UNIVERSITY OF COPENHAGEN
FACULTY OF SCIENCE

Master's thesis

Title

Subtitle

Tomas Ricardo Basile Alvarez

Advisor: Karel Proesmans

Submitted: January 17, 2025

Contents

Abstract

1 Introduction

Living organisms fundamentally operate out of equilibrium, driven by a continuous influx and dissipation of energy. This persistent non-equilibrium state is crucial for sustaining life in cells [19, 6], playing a key role in processes such as DNA replication [47, 30], intracellular transport, and cellular organization [20, 48, 29].

Describing living systems, therefore, requires a framework grounded in non-equilibrium thermodynamics [34]. Over the past few decades, the theory of stochastic thermodynamics has emerged as a powerful tool for understanding non-equilibrium systems. It has revealed fundamental principles that apply even to systems far from equilibrium [39].

One particularly intriguing class of non-equilibrium systems is active matter [27, 16, 36, 49, 15]. Active matter consists of large assemblies of entities (active particles) that extract energy from their environment and convert it into motion. These systems inherently break time-reversal symmetry due to the continuous dissipation of energy and exhibit remarkable emergent phenomena, such as active turbulence [2], collective motion [10, 3], flocking [45], and active stress [31].

Biological systems provide many examples of active matter, including the cytoskeleton of living cells [41], nuclear fluctuations within cells [14], swarming bacteria [18], and, on larger scales, animal groups such as flocking birds and human crowds [13].

In this work, we investigate methods for determining whether a system is active or passive using neural networks. Machine learning, particularly neural networks, has been widely applied across nearly every area of physics [12]. In particular, in stochastic thermodynamics, recent studies have used these techniques to classify the direction of time's arrow [22] and estimate a system's entropy production [23, 32, 21]. Here, we address a different challenge: developing a classifier capable of distinguishing between active and passive systems, and then also estimating the active diffusion coefficient.
Probably add more about what we are doing and why its important

1.1 Outline

Section 2 introduces the foundational concepts of stochastic thermodynamics and the Langevin formalism, providing the necessary tools for the rest of the thesis. In Section 3, we discuss an existing method from the literature [23] that employs a neural network to predict a system's entropy production. However, we find that this approach often provides a poor lower bound on entropy production in cases involving hidden entropy contributions, such as in active matter.

In Section 4, we develop a neural network designed to classify whether a system exhibits active noise. We do so with a feed forward neural network and also with a recurrent neural network. Moreover, we present a way of doing this classification theoretically by using path integrals to calculate the probability of a system's trajectory given it is active or passive.

In Section 5, we apply these neural networks to analyze real data of the motion of cells and draw conclusions about its activity. Section 6 extends this analysis to the trajectories of cellular nuclei. Finally, Section 7 summarizes the results obtained in this thesis, discusses their implications, and explores future perspectives and challenges.

2 Theory

We begin with an overview of the theoretical framework needed for this thesis. To do so, we introduce the fundamental concepts of stochastic thermodynamics and the Langevin equation. Building on this, we explore the notion of entropy production within stochastic thermodynamics. Finally, we introduce the mathematical model we will use of 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. Over the past few decades, however, this perspective has shifted. The application of thermodynamic principles to microscopic systems has become increasingly widespread [26], driven in part by the study of biomolecules, which are analyzed as small machines that consume fuel and perform work [42].

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 applying thermodynamic concepts like heat, work, and entropy at the level of individual trajectories [9]. Using stochastic processes such as Langevin equations, it 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. Crucially, the particle under study need not be in equilibrium with its thermal environment; it may be externally manipulated or, more relevant for this work, it can be self-propelled, as in active systems [26].

Stochastic thermodynamics has found applications across many fields [26]. 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 under-

standing 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. This erratic motion, known as Brownian motion, arises from the unpredictable nature of the molecular collisions [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 to so 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.

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

$$\langle \xi(t) \rangle = 0. \tag{3}$$

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:

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

with $\langle \Delta W \rangle = 0$ and $\langle \Delta W^2 \rangle = \Delta t$. These two moments of ΔW can be proven from the properties defined for $\xi(t)$. However, the extra assumption done 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 can 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.

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, the coefficient of $\xi(t)$ is written as $\gamma\sqrt{2D}$ with D identified as the diffusion coefficient. To demonstrate this, 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:

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

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 parameter λ that can vary in time. Also, as before, the thermal noise ξ has

correlations given by $\langle \xi(t)\xi(t') \rangle = \delta(t - t')$.

We will 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 [39]:

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

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 it's entropy $s(t)$ along the path, known as **stochastic entropy**. Defining this may sound absurd, since entropy was first conceived as a property of an ensemble of particles, not something defined for a trajectory of one particle. However, we can define the trajectory dependent entropy of the particle as [39, 38, 9]:

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

where $p(x, t)$ is the solution of the Fokker-Planck equation. The definition should be multiplied by k_B, but from here on we are setting $k_B = 1$. 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:

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

This coincides with the well-known definition of Shannon entropy [26, 33].

Rate of entropy change

We can find the rate of change of the entropy of the system [38] by differentiating Eq. 10. To get the total time derivative of $s_{sys}(t)$, we need to do the partial derivative with respect to 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.

The last term can be identified as the change in entropy due to the heat dissipation to the medium. That is, the rate of change of the medium's entropy, due to the heat that the particle dissipates is:

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

where $\dot{q}(t) = f(x,\lambda)\dot{x}$ is the heat flow to the medium [39].

