JPhys Photonics

INTERNATIONAL YEAR OF Quantum Science and Technology

PURPOSE-LED PUBLISHING™

PAPER • OPEN ACCESS

Bottom-up iterative anomalous diffusion detector (BI-ADD)

To cite this article: Junwoo Park *et al* 2025 *J. Phys. Photonics* **7** 045027

View the article online for updates and enhancements.

You may also like

- X-Ray Superflares from Pre-main-sequence Stars: Flare Energetics and Frequency
Konstantin V. Getman and Eric D. Feigelson
- Detecting global financial crises with scarce data by multivariate nonlinear filtering
Cécile Bastidon, Antoine Parent, Patrice Abry et al.
- Multi-component diffusion theory of quantum well intermixing
Tommy Murphy, Christopher A Broderick, Frank H Peters et al.

This content was downloaded from IP address 193.54.89.80 on 13/10/2025 at 09:54

Journal of Physics: Photonics

OPEN ACCESS

RECEIVED
29 January 2025

REVISED
3 August 2025

ACCEPTED FOR PUBLICATION
15 August 2025

PUBLISHED
30 September 2025

Original Content from this work may be used under the terms of the Creative Commons Attribution 4.0 licence.

Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

PAPER

Bottom-up iterative anomalous diffusion detector (BI-ADD)

Junwoo Park[1,*], Nataliya Sokolovska[1], Clément Cabriel[2], Ignacio Izeddin[2] and Judith Miné-Hattab[1]

[1] Laboratory of Computational, Quantitative and Synthetic Biology, CNRS, Institut de Biologie Paris-Seine, Sorbonne Université, Paris, France
[2] Institut Langevin, ESPCI Paris, Université PSL, CNRS, Paris, France
[*] Author to whom any correspondence should be addressed.

E-mail: junwoo.park@sorbonne-universite.fr

Keywords: single particle tracking (SPT), anomalous diffusion, fractional Brownian motion, AnDi2 challenge 2024, trajectory segmentation, neural network

Supplementary material for this article is available online

Abstract

In recent years, the segmentation of short molecular trajectories with varying diffusive properties has drawn particular attention of researchers, since it allows studying the dynamics of a particle. In the past decade, machine learning methods have shown highly promising results, also in changepoint detection and segmentation tasks. Here, we introduce a novel iterative method to identify the changepoints in a molecular trajectory, i.e. frames, where the diffusive behavior of a particle changes. A trajectory in our case follows a fractional Brownian motion and we estimate the diffusive properties of the trajectories. The proposed Bottom-up iterative anomalous diffusion detector (BI-ADD) combines unsupervised and supervised learning methods to detect the changepoints. Our approach can be used for the analysis of molecular trajectories at the individual level and also be extended to multiple particle tracking, which is an important challenge in fundamental biology. We validated BI-ADD in various scenarios within the framework of the 2nd anomalous diffusion challenge 2024 dedicated to single particle tracking. Our method is implemented in Python and is publicly available for research purposes.

1. Introduction

Finding a pattern in a continuous random process to predict the future from limited observations has been extensively studied to predict stock index, consumer's patterns in economics, trajectories of molecules in physics and fractals in mathematics [1–3]. In 1951, the British hydrologist Harold Edwin Hurst introduced the Hurst exponent H to determine the dam size of the Nile river depending on the periodical rain observed in Egypt [4]. Since then, various methods have been developed to estimate H, the long-range dependence in a random process, from classical methods such as rescaled range (R/S) analysis [4] to machine learning based methods [5–8], with the goal to estimate H as accurate as possible.

In biology, with the recent development of single molecule microscopy [9, 10], it became possible to rapidly image the dynamics of thousands of proteins in living cells. By quantifying the dynamics of protein populations, we can observe protein interactions or explore the activity of novel pharmaceuticals with previously unmatched resolution. The molecular diffusion can be described by the anomalous diffusion coefficient α (where $\alpha = 2H$) and the generalized diffusion coefficient K; α and K represent the dependence of molecular diffusion over time and the magnitude of diffusivity respectively. More precisely, the anomalous exponent α represents the degree of recurrence of DNA exploration, that is, the number of times a DNA locus re-iteratively scans neighboring regions before reaching a distant position. When α is low, the locus explores recurrently the same environment for a long time reaching the same targets, while a high α indicates that the locus is able to explore new environments often. However, the prediction of α still remains challenging mainly due to the short length of molecular trajectories coming from experimental limitations.

© 2025 The Author(s). Published by IOP Publishing Ltd

The low number of observations is a major hurdle which limits the improvement of α estimation and further analysis. The physical limitations of empirical methods in optics and noise in low-resolution images of molecules also increase uncertainty in statistical results. Thus, the accurate estimation of α from short sequences is absolutely needed for the prediction of molecular dynamics and its change in a heterogeneous environment.

In our contribution, we introduce a method to estimate the α and K with neural networks (NNs) from an individual short molecular trajectory following fractional Brownian motion (fBm) [11, 12] which is a generalization of Brownian motion having an auto-covariance function given as:

$$\boldsymbol{E}\left[B_H(t)\,B_H(s)\right] = \frac{1}{2}\left(|t|^{\alpha} + |s|^{\alpha} - |t-s|^{\alpha}\right), \tag{1}$$

where $t \in \mathbb{R}$ and $\alpha \in (0,2)$. The accurate estimation of α and K is crucial to determine the ***changepoint*** in a trajectory, which is the moment where molecular dynamics changes are induced by changes of K and/or α. These changes can reflect the activity of a molecule, such as the binding to a substrate, a sudden change of molecular crowding when entering or exiting a macro-domain.

The classical method to analyze molecular properties at ensemble-level in homogeneous systems is mean squared displacement (MSD) [13]:

$$\mathrm{MSD}(t) = 2nKt^{\alpha}, \tag{2}$$

where n is the dimension. In biology, the MSD represents the amount of space a locus has explored in the nucleus, cytoplasm and membrane which can reveal the nature of molecular motion at ensemble-level. When molecules freely diffuse ($\alpha = 1$), its MSD curve is linear in time and its motion is called Brownian. However, in living cells, the molecular motion is often slower than Brownian diffusion and is called subdiffusive ($0 < \alpha < 1$) [14] and the future trajectory is negatively correlated to the past trajectory. Several types of subdiffusive motion have been observed in biological tasks [15]. When a chromosomal locus is confined inside a sub-volume of the nucleus, the motion is called confined subdiffusion and the MSD exhibits a plateau [16]. When a force or structure that restricts the motion is not a simple confinement but is modulated in time and space with scaling properties, the motion is called subdiffusion [14, 17]. The superdiffusion ($1 < \alpha < 2$) usually indicates that the molecules are able to explore new environments and the future trajectory is positively correlated to the past trajectory. However, MSD is not a proper method in a heterogeneous environment of real data, since it averages over the ensemble of trajectories without considering the change of dynamics for each individual trajectory, and does not fit well in general due to its requirement of long observations. Moreover, MSD is meaningfully valid for ergodic processes such as Brownian motion or continuous-time random walk with limited waiting time.

To analyze the molecular trajectory at individual level and to approximate the changepoints, we propose a novel method to classify molecular trajectories by dividing each trajectory into multiple sub-trajectories. We introduce bottom-up iterative anomalous diffusion detector (BI-ADD) which identifies changepoints of molecular trajectories at individual level when molecules diffuse under fBm. BI-ADD integrates both classical and deep learning approaches for the analysis of molecular trajectories. The predicted changepoints with BI-ADD can indicate the significant changes of α and/or K along the trajectory.

The AnDi challenge [18–20] aims to analyze biological phenomena with inspections of molecular trajectories in cells. The molecular trajectories can be studied collectively in a homogeneous system or individually in a heterogeneous system if the diffusive properties change over time and space. The 2nd anomalous diffusion challenge (AnDi2 Challenge), which is a successive challenge of 1st challenge [20], has been held in 2024 to answer the biological questions in terms of molecular trajectory and we participated in the challenge as SU-FIONA team. The performance of BI-ADD compared to other methods participated in the challenge is available in [19, 21]; we were ranked 1st of 3 teams for the video task and 6th of 18 teams for the trajectory task (http://andi-challenge.org/challenge-2024/#andi2leaderboard). Since the BI-ADD detects the changepoints and the diffusive properties from the molecular trajectory only, we applied FreeTrace [22] to infer the trajectory of molecules from microscopy videos of AnDi2 challenge. The simulated fBm trajectories to train the deep learning models and also to evaluate the obtained results were provided by the AnDi2 challenge [18]. The notations utilized in our algorithm are listed in table 1.

