol{Q}^{(l)}$ can be written in irreducible forms as $\boldsymbol{Q}^{(l\sigma)}$. Here, $\boldsymbol{Q}^{(l\sigma)}$ are irreducible tensors of rank l, $l-1$ and $l-2$ for $\sigma = s, a$ and t respectively. Here $\sigma = s$ represents symmetric-traceless part of the tensor, while $\sigma = a$ represents the antisymmetric part, and $\sigma = t$ denotes the trace. The final expression of the flow – (2.2) of the main text – in terms of irreducible components is obtained when (A9) is written as a sum over the irreducible components of $\boldsymbol{Q}^{(l)}$.

Appendix B. Quantitative analysis

B.1. *Root mean square error*

The convergence plot (average of mean squared error over batches of 32 versus epoch) is given in figure 5 of the main text. It shows the good fit of the neural network. The network was tested over 600 test cases in each of the single particle, multiple particle and variable particle instances. The RMSE of the true force density versus that predicted by the network over the cases is plotted for each of the instances as a histogram in figure 6. The RMSE is calculated over all grid points for a single instance.

B.2. *Variable resolution*

In figure 7, we demonstrate that our network works for variable spatial resolution. In the main text, we have set the spatial resolution to be 128×128 while we vary the number

https://doi.org/10.1017/jfm.2025.10502 Published online by Cambridge University Press

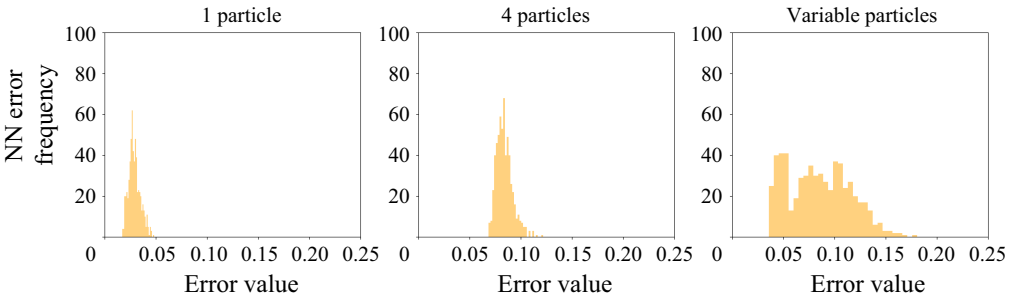


Figure 6. Histogram of root mean squared error from the predictions by CNN.

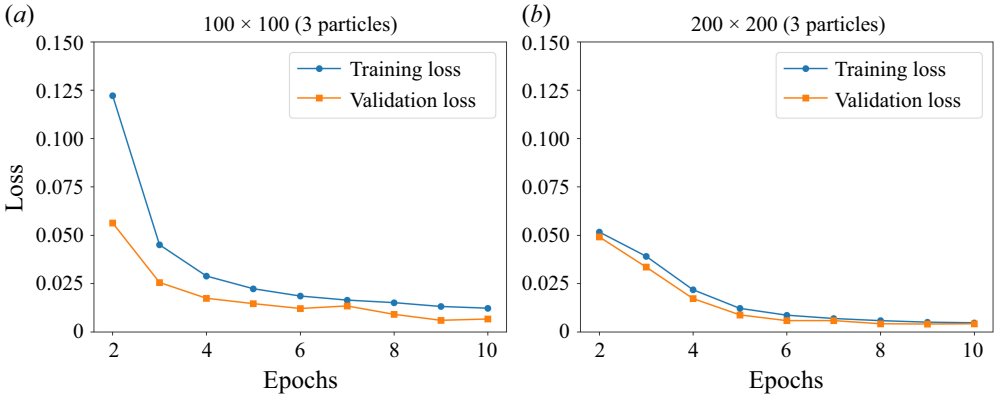


Figure 7. Loss function as a function of epochs for two different resolutions of the data.