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 (Eq. 13) [38]. This is what we call the **entropy production:**

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

We can also obtain the ensemble averages of these quantities, for which we use that $\langle \dot{x}|x,t\rangle = j(x,t)/p(x,t)$ [38]. For the medium entropy, we get as a result:

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

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

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

Note that the ensemble average of the rate of entropy production 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 [27, 43]. These forces need to be incorporated to Langevin's equation and they can be described in a variety of ways, as we mention 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 [16]. Therefore, it can't be described by the term we already use for normal diffusion.

Then, to model the motion of an active particle (and overdamped), 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:

- 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 describe $\eta(t)$ in Eq. 17. Below are some of the most common approaches:

- **Ornstein-Uhlenbeck Process:** The active noise follows the equation [8, 15, 43]:

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

  Here $\zeta(t)$ is Gaussian noise with mean zero and delta correlation, that is, $\langle\zeta(t)\rangle = 0$ and $\langle\zeta(t)\zeta(t')\rangle = \delta(t-t')$. On the other hand, 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 [17]. This type of noise is frequently used to describe bacterial motion [16].
- **Brownian Active Noise:** Here, the active noise is modeled as $\eta = \cos(\theta(t))$, where $\theta(t)$ evolves by:

$$\dot{\theta} = C\zeta(t),$$

  with C 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. We will mostly use the AOUP model.

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 [25]. 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 [32, 46, 4].

Recently, machine learning has opened new avenues for estimating entropy production [21, 23], including applications like inferring the arrow of time [22]. In particular, the method proposed in [23] employs a neural network to estimate not only the ensemble-averaged entropy production but also the stochastic entropy production along individual trajectories. Here, we summarize this innovative approach and its implications for nonequilibrium thermodynamics.

3.1 NEEP

The method introduced in [23], called the **Neural Estimator for Entropy Production (NEEP)**, is designed to estimate the stochastic 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 define a function $h_\theta(s_t, s_{t+1})$, which takes two consecutive states of the system and 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 (which we denote

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 $\Delta S_\theta(s_t, s_{t+1}) = \Delta S_\theta(s_{t+1}, s_t)$. 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, samples 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} 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 $\Delta S(s_t, s_{t+1})$ 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 [23]:

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

where $h_{ij} := h(s_i, s_j)$, and $p_i := p(s_i)$ is the steady state probability density. 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 [39]. 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 $f \geq 0$. Therefore, the particle feels a force of [35]:

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

and the effective potential is given by:

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

The particle's overdamped motion is governed by the Langevin equation [44]:

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

where as always, the thermal noise $\xi(t)$ has zero mean and 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 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 [35]:

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

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 $f = 0.3$.

Figure 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.

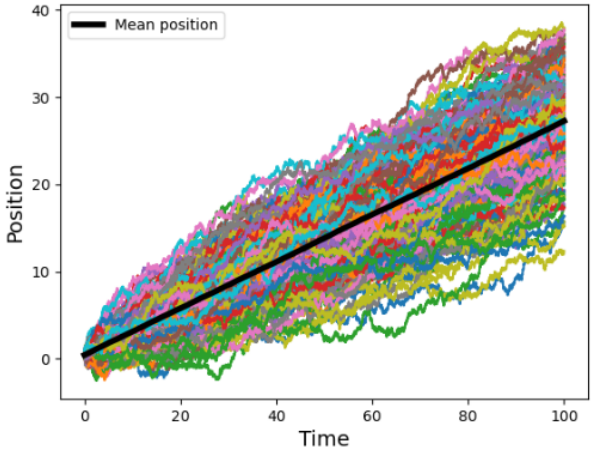


Figure 1: Sample trajectories of a Brownian particle on a ring. Parameters: $D = 0.2, f = 0.3, V(x) = -x + 3x^2 - 2x^3$. Although x is periodic with period 1, the plot shows the absolute position for clarity and to observe how many trips around the ring the particle has done.

NEEP

Given all these trajectories, we use the NEEP to estimate their stochastic entropy production. To do so, we create a feed forward 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]).

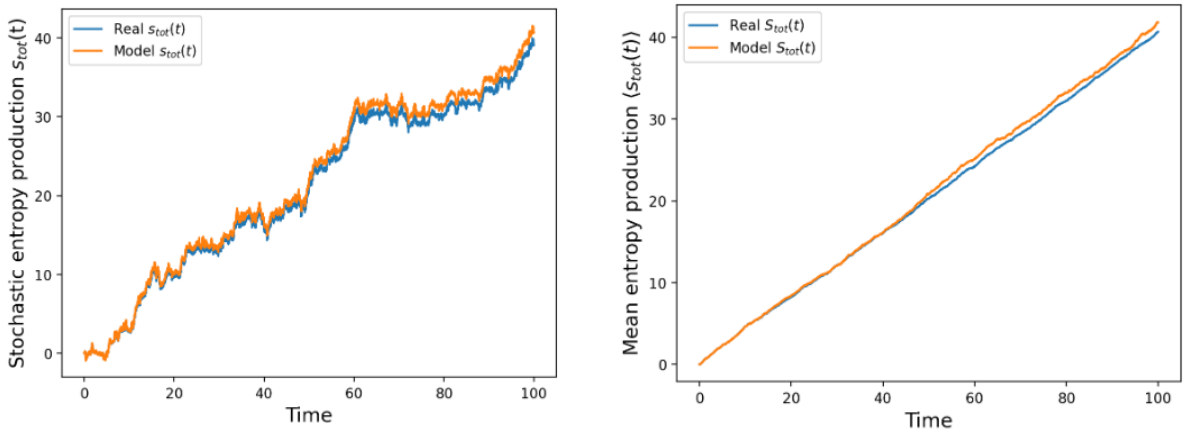
The network is trained to maximize the objective function $J(\theta)$ (Eq. 20). Once trained, the entropy production for a single transition is approximated as:

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

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

Results

The result of the stochastic 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 $S_{tot}(t) := \langle s_{tot}(t) \rangle$ 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).



