Learning physical properties of anomalous random walks using graph neural networks

Hippolyte Verdier, Maxime Duval, François Laurent, Alhassan Cassé, Christian L. Vestergaard, Jean-Baptiste Masson

Learning physical properties of anomalous random walks using graph neural networks

Hippolyte Verdier[1,2,3,*], Maxime Duval[1], François laurent[1], Alhassan Cassé[2], Christian L. Vestergaard[1], and Jean-Baptiste Masson[1,*]

[*]*Correspondence should be addressed to hverdier@pasteur.fr & jbmasson@pasteur.fr*
[1]*Decision and Bayesian Computation, USR 3756 (C3BI/DBC) & Neuroscience department CNRS UMR 3751, Institut Pasteur, CNRS, Paris, France*
[2]*Histopathology and Bio-Imaging Group, Sanofi R&D, Vitry-Sur-Seine, France*
[3]*Université de Paris, UFR de physique, 75013 Paris, France*

March 22, 2021

Abstract

Single particle tracking allows probing how biomolecules interact physically with their natural environments. A fundamental challenge when analysing recorded single particle trajectories is the inverse problem of inferring the physical model or class of models of the underlying random walks. Reliable inference is made difficult by the inherent stochastic nature of single particle motion, by experimental noise, and by the short duration of most experimental trajectories. Model identification is further complicated by the fact that main physical properties of random walk models are only defined asymptotically, and are thus degenerate for short trajectories.

Here, we introduce a new, fast approach to inferring random walk properties based on graph neural networks (GNNs). Our approach consists in associating a vector of features with each observed position, and a sparse graph structure with each observed trajectory. By performing simulation-based supervised learning on this construct [1], we show that we can reliably learn models of random walks and their anomalous exponents. The method can naturally be applied to trajectories of any length. We show its efficiency in analysing various anomalous random walks of biological relevance that were proposed in the AnDi challenge [2]. We explore how information is encoded in the GNN, and we show that it learns relevant physical features of the random walks. We furthermore evaluate its ability to generalize to types of trajectories not seen during training, and we show that the GNN retains high accuracy even with few parameters. We finally discuss the possibility to leverage these networks to analyse experimental data.

Keywords: inverse problems, graphical models, random walks, amortised inference, single particle tracking, deep learning, graph neural networks.

1 Introduction

Random walks are encountered throughout a large variety of scientific domains, spanning diverse fields such as condensed matter physics [3], molecular biology [4], ecology [5], and finance [6]. The random walkers can be considered probes of their environment, and their recorded trajectories contain information on the properties of both the walkers and their environments. When analysing experimental data from such systems, one needs to solve the inverse problem of inferring an appropriate model or class of models and their parameters from a given set of trajectories.

Whether the recorded trajectories come from single-molecule experiments [7, 8] or from other types of experimental recordings [9–11], the inverse problem of identifying the nature (i.e. model class and parameters) of the random walks is a complex one. The reasons for this complexity are multiple: the inherent stochastic nature of random walks makes it difficult to identify stereotypical behaviours, characteristic features of different random walks models overlap for finite size trajectories, experimental noise can mask subtle local dynamics that could differentiate some models, and the distributions of selective features associated with many random walks are difficult to compute.

A continually growing number of approaches have been developed to tackle the task of identifying and characterising random walks from their individual realisations. A popular class of approaches consists in fitting temporal curves of moments of the displacements [12, 13]. The main such statistics is the mean square displacement (MSD), either estimated from an ensemble of random walks or evaluated for individual trajectories [12, 14–16]. The MSD is widely used and remains a tool of choice for analysis, both for historical reasons and for its simple interpretation [3] (see Section 2 below). The MSD however has several undesirable statistical properties which can make conventional analysis suboptimal. It is nonstationary, highly autocorrelated, and has a non-Gaussian distribution [17], making the design of efficient inference procedures difficult. Moreover, it may be insufficient on its own to distinguish between different random walk models, especially for heterogeneous systems [18, 19]. The inferences of the model class and associated parameters degrade with decreasing trajectory lengths, with increasing positioning noise and with positional correlations induced by the experimental measurement process. Many complementary approaches, based on predefined features extracted from trajectories, have been developed. Some are ensemble-based, relying on distributions of features evaluated in sets of trajectories, such as the distribution of tangent velocities [20], of displacements [21], of "excursions" [22], or of first time passage statistics [23]. Others are centred on individual trajectories, such as non-ergodicity estimators [24], renormalisation-group-based second moment analysis [25], mixed use of multiple estimators in decision trees [26], asymmetry samplers [27], or filtered temporally-averaged MSDs [28].

Statistical learning has provided new tools to address these inverse problems in a principled way. Hidden Markov Models (HMM) [29] are efficiently used to probe diffusion modes [30–32], detect confined-like motion [33], and mix local and global information to perform inference on very short trajectories [34]. (It is worth noting here that the temporally averaged MSD (TAMSD) can also be used in detecting diffusion mode changes [35].) Other Bayesian approaches have allowed to infer fBM characteristics [36–38] and to perform model selection between several different models [39].

The main drawback of statistical learning approaches is that they often require an analytical expression for the likelihood function. This limits the applicability of these approaches as many random walk models of interest do not have a tractable likelihood.

Recently, machine learning approaches have been developed to tackle this problem and to provide a broader class of random walks with efficient means for model selection and inference. To overcome the lack of ground truth annotations for experimental random walks, applications of machine learning generally involve a simulation-based approach, where a machine learning model is trained on simulated data from a selection of random walk models using supervised learning. Assuming that the selected training set contains models that describe the experimental system well enough, the learned model can then be transferred to classify and/or infer the parameters of experimental random walks. This differs from the typical approach to supervised learning where the model is trained on an annotated experimental dataset. Within this type of approaches, ensemble learning methods such as random forest have been found to be remarkably effective when applied to appropriately selected features of recorded trajectories [2, 40, 41]. Neural networks applied on windowed MSDs have also been found to be efficient for differentiating between modes of motion and detecting their transitions in time [42].

As larger datasets become available, deep learning approaches, which automatically learn features from the raw data, have consistently been found to outperform classic machine learning models relying on predefined features [43]. Following this trend, deep learning approaches have recently been developed to infer properties of random walks. Recurrent neural networks (RNNs), tools of choice for time series analysis, have been used to infer parameters associated to sub-diffusive behaviour [44] and to detect changes between different diffusion modes [45]. An approach based on a deep learning architecture search platform, leveraging convolutional neural networks (CNNs), LSTM networks (a class of RNNs), and the ResNet and InceptionTime architectures known for their efficiency for natural language processing (NLP) [46], has proven to be efficient even for short trajectories [40]. Another recent approach, inspired from generative audio generation models [47], leverages a temporal convolution network [48] where convolutions are applied with various dilation effect to detect long term correlations, has shown impressive performance in inferring the nature of random walks [49].

The limitations of deep learning approaches applied to random walks are similar to other applications of deep learning: identifying overfitting can be difficult, and learned models typically lack interpretability (they are usually referred to as black box models for this reason). Furthermore, the computational cost of training generally grows with the number of parameters and with the level of recurrence of the neural network.

The particular nature of random walk data and the simulation-based approach also adds difficulties when adapting to experimental data: varying trajectory lengths complicate the application of non-recurrent neural networks (e.g., CNN architectures), and varying experimental noise as well as mislinking between particle positions in experimental images may degrade statistical performance. Finally, inevitable differences between our random

walk models and the true data generating process (known as distribution drift, but here exacerbated by the simulation-based approach), makes it hard to define the conditions under which transfer learning performs well.

To address some of these problems, we here propose to leverage graph neural networks (GNNs) [50], a recent extension of neural networks to general dependency structures between data points (Fig. 1). Our goal is two-fold:

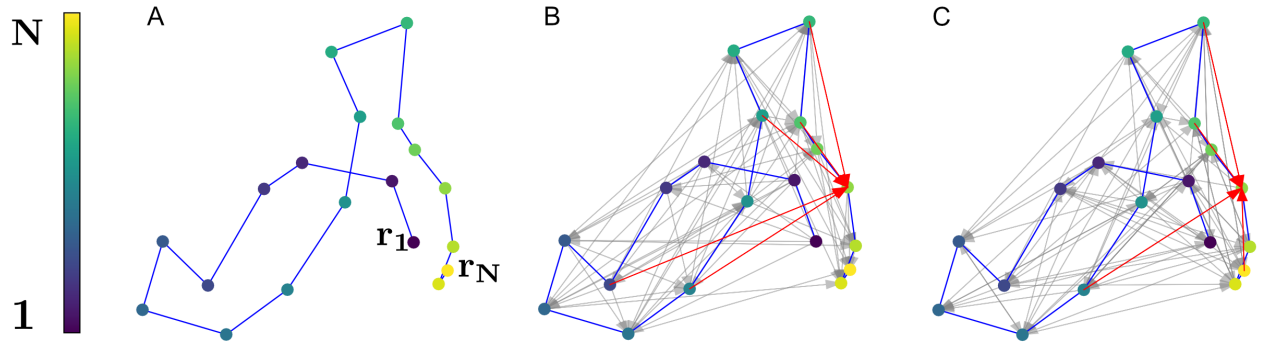

Figure 1: Examples of graphs associated to a single trajectory. A) An example of a short, subdiffusive fBM trajectory. B) Graph associated to the trajectory following a causal geometric edge wiring scheme (see Section 3.2). C) Graph obtained by wiring edges at random. Edges linking to a single selected node are shown in red in B and C.

(i) To develop an inference procedure that can be applied to trajectories of any length, is robust w.r.t. noise, and is numerically efficient, while keeping the number of parameters of the neural network low.

(ii) To provide physical interpretability of the neural network model through analysis of the latent representations it learns, and to study its robustness w.r.t. differences in the statistical properties of the training and testing data which is essential for transfer to experimental data analysis.

In this paper, we will focus our approach and applications on a limited subset of random walk models. However, the approach is transferable to any type of random walk. We focus here on anomalous motion stemming from the following 5 different models, which were selected for the anomalous diffusion (AnDi) challenge [51]: subdiffusive continuous time random walks (CTRW) [52, 53], fractional Brownian motion (fBM) [54], superdiffusive Levy walks (LW) [55–57], annealed transient time motion (ATTM) [58], and scaled Brownian motion (sBM) [59–61]. We will also consider Brownian motion and the Ornstein-Uhlenbeck process [62] in the analysis of the representations learned by the GNN.

Although the GNNs will always be trained on random walks whose nature does not change with time (i.e. each trajectory is generated by a single model with constant parameter values), we will explore the robustness of these trained networks to changes in the underlying generative model over time. We refer the reader to Refs. [63–67] for methods addressing spatial and temporal heterogeneities in the overdamped Langevin equation and to Refs. [38, 40] for fBM.

The paper is organised as follows. In Section 2 we briefly introduce the different models of random walks and the motivation for our learning procedure. In Section 3 we introduce the graph neural network approach to random walk inference, the graph building process and properties of the architecture. In Section 4 we discuss the results, quantify the properties of the learning procedure and explore what has been learnt by the network. We also explore neural network design choices and their effect on learning. Finally, in Section 5 we discuss the efficiency of the approach, challenges in its application to experimental data and possible extensions to unsupervised learning.

2 Anomalous random walks

Here we briefly introduce important properties of the anomalous random walk models [3, 62] that we will consider in the following. Anomalous diffusion emerges when a random walker diffuses in a disordered system or in an out-of-equilibrium environment. Usually, the principal property used to characterise an anomalous diffusion process is the temporal scaling behaviour of its mean squared displacement,

$$\langle \mathbf{r^2}(\mathbf{t}) \rangle \propto \mathbf{t}^{\alpha} \ , \tag{1}$$

where $\mathbf{r}$ is the position of the random walker, $\langle\cdot\rangle$ denotes ensemble averaging, and $\alpha \in [0, 2]$ is the anomalous exponent. The process is subdiffusive when $\alpha < 1$, superdiffusive when $\alpha > 1$ and Brownian (i.e. non anomalous) when $\alpha = 1$.