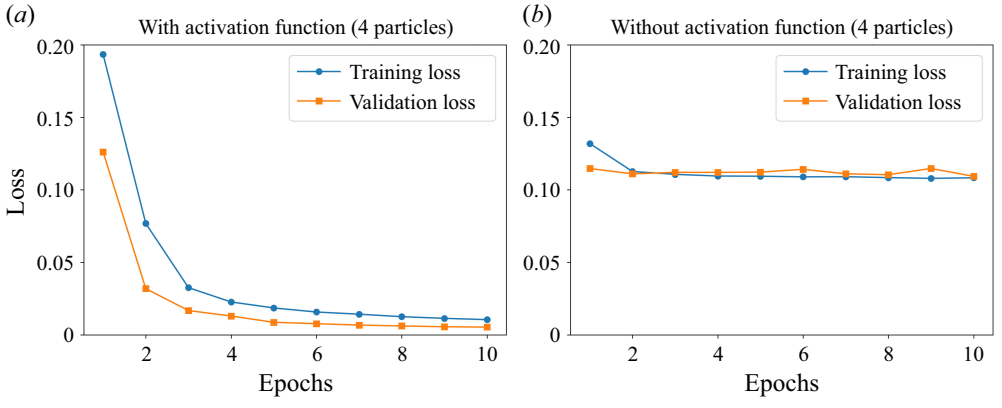


Figure 8. Ablation study. Measuring loss with epochs using CNN with activation functions (*a*) and without activation function (*b*).

of particles. Here, we plot the loss function as a function of epochs for two different resolutions (100 × 100 and 200 × 200) for an inference study of three particles.

B.3. *Ablation study*

The advantages of CNNs in this particular problem is studied by checking the solutions without the components of CNNs. To evaluate it, the dataset was trained with just

https://doi.org/10.1017/jfm.2025.10502 Published online by Cambridge University Press

convolution layers without any activation functions, which essentially makes it a linear map. The comparison of this case against the CNN with activation function is shown by the convergence plot in figure 8. The former is not converging with similar iterations whereas the latter converges rapidly. The CNNs learn nonlinear mappings using activation functions. Thus, we demonstrate the importance of nonlinear mappings in deep learning by showing that the CNN with activation function is essential for the model presented here to be effectively trained against data.

References

Adrian, R.J. 1991 Particle-imaging techniques for experimental fluid mechanics. *Annu. Rev. Fluid Mech.* **23** (1), 261–304.

Baruah, P.V., Padhan, N.B., Maji, B., Pandit, R. & Sharma, P., 2025 Emergent active turbulence and intermittency in dense algal suspensions of *Chlamydomonas reinhardtii*. arXiv:2503.03834.

Blake, J.R. 1971 A spherical envelope approach to ciliary propulsion. *J. Fluid Mech.* **46** (1), 199–208.

Boffi, N.M. & Vanden-Eijnden, E. 2024 Deep learning probability flows and entropy production rates in active matter. *Proc. Natl Acad. Sci.* **121** (25), e2318106121.

Brennen, C. & Winet, H. 1977 Fluid mechanics of propulsion by cilia and flagella. *Annu. Rev. Fluid Mech.* **9** (1), 339–398.

Cichos, F., Gustavsson, K., Mehlig, B. & Volpe, G. 2020 Machine learning for active matter. *Nat. Machine Intell.* **2** (2), 94–103.

Deb, M. & Singh, R. 2025 Ewald summing irreducible components of flow around active particles. *J. Chem. Phys.* **163** (2), 024901.

Delmotte, B., Keaveny, E.E., Plouraboué, F. & Climent, E. 2015 Large-scale simulation of steady and time-dependent active suspensions with the force-coupling method. *J. Comput. Phys.* **302**, 524–547.

Drescher, K., Dunkel, J., Cisneros, L.H., Ganguly, S. & Goldstein, R.E. 2011 Fluid dynamics and noise in bacterial cell–cell and cell–surface scattering. *Proc. Natl Acad. Sci.* **108** (27), 10940–10945.

Drescher, K., Goldstein, R.E., Michel, N., Polin, M. & Tuval, I. 2010 Direct measurement of the flow field around swimming microorganisms. *Phys. Rev. Lett.* **105** (16), 168101.

Ebbens, S.J. & Howse, J.R. 2010 In pursuit of propulsion at the nanoscale. *Soft Matt.* **6** (4), 726–738.

Ghose, S. & Adhikari, R. 2014 Irreducible representations of oscillatory and swirling flows in active soft matter. *Phys. Rev. Lett.* **112** (11), 118102.

Glorot, X., Bordes, A. & Bengio, Y. 2011 Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, vol. 15 (ed. G. Gordon, D. Dunson & M. Dudík), Proc. Mach. Learn. Res.), pp. 315–323. PMLR.

Goldstein, R.E. 2015 Green algae as model organisms for biological fluid dynamics. *Ann. Rev. Fluid Mech.* **47**, 343–375.