(a) Stochastic entropy production.

(b) Mean entropy production.

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

From figure 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.

3.3 NEEP for an Ornstein-Uhlenbeck Active 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

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

with $U(x)$ the potential experienced by the particle, D is the diffusion coefficient, D_a is the active diffusion coefficient, τ_a is 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 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 [23]. Consequently, the NEEP framework in this case does not capture the full entropy dynamics but provides insight into the observable portion of the system's non-equilibrium behavior.

3.3.1 Harmonic Potential

We begin by considering a harmonic potential, $U(x) = \frac{1}{2}kx^2$, where 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 retained 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.

Stochastic Entropy Production

Using the definition of stochastic entropy production from Eq. 14, we calculate the entropy production 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 $D_{\text{mat}} = \text{diag}\left(D, \frac{1}{2\tau_a^2}\right)$.

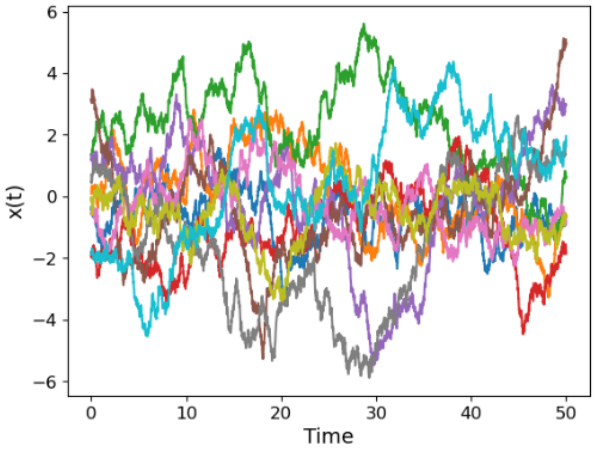


Figure 3: Sample trajectories of an active Ornstein-Uhlenbeck particle in a harmonic potential, with $D = 0.2$, $D_a = 0.2$, $\tau_a = 0.2$, $k = 0.1$, and $\Delta t = 0.01$.

The stochastic 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 [15]:

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

Using this distribution $p(\mathbf{x},t)$, we can easily compute the stochastic 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 results in $\langle \dot{s}_{\text{tot}} \rangle = 4.912$.

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.

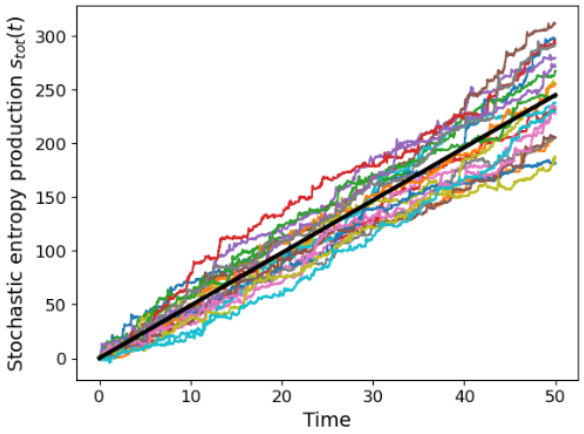


Figure 4: Stochastic entropy production $s_{\text{tot}}(t)$ for 20 trajectories of an AOUP in a harmonic potential. The black line represents the mean entropy production $\langle s_{\text{tot}}(t)\rangle$, growing at a practically constant rate of $\langle \dot{s}_{\text{tot}}(t)\rangle = 4.912$.

This result is expected, as the NEEP only captures the dynamics observable through $x(t)$, neglecting contributions from the hidden variable $\eta(t)$.

Moreover, this estimated entropy production aligns with findings in [15], 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. Notices that the definition only depends on $x(t)$ 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.

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 force of $-U'(x) = -kx^3$. Similar to the harmonic potential case, we generate $10,000$ trajectories using the same simulation approach, but now with the quartic potential.

Again, the stochastic entropy production for each trajectory is calculated using Eq. **??**, 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 calcualte, the theoretical mean entropy production rate obtaines 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 [15]. This is shown in fig. 5, where we plot the stochastic entropy production for a given trajectory, calculated both with the coarse grained entropy $\Delta\Sigma$ of Eq. 24 and with the NEEP trained only on trajectories $x(t)$.

