UNIVERSITY OF COPENHAGEN
FACULTY OF SCIENCE

Master's thesis

Learning the source of non-equilibrium dynamics

Tomás Ricardo Basile Álvarez

Advisor: Karel Proesmans

Submitted: May 2, 2025

Contents

1 Introduction 6
2 Theory 8
2.1 Stochastic Thermodynamics 8
2.2 Langevin equation 9
2.3 Entropy Production 12
2.4 Active Noise 14
3 Estimating Entropy Production 17
3.1 NEEP 17
3.2 Applying NEEP to a Brownian Particle on a Ring 19
3.3 NEEP for an Active Ornstein-Uhlenbeck Particle 22
4 Classifying Active Noise 27
4.1 Feedforward Neural Network (FNN) 29
4.2 Recurrent Neural Network (RNN) 30
4.3 Theoretical Computation 32
4.4 Results for Varying Trajectory Lengths 35
5 Experimental Cellular Data 37
5.1 Data Analysis 37
5.2 Classifying Active Noise 39
5.3 Type of Active Noise 40
5.4 Predicting Diffusion Coefficients 41
5.5 Power Spectrum Analysis 44
6 Maximum Likelihood to Predict the Force on a Stochastic System 45
6.1 Problem Statement 46
6.2 Calculation and Maximization of the Log Likelihood . . . 46
6.3 Results on Simulated Data 48
6.4 Application to Cellular Data 50
7 Cellular Nuclei Data 51
7.1 Data Analysis 51
7.2 Neural Network Training 54
7.3 Predictions on real data 54
7.4 Power Spectrum Analysis 56
8 Conclusion 57

Appendix A: Computationally solving Langevin's equation 59
Appendix B: Theoretical path integral calculation 61
Appendix C: Correlation calculation 73

Bibliography **81**

Abstract

Understanding the motion of living cells and other active matter systems requires distinguishing between passive fluctuations and actively driven dynamics. However, identifying active noise from experimental trajectories remains a challenge. In this thesis, we demonstrate the effectiveness of machine learning for detecting active noise in trajectory data. By training a long short-term memory (LSTM) neural network on synthetic trajectories, we develop a classifier that reliably distinguishes between active and passive motion. When applied to experimental data, our method confirms that the observed cellular motion is actively driven and is best described by an Ornstein–Uhlenbeck active noise model. Furthermore, tests on simulated data show that the classifier's performance approaches the theoretical optimal limit, which is derived in this work. This approach provides a powerful and generalizable framework for analyzing non-equilibrium dynamics and could be applied to a wide range of experimental systems exhibiting active motion.

Acknowledgments

I would like to express my deepest appreciation to my advisor, Karel Proesmans, whose guidance and expertise made this thesis possible. It was a pleasure to work and learn with you and take part on this fascinating research.

I am very grateful to my mother Angeles, my father Roberto and my siblings Ale, Pablo, Martin and Andrés for their unconditional and everlasting support.

To my girlfriend, Jessie, I cannot begin to express my gratitude. Thank you for always being by my side, for the patience, support and inspiration you bring me every day.

I am grateful to all the friends I have made at KU: Niels, Joris, Effry, Filippo, Nathaniel, and many more. I would like to specifically thank Daniel for always offering his help and making this experience even more memorable.

Last but not least, I would like to offer my appreciation to Namiko, Kim, Thomas, Amin, and other excellent professors I had the privilege to learn from.

1 Introduction

Living organisms operate fundamentally out of equilibrium, driven by a continuous influx and dissipation of energy. This persistent non-equilibrium state is crucial for sustaining life in cells [20, 6], playing a key role in processes such as DNA replication [50, 33], intracellular transport, and cellular organization [21, 52, 32].

Describing living systems, therefore, requires a framework grounded in non-equilibrium thermodynamics [37]. Over the past few decades, the theory of stochastic thermodynamics has emerged as a powerful tool for understanding such non-equilibrium systems. This framework has revealed fundamental principles that apply even to systems far from equilibrium [42].

One particularly intriguing class of non-equilibrium systems is active matter [30, 17, 39, 53, 16]. Active matter consists of particles that extract energy from their environment and convert it into motion. These systems inherently break time-reversal symmetry due to constant energy dissipation and exhibit remarkable emergent phenomena, such as active turbulence [2], collective motion [10, 3], flocking [48], and active stress [34]. Biological systems provide many examples of active matter, including the cytoskeleton of living cells [44], nuclear fluctuations within cells [15], swarming bacteria [19], and, on larger scales, animal groups such as flocking birds and human crowds [14].

A key challenge in studying biological motion is determining whether an observed trajectory results from active dynamics or if it can be explained by passive thermal fluctuations. Experimentally distinguishing between these two cases is often difficult, as active fluctuations can be subtle or masked by environmental noise.

In this work, we present a machine learning-based method for classifying whether a system is active or passive, applicable to experimental trajectories. Specifically, we train a long short-term memory (LSTM) neural network on synthetic data generated from mathematical models of passive and active motion, including the Active Ornstein-Uhlenbeck Particle (AOUP) [8, 16, 46] and the Rotational Brownian Particle [11, 5]. We also derive a theoretical bound to the effectiveness of such a classifier and compare it with our machine learning results, finding that the LSTM approaches the theoretical limit, while a simpler feed forward neural network fails to capture important trajectory information. By applying this model to experimental data, we identify the presence of ac-

tive noise in living cell trajectories, demonstrating the method's effectiveness in real-world applications.

This approach builds on previous studies that apply machine learning (particularly neural networks) to problems in physics [13]. In particular, in stochastic thermodynamics, recent studies have used these techniques to classify the direction of time's arrow [24] and estimate a system's entropy production [26, 35, 23], a topic we briefly explore in chapter 3. As mentioned before, here we address a different challenge: developing a classifier capable of distinguishing between active and passive systems and estimating the active diffusion coefficient.

Outline

Chapter 2 introduces the foundational concepts of stochastic thermodynamics and the Langevin formalism, providing a short introduction of the necessary tools for the rest of the thesis. In Chapter 3, we discuss an existing method from the literature [26] that employs a neural network to predict a system's entropy production. However, we find that this approach fails to estimate entropy production in cases involving hidden entropy contributions, such as active matter.

In Chapter 4, we develop a neural network designed to classify whether a system has active noise. We explore both a feed forward neural network and a recurrent neural network. Moreover, we introduce a theoretical approach for classification using path integrals to compute the probability of a system's trajectory being active or passive and compare it with the machine learning results.

In Chapter 5, we apply this neural network to analyze experimental cell motion data and draw conclusions about its activity. In Chapter 6 we present a different method, with the goal of estimating the conservative force that is felt by a stochastic system. This method is initially used for passive systems in [22], so we extend it to active ones and apply it on the cellular data trajectory, serving as a consistency check for the results of Chapter 5. Next, in Chapter 7 we extend our analysis to trajectories of cellular nuclei, uncovering limitations in our results.

2 Theory

We begin with an overview of the theoretical framework needed for this thesis. First, we introduce the fundamental concepts of stochastic thermodynamics and the Langevin equation. Then, we examine the notion of entropy production within stochastic thermodynamics. Finally, we present the mathematical model used to describe an active particle.

2.1 Stochastic Thermodynamics

Originally, thermodynamics was conceived as a discipline focused exclusively on macroscopic systems, with statistical mechanics developed to explain how macroscopic thermodynamic properties emerge from microscopic interactions. However, over the past few decades, this perspective has shifted. The application of thermodynamic principles to microscopic systems has become increasingly widespread [29], driven in part by the study of biomolecules, which can be analyzed as small machines that consume fuel and perform work [45].

In this context, stochastic thermodynamics has emerged as a theoretical framework extending classical thermodynamics to small-scale systems. At these scales, characteristic energies are comparable to $k_B T$, making thermal fluctuations a dominant factor. Unlike macroscopic systems, where properties such as temperature, pressure, and entropy can be treated as deterministic, microscopic systems experience significant random fluctuations due to their interaction with a thermal environment [9].

Stochastic thermodynamics provides tools to describe and analyze such systems by defining thermodynamic concepts like heat, work, and entropy at the level of individual trajectories [9].This offers a framework to understand non-equilibrium behavior and energy conversion at microscopic scales. This framework is particularly valuable for studying systems far from thermodynamic equilibrium. Importantly, the particle under study does not need to be in equilibrium with its thermal environment; it may be externally driven or, as in active systems, it can be self-propelled [29].

Stochastic thermodynamics has found applications across many fields [29]. In biophysics, it helps elucidate the functioning of molecular motors, while in nanotechnology, it informs the design of efficient nanoscale engines and devices. By linking microscopic fluctuations to macroscopic thermodynamic quantities

through fluctuation theorems and other fundamental principles, stochastic thermodynamics has become an essential tool for understanding energy and matter at small scales.

2.2 Langevin equation

To introduce the Langevin equation, consider a small object, such as a cell or organelle, floating in a fluid. If the object is sufficiently small, collisions with the fluid molecules become significant, causing the object to move in a complex and random manner, known as Brownian motion [7].

To describe this phenomenon mathematically, we use the Langevin equation, which combines deterministic forces (such as those from external potentials) with stochastic forces arising from the random collisions of the surrounding fluid molecules. For a one-dimensional system, we start with Newton's second law:

$$m\dot{v} = F, \tag{1}$$

where m is the object's mass, v its velocity and F the net force acting on it.

Modeling F is crucial, as it includes both deterministic forces, such as those due to external potentials or dragging, and stochastic forces from countless fluid molecule collisions. Since modeling each collision directly is impractical, we represent the stochastic component as a random force. The resulting Langevin equation is:

$$m\dot{v} = -\gamma v + f(x) + \gamma\sqrt{2D}\,\xi(t), \tag{2}$$

where $-\gamma v$ represents the drag force proportional to velocity (γ being the drag coefficient), $f(x)$ is the deterministic force (for example, coming from a potential $U(x)$ or from dragging the particle), and $\xi(t)$ is the random force due to molecular collisions. This random force is multiplied by a constant factor $\gamma\sqrt{2D}$, written in such a way that D (the diffusion coefficient) appears explicitly in this factor, as we will see later.

The random force $\xi(t)$ is a stochastic variable characterized by two key properties:

- **Zero mean:** Since molecular collisions occur uniformly in all directions, the average force is zero:

$$\langle \xi(t) \rangle = 0,$$

where $\langle \cdot \rangle$ indicates an ensemble mean, meant to be taken over a large amount of copies of the system, all created in the same way but evolving differently due to the randomness of ξ.

- **Delta-correlated in time:** The instantaneous nature of collisions implies no correlation between forces at different times:

$$\langle \xi(t)\xi(t') \rangle = \delta(t - t') \tag{3}$$

To fully define $\xi(t)$, we assume it is Gaussian noise. This means that integrating $\xi(t)$ over a short time interval Δt produces a Gaussian random variable, which we call ΔW:

$$\Delta W := \int_t^{t+\Delta t} \xi(s) ds, \tag{4}$$

with $\langle \Delta W \rangle = 0$ and $\langle \Delta W^2 \rangle = \Delta t$. These two moments of ΔW can be derived from the properties defined for $\xi(t)$. However, the extra assumption here is that ΔW doesn't follow just any distribution with these two moments, but a Gaussian distribution. This Gaussian assumption is important when computationally solving Langevin's equation, as explained in Appendix A.

Now we will find an important relation between the constants in the Langevin equation 2. We do so by solving the equation for the case $f(x) = 0$ and then calculating the mean square velocity of the particle in the long-time limit, resulting in:

$$\lim_{t\to\infty} \langle v(t)^2 \rangle = \frac{D}{m\gamma}. \tag{5}$$

In this long time limit, the system has reached equilibrium and therefore the equipartition theorem holds, meaning that:

$$\lim_{t\to\infty} \frac{1}{2} m \langle v(t)^2 \rangle = \frac{1}{2} k_B T. \tag{6}$$

Combining Eq. 5 and Eq. 6, we find that:

$$D\gamma = k_B T, \tag{7}$$

known as **Einstein's relation** [9]. This equation connects the fluctuations (via D) with the friction (via γ), ensuring that thermal equilibrium is achieved as friction and fluctuations balance over time. This is a specific case of the more general fluctuation-dissipation theorem.

2.2.1 Overdamped Equation

When the particle's inertia is negligible compared to the damping force, the inertial term can be ignored, effectively setting it to zero. This assumption is valid for small particles in highly viscous media, such as organelles or cells moving in a fluid environment. In this overdamped limit, the Langevin equation 2 simplifies to:

$$0 = -\gamma v + f(x) + \gamma\sqrt{2D}\,\xi(t),$$

and therefore, considering that $v = \dot{x}$, we get the Overdamped Langevin equation:

$$\dot{x} = \frac{1}{\gamma} f(x) + \sqrt{2D}\,\xi(t). \tag{8}$$

This formulation explains why, in the original Langevin equation 2, I wrote the coefficient of $\xi(t)$ as $\gamma\sqrt{2D}$ and said that D was identified as the diffusion coefficient. To demonstrate that D is indeed the diffusion coefficient, consider the case where no external force is present ($f(x) = 0$), leaving only drag and random forces. The equation then reduces to:

$$\dot{x} = \sqrt{2D}\,\xi(t).$$

This describes the free motion of a Brownian particle. Solving formally by integration yields:

$$x(t) = x_0 + \sqrt{2D}\int_0^t \xi(s)ds.$$

Then, we can compute the variance of $x(t)$ as:

$$\begin{aligned}
\langle (x(t) - x(0))^2 \rangle &= \left\langle \left(\sqrt{2D}\int_0^t \xi(s)ds \right)^2 \right\rangle = \left\langle 2D\int_0^t \int_0^t \xi(s)\xi(s')dsds' \right\rangle \\
&= 2D\int_0^t \int_0^t \langle \xi(s)\xi(s') \rangle dsds' \\
&= 2D\int_0^t \int_0^t \delta(s - s')dsds' \\
&= 2D\int_0^t ds' \\
&= 2Dt.
\end{aligned}$$

Thus, the variance of $x(t)$ grows linearly with time t, with a proportionality constant of $2D$. This result matches the behavior predicted by the diffusion equation, where D is defined as the diffusion coefficient.

2.3 Entropy Production

In this section, we examine a particle described by the overdamped Langevin equation, which, as derived in Eq. 8, takes the form:

$$\dot{x} = \frac{1}{\gamma} f(x, \lambda) + \sqrt{2D}\, \xi(t),$$

where the force $f(x, \lambda)$ arises from a conservative potential $U(x, \lambda)$ which in general may depend on a time-varying parameter λ. Also, as before, the thermal noise ξ has correlations given by $\langle \xi(t)\xi(t') \rangle = \delta(t - t')$.

We now define a mathematical expression for the stochastic entropy of this particle. We start by first considering the Fokker-Planck equation for the system.

2.3.1 Fokker-Planck Equation

The Fokker-Planck equation is a partial differential equation whose solution $p(x, t)$ represents the probability distribution of the particle's position at time t. That is, the probability of finding the particle at position x at time t. This equation is given by [42]:

$$\begin{aligned} \partial_t p(x,t) &= -\frac{\partial}{\partial x}\left(\frac{1}{\gamma} f(x,\lambda) p(x,t) - D\partial_x p(x,t)\right) \\ &:= -\frac{\partial}{\partial x} j(x,t), \end{aligned} \tag{9}$$

where $j(x, t)$ denotes the probability current. For a normalized initial distribution $p(x, 0) = p_0(x)$, the equation can be solved to find the time evolution of $p(x, t)$.

2.3.2 Entropy

Given a trajectory $x(t)$, we want to define its entropy $s(t)$ along the path, known as **stochastic entropy**. Defining this may seem counterintuitive, as entropy was initially conceived as a property of an ensemble of particles rather than a quantity defined for a single-particle trajectory. However, we can define the trajectory dependent entropy of the particle as [42, 41, 9]:

$$s_{sys}(t) = -\ln p(x(t), t), \tag{10}$$

where $p(x, t)$ is the solution of the Fokker-Planck equation. This definition should be multiplied by k_B, but for simplicity, we set $k_B = 1$. Large values of

$s_{sys}(t)$ indicate the particle is in a highly improbable state, while smaller values correspond to more likely states, therefore reflecting the "surprise" of observing the particle at x at time t.

For an ensemble of particles governed by the same dynamics, the **ensemble-averaged entropy** is given by:

$$\langle s_{sys}(t) \rangle = -\int dx\ p(x,t) \ln p(x,t). \tag{11}$$

This coincides with the well-known definition of Shannon entropy [29, 36].

Rate of Entropy Change

We can determine the rate of change of the system's entropy [41] by differentiating Eq. 10. To get the total time derivative of $s_{sys}(t)$, we need to take the partial derivative with respect to both t and x:

$$\begin{aligned} \dot{s}_{sys}(t) &= -\frac{\partial_t p(x,t)}{p(x,t)}\bigg|_{x(t)} - \frac{\partial_x p(x,t)}{p(x,t)}\bigg|_{x(t)} \dot{x} \\ &= -\frac{\partial_t p(x,t)}{p(x,t)}\bigg|_{x(t)} + \frac{j(x,t)}{Dp(x,t)}\bigg|_{x(t)} \dot{x} - \frac{f(x,\lambda)}{\gamma D}\bigg|_{x(t)} \dot{x}, \end{aligned} \tag{12}$$

where $\Big|_{x(t)}$ means that we evaluate along the trajectory $x(t)$. For the second line, we used the definition of the current $j(x,t) = \frac{1}{\gamma} f(x,\lambda)p(x,t) - D\partial_x p(x,t)$ to substitute the derivative with respect to x.

Using Einstein's relation 7, the last term can be identified as the change in entropy due to heat dissipation into the surrounding medium. That is, the rate of change of the medium's entropy due to the heat dissipated by the particle:

$$\dot{s}_m := \frac{\dot{q}(t)}{T} = \frac{f(x,\lambda)\dot{x}}{T}, \tag{13}$$

where $\dot{q}(t) = f(x,\lambda)\dot{x}$ is the heat flow to the medium [42].

Total Entropy and Entropy Production

Then, we can find the rate of change of the total entropy. This is done by adding the rate of change of the system's entropy (Eq. 12) and the medium's entropy (Eq. 13) [41]. This is what we call the **stochastic entropy production rate:**

$$\begin{aligned}\dot{s}_{tot}(t) &= \dot{s}_m(t) + \dot{s}_{sys}(t) \\ &= -\frac{\partial_t p(x,t)}{p(x,t)}\bigg|_{x(t)} + \frac{j(x,t)}{Dp(x,t)}\bigg|_{x(t)} \dot{x}.\end{aligned} \tag{14}$$

We can also obtain the ensemble averages of these quantities as done in [41]. For the medium entropy rate, we get as a result:

$$\langle \dot{s}_m(t) \rangle = \int dx\ F(x,t) j(x,t)/T.$$

And for the total entropy we get the ensemble average entropy production rate:

$$\langle \dot{s}_{tot}(t) \rangle = \int dx\ \frac{j(x,t)^2}{Dp(x,t)} \geq 0.$$

Note that the ensemble average entropy production rate is non negative, reflecting the second law of thermodynamics.

Fluctuation Theorem

From these results, it is possible to derive the integral fluctuation theorem:

$$\langle e^{-\Delta s_{tot}} \rangle = 1, \tag{15}$$

where Δs_{tot} is the change of total stochastic entropy along a trajectory. From there, one can also prove that

$$\frac{p(-\Delta s_{tot})}{p(\Delta s_{tot})} = e^{-\Delta s_{tot}}, \tag{16}$$

where $p(\Delta s_{tot})$ is the probability of observing a trajectory in which the entropy production is Δs_{tot}. We see that the probability of having negative stochastic entropy production is exponentially smaller than the probability of observing positive entropy production.

2.4 Active Noise

Active noise models the additional forces acting on particles in systems capable of self-propulsion, such as those powered by molecular motors [30, 46]. These forces need to be incorporated to Langevin's equation and they can be described in a variety of ways, as we discuss below.

2.4.1 Incorporating Active Noise into the Langevin Equation

To account for active noise in Langevin dynamics, we extend the overdamped Langevin equation (Eq. 8) by introducing a term representing self-propulsion. In active matter systems, particles consume energy from their surroundings to generate motion. This motion exhibits directional persistence over a characteristic timescale, requiring a noise term with finite time correlation [17]. Therefore, it can't be described by the term we already use for normal diffusion.

Then, to model the motion of an active (and overdamped) particle, we start with the Overdamped Langevin equation we had in Eq. 8 and add to it an active noise term, obtaining:

$$\dot{x} = \frac{1}{\gamma} f(x) + \sqrt{2D}\xi(t) + \sqrt{2D_a}\eta(t), \tag{17}$$

where the new terms have the following meanings:

- D_a: Active diffusion coefficient.
- $\sqrt{2D_a}\eta(t)$: Represents the active force acting on the particle.
- $\eta(t)$: Active fluctuations, modeling a nonthermal noise source associated with the particle's self-propulsion. This is generally not a directly measurable quantity, since it embodies the stochastic self-propulsion mechanism of the particle.

2.4.2 Models for active noise

Various models are used to define $\eta(t)$ in Eq. 17. Below are some of the most common approaches:

- **Ornstein-Uhlenbeck Process:** In this model, the active noise follows the equation [8, 16, 46]:

$$\dot{\eta} = -\frac{1}{\tau_a}\eta(t) + \frac{1}{\tau_a}\zeta(t). \tag{18}$$

  Here $\zeta(t)$ is Gaussian noise with mean zero and delta correlation, that is, $\langle\zeta(t)\rangle = 0$ and $\langle\zeta(t)\zeta(t')\rangle = \delta(t-t')$. By solving eq. 18, we can obtain the correlation of η:

$$\langle\eta(t)\eta(t')\rangle = \frac{1}{2\tau_a} e^{-|t-t'|/\tau_a},$$

and therefore we see that τ_a measures the characteristic time of the correlations of η, so it quantifies how long the active fluctuations are persistent.

When we use this type of noise, the whole model of Eq. 17 with Eq. 18 is called the **Active Ornstein-Uhlenbeck Particle (AOUP)**.

- **Run and Tumble:** In this model, η alternates between the discrete values $+1$ and -1, with random switching. The time between transitions follows an exponential distribution with mean τ_a [18]. This type of noise is frequently used to describe bacterial motion [17].
- **Rotational Brownian Particle (RBP):** Here, the active noise is modeled as $\eta = \cos(\theta(t))$, where $\theta(t)$ evolves by:

$$\dot{\theta} = \sqrt{2D_{rot}}\,\zeta(t),$$

with D_{rot} being a constant and $\zeta(t)$ a Gaussian white noise [11, 5].

Each of these models provides a different perspective on active noise, tailored to the specific characteristics of the system being studied.

3 Estimating Entropy Production

As discussed in the introduction, nonequilibrium states are ubiquitous, particularly in biological systems [6]. Understanding these systems requires a detailed analysis of their energetics. However, directly measuring heat flow (a key component of nonequilibrium thermodynamics) can be highly challenging. A practical alternative is to estimate the mean entropy production rate $\langle \dot{s}_{tot} \rangle$, which provides a measure of the average heat dissipated into the environment per unit time.

Over the years, various techniques have been developed to estimate entropy production. Some of these methods rely on detailed knowledge of the underlying dynamics, such as calculating probability currents and densities [28]. More broadly applicable methods have also emerged, which do not require explicit knowledge of the governing equations. Many of these approaches utilize the thermodynamic uncertainty relation (TUR) to establish lower bounds on entropy production [35, 49, 4].

Recently, machine learning has opened new ways of estimating entropy production [23, 26], and other applications such as inferring the arrow of time [24]. In particular, the method proposed in [26] employs a neural network to estimate not only the ensemble-average entropy production but also the entropy production along individual trajectories. Here, we summarize this innovative approach and its implications for nonequilibrium thermodynamics.

3.1 NEEP

The method introduced in [26], called the **Neural Estimator for Entropy Production (NEEP)**, is designed to estimate the entropy production of a particle solely from its trajectory data, without requiring knowledge of the underlying dynamics. Below, we outline how the method works.

We begin with a Markov chain trajectory $s_1, s_2, \cdots, s_L$, where L is the length of the observed trajectory, and each s_i represents the state of the system at time step i. This state can be a scalar (e.g., the position of a particle in one-dimensional space) or a multidimensional array (e.g., the positions of multiple particles, or a particle that requires more than one dimension to be described).

Then, we will define a function $h_\theta(s_t, s_{t+1})$, which takes two consecutive states of the system, outputs a real number and depends on a big number of trainable parameters θ. This function $h_\theta(s_t, s_{t+1})$ represents what the neural network of NEEP will do. That is, the neural network takes as input two consecutive states s_t, s_{t+1}, then it does some operations to them, which depend on the weights and biases of the neural network (denoted as a whole as θ) and it outputs a single real number.

