

PAPER • OPEN ACCESS

Characterization of anomalous diffusion classical statistics powered by deep learning (CONDOR)

To cite this article: Alessia Gentili and Giorgio Volpe 2021 *J. Phys. A: Math. Theor.* **54** 314003

View the [article online](#) for updates and enhancements.

You may also like

- [WaveNet-based deep neural networks for the characterization of anomalous diffusion \(WADNet\)](#)
Dezhong Li, Qiujiu Yao and Zihan Huang
- [Cusping, transport and variance of solutions to generalized Fokker–Planck equations](#)
Sean Carnaffan and Reiichiro Kawai
- [Decomposing the effect of anomalous diffusion enables direct calculation of the Hurst exponent and model classification for single random paths](#)
Philipp G Meyer, Erez Aghion and Holger Kantz

Characterization of anomalous diffusion classical statistics powered by deep learning (CONDOR)

Alessia Gentili and Giorgio Volpe* 

Department of Chemistry, University College London, 20 Gordon Street, London WC1H 0AJ, United Kingdom

E-mail: g.volpe@ucl.ac.uk

Received 8 April 2021, revised 11 June 2021

Accepted for publication 17 June 2021

Published 7 July 2021



CrossMark

Abstract

Diffusion processes are important in several physical, chemical, biological and human phenomena. Examples include molecular encounters in reactions, cellular signalling, the foraging of animals, the spread of diseases, as well as trends in financial markets and climate records. Deviations from Brownian diffusion, known as anomalous diffusion (AnDi), can often be observed in these processes, when the growth of the mean square displacement in time is not linear. An ever-increasing number of methods has thus appeared to characterize anomalous diffusion trajectories based on classical statistics or machine learning approaches. Yet, characterization of anomalous diffusion remains challenging to date as testified by the launch of the AnDi challenge in March 2020 to assess and compare new and pre-existing methods on three different aspects of the problem: the inference of the anomalous diffusion exponent, the classification of the diffusion model, and the segmentation of trajectories. Here, we introduce a novel method (CONDOR) which combines feature engineering based on classical statistics with supervised deep learning to efficiently identify the underlying anomalous diffusion model with high accuracy and infer its exponent with a small mean absolute error in single 1D, 2D and 3D trajectories corrupted by localization noise. Finally, we extend our method to the segmentation of trajectories where the diffusion model and/or its anomalous exponent vary in time.

*Author to whom any correspondence should be addressed.



Original content from this work may be used under the terms of the [Creative Commons Attribution 4.0 licence](https://creativecommons.org/licenses/by/4.0/). Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

Keywords: anomalous diffusion, single trajectory characterization, classical statistics analysis, supervised deep learning, deep feed-forward neural networks

(Some figures may appear in colour only in the online journal)

1. Introduction

Anomalous diffusion (AnDi) refers to diffusion phenomena characterized by a mean square displacement (MSD) that grows in time with an exponent α that is either smaller (subdiffusion) or greater (superdiffusion) than one (standard Brownian diffusion) [1]. Typically, this is represented by a nonlinear power-law scaling ($\text{MSD}(t) \sim t^\alpha$) [1]. The growing interest in anomalous diffusion shown by the physics community lies in the fact that it extends the concept of Brownian diffusion and helps describe a broad range of phenomena of both natural and human origin [2]. Examples include molecular encounters in reactions [3], cellular signalling [4, 5], the foraging of animals [6], search strategies [7], human travel patterns [8, 9], the spread of epidemics and pandemics [10, 11], as well as trends in financial markets [12, 13] and climate records [14].

Various methods have been introduced over time to characterize anomalous diffusion in real data. Traditional methods are based on the direct analysis of trajectory statistics [15] via the examination of features such as the MSD [16–19], the velocity autocorrelation function [20] and the power spectral density [21, 22] among others [15, 23–34]. More recently, the blossoming of machine learning approaches has expanded the toolbox of available methods for the characterization of anomalous diffusion with algorithms based on random forests [35, 36], gradient boosting techniques [35], recurrent neural networks [37] and convolutional neural networks [38, 39].

Despite the significant progress made, the characterization of anomalous diffusion data still present non-negligible challenges, especially to maintain the approach insightful, easy to implement and parameter-free [40]. First, the measurement of anomalous diffusion from experimental trajectories is often hampered by the lack of significant statistics (i.e. single and short trajectories), by limited localization precision, by irregular sampling as well as by changes of the diffusion properties in time. Then, these limitations become even more pronounced when dealing with non-ergodic processes [1], as the information of interest can be well hidden within individual trajectories. Finally, while machine learning approaches are powerful and parameter-free, more traditional statistics-based approaches typically offer deeper insight on the underlying diffusion process.

These shortcomings have resulted in the launch of the AnDi challenge by a team of international scientists in March 2020 (<https://andi-challenge.org>) [40, 41]. The goal of the challenge was to push the methodological frontiers in the characterization of three different aspects of anomalous diffusion for 1D, 2D and 3D noise-corrupted trajectories [41]: (1) the inference of the anomalous diffusion exponent α ; (2) the classification of the diffusion model; and (3) the segmentation of trajectories characterized by a change in anomalous diffusion exponent and/or model at an unspecified time instant.

Here, in response to the AnDi challenge, we describe a novel method (CONDOR: Classifier Of aNomalous DiffusiOn tRajectories) for the characterization of single anomalous diffusion trajectories based on the combination of classical statistics analysis tools and supervised deep learning. Specifically, our method applies a deep feed-forward neural network to cluster parameters extracted from the statistical features of individual trajectories. By combining advantages from both approaches (classical statistics analysis and machine learning), CONDOR is highly

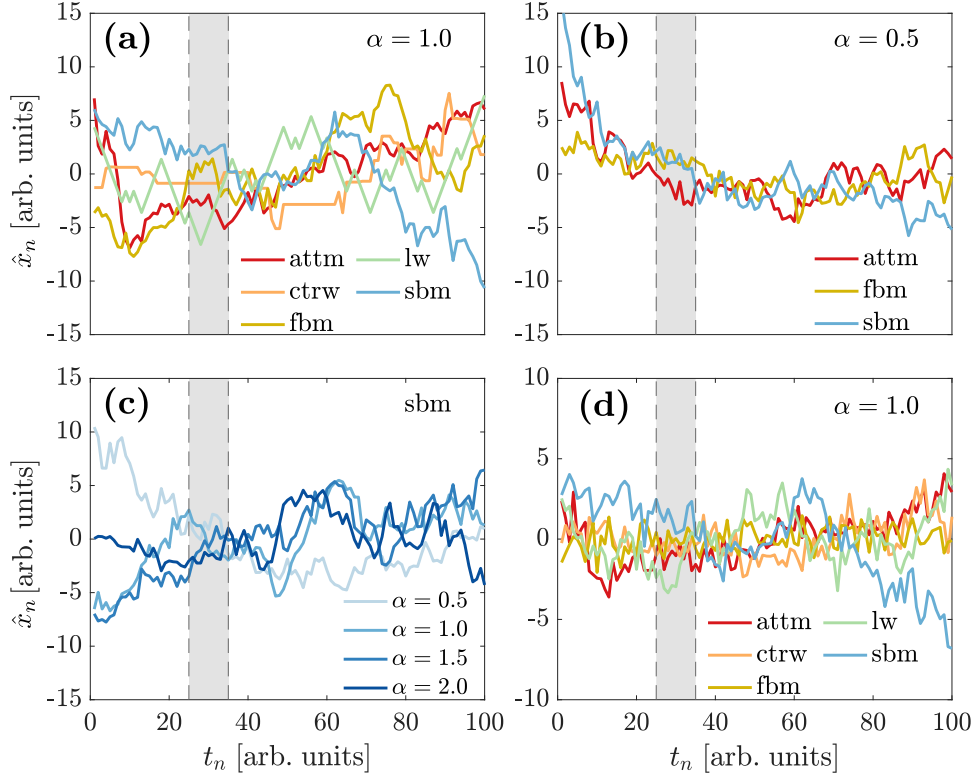


Figure 1. Representative anomalous diffusion trajectories. (a)–(c) Examples of single 1D anomalous diffusion trajectories in the absence of corrupting noise for (a) different models of standard diffusion (anomalous diffusion exponent $\alpha = 1$), for (b) different models of anomalous diffusion (anomalous diffusion exponent $\alpha = 0.5$), and for (c) the same anomalous diffusion model (sbm) with different values of α . The models considered here are the annealed transient time motion (attm), the continuous-time random walk (ctrw), the fractional Brownian motion (fbm), the Lévy walk (lw) and the scaled Brownian motion (sbm). Note that fbm and sbm processes are undistinguishable when $\alpha = 1$ as in (a). (d) Same trajectories as in (a) corrupted with Gaussian noise (of zero mean and variance $\sigma_{\text{noise}} = 1$) associated to a finite localization precision, corresponding to a signal-to-noise ratio (SNR) of 1. For better comparison, each sequence of positions x_n taken at times t_n is represented after subtracting its average and after rescaling its displacements to their standard deviation to give the normalized sequence \hat{x}_n [35]. The grey-shaded areas highlight a small portion (10 data points) of each trajectory.

performant and parameter-free. It allows us to identify the underlying anomalous diffusion model with high accuracy ($\gtrsim 84\%$) and infer its exponent α with a small mean absolute error ($\lesssim 0.16$) in single 1D, 2D and 3D trajectories corrupted by localization noise. CONDOR was always among the top performant methods in the AnDi challenge for both the inference of the α -exponent and the model classification of 1D, 2D and 3D anomalous diffusion trajectories. In particular, CONDOR was the leading method for the inference task in 2D and 3D and for the classification task in 3D. Finally, in this article, we extend CONDOR to the segmentation of trajectories.