Table 1. Notations.

Symbol	Description	Units
$\mathbf{X}$	x coordinate sequence of trajectory	Pixel
$\mathbf{Y}$	y coordinate sequence of trajectory	Pixel
α	Anomalous diffusion exponent	
K	Generalized diffusion coefficient	$\frac{\text{pixel}^2}{\text{frames}^\alpha}$
T	Length of trajectory	Frames
T_{sub}	Length of sub-trajectory	Frames
cp	Changepoint	Frame
$\mathbf{R}$	Radial displacement sequence	Pixel
λ	Changepoint threshold for $\mathbf{S}$	
N	Number of changepoints in a trajectory	
k	Number of different states (clusters) in a sample	
$\mathbf{X}^t_{w_i}$	tth element inside a sliding window of size w at ith position of $\mathbf{X}$	
$\sigma_{\mathbf{X}^w_i}$	Standard deviation of sliding window of size w at ith position of $\mathbf{X}$	
$\mathbf{V}^w$	Converted signal, generated with sliding window of size w	
$\mathbf{S}$	Signal averaging $\mathbf{V}^w$ With w_{set}	
w_{set}	Set of sliding window sizes	
l	Extension length for each first and last frames	
N_{sub}	Number of sub-trajectories divided for the regression of α	
$\mathbb{T}$	Set of input trajectory lengths for the model$_\alpha$	

2. Methods

2.1. Workflow overview

The main objective of BI-ADD is two-fold:

1. Changepoints detection along the trajectory.
2. Estimation of the molecular diffusive properties α and K of sub-trajectories divided by the changepoints in a heterogeneous molecular system.

In the preprocessing step, we first generate N candidate changepoints by converting the 2-dimensional coordinates of a trajectory into a signal. Second, $N+1$ sub-trajectories are generated by dividing the trajectory with the potential changepoints from the converted signal, and the $\hat{\alpha}$ and $\hat{K}$ of each sub-trajectory are estimated. The diffusive properties are estimated with convolutional Long short-term memory (ConvLSTM) [23–25] and a NN [26] for $\hat{\alpha}$ and $\hat{K}$, respectively. Note that $\hat{\alpha}$ and $\hat{K}$ can be done simultaneously. We repeat the estimation for all trajectories from a given sample. Third, BI-ADD clusters $\hat{\alpha}$ and $\hat{K}$ estimated from the entire sample to determine the total number of different states of the trajectories with the Gaussian mixture model (GMM) [27]. In the main processing step, BI-ADD re-visits each individual sub-trajectory iteratively to determine whether two consecutive sub-trajectories should be merged. This decision is made based on the clustering results and the proximity of two sub-trajectories given a fixed threshold. If BI-ADD merges two consecutive sub-trajectories, it re-estimates the $\hat{\alpha}$ and $\hat{K}$ of the merged new sub-trajectory and repeats the procedure until there are no more unexplored sub-trajectories. The visualized workflow is shown in figure 1(A).

2.2. Detection of potential changepoints

Data. A set of 2-dimensional fBm trajectories (figure 1(B)). The trajectory of length T is a sequence of coordinates $\{(x_1,y_1),\dots,(x_T,y_T)\}$.

Definition 1. *Changepoint* is the time point where the K and/or α change for a given trajectory following fBm and has the following properties.

(a) cp_n (nth changepoint of a trajectory) divides a trajectory into two consecutive sub-trajectories from cp_{n-1} to cp_n and from cp_n to cp_{n+1}.
(b) Two consecutive sub-trajectories divided by a changepoint always have different α and/or K.
(c) N changepoints divde a trajectory into $N+1$ sub-trajectories.
(d) The number of possible changepoints lies between 0 and $T-2$ inclusive for a trajectory of length T.

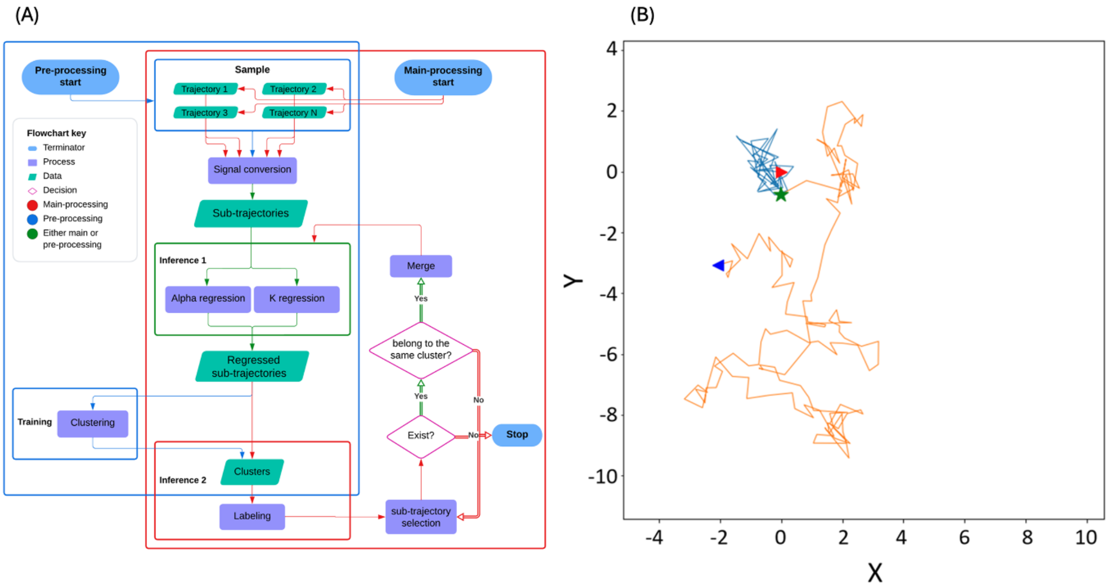

Figure 1. (A) shows the workflow of BI-ADD. In the pre-processing step (blue box), BI-ADD generates clusters from a given sample of trajectories in $\hat{\alpha}$ and $\log\hat{K}$ space. In the main processing step (red box), the selection process contains an ordering of the potential changepoints in the order of lowest value from the normalized signal **S** converted with the equations (3)–(7), and selects the two first non-selected consecutive sub-trajectories. If two selected consecutive sub-trajectories belong to the same cluster with a given significance level, BI-ADD merges the sub-trajectories and repeats the process. (B) Shows an example of two-dimensional fBm trajectory, where α changes from 0.2 to 1.0 while K remains constant. The red and blue triangles indicate the start and end points of the trajectory, respectively. The green star indicates the ground truth changepoint. The length of the simulated fBm trajectory is 200, and the α changes at the 49th frame.

We can see that the finding of true positive changepoints relies on the performance of $\hat{K}$ and $\hat{\alpha}$ for a given sequence length. If we can have a model estimating the $\hat{K}$ and $\hat{\alpha}$ with errors less than the minimum difference of the true K and α for a given set of trajectories, this model also can detect the true positive changepoints accurately since the difference of the K or α is relatively higher than the estimation error. However, a trajectory of length T has $T-2$ changepoints at maximum, which can be divided into $T-1$ sub-trajectories. The sub-trajectories have only one observation of coordinates, except for the last sub-trajectory in the extreme case. The short length of the sub-trajectory is a major hurdle to precisely estimating the diffusion properties of fBm and changepoints due to the stochastic nature of the process. To simplify the problem of detecting changepoints, BI-ADD is built under the following assumptions.

Assumption 1. Given a set of trajectories following fBm, there is a fixed number of populations (i.e. states) with different K and/or α.

Assumption 2. Each population has *a sufficient number of trajectories* for the clustering in the K and α space.

Assumption 3. The frequency of changepoints as a function of trajectory length is unknown.

Note that the clustering is performed with GMM and we do not cope with '*a sufficient number of trajectories*' of the assumption 2 in this article, which depends on the quality of trajectories, noises and the difference of K and α between populations. To identify the clusters, we find potential candidates of changepoints which divide a trajectory into multiple sub-trajectories, including false positives. The $\hat{K}$ and $\hat{\alpha}$ of sub-trajectories decide the total number of different populations $k = |\{(\hat{K}_n, \hat{\alpha}_n)|n \in \{1, 2, \ldots, k\}\}|$. We do not assume the distribution of changepoints for the flexibility in assumption 3. However, given a prior changepoint distribution of a sample as a function of elapsed time in each state, the model utilizing the prior information can decrease the number of false changepoints. The strategy in the next section decreases the computational time by avoiding the search for the true changepoint at every frame.