We still don't know what the parameters θ should be or how is this related to physics. But given this, we define $\Delta_\theta S$ as

$$\Delta S_\theta(s_t, s_{t+1}) = h_\theta(s_t, s_{t+1}) - h_\theta(s_{t+1}, s_t), \tag{19}$$

which is antisymmetric, since $\Delta S_\theta(s_t, s_{t+1}) = \Delta S_\theta(s_{t+1}, s_t)$. As we will see later, this quantity will be our approximation for the stochastic entropy production when the system jumps from state s_t to s_{t+1} in steady state.

Now we need to define how to train the neural network to find the optimal values of the parameters θ, such that $\Delta_\theta S$ defined in Eq. 19 indeed approximates the entropy production. As we will prove later, this is done by finding the parameters that maximize the following objective function:

$$J(\theta) = \mathbb{E}_t \mathbb{E}_{s_t \to s_{t+1}} \left[\Delta S_\theta(s_t, s_{t+1}) - e^{-\Delta S_\theta(s_t, s_{t+1})} \right], \tag{20}$$

where $\mathbb{E}_t$ denotes the expectation over time steps t, sampled from $\{1, \cdots, L-1\}$ and $\mathbb{E}_{s_t \to s_{t+1}}$ is the expectation over transition $s_t \to s_{t+1}$.

In summary, we start with an ensemble of trajectories, each of the form $s_1, s_2, \cdots s_L$. A neural network is constructed to take as input two consecutive states s_t and s_{t+1} of a trajectory and output a real number $h_\theta(s_t, s_{t+1})$, where θ are all the trainable parameters of the neural network (the weights and biases). The network is then trained to maximize the objective function $J(\theta)$, defined in Eq. 20. Once trained, the network can be used to compute $\Delta S_\theta(s_t, s_{t+1})$, which approximates the entropy production for any transition $s_t \to s_{t+1}$ in the steady state.

We can quickly see why this method is valid, that is, why the function h_θ that minimizes J is such that $\Delta S(s_t, s_{t+1})$ indeed gives the entropy production of the transition $s_t \to s_{t+1}$. To see this, consider that in steady state, $J(\theta)$ can be written as [26]:

$$J[h] = \sum_{i,j} p_i T_{ij} \left((h_{ij} - h_{ji}) - e^{-h_{ij} - h_{ji}} \right),$$

where $h_{ij} := h(s_i, s_j)$, and $p_i := p(s_i)$ is the steady state probability density evaluated at state s_i. Also, T_{ij} is the probability of having the transition from s_i to s_j at time t.

Then, maximizing $J[h]$ with respect to the outputs of h requires the gradient to vanish, so that when we reach the maximum, we have:

$$\begin{aligned} 0 &= \partial_{h_{\alpha\beta}} J[h] \\ &= \sum_{i,j} \left[p_i T_{ij} (1 + e^{-(h_{ij} - h_{ji})})(\delta_{i\alpha}\delta_{j\beta} - \delta_{i\beta}\delta_{j\alpha}) \right] \\ &= p_\alpha T_{\alpha\beta}(1 + e^{-(h_{\alpha\beta} - h_{\beta\alpha})}) - p_\beta T_{\beta\alpha}(1 + e^{-(h_{\beta\alpha} - h_{\alpha\beta})}). \end{aligned}$$

Rearranging yields:

$$h_{\alpha\beta} - h_{\beta\alpha} = -\ln(p_\beta T_{\beta\alpha} / p_\alpha T_{\alpha\beta}).$$

This is the definition of stochastic entropy production when $T_{ji} = \tilde{T}_{ij}$, with $\tilde{T}$ being the time reversal of T [42]. Therefore, this proves that when J is maximized, h_θ is such that $\Delta S_\theta(s_t, s_{t+1}) = h_\theta(s_t, s_{t+1}) - h_\theta(s_{t+1}, s_t)$ gives the entropy production in steady state for the transition $s_t \to s_{t+1}$.

3.2 Applying NEEP to a Brownian Particle on a Ring

In this section, we demonstrate the application of the NEEP method, described earlier, to estimate the entropy production of a system known as a Brownian particle on a ring.

System Description

We start by defining the model, which consists of a Brownian particle with a periodic potential $V(x)$ of period 1. The particle is being dragged by a constant force $f_0 \geq 0$. Therefore, the particle feels a force of [38]:

$$f(x) = f_0 - V'(x), \quad V(x) = V(x + 1),$$

and the effective potential is given by:

$$U(x) = V(x) - f_0 x.$$

The particle's overdamped motion is governed by the Langevin equation [47]:

$$\dot{x}(t) = \frac{1}{\gamma}\left(-\frac{\partial V}{\partial x} + f_0\right) + \sqrt{2D}\,\xi(t) := \frac{1}{\gamma}f(x) + \sqrt{2D}\,\xi(t), \tag{21}$$

where as always, the thermal noise $\xi(t)$ has zero mean and delta correlation $\langle \xi(t)\xi(t')\rangle = \delta(t-t')$. Also, the heat bath is in equilibrium at constant temperature T. In this setup, the non-conservative force f_0 drives continuous dissipation, making it the source of entropy production.

Trajectory Generation

We generate 2000 trajectories by numerically solving the Langevin equation, starting from initial positions $x(0)$ sampled from the steady state probability distribution, which is [38]:

$$p_{ss}(x) = C\exp\left(-\frac{1}{D}U(x)\right)\left(\int_0^x dy \exp\left(\frac{1}{D}U(y)\right) + e^{f_0/D}\int_x^1 dy \exp\left(\frac{1}{D}U(y)\right)\right),$$

where C is a normalization constant. The trajectories created have a time step of $\Delta t = 1/100$ and each trajectory spans 100 units of time (so the length of the trajectories is $L = 10000$). The potential used is cubic, chosen in particular to be $V(x) = -x + 3x^2 - 2x^3$. Finally, the diffusion coefficient was chosen as $D = 0.2$ and the dragging force as $f_0 = 0.3$.

Fig. 1 shows some of the generated trajectories, where x is allowed to exceed the interval $[0, 1]$ to illustrate how many times the particles goes around the ring.

NEEP

Given all these trajectories, we use the NEEP to estimate their stochastic entropy production. To do so, we create a feedforward neural network which takes as input two consecutive positions in a trajectory, $x(t), x(t+\Delta t)$ and outputs a real number. This output is denoted as $h_\theta(x(t), x(t+\Delta t))$, where θ contains all the trainable parameters of the neural network. The architecture chosen for the network includes three hidden layers with 512 neurons each and ReLU activations (for more about the theory of neural networks, see [1]).

The network is trained to maximize the objective function $J(\theta)$ (Eq. 20). Once trained, the entropy production for a single transition is approximated as:

$$\Delta S_\theta(x(t), x(t+\Delta)) = h_\theta(x(t), x(t+\Delta t)) - h_\theta(x(t+\Delta t), x(t))$$

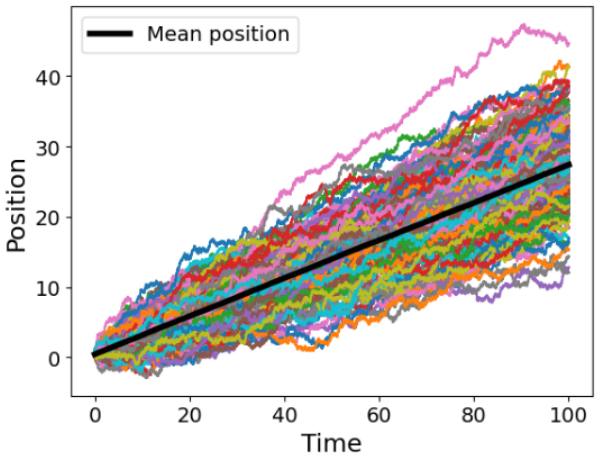


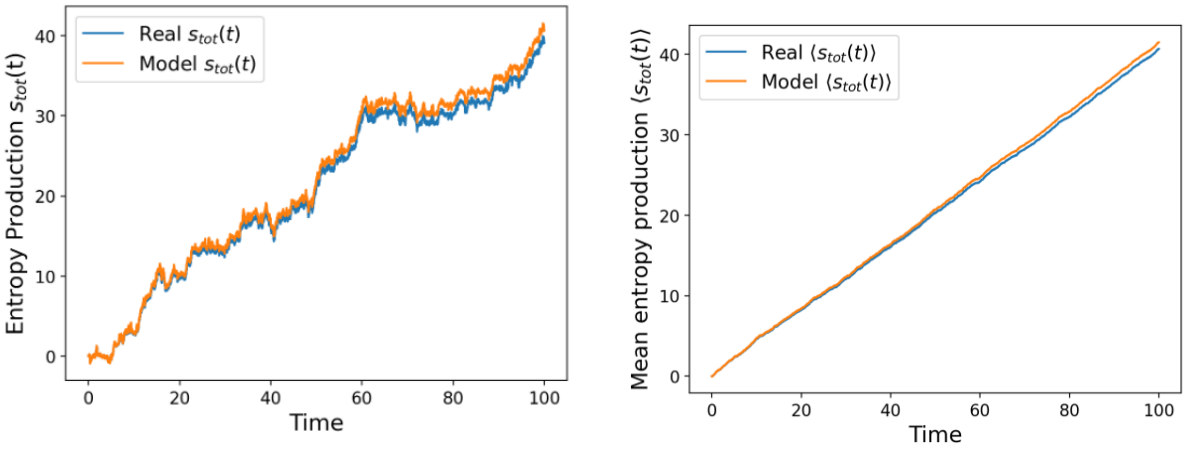
Figure 1: Sample trajectories of a Brownian particle on a ring. Parameters: $D = 0.2$, $f_0 = 0.3$, $V(x) = -x + 3x^2 - 2x^3$. Although x is periodic with period 1, the plot shows the absolute position for clarity and to observe how many trips around the ring the particle has done.

After training, the total entropy production for a trajectory is then obtained by cumulatively summing ΔS_θ over all time steps.

Results

The result of the total entropy production $s_{tot}(t)$ approximated this way is shown in Fig. 2 for a specific trajectory. Also, the calculation of $s_{tot}(t)$ using the theoretical result of Eq. 14 is shown for comparison. It's important to remember that the NEEP approximation to $s_{tot}(t)$ doesn't require any knowledge about the system, only the trajectories, while the theoretical calculation using Eq. 14 requires knowing the equation governing the system and the values of all the parameters. On the other hand, Fig. 2b shows the ensemble average entropy production as a function of time $\langle s_{tot}(t) \rangle$ for the result using NEEP and the theoretical result (both of them obtained by calculating $s_{tot}(t)$ for many trajectories and averaging all of them at each time step).

From Fig. 2, we can see that the NEEP effectively approximates both the stochastic entropy production for individual trajectories and the mean entropy production over the ensemble. Remarkably, this method achieves these estimates using only trajectory data, without requiring explicit knowledge of the underlying dynamics.



(a) Stochastic entropy production.

(b) Mean entropy production.

Figure 2: Theoretical and NEEP results of entropy production for a) the entropy production of a specific trajectory and b) the ensemble mean trajectory of the 2000 trajectories.

3.3 NEEP for an Active Ornstein-Uhlenbeck Particle

We now apply the NEEP framework to estimate the entropy production of an active Ornstein-Uhlenbeck particle (AOUP). As we saw before, the dynamics of the AOUP are governed by the following equations

$$\begin{aligned} \dot{x} &= -\frac{1}{\gamma}U'(x) + \sqrt{2D}\,\xi(t) + \sqrt{2D_a}\,\eta(t), \\ \dot{\eta} &= -\frac{1}{\tau_a}\eta(t) + \frac{1}{\tau_a}\zeta(t), \end{aligned} \tag{22}$$

with $U(x)$ the potential experienced by the particle, D is the diffusion coefficient, D_a the active diffusion coefficient, and τ_a the characteristic time of the active noise. The terms $\xi(t)$ and $\zeta(t)$ represent independent, delta-correlated Gaussian noise sources. In this setup, the active noise $\eta(t)$ introduces a non-equilibrium driving active force. However, $\eta(t)$ is not directly observable; only the particle position $x(t)$ is assumed to be measurable.

Limitations of NEEP in this Case

Because the NEEP method is applied using only the observable $x(t)$, the entropy production estimation obtained will be coarse-grained. This means that it represents a lower bound on the true entropy production, as contributions from the hidden active noise $\eta(t)$ are not accounted for [26]. We will see that this

is indeed the case for two examples, an AOUP in a harmonic potential and an AOUP in a quartic potential.

3.3.1 Harmonic Potential

We begin by considering a harmonic potential, $U(x) = \frac{1}{2}kx^2$, such that the force due to the potential is $-kx$. Similar to the Brownian particle on a ring, we generate multiple trajectories for this model. Specifically, we simulate $10,000$ trajectories using the following parameters: $D = 0.2$, $D_a = 0.2$, $\tau_a = 0.2$, $k = 0.1$, and a time step of $\Delta t = 0.01$.

The trajectories start at $x = 0$ and are simulated over a total time $t = 100$. To ensure that the trajectories are in steady state, we discard the initial half of each trajectory, keeping only the segment from $t = 50$ to $t = 100$. Thus, the effective duration of the retained trajectories is $\tau = 50$. Sample trajectories are shown in Fig. 3.

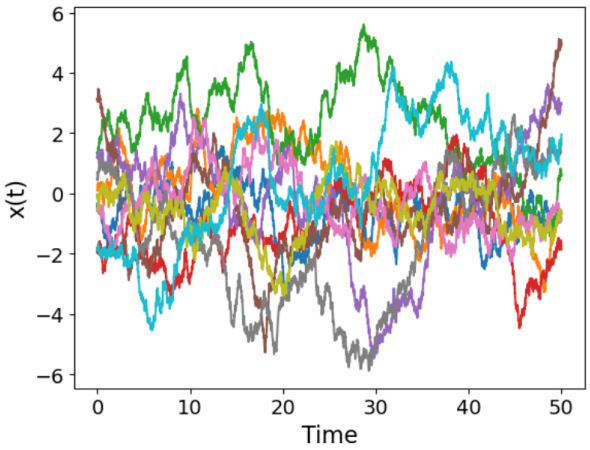


Figure 3: Sample trajectories of an active Ornstein-Uhlenbeck particle in a harmonic potential, with $D = 0.2$, $D_a = 0.2$, $\tau_a = 0.2$, $k = 0.1$, and $\Delta t = 0.01$.

Entropy Production

Using the definition of entropy production from Eq. 14, we calculate it for each trajectory. In this case, the state of the system is two-dimensional, represented by the vector $\mathbf{x} = (x, \eta)$. The dynamics of the AOUP can be written in vector form as:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \sqrt{2D_{\text{mat}}}\,\boldsymbol{\xi}(t),$$

where: $\mathbf{f}(\mathbf{x}) = \left(-kx + \sqrt{2D_a}\,\eta, -\frac{1}{\tau_a}\eta\right)$, $\boldsymbol{\xi}(t) = (\xi(t), \zeta(t))$, and $D_{\text{mat}} = \text{diag}\left(D, \frac{1}{2\tau_a^2}\right)$.

The entropy production rate in two dimensions is given by a generalization of Eq. 14:

$$\dot{s}_{\text{tot}} = -\frac{\partial_t p(\mathbf{x},t)}{p(\mathbf{x},t)}\bigg|_{\mathbf{x}(t)} + \frac{\mathbf{j}(\mathbf{x},t)^T D_{\text{mat}}^{-1}}{p(\mathbf{x},t)}\bigg|_{\mathbf{x}(t)} \dot{\mathbf{x}}, \tag{23}$$

where $p(\mathbf{x},t)$ is the probability distribution of vector $\mathbf{x}$ at time t, and $\mathbf{j}(\mathbf{x},t) = \frac{1}{\gamma}\mathbf{f}(\mathbf{x})p(\mathbf{x},t) - D_{\text{mat}}\nabla p(\mathbf{x},t)$ is the probability current. Since the trajectories are in steady state, $\partial_t p(\mathbf{x},t) = 0$.

In this particular case of a harmonic potential, the steady state distribution $p(\mathbf{x},t)$ is solvable and given by a 2D Gaussian with a covariance matrix C, given by [16]:

$$C = \begin{pmatrix} \frac{D(1+k\tau_a)+D_a}{k(1+k\tau_a)} & \sqrt{\frac{D_a}{2}}\frac{1}{1+k\tau_a} \\ \sqrt{\frac{D_a}{2}}\frac{1}{1+k\tau_a} & \frac{1}{2\tau_a} \end{pmatrix}.$$

Using this distribution $p(\mathbf{x},t)$, we can easily compute the entropy production $s_{tot}(t)$ for any trajectory using Eq. 23. Fig. 4 shows the results for a sample of trajectories, along with the mean rate of entropy production, which turns out to be in this case $\langle \dot{s}_{\text{tot}} \rangle = 4.912$.

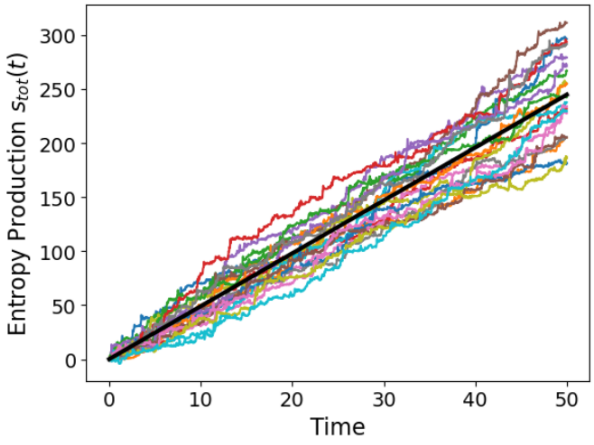


Figure 4: Entropy production $s_{\text{tot}}(t)$ for 20 trajectories of an AOUP in a harmonic potential. The black line represents the mean entropy production $\langle s_{\text{tot}}(t) \rangle$, growing at a practically constant rate of $\langle \dot{s}_{\text{tot}}(t) \rangle = 4.912$.

Entropy Production Using NEEP

Next, we use the NEEP framework to estimate the entropy production from the trajectories. Since only the observable $x(t)$ is input to the NEEP (excluding the hidden variable $\eta(t)$), the result represents a coarse-grained entropy production. After training, the NEEP predicts a mean entropy production rate of $\langle \dot{s}_{\text{tot}}(t) \rangle = 2 \times 10^{-6}$, a drastic underestimation compared to the theoretical value of 4.912. This result is expected, as the NEEP only captures the dynamics observable through $x(t)$, neglecting contributions from the hidden variable $\eta(t)$.

Interestingly, this estimated entropy production by the NEEP aligns with findings in [16], where they study this particular system and define a coarse grained entropy production $\Delta\Sigma$ for a trajectory as:

$$e^{-\Delta\Sigma[\overline{x}]} = \frac{\tilde{p}[\underline{\tilde{x}}|\tilde{x}_0]}{\tilde{p}[\underline{x}|x_0]}, \tag{24}$$

where $\overline{x}$ is the whole trajectory of x, $\underline{x}$ is the trajectory omitting the first position, which is x_0 and $\tilde{x}$ indicates the time reversed trajectory. Finally, p indicates the probability of observing a trajectory $x(t)$ in the forward direction and $\tilde{p}$ in the reversed direction. Notice that the definition only depends on the observable $x(t)$ and not on $\eta(t)$, which is why it is coarse grained.

Having defiend this, they find that for an AOUP with harmonic potential, the coarse-grained entropy production rate vanishes in steady state:

$$\lim_{\tau \to \infty} \frac{\langle \Delta\Sigma \rangle}{\tau} = 0.$$

Therefore, our result with the NEEP, which predicted a mean entropy production rate of around 0 coincides with this theoretical coarse-grained entropy production, even though they both underestimate the true entropy production.

Validation with Full State Information

Finally, as a validation step, we trained the NEEP using both $x(t)$ and $\eta(t)$ as input, as if $\eta(t)$ were observable. In this case, the NEEP predicts a mean entropy production rate of 4.925, closely matching the theoretical value of 4.912. This confirms that the NEEP framework can accurately estimate entropy production when all relevant state variables are included, but gives a coarse grained value when only including $x(t)$.

3.3.2 Quartic Potential

We now repeat the analysis, but for a quartic potential defined as $U(x) = \frac{1}{4}kx^4$, leading to a conservative force of $-kx^3$. As in the harmonic potential case, we generate $10,000$ trajectories using the same simulation approach.

Again, the entropy production for each trajectory is calculated using Eq. 23, only that now the force is $\mathbf{f}(\mathbf{x}) = \left(-kx^3 + \sqrt{2D_a}\,\eta\ ,\ -\frac{1}{\tau_a}\eta\right)$. Unlike the harmonic potential, theres is no analytical expression for $p(\mathbf{x}, t)$ in this case; instead, we approximate it using the generated trajectories.

From this calculation, the theoretical mean entropy production rate obtained is $\langle \dot{s}_{tot} \rangle = 99.76$. Training the NEEP using only the trajectories $x(t)$ yields as expected a much lower result, $\langle \dot{s}_{tot} \rangle = 0.243$, which represents a very crude lower bound to the entropy production rate. However, when the NEEP is trained with both x and the hidden variable η it produces a significantly improved estimate of $\langle \dot{s}_{tot} \rangle = 91.45$.

As with the harmonic potential, the NEEP trained only with $x(t)$ estimates a coarse-grained entropy production. Again, this coarse-grained entropy production coincides with the one defined by Eq. 24. This is shown in fig. 5, where we plot the entropy production for a given trajectory, calculated both with the coarse grained entropy $\Delta\Sigma$ of Eq. 24 (computed as in [16]) and with the NEEP trained only on trajectories $x(t)$.

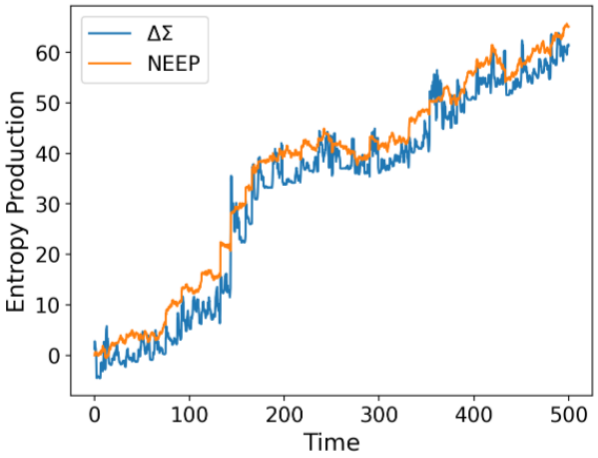


Figure 5: The entropy production $s_{tot}(t)$ for a single trajectory estimated using the coarse-grained entropy production defined in Eq. 24 and using a NEEP trained on trajectories $x(t)$

4 Classifying Active Noise

In the previous chapter, we observed that predicting the entropy production of an active system using neural networks is challenging and often provides only a very loose lower bound. To address this limitation, we shift our focus to something simpler. Here we see if we can train a neural network capable of classifying whether the measured trajectory of a particle exhibits active noise or not.

This chapter details the construction and evaluation of neural networks trained on data generated using the AOUP model, as described around Eq. 18. In later sections, we will apply the trained models to real data from live cells.

The task of designing a neural network to distinguish between trajectories with active noise and those without is initially addressed for trajectories under a harmonic potential, as it serves as a reasonable approximation for the real data we will analyze in Chapter 5. For more complex potentials encountered in Chapter 7, we train a separate neural network with tailored data and present the results there.

Problem Definition

As mentioned before, the system under consideration is an AOUP with har-

monic potential $U(x) = \frac{1}{2}kx^2$, which is governed by the equations:

$$\dot{x} = -\frac{k}{\gamma}x + \sqrt{2D}\,\xi(t) + \sqrt{2D_a}\,\eta(t)$$
$$\dot{\eta} = -\frac{1}{\tau_a}\eta(t) + \frac{1}{\tau_a}\zeta(t). \tag{25}$$

This system is fully described by four parameters: D, D_a, τ_a, and k/γ. We begin by generating synthetic trajectories with and without active noise, which we will use to train our classifier. To generate the trajectories, we choose $D = 0.2$, $\tau_a = 0.5$, and $k/\gamma = 0.1$, and generate 10,000 trajectories: 5,000 with $D_a = 0$ (no active noise) and 5,000 with $D_a = 0.1$. Trajectories are simulated with a time step $\Delta t = 0.01$, starting at $x(0) = 0$. To ensure steady-state conditions, each trajectory is initially simulated for $t_{in} = 100$ (this first part of the trajectory is not saved), followed by saving the trajectory for a total duration of $\tau = 100s$. Examples of trajectories with and without active noise are shown in Fig. 6.

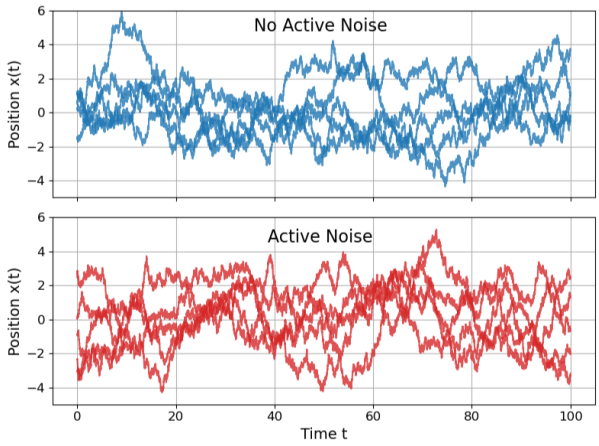


Figure 6: Example of simulated trajectories without active noise (top) and with active noise (bottom).

