Amortised inference of fractional Brownian motion with linear computational complexity

Hippolyte Verdier, François Laurent, Christian Vestergaard, Alhassan Cassé, Jean-Baptiste Masson

Amortised inference of fractional Brownian motion with linear computational complexity

Hippolyte Verdier*
Decision and Bayesian Computation, USR 3756 (C3BI/DBC) & Neuroscience department CNRS UMR 3751, Institut Pasteur, Université de Paris, CNRS, Paris, France and Histopathology and Bio-Imaging Group, Sanofi, R&D, Vitry-Sur-Seine, France

François laurent, Christian L. Vestergaard, and Jean-Baptiste Masson†
Decision and Bayesian Computation, USR 3756 (C3BI/DBC) & Neuroscience department CNRS UMR 3751, Institut Pasteur, Université de Paris, CNRS, Paris, France

Alhassan Cassé
Histopathology and Bio-Imaging Group, Sanofi, R&D, Vitry-Sur-Seine, France

We introduce a simulation-based, amortised Bayesian inference scheme to infer the parameters of random walks. Our approach learns the posterior distribution of the walks' parameters with a likelihood-free method. In the first step a graph neural network is trained on simulated data to learn optimized low-dimensional summary statistics of the random walk. In the second step an invertible neural network generates the posterior distribution of the parameters from the learnt summary statistics using variational inference. We apply our method to infer the parameters of the fractional Brownian motion model from single trajectories. The computational complexity of the amortized inference procedure scales linearly with trajectory length, and its precision scales similarly to the Cramér-Rao bound over a wide range of lengths. The approach is robust to positional noise, and generalizes well to trajectories longer than those seen during training. Finally, we adapt this scheme to show that a finite decorrelation time in the environment can furthermore be inferred from individual trajectories.

I. INTRODUCTION

Fractional Brownian motion (fBm) [1, 2] is a paradigmatic model of anomalous transport. It is a non-Markovian Gaussian process characterized by stationary increments and long temporal correlations in the noise driving the process. It allows capturing long-range temporal correlations in the dynamics of a walker or its environment, and it is a model of choice to describe a multitude of dynamic processes in numerous scientific fields [3–21]. Following the classification given in [22] of the three main sources of anomalous diffusion, the anomalous dynamics of fBm stems from the statistical dependency of the displacements at all time scales. Since fBm is a Gaussian process, it admits an analytical expression of the joint likelihood of the recorded signal. It is thus an ideal model to investigate the performance of approximate schemes to infer anomalous diffusion, such as variational inference or machine-learning-based approaches, since it allows direct comparison to statistically optimal exact inference.

The position of a random walker undergoing fBm is described by a Langevin equation [1] of the form

$$\frac{dr\left(t\right)}{dt} = \sqrt{K_\alpha}\,\eta\left(t\right), \tag{1}$$

where η is a zero-mean Gaussian noise process with covariance $\langle \eta\left(t_1\right)\eta\left(t_2\right)\rangle = \alpha\left(\alpha-1\right)\left|t_1-t_2\right|^{\alpha-2}$ and K_α is a generalized diffusion constant that sets the scale of the process. fBm is self-similar and ergodic [23, 24]. However, it has been shown to exhibit transient non-ergodic behaviour when confined [24, 25] and it is worth noting that the ergodic regime is witnessed only after a long transient passage exhibiting non-ergodic properties [24, 26–28]. The noise η is negatively correlated in the subdiffusion regime ($\alpha < 1$), while it is positively correlated in the super-diffusion regime ($\alpha > 1$).

Methods for estimating a random walk's parameters can roughly be divided into two types: heuristic approaches using features extracted from the trajectories [17, 29–32], and likelihood-based (e.g., Bayesian) approaches [33–36]. Each has its strengths and weaknesses. Likelihood-based approaches are provably asymptotically optimal, but they are often computationally intensive and are only applicable to random walk models that have a tractable likelihood. Feature-based approaches are typically computationally cheaper, and they can be applied to a much larger range of models since they do not require a tractable likelihood. However, they are generally not statistically efficient, are prone to bias when used on experimental data and their precision can be difficult to evaluate. It is worth noting the rapid progress of machine learning based approaches [32, 37], which fall in the category of feature-based approaches, and which allow to learn high quality features to perform both parameter estimation and model classification. While such machine learning approaches generally outperform hand-

* hverdier@pasteur.fr
† jbmasson@pasteur.fr

crafted features on numerically generated data, it remains difficult to evaluate their actual performance and robustness on empirical data.

Here, we develop an amortized Bayesian inference approach to estimate the parameters of a fBm from a single recorded trajectory. More precisely, this paper focuses on two tasks: (i) amortizing the inference of the anomalous exponent to reduce the computational cost of inference and test how much information about temporal correlation can be inferred by a computational scheme of linear complexity, and (ii) exploring the possibility of retrieving information about finite decorrelation times of the walker's dynamics. We use a graph neural network (GNN) to encode a set of summary features of the trajectory. The GNN is trained on simulated trajectories , and allows capturing long-range interactions while retaining a linear scaling of the computational complexity with the length of the trajectories. We train an invertible network to generate the posterior distribution from the summary features using a variational objective. Focusing on the fBm model allows us to compare the performance of the amortized approach to maximum likelihood estimation and to the Cramér-Rao bound which provides a lower bound on the variance of any unbiased estimator. We show that our amortized inference attains near-optimal performance as compared to exact likelihood-based inference and to the Cramér-Rao bound. We furthermore discuss the latent space structure learned by the summary network and its ability to encode physical properties. We test the applicability of the approach to trajectories corrupted by positional noise and its potential to generalize to trajectories that are longer than those seen during training. Finally, we extend the inference procedure to capture a finite decorrelation time in the dynamics which may typically arise in physical environments.

II. AMORTISED BAYESIAN INFERENCE

In the context of parameter estimation, Bayesian inference uses Bayes' theorem to compute the posterior probability distribution of the parameters $\boldsymbol{\theta}$ given recorded data $\mathbf{R}$ (here a trajectory) and a probabilistic model of these data,

$$p\left(\boldsymbol{\theta}\,|\mathbf{R}\right) = \frac{p\left(\mathbf{R}|\,\boldsymbol{\theta}\right)p\left(\boldsymbol{\theta}\right)}{p\left(\mathbf{R}\right)}. \tag{2}$$

Equation (2) relates the posterior distribution, $p\left(\boldsymbol{\theta}\,|\mathbf{R}\right)$ to the likelihood $p\left(\mathbf{R}|\,\boldsymbol{\theta}\right)$, the prior $p\left(\boldsymbol{\theta}\right)$ and the evidence $p\left(\mathbf{R}\right)$. Here, we only consider one single model, i.e., the fBm, and thus do not explicitly refer to it. The principle of amortized inference [39] is to split the estimation of the posterior $p\left(\boldsymbol{\theta}\,|\mathbf{R}\right)$ into two independent steps. The first is computationally costly and involves learning an approximate posterior density $\hat{p}\left(\boldsymbol{\theta}\,|\mathbf{R}\right)$ from numerically generated data. Then, the second step consists in running the pre-trained approximate system on the experimental data to infer the posterior density, assuming that they are similar to the training data.