First, we extend a trajectory with a length of $2l$ by reflecting the extremities, where the first and last coordinates are used as pivot points in the 2-dimensional Euclidean space. This extension of the trajectory is needed to approximate the changepoints, to estimate $\hat{\alpha}$ and $\hat{K}$ of sub-trajectories that are near the extremities of the original trajectory. Next, we convert the extended 2-dimensional trajectory into a signal using multiple sliding windows with the set of window sizes $w_{\text{set}} = \{20, 22, \ldots, 40\}$, i.e. the different numbers of frames considered, as follows:

$$\mathbf{X}^{w}_{i,l} = \left\{\mathbf{X}^{k}_{w_i} - \mathbf{X}^{0}_{w_i} \mid k \in [0, w/2]\right\} \tag{3}$$

$$A^{w}_{i,l} = \sum \left|\mathbf{X}^{w}_{i,l}\right| \tag{4}$$

where $\mathbf{X}^{k}_{w_i}$ corresponds to the kth element of extended $\mathbf{X}$, sliced with a sliding window of size w at ith position of extended sequence. The subscripts l and r of equations (3)–(5) stand for the range of $k \in [0, w/2)$ and $k \in [w/2, w)$ which correspond to the first and second half inside the sliding window respectively. The idea behind equation (3) is to identify the maximal distance a molecule can move in a given time. $A^{w}_{i,l}$ corresponds to the last value in a sequence of CAS (Cumulative absolute sum) inside the first half of the sliding window of size w in X space. $B^{w}_{i,l}$ is computed similarly to A for $Y^{w}_{i,l}$ space. Figure S3 shows the corresponding CAS obtained from the simulated fBm trajectories with the same K but different α values when the length of the sequence is 256. It shows how the CAS of fBm can evolve as a function of sequence length. Equation (5) is the comparison process for a time point i where the sequence is sliced with a sliding window of size w, in terms of the last value of CAS of fBm. The difference of the empirical standard deviation of fGn is added as a second term in equation (5).

$$v^{w}_{i} = \left| \frac{A^{w}_{i,l} - A^{w}_{i,r}}{\max\left(A^{w}_{i,l}, A^{w}_{i,r}\right)} + \frac{B^{w}_{i,l} - B^{w}_{i,r}}{\max\left(B^{w}_{i,l}, B^{w}_{i,r}\right)} \right| + \delta \tag{5}$$

where $\delta = |\ \sigma_{\mathbf{X}^{w}_{i,l}} - \sigma_{\mathbf{X}^{w}_{i,r}}\ | + |\ \sigma_{\mathbf{Y}^{w}_{i,l}} - \sigma_{\mathbf{Y}^{w}_{i,r}}\ |$, σ corresponds to the standard deviation.

The main intuition behind this transformation is that the newly constructed signal reflects the relative α and K differences of two sub-sequences of size $w/2$ inside the sliding window. Note that equation (5) can be extended to a higher-dimensional trajectory since the transformed signal is a linear combination of simple terms.

The $\mathbf{V}^{w}$ is a signal representing the relative difference of sub-sequences inside sliding windows measured for every i except extended range:

$$\mathbf{V}^{w} = \left\{v^{w}_{i} | i \in [l, l+T]\right\}. \tag{6}$$

We sum $\mathbf{V}^{w}$ over all possible window sizes to get $\mathbf{S}$—our signal of interest—the signal integrating information of all considered sliding windows w on multiple scales:

$$\mathbf{S} = \sum_{w \in w_{\text{set}}} \mathbf{V}^{w}, \tag{7}$$

where $i, j, l \in \mathbb{N}$, $w \in 2\mathbb{N}$. Note that the additional points added on the extremities do not influence the result since they are excluded from equation (6).

The choice of the sliding window size depends on the transition probability between sub-trajectories assuming the Markov model in a heterogeneous system: a small size of a sliding window fits well with the high transition probability, large sizes of sliding windows reflect the case of low probability of the transition. An example of a normalized signal $\mathbf{S}$ is illustrated on figure 2. We can observe that a threshold λ divides a trajectory into multiple sub-trajectories by finding the local maxima lying on the normalized signal $\mathbf{S}$. The local maxima above λ can be considered as potential changepoints where the relative difference is highest between two sub-sequences. We can approximate potential changepoints including false positives from the normalized $\mathbf{S}$ and can estimate $\hat{\alpha}$ and $\hat{K}$ for each individual sub-trajectory.

2.3. Estimation of $\hat{\alpha}$ and $\hat{K}$ for each sub-trajectory

The estimations of $\hat{\alpha}$ and $\hat{K}$ are performed with a ConvLSTM network (Model$_{\alpha}$, figure 3) and a fully connected NN (Model$_{K}$, figure 4) respectively. For the training, we generate 1 million observations without noise, 2-dimensional fBm sequences with uniformly selected lengths from 5 to 256. The model architectures are shown on figures 3 and 4. The models are extendable to higher-dimensional sequences for the estimation of both $\hat{\alpha}$ and $\hat{K}$. To optimize the models, we use RMSprop for model$_{\alpha}$ and Adam [28] for model$_{K}$.

Since the convolutional layer accepts a fixed length of input only, the trajectory sequence of length T is divided into N_{sub} sub-sequences where $N_{\text{sub}} = T \bmod 2^{m} + 1$, m is the maximum number which makes 2^{m} smaller or equal to T. We trained the models with $\mathbb{T} = \{5, 8, 12, 16, 32, 64, 128\}$. During the inference, the regressed values of α are averaged if $T - N_{\text{sub}} + 1 \notin \mathbb{T}$. The input features for the α regression are as follows:

$$\boldsymbol{\alpha}_{\mathbf{1}} = \left\{ \frac{1}{\sigma_{\mathbf{X}} T} \sum_{t=0}^{j} |x^{t+1} - x^{t}|\ |\ j \in [0,\ T-1] \right\} \tag{8}$$

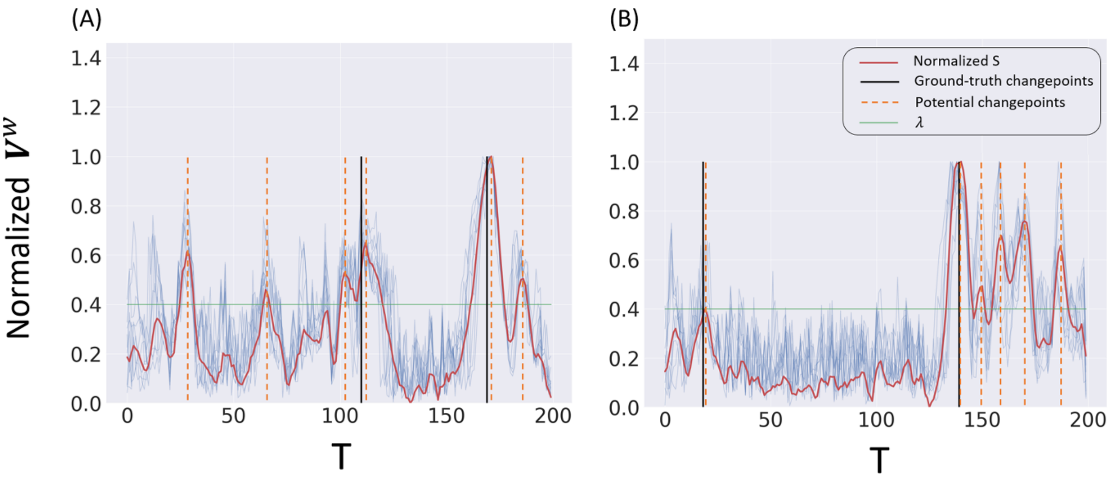

Figure 2. A normalized signal **S** computed with multiple sizes of sliding windows on two simulated trajectories. The black vertical lines show the ground-truth changepoints for each simulated trajectory. The orange dashed vertical lines show the potential changepoints above λ. The green horizontal line corresponds to the threshold value $\lambda = 0.4$. The red signal is normalized **S**. The transparent blue curves correspond to the $\mathbf{V}^w$ with various sizes of sliding window, where $w \in w_{\text{set}}$. The fBm trajectory of (A) is simulated with two states $(K_1, \alpha_1) = (0.01, 0.2)$ and $(K_2, \alpha_2) = (0.1, 1.0)$. (B) is simulated with $(K_1, \alpha_1) = (0.05, 0.5)$ and $(K_2, \alpha_2) = (0.1, 1.5)$. The transition probability between the states for both (A) and (B) is 0.01. The peaks of the signal **S** indicate the detected potential changepoints along the trajectory, where molecular dynamics changes.

