nature machine intelligence

Article

https://doi.org/10.1038/s42256-022-00595-0

Geometric deep learning reveals the spatiotemporal features of microscopic motion

Received: 17 February 2022

Accepted: 18 November 2022

Published online: 16 January 2023

Check for updates

Jesús Pineda[1], Benjamin Midtvedt[1], Harshith Bachimanchi[1], Sergio Noé[2], Daniel Midtvedt[1], Giovanni Volpe[1] ✉ & Carlo Manzo[2] ✉

The characterization of dynamical processes in living systems provides important clues for their mechanistic interpretation and link to biological functions. Owing to recent advances in microscopy techniques, it is now possible to routinely record the motion of cells, organelles and individual molecules at multiple spatiotemporal scales in physiological conditions. However, the automated analysis of dynamics occurring in crowded and complex environments still lags behind the acquisition of microscopic image sequences. Here we present a framework based on geometric deep learning that achieves the accurate estimation of dynamical properties in various biologically relevant scenarios. This deep-learning approach relies on a graph neural network enhanced by attention-based components. By processing object features with geometric priors, the network is capable of performing multiple tasks, from linking coordinates into trajectories to inferring local and global dynamic properties. We demonstrate the flexibility and reliability of this approach by applying it to real and simulated data corresponding to a broad range of biological experiments.

The biological functions of living systems rely on interactions that dynamically change in response to endogenous and exogenous stimuli. Studying the motion of the components of these systems sets the basis for mechanistic insights to understand health and disease[1]. Over the past 20 years, microscopy has advanced to the point where it can monitor dynamic processes at multiple scales with unprecedented spatiotemporal resolution. Time-lapse microscopy experiments have unveiled the strategies that unicellular organisms employ to search for food or to avoid adverse conditions, and have helped to understand tissue growth and repair, cancer metastasis, quorum sensing, the emergence of multicellularity and immune responses in multicellular organisms[2,3]. Fluorescence microscopy has monitored biological motion down to the nanoscale, detailing the diffusion of individual organelles and molecules within the cellular environment and disclosing their role, for example, in the fundamental processes of signalling and function regulation[4,5].

The momentous improvement of microscopy acquisition techniques has led to a substantial effort to develop and improve algorithms to automatically extract quantitative information from these experiments[6,7]. The standard analysis pipeline of tracking-by-detection methods entails the following steps[4,7,8]: (1) video frames are partitioned (segmentation) and/or otherwise processed to detect and locate objects of interest (detection/localization); (2) detected positions at different times are connected into trajectories (linking); (3) reconstructed trajectories are finally analysed to quantify dynamical parameters (motion characterization). The first two steps are often presented together and referred to as tracking. Several factors complicate the analysis of biological experiments, such as imaging noise, high object density, fusion or splitting events, random and heterogeneous motion, and shape-changing objects. Errors at each step propagate along the pipeline and ultimately impact the extraction of dynamic information.

[1]Department of Physics, University of Gothenburg, Gothenburg, Sweden. [2]Facultat de Ciències, Tecnologia i Enginyeries, Universitat de Vic – Universitat Central de Catalunya (UVic-UCC), Vic, Spain. ✉e-mail: giovanni.volpe@physics.gu.se; carlo.manzo@uvic.cat

Numerous algorithmic solutions have been proposed to tackle the limitations of tracking algorithms and their performance has been compared in open challenges[6,7]. However, most of these methods are specific to a given experiment or dynamic model, and often require manual tuning of parameters. The current deep-learning revolution has fostered the development of various methods for both tracking[9–13] and motion characterization[14].

Geometric deep learning provides compelling approaches to tackle tracking and motion characterization from a different perspective. It generalizes neural networks to problems that can be described by mathematical objects such as graphs that encode information about the structure of the input[15]. Deep-learning methods based on graphs are typically referred to as graph neural networks (GNNs)[16] and have been successfully applied, for example, to molecular property prediction[17], drug discovery[18], computer-assisted retrosynthesis[19] and human trajectory prediction[20]. Besides being ubiquitously used in science to represent complex systems[21,22], graphs provide a natural and intuitive way to represent the information contained in tracking experiments[23,24].

Here we describe a framework for Motion Analysis through GNN Inductive Knowledge (MAGIK), which provides the accurate estimation of dynamical properties from time-lapse microscopy. MAGIK models the system's motion and interactions through a graph representation. This graph is processed through an interpretable and adaptive attention-based GNN that estimates the associations among the objects and provides insights into the intrinsic dynamics of the systems. We demonstrate the flexibility and reliability of MAGIK by quantifying its performance on real and simulated data corresponding to a broad range of biological experiments. First, we benchmark it on its most natural application, that is, trajectory linking, in a variety of challenging experimental scenarios. Then, we show that MAGIK can estimate local and global dynamical properties without explicit linking even in highly heterogeneous scenarios.

Results

MAGIK represents spatiotemporal relations in a graph

MAGIK provides a GNN framework to estimate the dynamical properties of moving objects from time-lapse experiments. MAGIK models the objects' motion and physical interactions using a graph representation. The details of the algorithm are given in Methods ('Description of MAGIK') and Extended Data Fig. 1. In this section, we provide a high-level description of the architecture (Fig. 1).

Graphs can define arbitrary relational structures between nodes connecting them pairwise through edges. When training a GNN, the graph architecture guides the learning process about the objects and their relations by introducing a relational inductive bias[16]. In MAGIK, each node describes an object detection at a specific time, the edges connect spatiotemporally close objects, and a set of global attributes encodes system-level properties. As an example, for subsequent frames of a cell migration experiment, each detected object (orange crosses in Fig. 1a) is represented as a node with a vector of node features (Fig. 1b). Directed edges with relational features connect each node to objects detected in the future in its proximity (Fig. 1b). There are no intrinsic restrictions on the type or number of descriptors (for example, location and morphological features, image-based quantities, biological events, interaction strength, distance, direction) that can be encoded in the graph feature representation. The basic graph relational structure is established through a set of rules that link nodes pairwise based on distance metrics between features. Node and edge features are encoded through learnable functions implemented by neural networks (Fig. 1c,d). An extra learnable token is added to aggregate global attributes from the whole graph[25–27] (Fig. 1e).

The graph is processed through a sequence of attention-based fingerprinting graph neural networks (FGNNs; see also Methods 'Description of MAGIK' and Extended Data Fig. 1) that propagate information via message-passing steps (Fig. 1f–i). The relational inductive knowledge implemented in the graph structure sketches a network of redundant object associations. The objective of the FGNN is to modulate the association strength to identify the edges majorly influencing the dynamic properties of each object. For this, the FGNN implements two mechanisms that combine information from multiple objects at the local and global levels. The first mechanism intervenes when aggregating edge features to a node (equation (3)). The contribution of each edge has a weight that depends on the distance between the connected nodes through a function with learnable parameters (equation (2)), thus defining a learnable local receptive field[28,29]. The second is a gated self-attention mechanism[30] that sets in when updating the latent representation of nodes (equation (4)). The node update operation involves also information stemming beyond each node's topological neighbourhood, thus effectively expanding the receptive field to objects that, although not physically connected, can offer relevant information about the overall dynamics. The FGNN further updates the extra token for global attributes using information from all the nodes; thus, this extra token serves as an antenna to provide system-level insights.

The output of the FGNN is decoded by the last block of the GNN into an output graph, whose nodes, edges, and global attributes can be used to solve specific problems (Fig. 1j–l).

MAGIK accurately links trajectories

We benchmark MAGIK performance on a classical trajectory linking task, consisting of establishing temporal associations between identified objects. The graph structure includes a redundant number of edges with respect to the actual associations between objects. MAGIK aims to prune the wrong edges while retaining the true connections. We thus model this task as an edge-classification problem with a binary label (linked/unlinked) by minimizing the binary cross-entropy. From the predicted edge features, trajectories are built through a postprocessing algorithm that eliminates spurious connections (Methods 'Post-processing algorithm for trajectory linking').

To test MAGIK, we use the silver-standard segmentation datasets provided for the training of the sixth edition of the Cell Tracking Challenge[7]. A representative segmentation of the dataset DIC-C2DH-HELA, corresponding to HeLa cells on a flat glass imaged through differential interference contrast, is shown in Fig. 2a. From the segmentation, we calculate the mean pixel intensity, area, perimeter, eccentricity and solidity of the segmented objects, which we use as input node features. The Euclidean distance between neighbouring objects is used as the sole edge feature. To limit memory usage, we generate graphs by drawing edges only between objects within a limited spatial and temporal reach (Fig. 2b).

The DIC-C2DH-HELA dataset presents several challenges, such as the heterogeneity in cell shape and dynamics as a consequence of migration and proliferation (Fig. 2g). Examples of ground-truth and predicted graphs are shown in Fig. 2c,e showing a good agreement, as confirmed by an F_1 score of 99.4% in edge prediction. For the evaluation of performance at the trajectory level (Fig. 2d,f), we calculated the tracking accuracy measure (TRA), a normalized weighted distance between the tracking prediction and the reference tracking ground truth[31] (Methods, 'Quantification of cell-tracking results'). MAGIK reached a TRA = 99.2%, showing a great capability of correctly following objects despite shape changes and cell divisions (Fig. 2g and Supplementary Video 1).

We applied MAGIK to several other datasets of the 6th Cell Tracking Challenge, obtaining outstanding results for different microscopy techniques and cell types. Representative video frames with segmentation are shown in Fig. 3. Even though a strict objective comparison of MAGIK linking capability with other methods is limited by the fact that different algorithms rely on a different segmentation (whose errors influence linking and thus indirectly affect the value of the TRA metric),