A tractable likelihood can be computed for fBm. We consider a trajectory $\mathbf{R} = (r_0, r_2, \ldots, r_N)$ to be a 1-dimensional time-series of positions r_i recorded at equidistant points in time $t_i \in \{0, \Delta t, 2\Delta t, \ldots, N\Delta t\}$. The likelihood of a trajectory reads

$$p\left(\mathbf{R}|\,\boldsymbol{\theta}\right) = \frac{1}{\left(2\pi\right)^{N/2}\sqrt{\det\Sigma(\boldsymbol{\theta})}}\exp\left(-\frac{1}{2}\left(\Delta\mathbf{r}\right)^{\top}\Sigma(\boldsymbol{\theta})^{-1}\Delta\mathbf{r}\right), \tag{3}$$

where $\Delta r = \left(\Delta r_1, .., \Delta r_N\right)^{\top}$, with $\Delta r_i = r_i - r_{i-1}$ the individual displacements. Then, $\boldsymbol{\theta} = (K_\alpha, \alpha)$ are the fBm's parameters to infer, and Σ is the displacements' covariance matrix whose coefficients are given by

$$\left[\Sigma(\boldsymbol{\theta})\right]_{ij} = K_\alpha \Delta t^\alpha \left(|i-j+1|^\alpha + |i-j-1|^\alpha - 2|i-j|^\alpha\right). \tag{4}$$

We choose to rely on a likelihood-free approach to amortize our inference procedure. This may seem a counter-intuitive choice for the precise case of fBm because the likelihood is analytically tractable, but this method has the advantage of relying solely on computations of linear complexity. Furthermore, the approach is also directly portable to more complex problems for which a tractable likelihood may not be available or may be too computationally costly. Indeed, likelihood-free inference is a method of choice to address such problems. As more and more complex models are encountered in numerous fields of science, the field of simulation-based inference [39] is growing very rapidly to address the associated challenging inverse problems. The shift towards amortization of the likelihood is notably driven by new tools and conceptual approaches derived from machine learning [40, 41].

The architecture of the amortized model of the posterior distribution is shown in Figure 1. It is based on the recently introduced Bayes Flow (BF) [42] procedure. In this framework, a first neural network, working as an encoder, creates a fixed-dimension vector of summary statistics from a set of observations. In our case, the encoder takes the form of a GNN (Fig. 1A). The encoder's output, the summary statistics vector, parametrizes an invertible transformation between easily sampled distributions (Gaussian) and the posterior distribution of the parameters (Fig. 1B). The full procedure generates the posterior distribution of the parameters. Such flow-based approaches, derived from normalizing flows [43], have the advantage that they provide an estimation of the posterior without requiring extensive sampling. The whole module is trained on numerically generated data and can then be used for inference. In the two following subsections, we first present the GNN (Section II A) and then the invertible network (Section II B).

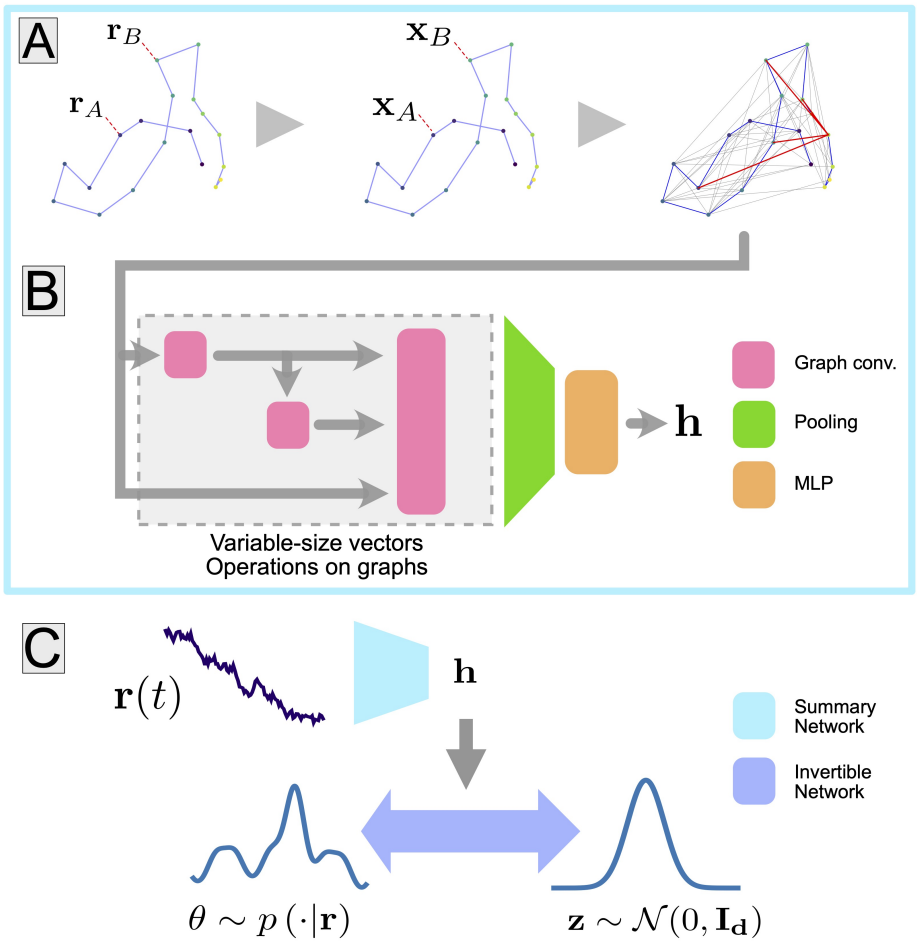

FIG. 1. **Model Architecture.** A: Construction of a graph from a single trajectory, on which graph learning is performed by the summary network shown in B, see [38] for details. B: Summary network consisting of graph convolution layers, a pooling layer and a multi-layer perceptron. The vector of statistics is indicated by $\mathbf{h}$. C: General structure of the model, with the summary network parametrizing the invertible network. In training mode, the invertible network is used from left to right, and in inference mode it is used from right to left.

A. Graph neural network for learning summary statistics

GNNs have been introduced to model and analyse graphs, meshes and point clouds [44–46]. They are well suited to capture geometric properties from point clouds and other datasets of variable size [46, 47], they can keep a sparse architecture while encoding long temporal correlations [48, 49] and they exhibit good performance with a limited number of parameters compared to other modern architectures. For these reasons, they are well adapted to analysing random walks [38], and we adopt a GNN architecture for our summary network.