The goal is to train a neural network to classify each trajectory as either having active noise or not. Trajectories without active noise are labeled as 0, while those with active noise are labeled as 1. The neural network outputs a value between 0 and 1, representing its confidence that a given trajectory exhibits active noise.

Training and Testing Setup

The training dataset consists of 8000 randomly selected trajectories, with the remaining 2000 used as a test set to evaluate classification performance. To quantify the classification error, we use a very common loss function known as Log Loss [1], which is defined as follows: given a trajectory, we define $y \in \{0, 1\}$ as its true label (0 if it has no active noise and 1 if it does) and $p \in [0, 1]$ as the output of the neural network (which as mentioned before, is the confidence of the neural network that the trajectory has active noise). Then, the Log Loss for this prediction is:

$$\text{Log Loss (one prediction)} = -y \log p - (1 - y) \log(1 - p). \tag{26}$$

This can be interpreted as the surprise of the actual label y relative to the prediction p. For example, if the actual label is $y = 1$, then the Log Loss is simply $-\log p$, which is zero if $p = 1$ and gets infinitely big as p approaches 0.

If instead of having only one trajectory, we have a total of N trajectories, each with true label y_k and a predicted probability of having active noise of p_k, then the Log Loss is simply the average of Eq. 26, that is:

$$\text{Log Loss} = -\frac{1}{N} \sum_{k=1}^{N} \left(y_k \log(p_k) + (1 - y_k) \log(1 - p_k) \right). \tag{27}$$

One of the reasons for defining the Log Loss this way is that minimizing it corresponds with maximizing the likelihood function, that is, the probability that the given data set is produced according to the probabilities p_k predicted, which is

$$\text{Likelihood} = \prod_{k|y_k=1} p_k \prod_{k|y_k=0} (1 - p_k).$$

In what follows, we first try with two different architectures of neural networks (feed forward and recurrent), use them for the classification task and measure their error with the Log Loss function. After that, we also present a way to classify the trajectories theoretically.

4.1 Feedforward Neural Network (FNN)

We begin with a Feedforward Neural Network (FNN), a widely used and simple architecture where connections between neurons are directed and do not form cycles, flowing sequentially from one layer to the next [1]. For our task, the network's architecture is as follows:

- **Input layer:** Contains 10,000 neurons, corresponding to the number of measurements in each trajectory ($\tau = 100s$ with a time step $\Delta t = 0.01s$, giving $\tau/\Delta t$ input neurons).
- **Hidden layers:** Three layers with 1,000, 250, and 125 neurons, respectively, each using the ReLU activation function [1].
- **Output layer:** A single neuron with a sigmoid activation function, producing an output between 0 and 1, which represents the confidence of the network for whether the trajectory contains active noise or not.

The hyperparameters, such as the number of layers, neurons in each layer, and learning rate were selected using a grid search [1], choosing the values that best generalize to unseen data. After training the network, we evaluated its performance on the test dataset, and the results are shown as violin plots in Fig. 7.

Fig. 7 demonstrates that while the network can perform some level of classification, its accuracy is very limited. The mean predicted probability of having active noise for trajectories without active noise is 0.41, and for those with active noise, it is 0.61. Showing that on average, the neural network barely has any confidence on the type of trajectory it is looking at. Also, 70.1% of the trajectories without active noise are correctly classified with a probability below 0.5 and 66.4% of the trajectories with active noise are correctly classified with a probability above 0.5. The overall Log Loss for these predictions is 0.589, indicating room for improvement.

4.2 Recurrent Neural Network (RNN)

Next, we attempt the classification task using a Recurrent Neural Network (RNN) [40, 31], which is well-suited for sequential data processing. Unlike FNNs, RNNs incorporate cycles in their architecture, allowing them to maintain a form of memory. This feature makes them particularly effective for analyzing temporal or sequential data.

The basic architecture of an RNN comprises an input layer (denoted by x), a hidden layer (h), and an output layer (y). When processing a trajectory, the RNN operates as follows:

1. The trajectory x is divided into n consecutive segments, each of length "input-dim". For instance, with a trajectory length of $10,000$ and an input-dim of 200, the trajectory is split into $n = 50$ sections: $x_1, x_2, \ldots, x_n$, where each one will contain 50 data points.

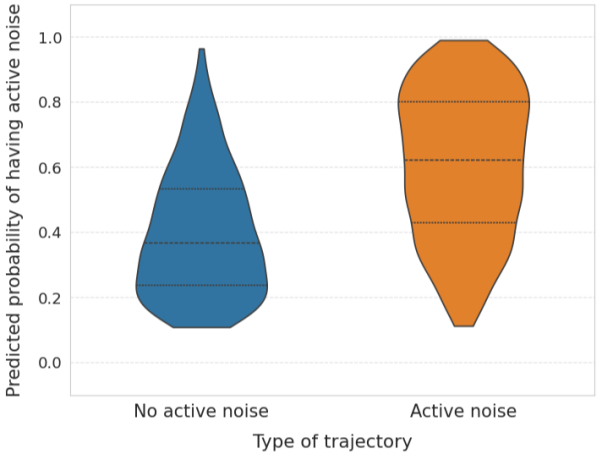


Figure 7: Test data results of the FNN in classifying trajectories as having active noise or not. The left side (blue) is for trajectories without active noise and the right side (orange) for trajectories with active noise. On both sides, the results of the neural network are shown by violin plots, with the vertical axis indicating the output of the neural network, interpreted as the predicted probability of having active noise.

2. At each time step t (from 1 to n), we will define a hidden state h_t (which is a vector of size "hidden-size"). This hidden state contains information about the values of x before time t. The hidden state is updated at each time step by:

$$h_t = \sigma_h(W_{xh}x_t + W_{hh}h_{t-1} + b_h),$$

where W_{xh} is the weight matrix between the input and hidden layer, W_{hh} is the weight matrix for the hidden state, b_h is the bias vector and σ_h is the activation function (usually a hyperbolic tangent) [31]. This way, h_t contains information about the previous time step x_{t-1} (and actually all previous time steps through the term $W_{hh}h_{t-1}$). Note that the weight matrices W_{xh}, W_{hh} are constant for all time steps, and this part is the recurrent operation.

- Then, there is an output y_t defined for each time step, given by:

$$y_t = \sigma_y(W_{hy}h_t + b_y),$$

with W_{hy} the weight matrix between the hidden and output layers, b_y a bias vector and σ_y an activation function (again a hyperbolic tangent).

Abstractly, an RNN is a function f_θ that maps inputs and hidden states to outputs and updated hidden states:

$$(y_t, h_{t+1}) = f_\theta(x_t, h_t) = (\sigma_h(W_{xh}x_t + W_{hh}h_{t-1} + b_h),\ \sigma_y(W_{hy}h_t + b_y)),$$

with $t \in \{1, n\}$, and θ represents all the parameters (weight matrices and biases). In summary, the RNN takes the sequence $x_1, x_2, \cdots, x_n$ and outputs a sequence $y_1, \cdots, y_t$, where each output is informed on previous states of the system through the memory saved in the hidden state. In our case, since we want the final output to be only a single number (representing the probability that the trajectory has active noise), the values $y_1, \cdots y_t$ are then passed through a FNN with two hidden layers that give out a scalar output.

Training of RNNs works as usual, by slowly updating the parameters (in this case $W_{xh}, W_{hh}, b_h, W_{hy}, b_y$) using gradient descent until the network's output minimizes some sort of loss metric in the training set. However, the recurrence in RNNs requires a slightly more complicated calculation of gradients than that done in FNNs, known as backpropagation through time [51].

This standard RNN architecture, suffers from the vanishing and exploding gradient problem, where gradients become too small or too big during backpropagation, due to many multiplications of the same number when backpropagating through the recurrent network. To address this issue, more advanced architectures were developed, such as the Long Short-Term Memory (LSTM) networks [25]. LSTMs mitigate the vanishing gradient problem by discarding or saving information between time steps [31]. This type of network can learn to retain information without changing it during many time steps or to discard it when it is no longer useful, allowing it to keep the gradients from vanishing or exploding.

Therefore, we used an LSTM for our classification task. After training it on the training trajectories, we evaluate it on the test dataset, with results shown in Fig. 8. Compared to the FFN, the LSTM demonstrates significantly better performance in classifying trajectories. The mean probability of having active noise predicted for trajectories without active noise is 0.279, while for those with active noise, it is 0.81. Also, 81.1% of the trajectories without active noise are correctly classified with a probability below 0.5 and 89.7% of the trajectories with active noise are correctly classified with a probability above 0.5. The overall Log Loss is 0.355, reflecting a notable improvement with respect to the FNN.

4.3 Theoretical Computation

In this section, we outline a theoretical method for performing the classification task we've been working with. This will serve us as a limit to evaluate the

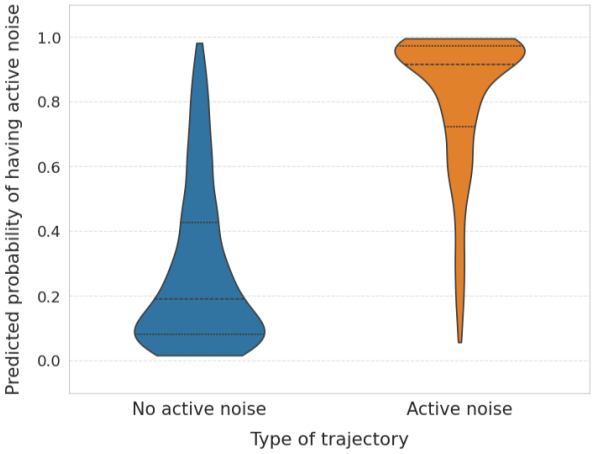


Figure 8: Test data results of the LSTM in classifying trajectories as having active noise or not. Presented in the same way as in Fig. 7.

performance of the neural networks.

As before, we consider two types of trajectories, both characterized by the same parameters k/γ, D, and τ_a. One category corresponds to trajectories without active noise ($D_a = 0$), while the other category has active noise with a fixed active diffusion coefficient $D_a = D_a'$. Given a trajectory, our goal is to calculate the probability that it belongs to the category with active noise, $P(D_a = D_a' \mid \underline{x})$, which is the quantity we were estimating using neural networks. Here $\underline{x}$ represents the trajectory of the particle omitting the starting point.

Using Bayes' theorem, we can express this probability as:

$$\begin{aligned} P(D_a = D_a' \mid \underline{x}) &= \frac{P(\underline{x}|D_a = D_a')P(D_a = D_a')}{P(\underline{x})} \\ &= \frac{P(\underline{x}|D_a = D_a')P(D_a = D_a')}{P(\underline{x}|D_a = D_a')P(D_a = D_a') + P(\underline{x}|D_a = 0)P(D_a = 0)} \\ &= \frac{P(\underline{x}|D_a = D_a')}{P(\underline{x}|D_a = D_a') + P(\underline{x}|D_a = 0)}, \end{aligned}$$

where in the second line we used the law of total probability: $P(\underline{x}) = P(\underline{x}|D_a = D_a')P(D_a = D_a') + P(\underline{x}|D_a = 0)P(D_a = 0)$. We further assume that, a priori, we don't have any preference about the trajectory, so that $P(D_a = 0) = P(D_a =$

$D'_a) = 1/2$. Then, after simplifying we get:

$$P(D_a = D'_a|\underline{x}) = \frac{1}{1 + \dfrac{P(\underline{x}|D_a = 0)}{P(\underline{x}|D_a = D'_a)}}. \tag{28}$$

Therefore, finding the quantity that we care about, the probability that a given trajectory has active noise, can be done by calculating both $P(\underline{x}|D_a = 0)$ and $P(\underline{x}|D_a = D'_a)$, and then using Eq. 28.

$P(\underline{x}|D_a = 0)$ is the probability that the given trajectory $\underline{x}$ is observed for a model without active noise, while $P(\underline{x}|D_a = D'_a)$ is the probability that the trajectory is observed for a model with active diffusion coefficient D'_a. Both of these probabilities require path integrals to be calculated. We show how to compute them in Appendix B, which is based on the results of [16]. The results obtained there are:

$$P(\underline{x}|D_a = 0) = \left(\frac{1}{\sqrt{4\pi D\Delta t}}\right)^N \exp\left\{ -\int_0^\tau dt \left[\frac{(\dot{x}_t - g_t)^2}{4D} + \frac{1}{2}\frac{\partial g_t}{\partial x}\right]\right\},$$

where the subindex t means that the variable is evaluated at time t (and position $x(t)$) and $g(x) := \dfrac{1}{\gamma} f(x)$. Meanwhile, for the active noise case, we get:

$$P(\underline{x}|D_a = D'_a) = K \exp\left(-\int_0^\tau dt \left[\frac{(\dot{x}_t - g_t)^2}{4D} + \frac{1}{2}\frac{\partial g_t}{\partial x}\right] + \frac{D_a}{4D^2}\int_0^\tau dt \int_0^\tau dt' (\dot{x}_t - g_t)\Gamma_\tau(t,t')(\dot{x}_{t'} - g_{t'})\right),$$

with K given by:

$$K = L^{1/4}\left(\frac{1}{\sqrt{4\pi D\Delta t}}\right)^N e^{\tau k_-/(2\tau_a)} \frac{2}{\sqrt{4\sqrt{L} + (1-\sqrt{L})^2(1-\rho^2)}}.$$

$\Gamma_\tau(t,t')$ defined as:

$$\Gamma_\tau(t,t') = \left(\frac{1}{2\tau_a\sqrt{L}}\right) \frac{k_+^2 e^{-\sqrt{L}|t-t'|/\tau_a} + k_-^2 e^{-\sqrt{L}(2\tau - |t-t'|)/\tau_a} - k_+k_- \left[e^{-\sqrt{L}(t+t')/\tau_a} + e^{-\sqrt{L}(2\tau - t - t')/\tau_a}\right]}{k_+^2 - k_-^2 e^{-2\sqrt{L}\tau/\tau_a}},$$

and also $L = 1 + D_a/D$, $k_- = 1 - \sqrt{L}$, and $\rho = e^{-\sqrt{L}\tau/\tau_a}$.

Given these results, we can substitute $P(\underline{x}|D_a = 0)$ and $P(\underline{x}|D_a = D'_a)$ into Eq. 28 to calculate $P(D_a = D'_a|\underline{x})$ for any trajectory. When doing so, some things cancel out and we have our final result:

$$P(D_a = D_a'|\underline{x}) = \frac{1}{1 + C\exp\left\{ -\frac{D_a}{4D^2}\int_0^\tau dt \int_0^\tau dt'(\dot{x}_t - g_t)\Gamma_\tau(t,t')(\dot{x}_{t'} - g_{t'}) \right\}}, \tag{29}$$

with C given by

$$C = \frac{1}{2}L^{-1/4}e^{-\tau k_-/(2\tau_a)}\sqrt{4\sqrt{L} + (1-\sqrt{L})^2(1-\rho^2)}.$$

Eq. 29 allows us to calculate the probability of having active noise for our generated trajectories. We do so for the trajectories in the same test set as used for the neural networks and the results are shown in Fig. 9. The mean predicted probability for trajectories without active noise is now 0.237 and with active noise, it is 0.808. Also, 85.3% of the trajectories without active noise are correctly classified with a probability below 0.5 and 86.6% of the trajectories with active noise are correctly classified with a probability above 0.5. The Log Loss of all these predictions is 0.337.

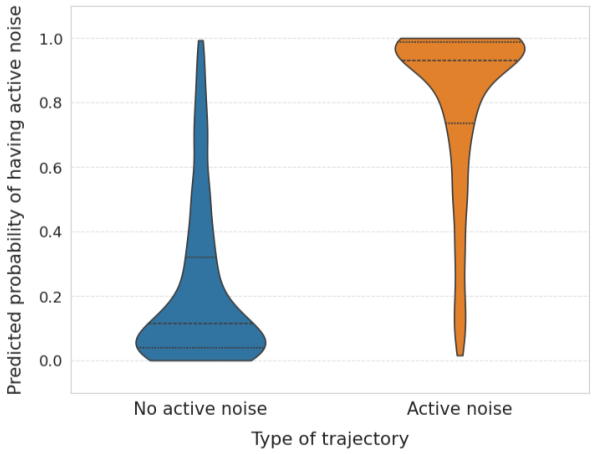


Figure 9: Results of theoretically classifying trajectories of the test set as having active noise or not. Presented in the same way as in Fig. 7.

4.4 Results for Varying Trajectory Lengths

In this section, we extend the classification experiments from the previous sections, conducted with a FNN, a LSTM, and the theoretical approach. Instead of fixing the total trajectory time τ to 100, we perform the analysis for various

values of τ.

Fig. 10 illustrates the Log Loss obtained on the test set for each method as a function of the total trajectory time τ. As expected, all models improve with increasing τ. Longer trajectories provide more information about the particle's dynamics, particularly for identifying processes like active noise, which are more evident over timescales comparable to or exceeding the characteristic time length τ_a.

Fig. 10 highlights an important trend: as the total trajectory length increases and more information is present for the classifiers, the efficiency of both the LSTM and the analytical classifier improves exponentially, while the feedforward neural network only shows linear improvement. This suggests that incorporating memory effects, as the LSTM does, significantly enhances classification performance as the processing time increases.

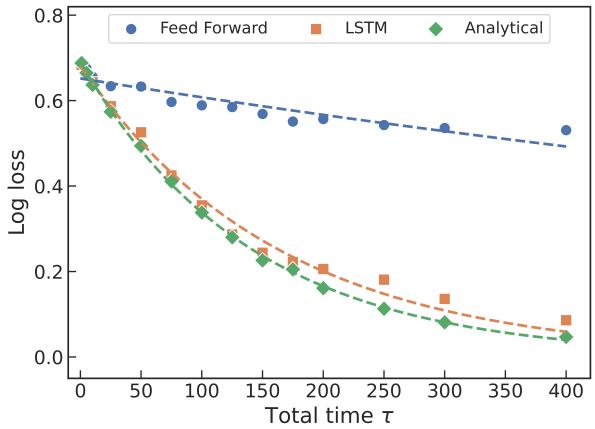


Figure 10: Log Loss on the test set for the three methods across different total trajectory times τ, along with the best-fitting exponential curves of the form $ae^{b\tau}$ for each method.

5 Experimental Cellular Data

In this chapter, we analyze real trajectory data from a experimental measurement of the motion of a living cell within a dense environment of other cells. We apply a LSTM network, just as the one used in the previous chapter, to classify whether the trajectory is influenced by active noise or not. Additionally, we extend the approach by training a separate LSTM to predict the specific values of the diffusion coefficients D and D_a that best characterize the data.

5.1 Data Analysis

We begin by examining the cellular trajectory data, which consists of a time series $x(t)$ recorded with a time step of $\Delta t = 1.28 \times 10^{-5}\,\mathrm{s}$, spanning a total of 655076 steps. This time series is measured by following the motion of a cell using optical tweezers.

To explore the statistical properties of the data, we first plot a histogram of the position x, as shown in Fig. 11. The histogram closely resembles a Gaussian distribution, which we confirm quantitatively using the Shapiro-Wilk test of normality [43], yielding a value of 0.9997. This Gaussian nature suggests that the particle is likely subject to a harmonic potential, so we will model it as an AOUP under a harmonic potential (Eq. 25).

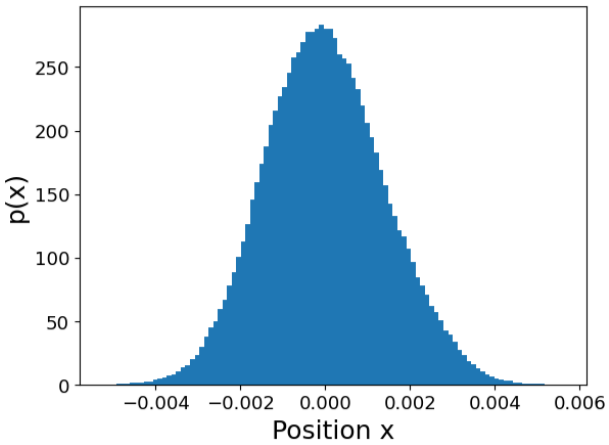


Figure 11: Histogram of the position x (in mm) for the cellular data.

Before we can classify if the trajectory is active or not, we need some information about the 4 parameters $k/\gamma, \tau_a, D, D_a$ that define an AOUP in a harmonic potential (Eq. 25), so that we can approximate a range of values for them.

With these ranges, we can generate similar trajectories and use them to train the neural network. The way we will get these approximate values is by comparing the computed correlation $\langle x(t)x(t+t')\rangle$ of the trajectory to the theoretical result of this correlation for an AOUP under a harmonic potential.

Correlation

The analytical expression for the correlation function in a harmonic potential $U(x) = \frac{1}{2}kx^2$, assuming steady-state conditions, is derived in Appendix C. It is given by:

$$\langle x(t)x(t+t')\rangle = \left(\frac{\gamma D}{k} + \frac{\gamma^3 D_a}{k(\gamma^2 - k^2\tau_a^2)}\right) e^{-kt'} - \frac{\gamma^2 D_a \tau_a}{\gamma^2 - k^2\tau_a^2}\, e^{-t'/\tau_a}. \tag{30}$$

To calculate the correlations $\langle x(t)x(t+t')\rangle$ from the real trajectory data, we fix a value of t', compute the product $x(t)x(t+t')$ at multiple time points t along the trajectory, and then average these products. Repeating this for various values of t', we obtain the empirical correlation function, shown in Fig. 12. The figure also includes a fit of the theoretical result from Eq. 30 to the calculated data.

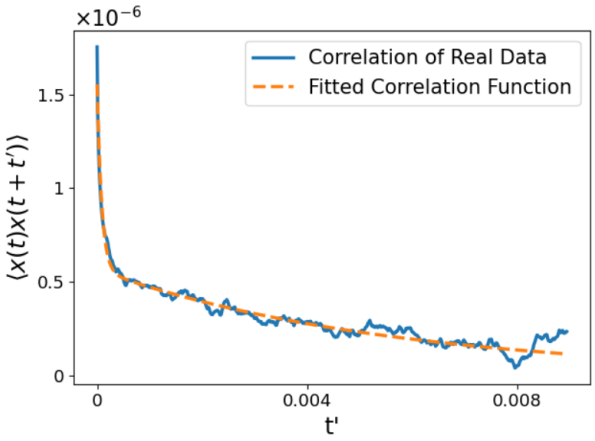


Figure 12: Correlation $\langle x(t)x(t+t')\rangle$ of real data and the result of fitting Eq. 30 to this data.

The fitting process yields an approximation of the system's four defining parameters: k/γ, τ_a, D, and D_a:

$$\begin{aligned} k/\gamma &= 157 \pm 7 s^{-1}, \\ \tau_a &= 0.33 \pm 0.011 s, \\ D_a &= (6.3 \pm 0.5) \times 10^{-3}\ mm^2 s^{-1}, \\ D &= (7 \pm 1.2) \times 10^{-4}\ mm^2 s^{-1}. \end{aligned} \tag{31}$$

This gives us an idea of the value of the parameters that could define the system. As a check, we can compute D in a different way, since for an overdamped Langevin equation it can be approximated as $D \simeq \frac{\langle \Delta x^2 \rangle}{2\Delta t}$ [22]. Doing this approximation for our data, we get a result of $0.00055 mm^2 s^{-1}$, reasonably close to the value obtain by fitting the correlation.

5.2 Classifying Active Noise

Training Data

In this section, we train a LSTM to classify whether a given trajectory contains active noise, i.e., whether the diffusion coefficient D_a is nonzero. We will train the LSTM with trajectories similar to the cellular data, so that at the end we can use the model on said data.

To create the training data, we generate trajectories of an AOUP under a harmonic potential, with random values for k, τ_a, and D chosen to be within a range based on the parameters obtained in Eq. 31. Some of these trajectories are generated with $D_a = 0$, while others have a non-zero value of D_a. In particular, the values of k/γ are chosen uniformly at random from $[50 s^{-1}, 250 s^{-1}]$, the values of τ_a from $[0.05s, 0.8s]$, D from $[10^{-6} mm^2 s^{-1}, 0.002 mm^2 s^{-1}]$ and D_a from $[10^{-6} mm^2 s^{-1}, 0.01 mm^2 s^{-1}]$.

We generate 5000 trajectories with $D_a = 0$ and 5000 trajectories with $D_a > 0$ (randomly selected from the already mentioned range). Each trajectory consists of 5000 time steps (after waiting for an initial 5000 time steps to make sure they are already at steady state). The LSTM is then trained to predict whether a given trajectory has $D_a > 0$ or $D_a = 0$. After training, we test the model on 2000 test trajectories to assess its ability to classify whether D_a is non-zero. The results are shown in Fig. 13.