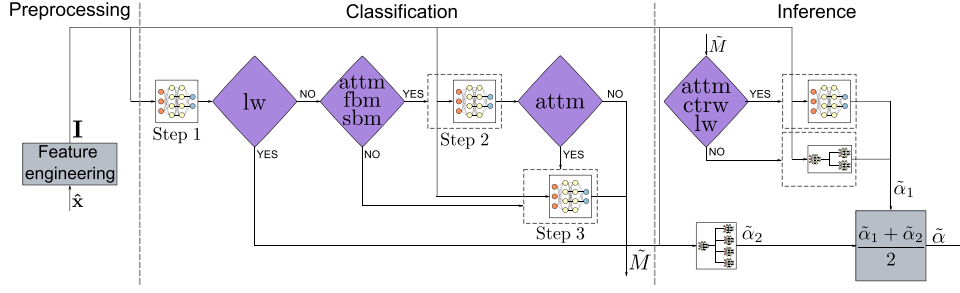


Figure 2. CONDOR workflow. CONDOR workflow is divided in three main parts: pre-processing (section 2.2), classification and inference (section 2.3). In the preprocessing step, the vector of positions \mathbf{x} associated to a single trajectory is used to extract a vector \mathbf{I} of $N = 1 + 92d$ inputs primarily based on classical statistics (with $d = 1, 2, 3$ being the trajectory dimension). The vector \mathbf{I} is the basic input for each neural network in the classification and inference algorithms. The output of the classification is determined by the output of up to three deep feed-forward neural networks and is the predicted model \hat{M} among five possible categories: the annealed transient time motion (attm), the continuous-time random walk (ctrw), the fractional Brownian motion (fbm), the Lévy walk (lw) and the scaled Brownian motion (sbm). Finally, the inference algorithm predicts the value of the exponent $\alpha \in [0.05, 2]$ basing its decision on \mathbf{I} and the predicted \hat{M} for each trajectory. In particular, the predicted $\hat{\alpha}$ is the arithmetic average of two distinct predictions: $\hat{\alpha}_1$ and $\hat{\alpha}_2$ where \hat{M} is used as selection element or additional input for the deep learning algorithm, respectively.

2. Methods

CONDOR combines classical statistics analysis of single anomalous diffusion trajectories with supervised deep learning. The trajectories used in this work are generated numerically (with the `andi-dataset` Python package [42]) from the 5 models considered in the AnDi challenge: the annealed transient time motion (attm) [43], the continuous-time random walk (ctrw) [44], the fractional Brownian motion (fbm) [45], the Lévy walk (lw) [46] and the scaled Brownian motion (sbm) [47]. Briefly, AnDi challenge datasets contain 1D, 2D or 3D trajectories from these 5 models sampled at regular time steps ($t_n = n\Delta t$, with n a positive integer and Δt a unitary time interval). The anomalous exponent α varies in the range $[0.05, 2]$ with steps of 0.05, as allowed by the respective model [41]. The shortest trajectories have at least 10 samples per dimension ($T_{\min} = 10\Delta t$). Prior to the addition of the localization error, the trajectories have a unitary standard deviation σ_s of the distribution of the displacements. Along each dimension i ($i = 1, \dots, d$ with $d = 1, 2, 3$), the trajectories are then corrupted with additive Gaussian noise (with zero mean and variance σ_i) associated to a finite localization precision to better simulate experimental endeavours [41, 42]. The variance σ_i of the additive Gaussian noise can assume values of 0.1, 0.5 or 1.0. The overall level of noise can then be computed as $\sigma_{\text{noise}} = \sqrt{\sum_{i=1}^d \sigma_i^2}$ [42]. The inverse of this value is then used to compute the signal-to-noise ratio (SNR). Figure 1 shows examples of such trajectories without noise (figures 1(a)–(c)) and with the addition of corrupting noise (figure 1(d)). By visual inspection, these examples show how differences between single trajectories are not always explicit, thus making the characterization of anomalous diffusion challenging. Already in the absence of noise, the same diffusion process (e.g. for $\alpha = 1$ in figure 1(a) and for $\alpha = 0.5$ in figure 1(b)) can be described by different models and the same model (e.g. sbm in figure 1(c)) can assume different values of the exponent α . These occurrences are not always easy to set apart (e.g. attm and fbm in figure 1(b)). The presence

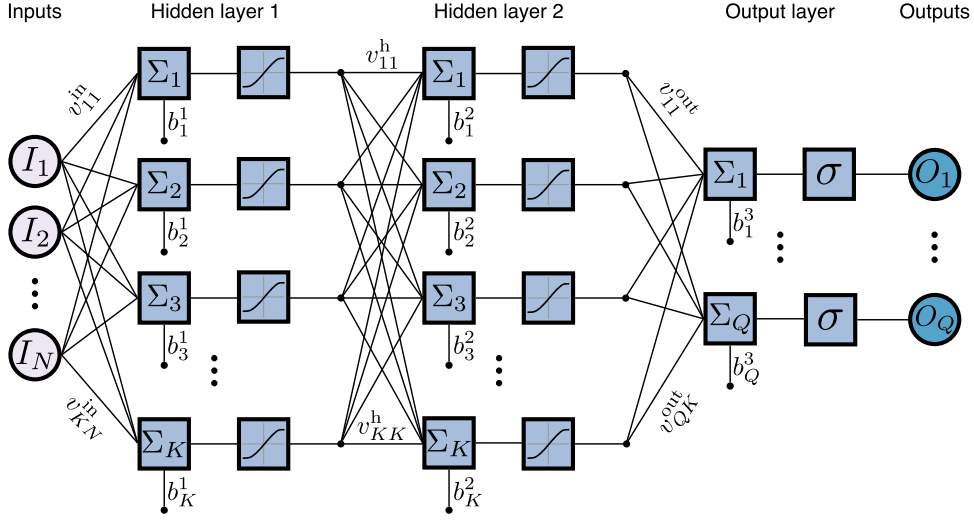


Figure 3. Basic neural network architecture in CONDOR. Architecture of the deep three-layer feed-forward neural network chosen for both classification and inference tasks. Inputs I_n (with $n = 1, 2, \dots, N$) are processed by two hidden layers and an output layer to yield outputs O_q (with $q = 1, 2, \dots, Q$). The hidden layers have $K = 20$ neurons each with a sigmoidal activation function (f). The output layer has Q neurons with a softmax activation function (σ). Each neuron has a summer (Σ) that gathers its weighted inputs and bias to return a scalar output. For each neuron, weights v and bias b are optimised during the training of the network.

Table 1. Set of $92d$ quantities (with $d = 1, 2, 3$) used to extract statistical information from each dimension of every trajectory as part of the N inputs for the deep learning algorithm. The approximate entropy (ApEn^{*i*}) is introduced as a way to quantify the amount of regularity and unpredictability of fluctuations over the considered time series. $m = 2, 3, \dots, \frac{T_{\min}}{\Delta t} - 1$ and $u = 1, 2$. Note that we do not define ApEn^{*i*} for $\Delta^i(m)$ when $m = \frac{T_{\min}}{\Delta t} - 2 = 8$ and $m = \frac{T_{\min}}{\Delta t} - 1 = 9$, as the numerical calculation of the approximate entropy requires at least three data points, which is not always the case for short trajectories.

	$\hat{\mathbf{w}}^i$	$\hat{\mathbf{W}}^i$	$\Delta^i(m)$	\mathbf{r}^i	\mathbf{f}^i	IPSD ^{<i>i</i>}	ΔMSD^i	WL ^{<i>i(u)</i>}	IACF ^{<i>i</i>}	fAsym ^{<i>i</i>}	tAsym ^{<i>i</i>}	AbCh ^{<i>i</i>}
μ^i	✓	✓	✓		✓	✓	✓	✓				
$\mu_{1/2}^i$	✓	✓	✓	✓	✓	✓	✓	✓				
σ^i		✓	✓		✓	✓	✓	✓				
β_1^i	✓	✓	✓		✓	✓	✓	✓				
β_2^i	✓	✓	✓		✓	✓	✓	✓				
ApEn ^{<i>i</i>}	✓	✓	(✓)		✓	✓	✓	✓				
Other									✓	✓	✓	✓
#/Dimension	5	6	46	1	6	6	6	12	1	1	1	1

of corrupting noise (figure 1(d)) and/or short trajectories (shaded areas in figures 1(a)–(c)) add extra complexity.

In this section, we provide a detailed description of our method (CONDOR) to characterise such single anomalous diffusion trajectories. The fundamental intuition behind our approach is that different trajectories produced by the same diffusion model should share similar statistical

features to set them apart from other models [1]. Nonetheless, in the presence of noise or with few data points, these similarities and differences do not need be explicit in single trajectories (figure 1). Thus, CONDOR first manipulates single trajectories to extract this hidden statistical information and then employs the power of deep learning to analyse and cluster it. CONDOR workflow is described in section 2.1 and shown in figure 2. We will first discuss the feature engineering step used to preprocess the trajectories and extract a set of N relevant features primarily based on classical statistics analysis (section 2.2). These features become the inputs for the deep-learning part of the characterization algorithm based on deep feed-forward neural networks (figure 3), which, in order, classifies the diffusion model and infers the anomalous diffusion exponent α (figure 2). These two steps are performed by multiple neural networks, whose architecture and connection is explained in section 2.3. Section 2.4 discusses the training of CONDOR neural networks. Section 2.5 discusses how CONDOR's performance varies with the number of inputs. In section 2.6, we explain how CONDOR outputs can be used to perform the segmentation of trajectories, where the model and/or its anomalous exponent change in time. Finally, in section 2.7, we provide a brief explanation of the code behind CONDOR [48].

2.1. CONDOR workflow

CONDOR's workflow comprises three main activities executed in order (figure 2):

- The preprocessing of the trajectories to extract the inputs for CONDOR's neural networks;
- The classification of the anomalous diffusion model M ;
- The inference of the anomalous diffusion exponent α .

Additionally, a forth activity (i.e. segmentation) can be executed in case one is also interested in identifying a change in time of the underlying model and/or of the α exponent in a trajectory.

The first step (preprocessing) manipulates the trajectories to extract a vector \mathbf{I} of inputs primarily based on classical statistics. The number of inputs is independent of the length of the trajectory but depends on its dimensionality. A detailed description of these inputs and of the feature engineering process used to extract them follows in section 2.2.

In the following activity (classification), the inputs calculated in the previous step are used by a series of three neural networks to identify the underlying anomalous diffusion model behind each trajectory (figure 2). A detailed description of the architecture of these networks follows in section 2.3. The output of the first network used for classification (step 1 in figure 2) is the main classification step, where most of the trajectories are already allocated to the correct model. The following two networks (steps 2 and 3 in figure 2) perform an additional check on the classification obtained from the previous steps (step 2 on step 1 and step 3 on step 1 and step 2 combined) to either confirm their classification or reassign it for a small percentage of the trajectories (usually $\leq 10\%$). Practically, the output of the first network is directly used to classify trajectories as lw and is key to inform the following two networks (steps 2 and 3 in figure 2) on which trajectories in the dataset to select for the additional check. Specifically, the second network (step 2) accepts the trajectories classified as atm , fbm or sbm by the first network (step 1) and re verifies their classification; the third network (step 3) instead takes the trajectories classified as $ctrw$ by the first network (step 1) and those classified as atm by the second (step 2) and re verifies their classification. Steps 2 and 3 are particularly useful to avoid misclassification of atm trajectories. Indeed, in the atm model, the diffusion coefficient can vary in space or time (in the AnDi challenge only time variations were considered [41, 43]). This feature can be easily confused with features of other processes: namely, the time-dependent diffusivity of the sbm [47], the time irregularities between consecutive steps in the $ctrw$ [44] and the correlations among the increments in the fbm [45].