As indicated by their name, GNNs process graphs, and the first step of our inference pipeline is thus to build a graph from a trajectory. To do so, we represent each trajectory $\mathbf{R} = (r_0, r_1, r_2, \ldots, r_N)$ by a directed graph $G = (V, E, \mathbf{X}, \mathbf{Y})$. Here $V = \{1, 2, \ldots, N\}$ is the set of nodes, each corresponding to a recorded position of the observed walker. $E \subseteq \{(i,j)|(i,j) \in V^2\}$ is the set of edges connecting pairs of nodes. Each node has a vector of features, $\mathbf{x}_i^{(0)}$ (of size n_x) which initially encapsulates information linked to the i-th position of the trajectory and to the displacement that led to it. $\mathbf{X}^{(0)} = (\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}, \ldots, \mathbf{x}_N^{(0)})$ is the (N, n_x) matrix of initial node feature vectors. Amongst features associated to node i are the normalized time i/N, as well as the total distance covered by the walker, its mean square displacement, and the maximal jump size, all measured since the beginning of the trajectory. Finally, $\mathbf{Y} = (\mathbf{y}_1^{(0)}, \mathbf{y}_2^{(0)}, \ldots, \mathbf{y}_{|E|}^{(0)})$ is a matrix of edge features, $\mathbf{y}_e^{(0)}$, each associated to an edge e in E. The features vector of a given edge e, $\mathbf{y}_e^{(0)}$, of size n_y, encapsulates information about the trajectory's course between the two nodes i and j it connects, such as the normalized time difference $(j-i)/N$ and various dis-

tance measures (see Supplemental Material for details). All distance measures are provided with multiple normalization factors (see Supplementary Material). The edges in G are placed such that incoming edges of each node originate only from nodes in the past (i.e., respecting causality): node i receives connections from nodes $i-\Delta_1,\ldots,i-\Delta_{max}$, where $(\Delta_i)_{i\geq 1}$ is a geometric series. The training is specific to the dimension of the trajectories. Yet, the GNN architecture can be adapted to trajectories of any dimension by adapting the features' initialization. A key point about the graph construction procedure is that it has linear algorithmic complexity.

Following the graph initialization step, the summary network performs several graph convolution operations [50–52]. It then passes the learnt node feature vectors as inputs to a pooling layer that aggregates features across all nodes of a trajectory graph into a fixed-length vector. The vector is finally passed through a multi-layer perceptron to obtain the summary statistics vector $\mathbf{h} = g_\psi(\mathbf{R})$, where ψ denotes the neural network coefficients. We refer the interested reader to [38] for details about the graph generation and GNN implementation.

B. Invertible network for generating a variational posterior density

The Bayes Flow approach provides an invertible transformation, $f_\phi(\cdot;\mathbf{h})$, between the parameter space (in $\mathbb{R}^D$, with $D \geq 2$) and the prior space (in $\mathbb{R}^D$), on which a D-dimensional standard Gaussian density is assumed. The transformation $f_\phi(\cdot;\mathbf{h})$ is parametrized by a conditional invertible neural network (cINN) [53] made of a succession of affine coupling blocks [54] (multiple blocks sequentially applied) and maps $\boldsymbol{\theta}$ to the prior conditioned on $\mathbf{h}$, the summary statistics of the trajectory.

By design, these blocks can be inverted and the determinant of the Jacobian matrix $\mathbf{J}_{f_\phi}$ of the transformation is retrieved from the forward pass. During training we seek to approximate the true posterior $p\left(\theta|\mathbf{R}\right)$ by the learnt posterior $p_\phi\left(\theta|\mathbf{R}\right) = \exp\left(-\frac{\|f_\phi(\boldsymbol{\theta};\mathbf{h})\|_2^2}{2}\right)$. The loss function is the Kullback-Leibler divergence between $p\left(\theta|\mathbf{R}\right)$ and $p_\phi\left(\theta|\mathbf{R}\right)$ which reads as

$$\mathcal{L}(\mathbf{R}) = \frac{1}{2}\|f_\phi\left(\boldsymbol{\theta};\mathbf{h}\right)\|_2^2 - \log|\det \mathbf{J}_{f_\phi}|, \tag{5}$$

where $\mathbf{h} = g_\psi(\mathbf{R})$. Sampling the posterior distribution consists in computing $\mathbf{h}$ from the trajectory $\mathbf{R}$, and generating the required number of sample as $\boldsymbol{\theta} = f_\phi^{-1}(\mathbf{z};\mathbf{h})$ with $\mathbf{z}$ generated from a standard D-dimensional Gaussian distribution.

III. ESTIMATION OF THE ANOMALOUS EXPONENT

We evaluate the performance of our amortized inference procedure on numerically generated trajectories. Estimating the anomalous exponent α is the most challenging part of the inference, and we thus focus on this here, but our approach infers a joint posterior density for $\boldsymbol{\theta} = (K_\alpha, \alpha)$. Figure 2A shows the inferred posteriors of α on portions of increasing length of two example trajectories. The amortized posterior is consistent with the exact one (See Supplementary). Both become increasingly peaked around the true value of α as the length of the trajectory increases. The inferred posterior distributions do not exhibit broad tails or divergences, and are thus proper distributions, i.e., they are normalizable.

We show the precision of the inference on trajectories with lengths varying across two orders of magnitudes. Both the variance of the approximated posterior and the square error of the estimator follow a power-law decrease, as can be seen Fig. 2B. Using the exact likelihood shown in Eq. 3 we can evaluate the Cramér-Rao bound and compare the amortised inference's performance to it. The amortised inference is suboptimal (as expected from a variational inference), but its variance shows a fast decreasing trend similar to the Cramér-Rao bound, i.e., close to $\propto 1/N$.

We looked at the learnt summary statistics h, that after training constitutes a low-dimensional representation of the random walks which is use as features to compute the posterior distribution. The latent representation can be interpreted for its own sake as the way the encoder represents information about the trajectories. An assumption in representation learning [55] is that interpretable representation lead to better generalisation. We projected h onto a 2D plane using UMAP [56] (a non-linear dimenstionality reduction algorithm) and mapped α on it (see Supplementary Fig. S1A). We see that the latent space is organised according the value of α, a good indication that the learning process properly captured the underlying physical properties. We tested the robustness of the inference procedure when applied to trajectories corrupted by positional noise. We show in Supplementary Figure S2 the evolution of the mean square error of the amortised inference of α and compare it with the corresponding Cramér-Rao bound. The precision of the amortised inference procedure closely follows the lower bound set by the Cramér-Rao inequality. This was obtained by training models specifically on trajectories corrupted with increasing amounts of noise.