As seen in Fig. 13, the model performs well in distinguishing between trajectories with and without active noise, demonstrating its ability to correctly identify whether D_a is non-zero. For trajectories without active noise, the model gives on average a probability of 0.046 and it correctly classifies 97.33% of them (gives them a probability below 0.5). For trajectories with active noise, the model gives on average a probability of 0.941 and it correctly classifies 94.33% of them (gives them a probability above 0.5).

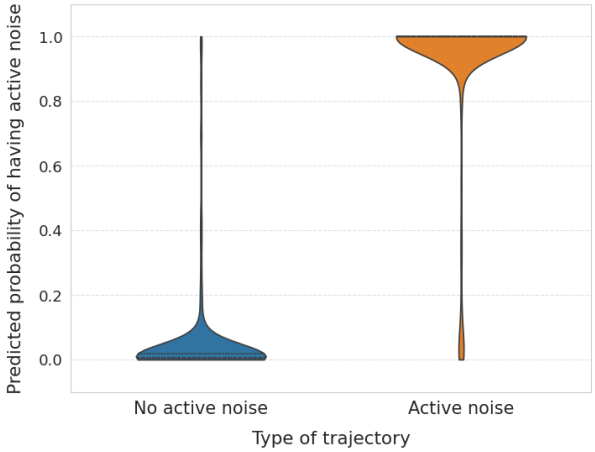


Figure 13: Results of the LSTM model on the test set to classify whether a trajectory has $D_a = 0$ (no active noise) or $D_a > 0$ (active noise). The left side (blue) is for trajectories without active noise and the right side (orange) for trajectories with active noise. On both sides, the results of the LSTM are shown by violin plots, with the vertical axis indicating its output, interpreted as the predicted probability of having active noise.

Application to Experimental Data

We then apply the trained model to the experimental data. The model predicts a score of 0.997, indicating a high confidence that the trajectory exhibits active noise.

5.3 Type of Active Noise

So far, we have always been using the AOUP model, in which the active noise is given by an Ornstein-Uhlenbeck process as described in Eq. 18. However, we now consider the possibility of having a different type of process, known as Rotational Brownian Particle (RBP). In that case, the active noise variable η is defined by $\eta = \cos(\theta(t))$, where $\theta(t)$ evolves by:

$$\dot{\theta} = \sqrt{2D_{rot}}\,\zeta(t),$$

with D_{rot} a constant and $\zeta(t)$ as always a Gaussian white noise with delta correlation.

To see if Rotational Brownian noise fits better our data than the Ornstein-Uhlenbeck process, we will train a LSTM to differentiate between these two types of active noises and then apply it to the experimental data. First, as we did before for an AOUP, we need to construct the train trajectories of RBP

with random values of the parameters k, D_{rot}, D, D_a. To get ranges for these parameters, we start by using the correlation function for a RBP system in a harmonic potential, which is [12]:

$$\langle x(t)x(t')\rangle = \left(\frac{D}{k_\gamma} + \frac{D_a D_{rot}}{k_\gamma(D_{rot}^2 - k^2)}\right) e^{-k_\gamma|t-t'|} - \frac{D_a}{D_{rot}^2 - k_\gamma^2} e^{-D_{rot}|t-t'|}. \quad (32)$$

Next, we fit this expression to the result of the correlation function for the experimental trajectory. Since the expression has the same shape as the one in Eq. 30, the fit is actually the same, but yielding now the following parameters: $k/\gamma = 157 \pm 7s^{-1}$, $D_{rot} = 3.0 \pm 0.05$, $D_a = 0.021 \pm 0.002\ mm^2s^{-1}$, and $D = (7 \pm 1.2) \times 10^{-4}\ mm^2s^{-1}$.

Given this, we create the training set with 5000 trajectories with random values of k, D_{rot}, D, D_a taken respectively from $[50s^{-1}, 250s^{-1}], [1s^{-1}, 10s^{-1}], [10 - 6mm^2s^{-1}, 0.002mm^2s^{-1}]$ and $[10^{-6}mm^2s^{-1}, 0.01mm^2s^{-1}]$. With these trajectories and the ones we have for an AOUP, we train the LSTM to distinguish between RBP and AOUP. AOUP trajectories are labeled with 0 and RBP trajectories are labeled with 1, so that the output of the LSTM is a number between 0 and 1 representing the confidence of the network in identifying the type of active noise present.

The result of applying the trained LSTM to 2000 randomly selected test trajectories is shown in Fig. 14. We can see that the model performs well in distinguishing these types of noises. On average, it gives a value of 0.022 to trajectories with Ornstein-Uhlenbeck noise and 0.985 for Rotational Brownian noise. Furthermore, it correctly classifies 98.2% of the Ornstein-Uhlenbeck trajectories and 98.3% of the Rotational Brownian trajectories.

Application to Experimental Data

We then apply the trained model to the real data and it outputs a value 0.017. This means that choosing between Rotational Brownian noise and Ornstein-Uhlenbeck active noise, the model gives a 98.3% probability that the type of active noise of the system is Ornstein-Uhlenbeck.

5.4 Predicting Diffusion Coefficients

In this section we attempt to use a LSTM to predict the actual values of the diffusion coefficients D and D_a, rather than only classifying whether D_a is zero or greater than zero.

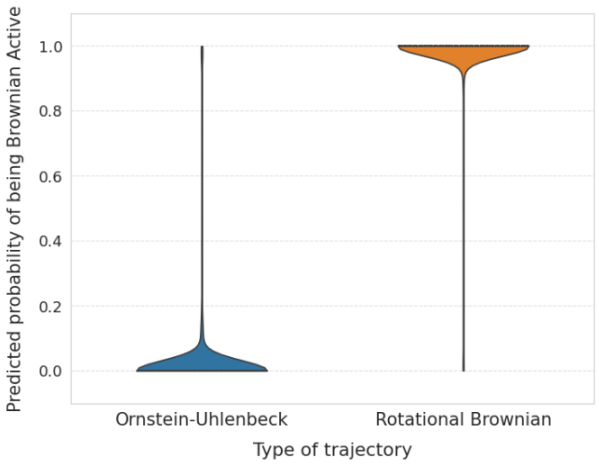


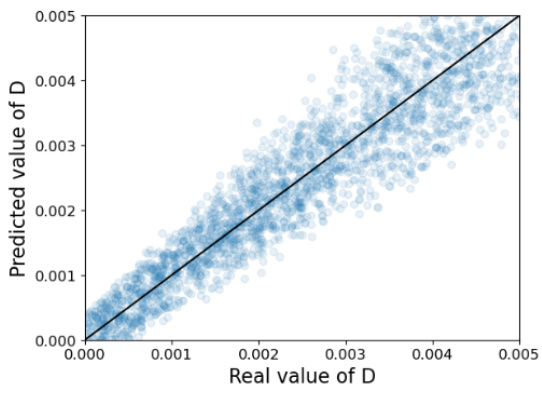
Figure 14: Results of the LSTM model on the test set to classify whether a trajectory is an AOUP or RBP. The left side (blue) is for AOUP trajectories, while the right side (orange) for RBP trajectories. On both sides, the results of the LSTM are shown by violin plots, with the vertical axis representing the output, interpreted as the predicted probability of having Rotational Brownian noise instead of Ornstein-Uhlenbeck.

Training and Testing

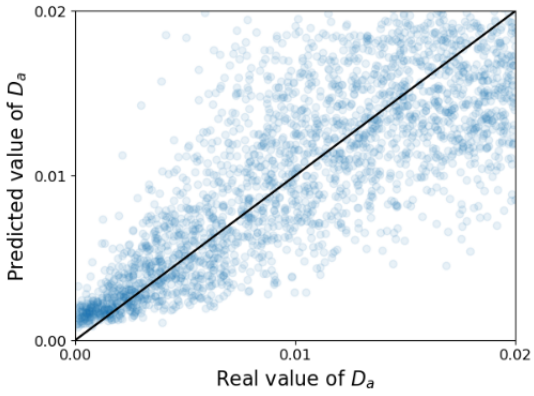
We generate $10,000$ AOUP trajectories, with k/γ randomly chosen in the range $[50s^{-1}, 250s^{-1}]$, and τ_a in the range $[0.05s, 0.8s]$. Furthermore, the trajectories have varying values of D chosen from the range $[10^{-4}mm^2s^{-1}, 0.005mm^2s^{-1}]$ and D_a from $[10^{-4}mm^2s^{-1}, 0.02mm^2s^{-1}]$. We then build and train a LSTM to predict D and a different LSTM to predict D_a from a given trajectory. These LSTMs are trained with the objective of minimizing the mean square error between the real values of D (or D_a) and the predicted values.

The results on the test set of simulated trajectories are shown in Fig. 15, where the predicted values of D and D_a are plotted against the true values. As shown in the figure, the model performs well for predicting D, with the predicted values closely matching the true values. However, predicting D_a is more challenging, and while the model's predictions are not as accurate, they are still reasonably good.

This discrepancy is to be expected, as D can be directly estimated from trajectory data using $D \simeq \frac{\langle \Delta x^2 \rangle}{2\Delta t}$ [22], a straightforward quantity for the network to learn. In contrast, D_a lacks a simple empirical estimation and requires the model to account for memory effects, making its prediction more challenging.



(a) Results for D

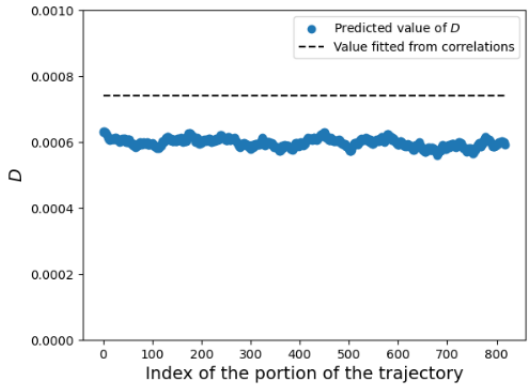


(b) Results for D_a

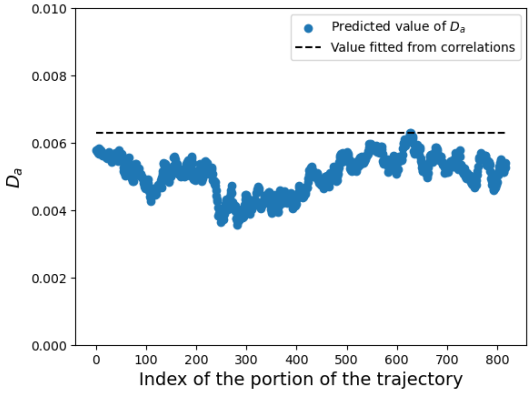
Figure 15: Results of the LSTM model on the test set for predicting the values of D and D_a. The graphs show the predicted values against the true values for each parameter.

Application to Real Data

We now apply the LSTMs to predict the values of D and D_a from the experimental data. To do this, we divide the long trajectory into 818 segments, each containing 800 time steps, and apply the model to each segment. The results are shown in Fig. 16, where we compare the predicted values to those obtained in Eq. 31 by fitting the data to the analytical correlation model.



(a) Results for D



(b) Results for D_a

Figure 16: Results for the predicted D and D_a on the experimental data split into many portions of the trajectory. Also included are the values of D_a and D obtained by fitting to the correlations.

The predicted values of D and D_a are consistent across the different trajectory segments and are close to the values obtained from fitting the correlation data.

5.5 Power Spectrum Analysis

Next, we change focus and conduct an analysis based on the method outlined in [27], which uses the spectral density of an observable $x(t)$ to determine whether a system is out of equilibrium.

The first step is to define the finite-time Fourier transform of the time series $x(t)$ over a measurement interval $[0, \tau]$:

$$\hat{x}_\tau(\omega) = \int_0^\tau dt e^{i\omega t} x(t).$$

The power spectral density (PSD) is then given by the long-time limit of the variance of $\hat{x}_\tau(\omega)$, defined as:

$$S^x(\omega) = \lim_{\tau\to\infty} \left(\frac{1}{\tau} \left(\langle |\hat{x}_\tau(\omega)|^2 \rangle - |\langle \hat{x}_\tau(\omega) \rangle|^2 \right) \right).$$

The quantity $\omega^2 S^x(\omega)$ is then computed. At large frequencies, this quantity converges to a constant value, which characterizes the short-time fluctuations of the observable. As shown in [27], for systems in equilibrium, $\omega^2 S^x(\omega)$ monotonically approaches this constant value from below as ω increases. However, for systems out of equilibrium, there is no such rule, it might approach the constant value monotonically or it might have a bump before reaching it.

The results of calculating $\omega^2 S^x(\omega)$ for the experimental data $x(t)$ are shown in Fig. 17. Since we have only one trajectory, but we need to compute averages, we first divide it into 655 segments, each of length 1000, and take the Fourier transform of each segment. This allows us to compute the ensemble average of the Fourier transforms, which is required for the calculation.

We see from the figure that there is a noticeable bump in $\omega^2 S^x(\omega)$ before it goes back down to its large frequency value. According to [27], this means that the trajectory is out of equilibrium, since an equilibrium trajectory would grow monotonously. This agrees with the results we've got before about the system being out of equilibrium.

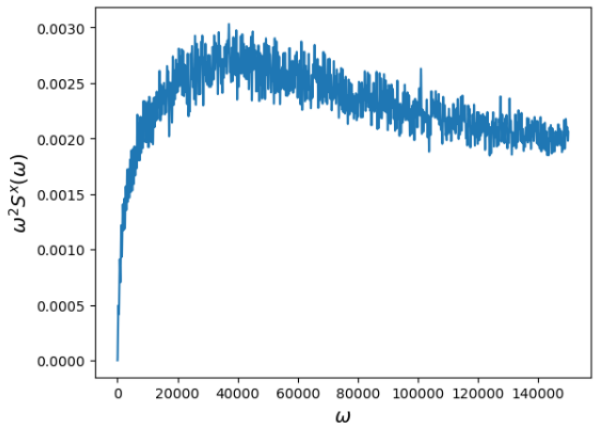


Figure 17: Result of $\omega^2 S^x(\omega)$ for the experimental cellular trajectory.

6 Maximum Likelihood to Predict the Force on a Stochastic System

In this chapter, we turn our attention to a related but distinct problem from those explored earlier. Our approach is inspired by the recent work of [22], in which the authors propose a method for inferring the shape of the conservative force acting on a stochastic system. In that work, they consider a system modeled by a Langevin equation without active noise:

$$\dot{x}_t = \frac{1}{\gamma} f(x_t) + \sqrt{2D}\xi(t),$$

and their goal is to estimate the force field $f(x_t)$ from an observed trajectory $\overline{x} = \{x_0, x_1, x_2, \cdots, x_N\}$.

The method they follow in [22] begins by expressing the force $f(x_t)$ as a linear combination of basis functions $\mathcal{B} = \{b_i(x_t)\}$ for $i = 1$ to n, with coefficients F_i, such that

$$f(x_t) = \sum_{i=1}^{n} F_i b_i(x_t).$$

They then analytically derive the likelihood of observing the trajectory $\overline{x}$ as a function of the coefficients $\mathbf{F} = \{F_i\}$, and find the values that maximize it. This leads to an explicit expression for the optimal coefficients:

$$F_i = \sum_j \frac{\tau}{2D} G_{ij}^{-1} \left\langle b_i(x_t) \frac{\Delta x_t}{\Delta t} \right\rangle_t, \tag{33}$$

where the time average is defined as $\langle \# \rangle_t = \frac{1}{\tau} \sum_t \# \Delta t$ (the summation is done step by step over the whole trajectory). The matrix G_{ij} is given by:

$$G_{ij} = \frac{\tau}{2D} \langle b_i(x_t) b_j(x_t) \rangle.$$

In the rest of this chapter, we extend this method to systems with active noise. We then apply the generalized version to both simulated trajectories and the cellular trajectory analyzed in the previous chapter. Also, in [22], they investigate the best way to select the basis $\mathcal{B} = \{b_i(x_t)\}_{i=1,\cdots,n}$, but for our purposes, we will not go deeply into this.

6.1 Problem Statement

We now consider a system influenced by both thermal and active noise, described by the following dynamics:

$$\begin{aligned} \dot{x}_t &= \frac{1}{\gamma} f(x_t) + \sqrt{2D_a}\eta_t + \sqrt{2D}\xi(t), \\ \dot{\eta}_t &= -\frac{1}{\tau_a}\eta_t + \frac{1}{\tau_a}\zeta(t). \end{aligned}$$

Then, the problem is the following: given a trajectory $\overline{x} = \{x_0, x_1, \cdots, x_N\}$, our goal is to estimate the conservative force $f(x_t)$ acting on the particle.

As in [22], we begin by writing the force as a general linear combination of basis functions $b_i(x_t)$:

$$f(x_t) = \sum_{i=1}^{n} F_i b_i(x_t),$$

and aim to find the coefficients F_i that best explain the observed trajectory. To do this, we will generalize the maximum likelihood approach from [22]: first computing the Log Likelihood of observing the trajectory as a function of the F_i, and then finding the values that maximize it.

6.2 Calculation and Maximization of the Log Likelihood

First, we want to calculate the likelihood of observing a trajectory as a function of the force coefficients $F_1, F_2, \cdots, F_n$. As shown in Appendix B, the probability

of observing a given trajectory $p(\underline{x})$, starting from an initial point x_0 is:

$$p(\underline{x}|x_0) = C \exp\Bigg(- \int_0^\tau dt \left[\frac{(\dot{x}_t - f_t/\gamma)^2}{4D} + \frac{1}{2\gamma}\frac{\partial f_t}{\partial x} \right] + \frac{D_a}{4D^2} \int_0^\tau dt \int_0^\tau dt' (\dot{x}_t - f_t/\gamma)\Gamma_\tau(t,t')(\dot{x}_{t'} - f_{t'}/\gamma) \Bigg).$$

Here, $f_t = f(x(t))$ and $f_{t'} = f(x(t'))$. The normalization constant C is:

$$C = L^{1/4} \left(\frac{1}{\sqrt{4\pi D \Delta t}} \right)^N e^{\tau k_-/(2\tau_a)} \frac{2}{\sqrt{4\sqrt{L} + (1-\sqrt{L})^2(1-\rho^2)}},$$

with $L = 1 + D_a/D$, $k_\pm = 1 \pm \sqrt{L}$, $\rho = e^{-\sqrt{L}\tau/\tau_a}$ and

$$\Gamma_\tau(t,t') = \left(\frac{1}{2\tau_a\sqrt{L}} \right) \frac{k_+^2 e^{-\sqrt{L}|t-t'|/\tau_a} + k_-^2 e^{-\sqrt{L}(2\tau - |t-t'|)/\tau_a} - k_+k_- \left[e^{-\sqrt{L}(t+t')/\tau_a} + e^{-\sqrt{L}(2\tau - t - t')/\tau_a} \right]}{k_+^2 - k_-^2 e^{-2\sqrt{L}\tau/\tau_a}}.$$

So, this gives us the probability of observing a trajectory as a function of the force values $f(x_t)$, and therefore as a function of the coefficients F_i.

Before we maximize this with respect to F_i, we take the logarithm of $p(\underline{x}|x_0)$ to obtain the Log Likelihood, which simplifies the maximization without changing the result. Since we are only interested in how the Log Likelihood depends on the force, we can ignore terms that do not include f. This gives:

$$\mathcal{L}[\underline{x}|\mathbf{F}] = - \int_0^\tau dt \left[\frac{(\dot{x}_t - f_t/\gamma)^2}{4D} \right] + \frac{D_a}{4D^2} \int_0^\tau dt \int_0^\tau dt' (\dot{x}_t - f_t/\gamma)\Gamma_\tau(t,t')(\dot{x}_{t'} - f_{t'}/\gamma).$$

We then expand this expression and substitute the force $f(x_t)$ as a linear combination of basis functions $f(x_t) = \sum_i F_i b_i(x)$. This leads to:

$$\begin{aligned}\mathcal{L}[\underline{x}|\mathbf{F}] = {} & \frac{1}{2D\gamma} \sum_{i=1}^n \int_0^\tau F_i b_i(x_t)\dot{x}_t dt \; - \; \frac{1}{4D\gamma^2} \sum_{i=1}^n \sum_{j=1}^n \int_0^\tau F_i F_j b_i(x_t) b_j(x_t) dt \; + \; \frac{D_a}{4D^2} \int_0^\tau \int_0^\tau \dot{x}_t \dot{x}_{t'} \Gamma_\tau(t,t') dt dt' \\ & - \frac{D_a}{4D^2\gamma} \sum_{i=1}^n \int_0^\tau \int_0^\tau F_i b_i(x_t) \Gamma_\tau(t,t') \dot{x}_{t'} dt dt' - \frac{D_a}{4D^2\gamma} \sum_{i=1}^n \int_0^\tau \int_0^\tau F_i b_i(x_{t'}) \Gamma_\tau(t,t') \dot{x}_t \dot{x}_{t'} dt dt' \\ & + \frac{D_a}{4D^2\gamma^2} \sum_{i=1}^n \sum_{j=1}^n \int_0^\tau \int_0^\tau F_i F_j b_i(x_t) b_j(x_{t'}) \Gamma_\tau(t,t') dt dt'.\end{aligned}$$

Maximizing the Log Likelihood

To find the optimal coefficients F_i, we take the derivative of the Log Likelihood with respect to F_i and set it equal to zero:

$$0 = \frac{\partial \mathcal{L}}{\partial F_i} = \frac{\tau}{2D\gamma} \langle b_i(x_t)\dot{x}_t \rangle_t - \frac{\tau}{2D\gamma^2} \sum_{j=1}^{n} F_j \langle b_i(x_t) b_j(x_t) \rangle_t - \frac{\tau^2 D_a}{4D^2\gamma} \langle b_i(x_t)\Gamma_\tau(t,t')\dot{x}_{t'} \rangle_{t,t'}$$
$$- \frac{\tau^2 D_a}{4D^2\gamma} \langle b_i(x_{t'})\Gamma_\tau(t,t')\dot{x}_t \rangle_{t,t'} + \frac{\tau^2 D_a}{2D^2\gamma^2} \sum_{j=1}^{n} F_j \langle b_i(x_t) b_j(x_{t'})\Gamma_\tau(t,t') \rangle_{t,t'} .$$

Here the means are calculated as $\langle \# \rangle_t = \frac{1}{\tau} \int_0^\tau \# \, dt$, and $\langle \# \rangle_{t,t'} = \frac{1}{\tau^2} \int_0^\tau \int_0^\tau \# \, dt dt'$.

This equation can be solved for $\mathbf{F} = (F_1, F_2, \cdots, F_n)$ by realizing that it has the shape $G\mathbf{F} = \mathbf{C}$, so that the solution is:

$$\mathbf{F} = G^{-1} \, \mathbf{C}, \tag{34}$$

where $\mathbf{C}$ is a vector and G is a matrix given by:

$$\mathbf{C}_i = \frac{\tau}{2D} \langle b_i(x_t)\dot{x}_t \rangle_t - \frac{\tau^2 D_a}{4D^2} \langle b_i(x_t)\Gamma_\tau(t,t')\dot{x}_{t'} \rangle_{t,t'} - \frac{\tau^2 D_a}{4D^2} \langle b_i(x_{t'})\Gamma_\tau(t,t')\dot{x}_t \rangle_{t,t'} ,$$
$$G_{ij} = \frac{\tau}{2D\gamma} \langle b_i(x_t) b_j(x_t) \rangle_t - \frac{\tau^2 D_a}{2D^2\gamma} \langle b_i(x_t) b_j(x'_t)\Gamma_\tau(t,t') \rangle_{t,t'} .$$

This gives us a complete solution. Given a trajectory $\overline{x}$, and a set of basis functions $b_i(x_t)$, we can use this solution to calculate the coefficients F_i. Then, our best estimate for the conservative force acting on the system is: $f(x_t) = \sum_{i=1}^{n} F_i b_i(x_t)$.

Note that for this method to work, we need to know beforehand the values of D_a, τ_a and D, so that we can use them in the definition for $\mathbf{C}$ and $\mathbf{G}$.

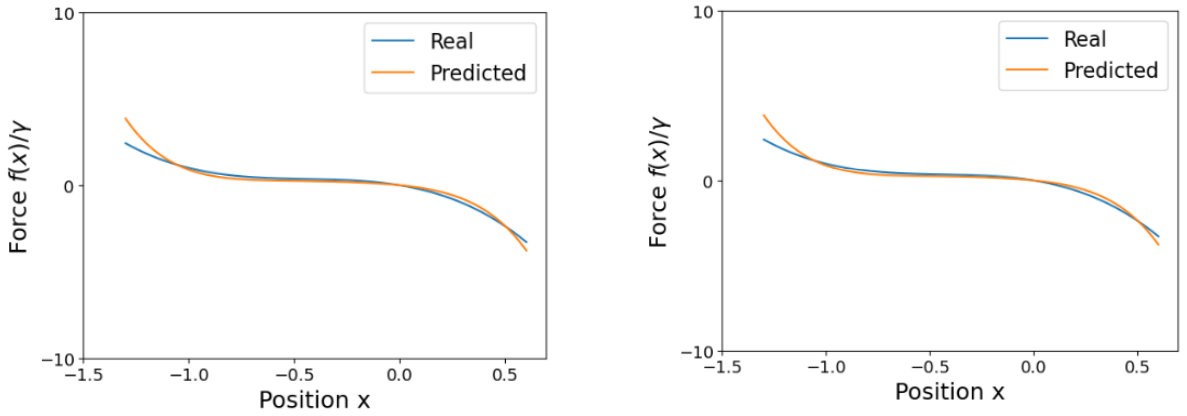
6.3 Results on Simulated Data

We now apply our method to estimate the conservative force acting on simulated trajectories. We begin with a trajectory generated using the following parameters: $D = 0.1$, $D_a = 0.1$, $\tau_a = 1$, and a force given by $f(x) = -3x^3 - 4x^2 - 2x$. We simulate this system for a total duration $\tau = 100$, with a time step of $\Delta t = 0.01$.