https://doi.org/10.1038/s42256-022-00595-0

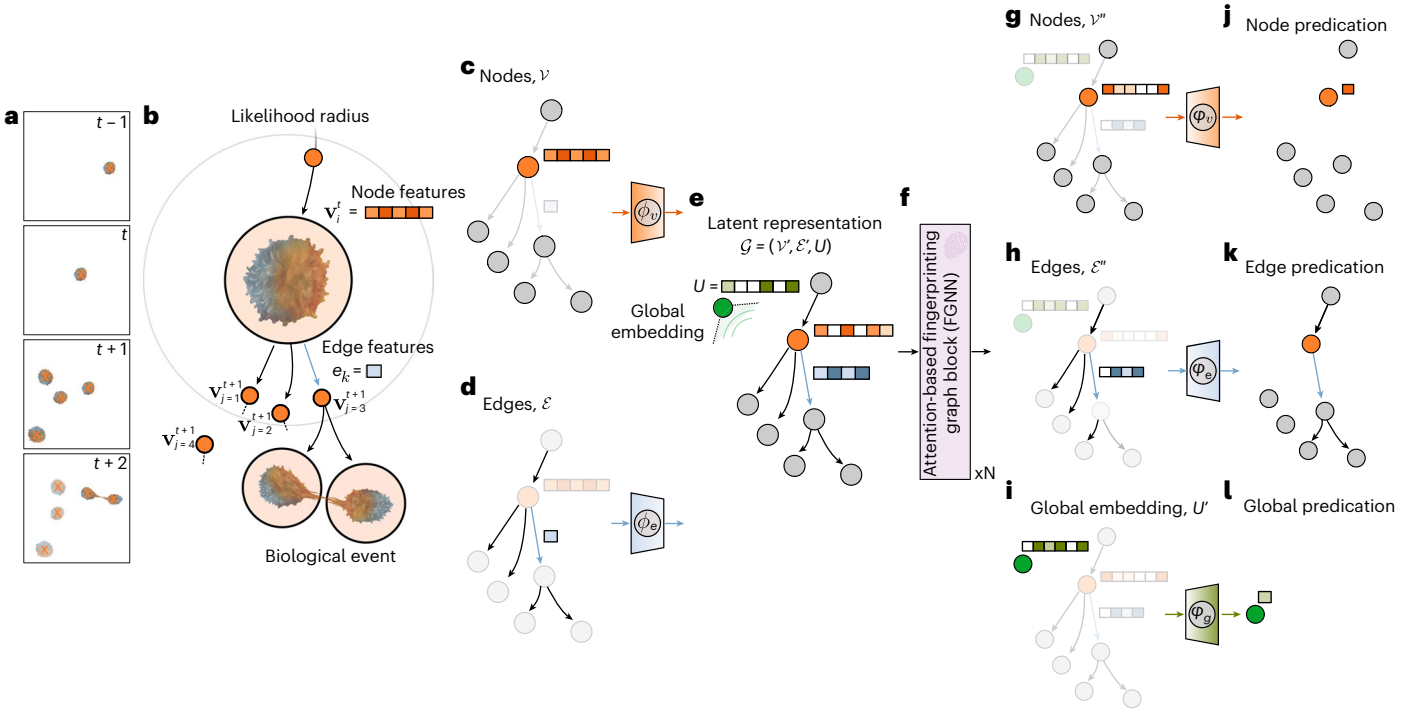


Fig. 1 | Spatiotemporal characterization of trajectories using MAGIK. **a**, Sequence of images illustrating the evolution of a group of cells over time, corresponding to frame numbers $t-1, t, t+1, t+2$. The orange crosses indicate a detection. **b**, The movement of the cells and their interactions are geometrically modelled using a directed graph, where the nodes ($\mathcal{V}$) represent the detections and the edges ($\mathcal{E}$) connect spatiotemporally close detections. Each node contains features (orange squares) such as the cell's centroid and some relevant descriptors (for example, the cell's morphological and intensity attributes). The edges contain features (blue squares) too, in this case encoding the Euclidean distance between the centroids of the cells. In this example, the node of interest, labelled with the subindex i, is connected to neighbouring nodes in the future, labelled with the subindex j within a distance-based likelihood radius (the edge between nodes with feature vectors $\mathbf{v}_i^t$ and $\mathbf{v}_{j=4}^{t+1}$ is dumped). Meaningful biological events (for example, cell divisions) are naturally encoded in the graph. **c**,**d**, The input node and edge features are mapped to a higher-level feature representation using learnable encoding functions implemented by the neural networks ϕ_v and ϕ_e, respectively. **e**, Importantly, we also append an extra learnable token $\mathbf{U}'$ to the graph latent representation $\mathcal{G} = \{\mathcal{V}', \mathcal{E}', U\}$, whose function is to provide global insights about the dynamics of the cells. **f**–**l**, MAGIK relies on attention-based fingerprinting graph blocks (FGNN; **f**) sequentially applied N times to process $\mathcal{G}$ and provide an updated representation for nodes ($\mathcal{V}''$; **g**), edges ($\mathcal{E}''$; **h**) and global information ($\mathbf{U}'$; **i**) (for further details regarding the FGNN architecture, refer to Methods 'Description of MAGIK' and Extended Data Fig. 1). Finally, $\mathcal{V}''$, $\mathcal{E}''$ and $\mathbf{U}'$ are decoded by applying learnable functions implemented by the neural networks φ_v, φ_e and φ_u, respectively, to obtain the sought-after node (**j**), edge (**k**) and global information (**l**).

MAGIK obtained TRA values that are competitive, often superior, to the best-in-class methods of the 6th Cell Tracking Challenge. The influence of segmentation on linking is particularly relevant when dealing with touching or overlapping cells; thus specific isolation strategies might be adopted to prevent over- or under-segmentation[32–35]. While we attempted to account for these errors through the augmentation procedure, segmentation errors could produce systematic changes in node features that might impair the linking performance.

MAGIK quantifies motion parameters without trajectory linking

In most applications, the ultimate objective of tracking is the characterization of the dynamics of the systems under investigation to gain insights into their underlying biological mechanisms. In this process, trajectory linking is often just an intermediate step necessary to obtain meaningful information from the data, but not the end goal itself.

MAGIK can characterize essentially any dynamic aspect without requiring the actual linking, owing to its capability of accounting for the whole spatiotemporal complexity contained in the associations between objects at multiple scales. Such linking-free analysis produces a twofold advantage. First, it bypasses the error-prone linking step, thus inherently preventing linking errors from propagating to the quantification of the ultimately relevant parameters. Second, it enables the analysis of experiments for which linking cannot be performed due to, for example, a high object density or low signal-to-noise ratio.

We apply MAGIK to analyse simulated data reproducing the diffusion of fluorescently labelled single molecules such as lipids or receptors in the plasma membrane of living cells (Fig. 4). We first consider the task of determining the diffusion coefficient from a heterogeneous ensemble of diffusing objects (Fig. 4a). We feed the network the centroid coordinates and the intensity of the localized fluorescence spots as node features and the Euclidean distance between neighbouring centroids as the edge feature. We define the problem as a node regression and minimize the mean absolute error (MAE). The target feature is the displacement scaling factor $\sqrt{2D}$, with D being the diffusion coefficient. Graphs are built by connecting localized objects with neighbours in space and time (Fig. 4b). Ground-truth and predicted graphs are shown in Fig. 4b,c, respectively. All the edges of the graph structure are drawn, representing the network of associations used to infer dynamic properties without direct linking. Nodes are colour-coded according to the value of the displacement scaling factor $\sqrt{2D}$. Their visual comparison suggests excellent agreement, further confirmed by the quantification in Fig. 4d. Notably, the same architecture can be applied at the single-trajectory level, opening interesting perspectives for the detection of dynamic changes and trajectory segmentation (Extended Data Fig. 2). The same approach can also be extended to estimate other parameters. In Extended Data Fig. 3a–d, we show the results of its application to the inference of the scaling exponent for objects undergoing anomalous diffusion, achieving

https://doi.org/10.1038/s42256-022-00595-0

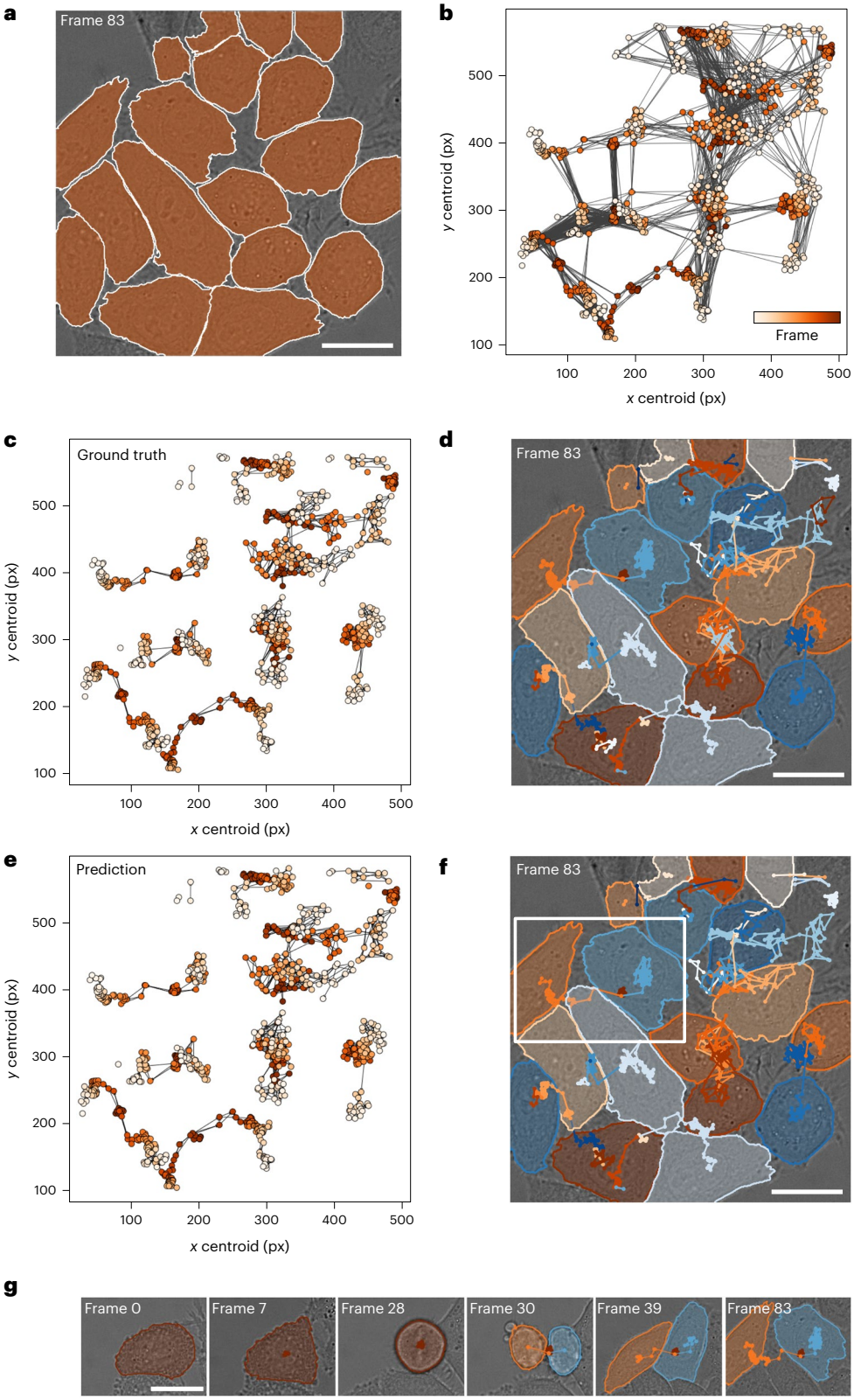


Fig. 2 | Trajectory linking using MAGIK. a, Representative frame of the HeLa cells in the DIC-C2DH-HELA validation video of the 6th Cell Tracking Challenge[7]. Scale bar, 20 µm. Segmentation (coloured regions) is used to extract relevant information from each cell along the sequence of images to build. **b**, The input graph structure including a redundant number of edges with respect to the actual associations between objects. Nodes are colour-coded with respect to the frame number (white, low; dark orange, high). **c**,**d**, Ground-truth graph (**c**) and ground-truth cell trajectories (**d**). Scale bar, 20 µm. **e**, The predicted graph agrees well with the expected solution, achieving an F_1 score equal to 99.4%. **f**, The predicted trajectories reach TRA = 99.2% compared with the ground truth. Scale bar, 20 µm. Cell divisions are detected correctly, and the network performs well also in edge regions where cells are only partially observed and move out of the field of view. **g**, Zoomed-in view of the inset in **f** showing the heterogeneity in cell shape and dynamics. A cell changes morphology during migration (frames 0–28) and divides into 2 daughter cells (frame 30) that spread and migrate apart (frames 30–83). Scale bar, 20 µm. A video visualizing the tracked cells is provided as Supplementary Video 1).

https://doi.org/10.1038/s42256-022-00595-0

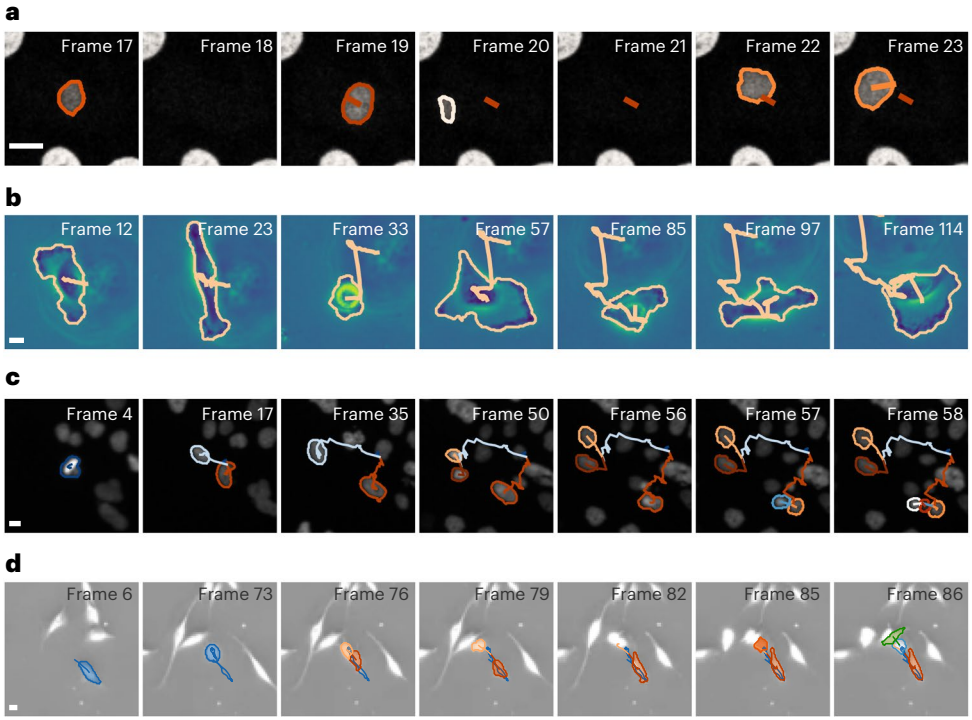


Fig. 3 | MAGIK reliably links trajectories in various experimental scenarios. **a**, Confocal microscopy of green fluorescent protein (GFP)-transfected GOWT1 mouse stem cells. MAGIK achieves an F_1 score of 99.8% and TRA = 99.2% despite the fact that the cells frequently leave the field of observation. Scale bar, 10 µm. **b**, Phase-contrast imaging of glioblastoma-astrocytoma U373 cells on a polyacrylamide substrate. MAGIK reaches an F_1 score of 99.8% and TRA = 100% even though the cells greatly change shape over time. Scale bar, 10 µm. **c**, Epifluorescence imaging of HeLa cells stably expressing histone H2b–GFP. MAGIK achieves an F_1 score of 98.8% and TRA = 98.4% despite the dense sample and frequent mitosis and collisions. Scale bar, 10 µm. **d**, Phase-contrast imaging of pancreatic stem cells on a polystyrene substrate. MAGIK obtains an F_1 score of 99.3% and TRA = 98.5% despite high cell density, elongated shapes, pronounced cell displacements and a significant number of division events. Scale bar, 10 µm. Interrupted trajectories correspond to cases where cells left the field of view or missed segmentation in the image sequence. All videos belong to the dataset of the 6th Cell Tracking Challenge[7]. Results can be observed in greater detail in Supplementary Videos 2–5.

similarly good results. It is also interesting to note that MAGIK's performance for node regression is less influenced by crowding than the performance for the linking task (Extended Data Fig. 4).