The summary network's architecture, with normalized initial features, leads to an approximately "length-invariant" inference, i.e., the vector of summary statistics captures relevant information regardless the length of the trajectories. Hence, the approach is not limited to trajectories seen during training. We show in Figure 2C an example of application for trajectories a hundred times longer than the maximal ones the inference procedure

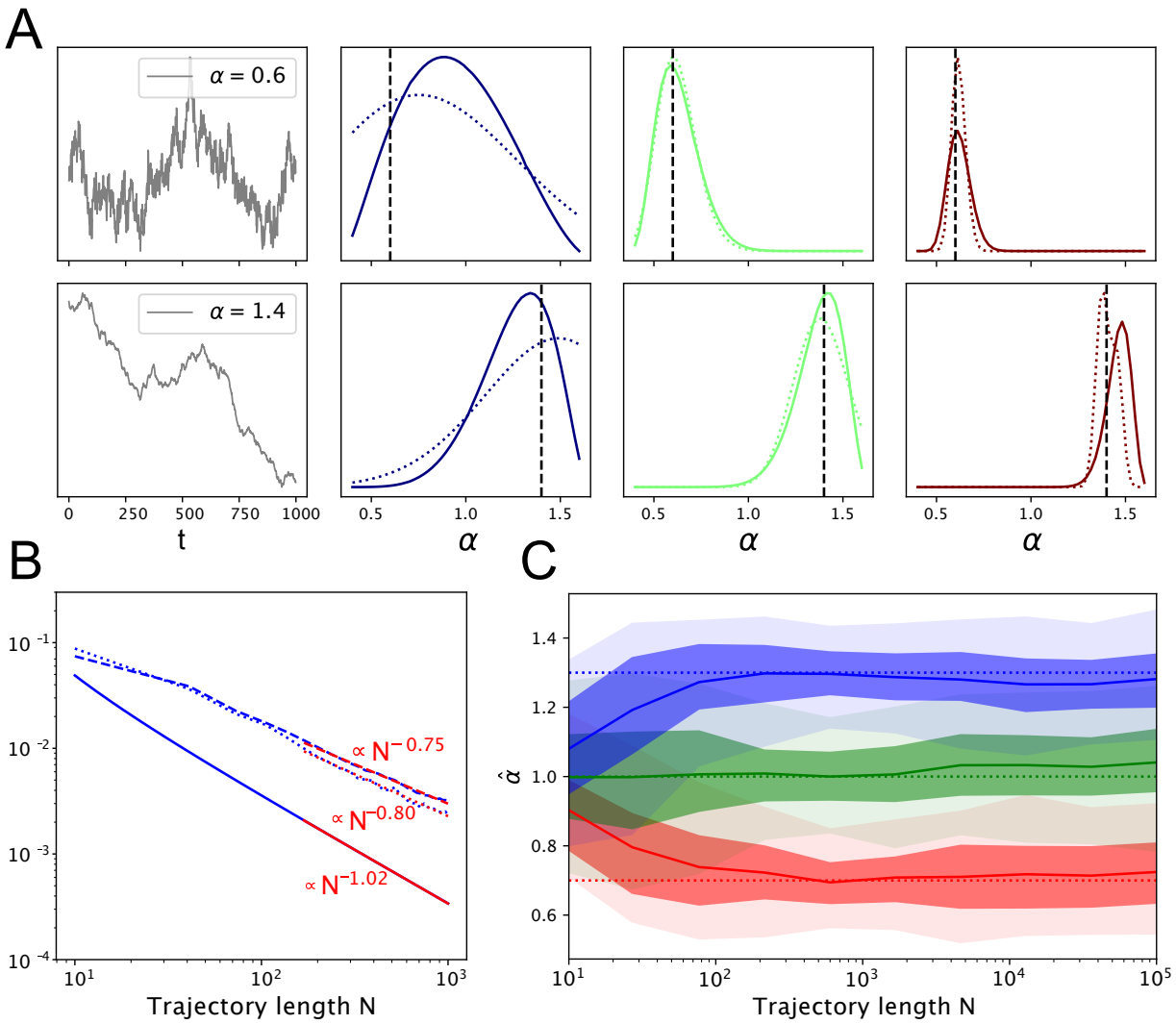

FIG. 2. **Model performance.** A: Evolution of the posterior density of α inferred by the model (plain lines) versus the true posterior (dotted lines) from two example trajectories with $\alpha = 0.6$ (top) and $\alpha = 1.4$ (bottom), respectively. The length of the portion of the trajectory used for inference is increased ten-fold between each panel, i.e., from left to right: $N = 10$ (blue), 100 (green), 1000 (red). B: Evolution of the variance of the estimator $\hat{\alpha}$ (dashed line), the mean square error of the mean posterior $\hat{\alpha}$ (dotted line) and the Cramér-Rao bound for an unbiased estimator of α, for increasing values of trajectory length N. In red: power-law fits on large values of N C: Convergence towards the true value of α as function of trajectory length. The model was trained on trajectories of length $10 \leq N \leq 1,000$. Darker zones correspond to the first and third quartiles, while lighter ones correspond to 5% and 95% quantiles We modified the normalization procedure so that it is able to generalize to trajectories longer than those seen during training.

was trained on. The inferred $\hat{\alpha}$ for long trajectories were still well ordered, but suffered from a slight bias, which we corrected using a simple polynomial of degree 3.

An important attribute of the amortized approach is that is has linear computational complexity at inference time, i.e., when applied to infer the parameters of a random walk. To show this, we subdivide the amortised inference procedure into three steps: (i) initial feature evaluation, (ii) forward pass through graph convolutions and pooling, and (iii) operations on summary statistics to generate the posterior. (i) The initial evaluation of node and edge features requires $O(N+|E|)$ time and memory, where N is the number of nodes (for a trajectory of $N+1$ points) and $|E|$ is the number of edges. Here $|E| \propto N$ by design (the in-degree of nodes is bounded), so this step has $O(N)$ complexity. (ii) The forward pass through the graph convolutions and the following pooling of node features requires $O(|E|)$ operations and memory slots, and hence this step also has $O(N)$ complexity. (iii) The latent space is of fixed dimensions, and hence all operations after the pooling layer have $O(1)$ complexity. The global complexity of the amortised architecture is thus linear

with respect to the number of points in the trajectory.

In comparison, calculating the exact likelihood [Eq. (3)] requires evaluating the determinant $\det\Sigma(\boldsymbol{\theta})$ and the quadratic form $(\Delta\mathbf{r})^\top\Sigma(\boldsymbol{\theta})^{-1}\Delta\mathbf{r}$, which can be done in $O(N^2)$ time [57]. This makes exact inference prohibitively expensive for very long trajectories, where our amortized inference scheme may instead be used (Fig. 2C). Note furthermore that for many models the exact likelihood cannot be calculated at all, in which case approximate inference is the only route possible. In all of the above cases, our amortized inference scheme retains its linear computational complexity.

IV. ESTIMATION OF A FINITE DECORRELATION TIME

When considering fBm as a model of biomolecule random walks, we have to keep in mind that many physical environments might exhibit a finite decorrelation time τ_c possibly stemming from motion occurring outside a polymer-dominated environment [58] or from changes of conformations of the bimolecule altering the nature of its interactions The characteristic time bears information on the local environment's physical properties, and it might be spatially dependent or specific to interactions with local partners. In practice, inferring τ_c from individual trajectories is challenging. Autocorrelation-based approaches for example give incomplete results on individual trajectories as the limited number of points prevents proper averaging [59, 60].

We adapted our amortised inference procedure to infer (α, τ_c) instead of (K_α, α). We left out K_α here since it is simply a scale factor and can be removed by rescaling the trajectories. We used the same node and edge features as above, and we thus conserve the procedure's linear computational complexity. A finite decorrelation time was modeled by multiplying the autocovariance of the fBm by an exponential factor, $\min(1, e^{\tau_c-\tau})$, where τ is the time difference. Examples of the autocorrelation function for several values of α and τ_c are given in Supplementary Figure S3. The modified covariance matrix thus reads

$$[\Sigma(\alpha,\tau_c)]_{i,j} = \min\left(1, e^{\tau_c - |i-j|\Delta t}\right) \times \left(|i-j+1|^\alpha + |i-j-1|^\alpha - 2|i-j|^\alpha\right),$$

where we have ignored the scale factor $K_\alpha \Delta t^\alpha$.