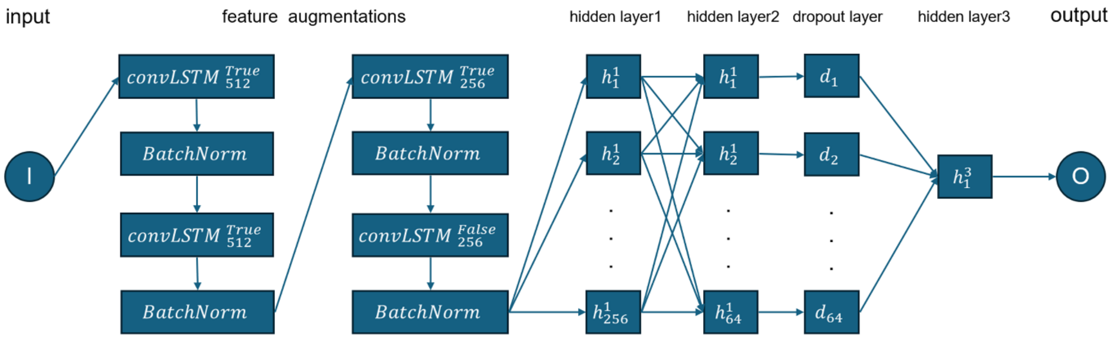

Figure 3. Model$_\alpha$ architecture: estimation of $\hat{\alpha}$. The convLSTM$^{\text{True}}_{512}$ stands for returning a full sequence with 512 convolutional filters, the last output otherwise; h^k_n represents the nth neuron in k_{th} hidden layer. This model takes 3 input features (equations (8)–(10)). The convLSTM layers perform the feature augmentation. The total number of parameters in the model is 17 M.

$$\boldsymbol{\alpha_2} = \left\{\mathbf{R}^j/\overline{\mathbf{XY}}/T \mid j \in [0,\ T]\right\} \tag{9}$$

$$\boldsymbol{\alpha_3} = \left\{\left(\mathbf{X} - \mathbf{X}^0\right)/\overline{\mathbf{XY}}/T\right\}, \tag{10}$$

where $\boldsymbol{\sigma}_\mathbf{X}$, T, $\mathbf{R}^k$ and x^t represent the standard deviation of x coordinate sequence, length of input sequence, radial displacement for a sequence length at k and the tth element in $\mathbf{X}$ respectively. Note that the first value of $\boldsymbol{\alpha_1}$ is padded with 0 for equal length between features. These input features are for the estimation of $\hat{\alpha}$ in terms of X space in a pointwise manner, and we took the average of $\hat{\alpha}$. The same procedure is repeated for Y space and the average of $\hat{\alpha}$ for each dimension is used.

The input feature for K regression is provided as:

$$\mathbf{K_1} = \log \overline{\mathbf{XY}}, \tag{11}$$

where $\overline{\mathbf{XY}} = \frac{1}{T}\sum_{i=0}^{T-1}\sqrt{(x^{t+1}-x^t)^2 + (y^{t+1}-y^t)^2}\ k, T \in \mathbb{N}$.

For a stable estimation of $\hat{\alpha}$, we consider that $\boldsymbol{\alpha_1}$ is the most distinguishable input feature compared to others that represents the normalized absolute cumulative sum of fractional Gaussian noise. The numerical results of model$_\alpha$ and model$_K$ are shown in figure 5. Note that $\boldsymbol{\alpha_1}$ in equation (8) is divided by the empirical standard deviation, which allows estimating the α and K separately. To test the models, we simulated 1000 trajectories for each α_{true} and K_{true}. We can observe that the performance decreases if T gets smaller; in general, this is due to the low number of observations and it leads possibly to a biased result, since fBm is a

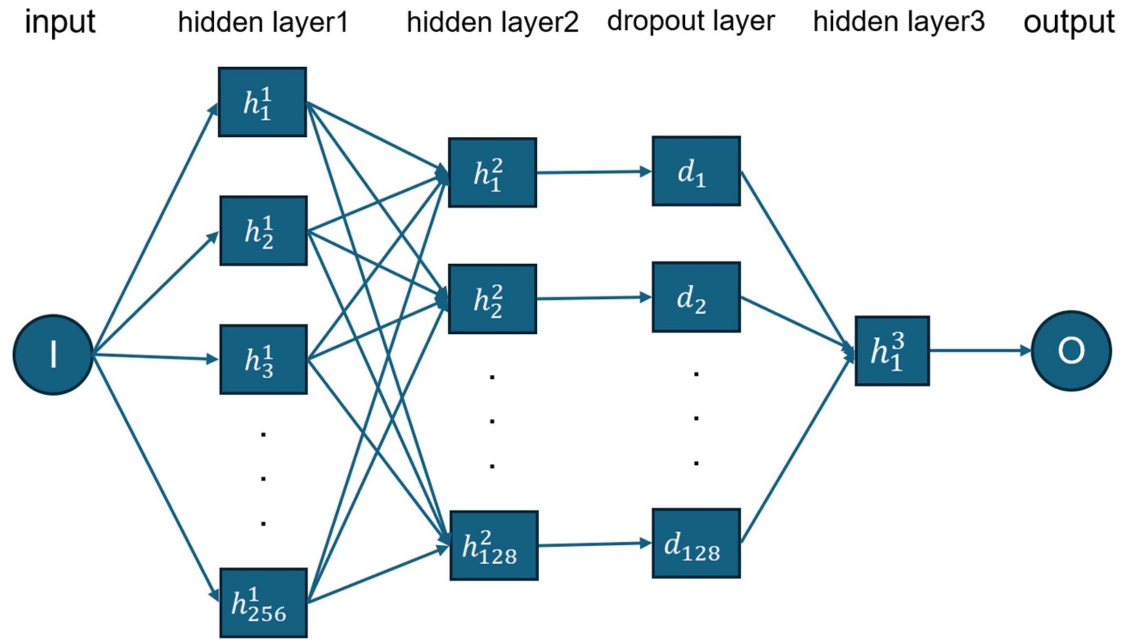

Figure 4. Model$_K$ architecture: estimation of log$\hat{K}$; h_n^j represents nth neuron in jth hidden layer. This model takes one input feature (equation (11)). The dropout layer avoids overfitting of the model. The total number of parameters is 33 K.

continuous Gaussian process. In the interval $\alpha_{\text{true}} = 0.05$ to $\alpha_{\text{true}} = 1.0$, $\sigma_{\hat{\alpha}}$ increases, and drops again from $\alpha_{\text{true}} = 1.0$ to $\alpha_{\text{true}} = 1.95$. For the longer trajectories, where $T > 32$, we can see that the estimated results are the most promising in the neighborhood near $\alpha_{\text{true}} = 1.0$. The estimation of $\hat{K}$ shows reasonable results in general, however, if α_{true} gets bigger, such as $\alpha_{\text{true}} > 1.5$, the MALE criterion (equation (13)) increases rapidly due to the auto-covariance of fBm. In this case, accurate estimation of $\hat{K}$ is nearly impossible unless the first coordinates of a short trajectory are sampled near the true mean of the Gaussian distribution.

2.4. Clustering of $\hat{\alpha}$ and $\hat{K}$ and sequential inference on a trajectory

To filter the regression noise and false positive changepoints, we cluster $\hat{\alpha}$ and log$\hat{K}$ of sub-trajectories from a given sample. The clustering is performed with the GMMs for the sub-trajectories where the length T_{sub} is longer than 16 due to high uncertainty of low number of observations. If the sample size is smaller, the length T_{sub} of sub-trajectories for the clustering should be smaller to obtain enough data for the clustering. However, if the sample size is big enough, we can increase λ to exclude uncertain short sub-trajectories. The optimal number of clusters k is inferred with the Bayesian Information Criterion (BIC) score [29] or can be decided manually. BI-ADD produces the distribution of $\hat{\alpha}$ and log$\hat{K}$ shown in figure 6 and the number of clusters can be deduced from the results.