Fluorescence microscopy experiments for object tracking must ensure that the number of visualized molecules is low enough to unambiguously link the trajectories, thus they are often performed at low labelling density[4]. However, these conditions are not optimal to probe the interactions between particles and make difficult the inference of spatial patterns of diffusion[36]. Enabling the quantification of diffusion properties without linking offers the possibility to process high-density videos to determine the underlying topology and spatial heterogeneity.

As an example, we used MAGIK to resolve a spatially modulated landscape with diffusion continuously varying over more than two orders of magnitude from the localizations of diffusing particles (Fig. 4e–h), treating the problem as a node feature regression, as above. At a number density of about 0.02 px^{-2}, MAGIK is capable of correctly retrieving the spatial map of D (Fig. 4f). Remarkably, most spatial features can be already resolved with a 100-frame-long video (Fig. 4g). The spatial resolution of the predicted map can be further improved using longer videos (1,500 frames; Fig. 4h), with the typical duration of single-molecule fluorescence microscopy experiments for measuring diffusion[4].

MAGIK quantifies global dynamic properties

We applied MAGIK to directly extract ensemble information through the inference of global attributes skipping the trajectory linking step. We considered two biologically relevant scenarios. First, we analysed fluorescence microscopy experiments in which objects in the same video undergo diffusion according to different microscopic models (namely, fractional Brownian motion (FBM), annealed transient time motion (ATTM) and continuous-time random walk (CTRW); Fig. 5a–e). Although these diffusion models can give rise to anomalous diffusion, in this example they are parameterized to have the same scaling of the mean-squared displacement of Brownian motion ($\alpha = 1$)[14]. Graphs are built as described above using centroid coordinates and intensity of the localized fluorescence spots as node features and the Euclidean distance between neighbouring centroids as the edge feature. MAGIK estimates the relative fraction of objects in each category, varying from experiment to experiment, as a regression problem by minimizing the sparse categorical cross-entropy of the global attribute. The results are summarized in Fig. 5a–e, showing an outstanding accuracy in predicting the correct fractions, even when the number of objects performing the same class of motion is very low. In Extended Data Fig. 3e–h, we further demonstrate that the same approach can also estimate the fraction of object moving according to different diffusion modes (subdiffusion with $\alpha < 1$, normal diffusion with $\alpha = 1$ and superdiffusion with $\alpha > 1$).

The second example refers to simulations of holographic imaging of microorganisms diffusing in a liquid environment, such as plankton (Fig. 5f–k). We model diffusion as either FBM (Fig. 5f,g), ATTM (Fig. 5h,i) or CTRW (Fig. 5j,k) with $\alpha = 1$. Objects in the same experiments follow the same physical model but with random diffusivity. Centroid three-dimensional coordinates, mean intensity, area and refractive index of the objects are used as node features in a classification problem to determine the common diffusion model of the objects in the same video, encoded as a global attribute. As shown in Fig. 5l, MAGIK

https://doi.org/10.1038/s42256-022-00595-0

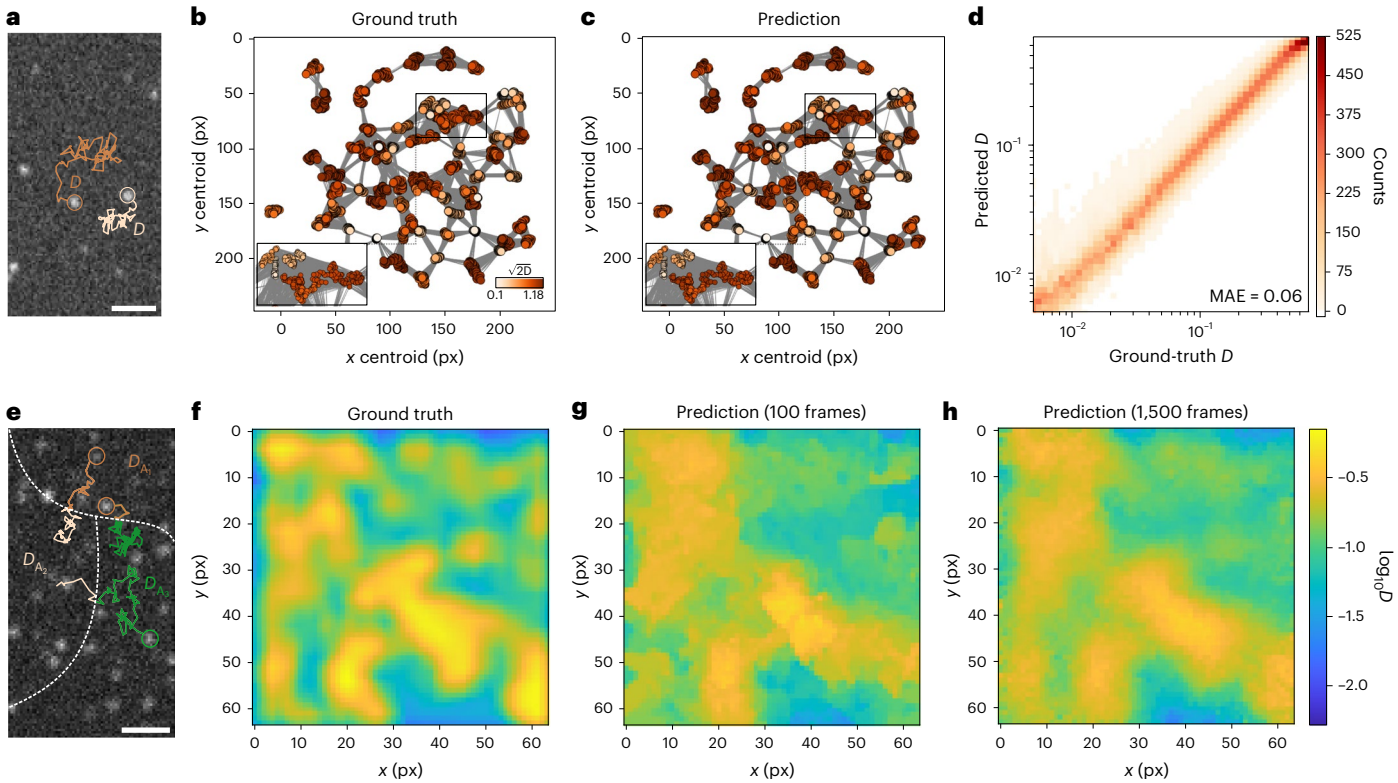


Fig. 4 | MAGIK determines local diffusion properties. a, Simulated single-object tracking experiment where fluorescence microscopy is used to follow the motion of single molecules performing Brownian motion with diffusivity D varying from particle to particle. Scale bar, 20 px. **b**,**c**, Ground-truth (**b**) and predicted (**c**) graphs. The edges depict the network of associations used to infer dynamic properties without direct linking. The nodes are colour-coded according to the value of the target feature, that is, the displacement scaling factor $\sqrt{2D}$ measured in pixels per frame (colour bar in **b**). **d**, Probability distribution of the predicted versus ground-truth diffusion coefficient D, showing a good agreement (MAE = 0.06). **e**, Simulated single-object tracking experiment where fluorescence microscopy is used to follow the motion of single molecules performing Brownian motion with diffusivity D randomly varying in space. Scale bar, 20 px. **f**–**h**, Ground-truth (**f**) and predicted (**g**,**h**) diffusion maps. Ground-truth spatial diffusivity pattern (**f**) and prediction obtained by MAGIK using 100-frame-long (**g**) and 1,500-frame-long (**h**) videos with about 0.02 localizations per px^2 per frame. The analysis is performed by breaking down the sequence into 2 and 30 videos of 50 frames each, respectively. Predicted maps are obtained by interpolating the values of diffusivity obtained for the nodes over the 64 px × 64 px grid through a triangulation-based nearest-neighbour algorithm.

correctly classifies the diffusion model even with largely overlapping objects. We find this result quite remarkable (equally so as that illustrated in Fig. 5e) since, for $\alpha = 1$, all models converge to Brownian motion and feature large similarities in their statistical properties, making their classification rather challenging even when linked trajectories are available[14]. We believe that MAGIK achieves this capability by detecting the fingerprint of each model's generative dynamics at the microscopic level.

Last, we explore MAGIK's performance for quantifying anomalous diffusion through the estimation of the exponent α (ref. 14) from a sequence of holographic images reproducing the motion of microorganisms. All the objects in the same video undergo FBM with random diffusivity and the same exponent α, varying from sequence to sequence (Fig. 5m). Also in this case, MAGIK provides remarkable results (MAE = 0.11) from short videos (about 50 frames) containing only a few objects.

Interpreting MAGIK

To determine the mechanisms that most contribute to MAGIK's performance, we evaluated different models on ablation of key components of the MAGIK architecture (Extended Data Table 1). As a baseline, we consider a version of MAGIK lacking both the learnable local receptive field and the gated self-attention and compare it with a model without gated self-attention and with the full MAGIK. As shown in Extended Data Table 1, both components contribute to improving the performance but, depending on the specific task, have a different weight. The learnable local receptive field seems to have little influence on the trajectory linking and the estimation of local diffusion properties. In these cases, the gated self-attention significantly affects the results as both problems can benefit from information originated from nodes that are distant in time and/or space. In contrast, the learnable local receptive field is responsible for most of the gain in performance when inferring the fraction of objects performing different kinds of motion. Differences between diffusion modes can be in fact detected from short-time displacements, reducing the contribution of the gated self-attention to this task.

The relative distance between objects, encoded in MAGIK as an edge feature, is crucial to tackling the tasks presented in this work. In fact, we first attempted to establish a baseline through an ablated model without edge features. When trained for the node-regression problem, such a model did not converge and produced an MAE ≈ 0.2, compatible with random predictions of the diffusion coefficient.