The next activity (inference) is to determine the anomalous diffusion exponent α (figure 2). The estimated value $\tilde{\alpha}$ is obtained as the arithmetic average of the outputs of two different approaches based on deep learning (figure 2), as the two approaches lead to predicting an underestimated α value ($\tilde{\alpha}_1$) and an overestimated α value ($\tilde{\alpha}_2$), respectively. In the first approach, each trajectory is sent to a different set of networks (out of five possible sets, one per model) using the output \tilde{M} of the classification as a selection element. In the second approach (figure 2), each trajectory is sent to the same set of networks where the output \tilde{M} of the model classification is used as an additional input together with the list of inputs identified in the feature engineering step. A detailed description of the architecture of the networks for the inference task follows in section 2.3.

Finally, the outputs of both classification and inference can additionally be used to identify the presence of a change point in a trajectory. This activity is not based on the direct use of neural networks but rather on the analysis of CONDOR's classification and inference outputs on all segments of each trajectory identified with an overlapping moving window. A detailed description of the algorithm used for the segmentation task follows in section 2.6.

2.2. Feature engineering and CONDOR input definition

Before the classification and inference steps (figure 2), we analyzed each trajectory to extract $N = 1 + 92d$ inputs (with $d = 1, 2, 3$ being the trajectory dimensionality) primarily based on classical statistics (table 1). We would like to stress the fact that the total number of inputs is independent of the duration of each trajectory and only depends on its dimensionality. The inputs used in this work were ultimately selected by trial and error based on the final performance of the deep learning algorithm.

To extract the N inputs for every trajectory in 1D, 2D and 3D, we analysed the vector of the positions along each dimension $\mathbf{x}^i \equiv (x_1^i, x_2^i, \dots, x_{\frac{T_{\max}}{\Delta t}}^i)$ (with $i = 1, \dots, d$) independently, after first subtracting its average and rescaling its displacements $w_j^i = x_{j+1}^i - x_j^i$ (with $j = 1, \dots, \frac{T_{\max}}{\Delta t} - 1$) to their standard deviation to obtain the normalised displacements \hat{w}_j^i [35]. Practically, for each dimension, this step produces new time sequences $\hat{\mathbf{x}}^i \equiv (\hat{x}_1^i, \hat{x}_2^i, \dots, \hat{x}_{\frac{T_{\max}}{\Delta t}}^i)$ of positions reconstructed from the cumulative sum of the normalised displacements of the original trajectories. Therefore these new time sequences have zero mean ($\langle \hat{\mathbf{x}}^i \rangle = \mathbf{0}$) and their displacements have a unitary standard deviation. Importantly, this step allows for a better comparison between trajectories of different dimensions, durations and noise levels while keeping the performance of the deep learning algorithm optimal [35].

The first input is $\ln T_{\max}$ with T_{\max} being the duration of each trajectory. Independently of the dimensionality d of the trajectory, this input is only defined once, and is useful to enable the deep learning algorithm to properly weight different trajectories based on their duration.

The next subset of the N inputs is then extracted from sixteen mathematical quantities calculated from the new time sequences $\hat{\mathbf{x}}^i$ to obtain as much statistical information as possible from each individual trajectory. For each dimension i of a trajectory, the quantities in question are (table 1):

- The vector of the normalised displacements $\hat{\mathbf{w}}^i \equiv (\hat{w}_1^i, \hat{w}_2^i, \dots, \hat{w}_{\frac{T_{\max}}{\Delta t}-1}^i)$, where $\hat{w}_j^i = \hat{x}_{j+1}^i - \hat{x}_j^i$ with $j = 1, \dots, \frac{T_{\max}}{\Delta t} - 1$;
- The vector $\hat{\mathbf{W}}^i \equiv (|\hat{w}_1^i|, |\hat{w}_2^i|, \dots, |\hat{w}_{\frac{T_{\max}}{\Delta t}-1}^i|)$ of the absolute values of the normalized displacements $\hat{\mathbf{w}}^i$;

- The eight vectors $\Delta^i(m) \equiv (\Delta_1^i(m), \dots, \Delta_{\frac{T_{\max}}{\Delta t} - m}^i(m))$ of the absolute values $\Delta_j^i(m) = |\hat{x}_{j+m}^i - \hat{x}_j^i|$ of the displacements $\hat{x}_{j+m}^i - \hat{x}_j^i$ (with $j = 1, \dots, \frac{T_{\max}}{\Delta t} - m$) taken at time lags $\tau = m\Delta t$ with $m = 2, 3, \dots, \frac{T_{\max}}{\Delta t} - 1$;
- The vector $\mathbf{r}^i \equiv (r_1^i, r_2^i, \dots, r_{\frac{T_{\max}}{\Delta t} - 2}^i)$ of the ratios between consecutive normalised displacements $r_j^i = \frac{\hat{w}_{j+1}^i}{\hat{w}_j^i}$ with $j = 1, \dots, \frac{T_{\max}}{\Delta t} - 2$;
- The vector $\mathbf{f}^i \equiv (|\hat{X}_1^i|, |\hat{X}_2^i|, \dots, |\hat{X}_{\lfloor \frac{T_{\max}}{2\Delta t} \rfloor}^i|)$ of the absolute values of the discrete Fourier transform of $\hat{\mathbf{w}}^i$ centered at the zero-frequency component and normalised to the trajectory length, $\hat{\mathbf{X}}^i = \frac{\Delta t}{T_{\max}} \mathcal{F}\{\hat{\mathbf{w}}^i\}$;
- The vector $\mathbf{IPSD}^i \equiv (|\mathbf{IPSD}_1^i|, |\mathbf{IPSD}_2^i|, \dots, |\mathbf{IPSD}_{\lfloor \frac{T_{\max}}{2\Delta t} \rfloor}^i|)$ with $\mathbf{IPSD}_j^i = \|\hat{\mathbf{X}}^i(j\delta)\|^2 = \frac{1}{\delta^2} \|\mathcal{F}\{\hat{\mathbf{w}}^i(j\delta)\}\|^2$ ($j = 1, \dots, \lfloor \frac{T_{\max}}{2\Delta t} \rfloor$), where $\hat{\mathbf{w}}^i(j\delta) \equiv (\hat{w}_{(j-1)\delta+1}^i, \dots, \hat{w}_{j\delta}^i)$ are all segments of $\hat{\mathbf{w}}^i$ selected with a non-overlapping moving window of width $\delta\Delta t$ (with $\delta = 3$). Practically, the values \mathbf{IPSD}_j^i correspond to the integrals of the normalised power spectral densities of $\hat{\mathbf{w}}^i(j\delta)$. Here, the power spectral densities are estimated as a periodogram and normalised to the length $\delta\Delta t$ of the segments $\hat{\mathbf{w}}^i(j\delta)$;
- The vector $\Delta\mathbf{MSD}^i \equiv (\Delta\mathbf{MSD}_1^i, \Delta\mathbf{MSD}_2^i, \dots, \Delta\mathbf{MSD}_{\lfloor \frac{T_{\max}}{2\Delta t} \rfloor - 2}^i)$ of the absolute values of the variations $\Delta\mathbf{MSD}_j^i = \left| \frac{\mathbf{MSD}_{j+1}^i - \mathbf{MSD}_j^i}{j^2 \Delta t^2} \right|$ (with $j = 1, \dots, \lfloor \frac{T_{\max}}{2\Delta t} \rfloor - 2$) of the time-averaged mean squared displacement (MSD) normalised to the corresponding time lag $\tau = j\Delta t$ of \mathbf{MSD}_j^i ;
- Two vectors $\mathbf{WL}^i(u)$ (with $u = 1, 2$) of the absolute values of the numerical approximation to the first two translations $\tau = (u-1)\Delta t$ of the continuous wavelet transform of $\hat{\mathbf{w}}^i$ (calculated with the default settings of the MATLABTM function `cwt`). The continuous wavelet transform is a continuous mathematical function used for the analysis of localised variations in noisy time series; it is calculated by performing the convolution of a signal (here, $\hat{\mathbf{w}}^i$) with a orthonormal family of functions $\phi_u^s(t) = \phi(\frac{t-u}{s})$ obtained by continuously scaling (by s) and translating (by τ) the mother wavelet $\phi(t)$ (here, the analytic Morse wavelet) [49].

For each of these quantities (but \mathbf{r}^i) and along each trajectory's dimension i , we calculated the mean (μ^i), the median ($\mu_{1/2}^i$), the standard deviation (σ^i), the skewness (β_1^i), the kurtosis (β_2^i) (table 1). Note that the standard deviation for $\hat{\mathbf{w}}^i$ is always one by definition and hence is not used as an input. Additionally, for each quantity (but \mathbf{r}^i), we also calculated the approximate entropy (ApEn^i), which is a statistical technique used to quantify the amount of regularity and unpredictability of fluctuations in time signals (here calculated numerically with the MATLABTM function `approximateEntropy`) [50]: the closer to zero the value of ApEn^i , the more regular and predictable the time series is. For \mathbf{r}^i , we only calculated the median $\mu_{1/2}^i$ instead. For every dimension i of the trajectory, based on the aforementioned quantities, we obtain 88 inputs.

To further improve the performance of the deep learning algorithm, the following extra four inputs were also added for each dimension to extract additional information about regular and irregular time patterns in the trajectories (correlations, power variations and abrupt changes):

- The integral IACF^i of the autocorrelation function $\text{ACF}^i(\tau)$ of $\hat{\mathbf{w}}^i$ normalised to the trajectory length over a time window of width $\delta\Delta t$ (with $\delta = 3$) at the right of the autocorrelation peak at $\tau = 0$, i.e. calculated as $\frac{\Delta t}{T_{\max}} \sum_{k=0}^2 \text{ACF}(k\Delta t)$;

- The level of frequency asymmetry $f\text{Asym}^i$ in the normalized power spectral density of $\hat{\mathbf{w}}^i$ calculated as $\sum_k |\hat{X}_k^i|^2 \text{sgn}(k - \frac{F_N}{2})$ with F_N being the Nyquist frequency and $k = 1, \dots, \lceil \frac{T_{\max}}{2\delta\Delta t} \rceil$;
- The level of time asymmetry $t\text{Asym}^i$ in the time series of all IPSDⁱ calculated as $\sum_j \text{IPSD}_j^i \text{sgn}(j - \lceil \frac{T_{\max}}{2\delta\Delta t} \rceil)$ with $j = 1, \dots, \lceil \frac{T_{\max}}{\delta\Delta t} \rceil$;
- The absence of abrupt changes AbCh^i in the variance of the elements of $\hat{\mathbf{x}}^i$ defined as $\text{AbCh}^i = 1 - \frac{N_{\text{AC}}\Delta t}{T_{\max}}$ being N_{AC} the number of points where an abrupt change is identified using the MATLABTM function `ischange` with a high threshold value (threshold = 20) to reduce these instances to points where there is a significant change in signal variance [51].