To test our method, we ain to recover the force $f(x)$ using maximum likelihood estimation. We start by selecting a basis $\mathcal{B} = \{b_1(x), \cdots, b_n(x)\}$, in this case,

we choose $\{1, x, x^2, x^3, x^4, x^5\}$. This basis can express the true force, but includes additional terms to better reflect the realistic scenario in which the force is not known a priori.

Assuming the force can be expressed as $f(x) = \sum_{i=1}^{6} F_i b_i(x)$, we estimate the coefficients F_i ussing two approaches: first, using Eq. 33 (the method from [22], which doesn't account for active noise), and second, using our modified equation, Eq. 34, which accounts for active noise. In both cases, we input the true value of D, and for the second approach, also the values of D_a, and τ_a, as they are requires to evaluate the Log Likelihood.



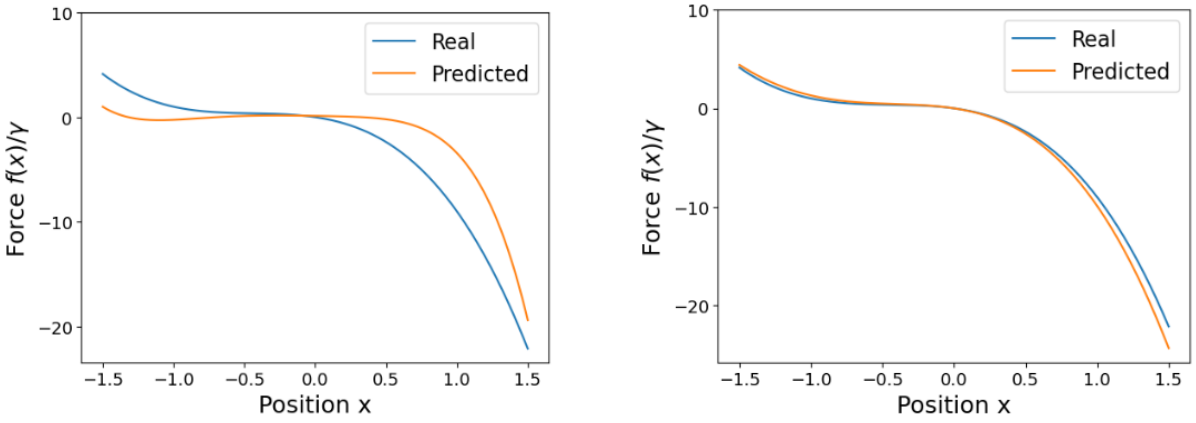
(a) Results with Eq. 33

(b) Results with Eq. 34

Figure 18: Results of the predicted and real force $f(x)$ for a trajectory with $D = 0.1$, $D_a = 0.1$, $\tau_a = 1$ and $f(x) = -3x^3 - 4x^2 - 2x$. Figure a) shows the result of using Eq. 33 (the method from [22], not accounting for active noise), and figure b) shows the result of our modified Eq. 34, which accounts for active noise.

We can see in Fig. 18 that both the standard method and our modified approach yield similar results. This suggests that, in this case, the contribution of active noise is small—possibly because D_a is not large enough for its effects to be significant.

To more clearly evaluate the benefit of incorporating active noise into the inference, we repeat the experiment with a larger value of $D_a = 5$, while keeping all other parameters unchanged. The results are shown in Fig. 19. In this case, the method which accounts for active noise produces a significantly better approximation of the true force. This highlights the importance of including active noise in the Log Likelihood when it plays an important role in the dynamics.



(a) Results with Eq. 33

(b) Results with Eq. 34

Figure 19: Results of the predicted and real force $f(x)$ for a trajectory with $D = 0.1$, $D_a = 5$, $\tau_a = 1$ and $f(x) = -3x^3 - 4x^2 - 2x$. Figure a) shows the result of using Eq. 33 (the method from [22], not accounting for active noise), and figure b) shows the result of our modified Eq. 34, which accounts for active noise.

6.4 Application to Cellular Data

We now apply our maximum likelihood method to estimate the conservative force acting on the cellular trajectory analyzed in Chapter5. Based on the results from fitting the velocity and position correlations using Eq.31, we previously inferred that the force acting on the cell takes the form $\frac{1}{\gamma}f(x) = -157x$.

Using the same trajectory, we now estimate the force $f(x)$ through the maximum likelihood approach described in the previous section. To do this, we select a polynomial basis $\{1, x, x^2, x^3\}$ for $f(x)$ and apply Eq. 34, which accounts for active noise. The values of D, D_a and τ_a were taken from the previous correlation fit.

The result is shown in Fig. 20, where we plot the force obtained from the maximum likelihood inference alongside the previously inferred force from the correlation analysis. As seen in the figure, the inferred force closely matches the linear force predicted by the correlation method, serving as a consistency check between both methods.

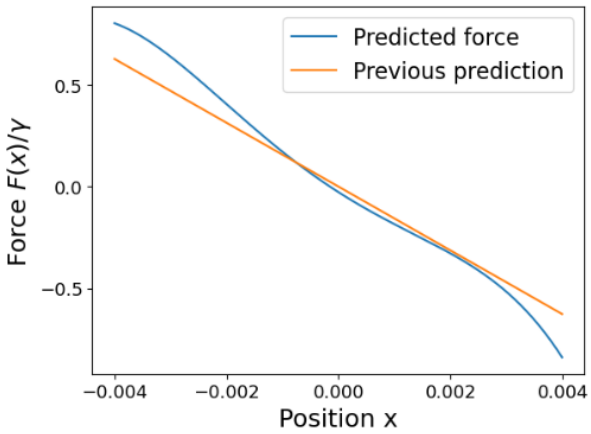


Figure 20: Result of predicting the force $f(x)$ for the cellular data trajectory, along with the force we got in Eq. 31 by fitting the correlation data.

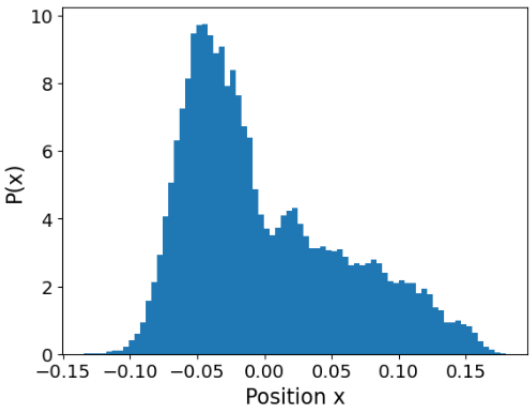
7 Cellular Nuclei Data

In this chapter, we perform a similar analysis to the one we did on cellular data in chapter 5, but now applied to experimental data of lipid granules within cellular nuclei of stem cells. That is, we attempt to use a neural network to infer if this system has active noise. As we will see, this dataset presents additional challenges and the prediction of active noise is inconclusive. This may indicate that the system cannot be accurately described by an overdamped Langevin equation, which limits the applicability of our method.

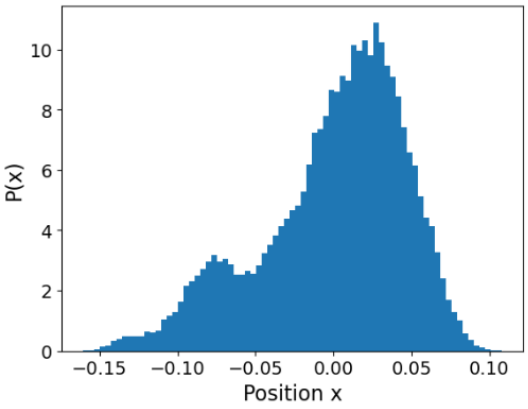
7.1 Data Analysis

We begin by analyzing the data, which consists of 7 different trajectories. Each trajectory has a time step of $\Delta t = 4.5454 \times 10^{-5} s$ and a total of $65,535$ steps. To gain an initial understanding of the system, we examine the probability distributions of the particle positions by plotting histograms of the trajectories (shown in Fig. 21 for the first four cases).

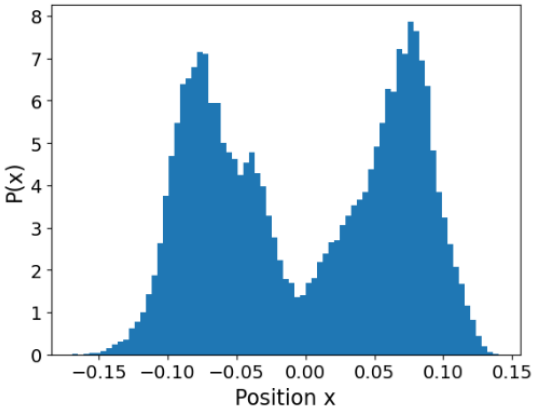
As shown in Fig. 21, the position distributions are clearly non-Gaussian, in contrast with the cellular trajectories analyzed in chapter 5. This prevents us from applying the same analysis methods used with the cellular data, where a clear Gaussian distribution allowed us to assume a harmonic potential and then approximate the coefficients of the system by fitting the correlation data (which can be calculated analytically for a harmonic potential). Moreover, there are considerable differences across the trajectories, suggesting a high degree of vari-



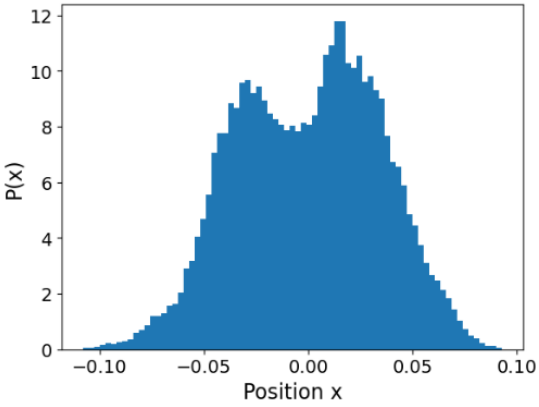
(a) Trajectory 1



(b) Trajectory 2



(c) Trajectory 3



(d) Trajectory 4

Figure 21: Histograms for four different cell nuclei trajectories.

ability in the underlying forces.

Despite this, we would still like to approximate the potential $U(x)$ for each trajectory. This is necessary in order to generate synthetic data with similar characteristics, which will later be used for training the neural network. As an initial model, we assume the potential takes a quartic form:

$$U(x) = k_1 x + k_2 x^2 + k_3 x^3 + k_4 x^4. \tag{35}$$

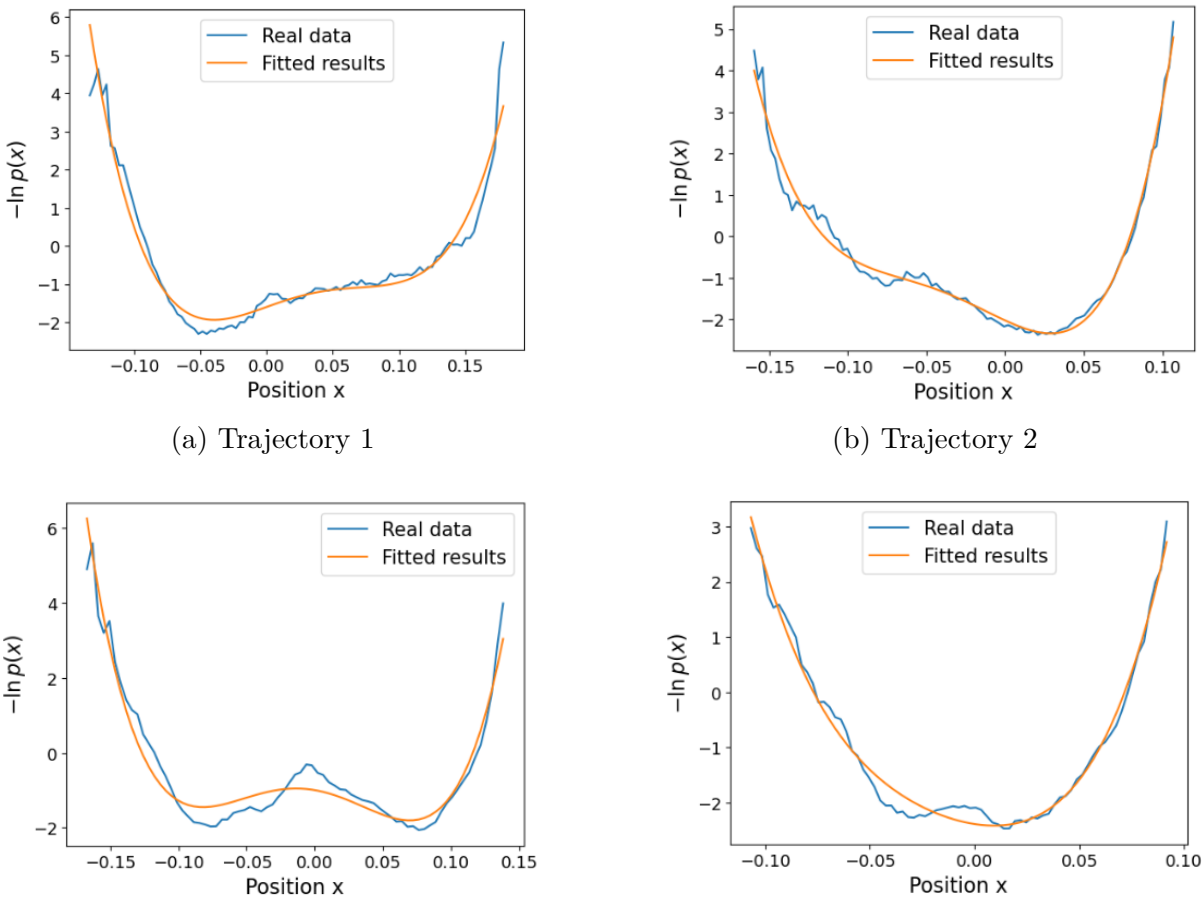
To generate realistic synthetic trajectories, we will eventually need to randomly select values for the coefficients k_1, k_2, k_3, k_4. However, we currently lack a good sense of the appropriate parameter ranges. To address this, we attempt to estimate them from the real data under the assumption that the system is close to equilibrium, which allows us to use Boltzmann's distribution as a proxy for

the steady-state behavior.

According to Boltzmann's distribution, for an equilibrium system we expect $\ln p(x) = -U(x)/k_BT = -U(x)/(\gamma D)$, where at the end we used Einstein's relation $k_BT = \gamma D$. Substituting Eq. 35, we get:

$$-\ln p(x) = \frac{k1}{\gamma D}x + \frac{k2}{\gamma D}x^2 + \frac{k3}{\gamma D}x^3 + \frac{k4}{\gamma D}x^4. \tag{36}$$

We now compute $-\ln p(x)$ directly from the histograms and fit the result to Eq. 36 to estimate the scaled coefficients $\frac{k_i}{\gamma D}$. The resulting fits are shown in Fig. 22 for the first four trajectories.



(a) Trajectory 1

(b) Trajectory 2

(c) Trajectory 3

(d) Trajectory 4

Figure 22: $-\ln p(x)$ calculated directly from the data (blue) for the first four cell nuclei trajectories and the result of fitting Eq. 36 to each one (orange).

This gives us a starting point to define a realistic parameter space for the coefficients k_i, which we can then sample to construct synthetic trajectories for training purposes.

7.2 Neural Network Training

7.2.1 Classifying active noise

Following the approach used in chapter 5 with cellular data, we train a LSTM to distinguish between trajectories generated with and without active noise.

The synthetic trajectories used for training are generated using the AOUP model with a quartic potential. For each trajectory, we randomly select values for the parameters k_1/γ, k_2/γ, k_3/γ, and k_4/γ, within wide ranges chosen around the results obtained from the fitting in the previous section. Additionally, each trajectory is assigned a random value for the diffusion coefficient D, the active diffusion coefficient D_a, and the active noise correlation time τ_a.

We generate a total of 5000 trajectories without active noise (i.e., with $D_a = 0$), and 5000 trajectories with active noise (where D_a is randomly chosen for each trajectory). This setup defines a binary classification problem, where the LSTM must determine whether a given trajectory includes active noise. The network outputs a probability between 0 and 1, representing the confidence that a given trajectory exhibits active noise.

After training the LSTM on this synthetic dataset, we evaluate its performance on a separate test set generate in the same way. The results are shown in Fig. 23. The model assigns a mean probability of 0.267 to trajectories without active noise, and 0.6727 to those with active noise. Out of 2000 test trajectories, 77.6% were correctly classified, based on whether the assigned probability was greater than 0.5 (active) or less than 0.5 (inactive).

7.3 Predictions on real data

We now apply the trained LSTM classifier to the real nuclei data.

Each of the 7 trajectories is divided into 80 sections, and the LSTM is applied to each segment independently. The results are shown in Fig.. 24.

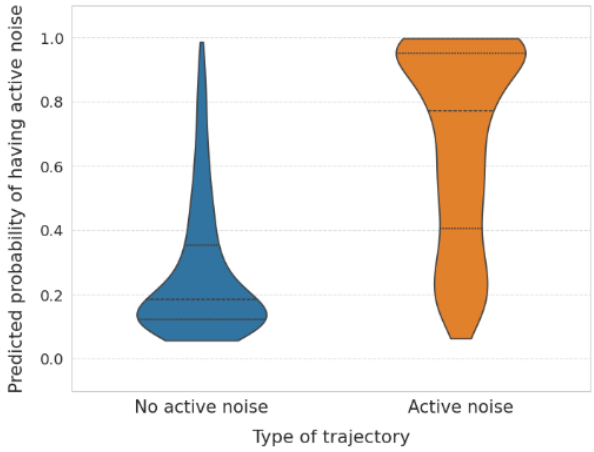


Figure 23: LSTM classification predictions on simulated data of an AOUP with quartic potential. The predictions are shown as violin plots. The right side corresponds to trajectories that have active noise, while the left side to those that don't.

As seen in Fig. 24, the LSTM does not classify the trajectories with high confidence. For most segments, it outputs probabilities close to 0.5, suggesting uncertainty. As mentioned before, this could be due to a limitation of the method, maybe because these experimental trajectories can not be modeled by an overdamped Langevin equation.

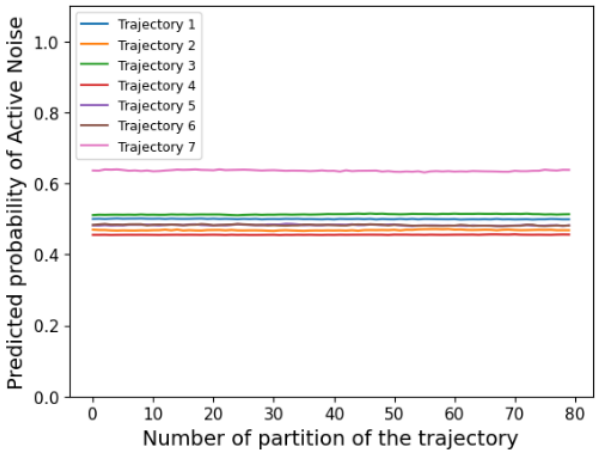


Figure 24: Results of applying the trained LSTM to classify nuclei trajectories as active or inactive. Each trajectory is split into 80 sections, and the LSTM is applied to each section.

It is important to mention that we tried many alternatives to get some conclusions from this data, such as: higher or lower order expansions of $U(x)$, tuning the ranges of values of k_i for generating training data, different neural network

architectures and designs, and using the power spectrum analysis as in section 5.5. These methods failed to give any conclusions about the existence of active noise in the data. A possible solution to this problem in future research could be to generalize what we have done by removing the overdamping assumption in the Langevin equation.

7.4 Power Spectrum Analysis

In this small section we show why the power spectrum analysis method that was introduced in section 5.5 to conclude if a system is in equilibrium or not, also fails in this dataset.

Recall that in the power spectrum analysis, we compute the finite-time Fourier transform of a trajectory, defined as:

$$\hat{x}_\tau(\omega) = \int_0^\tau dt\ e^{i\omega t} x(t).$$

Then, the power spectral density (PSD) is:

$$S^x(\omega) = \lim_{\tau\to\infty} \left(\frac{1}{\tau} \left(\langle |\hat{x}_\tau(\omega)|^2 \rangle - |\langle \hat{x}_\tau(\omega) \rangle|^2 \right) \right),$$

and the quantity $\omega^2 S_\omega$ is analyzed. If the process is in equilibrium, as we increase ω, this quantity grows monotonically until reaching a steady value. For instance, in the case of cellular data (Fig. 17), a bump in $\omega^2 S_\omega$ indicates that the trajectory is out of equilibrium.

However, this method is not applicable to the nuclear data. To understand why, consider that the finite-time Fourier transform is numerically computed as follows:

$$\hat{x}(\omega) \simeq \sum_{n=0}^{\tau/\Delta t} e^{i\omega\ n\Delta t} x(\Delta t\ n)\Delta t.$$

When ω becomes large enough so that $\omega\Delta t = 2\pi$, we have $\hat{x}(\omega) = \hat{x}(0)$. Consequently, the power spectrum $S(\omega)$ is periodic with a period of $\frac{2\pi}{\Delta t}$, and therefore $\omega^2 S_\omega$ will grow without bound as we increase ω, instead of reaching a steady value. This is not a problem if Δt is small enough so that we can explore big values of ω and clearly see the behavior of $\omega^2 S_\omega$ (either monotonously increasing or having a bump before reaching a steady value). The problem is that this is not the case for this data, as Δt is not small enough to get any conclusions from $\omega^2 S_\omega$.

8 Conclusion

Our results demonstrate the effectiveness of machine learning in identifying and characterizing active noise in experimental trajectories. First, we explored the related work of [26], where they use machine learning to approximate the entropy production of stochastic systems. However, we found that said method is insufficient for our aim of identifying active noise, since it is unable to recognize entropy production due to hidden variables (in this case, the active noise variable $\eta(t)$).

Therefore, we changed focus to train neural networks specifically for the task of identifying if a trajectory has active noise or nor. After training a Long Short-Term Memory (LSTM) network on synthetic data, we successfully classified the presence of active noise in a real trajectory of a living cell.

We found that for the particular type of cell observed, its motion is actively driven and best modeled by Ornstein-Uhlenbeck active noise. This finding aligns with previous studies on cellular motility, where active fluctuations often exhibit memory effects and finite correlation times. The ability of our model to distinguish between different types of active noise suggests that machine learning provides a powerful tool for analyzing stochastic processes in biological systems.

Moreover, we used synthetic trajectories with and without active noise to test the effectiveness of our method, and we found that the classification of active noise using a LSTM is almost as effective as the theoretically calculated best possible result. This suggests that the LSTM is capable of extracting important memory effects in the trajectories for detecting active fluctuations.

Then, we studied a method using maximum likelihood to estimate the force being applied on stochastic systems. We expanded said method by including the possibility of active noise and tested it with positive results on simulate data. Moreover, when applied to the same cellular data as before, it gave out results consistent with what we had previously found.

Finally, we tried to apply the methods developed to trajectories of cellular nuclei. However, this proved too challenging, suggesting that they can't be modeled as an overdamped Langevin equation. This brings the clear avenue for future research of relaxing the overdamping assumption.

In summary, our study provides a proof of concept that machine learning, and LSTMs in particular, offer a powerful framework for analyzing non-equilibrium stochastic dynamics in living systems and characterizing active motion. By learning directly from data, these models can uncover hidden structure in noisy trajectories and provide new insights into the nature of active processes.

Appendix A: Computationally solving Langevin's equation

In this appendix we look at how to numerically solve a Langevin equation, such as:

$$\dot{x} = f(x) + \sqrt{2D}\,\xi(t).$$

We start by integrating both sides of the equation from some value t to $t + \Delta t$:

$$\int_t^{t+\Delta t} \dot{x}(s)\,ds = \int_t^{t+\Delta t} f(x(s))\,ds + \sqrt{2D}\int_t^{t+\Delta t} \xi(s)\,ds. \tag{37}$$

The left hand side of Eq. 37 is

$$\int_t^{t+\Delta t} \dot{x}(s)\,ds = x(t+\Delta t) - x(t) := \Delta x. \tag{38}$$

We can integrate the first term of the right hand side of Eq. 37 by using a Taylor series expansion:

$$\int_t^{t+\Delta t} f(x(s))\,ds = f(x(t))\Delta t + \cdots .$$

For the second term on the right hand side of Eq. 37, we define $\Delta W(t) := \int_t^{t+\Delta t} \xi(s)ds$, as we did in Eq. 4. Putting it all together in Eq. 37, we get:

$$x(t+\Delta t) - x(t) = f(x(t))\Delta t + \Delta W(t),$$

and therefore:

$$x(t+\Delta t) = x(t) + f(x(t))\Delta t + \Delta W(t). \tag{39}$$

Eq. 39 tells us how to numerically solve Langevin's equation. We first divide the time range on which we want to solve the equation $[0, \tau]$ into steps of length Δt. We initialize x at time $t = 0$ as some value $x(0) = x_0$. Then, we can calculate $x(\Delta t)$, $x(2\Delta t)$, $x(3\Delta t)$, $\cdots$ by using Eq. 39 repeatedly starting from $x(0) = x_0$.