Goodfellow, I., Bengio, Y., Courville, A. & Bengio, Y. 2016 *Deep Learning*. vol. 1. MIT press Cambridge.

Guasto, J.S., Johnson, K.A. & Gollub, J.P. 2010 Oscillatory flows induced by microorganisms swimming in two dimensions. *Phys. Rev. Lett.* **105** (16), 168102.

Hannel, M.D., Abdulali, A., O'Brien, M. & Grier, D.G. 2018 Machine-learning techniques for fast and accurate feature localization in holograms of colloidal particles. *Opt. Express* **26** (12), 15221–15231.

Hess, S. 2015 *Tensors for Physics*. Springer.

Hornik, K., Stinchcombe, M. & White, H. 1989 Multilayer feedforward networks are universal approximators. *Neural Networks* **2** (5), 359–366.

Kingma, D.P. & Ba, J. 2014 Adam: a method for stochastic optimization. arXiv:1412.6980.

Krishnamurthy, D. & Prakash, M. 2023 Emergent programmable behavior and chaos in dynamically driven active filaments. *Proc. Natl Acad. Sci.* **120** (28), e2304981120.

Lauga, E. 2020 *The Fluid Dynamics of Cell Motility*, vol. 62. Cambridge University Press.

Lighthill, M.J. 1952 On the squirming motion of nearly spherical deformable bodies through liquids at very small Reynolds numbers. *Commun. Pure Appl. Maths* **5** (2), 109–118.

Marmol, M., Gachon, E. & Faivre, D. 2024 Colloquium: magnetotactic bacteria: from flagellar motor to collective effects. *Rev. Mod. Phys.* **96** (2), 021001.

Maxey, M.R. & Patel, B.K. 2001 Localized force representations for particles sedimenting in Stokes flow. *Intl J Multiphase Flow* **27** (9), 1603–1626.

https://doi.org/10.1017/jfm.2025.10502 Published online by Cambridge University Press

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.* **115** (36), 9026–9031.

PEDLEY, T.J. 2016 Spherical squirmers: models for swimming micro-organisms. *IMA J. Appl. Maths* **81** (3), 488–521.

POZRIKIDIS, C. 1992 *Boundary Integral and Singularity Methods for Linearized Viscous Flow*. Cambridge University Press.

RABAULT, J., KOLAAS, J. & JENSEN, A. 2017 Performing particle image velocimetry using artificial neural networks: a proof-of-concept. *Meas. Sci. Technol.* **28** (12), 125301.

SINGH, R. 2018 Microhydrodynamics of active colloids [HBNI Th131]. PhD thesis, HBNI.

SINGH, R., GHOSE, S. & ADHIKARI, R. 2015 Many-body microhydrodynamics of colloidal particles with active boundary layers. *J. Stat. Mech.* **2015** (6), P06017.

SUPEKAR, R., SONG, B., HASTEWELL, A., CHOI, G.P.T., MIETKE, A. & DUNKEL, J. 2023 Learning hydrodynamic equations for active matter from particle simulations and experiments. *Proc. Natl Acad. Sci.* **120** (7), e2206994120.

THUTUPALLI, S., GEYER, D., SINGH, R., ADHIKARI, R. & STONE, H.A. 2018 Flow-induced phase separation of active particles is controlled by boundary conditions. *Proc. Natl Acad. Sci.* **115** (21), 5403–5408.

TSANG, A.C.H., TONG, P.W., NALLAN, S. & PAK, O.S. 2020 Self-learning how to swim at low Reynolds number. *Phys. Rev. Fluids* **5**, 074101.

VAN DER WALT, S., SCHÖNBERGER, J.L., NUNEZ-IGLESIAS, J., BOULOGNE, F., WARNER, J.D., YAGER, N., GOUILLART, E. & YU, T. 2014 scikit-image: image processing in Python. *PeerJ* **2**, e453.

WANG, H., LIU, Y. & WANG, S. 2022 Dense velocity reconstruction from particle image velocimetry/particle tracking velocimetry using a physics-informed neural network. *Phys. Fluids* **34** (1), 017116.

WENG, W.G., FAN, W.C., LIAO, G.X. & QIN, J. 2001 Wavelet-based image denoising in (digital) particle image velocimetry. *Signal Process.* **81** (7), 1503–1512.

WILLERT, C.E. & GHARIB, M. 1991 Digital particle image velocimetry. *Exp. Fluids* **10** (4), 181–193.

https://doi.org/10.1017/jfm.2025.10502 Published online by Cambridge University Press