We also explored positional encoding to provide distance-aware information, by appending Laplacian eigenvectors to node features[27,37]. However, Laplace positional encoding did not produce any improvement in this architecture. To further explore the role of edge and node features in MAGIK, we trained a model where distances between detected objects are encoded as edge features but absolute coordinates are removed as node features. Interestingly, this model shows degradation of performance with an MAE = 0.0748 ± 0.0115 (compared with

https://doi.org/10.1038/s42256-022-00595-0

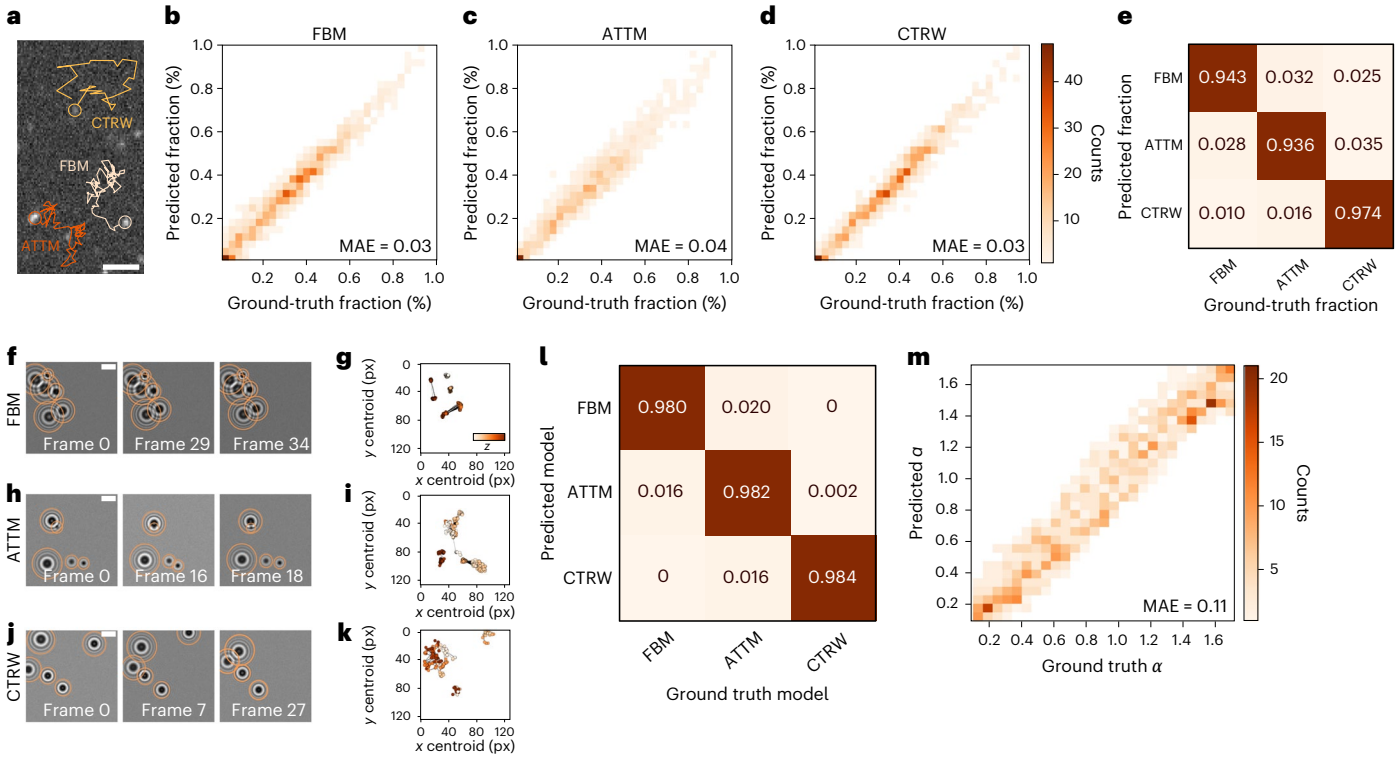


Fig. 5 | MAGIK estimates local and global dynamic properties at the ensemble and single-object levels. a, Simulated single-object tracking experiment where objects with different underlying diffusion models coexist (that is, FBM, ATTM and CTRW with anomalous diffusion exponents $\alpha = 1$). Scale bar, 20 px. **b**–**d**, Probability distribution of predicted versus ground-truth model fraction for FBM (**b**), ATTM (**c**) and CTRW (**d**). **e**, Confusion matrix demonstrating how the network classifies the underlying diffusion model exhibited by objects in 1,199 validation videos. The diagonal represents the percentage of correctly classified graph representations, constituting most cases. The off-diagonal cells represent incorrectly classified examples. Column-based normalization is applied, such that the sum along the columns adds up to 1, with minor deviations due to rounding. **f**–**k**, Representative frames of simulated holographic video (**f**,**h**,**j**) and corresponding graph representation (**g**,**i**,**k**) for the whole image sequence, where objects follow FBM (**f**,**g**), ATTM (**h**,**i**) and CTRW (**j**,**k**), with $\alpha = 1$. Nodes are colour-coded with respect to the value of axial coordinate z (white, low; dark orange, high). In the graphs, the edges depict the association network used to infer dynamic properties without trajectory linking. Scale bars, 20 px. **l**, Confusion matrix showing how the network classifies the underlying diffusion model presented in 1,496 validation videos. Column-based normalization is applied. **m**, MAGIK predicts the anomalous diffusion exponent governing the motion of ensembles of objects performing FBM in 1,097 holographic videos. The probability distribution of the predicted versus ground-truth anomalous diffusion exponent (α) exhibits a good performance throughout the evaluated range.

MAE = 0.0538 ± 0.0022 for the complete model), pointing towards the importance of absolute coordinates for capturing spatial dynamics through the calculation of further parameters (for example, directionality) beyond Euclidean distance.

The use of gated self-attention offers a feature-wise discriminatory power to the node update operation as it weights individual features of the attention node embedding with respect to their importance to the overall graph structure. Through this mechanism, MAGIK identifies only the meaningful features of each node. This leads MAGIK to apply non-uniform attention over the neighbourhood[38] and to differentially consider the contribution from nodes of the same trajectory with respect to other neighbouring nodes belonging to different objects (Fig. 6).

Related works

Among the tasks explored in this work, the trajectory linking in biological systems is undoubtedly the most popular and has been tackled with a variety of methods[6,7]. These methods typically employ Kalman filter[39], multiframe and/or multitrack optimization based on greedy algorithms that approximate the multiple-hypothesis tracking solution[36,40,41] or combinatorial optimization[42]. Most of these approaches offer their best performance when knowledge of the motion is explicitly used[6].

Recently, deep-learning approaches have also been introduced to track biological objects using recurrent neural networks[43,44] and long short-time memory networks[45]. From a computer vision point of view, trajectory linking is equivalent to what is generally referred to as data association in multi-object tracking. In this context, several approaches have used GNN to solve data association as an edge-classification problem[46] or to jointly optimize detection and data association[47,48]. More generally, the problem of classifying nodes in a graph has also been tackled using spectral graph convolutional neural networks[49]. Very recently, the leveraging of the standard transformer into the graph domains has produced state-of-the-art performance on a wide range of tasks[27].

MAGIK jointly processes spatial and temporal information in a static GNN. However, other approaches treat space and time differently[50,51]. In MAGIK, similar to other architectures, edge features are used together with node features in the aggregation of each node[25,49]. In these cases, unless a message-passing framework[16,25] or an attention layer are used[27], the edge features only propagate to the associated node. As MAGIK leverages edge information through a message-passing framework, we compared its performance with other methods using the same mechanism, namely a message-passing neural network[25] and a gated graph sequence neural network[52]. To assess differences in performance between the use of global and masked attention, we also compared MAGIK with a message-passing neural network having a graph transformer[37] as a node update function. The results of the comparison are shown in Extended Data Table 1. For all the tasks and datasets, these methods perform better than or in line with the baseline

https://doi.org/10.1038/s42256-022-00595-0

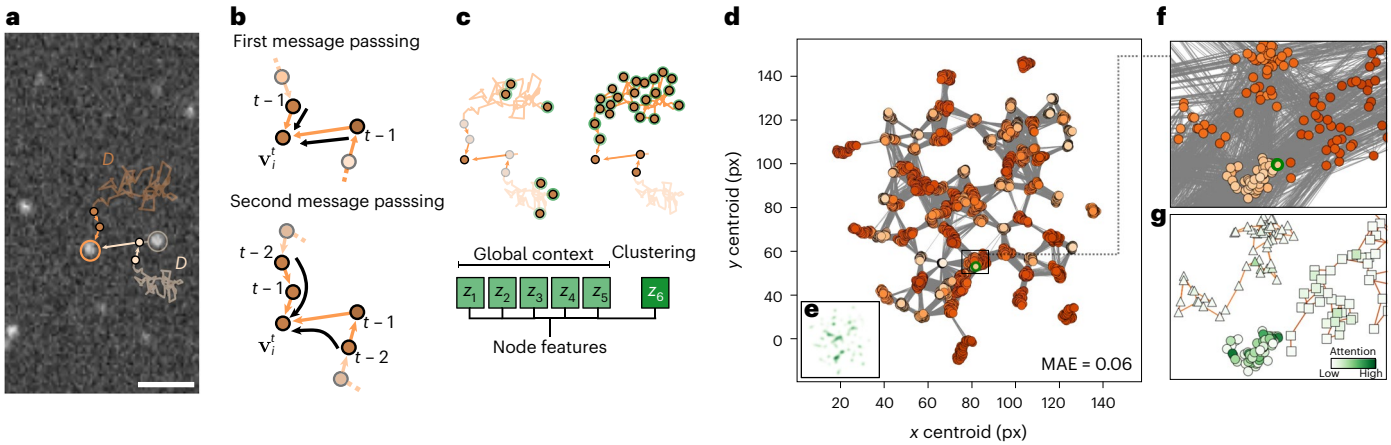


Fig. 6 | MAGIK attention mechanism. a, Exemplary frame from a video reproducing particles moving with different diffusion coefficients and their corresponding trajectories (segmented lines). Edges and nodes corresponding to detections of the two particles at the current (large circles) and at two previous (small circles) frames are also shown. Scale bar, 20 px. **b**, The message-passing steps only propagate information to a target node ($\mathbf{v}_i^t$) from a limited number (in this example 2) of previous frames. **c**, The gated self-attention mechanism further encodes information from all nodes. The different attention heads z_i spatiotemporally cluster other nodes and differentially consider their influence on the target node. **d**, Example of a ground-truth graph. The edges depict the network of associations used to infer dynamic properties without direct linking. The nodes are colour-coded according to the value of the target feature, that is, the displacement scaling factor $\sqrt{2D}$. The green circle highlights the target node, with respect to which the attention values shown in **e** and **g** are calculated. **e**, Attention map (heads 1–6) corresponding to the graph in **b** calculated with respect to the target node. **f**, Zoom-in of the rectangular region in **b**. **g**, Ground-truth trajectories (symbols and lines) corresponding to the graph in **f**. The symbols are colour-coded according to the value of the attention head 6 calculated with respect to the target node. Independently from the spatial distance, nodes from the same trajectory have a larger influence on the reference node.

but are outperformed by the full MAGIK. MAGIK performance is quite striking for the node-regression task, once more stressing the importance of the attention layer for this problem.

Discussion

MAGIK is a versatile framework for the characterization of dynamic properties from time-lapse microscopy that exploits geometric deep-learning capability to capture the full spatiotemporal complexity of biological experiments. MAGIK strongly relies on an attention-based GNN that can extract dynamic parameters from image-based features by assuming relational constraints between objects.

The examples analysed in this work highlight the wide versatility of MAGIK in different biological contexts. Remarkably, the same architecture can be applied to investigate other observables, can be trained to simultaneously estimate several parameters, and can even be used for applications beyond time-lapse microscopy, where time is substituted by another variable.

MAGIK provides a key enabling technology to estimate dynamic parameters from segmentation/localization in a complete linking-free fashion, whereas other methods require some level of knowledge about the linking between objects[53,54]. As such, it provides a powerful solution for those experiments where trajectory linking cannot be reliably performed, for example, as a consequence of high object density or probe blinking.

Methods

Description of MAGIK

The input to MAGIK is the graph representation of the movement and interactions of an ensemble of objects. The nodes ($\mathcal{V}$) contain features encoding meaningful information about the objects, whereas the edges ($\mathcal{E}$) connect spatiotemporally neighbouring nodes codifying relational features, such as the Euclidean distance between them (Fig. 1a,b). To improve efficiency, each node is connected to a limited number of spatial neighbours at subsequent frames. This is achieved through the choice of two parameters that set the maximum distance at which nodes must be located in space and time to be connected. We thus obtain a directed graph that is generally not fully connected. For most of the examples discussed in this work, we connect nodes within a maximum distance equal to 20% of the full field of view and up to 2 frames in the future. Exceptions are the reconstruction of the spatially varying diffusion map for which the maximum distance was set to 12% of the full field of view and the time-varying diffusion (segmentation) for which we connected nodes up to 4 frames in the future.