2.3. CONDOR architecture

In the model classification task (figure 2), the N inputs calculated in the previous step are first fed to a deep three-layer feed-forward neural network (architecture in figure 3) with two sigmoid hidden layers of twenty neurons each and one softmax output layer of five neurons to predict the model \tilde{M} of anomalous diffusion among the five possible categories (atm, ctrw, fbm, lw and sbm). We settled for this specific architecture with two dense layers because it is deep enough to efficiently classify noisy vectors arbitrarily while still avoiding that its learning rate is slowed down by an excessively complex architecture. The output of the previous neural network (step 1 in figure 2) is then refined by two additional networks (steps 2 and 3 in figure 2) with similar architectures (but with three output neurons and two output neurons, respectively) to improve the classification among trajectories respectively classified as atm, fbm and sbm and as atm and ctrw, as these models can be more easily confused. Alternatively, step 1 can be replaced by a binary classifier to extract trajectories classified as lw followed by a second classifier with four outputs to classify the remaining trajectories among atm, ctrw, fbm and sbm. Steps 2 and 3 are then applied directly to the outputs of this second classifier. While this alternative approach offers a slightly improved performance after step 1 in lower dimensions ($\approx 1\%$), the overall performance in all dimensions is optimal after step 2 and 3 and is independent of step 1 implementation.

Once the model is identified, the next step is to infer the anomalous diffusion exponent α . The estimated value $\tilde{\alpha}$ is obtained as the arithmetic average of the outputs of two different approaches based on deep learning (figure 2). In the first approach ($\tilde{\alpha}_1$ in figure 2), the output \tilde{M} of the classification is used to cluster the trajectories into five categories based on the identified model. Each category is analysed independently by a different set of neural networks (architectures as in figure 3). The neural networks for atm and ctrw have five output categories predicting $\tilde{\alpha}_1$ in the range 0.05 to 1 [41, 42]. The neural network for lw has five output categories predicting $\tilde{\alpha}_1$ in the range 1 to 2 [41, 42]. Finally, the prediction for fbm and sbm is based on two levels of neural networks (1×2) with a single network in the first level having two equally spaced output categories predicting $\tilde{\alpha}_1$ in the range 0.05 to 2; each category is then refined by an additional neural network with five equally spaced output categories in the corresponding subrange identified by the network in the first level. In the second approach ($\tilde{\alpha}_2$ in figure 2), the output \tilde{M} of the model classification is used as an additional input together with the list of inputs identified in section 2.2. These new set of inputs feed two levels of neural networks (1×4), each with the same overall architecture as in figure 3. The only neural network in the first level has four equally spaced output categories predicting $\tilde{\alpha}_2$ in the range 0.05 to 2; each of this category is then refined by an additional neural network with five equally spaced output categories in the corresponding α subrange identified by the network in the first level. As both approaches ($\tilde{\alpha}_1$ and $\tilde{\alpha}_2$) rely on the results of classification, misclassification of trajectories can

Table 2. Summary of CONDOR architecture and training key information for both classification task and inference task.

Task	Classification	Inference
Architecture	Feedforward neural network Two hidden layers (20, 20) Output layer (2, 3 or 5)	Feedforward neural network Two hidden layers (20, 20) Output layer (2, 4 or 5)
# Networks	3	14
Output	Probability of the input Belonging to each model Scaled conjugate	Probability of the α exponent Belonging to a specific range Scaled conjugate
Training algorithm	Gradient backpropagation	Gradient backpropagation
Training dataset size	300k trajectories (see tables S1 and S2) 70% training	300k trajectories (see tables S3–S12) 70% training
Training dataset usage	15% validation 15% test	15% validation 15% test
Performance function	Cross-entropy	Cross-entropy
Validation fails	100	100

be detrimental for the inference task. If misclassification is of concern (e.g. in the presence of strong localization noise), a more robust approach for inference is based on using a variation of the second approach alone ($\tilde{\alpha}_2$) where the model classification is no longer used as additional input.

2.4. Training of CONDOR

After defining their architecture, we trained each network of CONDOR independently on a set of trajectories for which either the ground-truth values of the models (for classification) or of α (for inference) was known. The networks that we used during the AnDi challenge (and whose results are discussed in this manuscript) were typically trained on well-balanced datasets containing trajectories of different length and disrupted by different noise levels (tables S1–S12). Due to how the AnDi challenge was set up, the ground truth information about the models was not available for the datasets used for the inference task in the challenge. We thus used the results from our own classification to train the networks for the inference task, which indeed rely on knowing the model (or its prediction). As our classification method is highly performant, errors due to misclassification are small. Nonetheless, if the ground truth about the model is also available, the networks for inference can also be trained directly based on this information, thus minimizing potential reduction in performance due to misclassification (see section 3.3). Each training was performed using the scaled conjugate gradient backpropagation (scg) algorithm (for speed) on mixed datasets with up to three hundred thousand trajectories from different models and values of α , with different durations and levels of corrupting noise [42]. The key training information is concisely summarised in table 2.

For training the networks used for classification, we typically used well-balanced sets of N_t trajectories extracted from a dataset of 300k trajectories per dimension ($\approx 60k$ trajectories per model per dimension). While the training of the network for step 1 (figures 2 and 4) requires a balanced combination of trajectories of all models (tables S1 and S2), the networks for steps 2 and 3 are trained on subsets of the full training dataset, which only contains trajectories of the relevant models (tables S1 and S2): hence, for step 2, we only used trajectories belonging to attm, fbm or sbm (i.e. up to $\approx 180k$ trajectories per dimension); and for step 3, we only

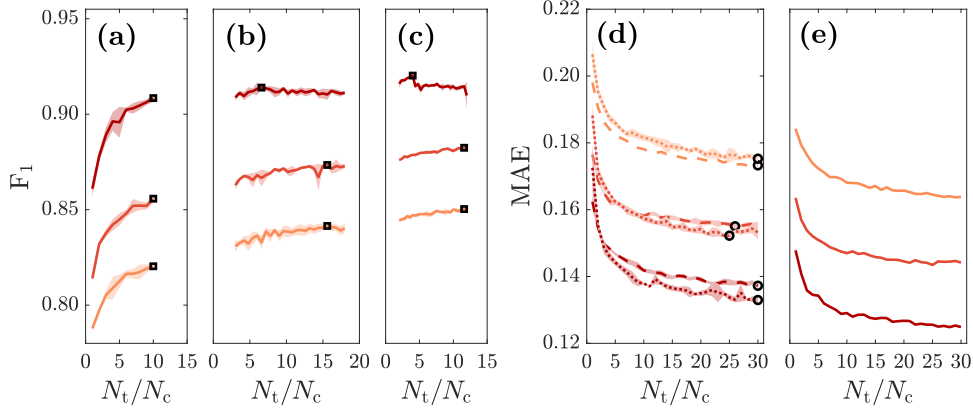


Figure 4. Evolution of CONDOR performance with the size of the training dataset. (a)–(c) F_1 score as a function of the overall number of trajectories (N_t) (i.e. including fractions for training, validation and testing) employed to train the three consecutive networks of the classification task (figure 2). The N_t trajectories are a subset of the total 1D, 2D or 3D training datasets for classification of 300k trajectories each and the F_1 score is calculated only on the test fraction (15% of trajectories). N_t is normalized to the size of a single AnDi challenge dataset ($N_c = 10k$ trajectories). The F_1 score is given at the output of (a) the first network, (b) the second network and (c) the third network used to refine the model predictions in the classification task (figure 2). At each of these steps, only the best performing network (black squares) is carried forward to the next step. (d) and (e) MAE as a function of overall number of trajectories N_t/N_c (i.e. including fractions for training, validation and testing) for (d) $\tilde{\alpha}_1$ (dashed lines) and $\tilde{\alpha}_2$ (dotted lines) and (e) for $\tilde{\alpha}$ (solid lines) as defined in figure 2. The N_t trajectories are a subset of the total 1D, 2D or 3D training datasets for inference of 300k trajectories each and the MAE is calculated only on the test fraction (15% of trajectories). For each N_t/N_c , the arithmetic average $\tilde{\alpha}$ is calculated using the best performing networks used to infer $\tilde{\alpha}_1$ and $\tilde{\alpha}_2$ with the same number of training trajectories. The networks with the lowest MAE for $\tilde{\alpha}_1$ and $\tilde{\alpha}_2$ are also shown in (d) (black circles). In all panels, each training point was obtained by incrementally increasing the previous set of N_t trajectories used for training with an additional well-balanced set of 10k trajectories from the full dataset. The shaded areas correspond to one standard deviation around the mean values (calculated from the performance of 5 distinct training attempts).

used trajectories belonging to attm and ctrw (i.e. up to $\approx 120k$ trajectories per dimension). Similarly, for training the networks used for inference, we typically used well-balanced sets of N_t trajectories extracted from a dataset of 300k trajectories per dimension ($\approx 75k$ trajectories per dimension for each of the following α ranges: $\alpha \in (0, 0.5]$, $\alpha \in (0.5, 1]$, $\alpha \in (1, 1.5]$ and $\alpha \in (1.5, 2]$). For the first approach ($\tilde{\alpha}_1$ in figures 2 and 4), the networks for each model are trained using a subset of the full dataset containing the trajectories that have been classified as the corresponding model by CONDOR. Each network is trained using only those trajectories classified in the corresponding model and with all possible α values returned by the specific network (i.e. allowed by the corresponding model, section 2.3) (tables S3–S10). In the case of attm, ctrw and lw, misclassified trajectories can have a ground truth value of α that falls outside the range of values allowed by the specific model. To account for this during the training of the networks associated to these three models, we assign these instances to the closest output α category (i.e. the category that minimises the error in estimating α). For the second approach ($\tilde{\alpha}_2$ in figures 2 and 4), each network is trained purely using trajectories selected based on their α value from the full training dataset, for which the ground truth value of α is known. In

particular, each network is trained using those trajectories with an α value falling among all possible output α categories returned by the specific network (tables S11 and S12).

The sets of N_t trajectories used for training each data point in figure 4 were randomly split among training (70%), validation (15%) and test (15%) trajectories. Overall, training with scg is very efficient on a standard desktop computer (processor Intel(R) Core(TM) i7-4790 CPU @ 3.60 GHz), being the size of the training dataset the main limiting factor. Training the three consecutive networks used for classification (figure 2) took at most 1 h and a half, while training the fourteen networks for the inference task took at most 3 h. Training time can be reduced by up to 2 orders of magnitude employing a GPU. After training, CONDOR is very computationally efficient on new data. For example, CONDOR can classify $\approx 80k$ trajectories/s on a standard desktop computer.