There is no simple means to relate the length of a trajectory to the difficulty of inferring its finite correlation time, so we performed this inference solely on trajectories of length 1,000 with τ_c integer-valued and ranging from 5 to 50.

We compared our estimator with the maximum likelihood estimator, obtained by choosing the value of (α,τ_c) that maximizes the likelihood of the observed trajectory.

To optimize the likelihood in practice, we computed the log-likelihood on a grid of values, of α and τ_c, with α taking 30 regularly spaced values between 0.4 and 1.6, and τ_c taking all possible values in its range. As shown in Figure 3B (upper panel), our amortised inference yields a slightly more biased estimate than the maximum-likelihood estimator (when taking the mean of the posterior distribution) but has a smaller variance. When $\alpha = 1$, successive increments are completely independent of each other and there is thus no information to retrieve regarding τ_c. This is observable on the lower panel of Figure 3, both by looking at the Cramér-Rao bound, which diverges, and at the variance of our estimator, which is maximal at $\alpha = 1$.

V. DISCUSSION

Simulation-based inference coupled with machine learning are a promising avenue to address challenging inverse problems. When applied to intractable systems, this combination allows splitting the inference task into two steps. In the first, computationally intensive simulations produce artificial data. These data are used to train neural networks to approximate the posterior distribution of the parameters using a variational objective. In the second step, which is computationally fast, inference is performed on experimental data and the posterior distribution is evaluated by a direct forward pass through the trained neural networks. The procedure is statistically efficient if the numerical data match the properties of experimental one and if the variational inference is able to capture the complex relations that might exist between the variables to be inferred.

There are two main challenges associated with amortised approaches. First, training variational inference often consists in minimising a Kullback-Liebler distance between the approximate distribution and the real (unknown) one [61]. Optimising such a non-convex function is challenging and is not generally guaranteed to converge towards a global optimum. The second challenge is linked to interpretability. Both the models used to learn the summary statistics and the variational posterior distribution are generally intractable. There is thus limited insurance that the process does not misbehave, especially when applied to real experimental data. Evaluation of the exact posterior distribution using sampling, such as in approximate Bayesian computation, may however lead to similar problems due to the difficulty of properly sampling complex likelihood landscapes.

We here used fBm to quantify the performance of our amortised inference approach. We chose to focus on fBm both due to its paradigmatic status as an anomalous random walk model and because it has a tractable likelihood, allowing us to compare our amortised method to exact likelihood-based inference and to the Cramér-Rao bound on estimator precision. We advocate more generally for the use of exactly solvable random walk models, such as the fBm, as benchmarks to evaluate the performance of machine-learning based inference methods.

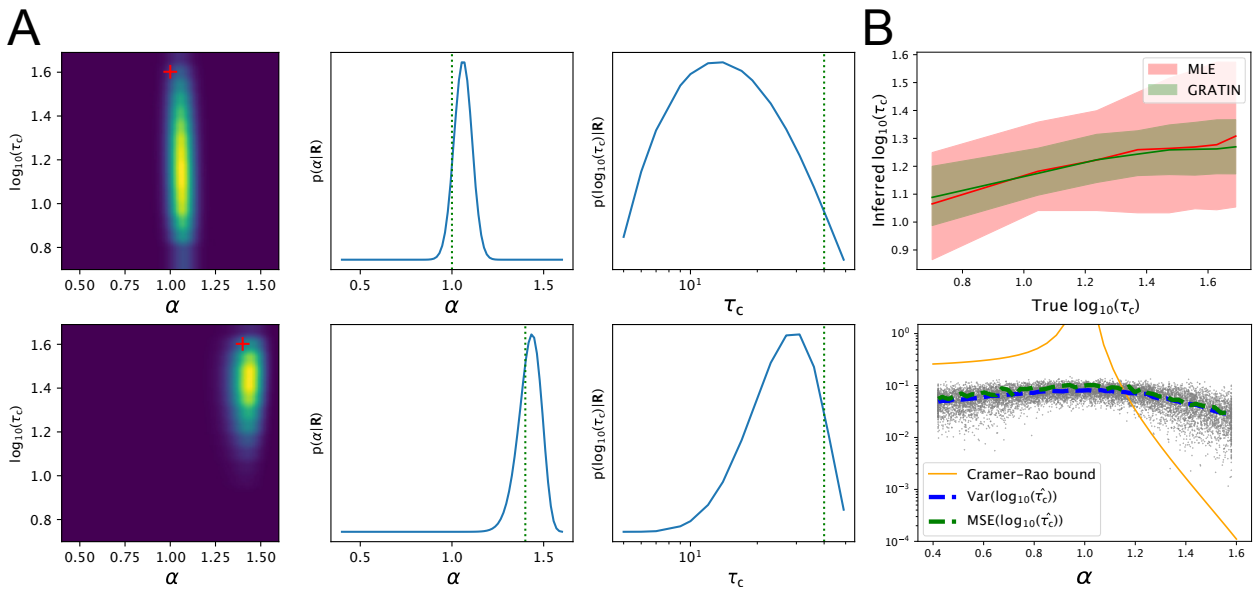

FIG. 3. **Performance of the anomalous exponent and decorrelation time estimation** A: Posteriors of α and $\log_{10}(\tau_c)$ for two trajectories with different α (plain lines). Dashed green lines indicate true parameter values. B: Top: Comparison of the values of $\log(\tau_c)$ inferred by our method (in green) and by a maximum-likelihood estimator (in red). The thick line represents the mean across all trajectories, while the filled regions correspond to the first and last quartiles. Bottom: Variance and mean square error of our inference of $\log_{10}(\tau_c)$ plotted as a function of α, compared with the Cramér-Rao bound for an unbiased estimator.

We showed that our amortized inference can successfully be applied to infer the parameters of fBm, with a precision that is lower than the Cramér-Rao bound but which increases with a scaling that is similar to it. Our algorithm has a linear complexity in the length of the trajectory and can be applied to trajectories of any length at inference time, even if the algorithm has not been specifically trained on trajectories of the same length. We furthermore showed that our amortised approach could be used to efficiently infer the parameters of a more realistic fBm-type model with a finite decorrelation time.

Our amortised inference framework can be used for any random walk model, even for models that do not have a tractable likelihood, provided that they can be simulated efficiently enough to provide a large number of trajectories for training. In all cases, our approach retains its linear computational complexity at inference time. For random walk models with intractable likelihoods, only empirical evaluation of the performance will in general be possible. Thus, it is not possible to make absolute statements about the statistical efficiency of the approach in these cases.