The architecture comprises three main blocks. First, an encoder neural network $\phi_v : \mathbb{R}^l \to \mathbb{R}^{l'}$ converts each node feature representation $\mathbf{v}_i \in \mathcal{V}$ of dimension l into an l'-dimensional feature representation $\mathbf{v}'_i$ (Fig. 1c). In parallel, another encoder neural network function $\phi_e : \mathbb{R}^f \to \mathbb{R}^{f'}$ transforms each edge feature $\mathbf{e}_k \in \mathcal{E}$ into a high-level feature vector $\mathbf{e}'_k$ of dimension f' (Fig. 1d). ϕ_v and ϕ_e are a series of multilayer perceptrons (MLPs) composed of a linear layer followed by a Gaussian error linear unit[55] as activation function and layer normalization.

Second, the resultant graph representation $\mathcal{G} = \{\mathcal{V}', \mathcal{E}'\}$ (Fig. 1e) is processed through repeated fingerprinting graph blocks (FGNN, described in detail in Extended Data Fig. 1). Each FGNN updates each edge in the graph by applying an MLP to the concatenation of the features of two neighbouring nodes and their connecting edge, that is

$$\mathbf{e}''_{ij} = \mathrm{MLP}\left(\left[\mathbf{v}'_i, \mathbf{v}'_j, \mathbf{e}'_{ij}\right]\right) \tag{1}$$

for $j \in \mathcal{N}_i$, where $\mathcal{N}_i$ is the neighbourhood of node i, and $[\cdot]$ represents the concatenation operation (Extended Data Fig. 1b). Subsequently, the learned representation $\mathbf{e}''_{ij}$ (of dimension f') is weighted by a Gaussian attention mechanism

$$w_{ij} = \exp\left(-\left(\frac{d_{ij}^2}{2\sigma^2}\right)^{\beta}\right) \tag{2}$$

where d_{ij} is the Euclidean distance between the centroids of the nodes i and j, and the standard deviation σ and the Gaussian order β are learnable parameters that allow the FGNN to adapt to varied object dynamics (Extended Data Fig. 1c,d). The FGNN computes a local representation

for the topological neighbourhood $\mathcal{N}_i$ by applying a linear transformation to the concatenation of the current state of node i and the aggregate of the weighted edge features, according to

$$\tilde{\mathbf{h}}_i = \mathbf{W}_{\tilde{H}}\left[\mathbf{v}'_i, \sum_{j\in\mathcal{N}_i} w_{ij}\mathbf{e}''_{ij}\right] \tag{3}$$

where $\mathbf{W}_{\tilde{H}}$ is an $l' \times (l' + f')$ linear projection matrix. The $\tilde{\mathbf{h}}_i$ are stored in a local representation matrix $\tilde{\mathbf{H}}$. Importantly, we prepend to this matrix a learnable node embedding $\mathbf{U} \in \mathbb{R}^{l'}$ through row-wise concatenation, that is, $\mathbf{H} = [\mathbf{U}; \tilde{\mathbf{H}}]$, whose state serves as a graph-level representation (Extended Data Fig. 1e)[26]. Finally, gated self-attention layers[30] are used to update the hidden states of the node features

$$\begin{aligned}\mathbf{V}''^{(z)} &= \mathrm{attn}^{(z)}(\mathbf{H}) \\ &= \mathbf{G}^{(z)} \odot \left(\mathrm{softmax}\left(\frac{1}{\sqrt{c}}\mathbf{Q}^{(z)}\mathbf{K}^{(z)\top}\right)\mathbf{P}^{(z)}\right)\end{aligned} \tag{4}$$

where $z = 1, \cdots, Z$, with Z representing the number of attention heads; $\mathbf{Q}^{(z)} = \mathbf{H}\mathbf{W}_Q^{(z)}$, $\mathbf{K}^{(z)} = \mathbf{H}\mathbf{W}_K^{(z)}$ and $\mathbf{P}^{(z)} = \mathbf{H}\mathbf{W}_P^{(z)}$ are the queries, key and values, embedding matrices of dimension c obtained by the $l' \times l'$ linear projection matrices $\mathbf{W}_Q^{(z)}$, $\mathbf{W}_K^{(z)}$ and $\mathbf{W}_P^{(z)}$, respectively; $\mathbf{G}^{(z)} = \sigma\left(\mathbf{H}\mathbf{W}_G^{(z)}\right)$ is the gate vector parameterized by the linear projection matrix $\mathbf{W}_G^{(z)} \in \mathbb{R}^{l'\times l'}$, followed by an element-wise sigmoid function σ; $\odot$ denotes the Hadamard product; and softmax normalizes the self-attention weights to be positive and add up to 1. The multi-head outputs $\mathbf{V}''^{(z)}$ are concatenated and passed through a MLP to capture nonlinear interactions between the node features to provide the set of updated node embeddings $\mathcal{V}''$ (Extended Data Fig. 1f). Note that $\mathbf{U}'$ needs to be retrieved from $\mathcal{V}''$ to obtain the updated global features.

Third, the final node ($\mathcal{V}''$), edge ($\mathcal{E}''$) and global features ($\mathbf{U}'$) are decoded to obtain node, edge and global-level predictions. The node features $\mathcal{V}''$ are processed using the decoding neural network φ_v to obtain predictions for nodes. Similarly, the decoder neural network φ_e receives $\mathcal{E}''$ and yields a prediction for each edge in the graph. φ_v and φ_e are reflections of the encoder networks ϕ_v and ϕ_e, respectively, with an additional (prediction) layer comprising a linear transformation tailed by an output activation function (for example, softmax or logistic sigmoid for classification problems, or linear activation for regression tasks). To compute global attributes, $\mathbf{U}'$ is processed by φ_u, an MLP followed by a linear layer and a task-dependent nonlinear activation.

To demonstrate the versatility of MAGIK, we use the same model architecture for all examples. The encoding neural networks ϕ_v and ϕ_e consist of a series of MLPs of dimensions 32, 64 and 96, respectively. The latent dimension for nodes and edges (that is, $l' = f' = 96$) is maintained across two FGNN layers in the trunk of the network and is chosen such that it is divisible by the number of self-attention heads in each layer ($Z = 6$ or $Z = 12$). The global embedding vector U is zero-initialized. The node and edge decoding neural networks φ_v and φ_e consist of three MLPs of dimensions 96, 64 and 32, followed by a final linear layer and an activation function that map the decoded node and edge features to the output dimension. φ_u consists of a 64-dimensional MLP followed by a linear output layer and an activation function that returns the global-level predictions.

MAGIK training

Once the network architecture is defined, MAGIK is trained using a set of graph feature representations and task-dependent targets. The input graphs follow the same relational structure regardless of the task, with nodes describing object detections and edges connecting the objects in time and space. Targets, in turn, represent different parameters depending on the specific task.

For trajectory linking (Figs. 2 and 3), MAGIK is trained to predict the probability of having a connection/link between two objects. This task is modelled as an edge-classification problem with a binary label (linked, labelled with 1, or unlinked, labelled with 0). Thus, during training, the network aims at minimizing the binary cross-entropy between the predicted probabilities and the ground-truth label for each edge. Accordingly, φ_e uses a sigmoid function as the final activation to produce probability estimates. For the training of the linking task, we use a single annotated video for each dataset/cell type, from which we stochastically extract 512 samples as sequences of consecutive frames with a duration of 10% to 20% of the whole video duration. Graphs are created using features obtained from video segmentation as described in Fig. 1a. Object coordinates are augmented by translations, rotations and mirroring. Other object descriptors are augmented by adding random noise to their values. To account for missed detections, we assign nodes a random number between 0 and 1 and remove those with values smaller than 0.05, together with the associated connections. For all the trajectory linking examples, the network is trained for 100 epochs using the 512 training samples processed in batches of 8 samples per iteration. Network performance was evaluated on samples extracted from different videos with respect to those used for training.

The inference of local properties is modelled as a node-regression problem (Fig. 4a–d), where MAGIK is trained to minimize the MAE between node predictions and ground truth. As a target feature, we use either the diffusion coefficient (Fig. 4b–d) or the anomalous diffusion exponent (Extended Data Figure 3b–d) of the object at the node level. Here, φ_v uses a linear activation function as the output activation. For the training, we generate a dataset of 2,000 videos with a duration of 50 to 55 frames corresponding to heterogeneous sets of moving objects (further details are provided in the 'Simulations' section). At each epoch, 1,024 samples are randomly extracted from the training dataset and their graph representations are augmented by translations, rotations and mirroring of the nodes' centroids. The network was trained for 100 epochs, processing the 1,024 samples in batches of 8 per iteration. Network performance was evaluated on independently simulated samples obtained using a different seed with respect to the training.

The quantification of global dynamic properties requires MAGIK to be trained to estimate global-level attributes from the input graphs. We have approached this problem from different perspectives: a classification problem to determine the underlying diffusion model of a set of particles (Fig. 4e–l) and a regression problem to estimate the relative fraction of objects moving according to different diffusion modes (Extended Data Fig. 3e–h). For classification tasks, the network is trained to minimize the sparse categorical cross-entropy between class predictions and ground-truth labels, with a softmax as the output activation of φ_u. For regression tasks, MAGIK minimizes the MAE between the network estimates and the target features and φ_u uses a linear activation function as the output activation. As target features, we use either class labels (for classification tasks) or continuous features (for regression tasks). In each of these examples, the training data come from 2,000 simulated videos. At each epoch, 1,024 samples are randomly extracted from the training dataset from which we extract graph representations and augment their topological structure by translations, rotations and mirroring of the nodes' centroids. The network was trained for 100 epochs, processing 1,024 samples in batches of 8 per iteration. Network performance was evaluated on independently simulated samples obtained using a different seed with respect to the training.

For all examples, the trainable parameters of MAGIK (that is, the weights of the artificial neurons in the neural networks and the parameters of Gaussian edge weighting function) were iteratively optimized using the backpropagation algorithm[56] and Adam optimizer (with a learning rate of 0.001)[57]. The training time of MAGIK ranges between 1 min and 5 min for trajectory linking and from 30 min to 60 min in the case of node and global-level regression on an NVIDIA A100 GPU (40 GB VRAM, 2,430 MHz effective core clock, 6,912 CUDA cores).

The capability of training using a minimal amount of annotated data without requiring prior knowledge of the underlying dynamics enables the application of MAGIK to a wide range of real data for which large labelled datasets are not available. In addition, we have also tested the possibility to perform transfer learning for the linking task between migration experiments employing different cell types, as shown in the GitHub repository.

Postprocessing algorithm for trajectory linking

Cell trajectories are built from the scores obtained for the edge-classification problem through a simple postprocessing algorithm. The algorithm starts from a random node at the initial frame $t = 0$ and connects it over time with other nodes at subsequent frames, considering only edges that have been classified as 'linked' by MAGIK. If no 'linked' edges connect the sender node at t with any receiver nodes at $t+1$, the algorithm checks future frames, until a maximum time lag. If no 'linked' edges are found within this lag, the trajectory is interrupted. If a sender node has two 'linked' edges connecting it to two receiver nodes at a later frame, the event is identified as a division. At this point, the algorithm treats the two nodes as independent and attempts to build two new trajectories. In the rare event that more than two 'linked' edges originate from the same sender, the one connecting the farthest receiver is dropped. The procedure is iterated until all the 'linked' edges have been taken into account. The datasets used in this work for the linking task did not contemplate merging events; therefore, this capability was not included in the postprocessing step used for the analysis. For other tasks and purposes, MAGIK output might be postprocessed with alternative algorithms that implement merging capabilities.

Quantification of cell-tracking results

Quantification of the method performance for cell tracking was obtained by calculating the TRA metric based on the acyclic oriented graph matching (AOGM) measure discussed in ref. 31. First, images corresponding to the incomplete cell segmentation provided for the 6th Cell Tracking Challenge were annotated according to their ground truth and then transformed into an acyclic oriented graph according to the instructions for participation in the challenge[7]. A similar graph was also obtained for the trajectories predicted by our methods. The quantification of the matching between the two graphs performed by the AOGM corresponds to the weighted sum of the executed operations to transform the predicted graph into the ground-truth one[31]. For this, we used the AOGM-A measure, which corresponds to the AOGM measure calculated by keeping only the edge-related weights positive ($w_{NS} = w_{FN} = w_{FP} = 0$; $w_{ED} = 1$, $w_{EA} = 1.5$, $w_{EC} = 1$)[31]. The AOGM-A thus evaluates the ability of an algorithm to follow objects in time (that is, its linking capability). The AOGM-A measure is normalized to obtain the tracking accuracy (TRA):

$$\mathrm{TRA} = 1 - \frac{\min(\mathrm{AOGM\text{-}A}, \mathrm{AOGM\text{-}A_0})}{\mathrm{AOGM\text{-}A_0}} \quad (5)$$

where AOGM-A_0 corresponds to the cost of linking the graph from scratch (that is, the cost of adding all the edges multiplied by the corresponding weights). The normalization bounds TRA in the interval [0, 1], with higher values corresponding to better tracking performance.