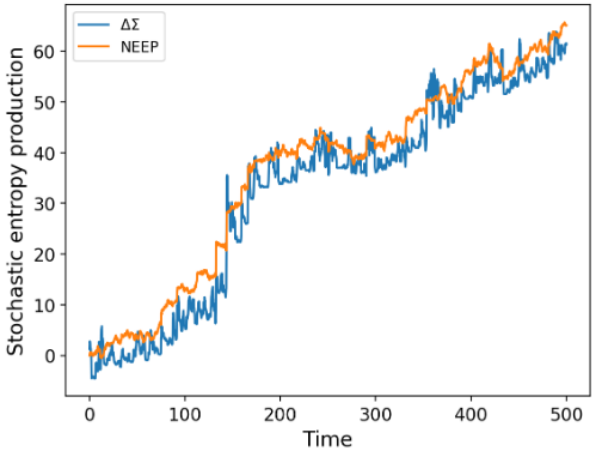


Figure 5: The stochastic entropy production $s_{tot}(t)$ for a single trajectory estimated using the coarse-grained entropy production defined in Eq. 24 and using a NEEP trained on trajectories $x(t)$

4 Classifying Active Noise

In the previous section, 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 determining whether the measured trajectory of a particle exhibits active noise or not.

This section 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 Section 5. For more complex potentials encountered in Section 6, 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 harmonic potential $U(x) = \frac{1}{2}kx^2$, which is governed by the equations:

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

This system is fully described by four parameters: D, D_a, τ_a, and k/γ. 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 $t_{in} = 100$ (not saved), followed by saving the trajectory for a total duration of $\tau = 100$.

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 $8,000$ randomly selected trajectories, with the remaining $2,000$ 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 we define $p \in [0, 1]$ as the probability of having active noise as predicted by the network. 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 outcome y relative to the prediction p. For example, if the actual outcome 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 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

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

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 Feed Forward Neural Network

We begin with a feedforward neural network, a widely used 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 points in each trajectory ($\tau = 100$ with a time step $\Delta t = 0.01$, 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 a probability between 0 and 1 for whether a trajectory contains active noise.

The hyperparameters, including the architecture and learning rate, were selected using a grid search [1]. After training the network, we evaluated its performance on the test dataset, and the results are shown in Fig. 6. The figure demonstrates that while the network can perform some level of classification, its accuracy is limited. The mean predicted probability for trajectories without active noise is 0.41, and for those with active noise, it is 0.61. The overall log-loss for these predictions is 0.589, indicating room for improvement.

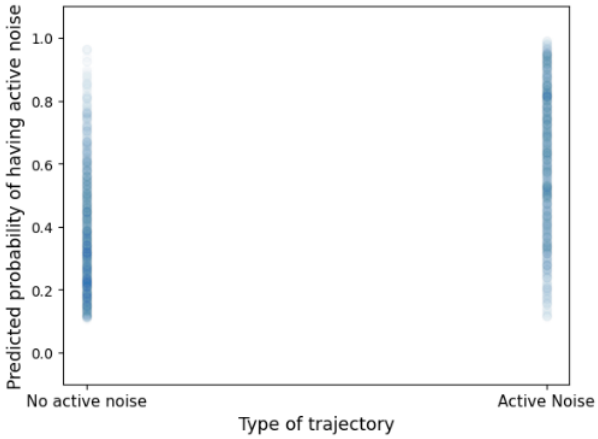


Figure 6: Test data results of the feed forward neural network in classifying trajectories as having active noise or not. Each point represents a trajectory, the horizontal axis indicates the true label of the trajectory and the vertical axis indicates the probability of having active noise as predicted by the feed forward neural network. The mean predicted probability for trajectories without active noise is 0.41 and with active noise is 0.61. The log loss of all these predictions is 0.589.

4.2 Recurrent Neural Network

Next, we attempt the classification task using a recurrent neural network (RNN) [37, 28], which is well-suited for sequential data processing. Unlike feedforward neural networks, 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, a hidden layer, and an output layer. 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 100, the trajectory is split into $n = 100$ sections: $x_1, x_2, \ldots, x_n$.

2. At each time step t (from 1 to n), the RNN takes the segment x_t and updates the hidden state using:

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

where W_{xh} is the weight matrix between the input and hidden layer, W_{hh} is the weight matrix for the recurrent connection, b_h is the bias vector and σ_h is the activation function (in this case a hyperbolic tangent) [28].

- The output at each time step t is computed as:

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

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:

$$f_\theta(x_t, h_t) = (y_t, h_{t+1}).$$

This standard RNN architecture, however, suffers from the vanishing gradient problem, where gradients become too small during backpropagation, hindering the training process. To address this issue, we employ a more advanced architecture: Long Short-Term Memory (LSTM) networks. LSTMs mitigate the vanishing gradient problem by using gating mechanisms to control the flow of information [28].

After training the RNN, we evaluate it on the test dataset, with results shown in Fig. 7. Compared to the feedforward network, the RNN demonstrates significantly better performance in classifying trajectories. The mean predicted probability for trajectories without active noise is 0.279, while for those with active noise, it is 0.81. The overall log loss is 0.355, reflecting a notable improvement.