Anomalous diffusion is associated to at least one of the following properties: (i) random walkers or their environment exhibit spatially or temporally varying properties, (ii) displacements are not statistically independent at any sufficiently small time scale, and (iii) displacements at small time scales exhibit anomalies that prevent the central limit theorem applying [19].

We consider the five following models of anomalous diffusion:

1. Fractional Brownian motion (fBM) [54, 62]. fBM is a Gaussian process characterised by long temporal correlations in the noise driving the process. It is generated by a Langevin equation [62] of the form $\frac{d\mathbf{r}(\mathbf{t})}{dt} = \sqrt{K_\alpha}\boldsymbol{\eta}(t)$, where $\boldsymbol{\eta}$ is a zero-mean Gaussian noise process with covariance structure $\langle \boldsymbol{\eta}(t_1)\boldsymbol{\eta}(t_2)\rangle = \alpha(\alpha-1)|t_1-t_2|^{\alpha-2}$, and K_α is a generalized diffusion constant that sets the scale of the process. fBM is a self-similar Gaussian process with stationary increments [54, 62]. Its associated noise is anti-persistent and negatively correlated in the subdiffusion regime but persistent and positively correlated in the superdiffusion regime. As such, this random walk model displays anomalous property (ii) as defined above. It is stationary and ergodic [68], and its likelihood is analytically tractable.

2. Scaled Brownian motion (sBM) [59, 69]. sBM is generated by a Langevin equation [62] with a time-dependent diffusion coefficient [59, 69] of the form $K(t) = \alpha K_\alpha t^{\alpha-1}$ and driven by an uncorrelated (white) noise. It can generate both subdiffusive and superdiffusive processes. sBM displays anomalous property (i), it is weakly non-ergodic [61] and has the same marginal probability density for the time-evolution of the walker's position as the fBM [60], but a different autocorrelation structure.

3. Subdiffusive continuous time random walk (CTRW) [52]. In the CTRW the random walker's motion is generated by a renewal process consisting of discrete jumps with a given waiting time distribution between jumps, $\psi(\tau)$, and a distance distribution of jump lengths, $f(\boldsymbol{\Delta})$. Within the context of this study f will be a centered Gaussian distribution and $\psi(\tau) \propto \tau^{-\alpha-1}$, giving rise to subdiffusive random walks. The definition of the CTRW used here corresponds to a physical model associated to an annealed environment [70]. The subdiffusive CTRW model displays anomalous property (iii) and does not have a tractable likelihood. It shows weak ergodicity breaking [71], ageing [72, 73], and non thermal plateau convergence when confined [74], and it has discontinuous paths.

4. Superdiffusive Levy walk (LW) [55, 75, 76]. LWs belongs to the CTRW class of models, but instead of performing discontinuous jumps their motion is continuous and composed of a series of uncorrelated "flights" of constant speed generated from given flight time and distance distributions, $\psi(\tau)$ and $f(\boldsymbol{\Delta})$, respectively. Here, we consider a subset of the LW class corresponding to superdiffusive motion with a flight time distribution scaling as $\psi(\tau) \sim \tau^{-1-\sigma}$, and with a distance distribution that is conditional on τ and given by $f(\boldsymbol{\Delta}) \propto \delta(|\boldsymbol{\Delta}| - v\tau)$, where v is a constant speed parameter and δ is the Dirac delta function. The superdiffusive LW displays anomalous property (iii), it exhibits weak ergodicity breaking and its likelihood is intractable.

5. Annealed transit time motion (ATTM) [58, 77]. The ATTM model considers a random walker in an annealed heterogeneous environment where the diffusivity varies over space and time. In the ATTM, the random walker has a diffusivity that is piecewise constant over time with values drawn from $(D) \sim D^{\sigma-1}$ and with resting time at each diffusivity level drawn from a conditional distribution $p(\tau|D)$. We consider here $p(\tau|D) = \delta(\tau - D^{-\gamma})$ and the parameter range $\sigma < \gamma < \sigma + 1$ (defined as Regime I in ref [58]), which leads to subdiffusive motion with an anomalous exponent of $\alpha = \sigma/\gamma$. Subdiffusive ATTM displays both anomalous properties (i) and (iii), and it exhibits weak ergodicity breaking and ageing.

We refer to realisations of random walks as trajectories. A trajectory, $\mathbf{R} = (\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_N)$, is a d-dimensional time-series of positions $\mathbf{r}_n$ recorded at equidistant points in time $t_n \in \{\Delta t, 2\Delta t, \ldots, N\Delta t\}$.

Experimentally, the positions of the trajectory are corrupted by various types of experimental noise. Here we assume an independent additive Gaussian white noise, i.e. $\hat{\mathbf{r}}_n = \mathbf{r}_n + \boldsymbol{\xi}_n$, where $\boldsymbol{\xi}_n \sim \mathcal{N}(\mathbf{0}, \sigma\mathbb{I})$ are d-dimensional Gaussian variables that are independent of both the past and of the random walker's motion. We refer to [17, 78] for how correlated noise induced by experimental settings can be taken into account. From a statistical analysis perspective, we consider individual random walk trajectories as noisy time series with two latent variables: a continuous anomalous exponent α and a discrete random walk model class.

3 From random walks to graph learning

We propose to leverage the flexibility and the capacity for representation learning of graph neural networks (GNNs) [79] to infer properties of random walks, in particular their anomalous exponent and model class, following a methodology which we detail below.

We start by associating a graph with each trajectory (see Figure 1). Inference is then performed by a GNN, capable of processing graphs of variable size, and which outputs an estimate of the anomalous exponent and a probability of belonging to each random walk model class. The weights of this neural network are set during training (on numerically generated trajectories) similarly to conventional supervised learning schemes, and the network can later be used for inference on other trajectories, not seen during training.

3.1 Rationale for learning random walks with graph neural networks

The use of a graphical representation has long been a method of choice to model and perform inferences of complex systems [80]. For example, factor graphs associated to mean field, belief propagation [81] and cavity methods [82] have been used to model spin glass dynamics and perform complex optimisation problems. Hidden Markov models [83] have been developed to model changes in random walker dynamics, and mixture models [84, 85] have been applied to approximate the point spread function in fluorescence microscopy and to model and analyse complex networks [86].

Over the past 4 years, extensions of deep learning approaches to graph data in the form of graph neural networks have attracted significant attention [79] and have demonstrated great efficiency for representation learning on point clouds, graphs and manifolds [87, 88]. GNNs meet several criteria that make them well suited for analyzing trajectories, and which motivated the design of our learning procedure. (i) They can be applied directly, using a shared architecture, to trajectories of different lengths [88, 89]. (ii) The choice of graph structure allows taking into account different time scales while retaining a sparse architecture. (iii) Numerous known features associated to random walks, such as the convex hull [90–92], first passage times [93–95], or the distributions of different features' extreme values [96–98], are linked to geometric properties which can be learned efficiently using GNNs. (iv) Finally, while advances in machine learning are associated to impressive achievements [43, 99] they are often obtained with large scale models (several millions to billions of parameters) that are prone to over-fitting and are challenging to interpret [100]. Hence, a model with a limited number of parameters and a means to quantify the acquired information during training would be beneficial to understand the requirements for random walk inference using machine learning. All these criteria point towards using graph neural networks for individual random walks analysis.

3.2 Graph representation of trajectories

We associate to each trajectory $\mathbf{R} = (\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_N)$ a directed graph $G = (V, E, \mathbf{X})$, with $V = \{1, 2, \ldots, N\}$ the set of nodes corresponding to the positions in the trajectory, $E \subseteq \{(i,j) | (i,j) \in V^2\}$ the set of edges connecting pairs of nodes, and $\mathbf{X} = (\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}, \ldots, \mathbf{x}_N^{(0)})$ a sequence of local feature vectors $\mathbf{x}_i^{(0)}$ associated to each node i in V (fig. 2A). Each node features vector $\mathbf{x}_i^{(0)}$, of dimension n_x, may contain any feature of the trajectory or of the graph and may depend only on i or on arbitrary neighborhoods of i. This graph-based encoding of trajectories was inspired by applications of GNNs to point cloud data [88, 89], but where time is an additional feature associated to each point here.

We attach to each node i the time t_i and three differently normalised versions of the i-th position $\mathbf{r}_i$: (1) normalised using the standard deviation of step sizes, (2) normalised using the standard deviation of positions, and (3) normalised using the mean step size. We also include in each node features the values of the cumulative sums of step sizes and of squared step sizes up to their time-point, computed using each of the three normalised positions. In order to prevent Levy walks from having a disproportionate influence in the learning process, due to the extreme distance values that the walk can induce, we clipped extreme jump lengths before normalisation. We noted during initial training that these rare events induced significant bias in the batch normalisation layers [101].

Thus, two matrices initially represent the graph associated to the random walk: the (sparse) adjacency matrix A, of size (N, N), and the node feature matrix X, of size (N, n_x) where n_x is the number of features initially attached t. Note that we may also add features to edges in the graph [50], represented by an edge feature matrix U, of size $(|E|, n_e)$. Training is much faster when omitting edge features, due to the efficient implementation of sparse matrix operations. As a consequence, we do not consider GNNs leveraging

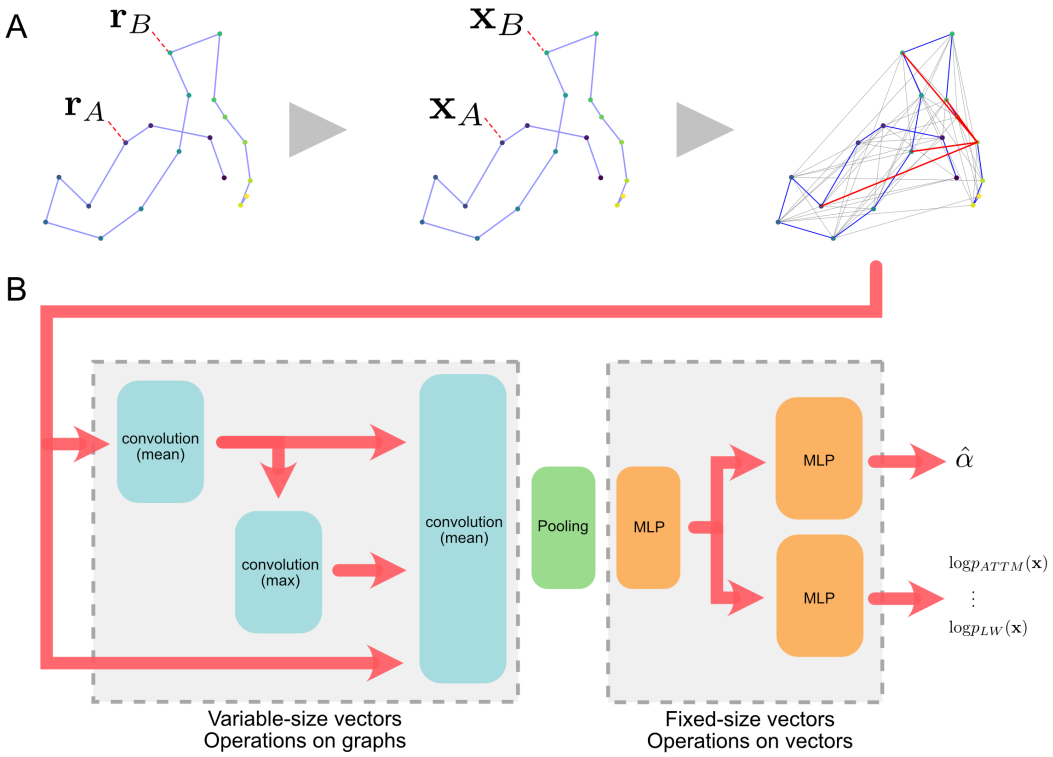

Figure 2: Graphical representation of the GNN model. A) Construction of a graph from a raw trajectory: (left) raw trajectory, (middle) positions in the trajectory are represented by nodes and a vector of local features is associated to each node; (right) nodes are connected to each other using a wiring scheme as described in section 3.2 to form a graph on which the GNN is applied. B) Overview of the GNN's architecture. Red arrows symbolise outputs of layers passed as inputs to other layers. When several arrows point to a same layer, their features vectors are concatenated before being passed to the layer. Graph convolution layers are shown in blue, they can process inputs of variable length (for graphs of variable size representing trajectories of variable length) using neural message passing (see Section 3.4). They use two different aggregation methods for messages: mean (averages over all the messages) and maximum (takes the maximal value of messages, for each feature). In green, the pooling operation embeds the variable-sized vector output from the graph convolution layers into a fixed-size vector representing the trajectory. Orange boxes downstream are multi-layer perceptrons, acting on vectors of fixed dimension as in conventional neural networks architectures.