Simulations

Trajectories were simulated using the andi-datasets Python package[58]. In addition, we used DeepTrack 2.1[59] to render imaged objects in different illumination modalities (fluorescence and holographic microscopy) reproducing optical conditions to provide realistic node features (Fig. 4). The localizations' crowding was estimated by $c = \rho D \Delta t$, an adimensional parameter in two dimensions that simultaneously accounts for changes in the number density ρ, diffusion coefficient D and sampling time Δt.

For the fluorescence microscopy experiments of Fig. 4a–d, we simulated objects performing Brownian motion in two dimensions with random diffusivities ($0.005 \leq D \leq 0.7$). For Fig. 4e–h, the diffusivity was defined by a random spatial map, smoothed with a Gaussian filter. For training, we typically use videos of 50–55 frames containing 30–35 objects for the inference of D and 70–80 frames for the diffusivity maps, initially positioned at random locations on 32×32 px^2. The localizations' crowding was estimated by $c = \rho D \Delta t$, an adimensional parameter in two dimensions that simultaneously accounts for changes in the number density ρ, diffusion coefficient D and sampling time Δt. Due to the variability of D and of the number of objects, the crowding factor for these examples could vary from video to video in the range [0.003, 0.04]. Each object was rendered as a diffraction-limited spot through the optics module of DeepTrack 2.1[59], with a random intensity from a uniform distribution between 20 and 80 counts, varying over time with a standard deviation of 3 counts.

For all the experiments of Fig. 5, we generated trajectories undergoing three different diffusion models, namely FBM, ATTM and CTRW, with a constant anomalous exponent $\alpha = 1$ and random diffusivity. For Fig. 5a–e, each object in the video undergoes two-dimensional diffusion with a randomly assigned model, with all other properties (sequence length, number of particles, intensity) being the same as described for the data in Fig. 4.

For the plankton trajectories illustrated in Fig. 5f–m, all microorganisms in the same video move according to the same three-dimensional model, varying from video to video. We generated holographic videos of 100 frames including 3–7 microorganisms, each with a randomly sampled refractive index from a uniform distribution between 1.35 and 1.55, covering a wide variety of plankton species in the literature[60].

For the data of Extended Data Fig. 2, we generated trajectories undergoing Brownian motion with random diffusivity drawn from an exponential distribution with an average of 0.1 px^2 per frame (truncated at 0.001 and 1 px^2 per frame) and with a random intensity from a uniform distribution between 20 and 80 counts, varying over time with a standard deviation of 3 counts. The diffusivity was kept constant over dwell times extracted from a geometrical distribution with $p = 0.05$ truncated at values >5 frames. Sequence length was 400 frames.

For the examples in Extended Data Fig. 3a–d, we simulated objects performing FBM in two dimensions with random anomalous exponents ($0.2 \leq \alpha < 1.8$) and diffusivities ($0.005 \leq D < 0.7$). For the examples illustrated in Extended Data Fig. 3e–h, we generated fluorescence images of objects undergoing FBM in two dimensions in sub-diffusive ($0.2 \leq \alpha \leq 0.6$), normal ($\alpha = 1$) and super-diffusive mode ($1.4 \leq \alpha \leq 1.8$). All other properties (sequence length, number of particles, intensity) are the same as described for the data in Fig. 4.

For the data of Extended Data Fig. 4, we generated trajectories undergoing FBM with a constant anomalous exponent $\alpha = 1$ and random diffusivity drawn from an exponential distribution with an average of 0.1 px^2 per frame (truncated at 0.001 and 1 px^2 per frame) and with a random intensity from a uniform distribution between 20 and 80 counts, varying over time with a standard deviation of 3 counts. Particles undergo diffusion with reflecting boundaries in a square box with a side of 32×32 px^2. The number of particles was kept constant at 30. Trajectories were generated at sampling times $\Delta t = 0.5, 1, 2, 4, 8, 16, 32$ corresponding to crowding factor $c = 0.0015, 0.0029, 0.0059, 0.0117, 0.0234, 0.0468, 0.0936$. Sequence length was 100 frames.

Data availability

The cell tracking datasets were obtained from the Cell Tracking Challenge webpage http://celltrackingchallenge.net/2d-datasets/, where they can be accessed from. Further datasets and examples to run the code are publicly available at the DeepTrack 2.1 GitHub repository[59]. Source data are provided with this paper.

Code availability

All source code is made publicly available at the DeepTrack 2.1 GitHub repository[59].

References

1. Brückner, D. B. et al. Learning the dynamics of cell–cell interactions in confined cell migration. *Proc. Natl Acad. Sci. USA* **118**, e2016602118 (2021).
2. Ladoux, B. & Mège, R.-M. Mechanobiology of collective cell behaviours. *Nat. Rev. Mol. Cell Biol.* **18**, 743–757 (2017).
3. Ramos, C. H. et al. The environment topography alters the way to multicellularity in *Myxococcus xanthus*. *Sci. Adv.* **7**, eabh2278 (2021).
4. Manzo, C. & Garcia-Parajo, M. F. A review of progress in single particle tracking: from methods to biophysical insights. *Rep. Prog. Phys.* **78**, 124601 (2015).
5. Shen, H. et al. Single particle tracking: from theory to biophysical applications. *Chem. Rev.* **117**, 7331–7376 (2017).
6. Chenouard, N. et al. Objective comparison of particle tracking methods. *Nat. Methods* **11**, 281–289 (2014).
7. Ulman, V. et al. An objective comparison of cell-tracking algorithms. *Nat. Methods* **14**, 1141–1152 (2017).
8. Tinevez, J.-Y. et al. Trackmate: an open and extensible platform for single-particle tracking. *Methods* **115**, 80–90 (2017).
9. Sarkar, R., Mukherjee, S., Labruyère, E. & Olivo-Marin, J.-C. Learning to segment clustered amoeboid cells from brightfield microscopy via multi-task learning with adaptive weight selection. In *2020 25th International Conference on Pattern Recognition (ICPR)* 3845–3852 (IEEE, 2021).
10. Helgadottir, S., Argun, A. & Volpe, G. Digital video microscopy enhanced by deep learning. *Optica* **6**, 506–513 (2019).
11. Berg, S. et al. Ilastik: interactive machine learning for (bio) image analysis. *Nat. Methods* **16**, 1226–1232 (2019).
12. Midtvedt, B. et al. Quantitative digital microscopy with deep learning. *Appl. Phys. Rev.* **8**, 011310 (2021).
13. Ershov, D. et al. TrackMate 7: integrating state-of-the-art segmentation algorithms into tracking pipelines. *Nat. Methods* **19**, 829–832 (2022).
14. Muñoz-Gil, G. et al. Objective comparison of methods to decode anomalous diffusion. *Nat. Commun.* **12**, 6253 (2021).
15. Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A. & Vandergheynst, P. Geometric deep learning: going beyond Euclidean data. *IEEE Signal Process. Mag.* **34**, 18–42 (2017).
16. Battaglia, P. W. et al. Relational inductive biases, deep learning, and graph networks. Preprint at https://arxiv.org/abs/1806.01261 (2018).
17. Liu, K. et al. Chemi-net: a molecular graph convolutional network for accurate drug property prediction. *Int. J. Mol. Sci.* **20**, 3389 (2019).
18. Stokes, J. M. et al. A deep learning approach to antibiotic discovery. *Cell* **180**, 688–702 (2020).
19. Somnath, V. R., Bunne, C., Coley, C., Krause, A. & Barzilay, R. Learning graph models for retrosynthesis prediction. *Adv. Neural Inf. Process. Syst.* **34**, 9405–9415 (2021).
20. Mohamed, A., Qian, K., Elhoseiny, M. & Claudel, C. Social-STGCNN: a social spatio-temporal graph convolutional neural network for human trajectory prediction. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition* 14424–14432 (IEEE, 2020).
21. Strogatz, S. H. Exploring complex networks. *Nature* **410**, 268–276 (2001).
22. Kipf, T., Fetaya, E., Wang, K.-C., Welling, M. & Zemel, R. Neural relational inference for interacting systems. In *International Conference on Machine Learning* 2688–2697 (PMLR, 2018).
23. Löffler, K., Scherr, T. & Mikut, R. A graph-based cell tracking algorithm with few manually tunable parameters and automated segmentation error correction. *PLoS ONE* **16**, e0249257 (2021).
24. Verdier, H. et al. Learning physical properties of anomalous random walks using graph neural networks. *J. Phys. A* **54**, 234001 (2021).
25. Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O. & Dahl, G. E. Neural message passing for quantum chemistry. In *International Conference on Machine Learning* 1263–1272 (PMLR, 2017).
26. Dosovitskiy, A. et al. An image is worth 16x16 words: transformers for image recognition at scale. Preprint at https://arxiv.org/abs/2010.11929 (2020).
27. Ying, C. et al. Do transformers really perform badly for graph representation? *Adv. Neural Inf. Process. Syst.* **34**, 28877–288888 (2021).
28. Chang, M. B., Ullman, T., Torralba, A. & Tenenbaum, J. B. A compositional object-based approach to learning physical dynamics. Preprint at https://arxiv.org/abs/1612.00341 (2016).
29. Zhang, K., Zhu, Y., Wang, J. & Zhang, J. Adaptive structural fingerprints for graph attention networks. In *International Conference on Learning Representations* (ICLR, 2019).
30. Jumper, J. et al. Highly accurate protein structure prediction with alphafold. *Nature* **596**, 583–589 (2021).
31. Matula, P. et al. Cell tracking accuracy measurement based on comparison of acyclic oriented graphs. *PLoS ONE* **10**, e0144959 (2015).
32. Qi, X., Xing, F., Foran, D. J. & Yang, L. Robust segmentation of overlapping cells in histopathology specimens using parallel seed detection and repulsive level set. *IEEE Trans. Biomed. Eng.* **59**, 754–765 (2011).
33. He, Y. et al. iCut: an integrative cut algorithm enables accurate segmentation of touching cells. *Sci. Rep.* **5**, 12089 (2015).
34. Winter, M. et al. Separating touching cells using pixel replicated elliptical shape models. *IEEE Trans. Med. Imaging* **38**, 883–893 (2018).
35. Mukherjee, S., Sarkar, R., Manich, M., Labruyère, E. & Olivo-Marin, J.-C. Domain adapted multi-task learning for segmenting amoeboid cells in microscopy. *IEEE Trans. Med. Imaging* (2022).
36. Jaqaman, K. et al. Robust single-particle tracking in live-cell time-lapse sequences. *Nat. Methods* **5**, 695–702 (2008).
37. Dwivedi, V. P. & Bresson, X. A generalization of transformer networks to graphs. Preprint at https://arxiv.org/abs/2012.09699 (2020).
38. Trivedi, R., Farajtabar, M., Biswal, P. & Zha, H. Dyrep: Learning representations over dynamic graphs. In *International Conference on Learning Representations* (National Science Foundation, 2019).
39. Godinez, W. J., Lampe, M., Eils, R., Müller, B. & Rohr, K. Tracking multiple particles in fluorescence microscopy images via probabilistic data association. In *2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro* 1925–1928 (IEEE, 2011).
40. Chenouard, N., Bloch, I. & Olivo-Marin, J.-C. Multiple hypothesis tracking in microscopy images. In *2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro* 1346–1349 (IEEE, 2009).
41. Coraluppi, S. & Carthel, C. Multi-stage multiple-hypothesis tracking. *J. Adv. Inf. Fusion* **6**, 57–67 (2011).
42. Sbalzarini, I. F. & Koumoutsakos, P. Feature point tracking and trajectory analysis for video imaging in cell biology. *J. Struct. Biol.* **151**, 182–195 (2005).
43. Spilger, R. et al. A recurrent neural network for particle tracking in microscopy images using future information, track hypotheses, and multiple detections. *IEEE Trans. Image Process.* **29**, 3681–3694 (2020).