Beyond random walks, amortized inference can more generally be instrumental in providing posterior distributions for models of complex systems with fractional noise and/or long memory. Numerous challenges have to be addressed to standardise the optimisation of the variational inference, especially in cases where some parameters are not sufficiently constrained by data or when there are sloppy directions in the parameter space [62]. Furthermore, variational inference does not necessarily lead to physically realistic parameters. Ensuring the physics-informed [63, 64] nature of the inference may require imposing constraints on the network generating the summary statistics. Though our results show that the network is able to learn physically meaningful features without inductive bias. Finally, the statistical efficiency of amortised approaches will depend on the ability of numerically generated data to match experimental observations.

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

The funding sources had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Conflicts of interest. Hippolyte Verdier and Alhassan Cassé are Sanofi employees and may hold shares and/or stock options in the company. The other author declare to have no financial or non-financial conflicts of interest.

[1] G. Crispin, *Handbook of Stochastic Methods: for Physics, Chemistry and natural sciences*, 4th ed. (Springer).

[2] B. B. Mandelbrot and J. W. V. Ness, SIAM Review **10**, 422 (1968).

[3] J.-P. Bouchaud and A. Georges, Physics Reports **195**, 127 (1990).

[4] W. Leland, M. Taqqu, W. Willinger, and D. Wilson, IEEE/ACM Transactions on Networking **2**, 1 (1994).

[5] N. J. Cutland, P. E. Kopp, and W. Willinger, in *Seminar on Stochastic Analysis, Random Fields and Applications* (Birkhäuser Basel, 1995) pp. 327–351.

[6] V. Kukla, J. Kornatowski, D. Demuth, I. Girnus, H. Pfeifer, L. V. C. Rees, S. Schunk, K. K. Unger, and J. Karger, Science **272**, 702 (1996).

[7] L. Decreusefond and A. S. Üstünel, ESAIM: Proceedings **5**, 75 (1998).

[8] J. Bouchaud, *Theory of financial risk and derivative pricing : from statistical physics to risk management* (Cambridge University Press, Cambridge, 2003).

[9] S. C. Weber, A. J. Spakowitz, and J. A. Theriot, Physical Review Letters **104**, 10.1103/physrevlett.104.238102 (2010).

[10] J. L. A. Dubbeldam, V. G. Rostiashvili, A. Milchev, and T. A. Vilgis, Physical Review E **83**, 10.1103/physreve.83.011802 (2011).

[11] J.-H. Jeon, V. Tejedor, S. Burov, E. Barkai, C. Selhuber-Unkel, K. Berg-Sørensen, L. Oddershede, and R. Metzler, Physical Review Letters **106**, 10.1103/physrevlett.106.048103 (2011).

[12] J.-C. Walter, A. Ferrantini, E. Carlon, and C. Vanderzande, Physical Review E **85**, 10.1103/physreve.85.031120 (2012).

[13] D. Ernst, M. Hellmann, J. Köhler, and M. Weiss, Soft Matter **8**, 4886 (2012).

[14] S. Rostek and R. Schöbel, Economic Modelling **30**, 30 (2013).

[15] V. V. Palyulin, T. Ala-Nissila, and R. Metzler, Soft Matter **10**, 9016 (2014).

[16] A. Javer, N. J. Kuwada, Z. Long, V. G. Benza, K. D. Dorfman, P. A. Wiggins, P. Cicuta, and M. C. Lagomarsino, Nature Communications **5**, 10.1038/ncomms4854 (2014).

[17] D. Han, N. Korabel, R. Chen, M. Johnston, A. Gavrilova, V. J. Allan, S. Fedotov, and T. A. Waigh, eLife **9**, 10.7554/elife.52224 (2020).

[18] W. Wang, A. G. Cherstvy, A. V. Chechkin, S. Thapa, F. Seno, X. Liu, and R. Metzler, Journal of Physics A: Mathematical and Theoretical **53**, 474001 (2020).

[19] M. Gherardi, L. Calabrese, M. Tamm, and M. C. Lagomarsino, Physical Review E **96**, 10.1103/physreve.96.042402 (2017).

[20] M. Arutkin, B. Walter, and K. J. Wiese, Physical Review E **102**, 10.1103/physreve.102.022102 (2020).

[21] S. Yu, J. Wu, X. Meng, R. Chu, X. Li, and G. Wu, Entropy **23**, 542 (2021).

[22] R. Metzler, J.-H. Jeon, A. G. Cherstvy, and E. Barkai, Phys. Chem. Chem. Phys. **16**, 24128 (2014).

[23] W. Deng and E. Barkai, Physical Review E **79**, 10.1103/physreve.79.011112 (2009).

[24] S. Burov, J.-H. Jeon, R. Metzler, and E. Barkai, Physical Chemistry Chemical Physics **13**, 1800 (2011).

[25] J.-H. Jeon and R. Metzler, Physical Review E **85**, 10.1103/physreve.85.021147 (2012).

[26] E. Geneston, R. Tuladhar, M. T. Beig, M. Bologna, and P. Grigolini, Physical Review E **94**, 10.1103/physreve.94.012136 (2016).

[27] H. Loch-Olszewska, G. Sikora, J. Janczura, and A. Weron, Physical Review E **94**, 10.1103/physreve.94.052136 (2016).

[28] J. Kursawe, J. Schulz, and R. Metzler, Physical Review E **88**, 10.1103/physreve.88.062124 (2013).

[29] Y. Meroz and I. M. Sokolov, Physics Reports **573**, 1 (2015).

[30] T. Kosztołowicz, K. Dworecki, and S. Mrówczyński, Physical Review Letters **94**, 10.1103/physrevlett.94.170602 (2005).

[31] L. P. Sanders and T. Ambjörnsson, The Journal of Chemical Physics **136**, 175103 (2012).

[32] G. Muñoz-Gil, G. Volpe, M. A. Garcia-March, R. Metzler, M. Lewenstein, and C. Manzo, Emerging Topics in Artificial Intelligence 2020 10.1117/12.2567914 (2020).

[33] M. Lysy, N. S. Pillai, D. B. Hill, M. G. Forest, J. W. R. Mellnik, P. A. Vasquez, and S. A. McKinley, Journal of the American Statistical Association **111**, 1413 (2016).

[34] J. Krog, L. H. Jacobsen, F. W. Lund, D. Wustner, and M. A. Lomholt, Journal of Statistical Mechanics: Theory and Experiment **2018**, 10.1088/1742-5468/aadb0e (2018).

[35] P. K. Koo and S. G. J. Mochrie, Physical Review E **94**, 10.1103/physreve.94.052412 (2016).

[36] S. Thapa, M. A. Lomholt, J. Krog, A. G. Cherstvy, and R. Metzler, Physical Chemistry Chemical Physics **20**, 29018 (2018).

[37] G. Muñoz-Gil, G. Volpe, M. A. Garcia-March, E. Aghion, A. Argun, C. B. Hong, T. Bland, S. Bo, J. A. Conejero, N. Firbas, *et al.*, Nature communications **12**, 1 (2021).

[38] H. Verdier, M. Duval, F. Laurent, A. Casse, C. L. Vestergaard, and J.-B. Masson, Journal of Physics A: Mathematical and Theoretical 10.1088/1751-8121/abfa45 (2021).