edge features in the main text, but we investigate their performance for specialised training involving limited information (see Section 7.2 in Supplementary Information)

A known limitation of message passing GNNs has motivated us to choose particular wiring schemes for the graph. The mechanism of information propagation in a GNN involves iteratively passing messages between neighboring nodes, aggregating them in each step. The latter creates an information bottleneck [102], leading to a limitation of information encoding in finite sized vectors. A GNN may fail to faithfully propagate local information stemming from nodes separated by long paths in the graph. It can hence perform poorly if the properties to be predicted depend on long-range information, which is generally the case for the task of classifying various random walks and inferring their anomalous exponents. Our approach overcomes these limitations by using structured wiring schemes to ensure more direct message passing from distant nodes. We discuss here two different wiring schemes, (i) hierarchical causal and (ii) regular random, but many options are possible. In the hierarchical causal scheme (i), the incoming edges of each node connect only to nodes in the past (respecting causality): node i is connected to nodes $i-\Delta_1,\ldots,i-\Delta_{max}$, where $(\Delta_i)_{i\geq 1}$ is a geometric series (see details in 7.1). In the regular random scheme (ii), edges are drawn at random, with the only constraint that all nodes have the same *in*-degree, generating a type of random regular graph. Example graphs can be seen in Figure 1. In both schemes, the graph structure ensures that distant time points of the random walk are connected by short paths.

3.3 Neural network architecture

We used a two-part architecture for the graph neural network, starting with an encoder followed by task-specific multi-layer perceptrons [100], each estimating a property of interest from the latent representation built by the encoder – here the anomalous exponent and the random walk class. This architecture, shown in figure 2, enables multi-task training (i.e. simultaneous inference of a random walk's class and anomalous exponent).

The encoder is the entry-point to the model. It embeds the graph representation of the trajectory into a latent space whose dimension is independent of the trajectory length. To do so, it performs several graph convolution operations [103–106] (described in Section 3.4 below) which propagate learnt features through the graph. It is terminated by a pooling layer, i.e. an operator that combines an aggregation of features across nodes with a multi-layer perceptron that outputs a fixed-size vector. We used convolution layers using both "mean" and "max" operations to aggregate messages they receive from their neighbors, thus enabling the network to compute a broader variety of features. We also chose to wire convolution layers so that the last one receives both the output of its predecessor and the initial features to prevent the information from vanishing through bottlenecks created by successive graph convolutions.

3.4 Graph convolution layers and neural message passing

The core of GNN operations is formulated in terms of neural message passing, which gathers and transmits information from nodes to nodes through connecting edges (Fig. 3) and aggregates it using basic operations such as convolution and pooling [103–107].

Graph convolution layers implement operations on node feature vectors following a message passing scheme [107] (Fig. 3):

$$\vec{x}_i^{(k)} \leftarrow \gamma_k \left(\vec{x}_i^{(k-1)}, \mathcal{X}_{j \in \mathcal{N}(i)} \vec{x}_i^{(k-1)} \right) \quad . \tag{2}$$

Here, the exponent $(k-1)$ denotes a vector's value before the k-th convolution, and (k) its updated value after the convolution. The functions γ_k are neural networks (in our case, multi-layer perceptrons) whose weights are learned during training. The dimension of $\vec{x}_i^{(k)}$ is not constrained to be the same as that of $\vec{x}_j^{(k-1)}$: drawing analogy with convolution layers in classic architectures used for image processing, each output dimension of γ corresponds to a convolution kernel. χ is a permutation-invariant aggregation operator which reduces the set of input vectors to a single vector of fixed dimension. In our case, it is either a feature-wise mean or maximum across nodes. We illustrate both the graph building process and the learning in Figure 2.

We point out here that, while initial motivations for using GNNs stemmed from the known efficiency (see Section 3.1) of graph models in physics, GNNs differ strongly from physically motivated graph models and message passing techniques. The neural messages do not represent beliefs about features or variables of interest, they are not normalised by conservation of probability and optimisation is not performed by sampling. The graph serves as a means to link features to sets of neighbor features and allow "classical" learning to be performed by optimisation.

4 Results

We test our model's performance on classification (model selection) and regression (parameter estimation) tasks for simulated trajectories of varying lengths and with a range of localisation noise amplitudes and characteristic scales of motion. We use the same convention as in the AnDi Challenge [51] regarding the localisation noise, i.e. we apply an independent Gaussian noise to all positions in a trajectory with a standard deviation equal to a constant factor of the expected standard deviation of the jump sizes of the random walk. We refer to this proportionality factor as the "noise amplitude", and we consider noise amplitudes in the range $[0, 1]$.

The lengths and anomalous exponents of trajectories used for both training and evaluation were sampled uniformly between their respective extreme values (unless otherwise specified, $N = 10$–1000 and $\alpha = 0.05$–1.95). For each evaluation, performance metrics were computed using one million trajectories (200 000 from each model), generated using the AnDi package [51]. We added localisation errors with the same amplitude to all dimensions of a trajectory. Here, we will show only results for 3D trajectories, but the dimension of the trajectories does not qualitatively change results.

Unless otherwise specified, we trained the GNN to perform both classification and regression simultaneously by using a training objective given by a simple sum of the mean squared error (MSE) of the estimated anomalous exponent and the cross-entropy between the true and predicted class labels. To quantify regression accuracy,

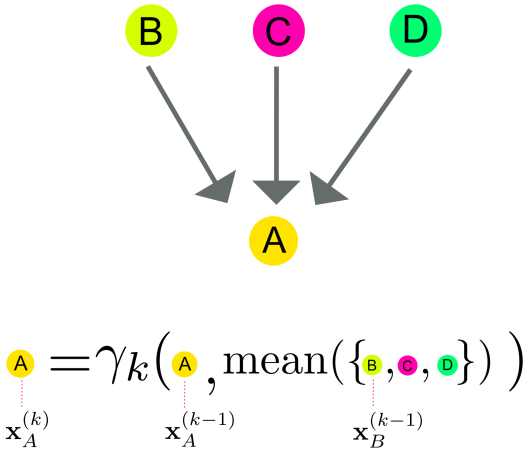

Figure 3: Illustration of a graph convolution. At each iteration k, each node (A) aggregates messages sent by nodes that are connected to it. The color code illustrates how the information is propagated between nodes. The weight parameters of the multi-layer perceptron γ_k, corresponding to the kth layer of the GNN, are learnt during the training. Here, the aggregation scheme for feature vectors shown is a mean over neighbor nodes.

we follow the AnDi settings [51] and use the mean absolute error (MAE) between the estimated and true values of the anomalous exponent α, while we use the F_1 score to quantify overall classification accuracy and confusion matrices for class-by-class evaluation. Section 7.1 gives detailed definitions of each of the measures.

4.1 Performance in absence of localisation noise

We show in Figures 4 and 5 the performance of a GNN trained on trajectories of lengths between 10 and 1000 and in the absence of localisation noise. Figure 4A shows that the accuracy of the inference of the anomalous exponent α from a single trajectory depends on trajectory length and the class of random walk considered: the anomalous exponent of ATTM is harder to infer than that of CTRWs or fBMs. Looking at how the estimation error on α depends on its true value, we see as can be expected a conservative bias shifting estimates away from extreme α values (i.e. 0, 1 and 2). The bias is pronounced for short trajectories and decreases with trajectory length (Fig. 4B,C). The bias stems in majority from the poor performance on ATTM trajectories.

Looking at the confusion matrices to assess classification accuracy (Fig. 5), we see that even for short trajectories, most walks are accurately classified, and misclassifications mainly confuse sBM and fBM. For trajectories longer than 200 points, the classification exhibits high performance. The GNNs classification performance for short trajectories illustrates that it relies not only on the asymptotic properties but also on finite-scale features of the random walks. Hence, even for short trajectories identification is possible. Furthermore, the confusion between sBM and fBM can be explained by the similarity of the processes for short trajectories, with a more pronounced effect for α close to one (in the range of approximately 0.7 to 1.4). sBM and fBM indeed share the same marginal probability density for the time-evolution of the walker's position [59] and they both approach Brownian motion as α approaches one.

4.2 Robustness to noise

Experimentally recorded trajectories are subject to various sources of noise. We here focus only on localisation noise, modelled as an uncorrelated Gaussian noise, but correlated noise sources may also be present [108, 109]. We investigated the performance of the inference when both training and inferring on trajectories observed with a broad range of noise amplitudes.

As shown in Figure 4, high localisation noise may significantly impair the accuracy of inferences when it has not been taken into account in the training data. We tested a global approach to induce noise robustness by assuming that no information on noise was accessible safe for a range of possible amplitudes. Hence, we trained on trajectories with added localisation error whose noise amplitude was randomly drawn from $[0, 1]$. Remarkably,

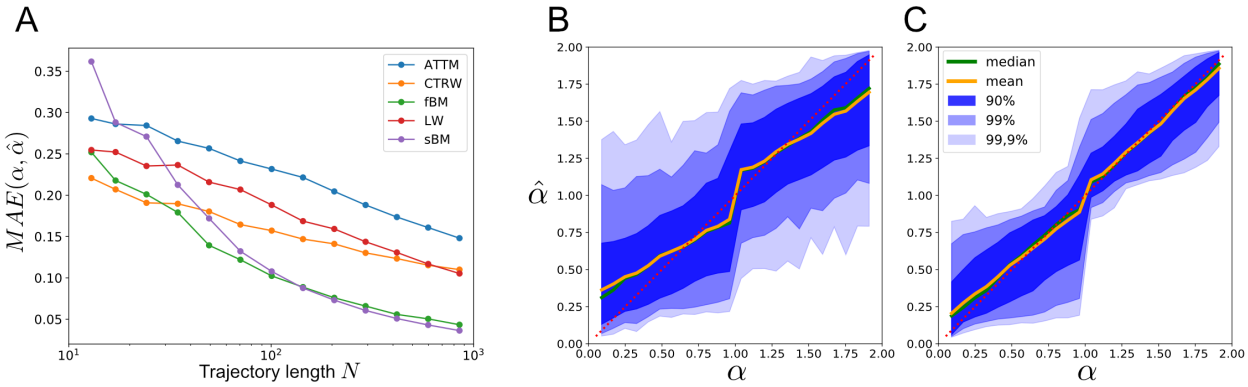

Figure 4: Performance of the inference of the anomalous exponent α on 3D trajectories. A) Mean absolute error (MAE) of estimate of the anomalous exponent, $\hat{\alpha}$, as a function of trajectory length N. B) & C) Distribution of $\hat{\alpha}$ values as a function of true value of α for B) trajectories of 10 to 100 points and C) 100 to 1000 points. x-axis : true exponent α, y-axis : inferred exponent $\hat{\alpha}$.