44. Spilger, R. et al. Deep probabilistic tracking of particles in fluorescence microscopy images. *Med. Image Anal.* **72**, 102128 (2021).
45. Yao, Y., Smal, I., Grigoriev, I., Akhmanova, A. & Meijering, E. Deep-learning method for data association in particle tracking. *Bioinformatics* **36**, 4935–4941 (2020).
46. Lee, J., Jeong, M. & Ko, B. C. Graph convolution neural network-based data association for online multi-object tracking. *IEEE Access* **9**, 114535 (2021).
47. Gao, J., Zhang, T. & Xu, C. Graph convolutional tracking. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition* 4649–4659 (IEEE, 2019).
48. Wang, Y., Kitani, K. & Weng, X. Joint object detection and multi-object tracking with graph neural networks. In *2021 IEEE International Conference on Robotics and Automation (ICRA)* 13708–13715 (IEEE, 2021).
49. Kipf, T. N. & Welling, M. Semi-supervised classification with graph convolutional networks. Preprint at https://arxiv.org/abs/1609.02907 (2016).
50. Yu, B., Yin, H. & Zhu, Z. Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In *Proc. 27th International Joint Conference onArtificial Intelligence* 3634–3640 (2018).
51. Nicolicioiu, A., Duta, I. and Leordeanu, M. Recurrent space-time graph neural networks. *Adv. Neural Inf. Process. Syst.* **32**, 12838–12850 (2019).
52. Li, Y., Tarlow, D., Brockschmidt, M. & Zemel, R. Gated graph sequence neural networks. Preprint at https://arxiv.org/abs/1511.05493 (2015).
53. El Beheiry, M., Dahan, M. & Masson, J.-B. Inferencemap: mapping of single-molecule dynamics with Bayesian inference. *Nat. Methods* **12**, 594–595 (2015).
54. Xiang, L., Chen, K., Yan, R., Li, W. & Xu, K. Single-molecule displacement mapping unveils nanoscale heterogeneities in intracellular diffusivity. *Nat. Methods* **17**, 524–530 (2020).
55. Hendrycks, D. & Gimpel, K. Gaussian error linear units (GELUs). Preprint at https://arxiv.org/abs/1606.08415 (2016).
56. McClelland, J. L. et al. *Parallel Distributed Processing, Volume 2: Explorations in the Microstructure of Cognition: Psychological and Biological Models* Vol. 2 (MIT Press, 1987).
57. Kingma, D. P. & Ba, J. Adam: a method for stochastic optimization. Preprint at https://arxiv.org/abs/1412.6980 (2014).
58. Muñoz-Gil, G., Requena, B., Volpe, G., Garcia-March, M. A. & Manzo, C. AnDiChallenge/ANDI_datasets: Challenge 2020 release. *Zenodo* https://doi.org/10.5281/zenodo.4775311 (2020).
59. Midtvedt, B., Pineda, J., Manzo, C. & Volpe, A. softmatterlab/deeptrack2: Deeptrack2 1.4.0 *Zenodo* https://doi.org/10.5281/zenodo.7175126 (2022).
60. Aas, E. Refractive index of phytoplankton derived from its metabolite composition. *J. Plankton Res.* **18**, 2223–2249 (1996).

Acknowledgements

This work was supported by the H2020 European Research Council (ERC) Starting Grant ComplexSwimmers (grant number 677511, received by G.V.); the Horizon Europe ERC Consolidator Grant MAPEI (grant number 101001267, received by G.V.); the Knut and Alice Wallenberg Foundation (grant number 2019.0079, received by G.V.); the grant RYC-2015-17896 (received by C.M.) funded by MCIN/AEI/10.13039/501100011033 and "ESF Investing in your future"; the grants BFU2017-85693-R and PID2021-125386NB-I00 (received by C.M.) funded by MCIN/AEI/10.13039/501100011033/ and "ERDF A way of making Europe"; the Generalitat de Catalunya (AGAUR grant number 2017SGR940, received by C.M.). C.M. acknowledges the support of NVIDIA Corporation with the donation of the Titan Xp GPU. We thank G. Muñoz-Gil, H. Klein and F. Skärberg for enlightening discussions, and AI Sweden for providing access to their computational resources.

Author contributions

C.M. and G.V. conceived the project. J.P. and C.M. designed the method. J.P., B.M. and S.N. implemented the architecture. J.P., H.B., B.M. and C.M. analysed the data and generated the figures. H.B., J.P. and C.M. carried out the simulations. J.P., G.V. and C.M. wrote the paper with input from all of the authors. D.M., G.V. and C.M. supervised the project.

Funding

Open access funding provided by University of Gothenburg.

Competing interests

J.P., B.M., D.M., G.V. and C.M. hold shares and/or stock options of the company IFLAI AB. The other authors declare no competing interests.

Additional information

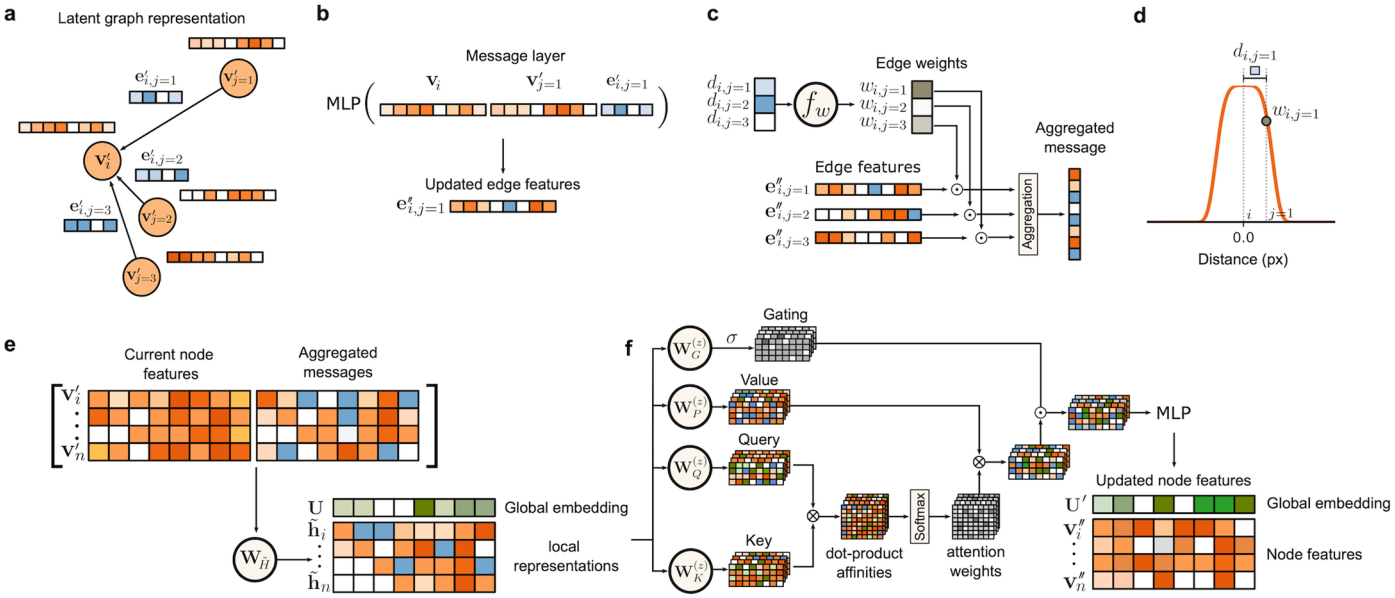
Extended data is available for this paper at https://doi.org/10.1038/s42256-022-00595-0.

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s42256-022-00595-0.

Correspondence and requests for materials should be addressed to Giovanni Volpe or Carlo Manzo.

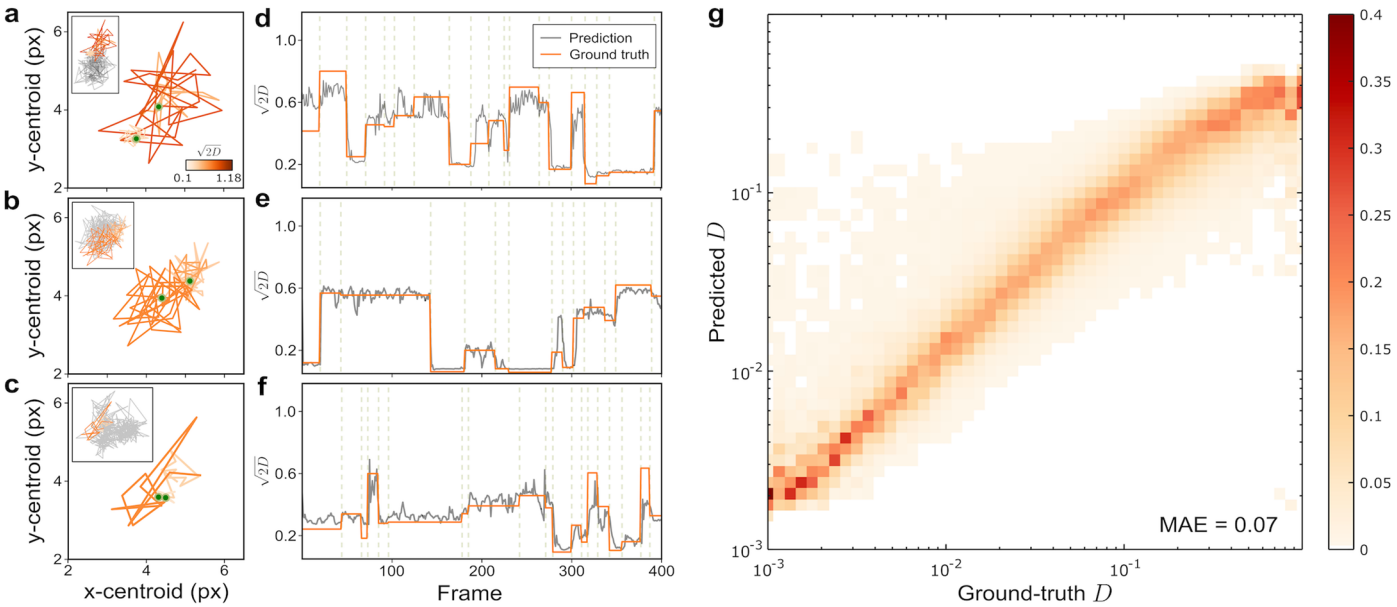
Peer review information *Nature Machine Intelligence* thanks Boris Knyazev, Suvadip Mukherjee, Bahare Fatemi and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