[39] K. Cranmer, J. Brehmer, and G. Louppe, Proceedings of the National Academy of Sciences **117**, 30055 (2020).

[40] G. Papamakarios, D. Sterratt, and I. Murray, in *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, Proceedings of Machine Learning Research, Vol. 89, edited by K. Chaudhuri and M. Sugiyama (PMLR, 2019) pp. 837–848.

[41] J. Alsing, T. Charnock, S. Feeney, and B. Wandelt, Monthly Notices of the Royal Astronomical Society 10.1093/mnras/stz1960 (2019).

[42] S. T. Radev, U. K. Mertens, A. Voss, L. Ardizzone, and U. Kothe, IEEE Transactions on Neural Networks and Learning Systems , 1 (2020).

[43] I. Kobyzev, S. Prince, and M. Brubaker, IEEE Transactions on Pattern Analysis and Machine Intelligence , 1 (2020).

[44] M. Fey and J. E. Lenssen, Fast graph representation learning with pytorch geometric (2019), arXiv:1903.02428.

[45] T. N. Kipf and M. Welling, Semi-supervised classification with graph convolutional networks (2016), arXiv:1609.02907.

[46] C. R. Qi, L. Yi, H. Su, and L. Guibas, ArXiv **abs/1706.02413** (2017).
[47] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2017).
[48] T. Azevedo, A. Campbell, R. Romero-Garcia, L. Passamonti, R. A. Bethlehem, P. Liò, and N. Toschi 10.1101/2020.11.08.370288 (2020).
[49] Y.-J. Lu and C.-T. Li, Agstn: Learning attention-adjusted graph spatio-temporal networks for short-term urban sensor value forecasting (2021), arXiv:2101.12465.
[50] M. Defferrard, X. Bresson, and P. Vandergheynst, (2016), arXiv:1606.09375.
[51] R. Ying, J. You, C. Morris, X. Ren, W. L. Hamilton, and J. Leskovec, Hierarchical graph representation learning with differentiable pooling (2018), arXiv:1806.08804.
[52] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, Dynamic graph cnn for learning on point clouds (2018), arXiv:1801.07829.
[53] L. Ardizzone, J. Kruse, S. Wirkert, D. Rahner, E. W. Pellegrini, R. S. Klessen, L. Maier-Hein, C. Rother, and U. Köthe, 1808.04730.
[54] L. Dinh, J. Sohl-Dickstein, and S. Bengio, arXiv preprint arXiv:1605.08803 (2016).
[55] Y. Bengio, A. Courville, and P. Vincent, IEEE transactions on pattern analysis and machine intelligence **35**, 1798 (2013).
[56] L. McInnes, J. Healy, and J. Melville, arXiv preprint arXiv:1802.03426 (2018).
[57] J. F. Monahan, *Numerical methods of statistics* (Cambridge University Press, 2011).
[58] S. C. Weber, A. J. Spakowitz, and J. A. Theriot, Physical review letters **104**, 238102 (2010).
[59] J. F. Reverey, J.-H. Jeon, H. Bao, M. Leippe, R. Metzler, and C. Selhuber-Unkel, Scientific reports **5**, 1 (2015).
[60] G. Crispin, *Handbook of Stochastic Methods: for Physics, Chemistry and natural sciences*, 4th ed. (Springer).
[61] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)* (Springer).
[62] J. J. Waterfall, F. P. Casey, R. N. Gutenkunst, K. S. Brown, C. R. Myers, P. W. Brouwer, V. Elser, and J. P. Sethna, **97**, 150601.
[63] M. Raissi, P. Perdikaris, and G. E. Karniadakis, **378**, 686.
[64] G.-J. Both and R. Kusters, 2106.04886.
[65] T. N. Kipf and M. Welling, arXiv preprint arXiv:1609.02907 (2016).
[66] M. Simonovsky and N. Komodakis, in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017) pp. 3693–3702.

SUPPLEMENTARY MATERIAL

1. Amortized inference model architecture and training

a. Node and edge features

The features associated to each node $i \in \{1, \dots, N\}$ in the graph of a trajectory $(r_0, \dots, r_N)$ of length N are:

1. the normalized time: i/N;
2. the cumulative distance covered by the trajectory up to i: $\sum_{k\leq i} \|\Delta r_k\|_2$;
3. the cumulative squared distance covered by the trajectory up to i: $\sum_{k\leq i} \|\Delta r_k\|_2^2$;
4. the maximum step size up to i: $\max_{k\leq i} \|\Delta r_k\|_2$.

The features associated to an edge $e_{i,j}$ with $i < j$ are:

1. the normalized time difference: $(j-i)/N$;
2. the distance: $\|r_j - r_i\|_2$;
3. the dot product of jumps: $\Delta r_i^\top \Delta r_j$ (equal to $\Delta r_i \Delta r_j$ for 1D trajectories);
4. the distance covered by the trajectory between i and j: $\sum_{i<k\leq j} \|\Delta r_k\|_2 = \sum_{k\leq j} \|\Delta r_k\|_2 - \sum_{k\leq i} \|\Delta r_k\|_2$
5. sum of square step sizes between i and j : $\sum_{i<k\leq j} \|\Delta r_k\|_2^2 = \sum_{k\leq j} \|\Delta r_k\|_2^2 - \sum_{k\leq i} \|\Delta r_k\|_2^2$.

The computation of each features is done in linear or constant time. In particular, the last two features are calculated in linear complexity by leveraging the fact that they equal to the differences of two node features, which each solely depends on i or j. All features based on positions or steps are computed on normalized trajectories, with three normalization scales applied in parallel: (i) the standard deviation of the step sizes, (ii) the total covered distance and (iii) the standard deviation of the positions.

2. Neural network architectures and training

a. GNN Architecture

The architecture of the GNN used in the summary network is similar to the encoder network proposed in [38], with the difference that we here additionally apply edge features. Node and edge features are first passed to perceptrons, which embeds them in a 32-dimensional space. The network is then composed of three successive convolution layers (one taken from [65] and two edge-conditioned layers taken from [66]) outputting node features matrices $\mathbf{x}^{(1)}$, $\mathbf{x}^{(2)}$ and $\mathbf{x}^{(3)}$, each of 32 dimensions, which are summed to form $\mathbf{x}^{(f)}$. The rows of this matrix of nodes features are then averaged during the pooling step, to keep just one row per graph, i.e., per trajectory. This vector is subsequently passed to a three-layer perceptron, the output of which is the summary statistics vector.