Figure 4 shows the performance of the classification task and of the inference task with the size of the training dataset. We evaluated the performance of the classification with the F_1 score:

$$F_1 = \frac{TP}{TP + \frac{1}{2}(FP + FN)},$$

where TP, FP and FN are the true positive, false positive and false negative rates, respectively. In particular, we computed the *micro* average of the F_1 score with the Scikit-learn Python's library [52]. F_1 can vary between 0 (no trajectory classified correctly) and 1 (all trajectories classified correctly). For balanced datasets, as those of the AnDi challenge (i.e. where each class is equally represented), the *micro* average of the F_1 -score coincides with the accuracy obtained as TP/N_{traj} , where N_{traj} is the number of trajectories in the dataset. To evaluate the performance of the inference of the α -exponent, we used the mean absolute error (MAE) instead:

$$\text{MAE} = \frac{1}{N_{\text{traj}}} \sum_{j=1}^{N_{\text{traj}}} |\tilde{\alpha}_j - \alpha_j|,$$

where $\tilde{\alpha}_j$ and α_j are the predicted and the ground truth values of the anomalous diffusion coefficient of the j th trajectory, respectively. MAE has a lower limit of 0, meaning that the coefficients of all the N_{traj} trajectories have been predicted correctly.

Figures 4(a)–(c) shows the F_1 score measured on the test fraction (15% of trajectories) of the training dataset at the output of the three consecutive networks used for classification (figure 2) for the 1D, 2D and 3D cases. For each step, we increased the number of trajectories used for training progressively until the F_1 score reached near saturation. Past this point any gain in F_1 score is relatively small and by far outbalanced by the increased computational cost associated to longer training times. The performance improves with the dimension d of the trajectories and, interestingly, at each step, saturation of the F_1 score for higher d appears to take place with smaller training datasets, as each trajectory contains intrinsically d times more information. Only the best performing network at each step is selected (black squares in figures 4(a)–(c)) and carried forward to the next step (the details on how these networks were trained are provided in tables 2, S1–S12). Independently of the dimensionality of the problem, the first network produces the highest increase of F_1 score with the size of the training dataset (figure 4(a)): for example, going from 10k to 100k trajectories, the F_1 score increases approximately by 4% in 1D, 5% in 2D and 6% in 3D. The combined contribution of the following two networks to the overall F_1 score is less pronounced, especially for higher dimensions (figures 4(b) and (c)); nonetheless they play a key role in refining the classification among attm, fbm and sbm (figure 4(b)) or among attm and ctrw (figure 4(c)), thus boosting the final F_1 score approximately by 4% in 1D, 3% in 2D and 1% in 3D.

Similarly, figures 4(d) and (e) shows the MAE obtained on the test fraction (15% of trajectories) of the training dataset for the inference of the α -exponent according to $\tilde{\alpha}_1$, $\tilde{\alpha}_2$ and their arithmetic average $\tilde{\alpha}$ (as defined in figure 2) for the 1D, 2D and 3D cases. As for F_1 , we report the MAE of each method as a function of the size of the training dataset progressively until it reaches near saturation. In general, the MAE lowers with the number of trajectories used for training and with their dimension d . As noted earlier, independently of the dimensionality of the problem, the arithmetic average $\tilde{\alpha}$ provides a better estimate of the α exponent with respect to both $\tilde{\alpha}_1$ and $\tilde{\alpha}_2$, leading to an overall MAE at saturation that is better than those obtained with the individual methods by at least 0.01 in all the dimensions.

As can be seen in tables S2, S7–S10, S12 and in figure 4, the optimal size of the specific training subset varies among the different networks of CONDOR. Increasing this size past the optimal point can lead to overlearning the training dataset and has a detrimental effect on CONDOR performance. For instance, this can be clearly appreciated in the training of the second classification step (figure 4(b), step 2).

2.5. CONDOR performance vs number of inputs

In general, the performance of each network of CONDOR increases with the number of inputs used: namely, when very few inputs are used (approximately up to $N/3$), adding an extra input causes a big relative change in performance; when several inputs are already in use (approximately above $2N/3$), the relative change in performance obtained by adding any extra input is relatively smaller as CONDOR reaches a near-saturation performance value. On average, we observed that this optimal performance is always obtained when the full set of inputs is used. This near-saturation value is reached faster in higher dimensions as CONDOR has more statistical information available to learn from (i.e. removing an input is more detrimental in lower dimensions). Interestingly, different subsets of inputs can ultimately provide a similar performance (although not optimal). This means that different inputs can convey similar statistical information that CONDOR learns to extract and distinguish. The presence of some repeated information is not detrimental to the optimal performance achieved by CONDOR and, indeed, helps CONDOR convergence to the same performance each time the training of its networks is repeated; when a smaller subset of inputs is used instead, different trainings can lead to a higher variability in the final performance after training and training of CONDOR should be repeated for best results.

2.6. Segmentation with CONDOR

The output of the classification and the inference can also be used to identify possible change points in a trajectory. In practice, a trajectory of length T_{\max} is divided in $T_{\max} - B + 1$ smaller segments using an overlapping moving window of width B (with $B \ll T_{\max}$). The width of the moving window is a trade-off between the possibility of identifying change points close to either end of a trajectory (which increases for smaller B) and the accuracy in predicting the correct time of the change point (which increase with B due to smaller errors in predicting the model and the α exponent associated to each segment). Each of these segments is then fed to CONDOR for classification and inference, thus generating two time series of $T_{\max} - B + 1$ values of the predicted model and α exponent. The occurrence of abrupt changes (up to two) in the root-mean-square (rms) level of these time series is used to identify the point of change of the model and/or of the α exponent [51]. To identify abrupt changes in the algorithm for the segmentation of trajectories, we used the MATLABTM function `findchangepts` to simultaneously detect significant changes in rms level of the two-dimensional vector containing the time series of predicted models and α exponents. We set the function `findchangepts` to

detect at most two significant changes: if no abrupt change is identified, we set the change point at the time instant corresponding to the center of the first moving window; if just one abrupt change is identified, we set the change point for the corresponding trajectory at the time instant where it was identified; if two abrupt changes are identified, we set the change point half-way between the two time instants where they were identified. This approach unavoidably introduces errors when the actual change point in the trajectory appears for $t \leq B/2$ and $t \geq T_{\max} - B/2$. To minimise the impact associated to this uncertainty in the attribution of the change point near the beginning and the end of the trajectory, each trajectory where no change point is identified with a given width B can be reanalysed with windows of increasingly smaller width at the expenses of a lower performance in the determination of the model and the α exponent (due to the use of shorter segments for classification and inference).

To evaluate CONDOR's performance in identifying the presence of a change point, we calculated the root mean square error of the change point localization as in the AnDi challenge [41]:

$$\text{RMSE} = \sqrt{\frac{1}{N_{\text{traj}}} \sum_{j=1}^{N_{\text{traj}}} (\tilde{t}_j - t_j)^2},$$

where \tilde{t}_j is the prediction and t_j the ground truth value of the change point, respectively, of a single trajectory in a dataset of N_{traj} trajectories. Similar to the MAE, the root mean square error (RMSE) has a lower limit of 0 when all the change points of the N_{traj} trajectories are correctly identified.

2.7. CONDOR codes

We have implemented all neural networks using the `patternnet` function in the MATLABTM-based Deep Learning ToolboxTM. We provide the MATLABTM implementations of the key functionalities of CONDOR [48]. Nonetheless, our approach is independent of the deep-learning framework used for its implementation. Moreover, it is also possible to exchange neural network models between the Deep Learning ToolboxTM and other platforms, such as those based on TensorFlowTM.

3. Results

3.1. CONDOR overall performance on the datasets of the AnDi challenge

The best performing networks identified during the training step (black symbols in figure 4) were carried forward and used in the prediction step, where we classified the anomalous diffusion model and inferred the α exponent of the same datasets used in the AnDi challenge (with 10k trajectories each) for testing purposes [41]. Figure 5(a) shows the overall F_1 score obtained on these test databases: $F_1 = 0.844$, $F_1 = 0.879$ and $F_1 = 0.913$ for the 1D, 2D and 3D cases, respectively. Figure 5(b) shows the overall MAE instead: $\text{MAE} = 0.161$, $\text{MAE} = 0.141$ and $\text{MAE} = 0.121$ for the 1D, 2D and 3D cases, respectively. These values enabled CONDOR to be one of the top performant methods in the AnDi challenge (<https://andi-challenge.org>) [40].

3.2. Analysis of CONDOR performance in the model classification task

Figure 6 provides further detail on CONDOR performance in the classification task by value of α (figures 6(a)–(c)), by model (figure 6(d)), by strength of the localization noise (figure 6(e))

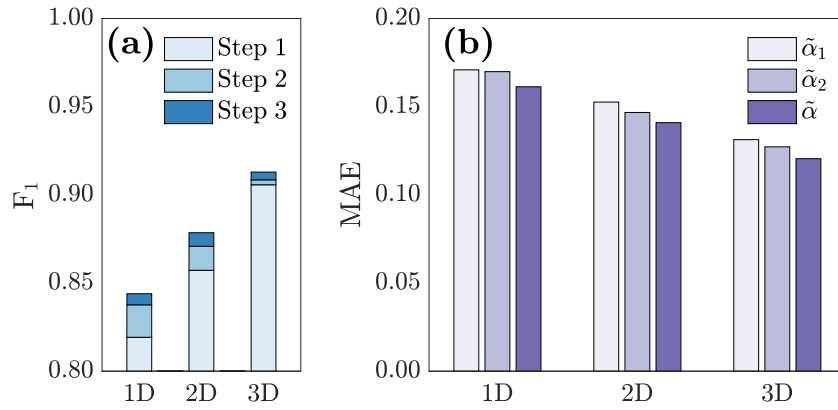


Figure 5. CONDOR performance on test datasets. (a) F_1 and (b) MAE obtained using CONDOR on the 1D, 2D and 3D datasets used in the classification and inference tasks of the AnDi challenge (10k trajectories each). In (a), the stacked bar charts shows the contribution of the three consecutive networks used for classification (figure 2) to the total F_1 score. In (b), the grouped bar charts shows the MAE associated to the inference of the α -exponent according to $\tilde{\alpha}_1$, $\tilde{\alpha}_2$ and their arithmetic average $\tilde{\alpha}$, which provides the best estimate of the α exponent (figure 2).