The only remaining thing needed to use Eq. 39 is to know how to compute $\Delta W(t)$. To do so, as we mentioned around Eq. 4, we assume that $\Delta W(t)$ is a Gaussian random variable. Moreover, from the definition of $\Delta W(t)$ it is easy to see that $\langle \Delta W \rangle = 0$ and $\langle \Delta W^2 \rangle = \Delta t$. Then, along with the Guassian assumption, we know the whole distribution of $\Delta W(t)$. Moreover, it can also be

proven quite easily that if $|t - t'| > \Delta t$, then $\langle W(t)\Delta W(t')\rangle = 0$, which means that the $\Delta W(t)$ at two different time steps are independent.

Therefore, at each time step, the value of $\Delta W(t)$ is obtained by drawing from a Gaussian distribution with mean 0 and variance Δt. This is done again and independently for each time step.

Appendix B: Path Probability Calculation

In this appendix we use path integrals to calculate the probability of observing a specific trajectory $x(t)$ for an active Ornstein-Uhlenbeck particle. We start by proving the Onsager-Machlup integral, which gives the probability density of a trajectory for a stochastic process. Then, we apply it to the specific system of an active Ornstein-Uhlenbeck particle.

Onsager-Machlup Integral

This is a derivation of the Onsager Machlup integral, which is used to calculate the probability that a process described by a Langevin equation follows a given trajectory $x(t)$ from a fixed initial position x_0. For this derivation, we follow what is done in [45]. We start by considering that our system is described by a stochastic differential equation, such as:

$$\dot{x}(t) = a(x(t), t) + b\xi(t),$$

with b a constant and $\xi(t)$ a Gaussian noise satisfying $\langle \xi(t)\xi(t') \rangle = \delta(t - t')$.

The particle follows a stochastic trajectory $x(t)$, during the time interval $t \in [0, \tau]$, and we assume that the initial position of the particle has some fixed value $x(0) = x_0$. We discretize time into N subintervals of length Δt, so that the discretized trajectory is $\overline{x} := \{x_0, x_1, x_2, \cdots, x_N\}$. We also define the part of the trajectory after the initial position as $\underline{x} := \{x_1, x_2, \cdots, x_N\}$.

The quantity we wish to calculate is the probability of having a certain trajectory $\underline{x}$ given the initial position x_0, that is $P(\underline{x}|x_0)$. If we define $t_n = n\Delta t$, the time evolution of the particle step by step can be rewritten as:

$$x_{n+1} = x_n + a(x_n, t_n)\Delta t + b\xi_{\Delta t}(t_n),$$

where $\xi_{\Delta t}(t_n)$ is a random variable with normal distribution, mean 0 and standard deviation $\sqrt{\Delta t}$. Therefore, $\xi_{\Delta t}(t_n)$ has the following probability density:

$$P(\xi_{\Delta t}) = \frac{1}{\sqrt{2\pi\Delta t}} e^{-\xi_{\Delta t}^2/2\Delta t}. \tag{40}$$

With the probability density of $\xi_{\Delta t}$, we can now calculate the path probability of the trajectory $\underline{x}$ as the product of the probabilities of $\xi_{\Delta t}(t_n)$ along the trajectory:

$$P_\xi(\underline{x}|x_0)\mathcal{D}[\xi] = \left(\frac{1}{\sqrt{2\pi\Delta t}}\right)^N \exp\left(-\sum_{n=0}^{N-1}\frac{\xi_{\Delta t}(t_n)^2}{2\Delta t}\right)\mathcal{D}[\xi]$$
$$= \left(\frac{1}{\sqrt{2\pi\Delta t}}\right)^N \exp\left(-\sum_{n=0}^{N-1}\frac{[(\dot{x}(t_n)-a(x_n,t_n))\Delta t]^2}{2b^2\Delta t}\right)\mathcal{D}[\xi], \quad (41)$$

where $D[\xi] := \prod_{n=0}^{N-1} d\xi_{\Delta t}(t_n)$ and $\dot{x}(t_n) := (x_{n+1}-x_n)/\Delta t$.
We now introduce the midpoint position at each step of time as $x'_n := \dfrac{x_n + x_{n+1}}{2}$.
Using this, the time-evolution is expressed as:

$$x_{n+1} = x_n + a(x'_n, t_n)\Delta t + b\xi_{\Delta t}(t_n). \quad (42)$$

We want to transform Eq. 41 into a form that depends on x'_n. First we change the variables in the path probability from $\mathcal{D}[\xi] = \prod_{n=0}^{N-1} d\xi(t_n)$ to $d\Gamma = \prod_{n=0}^{N-1} dx_n$, for which we need to calculate the Jacobian of this transformation. To calculate the Jacobian, we use Eq. 42 to obtain:

$$\frac{\partial \xi_{\Delta t}(t_k)}{\partial x_{n+1}} = \frac{\partial}{\partial x_{n+1}}\left(\frac{\Delta x_n - a(x'_n,t_k)\Delta t)}{b}\right)$$
$$= \frac{1}{b}\left(1 - \frac{1}{2}\frac{\partial}{\partial x'_n}a(x'_n,t_k)\Delta t\right)$$
$$= \frac{1}{b}\exp\left(-\frac{1}{2}\frac{\partial}{\partial x'_n}a(x'_n,t_k)\Delta t\right).$$

The last part is up to order $O(\Delta t)$. Since the Jacobian matrix $\partial(\xi_{\Delta t}(t_0),\cdots,\xi_{\Delta t}(t_{N-1}))/\partial(x_1,\cdots,x_N)$ is a lower triangular matrix, the Jacobian is the product of the diagonal elements $\dfrac{\partial \xi_{\Delta t}(t_n)}{\partial x_{n+1}}$. Therefore, continuing with Eq. 41, we have:

$$P_\xi(\underline{x}|x_0)\mathcal{D}[\xi] = \left(\frac{1}{\sqrt{2\pi\Delta t}}\right)^N \exp\left(-\sum_{n=0}^{N-1}\frac{[(\dot{x}(t_n)-a(x_n,t_n))\Delta t]^2}{2b^2\Delta t}\right)\mathcal{D}[\xi]$$
$$= \left(\frac{1}{\sqrt{2\pi\Delta t}}\right)^N \exp\left(-\sum_{n=0}^{N-1}\frac{[(\dot{x}(t_n)-a(x_n,t_n))\Delta t]^2}{2b^2\Delta t}\right)\frac{1}{b^N}\exp\left(-\sum_{n=0}^{N-1}\frac{1}{2}\frac{\partial}{\partial x'_n}a(x'_n,t_n)\Delta t\right)d\Gamma$$
$$= \left(\frac{1}{b\sqrt{2\pi\Delta t}}\right)^N \exp\left(-\sum_{n=0}^{N-1}\frac{[(\dot{x}(t_n)-a(x_n,t_n))\Delta t]^2}{2b^2\Delta t} + \frac{1}{2}\frac{\partial}{\partial x'_n}a(x'_n,t_n)\Delta t\right)d\Gamma$$
$$= \left(\frac{1}{b\sqrt{2\pi\Delta t}}\right)^N \exp\left(-\Delta t\sum_{n=0}^{N-1}\frac{[(\dot{x}(t_n)-a(x_n,t_n))]^2}{2b^2} + \frac{1}{2}\frac{\partial}{\partial x'_n}a(x'_n,t_n)\right)d\Gamma.$$

Taking the Δt limit, we conclude that:

$$P(\underline{x}|x_0)d\underline{x} = C_0 \exp\left(-\int_0^\tau dt \left[\frac{[(\dot{x}(t_n) - a(x_n, t_n))]^2}{2b^2} + \frac{1}{2}\frac{\partial}{\partial x'_n}a(x'_n, t_n)\right]\right), \quad (43)$$

where $C_0 = \lim_{\Delta t \to 0}\left(\frac{1}{b\sqrt{2\pi\Delta t}}\right)^N$. This result is the Onsager-Machlup integral, which gives us the probability density of observing a given trajectory $\underline{x}$ starting from an initial position x_0.

Calculation of $P(\underline{x}|x_0)$ without active noise

Here we use the Onsager-Machlup integral to calculate the probability of observing a given trajectory for an AOUP without active noise. In said case, the Langevin equation is:

$$\dot{x} = g(x(t), t) + \sqrt{2D}\xi(t),$$

with $g(x(t), t) = \frac{1}{\gamma}f(x(t), t)$ and $\xi(t)$ an unbiased Gaussian white noise such that $\langle \xi(t)\xi(t')\rangle = \delta(t - t')$.

Then, according to the Onsager-Machlup integral in Eq. 43 (with $a \to g$ and $b \to \sqrt{2D}$), we have that:

$$P(\underline{x} \mid x_0) = C_0 \exp\left\{-\int_0^\tau dt \left[\frac{(\dot{x}_t - g_t)^2}{4D} + \frac{1}{2}\frac{\partial g_t}{\partial x}\right]\right\},$$

where the subindex t means that the variable is evaluated at time t, and the normalization constant C_0 is given by

$$C_0 = \lim_{\Delta t \to 0}\left(\frac{1}{\sqrt{4\pi D\Delta t}}\right)^N.$$

Calculation of $P(\underline{x}|x_0)$ with active noise

Now we consider the case of an active Ornstein-Uhlenbeck particle, which is governed by the following equations:

$$\begin{aligned}\dot{x}(t) &= g(x(t), t) + \sqrt{2D_a}\eta(t) + \sqrt{2D}\xi(t)\\ \dot{\eta}(t) &= -\frac{1}{\tau_a}\eta(t) + \frac{1}{\tau_a}\zeta(t),\end{aligned}$$

where $g(x(t),t) := \frac{1}{\gamma} f(x(t),t)$. In this case, since we have active noise, the trajectory of x is non-Markovian and we can't apply directly the Onsager-Machlup integral. However, the evolution of the combined set of variables (x,η) is Markovian and we can find a probability $p(\underline{x},\underline{\eta} \mid x_0,\eta_0)$. Then, we can integrate out the variable η to get a probability density for $x(t)$.

The Onsager-Machlup integral for the two variables (x,η) is the one given in Eq. 43 but now with these two variables instead of only one, so that it becomes:

$$P(\underline{x},\underline{\eta} \mid x_0,\eta_0) = C_1 \exp\left(-\int_0^\tau dt \left[\frac{(\dot{x}_t - g_t - \sqrt{2D_a}\eta_t)^2}{4D} + \frac{(\tau_a\dot{\eta}_t + \eta_t)^2}{2} + \frac{1}{2}\frac{\partial g_t}{\partial x}\right]\right), \tag{44}$$

where the subindex t means that the variable is evaluated at time t and the normalization factor is

$$C_1 = \left(\frac{1}{\sqrt{2\pi\Delta t}}\right)^{2N} \left(\frac{\tau_a}{\sqrt{2D}}\right)^N e^{\tau/(2\tau_a)}.$$

The normalization factor comes from having two of the factors $\frac{1}{\sqrt{2\pi\Delta t}}$ from the Gaussian distribution of both noises. The term $\frac{1}{\sqrt{2D}}$ is the coefficient $1/b$ from the Onsager-Machlup integral for variable x and the factor τ_a comes from the same place but for the η variable. Finally, the factor $e^{\tau/2\tau_\alpha}$ comes from integrating $\exp\left(-\int_0^\tau \frac{1}{2}\frac{\partial}{\partial\eta}\left(-\frac{1}{\tau_a}\eta\right) dt\right)$, which is the last term of the Onsager-Machlup integral for the η variable.

Now we need to integrate out the η, which we do in basically the same way as it is done in [16]. From the law of total probability, we know that we can integrate out η by doing:

$$p(\underline{x}|x_0) = \int \mathcal{D}\underline{\eta}\, d\eta_0\, p(\underline{x},\underline{\eta}|x_0,\eta_0) p_0(\eta_0|x_0). \tag{45}$$

Here $p_0(\eta_0|x_0)$ is the initial distribution of η_0, which may depend on the value of x_0. However, we assume it to be independent from x_0, so that:

$$p_0(\eta_0|x_0) = p_s(\eta_0) = \sqrt{\frac{\tau_a}{\pi}} e^{-\tau_a\eta_0^2}. \tag{46}$$

The distribution for η_0 is a Gaussian with mean 0 and standard deviation of $\frac{1}{\sqrt{2\tau_a}}$, as shown in Eq. 58. The assumption that η_0 is independent from x_0 can come from assuming that $t = 0$ is the moment when the particle is placed into the medium, or that the correlations between x and η are not very significant. Substituting Eq. 46 and Eq.44 into Eq. 45, (and using the abbreviation $\overline{\eta} = \eta_0 \cup \underline{\eta}$), we get:

$$
\begin{aligned}
p(\underline{x}|x_0) &= C_1\sqrt{\frac{\tau_a}{\pi}}\int \mathcal{D}\overline{\eta}\, e^{-\tau_a\eta_0^2}\exp\left(-\int_0^\tau dt\left[\frac{(\dot{x}_t - g_t - \sqrt{2D_a}\eta_t)^2}{4D} + \frac{(\tau_a\dot{\eta}_t + \eta_t)^2}{2} + \frac{1}{2}\frac{\partial g_t}{\partial x}\right]\right)\\
&= C_1'\int \mathcal{D}\overline{\eta}\, e^{-\tau_a\eta_0^2}\exp\left(-\int_0^\tau dt\left[\frac{(\dot{x}_t - g_t)^2 - 2\sqrt{2D_a}(\dot{x}_t - g_t)\eta_t + 2D_a\eta_t^2}{4D} + \frac{\tau_a^2\dot{\eta}_t^2 + 2\tau_a\dot{\eta}_t\eta_t + \eta_t^2}{2} + \frac{1}{2}\frac{\partial g_t}{\partial x}\right]\right)\\
&= C_1'\exp\left(-\int_0^\tau dt\left[\frac{(\dot{x}_t - g_t)^2}{4D} + \frac{1}{2}\frac{\partial g_t}{\partial x}\right]\right)\\
&\qquad\times\int \mathcal{D}\overline{\eta}\, e^{-\tau_a\eta_0^2}\exp\left(-\int_0^\tau dt\left[\frac{-2\sqrt{2D_a}(\dot{x}_t - g_t)\eta_t + 2D_a\eta_t^2}{4D} + \frac{\tau_a^2\dot{\eta}_t^2 + 2\tau_a\dot{\eta}_t\eta_t + \eta_t^2}{2}\right]\right)\\
&= C_1'K\int \mathcal{D}\overline{\eta}\, e^{-\tau_a\eta_0^2}\exp\left(-\int_0^\tau dt\left[\frac{-2\sqrt{2D_a}(\dot{x}_t - g_t)\eta_t + 2D_a\eta_t^2}{4D} + \frac{\tau_a^2\dot{\eta}_t^2 + 2\tau_a\dot{\eta}_t\eta_t + \eta_t^2}{2}\right]\right),
\end{aligned}
$$

where $C_1' = \sqrt{\frac{\tau_a}{\pi}}\, C_1$ and $K = \exp\left(-\int_0^\tau dt\left[\frac{(\dot{x}_t - g_t)^2}{4D} + \frac{1}{2}\frac{\partial g_t}{\partial x}\right]\right)$.
Then, we integrate the term with $\dot{\eta}^2$ using integration by parts:

$$
\begin{aligned}
\int_0^\tau \dot{\eta}^2 dt &= \dot{\eta}\eta\Big|_0^\tau - \int_0^\tau \eta\ddot{\eta}\, dt\\
&= \dot{\eta}_\tau\eta_\tau - \dot{\eta}_0\eta_0 - \int_0^\tau \eta\ddot{\eta}\, dt,
\end{aligned}
$$

and we also integrate the term with $\dot{\eta}_t\eta_t$:

$$
\int_0^\tau \dot{\eta}_t\eta_t dt = \frac{1}{2}\eta_\tau^2 - \frac{1}{2}\eta_0^2
$$

Substituting these results, we get:

$$
\begin{aligned}
p(\underline{x}|x_0) = C_1'K\int \mathcal{D}\overline{\eta}\ \exp\Big(&-\frac{\tau_a\eta_0^2}{2} - \frac{\tau_a^2\dot{\eta}_\tau\eta_\tau}{2} + \frac{\tau_a^2\dot{\eta}_0\eta_0}{2} - \frac{\tau_a\eta_\tau^2}{2}\\
&- \int_0^\tau dt\left[\frac{-2\sqrt{2D_a}(\dot{x}_t - g_t)\eta_t + 2D_a\eta_t^2}{4D} + \frac{-\tau_a^2\eta\ddot{\eta} + \eta_t^2}{2}\right]\Big).
\end{aligned}
$$

This can be rewritten as:

$$
p(\underline{x}|x_0) = C_1'K\int \mathcal{D}\overline{\eta}\ \exp\left(\int_0^\tau dt\frac{\sqrt{2D_a}}{2D}\eta_t(\dot{x}_t - g_t) - \frac{1}{2}\int_0^\tau dt\int_0^\tau dt'\ \eta_t\hat{V}_\tau(t,t')\eta_{t'}\right),
$$

where $\hat{V}$ is a differential operator defined as:

$$\hat{V}_\tau(t,t') = \delta(t-t')\left[-\tau_a^2\partial_t^2 + (1+D_a/D) + \delta(t)(\tau_a - \tau_a^2\partial_t) + \delta(t-\tau)(\tau_a + \tau_a^2\partial_t)\right].$$

Then, the path integral over the active noise paths $\eta(t)$ can be done exactly [16]. The idea is to first complete the square so that we can write the integral as:

$$\int \mathcal{D}\overline{\eta}\exp\left(\frac{1}{2}\int_0^\tau dt\int_0^\tau dt'[w_t^T\Gamma_\tau(t,t')w_{t'} - (\eta_t+\epsilon_t)^T V_\tau(t,t')(\eta_{t'}+\epsilon_{t'})]\right)$$

where we don't know ϵ and Γ yet, but they are used to complete the square, and w is defined as $\dfrac{\sqrt{2D_a}}{2D}(\dot{x}_t - g_t)$. The path integral over noises can be shifted to an integral over $\eta+\epsilon$ instead of η, and this change of variable has the identity as its Jacobian, so that we don't need to multiply by any new factors. Then, the result we are looking for is:

$$\begin{aligned} p(\underline{x}|x_0) &= C_1' K\exp\left(\frac{1}{2}\int_0^\tau dt\int_0^\tau dt'\; w_t^T\Gamma_\tau(t,t')w_{t'}\right)\int\mathcal{D}\overline{\eta}\exp\left(-\frac{1}{2}\int_0^\tau dt\int_0^\tau dt'\;(\eta_t+\epsilon_t)^T V_\tau(t,t')(\eta_{t'}+\epsilon_{t'})\right)\\ &= C_1' KB\exp\left(\frac{1}{2}\int_0^\tau dt\int_0^\tau dt'\; w_t^T\Gamma_\tau(t,t')w_{t'}\right) \end{aligned} \tag{47}$$

where we defined B as:

$$B := \int\mathcal{D}\overline{\eta}\exp\left(-\frac{1}{2}\int_0^\tau dt\int_0^\tau dt'\;(\eta_t+\epsilon_t)^T V_\tau(t,t')(\eta_{t'}+\epsilon_{t'})\right) \tag{48}$$

Calculation of B

We still need to calculate the factor B of Eq.48 so that we can get the complete normalization factor of $p(\underline{x}|x_0)$. To recapitulate, we have the operator

$$\hat{V}_\tau(t,t') = \delta(t-t')[-\tau_a^2\partial_t^2 + L + \delta(t)(\tau_a - \tau_a^2\partial_t) + \delta(t-\tau)(\tau_a+\tau_a^2\partial_t)]$$

where $L := 1 + D_a/D$. And we want to calculate:

$$B = \int\mathcal{D}\overline{\eta}\exp\left(-\frac{1}{2}\int_0^\tau\int_0^\tau dtdt'\eta(t)\hat{V}_\tau(t,t')\eta(t')\right)$$

Substituting $\hat{V}$ into the expression for B we get:

$$\begin{aligned} B = \int\mathcal{D}\overline{\eta}\exp\Bigg[&-\frac{1}{2}\int_0^\tau\int_0^\tau dtdt'\;\delta(t-t')\big[-\tau_a^2\eta(t)\ddot{\eta}(t') + L\eta(t)\eta(t') + \\ &+\delta(t)(\tau_a\eta(t)\eta(t') - \tau_a^2\eta(t)\dot{\eta}(t')) + \delta(\tau)(\tau_a\eta(t)\eta(t') + \tau_a^2\eta(t)\dot{\eta}(t'))\big]\Bigg]. \end{aligned}$$

Then, getting rid of the Dirac deltas by doing the integration results in:

$$\int \mathcal{D}\overline{\eta}\ \exp\left[\frac{1}{2}\int_0^\tau dt\ \left(\tau_a^2\eta(t)\ddot{\eta}(t) - L\eta(t)^2\right) + \frac{1}{2}BT\right]$$

where $BT = -\tau_a\eta(0)^2 + \tau_a^2\eta(0)\dot{\eta}(0) - \tau_a\eta(\tau)^2 - \tau_a^2\eta(\tau)\dot{\eta}(\tau)$ are the boundary terms.
Then, we can use integration by parts to solve $\int_0^\tau dt\ \eta(t)\ddot{\eta}(t) = \eta(\tau)\dot{\eta}(\tau) - \eta(0)\dot{\eta}(0) - \int_0^\tau dt\ \dot{\eta}(t)^2$. Therefore:

$$\begin{aligned} B &= \int \mathcal{D}\overline{\eta}\ \exp\left[\frac{1}{2}\int_0^\tau dt\ \left(-\tau_a^2\dot{\eta}(t)^2 - L\eta(t)^2\right) + \frac{1}{2}\tau_a^2\eta(\tau)\dot{\eta}(\tau) - \frac{1}{2}\tau_a^2\eta(0)\dot{\eta}(0) + \frac{1}{2}BT\right] \\ &= \int \mathcal{D}\overline{\eta}\ \exp\left[-\frac{1}{2}\int_0^\tau dt\ \left(\tau_a^2\dot{\eta}(t)^2 + L\eta(t)^2\right) + \frac{1}{2}BT'\right], \end{aligned}$$

where $BT' = BT + \tau_a^2\eta(\tau)\dot{\eta}(\tau) - \tau_a^2\eta(0)\dot{\eta}(0) = -\tau_a\eta(0)^2 - \tau_a\eta(\tau)^2$. Then, we rewrite the term in the integral using $\tau_a^2\dot{\eta}(t)^2 + L\eta(t)^2 = (\tau_a\dot{\eta}(t) + \sqrt{L}\eta(t))^2 - 2\tau_a\sqrt{L}\dot{\eta}(t)\eta(t)$.

$$\begin{aligned} B &= \int \mathcal{D}\overline{\eta}\ \exp\left[-\frac{1}{2}\int_0^\tau dt\ \left((\tau_a\dot{\eta}(t) + \sqrt{L}\eta(t))^2 - 2\tau_a\sqrt{L}\dot{\eta}(t)\eta(t)\right) + \frac{1}{2}BT'\right] \\ &= \int \mathcal{D}\overline{\eta}\ \exp\left[-\frac{1}{2}\int_0^\tau dt\ \left(\tau_a\dot{\eta}(t) + \sqrt{L}\eta(t)\right)^2 + \int_0^\tau dt\ \tau_a\sqrt{L}\dot{\eta}(t)\eta(t) + \frac{1}{2}BT'\right] \\ &= \int \mathcal{D}\overline{\eta}\ \exp\left[-\frac{1}{2}\int_0^\tau dt\ \left(\tau_a\dot{\eta}(t) + \sqrt{L}\eta(t)\right)^2 + \frac{1}{2}\tau_a\sqrt{L}\eta(\tau)^2 - \frac{1}{2}\tau_a\sqrt{L}\eta(0)^2 + \frac{1}{2}BT'\right]. \end{aligned}$$

Therefore,

$$B = \int \mathcal{D}\overline{\eta}\exp\left[-\frac{1}{2}\int_0^\tau dt\ (\tau_a\dot{\eta}(t) + \sqrt{L}\eta(t))^2 + \frac{1}{2}BT''\right], \tag{49}$$

where now $BT'' = BT' + \tau_a\sqrt{L}\eta(\tau)^2 - \tau_a\sqrt{L}\eta(0)^2 = -\tau_a\eta(0)^2 - \tau_a\eta(\tau)^2 + \tau_a\sqrt{L}\eta(\tau)^2 - \tau_a\sqrt{L}\eta(0)^2 = -\tau_a(1+\sqrt{L})\eta(0)^2 - \tau_a(1-\sqrt{L})\eta(\tau)^2$. So that finally, $BT'' = -\tau_a k_+\eta(0)^2 - \tau_a k_-\eta(\tau)^2$, with $k_\pm = 1 \pm \sqrt{L}$.