https://doi.org/10.1038/s42256-022-00595-0



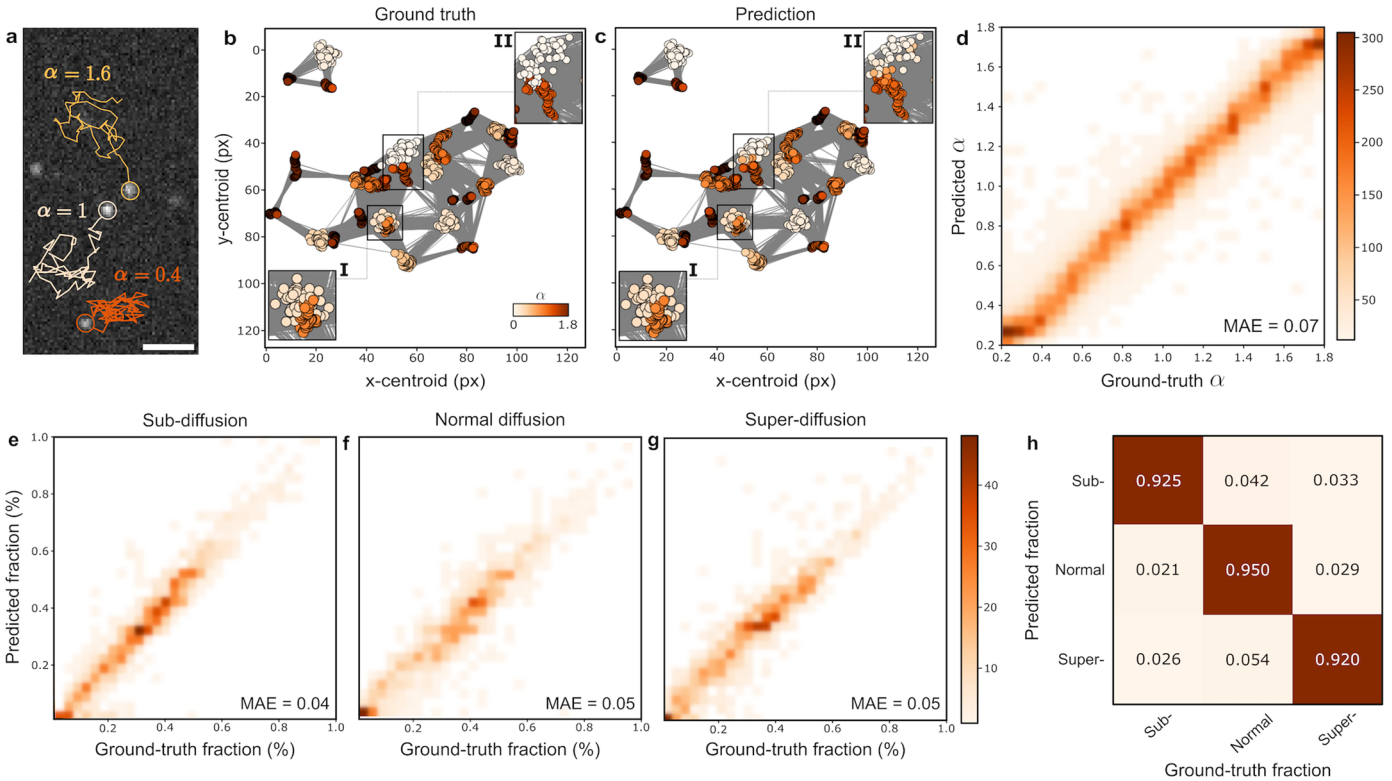
Extended Data Fig. 1 | Processing flow of the fingerprinting graph block (FGNN). FGNN, similar to other flavours of GNN layers, comprises three fundamental steps: edge feature update, edge feature aggregation, and node update. **a**, Input graph structure. Nodes contain features encoding the object's position and relevant descriptors. Edges encode relational features between neighbouring nodes. In this example, the node of interest, labelled with the subindex *i*, receives information from connected nodes, labelled with the subindex *j*. **b**, Each edge in the graph is updated by applying a multilayer perceptron (MLP) to the concatenation of the features of two nodes and the edge connecting them (equation (1)). **c**, During the aggregation of edge features to a node, the contribution of each edge has a weight that is determined by the distance between linked nodes using a function with free parameters, f_w (equation (2)). **d**, f_w is a super-Gaussian and defines a learnable local receptive field that allows the FGNN to adapt to heterogeneous dynamics. **e**, The current state of the nodes and the aggregate of the weighted edge features are concatenated and linearly transformed to obtain a local representation for each neighbourhood (equation (3)). Furthermore, the FGNN prepends a learnable node embedding **U** to the local representation matrix, whose features provide global system-level insights. **f**, The nodes are updated using gated self-attention layers. The matrix resulting from the concatenation of **U** with the local features is transformed by the trainable linear transformation matrices $\mathbf{Q}^{(z)}$, $\mathbf{K}^{(z)}$, $\mathbf{P}^{(z)}$ to obtain queries, key, and values, respectively. *z* denotes the index of the attention head. The self-attention weights are calculated by the dot-product of the queries with the key matrix. Softmax normalizes the weights to be positive and to add up to 1 (equation (4)). Finally, the weighted values are multiplied by the gatings and passed through an MLP to account for nonlinear interactions between nodes to obtain the updated node features.

https://doi.org/10.1038/s42256-022-00595-0



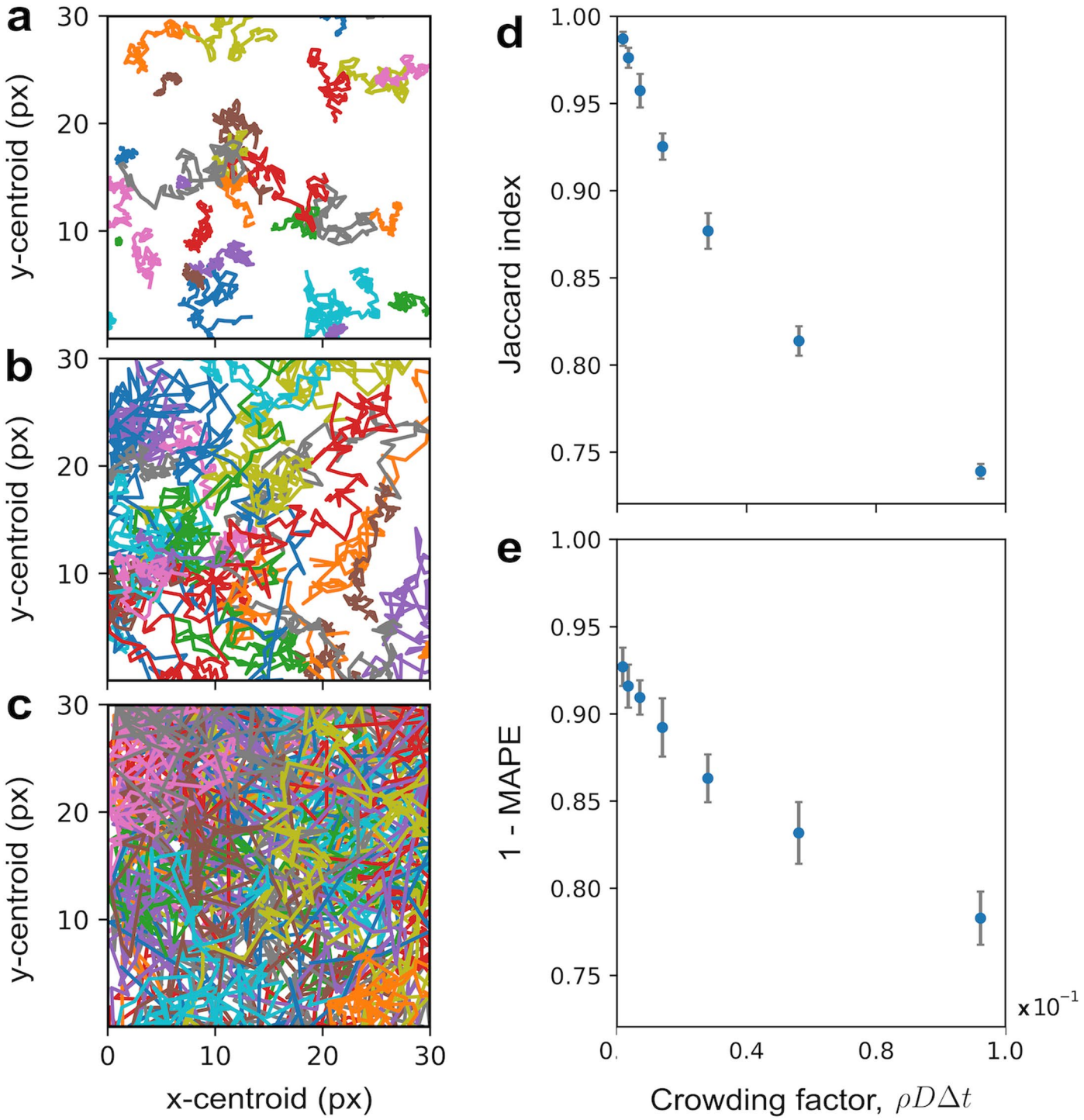
Extended Data Fig. 2 | MAGIK detects dynamic changes along individual trajectories. a-c, Portions of individual 2D trajectories undergoing changes of diffusion coefficient (circles). The colour of the segments represents the value of the displacement scaling factor $\sqrt{2D}$, with D being the diffusion coefficient. The full trajectories are reproduced in the insets, with the remainder of the trajectory depicted in grey. **d-f**, Time traces of the displacement scaling factor $\sqrt{2D}$. The ground-truth value used in the simulations is shown by the orange curve. The predictions obtained by MAGIK at the single-node level are shown in grey. **g**, Probability distribution of the predicted vs. ground-truth diffusion coefficient D, showing a good agreement (MAE = 0.07).

https://doi.org/10.1038/s42256-022-00595-0



Extended Data Fig. 3 | MAGIK estimates local and global anomalous diffusion properties at the ensemble and single-object levels. a, Simulated single-object tracking experiment. Fluorescence microscopy is used to follow the motion of single molecules characterized by a fractional Brownian motion (FBM) with varying anomalous diffusion exponent α. Scale bar = 20 px. **b-c**, Ground-truth and predicted graphs. Edges depict the network of associations used to directly infer dynamic properties without explicit linking. Nodes are colour-coded according to the value of the target feature α. The predicted node values agree with the ground truth also in crowded areas (for example, zoomed regions I and II). **d**, Probability distribution of the predicted vs. ground-truth anomalous diffusion exponent α. **e-h**, MAGIK estimates the relative fraction of objects following different diffusion modes, that is, sub- ($0.2 \leq \alpha \leq 0.6$), normal ($\alpha = 1$) and superdiffusion ($1.4 \leq \alpha \leq 1.8$). **e-g**, Probability distribution of predicted vs. ground-truth fraction for subdiffusion, normal diffusion, and superdiffusion, respectively. **h** Confusion matrix demonstrating how the network classifies the underlying diffusion model exhibited by objects in 1199 validation videos. Column-based normalization is applied, such as the sum along the columns adds up to 1, with minor deviations due to rounding.

https://doi.org/10.1038/s42256-022-00595-0



Extended Data Fig. 4 | MAGIK performance as a function of trajectory crowding. a-c, Examples of trajectories obtained at three different levels of crowding. The localizations' crowding was estimated by $c = \rho D \Delta t$, an adimensional parameter in 2D that simultaneously accounts for changes in the number density ρ, diffusion coefficient D, and sampling time Δt. Sequence length was 100 frames. Panels represent examples obtained for $c = 0.0015$ (**a-c**), $c = 0.0117$ (**b**), and $c = 0.0936$ (**c**). **d-e**, Performance obtained by MAGIK at varying the crowding factor for the trajectory linking (quantified through the Jaccard index, **d**) and the node-regression task (quantified through $1 - \frac{\text{MAPE}}{100}$, where MAPE represents the mean absolute percentage error, **e**). The performance for the linking task degrades faster ($\approx 25\%$ reduction) with respect to node regression ($\approx 15\%$ reduction) over the same range of c. Data in **d-e** correspond to averages (circles) and $3 \times \sigma$ (error bars) calculated over 5 runs.

https://doi.org/10.1038/s42256-022-00595-0

Extended Data Table 1 | Results of ablation study and methods' comparison. Ablated architectures are obtained by removing the learnable local receptive field (LLRF) and/or the gated self-attention (GSA) from MAGIK. Reported metrics correspond to averages over 20 runs for linking and over 5 runs for the other cases. Best-in-class performances are reported in bold. The dashed lines correspond to the metric values obtained for the baseline model

			Trajectory linking (DIC-C2DH-HeLa)	Node regression (Local diffusion properties)	Graph classification (Global diffusion properties)		
Model			Mean ± SD	Mean ± SD	Mean ± SD		
	LLRF	GSA			FBM	ATTM	CTRW
MAGIK	✓	✓	**0.9915 ± 0.0037**	**0.0538 ± 0.0022**	**0.8853 ± 0.1120**	**0.9137 ± 0.0928**	**0.8793 ± 0.1402**
ablated MAGIK	✗	✗	0.9845 ± 0.0077	0.1307 ± 0.0007	0.8586 ± 0.1362	0.8870 ± 0.1126	0.8705 ± 0.1394
	✓	✗	0.9856 ± 0.0051	0.1299 ± 0.0002	0.8798 ± 0.1126	0.9089 ± 0.1019	0.8771 ± 0.1383
MPNN [25]			0.9882 ± 0.0051	0.1306 ± 0.0007	0.8586 ± 0.1362	0.8870 ± 0.1126	0.8705 ± 0.1394
GGNN [37]			0.9894 ± 0.0062	0.1318 ± 0.0015	0.8593 ± 0.1338	0.8891 ± 0.1162	0.8676 ± 0.1452
Graph Transformer [38]			0.9889 ± 0.0034	0.1290 ± 0.0012	0.8552 ± 0.1311	0.8846 ± 0.1135	0.8575 ± 0.1519