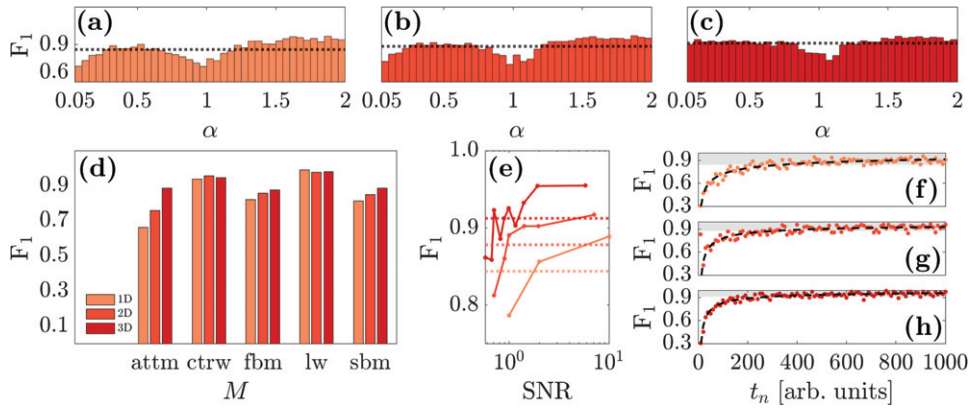


Figure 6. Detailed analysis of CONDOR performance for the classification task. CONDOR classification performance (F_1 score) on the 1D, 2D and 3D datasets of the AnDi challenge used for the classification task (as in figure 5(a)) (a)–(c) by value of α for the (a) 1D, (b) 2D and (c) 3D case, (d) by model, (e) by signal-to-noise ratio (SNR), and (f)–(h) by trajectory length for the (f) 1D, (g) 2D and (h) 3D case, respectively. In (a)–(c) each bar corresponds to an increment in α of 0.05. In (a)–(c) and (e) the dotted lines represent the overall F_1 as in figure 5(a). In (f)–(h), the trends are fitted to a power law ($y = c - ax^{-b}$, with $a, b, c > 0$) for ease of visualization. The shaded areas highlight the values of the F_1 score above the average value (figure 5(a)).

and by trajectory length (figures 6(f)–(h)). These data largely support CONDOR robustness of performance across the parameter space.

CONDOR is able to predict the correct model relatively consistently across the values of α , with the exception of $\alpha \simeq 0.05$ in 1D and 2D and of $\alpha \simeq 1$ for all dimensions

(figures 6(a)–(c)). This is not surprising. In fact, for $\alpha \simeq 0.05$ all trajectories are close to stationary and show a very low degree of motion, which could be dominated by the localization noise. Therefore, these trajectories are harder to classify, particularly in lower dimensions where a smaller amount of information is intrinsically available. Similarly, for $\alpha \simeq 1$ all models converge to Brownian motion, making it much more challenging to distinguish among them. Interestingly, in 1D and 2D, CONDOR shows an above-than-average performance in classifying superdiffusive trajectories, thus compensating what lost in the classification of subdiffusive trajectories.

Consistently across the dimensions (1D, 2D and 3D), CONDOR recognises two models (ctrw and lw) extremely well with $F_1 \geq 0.94$ for ctrw and $F_1 \geq 0.98$ for lw (figure 6(d)). This is understandable as both models show typical features that stand them apart from the remaining models and are relatively robust to the presence of localization noise (figure 1): i.e. long ballistic steps in lw trajectories and long waiting times in ctrw trajectories [1]. For the remaining three models (atm, fbm and sbm), CONDOR performs progressively better moving from 1D to 3D. As noticed earlier (figure 1), the distinctive features among these models are more subtle and can be better evaluated for higher dimensions as they are intrinsically richer in information content. While CONDOR performs equally well with fbm and sbm ($F_1 \approx 0.81$ – 0.82 in 1D, $F_1 \approx 0.85$ – 0.86 in 2D and $F_1 \approx 0.87$ – 0.89 in 3D), it struggles more with atm as this model can be more easily confused with others (figure 1). Nonetheless, CONDOR performance increases significantly with the dimension of the atm trajectories from $F_1 \approx 0.66$ in 1D to $F_1 \approx 0.88$ in 3D.

The performance of CONDOR with different values of SNR gives the expected results (figure 6(e)). The F_1 score increases for higher values of SNR, reaching F_1 scores of 0.89, 0.92 and 0.95 for 1D, 2D and 3D, respectively (i.e. well above the overall F_1 score in figure 5(a)). However, even when the noise overcomes the signal ($\text{SNR} \leq 1$), the lost in the classification performance is less than 10% of these high values for the level of noise considered here.

Finally, figures 6(f)–(h) shows CONDOR classification performance with the duration of the trajectory. Due to the richer information content, classification is increasingly better for longer trajectories and for higher dimensions. The trends in figures 6(f)–(h) are well fitted by power law functions (dashed lines), which show how, above a certain threshold in duration, CONDOR performs above the average F_1 scores (delimited by the shaded areas), reaching F_1 values of up to 0.96 in 1D, 0.97 in 2D and 0.99 in 3D. Moreover, this threshold decreases when increasing the dimension, from ≈ 340 time points in 1D to ≈ 280 in 3D. Note that, even though we did not perform a length specific training for our submissions to the AnDi challenge, by directly training CONDOR neural networks for classification only with short trajectories (≤ 50 time steps), its performance in classifying them can be boosted. This possibility is particularly useful for the classification of experimental trajectories that often are short or are interrupted by missing data points [37, 53]. For example, we were able to improve the F_1 score on short trajectories (≤ 30 time steps) in figures 6(f)–(h) by $\approx 15\%$ on average.

3.3. Analysis of CONDOR performance in the α -exponent inference task

Figure 7 provides further detail on CONDOR performance in the α -exponent inference task by value of α (figures 7(a)–(c)), by model (figure 7(d)), by strength of the localization noise (figure 7(e)) and by trajectory length (figures 5(f)–(h)). These data largely confirm the observations on CONDOR performance for the classification task, thus confirming its robustness across the parameter space.

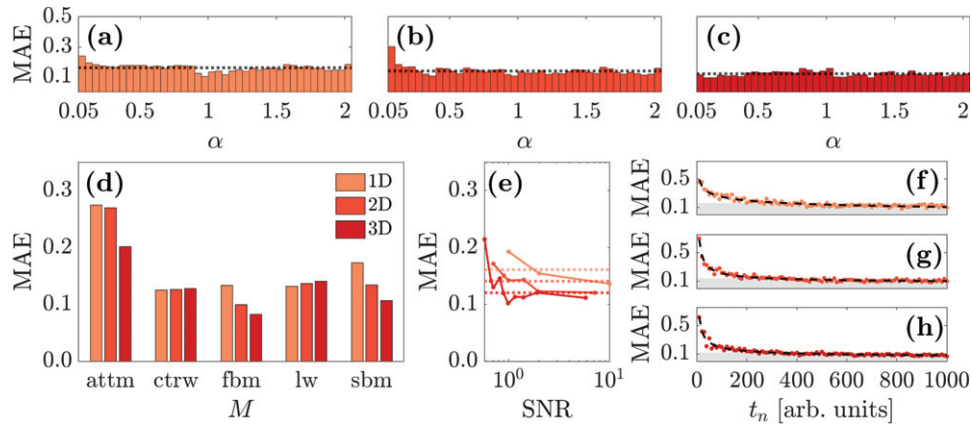


Figure 7. Detailed analysis of CONDOR performance for the α -exponent inference task. CONDOR α -exponent inference performance (MAE) on the 1D, 2D and 3D datasets of the AnDi challenge used for the inference task (as in figure 5(b)) (a)–(c) by value of α for the (a) 1D, (b) 2D and (c) 3D case, (d) by model, (e) by signal-to-noise ratio (SNR), and (f)–(h) by trajectory length for the (f) 1D, (g) 2D and (h) 3D case, respectively. In (a)–(c) each bar corresponds to an increment in α of 0.05. In (a)–(c) and (e) the dotted lines represent the overall MAE as in figure 5(b). In (f)–(h), the trends are fitted to a power law ($y = ax^{-b} + c$, with $a, b, c > 0$) for ease of visualization. The dashed areas highlight the values of the MAE below the average value (figure 5(b)).

In figures 7(a)–(c), we can see how CONDOR performs in a relatively consistent way across the α values, with the exception of $\alpha = 0.05$ in the 1D and 2D cases (MAE = 0.239 in 1D and MAE = 0.302 in 2D). This is primarily due to a relatively large proportion of 1D and 2D attm trajectories with $\alpha = 0.05$ being assigned to other α values: for example, for these trajectories, $\approx 10\%$ of the predicted values of α is greater than 1 (against only the 0.03% of the 3D attm trajectories with $\alpha = 0.05$). As in the classification case, being these subdiffusive trajectories almost stationary, the inference is largely influenced by the localization noise.

As already noted for the model classification (figure 6(a)), CONDOR can also infer the α -exponent very well and consistently across the dimensions for two models (ctrw and lw) with MAE ≤ 0.128 for ctrw and MAE ≤ 0.141 for lw (figure 7(d)). For the three remaining models (attm, fbm and sbm), CONDOR again performs progressively better moving from 1D to 3D. The best results across the dimensions are obtained for the fbm (MAE ≈ 0.133 in 1D, MAE ≈ 0.100 in 2D and MAE ≈ 0.083 in 3D), followed by sbm (MAE ≈ 0.173 in 1D, MAE ≈ 0.134 in 2D and MAE ≈ 0.107 in 3D). Once again, as in the classification task, CONDOR struggles more with attm (MAE ≈ 0.274 in 1D, MAE ≈ 0.269 in 2D and MAE ≈ 0.201 in 3D).

Figure 7(e) confirms what has been discussed for the classification regarding CONDOR performance with different SNR values. For high SNRs, the MAE values for each dimension are well below the average MAE (figure 5(b)): MAE = 0.136 in 1D, MAE = 0.121 in 2D and MAE = 0.112 in 3D. As expected, the MAE increases with higher levels of noise in all dimensions, but it is still below ≈ 0.2 for the values of noise considered here.

Figures 7(f)–(h) shows CONDOR performance in inferring the α exponent with the duration of the trajectory. Similar to the classification task, CONDOR performance improves for higher dimensions and for longer trajectories as the MAE becomes increasingly lower in agreement with prior observations [35, 37]. The trends in figures 7(f)–(h) are well fitted by power law functions (dashed lines), which show how, above a certain threshold in duration, CONDOR performance is significantly below the average MAE (delimited by the shaded areas),

Table 3. Comparison of the mean absolute errors (MAE) obtained with different approaches to the inference task on the AnDi challenge dataset: MAE_{Chal} corresponds to the standard outcome of CONDOR (figure 5(b)), MAE_{Best} and $\text{MAE}_{\text{Worst}}$ represent the outcomes of CONDOR when all trajectories are either classified correctly or completely misclassified, MAE_{Mod} is obtained from the second approach to the inference task ($\tilde{\alpha}_2$ in figure 2) when the predicted model is not added to the inputs for the corresponding neural networks. Finally, MAE_{Rand} is obtained assigning to each trajectory a random value of α uniformly; this value is an average over 5 different attempts (the standard deviation is also provided).