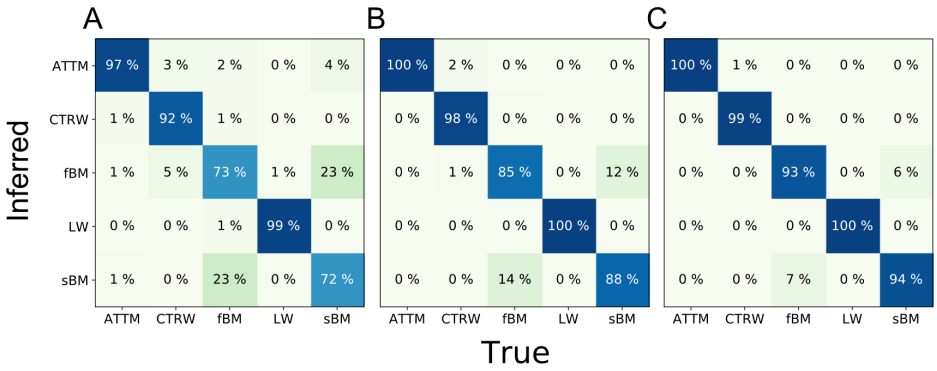

Figure 5: Confusion matrix for model classification, i.e. probability to identify a trajectory as having been generated by each random walk model (inferred class) given its true generating model (true class). For trajectory lengths in the range: A) 10-50, B) 50-200, and C) 200-1000.

training on such a wide distribution of noise leads to a nearly flat performance in of the anomalous exponent inference over the full range of noise amplitudes (green curve in Fig. 6). Similarly, the performance in classifying the trajectories, i.e. identifying their generative model, exhibited only a limited decrease with increasing noise amplitude. Conversely, when inference is performed on trajectories with localisation noise outside the range that the GNN was trained on, the performance of both regression and classification may degrade significantly.

In most experimental settings, there are means by which a range of possible values for positioning noise can be either deduced *a priori* or directly measured. Taking this in account by simulating with the same noise range to generate the training set increases the accuracy (curve corresponding to 0.2-0.4 in Fig. 6A), even if the GNN trained on the whole range of noise amplitudes already performs well. We refer the reader to [110] for an example of a procedure to estimate positioning noise within the context of single molecule experiments.

4.3 Improving performance on specific cases

We tested the performance of the architecture on data with parameters corresponding to typical experimental conditions, i.e. short trajectories (between 10 and 50 points) corrupted by a significant but bounded amount of noise (here, noise amplitudes between 0.2 and 0.4). The model was trained on this same range of trajectory lengths and noise amplitudes. We illustrate the model's performance in Figure 7. We show that despite the inherent difficulty of inferring properties from such short observations, the model is precise enough to extract relevant properties from these trajectories, e.g., it can reliably separate subdiffusive and superdiffusive trajectories

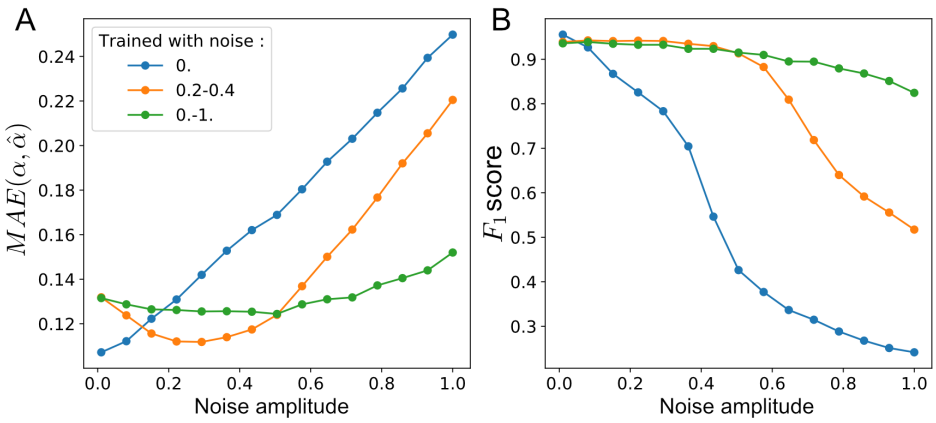

Figure 6: Inference performance as a function of noise amplitude and for different ranges of noise amplitudes included in the training data. A) Mean absolute error of the anomalous exponent inference, $\mathrm{MAE}(\hat{\alpha})$. B) F_1 score for model identification. The F_1 score is the harmonic mean of precision and recall (see details in Supplementary Section 7.1).

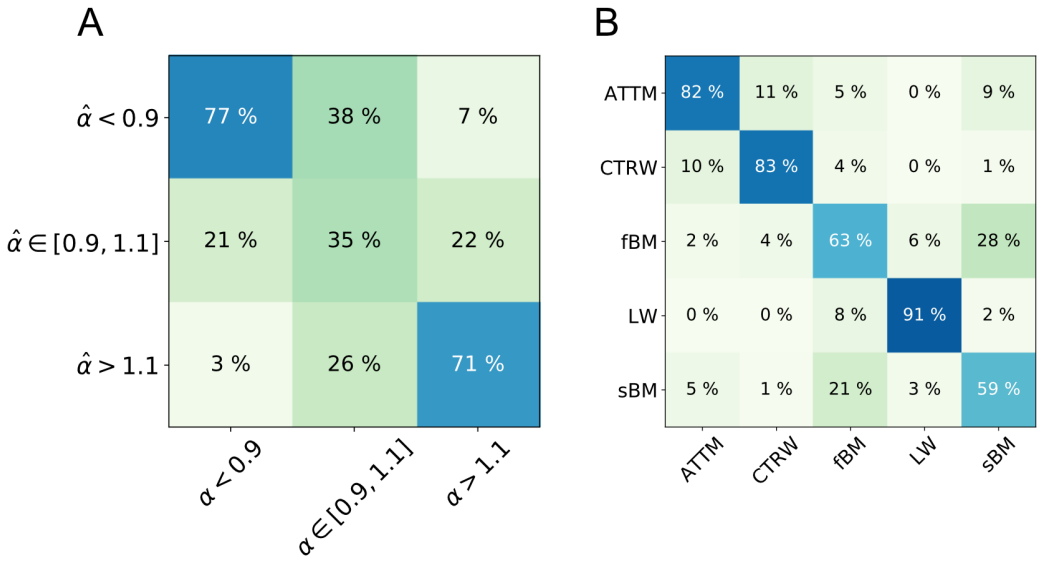

Figure 7: Performance on short trajectories ($10 \geq N \geq 50$) for a GNN trained with a known range of localisation noise amplitudes. A) Confusion matrix for identification of subdiffusive, normally diffusive and superdiffusive anomalous exponent values. B) Confusion matrix for model classification. The color code is identical to the one in figure 5.

and distinguish CTRWs from fBM. These results suggest that if we consider the random walk models to provide good approximations of the dynamics we may encounter in an experimental system, the approach may be efficiently leveraged to analyse single molecule experiments.

We next assessed more generally the effect of specialisation of the inference task on performance. First, we compared the performance of a model trained specifically on short trajectories ($10 \leq N \leq 100$) to that of a model trained on short and long trajectories ($10 \leq N \leq 1,000$) (Fig. 8A,B). We only considered model performance on the range of lengths that was common to both training sets. Then, we compared the performance of a model trained solely to infer the anomalous exponent with one trained both for regression and classification (Fig. 8C,D). We see that the performance increases with specialized training, although here only to a limited extent. This capacity of GNN models trained on numerical simulation to provide generalisable inference will be instrumental for their use in experimental data analysis.

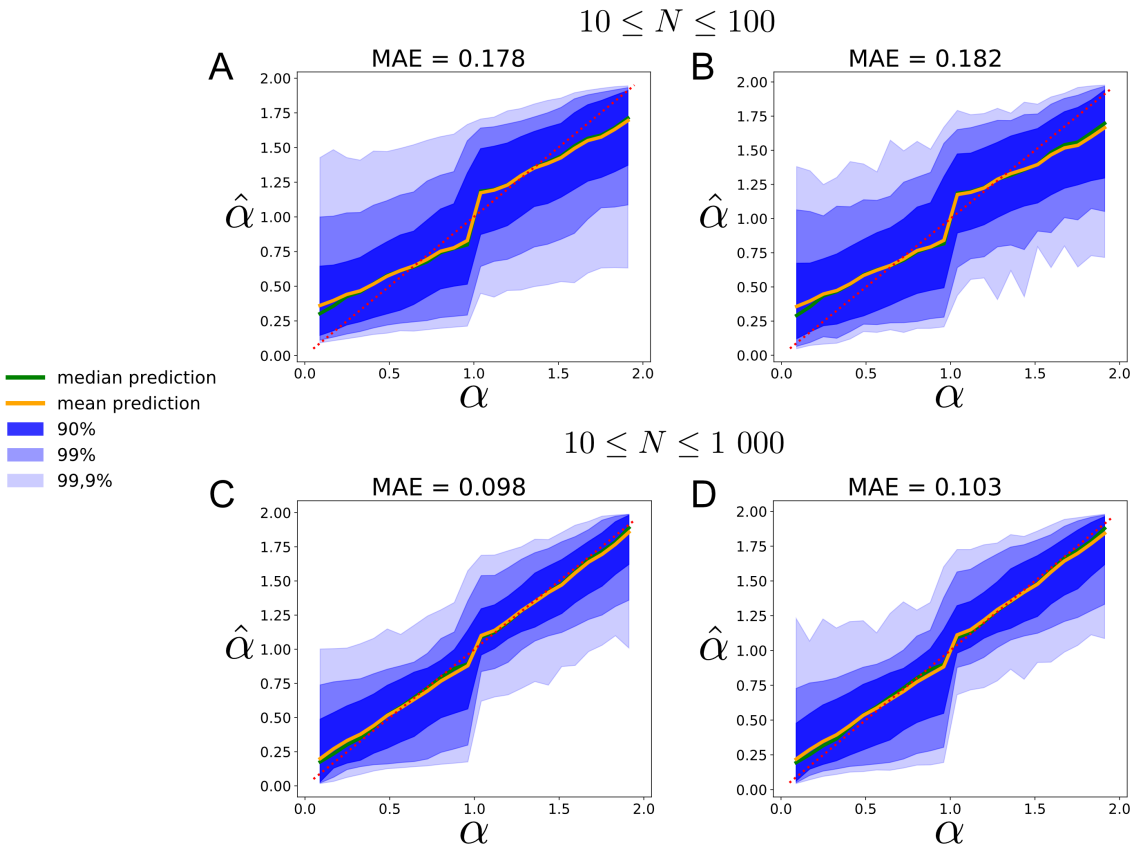

Figure 8: Improvement in model performance with specialized training: Distribution of estimated anomalous exponent values, $\hat{\alpha}$ as a function of the true value α for GNNs trained on only short or both short and long trajectories when applied to analyse short trajectories (A,B) and for GNNs trained only for regression or for both regression and classification (C,D). A) GNN trained only on short trajectories, $10 \leq N \leq 100$, and B) GNN trained on both short and long trajectories, $10 \leq N \leq 1000$, both applied to short trajectories of lengths $10 \leq N \leq 100$. C) GNN trained only on the anomalous exponent estimation task and D) GNN trained on both tasks.

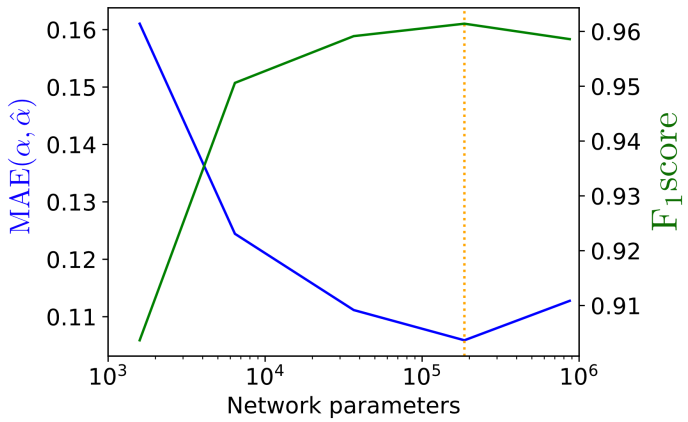

Figure 9: Inference performance (MAE and F_1 score) as a function of the number of neural network parameters. The vertical yellow line indicates the number of parameters used in the rest of the paper.

4.4 Performance as a function of the number of neural network parameters

GNNs generally do not require as many parameters as other recent deep neural architectures [87]. Here, we investigated the effect of the GNN's size (as measured in total number of parameters) on its performance. To do so, we modified the base architecture, first by making layers thinner and then by removing some layers (reducing by up to two orders of magnitude the number of parameters). Details are available in Table 7.1.