At this point, we can identify the integral in Eq. 49 as having the shape of an Onsager-Machlup integral for a system defined by the Langevin equation

$$\dot{y}(t) = -\frac{\sqrt{L}}{\tau_a}y(t) + \frac{1}{\tau_a}\xi(t). \tag{50}$$

For such a system, the probability of a given trajectory $\underline{y}$ is given by the Onsager-Machlup integral according to Eq. 43:

$$\begin{aligned} p(\underline{y}|y_0) &= C_0' \exp\left(-\int_0^\tau dt \left[\frac{\tau_a^2}{2}\left(\dot{y}(t) + \frac{\sqrt{L}}{\tau_a}y(t)\right)^2 - \frac{\sqrt{L}}{2\tau_a}\right]\right) \\ &= C_0' \exp\left(-\frac{1}{2}\int_0^\tau dt \left[(\tau_a\dot{y}(t) + \sqrt{L}y(t))^2 - \frac{\sqrt{L}}{\tau_a}\right]\right) \\ &= C_0' \exp\left(-\frac{1}{2}\int_0^\tau dt\ (\tau_a\dot{y}(t) + \sqrt{L}y(t))^2\ + \frac{\sqrt{L}\tau}{2\tau_a}\right), \end{aligned}$$

where $C_0' = \left(\frac{\tau_a}{\sqrt{2\pi\Delta t}}\right)^N$ is the normalization constant in the Onsager Machlup integral. Noting that this is very similar to the result we had for B in Eq. 49 is very useful, since we know that $p(\underline{y}|y_0)$ is normalized, so that $\int \mathcal{D}\underline{y}\ p(\underline{y}|y_0) = 1$. However, we are still missing something before being able to use this, since the integral for B is over $\overline{\eta}$, while the one on y is over $\underline{y}$, so we need to include the initial point y_0 in the y integral to make it comparable to the η integral.
From the Langevin equation for y, we know that the distribution of y_0 in steady state is a Gaussian with mean 0 and standard deviation $1/(\sqrt{2\tau_a}L^{1/4})$. Therefore, since the path integral is normalized, we have that:

$$\sqrt{\frac{\tau_a}{\pi}}L^{1/4}\int \mathcal{D}\overline{y}\ p(\underline{y}|y_0)e^{-\tau_a\sqrt{L}y(0)^2} = 1. \tag{51}$$

We notice that this integral is practically the same as we have for the η process in Eq. 49, but with a term $\frac{\sqrt{L}\tau}{2\tau_a} - \tau_a\sqrt{L}\eta(0)^2$, and without the $\frac{1}{2}BT''$ term. We know that when doing a path integral of this quantity, the result is 1 because it is normalized, therefore, we rewrite the η integral of Eq. 49 as:

$$\begin{aligned} B &= \int \mathcal{D}\overline{\eta}\exp\left[-\frac{1}{2}\int_0^\tau dt\ (\tau_a\dot{\eta}(t) + \sqrt{L}\eta(t))^2 + \frac{1}{2}BT''\right] \\ &= \int \mathcal{D}\overline{\eta}\exp\left[-\frac{1}{2}\int_0^\tau\ dt\ (\tau_a\dot{\eta}(t) + \sqrt{L}\eta(t))^2\ + \frac{\sqrt{L}\tau}{2\tau_a} - \tau_a\sqrt{L}\eta(0)^2 + \frac{1}{2}BT'' - \frac{\sqrt{L}\tau}{2\tau_a} + \tau_a\sqrt{L}\eta(0)^2\right] \\ &= \int \mathcal{D}\overline{\eta}\exp\left[-\frac{1}{2}\int_0^\tau\ dt\ (\tau_a\dot{\eta}(t) + \sqrt{L}\eta(t))^2\ + \frac{\sqrt{L}\tau}{2\tau_a} - \tau_a\sqrt{L}\eta(0)^2\right]\exp\left[\frac{1}{2}BT'' - \frac{\sqrt{L}\tau}{2\tau_a} + \tau_a\sqrt{L}\eta(0)^2\right] \end{aligned} \tag{52}$$

The integral of the first part is the one we have for the process y, so we know that it is normalized, and considering the $\sqrt{\frac{\tau_a}{\pi}}L^{1/4}$ factor in Eq. 51, the result

of this integral is $\frac{1}{C_0'}\sqrt{\frac{\pi}{\tau_a}}L^{-1/4}$. Therefore, we have:

$$B = \sqrt{\frac{\pi}{\tau_a}}L^{-1/4}\ \frac{\langle e^{BT''/2\,+\,\tau_a\sqrt{L}\eta(0)^2}\rangle e^{-\tau\sqrt{L}/(2\tau_a)}}{C_0'}. \tag{53}$$

Putting the Results Together

Now that we have the result for B, we can put everything together into equation 47 to get $p(\underline{x}|x_0)$:

$$p(\underline{x}|x_0) = C_1'B\exp\left(-\int_0^\tau dt\left[\frac{(\dot{x}_t - g_t)^2}{4D} + \frac{1}{2}\frac{\partial g_t}{\partial x}\right] + \frac{D_a}{4D^2}\int_0^\tau dt\int_0^\tau dt'(\dot{x}_t - g_t)\Gamma_\tau(t,t')(\dot{x}_{t'} - g_{t'})\right)$$

with $C_1'B$ given by:

$$\begin{aligned}
C_1'B &= \sqrt{\frac{\tau_a}{\pi}}L^{-1/4}\left(\frac{1}{\sqrt{2\pi\Delta t}}\right)^{2N}\left(\frac{\tau_a}{\sqrt{2D}}\right)^N e^{\tau/(2\tau_a)}\sqrt{\frac{\pi}{\tau_a}}\frac{\langle e^{BT''/2+\tau_a\sqrt{L}\eta(0)^2}\rangle e^{-\tau\sqrt{L}/(2\tau_a)}}{\left(\frac{\tau_a}{\sqrt{2\pi\Delta t}}\right)^N} \\
&= L^{-1/4}\left(\frac{1}{\sqrt{2\pi\Delta t}}\right)^N\left(\frac{1}{\sqrt{2D}}\right)^N e^{\tau(1-\sqrt{L})/(2\tau_a)}\langle e^{BT''/2+\tau_a\sqrt{L}\eta(0)^2}\rangle \\
&= L^{-1/4}\left(\frac{1}{\sqrt{4\pi D\Delta t}}\right)^N e^{\tau k_-/(2\tau_a)}\langle e^{BT''/2+\tau_a\sqrt{L}\eta(0)^2}\rangle.
\end{aligned}$$

To finish the calculation, we still need to obtain the expected value $\langle e^{BT''/2+\tau_a\sqrt{L}\eta(0)^2}\rangle$, where $BT'' = -\tau_a k_+\eta(0)^2 - \tau_a k_-\eta(\tau)^2$, with $k_\pm = 1 \pm \sqrt{L}$. This is:

$$\langle e^{-\tau_a k_-\eta_0^2/2 - \tau_a k_-\eta_\tau^2/2}\rangle = \langle e^{q(\eta_0^2+\eta_\tau^2)}\rangle,$$

with $q := -\tau_a k_-/2$ (the minus sign is so that q is positive, because k_- is always negative). Notice that, as we said earlier, the expected value is done by interpreting the variable η to follow the equation 50 and not the equation we had originally for η. Therefore, the statistics of η need to be obtained from equation 50 and the result is:

- $\sigma_0^2 := \langle\eta_0^2\rangle = \dfrac{1}{2\sqrt{L}\tau_a}$
- $\sigma_\tau^2 := \langle\eta_\tau^2\rangle = \dfrac{1}{2\sqrt{L}\tau_a}$

- $\rho := \dfrac{\langle \eta_0 \eta_\tau \rangle}{\sigma_0 \sigma_\tau} = e^{-\sqrt{L}\tau/\tau_a}$

Then, using the bivariate Gaussian distribution, we have that:

$$\langle e^{q(\eta_0^2+\eta_\tau^2)} \rangle = \frac{1}{2\pi\sigma_0\sigma_\tau\sqrt{1-\rho^2}} \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \exp\left[-\frac{1}{2(1-\rho^2)}\left(\frac{\eta_0^2}{\sigma_0^2}+\frac{\eta_\tau^2}{\sigma_\tau^2}-2\rho\frac{\eta_0\eta_\tau}{\sigma_0\sigma_\tau}\right)\right] e^{q\eta_0^2+q\eta_\tau^2}\, d\eta_0 d\eta_\tau,$$

Therefore:

$$\begin{aligned}
\langle e^{q(\eta_0^2+\eta_\tau^2)} \rangle &= \frac{\sqrt{L}\tau_a}{\pi\sqrt{1-\rho^2}} \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \exp\left[\frac{-1}{2(1-\rho^2)}\left(2\sqrt{L}\tau_a\eta_0^2 + 2\sqrt{L}\tau_a\eta_\tau^2 - 4\sqrt{L}\tau_a\rho\,\eta_0\eta_\tau\right) + q\eta_0^2 + q\eta_\tau^2\right] d\eta_0 d\eta_\tau \\
&= \frac{\sqrt{L}\tau_a}{\pi\sqrt{1-\rho^2}} \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \exp\left\{-\frac{1}{2}\left[\left(\frac{2\sqrt{L}\tau_a}{1-\rho^2}-2q\right)\eta_0^2 - \frac{4\sqrt{L}\tau_a\rho}{1-\rho^2}\eta_0\eta_\tau + \left(\frac{2\sqrt{L}\tau_a}{1-\rho^2}-2q\right)\eta_{\tau^2}\right]\right\} d\eta_0 d\eta_\tau \\
&= \frac{\sqrt{L}\tau_a}{\pi\sqrt{1-\rho^2}} \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \exp\left\{-\frac{1}{2}\begin{pmatrix}\eta_0 & \eta_\tau\end{pmatrix}\begin{pmatrix}\frac{2\sqrt{L}\tau_a}{1-\rho^2}-2q & -\frac{2\sqrt{L}\tau_a\rho}{1-\rho^2} \\ -\frac{2\sqrt{L}\tau_a\rho}{1-\rho^2} & \frac{2\sqrt{L}\tau_a}{1-\rho^2}-2q\end{pmatrix}\begin{pmatrix}\eta_0 \\ \eta_\tau\end{pmatrix}\right\} d\eta_0 d\eta_\tau
\end{aligned}$$

The result of this Gaussian integral is $\dfrac{2\pi}{\sqrt{\det A}}$, where A is the matrix inside the exponential. The determinant we have is:

$$\begin{aligned}
\det A &= \left(\frac{2\sqrt{L}\tau_a}{1-\rho^2}-2q\right)^2 - \frac{4L\tau_a^2\rho^2}{(1-\rho^2)^2} \\
&= \frac{4L\tau_a^2}{(1-\rho^2)^2} - \frac{8\sqrt{L}\tau_a q}{1-\rho^2} + 4q^2 - \frac{4L\tau_a^2\rho^2}{(1-\rho^2)^2} \\
&= \frac{4L\tau_a^2}{(1-\rho^2)^2}(1-\rho^2) - \frac{8\sqrt{L}\tau_a q}{1-\rho^2} + 4q^2 \\
&= \frac{4L\tau_a^2 - 8\sqrt{L}\tau_a q}{1-\rho^2} + 4q^2 \\
&= \frac{4L\tau_a^2 - 8\sqrt{L}\tau_a q + 4q^2(1-\rho^2)}{1-\rho^2}, \\
&= \frac{4\sqrt{L}\tau_a(\sqrt{L}\tau_a - 2q) + 4q^2(1-\rho^2)}{1-\rho^2}
\end{aligned}$$

where again $q = -\tau_a k_-/2$, $k_- = 1-\sqrt{L} = 1-\sqrt{1+D_a/D}$, and $\rho = e^{-\tau/\tau_a}$.

Then, we can substitute q and k_- to get:

$$\begin{aligned}\det A &= \frac{4\sqrt{L}\tau_a(\sqrt{L}\tau_a + \tau_a k_-) + \tau_a^2 k_-^2(1-\rho^2)}{1-\rho^2} \\ &= \frac{4\sqrt{L}\tau_a(\sqrt{L}\tau_a + \tau_a - \tau_a\sqrt{L}) + \tau_a^2(1-\sqrt{L})^2(1-\rho^2)}{1-\rho^2} \\ &= \frac{4\sqrt{L}\tau_a^2 + \tau_a^2(1-\sqrt{L})^2(1-\rho^2)}{1-\rho^2} \\ &= \frac{4\sqrt{L} + (1-\sqrt{L})^2(1-\rho^2)}{1-\rho^2}\tau_a^2\end{aligned}$$

Therefore, the expected value is:

$$\begin{aligned}\left\langle e^{q(\eta_0^2+\eta_\tau^2)}\right\rangle &= \frac{\sqrt{L}\tau_a}{\pi\sqrt{1-\rho^2}}\frac{2\pi}{\sqrt{\det A}} \\ &= \frac{2\sqrt{L}\tau_a\sqrt{1-\rho^2}}{\sqrt{1-\rho^2}\sqrt{4\sqrt{L}+(1-\sqrt{L})^2(1-\rho^2)}\ \tau_a} \\ &= \frac{2\sqrt{L}}{\sqrt{4\sqrt{L}+(1-\sqrt{L})^2(1-\rho^2)}}\end{aligned}$$

Finally, the constant $C_1'B$ in front of the exponential in the expression for $p(\underline{x}|x_0)$ is

$$C_1'B = L^{1/4}\left(\frac{1}{\sqrt{4\pi D\Delta t}}\right)^N e^{\tau k_-/(2\tau_a)}\frac{2}{\sqrt{4\sqrt{L}+(1-\sqrt{L})^2(1-\rho^2)}}$$

with $L = 1 + D_a/D$, $k_- = 1-\sqrt{L}$, and $\rho = e^{-\sqrt{L}\tau/\tau_a}$.

Final Result

Therefore, the final result is:

$$p(\underline{x}|x_0) = C_1'B\exp\left(-\int_0^\tau dt\left[\frac{(\dot{x}_t - g_t)^2}{4D} + \frac{1}{2}\frac{\partial g_t}{\partial x}\right] + \frac{D_a}{4D^2}\int_0^\tau dt\int_0^\tau dt'(\dot{x}_t - g_t)\Gamma_\tau(t,t')(\dot{x}_{t'} - g_{t'})\right), \tag{54}$$

with $C_1'B$ given by:

$$C_1'B = L^{1/4}\left(\frac{1}{\sqrt{4\pi D\Delta t}}\right)^N e^{\tau k_-/(2\tau_a)}\frac{2}{\sqrt{4\sqrt{L}+(1-\sqrt{L})^2(1-\rho^2)}},$$

and $L = 1 + D_a/D$, $k_\pm = 1 \pm \sqrt{L}$, $\rho = e^{-\sqrt{L}\tau/\tau_a}$.

We are just missing to mention the value of Γ_τ which was introduced before Eq. 47 for completing the square in the integral. We don't show it here, but the result is given in [11] and it is:

$$\Gamma_\tau(t,t') = \left(\frac{1}{2\tau_a\sqrt{L}}\right) \frac{k_+^2 e^{-\sqrt{L}|t-t'|/\tau_a} + k_-^2 e^{-\sqrt{L}(2\tau-|t-t'|)/\tau_a} - k_+k_- \left[e^{-\sqrt{L}(t+t')/\tau_a} + e^{-\sqrt{L}(2\tau-t-t')/\tau_a}\right]}{k_+^2 - k_-^2 e^{-2\sqrt{L}\tau/\tau_a}}.$$

Appendix C: Correlation calculation

In this appendix we calculate the correlation $\langle x(t)x(t')\rangle$ for an active Ornstein-Uhlenbeck particle on a harmonic potential. The equation describing the particle is

$$\dot{x}(t) = -k_\gamma x(t) + \sqrt{2D_a}\eta(t) + \sqrt{2D}\xi(t),$$
$$\dot{\eta}(t) = -\frac{1}{\tau_a}\eta(t) + \frac{1}{\tau_a}\zeta(t),$$

where for brevity, we define $k_\gamma = k/\gamma$. Also, ξ is an unbiased Gaussian white-noise, such that $\langle\xi(t)\rangle = 0$ and $\langle\xi(t)\xi(t')\rangle = \delta(t-t')$, and similarly for $\zeta(t)$.

Correlation for $\eta(t)$

We start by considering first the variable $\eta(t)$ and calculating its mean value $\langle\eta(t)\rangle$ and its correlation $\langle\eta(t)\eta(t')\rangle$, because we will need that to calculate the correlation of $x(t)$. First we formally solve the equation for $\eta(t)$ by introducing an integrating factor e^{t/τ_a}:

$$\begin{aligned}
&\dot{\eta}(t) + \frac{1}{\tau_a}\eta(t) = \frac{1}{\tau_a}\zeta(t) \\
&\Rightarrow\ \dot{\eta}(t)e^{t/\tau_a} + \frac{1}{\tau_a}\eta(t)e^{t/\tau_a} = \frac{1}{\tau_a}\zeta(t)e^{t/\tau_a} \\
&\Rightarrow\ \frac{d}{dt}\left(\eta(t)e^{t/\tau_a}\right) = \frac{1}{\tau_a}\zeta(t)e^{t/\tau_a} \\
&\Rightarrow\ \int_0^t \frac{d}{ds}\left(\eta(s)e^{s/\tau_a}\right)ds = \int_0^t \frac{1}{\tau_a}\zeta(s)e^{s/\tau_a}ds \\
&\Rightarrow\ \eta(t)e^{t/\tau_a} - \eta(0) = \int_0^t \frac{1}{\tau_a}\zeta(s)e^{s/\tau_a}ds \\
&\Rightarrow \eta(t) = \eta(0)e^{-t/\tau_a} + \frac{1}{\tau_a}\int_0^t \zeta(s)e^{(s-t)/\tau_a}ds. \qquad (55)
\end{aligned}$$

Now we can easily get the ensemble mean of $\eta(t)$ by using that $\langle\zeta(t)\rangle = 0$:

$$\langle\eta(t)\rangle = \langle\eta(0)\rangle e^{-t/\tau_a}. \qquad (56)$$

Therefore, we see that after a long time, the variable $\eta(t)$ will have a mean of 0, that is, once the particle has reached steady state, we can assume that $\langle\eta(t)\rangle = 0$.

Now we can calculate the correlation $\langle\eta(t)\eta(t')\rangle$. To do so, we use the result of Eq. 55 to get $\eta(t)\eta(t')$:

$$\eta(t)\eta(t') = \eta(0)^2 e^{-(t+t')/\tau_a} + \eta(0)\frac{1}{\tau_a}\int_0^{t'} \zeta(s)e^{(s-t'-t)/\tau_a}ds + \eta(0)\frac{1}{\tau_a}\int_0^{t} \zeta(s)e^{(s-t-t')/\tau_a}ds$$
$$+ \frac{1}{\tau_a^2}\int_0^t\int_0^{t'} \zeta(s)\zeta(s')e^{(s+s'-t-t')/\tau_a}ds'ds.$$

Then, we take the expected value of this quantity and use that $\langle\zeta(s)\rangle = 0$, $\langle\zeta(s)\zeta(s')\rangle = \delta(s-s')$. We also use that $\langle\eta(0)\zeta(s)\rangle = \langle\eta(0)\rangle\langle\zeta(s)\rangle = 0$ because η at time 0 is uncorrelated to the value of ζ at later times. Therefore:

$$\langle\eta(t)\eta(t')\rangle = e^{-(t+t')/\tau_a}\langle\eta(0)^2\rangle + \frac{1}{\tau_a^2}\int_0^t\int_0^{t'} \delta(s-s')e^{(s+s'-t-t')/\tau_a}ds'ds$$
$$= e^{-(t+t')/\tau_a}\langle\eta(0)^2\rangle + \frac{1}{\tau_a^2}\int_0^t \left[\int_0^t \delta(s-s')e^{(s+s'-t-t')/\tau_a}ds' + \int_t^{t'} \delta(s-s')e^{(s+s'-t-t')/\tau_a}ds'\right]ds$$
$$= e^{-(t+t')/\tau_a}\langle\eta(0)^2\rangle + \frac{1}{\tau_a^2}\int_0^t\int_0^{t} \delta(s-s')e^{(s+s'-t-t')/\tau_a}ds'ds.$$

The last step is true because in the second integral, s' is always bigger than s, so the delta function is 0. Then:

$$\begin{aligned}\langle\eta(t)\eta(t')\rangle &= e^{-(t+t')/\tau_a}\langle\eta(0)^2\rangle + \frac{1}{\tau_a^2}\int_0^t e^{(2s-t-t')/\tau_a}ds \\ &= e^{-(t+t')/\tau_a}\langle\eta(0)^2\rangle + \frac{1}{\tau_a^2}e^{-(t+t')/\tau_a}\left(\frac{\tau_a}{2}e^{2s}\right)\Big|_0^t \\ &= e^{-(t+t')/\tau_a}\langle\eta(0)^2\rangle + \frac{1}{\tau_a}e^{-(t+t')/\tau_a}\left(\frac{1}{2}e^{2t} - \frac{1}{2}\right) \\ &= e^{-(t+t')/\tau_a}\left(\langle\eta(0)^2\rangle - \frac{1}{2\tau_a}\right) + \frac{1}{2\tau_a}e^{-|t'-t|/\tau_a}.\end{aligned}$$

At the end I introduced the absolute value to $t'-t$, so that the result also applies also if $t > t'$. Finally, our result is

$$\langle\eta(t)\eta(t')\rangle = e^{-(t+t')/\tau_a}\left(\langle\eta(0)^2\rangle - \frac{1}{2\tau_a}\right) + \frac{1}{2\tau_a}e^{-|t'-t|/\tau_a}. \tag{57}$$

We can also directly obtain the second moment of $\eta(t)$ by setting $t' = t$:

$$\langle\eta(t)^2\rangle = e^{-2t/\tau_a}\left(\langle\eta(0)^2\rangle - \frac{1}{2\tau_a}\right) + \frac{1}{2\tau_a}. \tag{58}$$

Since we are interested in the results in steady state, we see what happens when $t \to \infty$. In such case, the second moment becomes:

$$\langle \eta(t)^2 \rangle = \frac{1}{2\tau_a}. \tag{59}$$

Therefore, if the system begins in steady state, the term $\langle \eta(0)^2 \rangle$ in Eq. 57 is equal to $\dfrac{1}{2\tau_a}$ and so the expression for the correlation in steady state reduces to:

$$\langle \eta(t)\eta(t') \rangle = \frac{1}{2\tau_a} e^{-|t'-t|/\tau_a}. \tag{60}$$

Correlation for $x(t)$

Now we calculate the correlation $\langle x(t)x(t') \rangle$. To do so, we first solve formally the equation for $x(t)$:

$$\begin{aligned}
&\dot{x}(t) + k_\gamma x(t) = \sqrt{2D_a}\eta(t) + \sqrt{2D}\xi(t) \\
&\Rightarrow \dot{x}(t)e^{k_\gamma t} + k_\gamma x(t)e^{k_\gamma t} = \sqrt{2D_a}\eta(t)e^{k_\gamma t} + \sqrt{2D}\xi(t)e^{k_\gamma t} \\
&\Rightarrow \frac{d}{dt}\left(x(t)e^{k_\gamma t}\right) = \sqrt{2D_a}\eta(t)e^{k_\gamma t} + \sqrt{2D}\xi(t)e^{k_\gamma t} \\
&\Rightarrow \int_0^t \frac{d}{ds}\left(x(s)e^{k_\gamma s}\right) ds = \sqrt{2D_a}\int_0^t \eta(s)e^{k_\gamma s} ds + \sqrt{2D}\int_0^t \xi(s)e^{k_\gamma s} ds \\
&\Rightarrow x(t)e^{k_\gamma t} = x(0) + \sqrt{2D_a}\int_0^t \eta(s)e^{k_\gamma s} ds + \sqrt{2D}\int_0^t \xi(s)e^{k_\gamma s} ds \\
&\Rightarrow \; x(t) = x(0)e^{-k_\gamma t} + \sqrt{2D_a}e^{-k_\gamma t}\int_0^t \eta(s)e^{k_\gamma s} ds + \sqrt{2D}e^{-k_\gamma t}\int_0^t \xi(s)e^{k_\gamma s} ds.
\end{aligned} \tag{61}$$

Given this, we can calculate the ensemble mean value of $x(t)$. We assume that we have reached steady state, so that $\langle \eta(s) \rangle = 0$. Therefore, the ensemble mean is:

$$\langle x(t) \rangle = \langle x(0) \rangle e^{-k_\gamma t}.$$

We can see that after a long time, this mean becomes 0. Therefore, since we want the correlation result in steady state, we will assume that a long time has already passed and so $\langle x(t) \rangle = 0$. To find the correlation, we first compute the

product of $x(t)x(t')$ from Eq. 61:

$$\begin{aligned}
x(t)x(t') = {} & x(0)^2 e^{-k_\gamma(t+t')} \; + \; x(0)\sqrt{2D_a}\, e^{-k_\gamma(t+t')} \int_0^t \eta(s) e^{k_\gamma s} ds \; + \; x(0)\sqrt{2D} e^{-k_\gamma(t+t')} \int_0^t \xi(s) e^{ks} ds \\
& + \; x(0)\sqrt{2D_a} e^{-k_\gamma(t+t')} \int_0^{t'} \eta(s') e^{k_\gamma s'} ds' \; + \; 2D_a e^{-k_\gamma(t+t')} \int_0^t \int_0^{t'} \eta(s)\eta(s') e^{k_\gamma(s+s')} ds' ds \\
& + \; 2\sqrt{DD_a} e^{-k_\gamma(t+t')} \int_0^t \int_0^{t'} \xi(s)\eta(s') e^{k_\gamma(s+s')} ds' ds \\
& + \; x(0)\sqrt{2D} e^{-k_\gamma(t+t')} \int_0^{t'} \xi(s') e^{k_\gamma s'} ds' \; + \; 2\sqrt{DD_a} e^{-k_\gamma(t+t')} \int_0^t \int_0^{t'} \xi(s')\eta(s) e^{k_\gamma(s+s')} ds' ds \\
& + \; 2D e^{-k_\gamma(t+t')} \int_0^t \int_0^{t'} \xi(s)\xi(s') e^{k_\gamma(s+s')} ds' ds.
\end{aligned}$$

Then, we take the mean of this, for which we use that $\langle \eta(s) \rangle = 0$ (in steady state), $\langle \xi(s) \rangle = 0$ and that ξ and η are uncorrelated, so that $\langle \xi(s')\eta(s) \rangle = \langle \xi(s') \rangle \langle \eta(s) \rangle = 0$. We also use that $x(0)$ and noises at later times are uncorrelated, so that $\langle x(0)\xi(s) \rangle = \langle x(0) \rangle \langle \xi(s) \rangle = 0$. Using this, we can see that the second, third, fourth, sixth, seventh, and eight terms in the result for $x(t)x(t')$ are equal to 0 and therefore we get:

$$\begin{aligned}
\langle x(t)x(t') \rangle = {} & \langle x(0)^2 \rangle e^{-k_\gamma(t+t')} \; + \; 2D_a e^{-k_\gamma(t+t')} \int_0^t \int_0^{t'} \langle \eta(s)\eta(s') \rangle e^{k_\gamma(s+s')} ds' ds \\
& + \; 2D e^{-k_\gamma(t+t')} \int_0^t \int_0^{t'} \langle \xi(s)\xi(s') \rangle e^{k_\gamma(s+s')} ds' ds \\
= {} & \langle x(0)^2 \rangle e^{-k_\gamma(t+t')} + \frac{D_a}{\tau_a} e^{-k_\gamma(t+t')} \int_0^t \int_0^{t'} e^{k_\gamma(s+s')} e^{-|s'-s|/\tau_a} ds' ds \\
& + 2D e^{-k_\gamma(t+t')} \int_0^t \int_0^{t'} \delta(s-s') e^{k_\gamma(s+s')} ds' ds. \qquad (62)
\end{aligned}$$

Here we used the result obtained for $\langle \eta(t)\eta(t') \rangle$. Now we will do each of these two integrals separately.