With the Gaussian mixtures built in the pre-processing step, i.e. GMM clustering with estimated $\hat{\alpha}$, $\hat{K}$ of sub-trajectories on α–K space, we can determine that the approximated potential changepoints from a given trajectory are true positives or false positives. BI-ADD starts the inferences from the lowest value of changepoint in the transformed signal **S**. If two consecutive sub-trajectories divided by the changepoint are estimated to belong to the same cluster with a given significance level (table 2) for each length of sub-trajectory, BI-ADD merges the selected two consecutive sub-trajectories and re-estimates α and K. The selection of clusters is determined by comparing the likelihoods of clusters constructed with GMM. BI-ADD repeats the inference and merges the sub-trajectories until BI-ADD does not detect any possible merging. The significance levels in table 2 were fixed by a heuristic approach. These alternating iterative steps are performed sequentially.

3. Results

To extensively test the proposed approach BI-ADD, we participated in the AnDi2 Challenge as team SU-FIONA for both video task and trajectory task at individual trajectory-level. The simulated scenarios of the AnDi2 Challenge aim to generate the molecular trajectories on micro or nanoscale by mimicking real world problems in biology. The details of comparative results between AnDi2 Challenge participants is available in [21]. For the video task, we applied FreeTrace [22] to predict the individual molecular trajectories from the video and analyzed the predicted trajectories with BI-ADD. For the inferred trajectories,

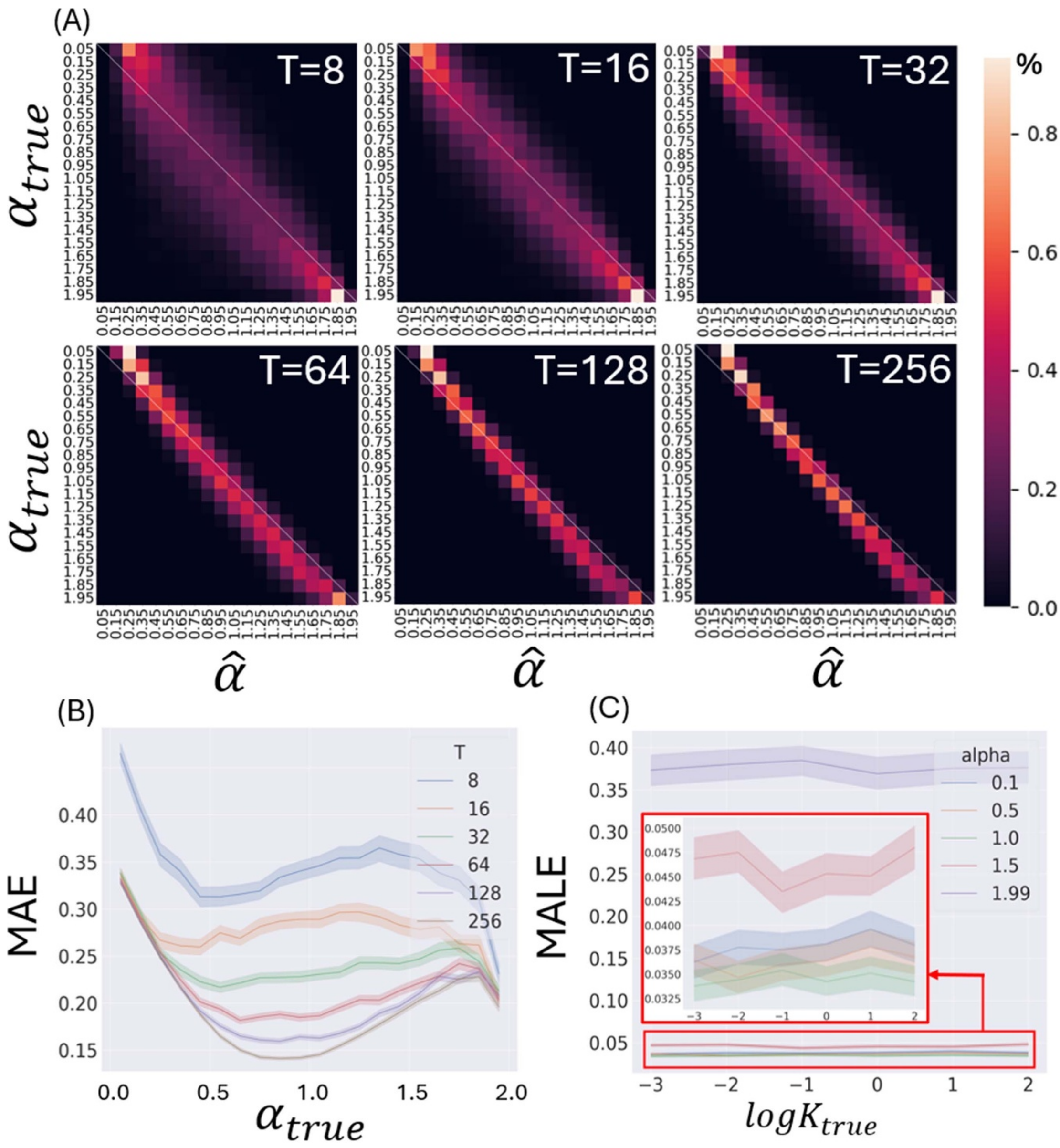

Figure 5. (A): Regression results of $\hat{\alpha}$ on different sequence lengths T with $\alpha_{\text{true}} \in \{0.05, 0.15, \ldots, 1.95\}$; $\hat{\alpha}$ is binned with window 0.1 (to fit into the confusion matrix); 1000 trajectories are simulated for each α_{true}. (B): Mean absolute error (MAE) for α_{true} with different lengths T. The blurred region shows 95% confidence interval. (C): log$\hat{K}$,mean absolute logarithmic error (MALE) for logK_{true} with $\alpha_{\text{true}} \in \{0.1, 0.5, 1.0, 1.5, 1.99\}$. The red box contains the zoomed MALE values of $\alpha_{\text{true}} \in \{0.1, 0.5, 1.0, 1.5\}$. The blurred region shows 95% confidence interval. The estimation of log$\hat{K}$ becomes challenging, if $\alpha_{\text{true}} \in \{\alpha | \alpha > 1.9\}$ due to the auto-covariance of fBm.

BI-ADD was used to predict the changepoints and the diffusive properties of molecular trajectories. We were ranked 1st for the video task, and 6th for the trajectory task.

3.1. Evaluation metrics

We evaluate the performance of the method using the following metrics, also used by the AnDi2 challenge:

MAE between an estimation $\hat{y}_i$ and true value y_i:

$$\mathbf{MAE} = \frac{1}{N}\sum_i |\hat{y}_i - y_i| \,. \tag{12}$$

Mean absolute log error:

$$\mathbf{MALE} = \frac{1}{N}\sum_i |\log \hat{y}_i - \log y_i| \,. \tag{13}$$

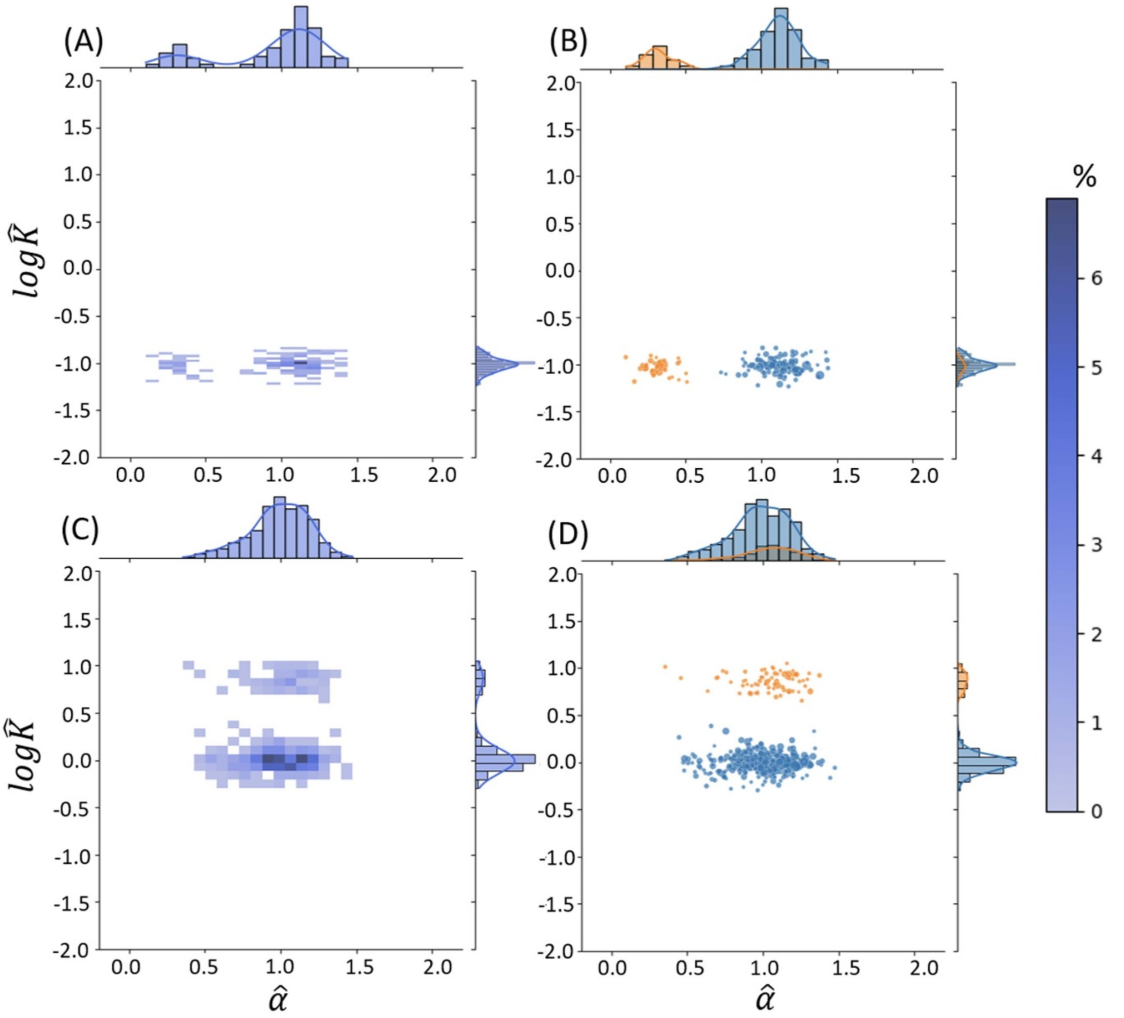

Figure 6. Distributions of $\hat{\alpha}$ and log$\hat{K}$ of sub-trajectories regressed from a given sample. The scatter size corresponds to the normalized length of sub-trajectory. The colorbar represents the percent. (B), (D) show clustered $\hat{\alpha}$ and log$\hat{K}$ of the distribution (A) and (C) with GMM. (A), (C) are the 2-dimensional histogram with marginal distributions of $\hat{\alpha}$ and log$\hat{K}$. (A) and (B) are obtained from the sample where α_{true} changes between 0.3 and 1.0 while K_{true} is 0.1. (C) and (D) are obtained from the sample where K_{true} changes between 0.8 and 1.0 while α_{true} is 1. Both samples are simulated with 1000 trajectories where the transition probability is 0.01. The clustering with GMM is performed for the sub-trajectories where the length is longer than 16.

Root mean squared error (RMSE):

$$\mathbf{RMSE} = \sqrt{\frac{1}{N}\sum_i (\hat{y}_i - y_i)^2}. \tag{14}$$

Jaccard similarity coefficient (JSC), which measures the true positive ratio of changepoints along the trajectory without considering true negatives:

$$\mathbf{JSC} = \frac{\text{TP}}{\text{TP + FP + FN}}, \tag{15}$$

where TP, FP and FN are the numbers of true positives, false positives and false negatives respectively.

Mean squared logarithmic error:

$$\mathbf{MSLE} = \frac{1}{N}\sum_i (\log(1+\hat{y}_i) - \log(1+y_i))^2. \tag{16}$$

3.2. A two-state scenario of molecular trajectories: evaluation of α and K

3.2.1. Data generation

We generate 20 000 2-dimensional fBm trajectories with $T = 200$. The two states of sub-trajectories are described by $\{(K_1, \alpha_1), (K_2, \alpha_2)\}$, where the order of the states is random, and the transition probability between them is 0.02.

Table 2. Significance level for each T_{sub}.

sub-trajectory length(T_{sub})	5	8	12	16	32	64	128
significance level	10^{-5}	10^{-3}	2.5×10^{-2}	10^{-1}	10^{-1}	10^{-1}	10^{-1}

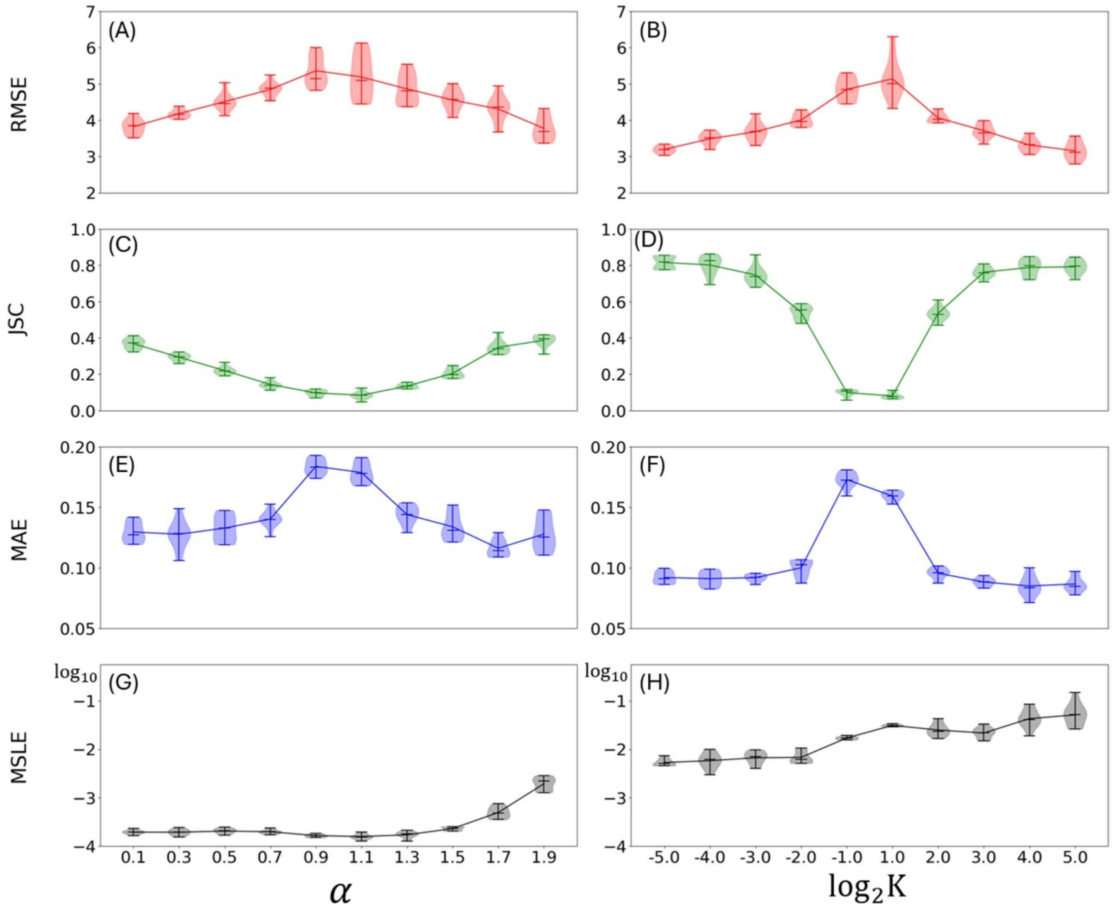

Figure 7. Performance of BI-ADD varying α and K in different scenarios. On the left: 10 sets of scenarios $\{(0.1,\alpha),(0.1,1.0) \mid \alpha \in \{0.1,0.3,\ldots,1.9\}\}$; on the right: $\{(K,1.0),(1.0,1.0) \mid log_2 K \in \{-5,-4,\ldots,5\}\}$, where set = $\{(K_1,\alpha_1),(K_2,\alpha_2)\}$. 1000 fBm trajectories in 2D are simulated for each scenario with the transition probability 0.01. (A) and (B) Show the root mean squared error (RMSE, equation (14)) of true positive changepoints. (C) and (D) show the JSC (equation (15)) of changepoints. (E) and (F) show the MAE of true positive changepoints. (G) and (H) Show the mean squared log error, equation (MSLE (16)) of true positive changepoints. We can observe that if the difference of α or K becomes smaller between two set of states, the errors increase in general. The MSLE is the highest, if $\alpha = 1.9$ due to auto-covariance of fBm and the result is coherent with (C) of the figure 5. The metrics are calculated with AnDi datasets [18].