4.3 Theoretical Computation

In this section, we outline a theoretical method for performing the classification task. As before, we consider two types of trajectories, both characterized by the same parameters

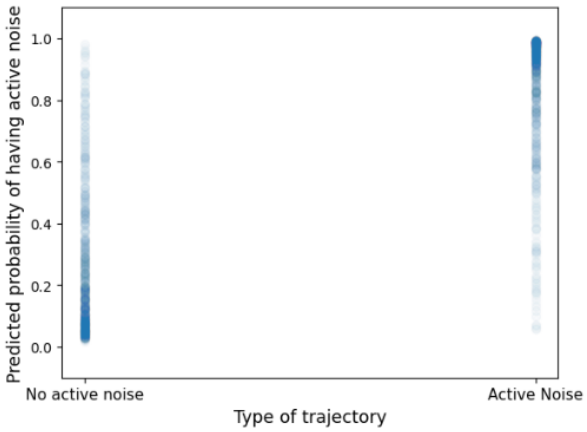


Figure 7: Test data results of the recurrent neural network in classifying trajectories as having active noise or not. Each point represents a trajectory, the horizontal axis indicates the true label of the trajectory and the vertical axis indicates the probability of having active noise as predicted by the RNN. The mean predicted probability for trajectories without active noise is 0.279 and with active noise is 0.81. The log loss of all these predictions is 0.355.

k/γ, D, and τ_a. However, 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. 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 same quantity we attempted to determine using neural networks. Note that $\overline{x}$ represents the whole trajectory of the particle and $\underline{x}$ represents the trajectory omitting the first point. From now on, we will always use $\underline{x}$ and the probabilities are computed conditional to the first point of the trajectory x_0.

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

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

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 $P(\underline{x}|D_a = D'_a)$.

$P(\underline{x}|D_a = 0)$ 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 noise. 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 [15]. The difference is that in said paper, they only calculate this quantity up to a normalization factor, but in appendix B we do a more complete derivation without ignoring the normalization constants.

The results obtained in appendix B are

$$P(\underline{x}|D_a = 0) = C_0 \exp\left\{ -\int_0^\tau dt \left[\frac{(\dot{x} - g_t)^2}{4D} + \frac{1}{2}\frac{\partial g_t}{\partial x} \right] \right\},$$

where the subindex t means that the variable is evaluated at time t (and position $x(t)$) and $g(x) := \frac{1}{\gamma} f(x)$. And

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

with K given by:

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

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

Then, we can use those results and Eq. 28 to calculate $P(D_a = D'_a|\underline{x})$ for any trajectory. We do so for the trajectories in the same test set as used for the neural networks and the results are shown in fig. 8

4.4 Results for Varying Trajectory Lengths

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

Figure 9 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

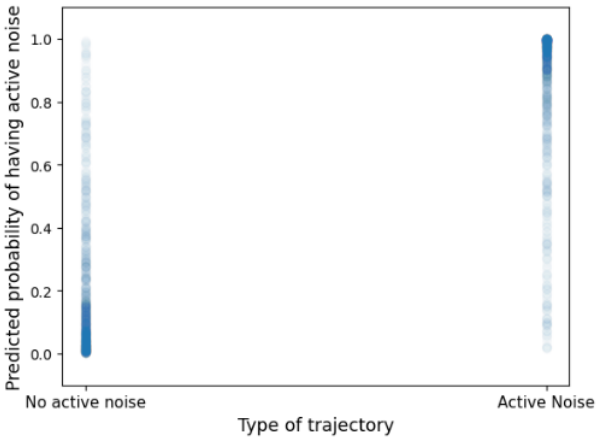


Figure 8: Results of theoretically classifying trajectories as having active noise or not. Each point represents a trajectory, the horizontal axis indicates the true label of the trajectory and the vertical axis indicates the probability of having active noise as calculated theoretically. The mean predicted probability for trajectories without active noise is 0.237 and with active noise is 0.808. The log loss of all these predictions is 0.3377.

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.
Note that the feedforward neural network consistently performs worse than the other methods across all trajectory lengths. Furthermore, its improvement with increasing τ is relatively slow compared to the other approaches. In contrast, the recurrent neural network and the theoretical method both show substantial improvement with longer trajectories. Their performance is closely matched, although the theoretical method slightly outperforms the RNN at most trajectory lengths.

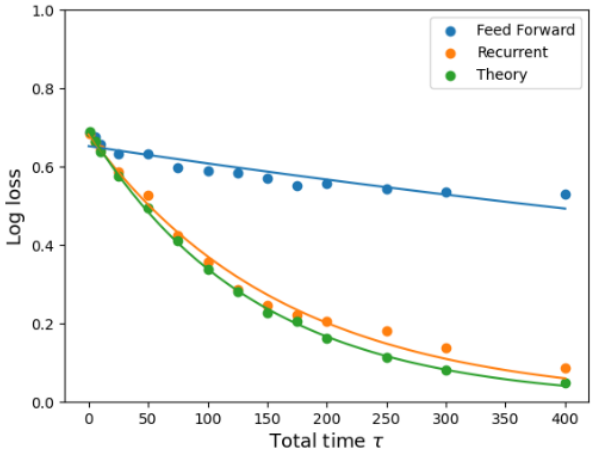


Figure 9: Log loss on the test set for the three methods across different total trajectory times τ, along with the best-fitting exponential curves of the form $ae^{b\tau}$ for each method.

5 Cellular data

In this section, we analyze real trajectory data representing the motion of a living cell within a dense environment of other cells. We apply a recurrent neural network (RNN), similar to the one used in the previous section, to classify whether the trajectory is influenced by active noise. Additionally, we extend the approach by training a separate RNN 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. To explore the statistical properties of the data, we first plot a histogram of the position x, as shown in Fig. 10. The histogram closely resembles a Gaussian distribution, which we confirm quantitatively using the Shapiro-Wilk test of normality [40], yielding a value of 0.9997. This Gaussian nature suggests that the particle is likely subject to a harmonic potential.