b. Invertible network

The invertible network is a succession of three affine coupling blocks. These blocks, introduced in [54], transform an input vector $\mathbf{u}$ into $\mathbf{v}$ in an invertible manner parametrized by the summary statistics vector $\mathbf{h}$. They do so by splitting $\mathbf{u}$ into two halves $\mathbf{u}_1$ and $\mathbf{u}_2$, used to compute the two halves $\mathbf{v}_1$ and $\mathbf{v}_2$ of $\mathbf{v}$ by consecutively performing the two following operations :

$$\mathbf{v}_1 = \mathbf{u}_1 \odot \exp\left(s_1\left(\mathbf{u}_2; \mathbf{h}\right)\right) + t_1(\mathbf{u}_2; \mathbf{h})$$
$$\mathbf{v}_2 = \mathbf{u}_2 \odot \exp\left(s_2\left(\mathbf{v}_1; \mathbf{h}\right)\right) + t_2(\mathbf{v}_1; \mathbf{h})$$

where $\odot$ denotes the element-wise multiplication (the Hadamart product) and where s_1, s_2, t_1 and t_2 are multi-layer perceptrons, which do not need to be invertible. In our case, they have five hidden layers and their activation function is an exponential linear unit. This procedure can be inverted to efficiently retrieve $\mathbf{u}$ from $\mathbf{v}$.

c. Training the networks

Not all parameter directions in the parameter space are equally constrained by the data. Thus ,we split the summary statistics vector in two halves (one per parameter to infer) and pre-train summary networks to infer each parameter individually.this is motivated by the expected misbehaviour of variational optimisation for an inference whose parameters are under significantly different constraints. Hence, the good performance at inferring an easily learnt parameter (such as K_α) does not prevent the network from converging towards a better optimum where it infers more challenging parameters as well. We do this by using the output of the encoder GNN as an input to a multi-layer perceptron, and optimizing the so-obtained regressor to infer a given parameter in a regression setting. The multi-layer perceptron is then discarded, and the outputs of the parameter-specific GNNs are concatenated to form the summary statistics vector used in the coupled inference. Weights of the summary networks are then frozen and only the invertible part of the network is trained with the objective function presented in equation 5.

3. Exact posterior inference

To compute exact posteriors, likelihood values were computed on grids of points in parameter space. We picked a uniform prior on $(0, 2)$ for α and a log-uniform

one for τ_c and K_α, which spanned 8 orders of magnitude. There was thus no coupling between parameters in the priors. The parameters used to generate trajectories during training were sampled from these same priors.

4. Cramér-Rao bound

Formally, we consider any estimator of the parameters $\boldsymbol{\theta}$ to be a (possibly implicit) function of the recorded trajectory, $\mathbf{R}$, i.e., $\hat{\boldsymbol{\theta}} = \mathbf{T}(\mathbf{R})$. We denote by $\boldsymbol{\psi}(\boldsymbol{\theta}) = E[\mathbf{T}(\mathbf{R})]$ its expectation, and by $\Gamma(\boldsymbol{\theta}) = E[(\mathbf{T}(\mathbf{R}) - \boldsymbol{\psi}(\boldsymbol{\theta}))(\mathbf{T}(\mathbf{R}) - \boldsymbol{\psi}(\boldsymbol{\theta}))^\top]$ its covariance matrix. Finally, $\mathbf{I}(\boldsymbol{\theta})$ is the Fischer information matrix, whose elements are given by $\mathbf{I}_{n,m}(\boldsymbol{\theta}) = E\left[\frac{\partial}{\partial\theta_n}\log p(\mathbf{R}|\,\boldsymbol{\theta})\frac{\partial}{\partial\theta_m}\log p(\mathbf{R}|\,\boldsymbol{\theta})\right]$.

The Cramér-Rao bound states that, for any unbiased estimator $\mathbf{T}$,

$$\Gamma(\boldsymbol{\theta}) \geq \nabla\boldsymbol{\psi}(\boldsymbol{\theta})\left[\mathbf{I}(\boldsymbol{\theta})\right]^{-1}\left[\nabla\boldsymbol{\psi}(\boldsymbol{\theta})\right]^\top, \tag{6}$$

where $\nabla\boldsymbol{\psi}$ is the Jacobian of $\boldsymbol{\psi}$. In particular, this matrix inequality implies the following lower bound on the variance of any unbiased estimator of a single parameter:

$$\mathrm{Var}_{\boldsymbol{\theta}}(T_n(\mathbf{R})) \geq \left[\mathbf{I}(\boldsymbol{\theta})^{-1}\right]_{n,n} \tag{7}$$

5. Supplementary figures

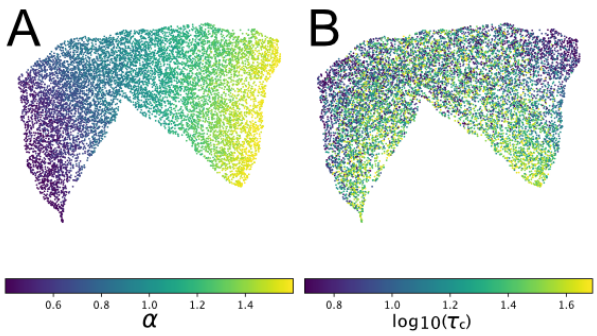

FIG. S1. **Latent space representations of individual trajectories.** 2D visualisation of summary vectors (one point per trajectory), obtained by UMAP and colored according to A: their anomalous diffusion exponent α, and B: their correlation time τ_c. Trajectories are of length $N = 1,000$.

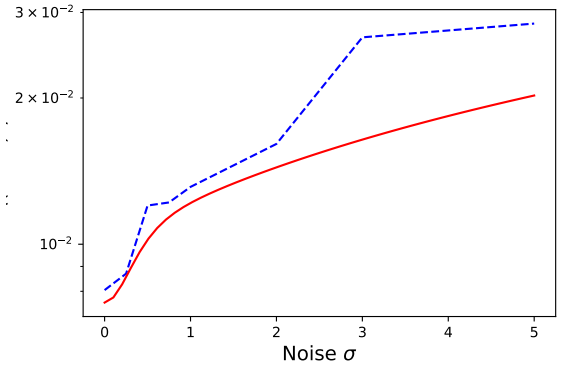

FIG. S2. **Robustness to noise.** Mean square error on α estimated with amortised inference compared to the Cramér-Rao bound as a function of positioning noise σ. Trajectories are of length $N = 200$ and generalised diffusivity 1. Positions are independently corrupted with Gaussian noise of variance σ^2.

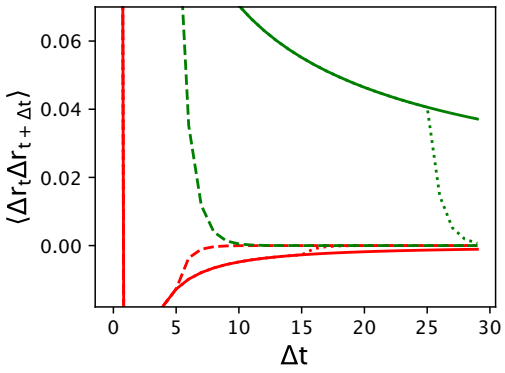

FIG. S3. **Temporal correlations of fBm with finite decorrelation time.** Autocovariance of increments of the fBm trajectory, with finite correlation time τ_c, in the sub-diffusive and super-diffusive case. Red curves correspond to $\alpha = 0.6$, and $\tau_c = 5$ (dashed line), 15 (dotted line), ∞ (plain line). Green curves correspond to $\alpha = 1.4$, and $\tau_c = 5$, 25, ∞.