To explore the effect of various α values, we created one scenario with different α values with fixed same K between two states, $\{(0.1,\alpha),(0.1,1.0) \mid \alpha \in \{0.1,0.3,\ldots,1.9\}\}$, and another scenario with different K with fixed same α between two states, $\{(K,1.0),(1.0,1.0) \mid \log_2 K \in \{-5,-4,\ldots,5\}\}$. The performance is assessed using the metrics mentioned above, the results are shown on figure 7.

To measure the detection rate of cp (frame of a changepoint) in a given sequence, we use the JSC. We consider $\hat{cp}$ (an estimated frame of a changepoint) as true positive if $|\hat{cp} - cp_{\mathrm{true}}| < 10$, and false positive otherwise. The obtained results are intuitive: when the difference between two states of molecular trajectories becomes smaller, BI-ADD underperforms compared to the case, where the difference between two states is significant. The identification of changepoints is more difficult in terms of α than K, and the maximum value of JSC is around 0.4 in figure 7(C), where the difference of α between two states is at the maximum. We can see, however, the maximum JSC is around 0.8 on figure 7(D) which shows the detection of changepoints is easier in terms of K than α. The estimated results of K on figure 7(G) emphasize the difficulty of accurate K estimation, especially, where $\alpha \simeq 1.9$ due to auto-covariance of fBm. In addition to the results of two-state scenarios, we simulated a 3-state scenario where the set of states is $\{(0.1,0.2),(1.0,1.0),(2.5,1.8)\}$. In the 3-state scenario, one population diffuse slowly showing anti-persistent motion, another Brownian population with intermediate diffusion coefficient and the last population with fast and persistent diffusive motion. The result is presented in figure S2 showing the RMSE $= 2.594 \pm 0.325$, JSC $= 0.736 \pm 0.012$,

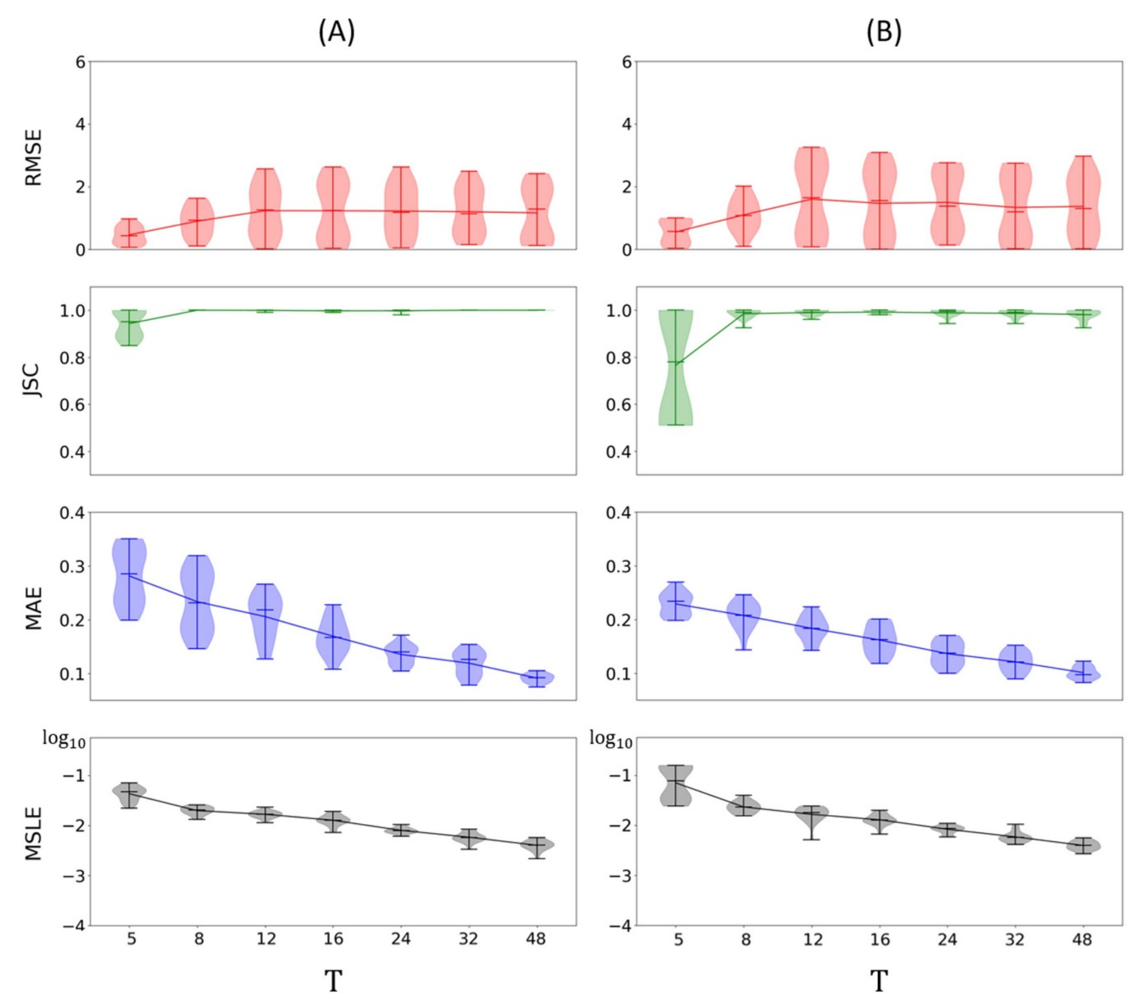

Figure 8. Performance of BI-ADD varying T in two different scenarios. The simulated sets of (K,α) of (A) are $\{(1.0,1.0),(0.01,0.2)\}$ and $\{(1.0,1.0),(0.1,0.5)\}$ for (B). Note that the simulated diffusivity-difference of (B) between two populations is smaller than (A). 500 trajectories of total length $2T$ are simulated for each T with 1 changepoint in the middle of trajectories and 500 trajectories are simulated for each T without changepoints to reflect the false negative sensitivity of BI-ADD. We can observe overall performance increases, if T gets longer except RMSE, since the metrics are calculated only for true positives. The corresponding generated clusters for each T is available in figure S1.

MAE$= 0.131 \pm 0.001$ and MSLE$= 0.051 \pm 0.001$. We can verify the good performance of BI-ADD for 3-state unless the diffusive properties between states are too similar which increase the difficulty of clustering with GMM.

3.3. A two-state scenario of molecular trajectories: the role of T

The evaluation of BI-ADD for different length of short trajectories is shown in the figure 8. Scenario (A) is simulated with $\{(1.0,1.0),(0.01,0.2)\}$ and $\{(1.0,1.0),(0.1,0.5)\}$ for (B). BI-ADD performs well for both scenarios in general except for short trajectories where $T = 5$. If the differences of clustered $\hat{\alpha}$ and $\hat{K}$ between two population are sufficiently large to make distinguishable clusters (figure S1), the score of JSC for short trajectories such as $T = 5$ is acceptable to use. If the distance between the generated clusters is not explicitly distinguishable, the variance of JSC is large for short trajectories and the identified changepoints might contain many false negatives and false positives.

4. Conclusion and discussion

We introduced a novel method—BI-ADD – to approximate the changepoints and estimate the diffusive properties α and K from molecular trajectory. The method is based on three central parameters of BI-ADD. The threshold (λ) of the transformed signal is set to 0.15 by default, λ acts as a false negative controller, its low value can decrease the number of false negatives, but can increase the computational time, if it creates more sub-trajectories. The automated estimation of λ was not tackled in the current contribution, since it is a challenging topic which is out of score in this paper. However, this problem is among our future research directions.