		MAE_{Chal}	MAE_{Best}	$\text{MAE}_{\text{Worst}}$	MAE_{Mod}	MAE_{Rand}
1D	$\tilde{\alpha}_1$	0.171	0.159	0.467		
	$\tilde{\alpha}_2$	0.170	0.159	0.316	0.176	0.665 ± 0.003
	$\tilde{\alpha}$	0.161	0.149	0.391		
2D	$\tilde{\alpha}_1$	0.153	0.142	0.496		
	$\tilde{\alpha}_2$	0.147	0.143	0.257	0.151	0.662 ± 0.002
	$\tilde{\alpha}$	0.141	0.133	0.377		
3D	$\tilde{\alpha}_1$	0.131	0.127	0.531		
	$\tilde{\alpha}_2$	0.127	0.125	0.245	0.132	0.666 ± 0.003
	$\tilde{\alpha}$	0.121	0.117	0.388		

reaching values as small as 0.083 in 1D, 0.085 in 2D and 0.065 in 3D. As for the classification task, this threshold decreases when increasing the dimension, from 350 time points in 1D to 300 in 3D. Also for the inference task, it is possible to improve the prediction of the value of α for short trajectories (≤ 40 time steps) by training CONDOR neural networks only using short trajectories (≤ 50 time steps). Differently from the classification task, training CONDOR's networks for both classification and inference on short trajectories (which we did not perform as part of our submissions to the AnDi challenge) does not significantly reduce the error on the inference of the α exponent. The main limiting factor is indeed the amount of information that can be extracted from short trajectories with the statistical observables used as inputs, which CONDOR is already performing at its best with networks trained on mixed length trajectories.

Finally, it is important to note that CONDOR's inference performance relies on the performance of the classification task. Indeed, the inference of the α exponent uses the predicted model either as a selection element or as an additional input, so the misclassification of trajectories can be directly detrimental for the inference task too. To test the robustness to misclassification more quantitatively, we computed the value of the MAE obtained with both inference methods ($\tilde{\alpha}_1$ and $\tilde{\alpha}_2$ in figure 2) and their average ($\tilde{\alpha}$) in the best-case scenario (all trajectories are classified correctly) and in the worst-case scenario (all trajectories are misclassified in any possible model but the correct one) using the ground truth information about the model in the AnDi challenge dataset for the inference task. For reference, we also compared our results with the MAE calculated assigning to each trajectory a random α value drawn from a uniform distribution (over 5 different attempts). Table 3 shows the correlation between CONDOR's inference performance and the misclassification issue. A perfect classification (MAE_{Best}) allows us to slightly reduce the overall MAE reported in the AnDi challenge (MAE_{Chal} as in figure 5(b)). This is further confirmation of the fact that CONDOR is highly performant in classifying anomalous diffusion trajectories and, consequently, in inferring their anomalous exponent. In the case of good classification, averaging between the two methods is also likely to help reduce the impact of potential misclassifications. Nonetheless, when misclassification becomes more of an issue (e.g. with trajectories heavily corrupted by localization noise), the overall performance of the inference starts to deteriorate. Interestingly,

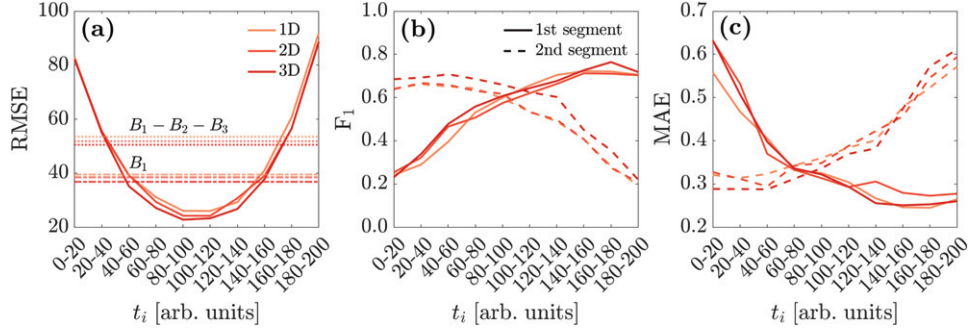


Figure 8. Detailed analysis of CONDOR performance for the segmentation task. CONDOR segmentation results on the 1D, 2D and 3D datasets of the AnDi challenge used for the segmentation task (10k trajectories each). (a) CONDOR segmentation performance is evaluated by the root mean square error (RMSE) of the change point localization as a function of the ground truth value of the change point (t_i). The dotted lines represent the overall RMSE for each dimension when using moving windows of width $B_1 = 50$, $B_2 = 40$ and $B_3 = 20$ sequentially (see text). The dash-dotted lines represent the average RMSE for each dimension on all change points identified only with a moving window of width B_1 (i.e. for $25 < t_i < 175$). (b) Model classification and (c) α -exponent inference with CONDOR on each of the two trajectory segments identified through segmentation. These results are reported by (b) F_1 score and (c) MAE, respectively, for the first (solid lines) and the second (dashed lines) trajectory segment as a function of t_i . The overall F_1 score (i.e. calculated over all possible segment lengths) is 0.533 in 1D, 0.534 in 2D, and 0.572 in 3D, respectively. The overall MAE is 0.372 in 1D, 0.381 in 2D, and 0.371 in 3D, respectively.

this trend is primarily driven by the first method ($\tilde{\alpha}_1$, where the classification outcome is a selection parameter for the choice of the network), while the second inference method ($\tilde{\alpha}_2$, where the classification outcome is an additional input rather than a selection parameter) is much more robust to classification errors and becomes, on its own, a better choice than the average of the two methods. An even better choice when misclassification is of serious concern is to use a revised version of the second inference method where the classification is no longer used as an additional input (MAE_{Mod}): this revised method (fully independent of the classification step) achieves a performance which is comparable to the performance obtained during the AnDi challenge with the first approach ($\tilde{\alpha}_1$) independently. Under all circumstances, CONDOR performs substantially better than assigning the α exponent at random (MAE_{Rand}).

4. Expanding CONDOR to the segmentation of trajectories

We tested our algorithm for the segmentation of trajectories on the same datasets used in the AnDi challenge for the segmentation task [42]. For each dimension, a dataset is composed of 10k trajectories 200 time steps long [41]. Similar to the datasets for the classification and the inference tasks, these datasets are also corrupted by Gaussian noise corresponding to a finite localization precision. Moreover, in these datasets, at least one parameter between the model and the α -exponent changes at a random time t_i within the duration of each trajectory. To predict the change point, we used a moving window of width $B_1 = 50$ at first as explained in section 2.6. We then used two moving windows of width $B_2 = 40$ and $B_3 = 20$ sequentially to refine the prediction of the change point in trajectories where the previous moving window did not detect a change. As our segmentation algorithm is based on using short segments,

we retrained CONDOR's networks for classification and inference using only short trajectories (≤ 50 data points) for an improved segmentation performance. The results of our segmentation are reported in figure 8. As shown in figure 8(a), the error in the determination of the change point is strongly influenced by its location in time, reaching its smallest value when the change point is located nearer the center of each trajectory: whilst the overall RMSE for 1D, 2D and 3D trajectories are 53.45, 51.81 and 50.54 respectively, the RMSE reaches average values of 26.00 ± 0.02 in 1D, 24.20 ± 0.04 in 2D and 23.01 ± 0.37 in 3D when the change point is located between 80 and 120 time steps. Interestingly, when the change point location can be determined using larger windows of width B_1 (i.e. for $25 < t_i < 175$), even the overall RMSE is significantly reduced (RMSE equal to 39.38 in 1D, 38.41 in 2D and 36.75 in 3D, figure 8(a)), due to CONDOR's better performance in predicting the model and the α exponent when more time points are available. Once the position of the change point is known, CONDOR can be used to classify the model and infer the α exponent for each segment of the trajectory determined by its segmentation. We either used the networks identified in figure 4 for long segments ($T_{\max} > 40$) or those trained on short trajectories for short segments ($T_{\max} \leq 40$). The overall F_1 score (i.e. calculated over all possible segment lengths) is 0.533 in 1D, 0.534 in 2D, and 0.572 in 3D, while the overall MAE is 0.372 in 1D, 0.381 in 2D, and 0.371 in 3D. When looking at the F_1 score and the MAE as a function of segment length, figures 8(b) and (c) confirms how CONDOR performance in classifying the model and inferring the α exponent increases with the length of the trajectory, which is here determined by the location of the change point in time for each identified trajectory segment. The extra uncertainty introduced by the localization of the change point however leads to slightly lower F_1 and slightly higher MAE values (for both their overall and length-specific values) across the dimensions with respect to those in figures 6(f)–(h) and 7(f)–(h) for trajectories of the same length. When compared against the other methods submitted to the AnDi challenge for the segmentation task (and using the same metric) [40], our method is always outperformed by the best method for segmentation in each dimension. Nonetheless, our method reaches an overall satisfactory performance comparable with the other submissions beyond the top-ranking one. Indeed, our method would have ranked 2nd in 3D (out of 4) and 4th in 1D and 2D (out of 5). Importantly, considering that the segmentation task of the AnDi challenge was the least subscribed, our method adds a further available tool to study the segmentation of anomalous diffusion trajectories, and could be useful for a preliminary assessment of the presence of segmented trajectories in real data based on CONDOR's outputs.

5. Conclusion

In conclusion, we have introduced a new method (CONDOR) for the characterisation of single anomalous diffusion trajectories in response to the AnDi challenge [40, 41]. Our method combines tools from classical statistics analysis with deep learning techniques. CONDOR can identify the anomalous diffusion model and its exponent with high accuracy, in relatively short times and without the need of any *a priori* information. Moreover, CONDOR performance is robust to the addition of localization noise. As a consequence, CONDOR was among the top performant methods in the AnDi challenge (<https://andi-challenge.org>), performing progressively better for higher dimension trajectories. A deeper understanding of the relative relevance of different inputs for the deep learning architecture can be gained by employing advanced interpretability algorithms [54]. This information could be important to optimise the performance of CONDOR as a function of the most relevant input parameters, and, in the future, to develop an in-depth understanding of how the various statistical observables are connected to each model being classified and each α value being inferred. At the expenses of an increased

computational cost, performance can be further improved by expanding the set of inputs fed to the deep learning algorithm and by employing more advanced network architectures (e.g. convolutional neural networks or recurrent neural networks [55]). While most advanced machine learning techniques often operate as black boxes, we envisage that CONDOR, being partially based on classical statistics analysis, can help characterise the underlying diffusion processes with greater physical insight in many real-life phenomena, from molecular encounters and cellular signalling, to search strategies and the spread of diseases, to trends in financial markets and climate records.