First Integral

We start solving this integral by separating the interval $[0, t']$ into $[0, t]$ and $[t, t']$.

$$\begin{aligned}
\int_0^t \int_0^{t'} e^{k_\gamma(s+s')} e^{-|s'-s|/\tau_a} ds' ds &= \int_0^t \left[\int_0^t e^{k_\gamma(s+s')} e^{-|s'-s|/\tau_a} ds' + \int_t^{t'} e^{k_\gamma(s+s')} e^{-|s'-s|/\tau_a} ds' \right] ds \\
&= \int_0^t \left[\int_0^t e^{k_\gamma(s+s')} e^{-|s'-s|/\tau_a} ds' + \int_t^{t'} e^{k_\gamma(s+s') - (s'-s)/\tau_a} ds' \right] ds \\
&= \int_0^t \left[\int_0^t e^{k_\gamma(s+s')} e^{-|s'-s|/\tau_a} ds' + e^{\Lambda_+ s} \int_t^{t'} e^{\Lambda_- s'} ds' \right] ds,
\end{aligned}$$

where $\Lambda_- := k_\gamma - 1/\tau_a$ and $\Lambda_+ := k_\gamma + 1/\tau_a$. Actually doing the integration we get:

$$
\begin{aligned}
&= \int_0^t \left[\int_0^t e^{k_\gamma(s+s')} e^{-|s'-s|/\tau_a} ds' + e^{\Lambda_+ s} \frac{1}{\Lambda_-} \left(e^{\Lambda_- t'} - e^{\Lambda_- t} \right) \right] ds \\
&= \int_0^t \int_0^t e^{k_\gamma(s+s')} e^{-|s'-s|/\tau_a} ds' ds + \frac{1}{\Lambda_+} \left(e^{\Lambda_+ t} - 1 \right) \frac{1}{\Lambda_-} \left(e^{\Lambda_- t'} - e^{\Lambda_- t} \right) \\
&= \int_0^t \int_0^t e^{k_\gamma(s+s')} e^{-|s'-s|/\tau_a} ds' ds \; + \; \frac{1}{\Lambda_+ \Lambda_-} \left(e^{\Lambda_+ t} - 1 \right) \left(e^{\Lambda_- t'} - e^{\Lambda_- t} \right). \quad (63)
\end{aligned}
$$

To finish the calculation, we still need to do the missing integral in Eq. 63:

$$
\begin{aligned}
\int_0^t \int_0^t e^{k_\gamma(s+s')} e^{-|s'-s|/\tau_a} ds' ds &= \int_0^t \left[\int_0^s e^{k_\gamma(s+s')} e^{-(s-s')/\tau_a} ds' + \int_s^t e^{k_\gamma(s+s')} e^{-(s'-s)/\tau_a} ds' \right] ds \\
&= \int_0^t \left[e^{\Lambda_- s} \int_0^s e^{\Lambda_+ s'} ds' + e^{\Lambda_+ s} \int_s^t e^{\Lambda_- s'} ds' \right] ds \\
&= \int_0^t \left[e^{\Lambda_- s} \frac{1}{\Lambda_+} \left(e^{\Lambda_+ s} - 1 \right) + e^{\Lambda_+ s} \frac{1}{\Lambda_-} \left(e^{\Lambda_- t} - e^{\Lambda_- s} \right) \right] ds \\
&= \frac{1}{\Lambda_+} \int_0^t e^{2ks} - e^{\Lambda_- s} \, ds + \frac{1}{\Lambda_-} \int_0^t e^{\Lambda_- t} e^{\Lambda_+ s} - e^{2k_\gamma s} ds \\
&= \frac{1}{\Lambda_+} \left[\frac{1}{2k_\gamma} \left(e^{2k_\gamma t} - 1 \right) - \frac{1}{\Lambda_-} \left(e^{\Lambda_- t} - 1 \right) \right] + \frac{1}{\Lambda_-} \left[\frac{e^{\Lambda_- t}}{\Lambda_+} \left(e^{\Lambda_+ t} - 1 \right) - \frac{1}{2k_\gamma} \left(e^{2k_\gamma t} - 1 \right) \right].
\end{aligned}
$$

Therefore, the complete result of the first integral in Eq. 62 is:

$$
\begin{aligned}
&\int_0^t \int_0^{t'} e^{k_\gamma(s+s')} e^{-|s'-s|/\tau_a} ds' ds = \\
&= \frac{1}{\Lambda_+} \left[\frac{1}{2k_\gamma} \left(e^{2k_\gamma t} - 1 \right) - \frac{1}{\Lambda_-} \left(e^{\Lambda_- t} - 1 \right) \right] + \frac{1}{\Lambda_-} \left[\frac{e^{\Lambda_- t}}{\Lambda_+} \left(e^{\Lambda_+ t} - 1 \right) - \frac{1}{2k_\gamma} \left(e^{2k_\gamma t} - 1 \right) \right] \\
&\quad + \frac{1}{\Lambda_+ \Lambda_-} \left(e^{\Lambda_+ t} - 1 \right) \left(e^{\Lambda_- t'} - e^{\Lambda_- t} \right).
\end{aligned}
$$

Second Integral

The second integral in Eq. 62 can be calculated directly:

$$
\begin{aligned}
\int_0^t \int_0^{t'} \delta(s-s') e^{k_\gamma(s+s')} ds' ds &= \int_0^t \left[\int_0^t \delta(s-s') e^{k_\gamma(s+s')} ds' + \int_t^{t'} \delta(s-s') e^{k_\gamma(s+s')} ds' \right] ds \\
&= \int_0^t e^{2k_\gamma s} ds \\
&= \frac{1}{2k_\gamma} \left(e^{2k_\gamma t} - 1 \right).
\end{aligned}
$$

Complete Result
Now that we have the results for both integrals in Eq. 62, we can substitute them to get the correlation:

$$\langle x(t)x(t')\rangle = \langle x(0)^2\rangle e^{-k_\gamma(t+t')} + \frac{D}{k_\gamma}e^{-k_\gamma(t+t')}\left(e^{2kt}-1\right)$$
$$+\frac{D_a}{\tau_a}e^{-k_\gamma(t+t')}\Bigg\{\frac{1}{\Lambda_+}\left[\frac{1}{2k_\gamma}\left(e^{2k_\gamma t}-1\right)-\frac{1}{\Lambda_-}\left(e^{\Lambda_- t}-1\right)\right]+\frac{1}{\Lambda_-}\left[\frac{e^{\Lambda_- t}}{\Lambda_+}\left(e^{\Lambda_+ t}-1\right)-\frac{1}{2k_\gamma}\left(e^{2k_\gamma t}-1\right)\right]$$
$$+\frac{1}{\Lambda_+\Lambda_-}\left(e^{\Lambda_+ t}-1\right)\left(e^{\Lambda_- t'}-e^{\Lambda_- t}\right)\Bigg\}.$$

We expand out the terms, so that later we may see a clear time dependence:

$$\langle x(t)x(t')\rangle = e^{-k_\gamma(t+t')}\left(\langle x(0)^2\rangle - \frac{D}{k_\gamma}\right)+\frac{D}{k_\gamma}e^{-k_\gamma(t'-t)}+\frac{D_a}{2k_\gamma\tau_a\Lambda_+}\left(e^{-k_\gamma(t'-t)}-e^{-k_\gamma(t+t')}\right)$$
$$-\frac{D_a}{\tau_a\Lambda_+\Lambda_-}\left(e^{-t/\tau_a-k_\gamma t'}-e^{-k_\gamma(t+t')}\right)$$
$$+\frac{D_a}{\Lambda_-\Lambda_+\tau_a}\left(e^{-k_\gamma(t'-t)}-e^{-t/\tau_a-k_\gamma t'}\right)-\frac{D_a}{2k_\gamma\Lambda_-\tau_a}\left(e^{-k_\gamma(t'-t)}-e^{-k_\gamma(t+t')}\right)$$
$$+\frac{D_a}{\tau_a\Lambda_+\Lambda_-}\left(e^{-1/\tau_a(t'-t)}-e^{-k_\gamma(t'-t)}-e^{-k_\gamma t-t'/\tau_a}+e^{-k_\gamma t'-t/\tau_a}\right).$$

Finally, let's put together the terms with the same exponent:

$$\langle x(t)x(t')\rangle = e^{-k_\gamma(t+t')}\left(\langle x(0)^2\rangle - \frac{D}{k_\gamma}-\frac{D_a}{2k_\gamma\tau_a\Lambda_+}+\frac{D_a}{\tau_a\Lambda_+\Lambda_-}+\frac{D_a}{2k_\gamma\Lambda_-\tau_a}\right)$$
$$+e^{-k_\gamma(t'-t)}\left(\frac{D}{k_\gamma}+\frac{D_a}{2k_\gamma\tau_a\Lambda_+}+\frac{D_a}{\Lambda_-\Lambda_+\tau_a}-\frac{D_a}{2k_\gamma\Lambda_-\tau_a}-\frac{D_a}{\tau_a\Lambda_+\Lambda_-}\right)$$
$$+e^{-t/\tau_a-k_\gamma t'}\left(-\frac{D_a}{\tau_a\Lambda_+\Lambda_-}-\frac{D_a}{\Lambda_-\Lambda_+\tau_a}+\frac{D_a}{\tau_a\Lambda_+\Lambda_-}\right)-e^{-t'/\tau_a-k_\gamma t}\left(\frac{D_a}{\tau_a\Lambda_+\Lambda_-}\right)+e^{-1/\tau_a(t'-t)}\left(\frac{D_a}{\tau_a\Lambda_+\Lambda_-}\right)$$
$$=\left(\langle x(0)^2\rangle+\frac{D-Dk_\gamma\tau_a+D_a}{k_\gamma(-1+k_\gamma\tau_a)}\right)e^{-k_\gamma(t+t')}+\left(\frac{D}{k_\gamma}+\frac{D_a}{k_\gamma(1-k_\gamma^2\tau_a^2)}\right)e^{-k_\gamma(t'-t)}+\frac{D_a\tau_a}{1-k_\gamma^2\tau_a^2}\,e^{-t/\tau_a-k_\gamma t'}$$
$$+\frac{D_a\tau_a}{1-k_\gamma^2\tau_a^2}\,e^{-t'/\tau_a-k_\gamma t}-\frac{D_a\tau_a}{1-k_\gamma^2\tau_a^2}\,e^{-1/\tau_a(t'-t)}. \tag{64}$$

However, we are only interested in the result after a long time has passed, so that we have reached steady state. To get this, we first need to see what is the value of $\langle x(0)^2\rangle$ in steady state. To do so, we set $t'=t$:

$$\langle x(t)^2\rangle = \left(\langle x(0)^2\rangle+\frac{D-Dk_\gamma\tau_a+D_a}{k_\gamma(-1+k_\gamma\tau_a)}\right)e^{-2k_\gamma t}+\frac{D+Dk_\gamma\tau_a+D_a}{k_\gamma(1+k_\gamma\tau_a)}+\frac{2D_a\tau_a}{1-k_\gamma^2\tau_a^2}e^{-t/\tau_a-k_\gamma t}.$$

We can see that after a long time has passed and we reach steady state, this result becomes:

$$\langle x(t)^2 \rangle = \frac{D + Dk_\gamma\tau_a + D_a}{k_\gamma(1 + k_\gamma\tau_a)}.$$

Therefore, if we start from steady state, we have that $\langle x^2(0) \rangle = \frac{D + Dk_\gamma\tau_a + D_a}{k_\gamma(1 + k_\gamma\tau_a)}$. Substituting this into Eq. 64, and disregarding the terms that become 0 if t and t' are large enough, we get the steady state correlation:

$$\langle x(t)x(t') \rangle = \left(\frac{D}{k_\gamma} + \frac{D_a}{k_\gamma(1 - k_\gamma^2\tau_a^2)} \right) e^{-k_\gamma(t'-t)} - \frac{D_a\tau_a}{1 - k_\gamma^2\tau_a^2}\, e^{-1/\tau_a(t'-t)} \tag{65}$$

Bibliography

[1] Charu C. Aggarwal. *Neural Networks and Deep Learning*. Springer, Cham, Switzerland, second edition edition, 2023.

[2] Ricard Alert, Jaume Casademunt, and Jean-François Joanny. Active turbulence. *Annual Review of Condensed Matter Physics*, 13(Volume 13, 2022):143–170, 2022.

[3] A. Altemose and A. Sen. Chapter 11: Collective behaviour of artificial microswimmers in response to environmental conditions. *RSC Theoretical and Computational Chemistry Series*, pages 250–283, 01 2019.

[4] Andre C. Barato and Udo Seifert. Thermodynamic uncertainty relation for biomolecular processes. *Phys. Rev. Lett.*, 114:158101, Apr 2015.

[5] Urna Basu, Satya N. Majumdar, Alberto Rosso, and Grégory Schehr. Active brownian motion in two dimensions. *Phys. Rev. E*, 98:062121, Dec 2018.

[6] William Bialek. Biophysics: Searching for principles. *Biophysics: Searching for Principles*, 11 2012.

[7] Xin Bian, Changho Kim, and George Em Karniadakis. 111 years of brownian motion. *Soft Matter*, 12:6331–6346, 2016.

[8] L. L. Bonilla. Active ornstein-uhlenbeck particles. *Phys. Rev. E*, 100:022601, Aug 2019.

[9] Christian BROECK. Stochastic thermodynamics: A brief introduction. *Proc. Of the International School of Physics 'Enrico Fermi'*, 184, 09 2014.

[10] Ivo Buttinoni, Julian Bialké, Felix Kümmel, Hartmut Löwen, Clemens Bechinger, and Thomas Speck. Dynamical clustering and phase separation in suspensions of self-propelled colloidal particles. *Phys. Rev. Lett.*, 110:238301, Jun 2013.

[11] Agnese Callegari and Giovanni Volpe. *Numerical Simulations of Active Brownian Particles*, pages 211–238. Springer International Publishing, Cham, 2019.

[12] Michele Caraglio and Thomas Franosch. Analytic solution of an active brownian particle in a harmonic well. *Phys. Rev. Lett.*, 129:158001, Oct 2022.

[13] Giuseppe Carleo, Ignacio Cirac, Kyle Cranmer, Laurent Daudet, Maria Schuld, Naftali Tishby, Leslie Vogt-Maranto, and Lenka Zdeborová. Machine learning and the physical sciences. *Rev. Mod. Phys.*, 91:045002, Dec 2019.

[14] Andrea Cavagna and Irene Giardina. Bird flocks as condensed matter. *Annual Review of Condensed Matter Physics*, 5(Volume 5, 2014):183–207, 2014.

[15] Fang-Yi Chu, Shannon C. Haley, and Alexandra Zidovska. On the origin of shape fluctuations of the cell nucleus. *Proceedings of the National Academy of Sciences*, 114(39):10338–10343, 2017.

[16] Lennart Dabelow, Stefano Bo, and Ralf Eichhorn. Irreversibility in active matter systems: Fluctuation theorem and mutual information. *Phys. Rev. X*, 9:021009, Apr 2019.

[17] G. De Magistris and D. Marenduzzo. An introduction to the physics of active matter. *Physica A: Statistical Mechanics and its Applications*, 418:65–77, 2015. Proceedings of the 13th International Summer School on Fundamental Problems in Statistical Physics.

[18] Abhishek Dhar, Anupam Kundu, Satya N. Majumdar, Sanjib Sabhapandit, and Grégory Schehr. Run-and-tumble particle in one-dimensional confining potentials: Steady-state, relaxation, and first-passage properties. *Phys. Rev. E*, 99:032132, Mar 2019.

[19] Jens Elgeti, R. Winkler, and Gerhard Gompper. Physics of microswimmers - single particle motion and collective behavior. *Reports on Progress in Physics*, 78:056601, 04 2015.

[20] Chung-Kang Peng et al. Non-equilibrium dynamics as an indispensable characteristic of a healthy biological system. *Springer Nature*, 29:283–293, 1994.

[21] Daniel A. Fletcher and R. Dyche Mullins. Cell mechanics and the cytoskeleton. *Nature*, 463:485–492, 2010.

[22] Andonis Gerardos and Pierre Ronceray. Principled model selection for stochastic dynamics, 2025.

[23] Federico S. Gnesotto, Grzegorz Gradziuk, Pierre Ronceray, and Chase P. Broedersz. Learning the non-equilibrium dynamics of brownian movies. *Nature Communications*, 11, 10 2020.

[24] Mohammad Hafezi and Christopher Jarzynski. Machine learning the thermodynamic arrow of time. *Nature Physics*, 17:1–9, 01 2021.

[25] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9:1735–1780, 11 1997.

[26] Dong-Kyum Kim, Youngkyoung Bae, Sangyun Lee, and Hawoong Jeong. Learning entropy production via neural networks. *Phys. Rev. Lett.*, 125:140604, Oct 2020.

[27] Artemy Kolchinsky, Naruo Ohga, and Sosuke Ito. Thermodynamic bound on spectral perturbations, with applications to oscillations and relaxation dynamics. *Phys. Rev. Res.*, 6:013082, 1 2024.

[28] Junang Li, Jordan M. Horowitz, Todd R. Gingrich, and Nikta Fakhri. Quantifying dissipation using fluctuating currents. *Nature Communications*, 10, 04 2019.

[29] Simone Pigolotti Luca Peliti. *Stochastic Thermodynamics: An Introduction*. Princeton University Press, Princeton, 2021.

[30] Gautam I. Menon. Active matter, 2010.

[31] Domor Mienye, Theo Swart, and George Obaido. Recurrent neural networks: A comprehensive review of architectures, variants, and applications. *Information*, 15:517, 08 2024.

[32] Andrew Missel, Mo Bai, William Klug, and Alex Levine. The equilibrium and nonequilibrium mechanics of cytoskeletal networks. *Biophysical Journal - BIOPHYS J*, 98, 01 2010.

[33] Arvind Murugan, David A. Huse, and Stanislas Leibler. Speed, dissipation, and error in kinetic proofreading. *Proceedings of the National Academy of Sciences*, 109(30):12034–12039, 2012.

[34] Daniel Needleman and Zvonimir Dogic. Active matter at the interface between materials science and cell biology. *Nature Reviews Materials*, 2, 2017.

[35] Shun Otsubo, Sosuke Ito, Andreas Dechant, and Takahiro Sagawa. Estimating entropy production by machine learning of short-time fluctuating currents. *Phys. Rev. E*, 101:062106, Jun 2020.

[36] R. K. Pathria and Beale Paul D. *Statistical Mechanics*. Academic Press, fourth edition edition, 2021.

[37] Vladimir N Pokrovskii. *Thermodynamics of Complex Systems*. 2053-2563. IOP Publishing, 2020.

[38] Karel Proesmans and Bernard Derrida. Large-deviation theory for a brownian particle on a ring: a wkb approach. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(2):023201, feb 2019.

[39] Sriram Ramaswamy. Active matter. *Journal of Statistical Mechanics: Theory and Experiment*, 2017(5):054002, may 2017.

[40] Robin Schmidt. Recurrent neural networks (rnns): A gentle introduction and overview. *Eberhard-Karls-University Tübingen*, 11 2019.

[41] Udo Seifert. Entropy production along a stochastic trajectory and an integral fluctuation theorem. *Phys. Rev. Lett.*, 95:040602, Jul 2005.

[42] Udo Seifert. Stochastic thermodynamics. fluctuation theorems and molecular machines. *Reports on progress in physics. Physical Society (Great Britain)*, 75:126001, 12 2012.

[43] S. S. Shapiro and M. B. Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, 52(3/4):591–611, 1965.

[44] Margaret L. Gardel Shiladitya Banerjee and Ulrich S. Schwarz. The actin cytoskeleton as an active adaptive material. *Annual Review of Condensed Matter Physics*, 11:421–439, 2020.

[45] Naoto Shiraishi. *An Introduction to Stochastic Thermodynamics*. Springer, Berlin, 2023.

[46] T. Speck, V. Blickle, C. Bechinger, and U. Seifert. Distribution of entropy production for a colloidal particle in a nonequilibrium steady state. *Europhysics Letters*, 79(3):30002, jul 2007.

[47] T. Speck, V. Blickle, C. Bechinger, and U. Seifert. Distribution of entropy production for a colloidal particle in a nonequilibrium steady state. *Europhysics Letters*, 79(3):30002, jul 2007.

[48] J. Toner. *The Physics of Flocking: Birth, Death, and Flight in Active Matter*. Cambridge University Press, 2024.

[49] Tan Van Vu, Van Tuan Vo, and Yoshihiko Hasegawa. Entropy production estimation with optimal current. *Phys. Rev. E*, 101:042138, Apr 2020.

[50] Hyung-June Woo and Anders Wallqvist. Nonequilibrium phase transitions associated with dna replication. *Phys. Rev. Lett.*, 106:060601, Feb 2011.

[51] Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola. Dive into deep learning, 2023.

[52] Ming-Li Zhang, Ziheng Zhang, Xue-Zhi Niu, Hui-Ying Ti, Yu-Xuan Zhou, Bo Gao, Yiwei Li, Ji-Long Liu, Xiaosong Chen, and Hui Li. Interplay between intracellular transport dynamics and liquid–liquid phase separation (adv. sci. 19/2024). *Advanced Science*, 11(19):2470107, 2024.

[53] Étienne Fodor and M. Cristina Marchetti. The statistical physics of active matter: From self-catalytic colloids to living cells. *Physica A: Statistical Mechanics and its Applications*, 504:106–120, 2018. Lecture Notes of the 14th International Summer School on Fundamental Problems in Statistical Physics.