Results are shown in Figure 9. It is noteworthy that even with only about 1,600 parameters, the GNN maintains a good performance. This suggests that increased model tractability may be possible for GNNs.

5 Discussion

5.1 Latent space encoding of physical properties and generalisation

Our work lies in the framework of simulation-based inference [1]. This allowed large scale data generation at low computational cost and ensured that features of one dataset could not impair the learning. This contributed to reducing possible overfitting bias [111]. However, since some processes are non-ergodic, it is still a concern that the learning procedure might have failed to learn relevant properties on a finite dataset. Moreover, the statistics of experimentally recorded trajectories are unlikely to exactly match those of any of the models that we trained our machine learning model on. Here, we show that the learned latent space of the GNN encodes physically relevant properties of random walks, and we explore how it can be used to evaluate the robustness and generalisation performance of the approach.

In Figure 10 we provide a representation of the latent space (output of the penultimate MLP module). Each point corresponds to the latent representation of a single trajectory. We relied on UMAP [112], a dimensionality reduction algorithm in the family of manifold learning techniques, for 2D visualisation of the high dimensional latent space while preserving its local topology.

First, the structure reveals how well the model is able to distinguish trajectories of different types and anomalous exponents. Levy Walks, CTRWs and ATTM form three well separated clusters while fBMs and sBMs form a more complex shape with an extended overlapping region corresponding to $\alpha \approx 1$, i.e. the regime of normal (Brownian) diffusion (Fig. 10A). Within each cluster and continuous region, the anomalous exponent is mapped in a stable fashion (Fig. 10B). In concordance with the lower performance on regression for the ATTM model, we see that the value of α is less clearly mapped than for the other random walk models. The length of the trajectories, which directly relates to the available quantity of information, is also encoded in an ordered manner within the latent space, along a direction that is roughly orthogonal to the direction encoding the exponent.

In order to investigate the capacity of the GNN to encode physical properties of the random walks and its ability to generalize to trajectories with unseen properties, we applied the GNN to trajectories of walkers starting with a given type of motion and ending with another, and to random walks generated by unseen models on which the GNN was not trained.

To evaluate how the GNN encodes random walks that change of motion class over time, we generated trajectories of fixed length ($L = 200$) where the first part was generated by a given model of subdiffusive walks and the second by another, and varied the relative importance of the two parts. Both segments have $\alpha = 0.5$. In Figure 10D we can see the trajectories' encoded positions in the latent space draw a transition from the fBM domain to the ATTM cluster as the percentage of the ATTM part in the trajectory increases. Some trajectories fall within a region of the latent space that was originally not occupied. The model is thus able to continuously encode random walk properties and interpolate between the properties of the two models based on previously unseen behaviour. It is an indication of its ability to generalize.

To investigate how the GNN behaves when used on trajectories generated by models not included in the training phase, we use the GNN to encode the properties of trajectories generated by pure Brownian motion (BM, with $\alpha = 1$), and by the Ornstein-Uhlenbeck (OU) process which models Brownian motion confined in a harmonic potential [62]. In Figure 10E we show that BM trajectories all fall within the portion of the latent space where fBM and sBM overlap, corresponding to the region where $\alpha = 1$ and their dynamics approach Brownian motion. The OU trajectories cover a region where subdiffusive fBMs with α values close to, or slightly below one are encoded. This is also a physically sensible encoding as the OU process shows anticorrelated dynamics, similar to the fBM but with a much faster, exponentially decaying kernel. In this respect, the latent space does encode relevant physical properties, suggesting that the GNN will generalize well to experimental data with statistical properties that may not be exactly equal to any of the random walk models it was trained on.

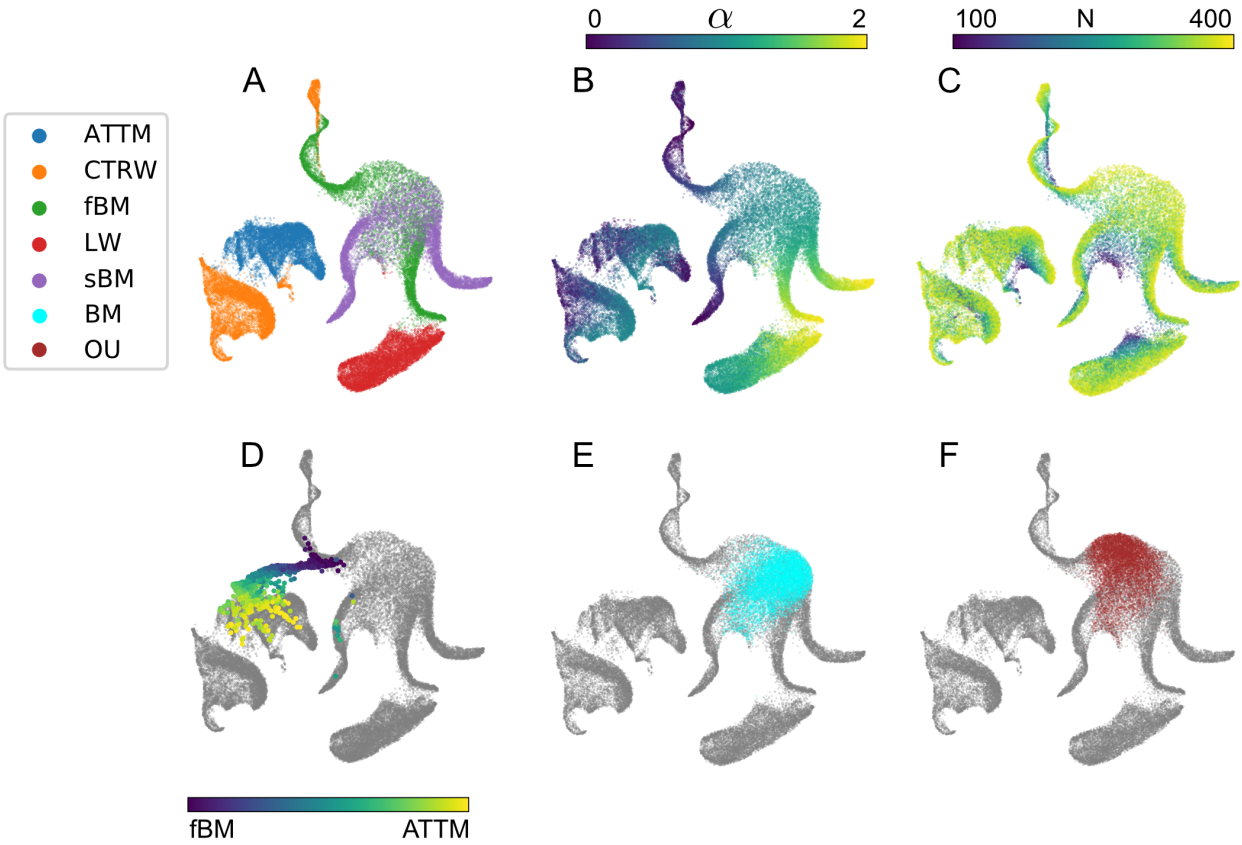

Figure 10: UMAP [112] representation of the latent space of the GNN. UMAP is a generalist manifold learning and dimension reduction algorithm relying on Riemannian geometry and algebraic topology to perform the embedding. Each point represents the encoded latent position of a single trajectory. A) Positions of each trajectory colored by the random walk's true model class. B) Mapping of the anomalous exponent α in the latent space. The color is associated with the value of α. C) Mapping of the trajectory length in the latent space. The color is associated with trajectory length. D) Positions of trajectories mixing fBM and ATTM in different proportions. Colors indicate the percentage of the trajectories generated by each of the two random walks. Note that trajectories continuously occupy the empty space between the ATTM cluster and the fBM domain. E) Projection of Brownian trajectories in the latent space. F) Projection of Ornstein-Uhlenbeck trajectories in the latent space.

5.2 Misclassified random walks

We investigated the sources of misclassification by the GNN. In Figure 11A we show the classification accuracy as a function of both trajectory length and noise factor. Error in model identification is concentrated in high noise and low length regions. Figure 11 helps to pinpoint the hardest samples to characterise. Intuitively, they are short and noisy. Furthermore, as could be expected from the latent space's structure, properties of trajectories whose anomalous exponent is close to one tend to be harder to infer (Fig. 11B).

5.3 Influence of graph structure

n iere, we investigated GNNs applied to graphs generated from recorded trajectories using a specific wiring scheme. As discussed in Section 3.2, we designed this scheme to ensure that the graph would connect distant portions of the trajectories, for efficient information exchange, while retaining a causal and hierarchical dependency structure. To test the influence of graph structure, we compare its performance to that of a random wiring scheme in Supplementary Section 7.3.

Recent work [113] has highlighted a strong link between the how a GNN's depth affects its performance and the connectivity of the graphs it is applied to. Following this direction, it has been proposed to learn the graph structure itself from an input point cloud in Euclidean space [114]. This could very well be applied to this setting in the future.

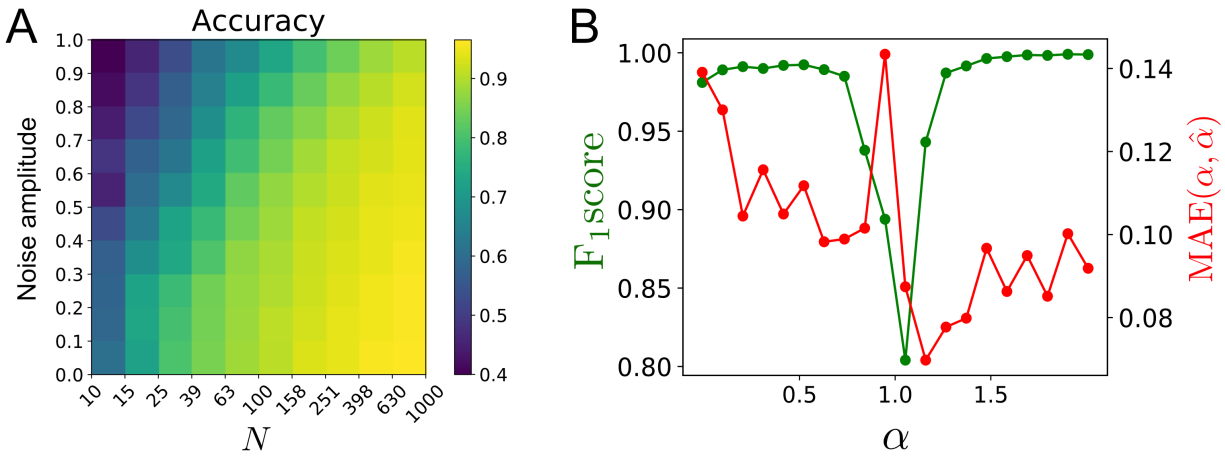

Figure 11: A) Accuracy of random walk classification as a function of trajectory length and noise amplitude. B) Classification accuracy and MAE of the estimated anomalous exponent $\hat{\alpha}$ as a function of the true value of the anomalous exponent. Trajectories are of length $10 \leq N \leq 1\ 000$, and without positioning noise.

5.4 Computational Complexity

The inference-time algorithmic complexity of the GNN model depends on the three main time-consuming parts when applying it to infer the properties of a random walk:

1. initial features evaluation;
2. forward pass through graph convolutions;
3. forward pass through subsequent layers.

1. The set of features we use to initialize each node's feature vectors can be computed in $O(N)$ time (N is the number of nodes, equal to the trajectory length).
2. From Eq. (2) it follows that the complexity of a convolution operation for all nodes of the graph is of $O(|E|)$ time complexity in the general case. In our case we restrict node's degrees to a maximum of k, so we have $E \leq kN$ and the convolution operation thus has $O(N)$ time complexity too.
3. Finally, as the dimension of the latent representation is independent of the graph's size, the forward pass through subsequent layers has $O(1)$ time complexity.