Acknowledgments

The authors would like to thank the organisers of the AnDi challenge for providing a stimulating scientific environment around anomalous diffusion and its characterization. The authors acknowledge sponsorship for this work by the US Office of Naval Research Global (Award No. N62909-18-1-2170).

Data availability statement

All data that support the findings of this study are included within the article (and any supplementary files).

ORCID iDs

Giorgio Volpe  <https://orcid.org/0000-0001-9993-5348>

References

- [1] Metzler R, Jeon J-H, Cherstvy A G and Barkai E 2014 Anomalous diffusion models and their properties: non-stationarity, non-ergodicity, and ageing at the centenary of single particle tracking *Phys. Chem. Chem. Phys.* **16** 24128–64
- [2] Joseph K and Sokolov I M 2005 Anomalous diffusion spreads its wings *Phys. World* **18** 29
- [3] Barkai E, Garini Y and Metzler R 2012 Strange kinetics of single molecules in living cells *Phys. Today* **65** 29
- [4] Felix H and Franosch T 2013 Anomalous transport in the crowded world of biological cells *Rep. Prog. Phys.* **76** 046602
- [5] Krapf D and Metzler R 2019 Strange interfacial molecular dynamics *Phys. Today* **72** 48–54
- [6] Viswanathan G M, GE Da Luz M, Raposo E P and Stanley H E 2011 *The Physics of Foraging: An Introduction to Random Searches and Biological Encounters* (Cambridge: Cambridge University Press)
- [7] Volpe G and Volpe G 2017 The topography of the environment alters the optimal search strategy for active particles *Proc. Natl Acad. Sci. USA* **114** 11350–5
- [8] Brockmann D, Hufnagel L and Geisel T 2006 The scaling laws of human travel *Nature* **439** 462–5
- [9] González M C, Hidalgo C A and Barabási A-L 2008 Understanding individual human mobility patterns *Nature* **453** 779–82
- [10] Zhang H and Li G H 2020 Anomalous epidemic spreading in heterogeneous networks *Phys. Rev. E* **102** 012315
- [11] Tirnakli U and Tsallis C 2020 Epidemiological model with anomalous kinetics: early stages of the covid-19 pandemic *Front. Phys.* **8** 557
- [12] Plerou V, Gopikrishnan P, Nunes Amaral L A, Gabaix X and Eugene Stanley H 2000 Economic fluctuations and anomalous diffusion *Phys. Rev. E* **62** R3023
- [13] Michael F and Johnson M D 2003 Financial market dynamics *Physica A* **320** 525–34

- [14] Massah M and Kantz H 2016 Confidence intervals for time averages in the presence of long-range correlations, a case study on earth surface temperature anomalies *Geophys. Res. Lett.* **43** 9243–9
- [15] Meroz Y and Sokolov I M 2015 A toolbox for determining subdiffusive mechanisms *Phys. Rep.* **573** 1–29
- [16] Golding I and Cox E C 2006 Physical nature of bacterial cytoplasm *Phys. Rev. Lett.* **96** 098102
- [17] Caspi A, Granek R and Elbaum M 2000 Enhanced diffusion in active intracellular transport *Phys. Rev. Lett.* **85** 5655
- [18] Kepten E, Weron A, Sikora G, Burnecki K and Garini Y 2015 Guidelines for the fitting of anomalous diffusion mean square displacement graphs from single particle tracking experiments *PLOS One* **10** e0117722
- [19] Michalet X 2010 Mean square displacement analysis of single-particle trajectories with localization error: Brownian motion in an isotropic medium *Phys. Rev. E* **82** 041914
- [20] Weber S C, Spakowitz A J and Theriot J A 2010 Bacterial chromosomal loci move subdiffusively through a viscoelastic cytoplasm *Phys. Rev. Lett.* **104** 238102
- [21] Krapf D, Marinari E, Metzler R, Oshanin G, Xu X and Squarcini A 2018 Power spectral density of a single Brownian trajectory: what one can and cannot learn from it *New J. Phys.* **20** 023029
- [22] Krapf D *et al* 2019 Spectral content of a single non-Brownian trajectory *Phys. Rev. X* **9** 011019
- [23] Tejedor V, Bénichou O, Voituriez R, Jungmann R, Simmel F, Selhuber-Unkel C, Oddershede L B and Metzler R 2010 Quantitative analysis of single particle trajectories: mean maximal excursion method *Biophys. J.* **98** 1364–72
- [24] Jeon J-H, Leijne N, Oddershede L B and Metzler R 2013 Anomalous diffusion and power-law relaxation of the time averaged mean squared displacement in worm-like Micellar solutions *New J. Phys.* **15** 045011
- [25] Burnecki K, Kepten E, Garini Y, Sikora G and Weron A 2015 Estimating the anomalous diffusion exponent for single particle tracking data with measurement errors-an alternative approach *Sci. Rep.* **5** 1–11
- [26] Thapa S, Wylomanska A, Sikora G, Wagner C, Krapf D, Kantz H, Chechkin A and Metzler R 2021 Leveraging large-deviation statistics to decipher the stochastic properties of measured trajectories *New J. Phys.* **23** 013008
- [27] Hinsin K and Kneller G R 2016 Communication: a multiscale Bayesian inference approach to analyzing subdiffusion in particle trajectories *J. Chem. Phys.* **145** 151101
- [28] Makarava N, Benmehdi S and Holschneider M 2011 Bayesian estimation of self-similarity exponent *Phys. Rev. E* **84** 021109
- [29] Monnier N, Guo S-M, Mori M, He J, Lénárt P and Bathe M 2012 Bayesian approach to MSD-based analysis of particle motion in live cells *Biophys. J.* **103** 616–26
- [30] Thapa S, Lomholt M A, Krog J, Cherstvy A G and Metzler R 2018 Bayesian analysis of single-particle tracking data using the nested-sampling algorithm: maximum-likelihood model selection applied to stochastic-diffusivity data *Phys. Chem. Chem. Phys.* **20** 29018–37
- [31] Cherstvy A G, Thapa S, Wagner C E and Metzler R 2019 Non-Gaussian, non-ergodic, and non-Fickian diffusion of tracers in Mucin hydrogels *Soft Matter* **15** 2526–51
- [32] Serov A S, Laurent F, Floderer C, Perronet K, Favard C, Muriaux D, Westbrook N, Vestergaard C L and Masson J-B 2020 Statistical tests for force inference in heterogeneous environments *Sci. Rep.* **10** 1–12
- [33] Magdziarz M, Weron A, Burnecki K and Klafter J 2009 Fractional Brownian motion versus the continuous-time random walk: a simple test for subdiffusive dynamics *Phys. Rev. Lett.* **103** 180602
- [34] Weron A, Janczura J, Boryczka E, Sungkaworn T and Calebiro D 2019 Statistical testing approach for fractional anomalous diffusion classification *Phys. Rev. E* **99** 042149
- [35] Muñoz-Gil G, Garcia-March M A, Manzo C, Martín-Guerrero J D and Lewenstein M 2020 Single trajectory characterization via machine learning *New J. Phys.* **22** 013010
- [36] Janczura J, Kowalek P, Loch-Olszewska H, Szwabiński J and Weron A 2020 Classification of particle trajectories in living cells: machine learning versus statistical testing hypothesis for fractional anomalous diffusion *Phys. Rev. E* **102** 032402
- [37] Bo S, Schmidt F, Eichhorn R and Volpe G 2019 Measurement of anomalous diffusion using recurrent neural networks *Phys. Rev. E* **100** 010102
- [38] Kowalek P, Loch-Olszewska H and Szwabiński J 2019 Classification of diffusion modes in single-particle tracking data: feature-based versus deep-learning approach *Phys. Rev. E* **100** 032410

- [39] Granik N, Weiss L E, Nehme E, Levin M, Chein M, Perlson E, Roichman Y and Shechtman Y 2019 Single-particle diffusion characterization by deep learning *Biophys. J.* **117** 185–92
- [40] Muñoz-Gil G *et al* 2021 Objective comparison of methods to decode anomalous diffusion (arXiv:2105.06766)
- [41] Muñoz-Gil G, Volpe G, Garcia-March M A, Metzler R, Lewenstein M and Manzo C 2020 Andi: the anomalous diffusion challenge (arXiv:2003.12036)
- [42] Muñoz-Gil G, Requena B, Volpe G, Garcia-March M A and Manzo C 2021 The anomalous diffusion challenge dataset <https://doi.org/10.5281/zenodo.4775311>
- [43] Massignan P, Manzo C, Torreno-Pina J A, García-Parajo M F, Lewenstein M and Lapeyre G J Jr 2014 Nonergodic subdiffusion from Brownian motion in an inhomogeneous medium *Phys. Rev. Lett.* **112** 150603
- [44] Scher H and Montroll E W 1975 Anomalous transit-time dispersion in amorphous solids *Phys. Rev. B* **12** 2455
- [45] Mandelbrot B B and Van Ness J W 1968 Fractional Brownian motions, fractional noises and applications *SIAM Rev.* **10** 422–37
- [46] Klafter J and Zumofen G 1994 Lévy statistics in a Hamiltonian system *Phys. Rev. E* **49** 4873
- [47] Lim S C and Muniandy S V 2002 Self-similar Gaussian processes for modelling anomalous diffusion *Phys. Rev. E* **66** 021114
- [48] Gentili A and Volpe G 2021 Classifier of anomalous diffusion trajectories (CONDORv1.0) <https://doi.org/10.5281/zenodo.4926564>
- [49] Lilly J M 2017 Element analysis: a wavelet-based method for analysing time-localized events in noisy time series *Proc. R. Soc. A* **473** 20160776
- [50] Pincus S M 1991 Approximate entropy as a measure of system complexity *Proc. Natl Acad. Sci.* **88** 2297–301
- [51] Killick R, Fearnhead P and Eckley I A 2012 Optimal detection of changepoints with a linear computational cost *J. Am. Stat. Assoc.* **107** 1590–8
- [52] Pedregosa F *et al* 2011 Scikit-learn: machine learning in Python *J. Mach. Learn. Res.* **12** 2825–30
- [53] Elf J and Barkefors I 2019 Single-molecule kinetics in living cells *Annu. Rev. Biochem.* **88** 635–59
- [54] Samek W, Montavon G, Lapuschkin S, Anders C J and Müller K-R 2021 Explaining deep neural networks and beyond: a review of methods and applications *Proc. IEEE* **109** 247–78
- [55] Argun A, Volpe G and Bo S 2021 Classification, inference and segmentation of anomalous diffusion with recurrent neural networks *J. Phys. A: Math. Theor.* <https://doi.org/10.1088/1751-8121/ac070a>