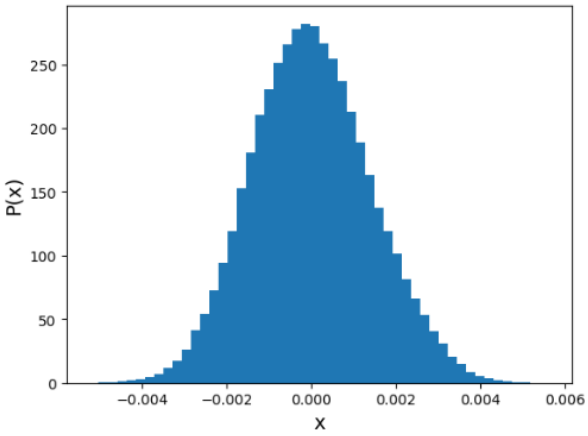


Figure 10: Histogram of the position x for the cellular data.

Because of this, we will train the RNNs using simulated trajectories of an AOUP in a harmonic potential. However, before that, we will use the correlations of the data to get an approximation of the range of values of the parameters $k/\gamma, \tau_a, D_a, D$. Then we can create simulated trajectories with said ranges and use them to train the RNN.

5.1.1 Correlations

Given the harmonic nature suggested by the histogram, we can compare the trajectory data to analytical results derived for an AOUP in a harmonic potential. A key comparison involves calculating the correlation function $\langle x(t)x(t+t')\rangle$ and matching it to

theoretical predictions.

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:

$$\langle x(t)x(t+t')\rangle = \left(\frac{D}{k} + \frac{D_a}{k(1-k^2\tau_a^2)}\right) e^{-kt'} - \frac{D_a\tau_a}{1-k^2\tau_a^2}\, e^{-t'/\tau_a}. \tag{29}$$

Furthermore, the variance of x in the steady state (assuming the data is centered so that $\langle x(t)\rangle = 0$) is:

$$\langle x(t)^2\rangle = \frac{D + Dk\tau_a + D_a}{k(1+k\tau_a)} \tag{30}$$

$$\tag{31}$$

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 along the trajectory, and then average these products. Repeating this for various values of t', we obtain the empirical correlation function, shown in Fig. 11. The figure also includes a fit of the theoretical result from Eq. 29 to the calculated data.

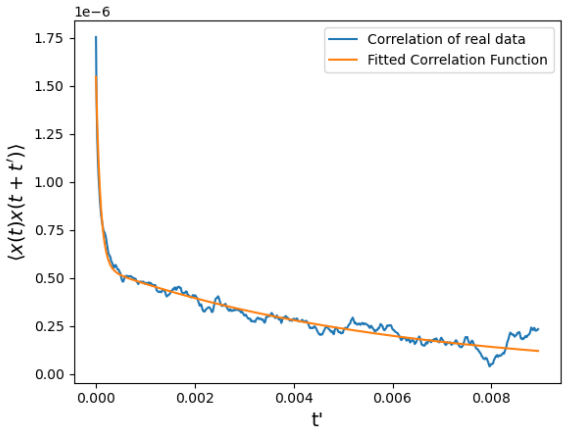


Figure 11: Correlation $\langle x(t)x(t+t')\rangle$ of real data and the result of fitting Eq. 29 to this data.

The fitting process yields the values of the system's four defining parameters: k, τ_a, D, and D_a:

$$\begin{aligned} k/\gamma &= 157, \\ \tau_a &= 0.333, \\ D_a &= 0.0063, \\ D &= 0.00014. \end{aligned} \tag{32}$$

5.2 Classifying Active Noise

Training Data

In this section, we train a Recurrent Neural Network (RNN) to classify whether a given trajectory contains active noise, i.e., whether the diffusion coefficient D_a is nonzero. We will use this model to identify if a trajectory is influenced by active noise (i.e., if $D_a > 0$) or not (i.e., if $D_a = 0$).

To create the training data, we generate trajectories with random values for k, τ_a, and D, chosen to be within a range based on the parameters obtained by fitting the correlation data in Eq. 32. 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 $[50, 250]$, the values of τ_a from $[0.05, 0.8]$, D from $[10^{-6}, 0.0005]$ and D_a from $[10^{-6}, 0.01]$.

We generate 5000 trajectories with $D_a = 0$ and 5000 trajectories with $D_a > 0$, each consisting of 5000 time steps. The RNN is then trained to predict whether a given trajectory has $D_a > 0$. 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 Figure 12.

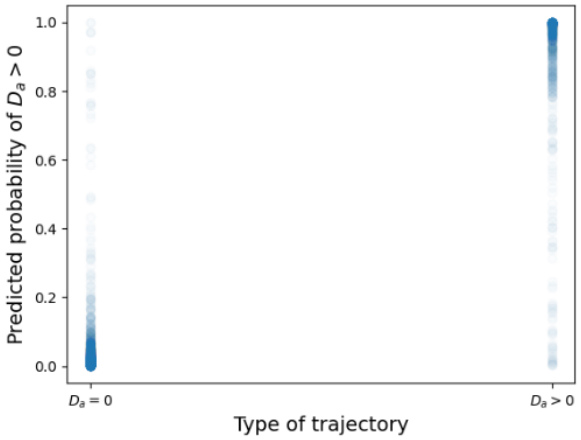


Figure 12: Results of the RNN model on the test set to classify whether a trajectory has $D_a = 0$ (no active noise) or $D_a > 0$ (active noise).

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.