Thus, provided that the number of edges scales linearly with the number of nodes, this architecture can scale well to long trajectories, inferring their model class and anomalous exponent in $O(N)$ time.

6 Conclusion

In this paper we have shown that we could learn a physical representation of anomalous random walks using GNNs. We leveraged this representation to infer both the anomalous exponent and the model of a random walk from single trajectories. We relied on simulations to train our procedure. The scheme was found to be efficient in performing regression and classification tasks as well as being robust to positional noise. We showed that the latent space learned by the GNN is linked to the physical properties of the random walks.

While GNNs provide a general and expressive framework, they are still a new approach. Future developments are likely to improve their computational efficiency (e.g., improving their scalability to large graphs [115] and the efficiency of GPU acceleration [115]), their statistical power (e.g., by incorporating higher-order geometric features [107, 116] or appropriately relaxing the permutation-invariance of the aggregation operator), as well as our theoretical understanding of their capacities [50].

Representation learning paves the way to new approaches to explore biomolecule random walks whose dynamics cannot be purely described by a unique canonical random walk model. We foresee two directions for developments: First, neural networks and feature learning may be used to accelerate likelihood-free inference [1], allowing to fit more complex and realistic simulation-based models for experimentally recorded random walks. Second, the ability to learn relevant representations from random walk realisations may be exploited to develop unsupervised approaches to analyse experimentally recorded random walks.

Future work may also involve imposing constrains during learning to reinforce known symmetries in the random walk (e.g. directional symmetries) to increase training efficiency and to ensure that the neural network does not learn spurious features.

Acknowledgments. We thank the organizers of the AnDi challenge, which motivated this work. We thank Thomas Blanc, Mohamed El Beheiry, Srini Turaga, Hugues Berry, Raphael Voituriez & Bassam Hajj for helpful discussions. This study was funded by the Institut Pasteur, *L'Agence Nationale de la Recherche* (TRamWAy, ANR-17-CE23-0016), the INCEPTION project (PIA/ANR-16-CONV-0005, OG), and the "*Investissements d'avenir*" programme under the management of Agence Nationale de la Recherche, reference ANR-19-P3IA-0001 (PRAIRIE 3IA Institute).

The funding sources had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript. **Conflicts of interest.** The authors declare to have no financial or non-financial conflicts of interest.

7 Supplementary Material

7.1 Numerical simulations, Graph neural network training and hyper parameters

Simulating trajectories Random walks were generated using the python package provided during the AnDi challenge [51]. We slightly modified it to generate the same noise amplitude along all dimensions, which corresponds more closely to experimental conditions.

We additionally considered pure Brownian motion and the Ornstein-Uhlenbeck process:

- We simulated Brownian motion by directly sampling its displacements according to $\mathbf{X}_{n+1} = \mathbf{X}_n + \boldsymbol{\Delta x_n}$ with $\boldsymbol{\Delta x_n} \sim \mathcal{N}(\mathbf{0}, \mathbb{I})$.
- We simulated the Ornstein–Uhlenbeck process using the Euler method according to the following update formula: $\mathbf{X}_{n+1} = \mathbf{X}_n(1 - \delta t) + \sqrt{\delta t}\boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, 0.1\mathbb{I})$ and $\delta t = 0.01$.

Neural network hyper-parameters The detailed architectures of the model's components are summarized in table 7.1, ordered from the smallest to the largest tested network in terms of the number of parameters. Within this paper we have mostly discussed results associated to the architecture showing the best performance (see vertical line in 9), which correspond to the fourth row in the table. The first convolution layer receives the initial nodes features (there are 28) as well as, optionally, some of their powers. In the first two architectures presented in table 7.1, it receives only the first power. In the third and fourth, it additionally receives the squared features. In the last architecture it also receives the cubed features (hence the initial width being a multiple of 28). This is meant to allow the network to compute moments of the features distributions. When building the graphs, we used a maximal *in*−degree value of 20.

parameters	γ layers	projector	α module	classifier
1 588	(56,8) (16,8) (32,8)	(24,6)	(6,16,1)	(6,5)
6 420	(56,16,16) (32,16,16) (64,16,16)	(48,8)	(8,64,16,1)	(8,16,5)
36 660	(84,32,32,32) (64,32,32,32) (128,32,32,32)	(96,64,16)	(16,128,64,16,1)	(16,16,16,5)
185 879	**(84,128,64,64)** **(128,128,64,64)** **(256,128,64,64)**	**(192,128,64,32)**	**(32,128,128,64,16,1)**	**(32,64,32,5)**
871 596	(112,256,128,128,128) (256,256,128,128,128) (512,256,128,128,128)	(384,512,256,128,64)	(64,128,128,128,64,1)	(64, 128,64,32,5)

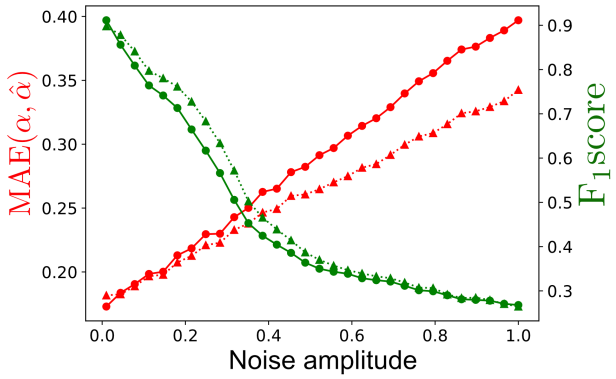

Figure 12: Performance of GNNs trained with (plain lines, circles) or without (dashed lines, triangles) features attached to edges as a function of the noise amplitude in the trajectories. MAE is shown in red, F_1 score in green.

We used batches of 128 trajectories for training, with a learning rate of 10^{-3}, exponentially decaying until it reaches $2 \cdot 10^{-4}$ after the network has seen $3 \cdot 10^6$ trajectories. Training lasts about 5 to 10 hours. We relied on the "PyTorch Geometric" package [117] to implement the graph convolutions and perform the learning.

Metrics The mean square error (MSE) is computed as follows: $\mathrm{MSE}(\hat{\alpha}) = \langle(\hat{\alpha} - \alpha)^2\rangle$.

The mean absolute error (MAE) reads: $\mathrm{MAE}(\hat{\alpha}) = \langle|\hat{\alpha} - \alpha|\rangle$

The cross-entropy loss, used for the classification task, reads: $\mathrm{CE} = -\langle\sum_{i=1}^{5} \delta_{m,i} \log \hat{p}_i\rangle$. Where $m \in \{1, 2, 3, 4, 5\}$ denotes the index of the true model of a trajectory.

To quantify overall classification performance, we used the F_1 score, which is the harmonic mean of the *precision* and the *recall* of the model. Using TP, FP & FN to denote the number of true positives, false positives and false negatives, respectively, we have:

- Precision: $P = \frac{\mathrm{TP}}{\mathrm{TP+FP}}$
- Recall: $R = \frac{\mathrm{TP}}{\mathrm{TP+FN}}$
- F_1 score: $F_1 = 2\frac{\mathrm{PR}}{\mathrm{P+R}}$

The confusion matrices, used to illustrate the ability of the GNN to infer random walk models, are defined as follows: $C_{i,j} = \langle\delta_{\hat{m}(\mathbf{R}),i}\rangle_{\mathbf{R} \text{ of type j}}$, where $\hat{m}$ is the index of the inferred model, i.e. the one which has been assigned the highest probability, and δ is the Kronecker symbol. That is, $C_{i,j}$ is the probability that a trajectory generated by the model class j (column) is classified as belonging to model class i (row). Defined this way, the diagonal elements $C_{i,i}$ are the per-class recall.

7.2 More complex Graph Neural Networks

In this paper, we focused on an approach where the GNN learns to build relevant features of random walks. We focused on a setting where features were only assigned to nodes, but the general framework of graph neural networks allows a more complex structure, by attaching features to edges, and by having two multi-layer perceptrons involved in the convolution operations : one before and one after the aggregation. In this more general setting, the neural message passing equations [107] read :

$$\vec{x}_i^{(k)} \leftarrow \gamma_k\left(\vec{x}_i^{(k-1)}, \mathcal{X}_{j\in\mathcal{N}(i)}\varphi_k\left(\vec{x}_i^{(k-1)}, \vec{x}_j^{(k-1)}, \vec{e}_{j,i}^{(k-1)}\right)\right) \tag{3}$$

We show in Figure 12 that edge features allow better performance on noise-free trajectories but is less robust to localisation noise.

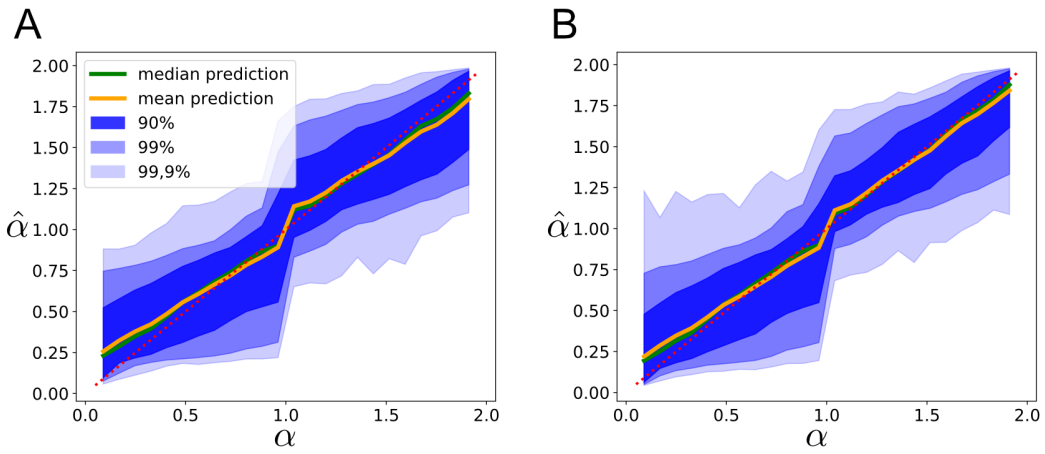

Figure 13: Regression performance of GNNs applied to different graph structures. A) Random regular graph structure. B) Causal hierarchical graph structure. Shaded regions represent probability intervals of the estimators.

7.3 Random versus structured connection patterns

As illustrated in Figure 13, connecting nodes according to a regular geometric pattern yields better performance than linking them at random.

In the geometric causal wiring scheme, node i receives edges from nodes $i-\lfloor\beta_1\rfloor, i-\lfloor\beta_2\rfloor, \ldots, \min(0, i-\lfloor\beta_k\rfloor)$, where k is the maximal *in*-degree, independent of trajectory length and which enforces sparsity, and where $\beta_1, \beta_2, \ldots, \beta_k$ is a geometric progression, parametrized such that the last node receives a connection from the first node, and $\lfloor\beta\rfloor$ denotes the integer part of β. We ensure that no edge is doubled. We used a value of $k = 20$ throughout this paper. Hence, nodes close to the start of the trajectory receive less connections than those located at the end.