The size of the sliding windows (w_i) are set to $i \in \{20, 22, \ldots, 40\}$ by default. An optimal size of a sliding window is related to the transition probability between the states of trajectories: if the transition probability is low, the sliding window should not be small, since small windows are sensitive to the noise and the number of false positives will increase. The total number k of the hidden states of a given sample is set to -1 by default, and an optimal k is estimated with the BIC. However, the BIC criterion might produce a large number of clusters, if the distribution in figure 6 contains many noisy sub-trajectories. This leads to multiple Gaussians and unnecessarily increases the number of estimated clusters k. Thus, manually choosing the number of clusters in real applications might be reasonable.

The memory storage and computational resources of BI-ADD are mostly consumed by the estimation of $\hat{\alpha}$ due to the size of model$_\alpha$. The approximate computation time of BI-ADD mainly depends on the length of trajectories. The elapsed time for the scenario (A) in figure 8 is available in figure S5, the overall processing time of 1000 trajectories is 522.9 seconds on a single machine equipped with RTX 3090, 24 GB of VRAM. We trained the model$_\alpha$ and model$_K$ on a A100 chip with 1 M simulated noiseless fBm 2-dimensional trajectories. BI-ADD infers the changepoint detection sequentially due to the merging mechanism. The sequential nature of the approach hinders parallelization of the changepoints approximation. A distributed detection of the changepoints for the molecular trajectories is an important research avenue.

Another research direction is to tackle challenging scenarios which arise in real data. The BI-ADD copes extremely well with the analysis of molecular trajectory in cases, where the diffusive properties between the states are significantly different. If the difference of diffusive properties of a trajectory is small in terms of both K and α, BI-ADD performs sub-optimally, since this setting is particularly difficult for all methods. More precisely, the distributions of $\hat{\alpha}$ and $\log\hat{K}$ (e.g. figure 6) can be easily clustered since the clusters are well-separated in our numerical experiments. Note that the clustering is done using the GMMs. In our contribution, we make use of the obtained clustering, however, we did not explore yet the transition probabilities between clusters. If the estimated transition probabilities between the states are time dependent, this change over time can be studied for the biological inter-molecular processes. Another open problem is an accurate estimation of $\hat{\alpha}$ from a short trajectory, where the length is smaller than 8. In a setting where a lot of sequences are short, it becomes hard to perform the estimation accurately due to the low number of observations. Since the accurate estimation of $\hat{\alpha}$ and $\hat{K}$ is nearly intractable for very short trajectories, the application BI-ADD for short trajectories in real data should be decided after analysis of the obtained clusters (such as the clusters in figure S1). This point is critical for the prediction of molecular trajectories in biology, since inferring long trajectories from real data is limited in empirical methods of optical physics and the data acquisition is, in general, expensive.

We are currently testing the proposed BI-ADD on the experimental data of the FIONA team, Sorbonne university.

Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: https://doi.org/10.5281/zenodo.13334951.

Software availability

The Python source code of BI-ADD is publicly available for research purposes. [30](https://github.com/JunwooParkSaribu/BI_ADD)

ORCID iDs

Junwoo Park 0009-0000-6126-2125
Clément Cabriel 0000-0002-0316-0312
Ignacio Izeddin 0000-0002-8476-3915
Judith Miné-Hattab 0000-0001-9986-4092

References

[1] Molz F J, Liu H H and Szulga J 1997 Fractional brownian motion and fractional gaussian noise in subsurface hydrology: a review, presentation of fundamental properties and extensions *Water Resour. Res.* **33** 2273–86

[2] Qian B and Rasheed K 2004 Hurst exponent and financial market predictability *Proc. 2nd IASTED Int. Conf. on Financial Engineering and Applications*

[3] Matos J A, Gama S M, Ruskin H J, Sharkasi A A and Crane M 2008 Time and scale Hurst exponent analysis for financial markets *Physica A* **387** 3910–5

[4] Hurst H E 1951 Long-term storage capacity of reservoirs *Trans. Am. Soc. Civil Eng.* **116** 770–99

[5] Gentili A and Volpe G 2021 Characterization of anomalous diffusion classical statistics powered by deep learning (CONDOR) *J. Phys. A: Math. Theor.* **54** 314003

[6] Muñoz-Gil G, i Corominas G G and Lewenstein M 2021 Unsupervised learning of anomalous diffusion data: an anomaly detection approach *J. Phys. A: Math. Theor.* **54** 504001

[7] Quiblier N, Rye J M, Leclerc P, Truong H, Hannou A, Héliot L and Berry H 2024 Enhancing fluorescence correlation spectroscopy with machine learning to infer anomalous molecular motion *Biophys. J.* **124** 844–56

[8] Verdier H, Duval M, Laurent F, Cassé A, Vestergaard C L and Masson J B 2021 Learning physical properties of anomalous random walks using graph neural networks *J. Phys. A: Math. Theor.* **54** 234001

[9] Betzig E, Patterson G H, Sougrat R, Lindwasser O W, Olenych S, Bonifacino J S, Davidson M W, Lippincott-Schwartz J and Hess H F 2006 Imaging intracellular fluorescent proteins at nanometer resolution *Science* **313** 1642–5

[10] Manley S, Gillette J M, Patterson G H, Shroff H, Hess H F, Betzig E and Lippincott-Schwartz J 2008 High-density mapping of single-molecule trajectories with photoactivated localization microscopy *Nat. Methods* **5** 155–7

[11] Mandelbrot B B and John W V N 1968 Fractional brownian motions, fractional noises and applications *SIAM Rev.* **10** 422–37

[12] Falconer K 2003 *Fractal Geometry Mathematical Foundations and Applications* 2 edn (Wiley)

[13] Lemons D S and Gythiel A 1997 Paul Langevin's 1908 paper "On the Theory of Brownian Motion" ["Sur la théorie du mouvement brownien," C. R. Acad. Sci. (Paris) 146, 530-533 (1908)] *Am. J. Phys.* **65** 1079–81

[14] Barkai E, Garini Y and Metzler R 2012 Strange kinetics of single molecules in living cells *Phys. Today* **65** 29–35

[15] Miné-Hattab J, Recamier V, Izeddin I, Rothstein R and Darzacq X 2017 Multi-scale tracking reveals scale-dependent chromatin dynamics after DNA damage *Mol. Biol. Cell* **28** 3323–32

[16] Marshall W, Straight A, Marko J, Swedlow J, Dernburg A, Belmont A, Murray A, Agard D and Sedat J 1997 Interphase chromosomes undergo constrained diffusional motion in living cells *Curr. Biol.* **7** 930–9

[17] 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

[18] Muñoz-Gil G, BorjaRequena F G F, Bachimanchi H, Pineda J and Manzo C 2023 AnDiChallenge/andi_datasets: AnDi challenge 2 *Zenodo* (https://doi.org/10.5281/zenodo.4775310)

[19] Muñoz-Gil G *et al* 2025 Quantitative evaluation of methods to analyze motion changes in single-particle experiments *Nat. Commun.* **16** 6749

[20] Muñoz-Gil G *et al* 2021 Objective comparison of methods to decode anomalous diffusion *Nat. Commun.* **12** 6253

[21] Muñoz-Gil G *et al* 2024 2nd anomalous diffusion challenge (available at: http://andi-challenge.org/challenge-2024/#andi2leaderboard)

[22] Park J, Sokolovska N, Cabriel C, Izeddin I and Miné-Hattab J 2024 FreeTrace *Zenodo* (https://doi.org/10.5281/zenodo.13336251)

[23] Hochreiter S and Schmidhuber J 1997 Long short-term memory *Neural Comput.* **9** 1735–80

[24] LeCun Y, Bengio Y and Hinton G 2015 Deep learning *Nature* **521** 436–44

[25] Shi X, Chen Z, Wang H, Yeung D Y, kin Wong W and chun Woo W 2015 Convolutional lstm network: a machine learning approach for precipitation nowcasting (arXiv:1506.04214)

[26] Goodfellow I, Bengio Y and Courville A 2016 *Deep Learning* (MIT Press) (available at: www.deeplearningbook.org)

[27] Pedregosa F *et al* 2011 Scikit-learn: machine learning in Python *J. Mach. Learn. Res.* **12** 2825–30 (available at: http://jmlr.org/papers/v12/pedregosa11a.html)

[28] Kingma D P and Ba J 2017 Adam: a method for stochastic optimization (arXiv:1412.6980)

[29] Schwarz G 1978 Estimating the dimension of a model *Ann. Stat.* **6** 461–4

[30] Park J, Sokolovska N, Cabriel C, Izeddin I and Miné-Hattab J 2024 Bi-add *Zenodo* (https://doi.org/10.5281/zenodo.13334951)