Application to Real Data

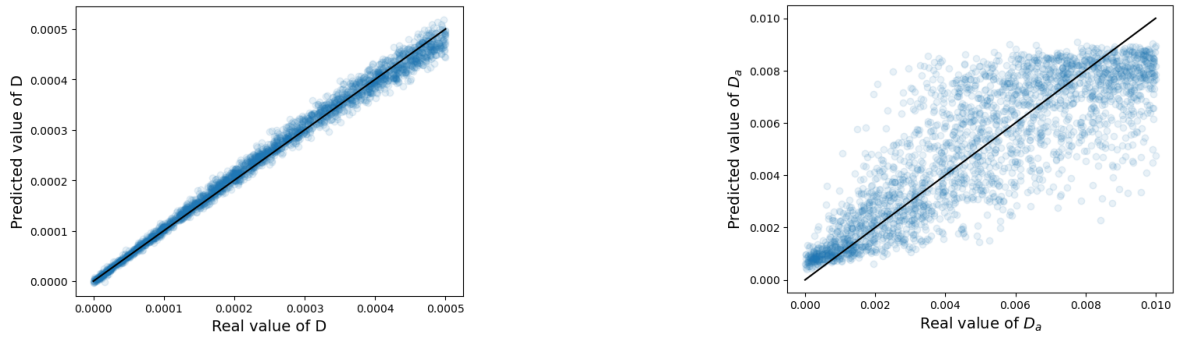
We then apply the trained model to real data. Since the RNN is designed to handle trajectories of varying lengths, we input the full trajectory of the experimental data. The model predicts a confidence score of 0.997, indicating a high probability that the trajectory exhibits active noise. Alternatively, we divide the long experimental trajectory into 131 shorter segments, each containing 5000 points, and apply the model to each segment. For every segment, the model consistently returns a prediction greater than 0.99, reinforcing the conclusion that active noise is present across the entire trajectory.

5.3 Predicting Diffusion Coefficients

Training

Next, we attempt to predict the actual values of the diffusion coefficients D and D_a, rather than simply classifying whether D_a is zero or non-zero.

For this, we generate $10,000$ trajectories, in the same way as before, but all of them with $D_a > 0$ (in the range $[10^{-6}, 0.01]$). We then train an RNN to predict D and a different RNN to predict D_a from a given trajectory. The results on the test set of simulated trajectories are shown in fig. 13, where the predicted values of D and D_a are plotted against the true values.



(a) Results for D

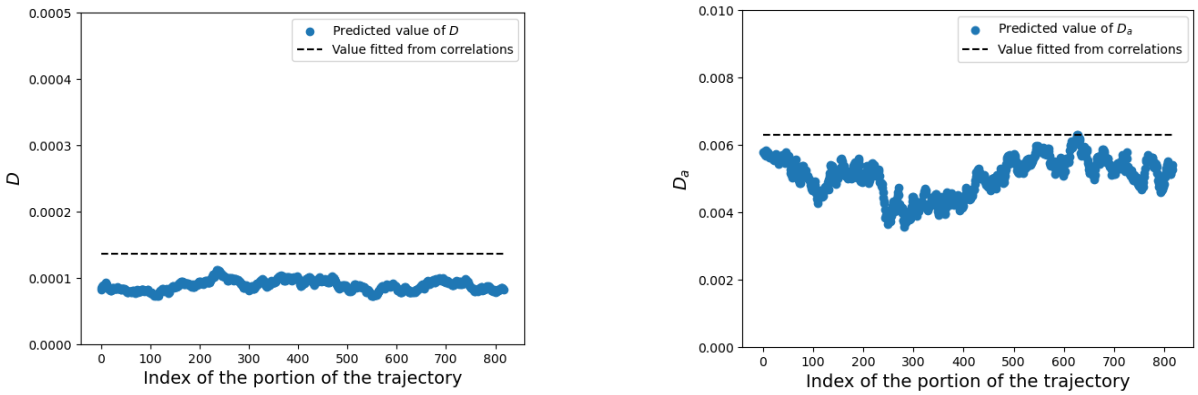
(b) Results for D_a

Figure 13: Results of the RNN 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.

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.

Application to Real Data

We now apply the RNNs trained in the previous section 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 Figure 14, where we compare the predicted values to those obtained by fitting the data to the analytical correlation model.



(a) Results for D

(b) Results for D_a

Figure 14: 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, though slightly lower than, the values obtained from fitting the correlation data.

5.4 Power Spectrum Analysis

Next, we conduct an analysis based on the method outlined in [24], which uses the spectral density of an observable $x(t)$ to determine whether a system is out of equilibrium and to place a lower bound on the entropy production in steady state.

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. 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 Figure 15. To compute this, we first divide the trajectory 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.

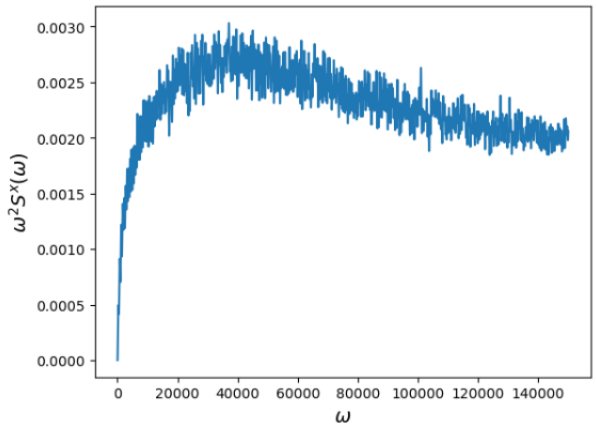


Figure 15: Result of $\omega^2 S^x(\omega)$ for the experimental trajectory $x(t)$.