References

1. Cranmer, K., Brehmer, J. & Louppe, G. The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences* **117,** 30055–30062. https://doi.org/10.1073/pnas.1912789117 (2020).
2. Muñoz-Gil, G. *et al. The anomalous diffusion challenge: single trajectory characterisation as a competition* in *Emerging Topics in Artificial Intelligence 2020* (eds Volpe, G., Pereira, J. B., Brunner, D. & Ozcan, A.) (SPIE, 2020). https://doi.org/10.1117/12.2567914.
3. Hughes, B. D. *Random walks and random environments* ISBN: 978-0198537885 (Clarendon Press Oxford University Press, Oxford New York, 1995).
4. Wachsmuth, M., Waldeck, W. & Langowski, J. Anomalous Diffusion of Fluorescent Probes inside Living Cell Investigated by Spatially-Resolved Fluorescence Correlation Spectroscopy. *J. Mol. Biol.* **298,** 677–689. ISSN: 00222836 (2000).
5. Sims, D. W. *et al.* Scaling laws of marine predator search behaviour. *Nature* **451,** 1098–1102. https://doi.org/10.1038/nature06518 (2008).
6. Shreve, S. *Stochastic calculus for finance* ISBN: 978-0387401010 (Springer, New York, 2004).
7. Betzig, E. *et al.* Imaging Intracellular Fluorescent Proteins at Nanometer Resolution. *Science* **313,** 1642–1645. https://doi.org/10.1126/science.1127344 (2006).
8. Sieben, C., Banterle, N., Douglass, K. M., Gönczy, P. & Manley, S. Multicolor single-particle reconstruction of protein complexes. *Nature Methods* **15,** 777–780. https://doi.org/10.1038/s41592-018-0140-x (2018).
9. Maiuri, P. *et al.* Actin Flows Mediate a Universal Coupling between Cell Speed and Cell Persistence. *Cell* **161,** 374–386. https://doi.org/10.1016/j.cell.2015.01.056 (2015).
10. Cavagna, A. *et al.* Scale-free correlations in starling flocks. *Proceedings of the National Academy of Sciences* **107,** 11865–11870. https://doi.org/10.1073/pnas.1005766107 (2010).
11. O'Toole, M. D., Lea, M.-A., Guinet, C., Schick, R. & Hindell, M. A. Foraging strategy switch of a top marine predator according to seasonal resource differences. *Frontiers in Marine Science* **2.** https://doi.org/10.3389/fmars.2015.00021 (2015).
12. Qian, H., Sheetz, M. & Elson, E. Single particle tracking. Analysis of diffusion and flow in two-dimensional systems. *Biophysical Journal* **60,** 910–921. https://doi.org/10.1016/s0006-3495(91)82125-7 (1991).
13. Weber, S. C., Thompson, M. A., Moerner, W. E., Spakowitz, A. J. & Theriot, J. A. Analytical tools to distinguish the effects of localization error, confinement, and medium elasticity on the velocity autocorrelation function. *Biophysical journal* **102,** 2443–2450 (2012).
14. Young, J.-G., Petri, G., Vaccarino, F. & Patania, A. Construction of and efficient sampling from the simplicial configuration model. *Physical Review E* **96.** https://doi.org/10.1103/physreve.96.032312 (2017).

15. Kepten, E., Weron, A., Sikora, G., Burnecki, K. & Garini, Y. Guidelines for the Fitting of Anomalous Diffusion Mean Square Displacement Graphs from Single Particle Tracking Experiments. *PLOS ONE* **10** (ed Levy, Y. K.) e0117722. https://doi.org/10.1371/journal.pone.0117722 (2015).
16. Lanoiselée, Y., Sikora, G., Grzesiek, A., Grebenkov, D. S. & Wyłomańska, A. Optimal parameters for anomalous-diffusion-exponent estimation from noisy data. *Physical Review E* **98.** https://doi.org/10.1103/physreve.98.062139 (2018).
17. Vestergaard, C. L., Blainey, P. C. & Flyvbjerg, H. Optimal Estimation of Diffusion Coefficients from Single-Particle Trajectories. *Phys. Rev. E* **89,** 022726. ISSN: 1539-3755 (2014).
18. Kepten, E., Bronshtein, I. & Garini, Y. Improved estimation of anomalous diffusion exponents in single-particle tracking experiments. *Physical Review E* **87,** 052713 (2013).
19. Metzler, R., Jeon, J.-H., Cherstvy, A. G. & Barkai, E. Anomalous diffusion models and their properties: non-stationarity, non-ergodicity, and ageing at the centenary of single particle tracking. *Phys. Chem. Chem. Phys.* **16,** 24128–24164. https://doi.org/10.1039/c4cp03465a (2014).
20. Burov, S. *et al.* Distribution of directional change as a signature of complex dynamics. *Proceedings of the National Academy of Sciences* **110,** 19689–19694. https://doi.org/10.1073/pnas.1319473110 (2013).
21. Schütz, G., Schindler, H. & Schmidt, T. Single-molecule microscopy on model membranes reveals anomalous diffusion. *Biophysical Journal* **73,** 1073–1080. https://doi.org/10.1016/s0006-3495(97)78139-6 (1997).
22. Tejedor, V. *et al.* Quantitative Analysis of Single Particle Trajectories: Mean Maximal Excursion Method. *Biophysical Journal* **98,** 1364–1372. https://doi.org/10.1016/j.bpj.2009.12.4282 (2010).
23. Condamin, S., Tejedor, V., Voituriez, R., Benichou, O. & Klafter, J. Probing microscopic origins of confined subdiffusion by first-passage observables. *Proceedings of the National Academy of Sciences* **105,** 5675–5680. https://doi.org/10.1073/pnas.0712158105 (2008).
24. Lanoiselée, Y. & Grebenkov, D. S. Revealing nonergodic dynamics in living cells from a single particle trajectory. *Physical Review E* **93.** https://doi.org/10.1103/physreve.93.052146 (2016).
25. O'Malley, D. & Cushman, J. H. A Renormalization Group Classification of Nonstationary and/or Infinite Second Moment Diffusive Processes. *Journal of Statistical Physics* **146,** 989–1000. https://doi.org/10.1007/s10955-012-0448-3 (2012).
26. Meroz, Y. & Sokolov, I. M. A toolbox for determining subdiffusive mechanisms. *Physics Reports* **573,** 1–29. https://doi.org/10.1016/j.physrep.2015.01.002 (2015).
27. Saxton, M. Lateral diffusion in an archipelago. Single-particle diffusion. *Biophysical Journal* **64,** 1766–1780. https://doi.org/10.1016/s0006-3495(93)81548-0 (1993).
28. Burnecki, K., Kepten, E., Garini, Y., Sikora, G. & Weron, A. Estimating the anomalous diffusion exponent for single particle tracking data with measurement errors - An alternative approach. *Scientific Reports* **5.** https://doi.org/10.1038/srep11306 (2015).
29. Bishop, C. M. *Pattern Recognition and Machine Learning (Information Science and Statistics)* ISBN: 0387310738. https://www.xarg.org/ref/a/0387310738/ (Springer, 2011).
30. Monnier, N. *et al.* Inferring transient particle transport dynamics in live cells. *Nature Methods* **12,** 838–840. https://doi.org/10.1038/nmeth.3483 (2015).
31. Das, R., Cairo, C. W. & Coombs, D. A Hidden Markov Model for Single Particle Tracks Quantifies Dynamic Interactions between LFA-1 and the Actin Cytoskeleton. *PLoS Computational Biology* **5** (ed Ranganathan, R.) e1000556. https://doi.org/10.1371/journal.pcbi.1000556 (2009).
32. Lindén, M. & Elf, J. Variational Algorithms for Analyzing Noisy Multistate Diffusion Trajectories. *Biophysical Journal* **115,** 276–282. https://doi.org/10.1016/j.bpj.2018.05.027 (2018).
33. Slator, P. J. & Burroughs, N. J. A Hidden Markov Model for Detecting Confinement in Single-Particle Tracking Trajectories. *Biophysical Journal* **115,** 1741–1754. https://doi.org/10.1016/j.bpj.2018.09.005 (2018).
34. Sgouralis, I. & Pressé, S. An Introduction to Infinite HMMs for Single-Molecule Data Analysis. *Biophysical Journal* **112,** 2021–2029. https://doi.org/10.1016/j.bpj.2017.04.027 (2017).
35. Grebenkov, D. S. Time-averaged mean square displacement for switching diffusion. *Physical Review E* **99,** 032133 (2019).
36. Hinsen, K. & Kneller, G. R. Communication: A multiscale Bayesian inference approach to analyzing subdiffusion in particle trajectories. *The Journal of Chemical Physics* **145,** 151101. https://doi.org/10.1063/1.4965881 (2016).
37. Krog, J., Jacobsen, L. H., Lund, F. W., Wüstner, D. & Lomholt, M. A. Bayesian model selection with fractional Brownian motion. *Journal of Statistical Mechanics: Theory and Experiment* **2018,** 093501. https://doi.org/10.1088/1742-5468/aadb0e (2018).
38. Loch-Olszewska, H. & Szwabiński, J. Impact of Feature Choice on Machine Learning Classification of Fractional Anomalous Diffusion. *Entropy* **22,** 1436. https://doi.org/10.3390/e22121436 (2020).
39. Koo, P. K. & Mochrie, S. G. J. Systems-level approach to uncovering diffusive states and their transitions from single-particle trajectories. *Physical Review E* **94.** https://doi.org/10.1103/physreve.94.052412 (2016).
40. Kowalek, P., Loch-Olszewska, H. & Szwabinski, J. Classification of diffusion modes in single-particle tracking data: Feature-based versus deep-learning approach. eprint: arXiv:1902.07942 (2019).
41. Wagner, T., Kroll, A., Haramagatti, C. R., Lipinski, H.-G. & Wiemann, M. Classification and Segmentation of Nanoparticle Diffusion Trajectories in Cellular Micro Environments. *PLOS ONE* **12** (ed Xu, B.) e0170165. https://doi.org/10.1371/journal.pone.0170165 (2017).
42. Dosset, P. *et al.* Automatic detection of diffusion modes within biological membranes using back-propagation neural network. *BMC Bioinformatics* **17.** https://doi.org/10.1186/s12859-016-1064-z (2016).
43. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521,** 436–444. https://doi.org/10.1038/nature14539 (2015).
44. Bo, S., Schmidt, F., Eichhorn, R. & Volpe, G. Measurement of Anomalous Diffusion Using Recurrent Neural Networks. eprint: arXiv:1905.02038 (2019).
45. Arts, M., Smal, I., Paul, M. W., Wyman, C. & Meijering, E. Particle Mobility Analysis Using Deep Learning and the Moment Scaling Spectrum. *Scientific Reports* **9.** https://doi.org/10.1038/s41598-019-53663-8 (2019).
46. Van Kuppevelt, D. *et al.* Mcfly: Automated deep learning on time series. *SoftwareX* **12,** 100548. https://doi.org/10.1016/j.softx.2020.100548 (2020).
47. Van den Oord, A. *et al. WaveNet: A Generative Model for Raw Audio* 2016. eprint: arXiv:1609.03499.
48. Bai, S., Kolter, J. Z. & Koltun, V. *An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling* 2018. eprint: arXiv:1803.01271.

49. Granik, N. *et al.* Single-Particle Diffusion Characterization by Deep Learning. *Biophysical Journal* **117,** 185–192. https://doi.org/10.1016/j.bpj.2019.06.015 (2019).
50. Xu, K., Hu, W., Leskovec, J. & Jegelka, S. *How Powerful are Graph Neural Networks?* 2018. eprint: arXiv:1810.00826.
51. Muñoz-Gil, G. *et al.* The anomalous diffusion challenge: single trajectory characterisation as a competition. *Emerging Topics in Artificial Intelligence 2020* (eds Volpe, G., Pereira, J. B., Brunner, D. & Ozcan, A.) http://dx.doi.org/10.1117/12.2567914 (2020).
52. Scher, H. & Montroll, E. W. Anomalous transit-time dispersion in amorphous solids. *Physical Review B* **12,** 2455–2477. https://doi.org/10.1103/physrevb.12.2455 (1975).
53. Magdziarz, M., Weron, A., Burnecki, K. & Klafter, J. Fractional Brownian Motion Versus the Continuous-Time Random Walk: A Simple Test for Subdiffusive Dynamics. *Physical Review Letters* **103.** https://doi.org/10.1103/physrevlett.103.180602 (2009).
54. Mandelbrot, B. B. & Ness, J. W. V. Fractional Brownian Motions, Fractional Noises and Applications. *SIAM Review* **10,** 422–437. https://doi.org/10.1137/1010093 (1968).
55. Klafter, J. & Zumofen, G. Lévy statistics in a Hamiltonian system. *Physical Review E* **49,** 4873–4877. https://doi.org/10.1103/physreve.49.4873 (1994).
56. Palyulin, V. V., Chechkin, A. V. & Metzler, R. Levy flights do not always optimize random blind search for sparse targets. *Proceedings of the National Academy of Sciences* **111,** 2931–2936. https://doi.org/10.1073/pnas.1320424111 (2014).
57. Koren, T., Klafter, J. & Magdziarz, M. First passage times of Lévy flights coexisting with subdiffusion. *Physical Review E* **76.** https://doi.org/10.1103/physreve.76.031129 (2007).
58. Massignan, P. *et al.* Nonergodic Subdiffusion from Brownian Motion in an Inhomogeneous Medium. *Physical Review Letters* **112.** https://doi.org/10.1103/physrevlett.112.150603 (2014).
59. Lim, S. C. & Muniandy, S. V. Self-similar Gaussian processes for modeling anomalous diffusion. *Physical Review E* **66.** https://doi.org/10.1103/physreve.66.021114 (2002).
60. Jeon, J.-H., Chechkin, A. V. & Metzler, R. Scaled Brownian motion: a paradoxical process with a time dependent diffusivity for the description of anomalous diffusion. *Phys. Chem. Chem. Phys.* **16,** 15811–15817. https://doi.org/10.1039/c4cp02019g (2014).
61. Sposini, V., Metzler, R. & Oshanin, G. Single-trajectory spectral analysis of scaled Brownian motion. *New Journal of Physics* **21,** 073043. https://doi.org/10.1088%2F1367-2630%2Fab2f52 (2019).
62. Crispin, G. *Handbook of Stochastic Methods: for Physics, Chemistry and natural sciences* 4th (Springer, 2009).
63. El Beheiry, M., Dahan, M. & Masson, J.-B. InferenceMAP: Mapping of Single-Molecule Dynamics with Bayesian Inference. *Nat. Methods* **12,** 594–595. ISSN: 1548-7091. pmid: 26125589 (2015).
64. Frishman, A. & Ronceray, P. Learning Force Fields from Stochastic Trajectories. *Physical Review X* **10.** https://doi.org/10.1103/physrevx.10.021009 (2020).
65. Serov, A. S. *et al.* Statistical Tests for Force Inference in Heterogeneous Environments. *Scientific Reports* **10.** https://doi.org/10.1038/s41598-020-60220-1 (2020).
66. Hozé, N. & Holcman, D. Statistical Methods for Large Ensembles of Super-Resolution Stochastic Single Particle Trajectories in Cell Biology. *Annu. Rev. Stat. Appl.* **4,** 189–223. ISSN: 2326-8298 (2017).
67. Laurent, F. *et al.* Mapping spatio-temporal dynamics of single biomolecules in living cells. *Physical Biology* **17,** 015003. https://doi.org/10.1088/1478-3975/ab5167 (2019).
68. Deng, W. & Barkai, E. Ergodic properties of fractional Brownian-Langevin motion. *Physical Review E* **79.** https://doi.org/10.1103/physreve.79.011112 (2009).
69. Saxton, M. J. Anomalous Subdiffusion in Fluorescence Photobleaching Recovery: A Monte Carlo Study. *Biophysical Journal* **81,** 2226–2240. https://doi.org/10.1016/s0006-3495(01)75870-5 (2001).
70. Bouchaud, J.-P. & Georges, A. Anomalous diffusion in disordered media: Statistical mechanisms, models and physical applications. *Physics Reports* **195,** 127–293. https://doi.org/10.1016/0370-1573(90)90099-n (1990).
71. Lomholt, M. A., Zaid, I. M. & Metzler, R. Subdiffusion and Weak Ergodicity Breaking in the Presence of a Reactive Boundary. *Physical Review Letters* **98.** https://doi.org/10.1103/physrevlett.98.200603 (2007).
72. Schulz, J. H., Barkai, E. & Metzler, R. Aging Renewal Theory and Application to Random Walks. *Physical Review X* **4.** https://doi.org/10.1103/physrevx.4.011028 (2014).
73. Krüsemann, H., Godec, A. & Metzler, R. First-passage statistics for aging diffusion in systems with annealed and quenched disorder. *Physical Review E* **89.** https://doi.org/10.1103/physreve.89.040101 (2014).
74. Burov, S., Metzler, R. & Barkai, E. Aging and nonergodicity beyond the Khinchin theorem. *Proceedings of the National Academy of Sciences* **107,** 13228–13233. https://doi.org/10.1073/pnas.1003693107 (2010).
75. Mandelbrot, B. *The fractal geometry of nature* ISBN: 978-0716711865 (W.H. Freeman, San Francisco, 1982).
76. Zaburdaev, V., Denisov, S. & Klafter, J. Lévy walks. *Reviews of Modern Physics* **87,** 483–530. https://doi.org/10.1103/revmodphys.87.483 (2015).
77. Akimoto, T. & Yamamoto, E. Distributional behavior of diffusion coefficients obtained by single trajectories in annealed transit time model. *Journal of Statistical Mechanics: Theory and Experiment* **2016,** 123201. https://doi.org/10.1088/1742-5468/2016/12/123201 (2016).
78. Calderon, C. P. Motion blur filtering: A statistical approach for extracting confinement forces and diffusivity from a single blurred trajectory. *Physical Review E* **93,** 053303 (2016).
79. Fey, M. & Lenssen, J. E. *Fast Graph Representation Learning with PyTorch Geometric* 2019. eprint: arXiv:1903.02428.
80. Koller, D. *Probabilistic Graphical Models: Principles and Techniques (Adaptive Computation and Machine Learning series)* (The MIT Press, 2009).
81. Yedidia, J. S. Message-Passing Algorithms for Inference and Optimization. *Journal of Statistical Physics* **145,** 860–890. https://doi.org/10.1007/s10955-011-0384-7 (2011).
82. Mezard, M. *Information, physics, and computation* ISBN: 978-0198570837 (Oxford University Press, Oxford England New York, 2009).
83. Bishop, C. *Pattern recognition and machine learning* ISBN: 978-0387310732 (Springer, New York, 2006).
84. Wainwright, M. J. & Jordan, M. I. Graphical Models, Exponential Families, and Variational Inference. *Foundations and Trends® in Machine Learning* **1,** 1–305. https://doi.org/10.1561/2200000001 (2007).

85. Samuylov, D. K., Purwar, P., Szekely, G. & Paul, G. Modeling Point Spread Function in Fluorescence Microscopy With a Sparse Gaussian Mixture: Tradeoff Between Accuracy and Efficiency. *IEEE Transactions on Image Processing* **28,** 3688–3702. https://doi.org/10.1109/tip.2019.2898843 (2019).
86. Peixoto, T. P. Nonparametric Bayesian inference of the microcanonical stochastic block model. *Physical Review E* **95.** https://doi.org/10.1103/physreve.95.012317 (2017).
87. Kipf, T. N. & Welling, M. *Semi-Supervised Classification with Graph Convolutional Networks* 2016. eprint: arXiv:1609.02907.
88. Qi, C. R., Yi, L., Su, H. & Guibas, L. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. *ArXiv* **abs/1706.02413** (2017).
89. Charles, R. Q., Su, H., Kaichun, M. & Guibas, L. J. *PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation* in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2017). https://doi.org/10.1109/cvpr.2017.16.
90. Luković, M., Geisel, T. & Eule, S. Area and perimeter covered by anomalous diffusion processes. *New Journal of Physics* **15,** 063034. https://doi.org/10.1088%2F1367-2630%2F15%2F6%2F063034 (2013).
91. Lanoiselée, Y. & Grebenkov, D. S. Unraveling intermittent features in single-particle trajectories by a local convex hull method. *Physical Review E* **96.** https://doi.org/10.1103/physreve.96.022144 (2017).
92. Grebenkov, D. S., Lanoiselée, Y. & Majumdar, S. N. Mean perimeter and mean area of the convex hull over planar random walks. *Journal of Statistical Mechanics: Theory and Experiment* **2017,** 103203. https://doi.org/10.1088/1742-5468/aa8c11 (2017).
93. Redner, S. *A Guide to First-Passage Processes* https://doi.org/10.1017/cbo9780511606014 (Cambridge University Press, 2001).
94. Guérin, T., Levernier, N., Bénichou, O. & Voituriez, R. Mean first-passage times of non-Markovian random walkers in confinement. *Nature* **534,** 356–359. https://doi.org/10.1038/nature18272 (2016).
95. Chupeau, M., Bénichou, O. & Majumdar, S. N. Survival probability of a Brownian motion in a planar wedge of arbitrary angle. *Physical Review E* **91.** https://doi.org/10.1103/physreve.91.032106 (2015).
96. Mori, F., Majumdar, S. N. & Schehr, G. Distribution of the time between maximum and minimum of random walks. *Physical Review E* **101.** https://doi.org/10.1103/physreve.101.052111 (2020).
97. Godrèche, C., Majumdar, S. N. & Schehr, G. Record statistics for random walk bridges. *Journal of Statistical Mechanics: Theory and Experiment* **2015,** P07026. https://doi.org/10.1088/1742-5468/2015/07/p07026 (2015).
98. Wang, W., Barkai, E. & Burov, S. Large Deviations for Continuous Time Random Walks. *Entropy* **22,** 697. https://doi.org/10.3390/e22060697 (2020).
99. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding* 2018. eprint: arXiv:1810.04805.
100. Goodfellow, I. *Deep learning* ISBN: 978-0262035613 (The MIT Press, Cambridge, Massachusetts, 2016).
101. Ioffe, S. & Szegedy, C. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift* 2015. eprint: arXiv:1502.03167.
102. Alon, U. & Yahav, E. *On the Bottleneck of Graph Neural Networks and its Practical Implications* 2020. eprint: arXiv:2006.05205.
103. Defferrard, M., Bresson, X. & Vandergheynst, P. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. eprint: arXiv:1606.09375 (2016).
104. Kipf, T. N. & Welling, M. *Variational Graph Auto-Encoders* 2016. eprint: arXiv:1611.07308.
105. Ying, R. *et al. Hierarchical Graph Representation Learning with Differentiable Pooling* 2018. eprint: arXiv:1806.08804.
106. Wang, Y. *et al. Dynamic Graph CNN for Learning on Point Clouds* 2018. eprint: arXiv:1801.07829.
107. Hamilton, W. L. Graph representation learning. *Synthesis Lectures on Artifical Intelligence and Machine Learning* **14,** 1–159 (2020).
108. Berglund, A. J. Statistics of Camera-Based Single-Particle Tracking. *Phys. Rev. E - Stat. Nonlinear Soft Matter Phys.* **82,** 1–8. ISSN: 15393755. pmid: 20866658 (2010).
109. Vestergaard, C. L., Blainey, P. C. & Flyvbjerg, H. Single-Particle Trajectories Reveal Two-State Diffusion-Kinetics of hOGG1 Proteins on DNA. *Nucleic Acids Res.* **46,** 2446–2458. ISSN: 1362-4962. pmid: 29361033 (2018).
110. Lindén, M., Ćurić, V., Amselem, E. & Elf, J. Pointwise error estimates in localization microscopy. *Nature Communications* **8.** https://doi.org/10.1038/ncomms15115 (2017).
111. Mostafa, Y. *Learning from data : a short course* ISBN: 978-1-60049-006-4 (AMLBook.com, United States, 2012).
112. McInnes, L., Healy, J. & Melville, J. *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction* 2018. eprint: arXiv:1802.03426.
113. Alon, U. & Yahav, E. *On the Bottleneck of Graph Neural Networks and its Practical Implications* 2020. arXiv: 2006.05205 [cs.LG].
114. Kazi, A., Cosmo, L., Navab, N. & Bronstein, M. *Differentiable Graph Module (DGM) for Graph Convolutional Networks* 2020. arXiv: 2002.04999 [cs.LG].
115. Gui, C. *et al. A Survey on Graph Processing Accelerators: Challenges and Opportunities* 2019. eprint: arXiv:1902.10130.
116. Gainza, P. *et al.* Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning. *Nature Methods* **17,** 184–192. https://doi.org/10.1038/s41592-019-0666-6 (2019).
117. Fey, M. & Lenssen, J. E. *Fast Graph Representation Learning with PyTorch Geometric* in *ICLR Workshop on Representation Learning on Graphs and Manifolds* (2019).