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 [24], this means that the trajectory is out of equilibrium, since an equilibrium trajectory would grow monotonously.

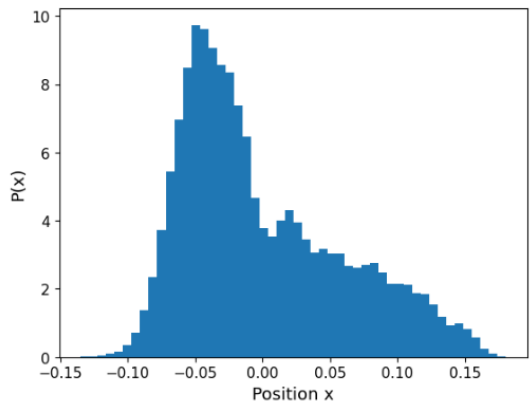
6 Nuclear data

In this section, we perform a similar analysis to that of the previous section for cellular data, but now applied to position-time data of lipid granules within cellular nuclei. Specifically, we have data for two types of nuclei: the EPI nucleus and the PrE nucleus.

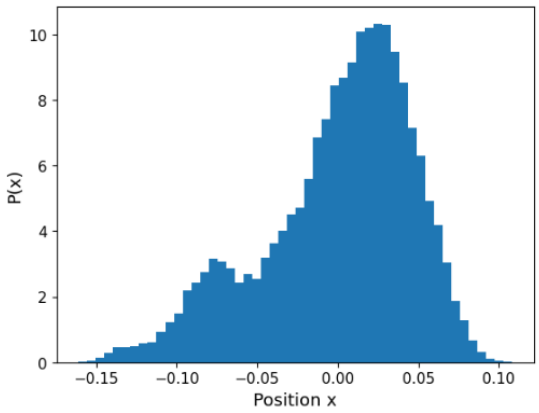
6.1 Data Analysis

6.1.1 EPI Nucleus

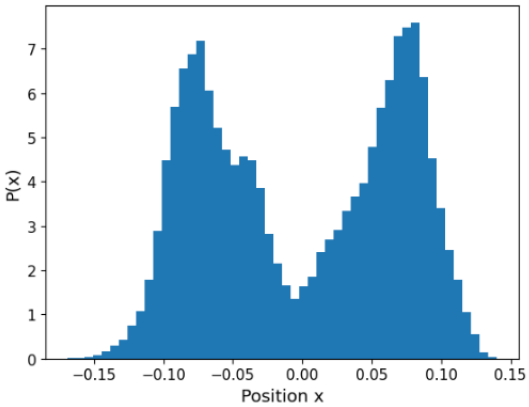
We begin by examining the EPI nuclear data. This dataset consists of 7 different trajectories, each with a time step of $\Delta t = 4.5454 \times 10^{-5}$ s and a total of 65,535 steps. To begin our analysis, we compute histograms for the first four trajectories, as shown in Figure 16.



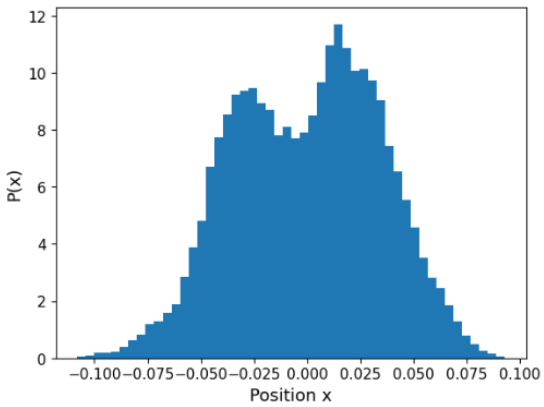
(a) Trajectory 1



(b) Trajectory 2



(c) Trajectory 3



(d) Trajectory 4

Figure 16: Histograms for four of the EPI nucleus trajectories.

As observed in Figure 16, the histograms do not follow a Gaussian distribution. This suggests that we cannot apply the same analysis methods used in the previous section with cellular data, where a clear Gaussian distribution allowed us to assume a harmonic potential $U(x)$.

Given the non-Gaussian nature of the histograms, we do not assume a harmonic potential for this case. However, we still aim to characterize the potential $U(x)$ for these trajectories to create similar synthetic trajectories, which can be used to train a neural network. To achieve this, we assume that the potential takes a quartic form:

$$U(x) = k_1 x + k_2 x^2 + k_3 x^3 + k_4 x^4. \tag{33}$$

From Boltzmann's distribution we expect that the logarithm of the probability $p(x)$ obtained in the histogram should be $\log p(x) = -U(x)/k_B T = -U(x)/(\gamma D)$, where at the end we used Eq. 7, therefore, if we assume $U(x)$ is of the form of Eq. 33, we get:

$$-\ln p(x) = \frac{k1}{\gamma D} x + \frac{k2}{\gamma D} x^2 + \frac{k3}{\gamma D} x^3 + \frac{k4}{\gamma D} x^4. \tag{34}$$

Next, we compute $-\ln p(x)$ for the trajectories shown in Figure 16, and fit the data to the equation above to determine the values of $\frac{k_1}{\gamma D}, \frac{k_2}{\gamma D}, \frac{k_3}{\gamma D}, \frac{k_4}{\gamma D}$. The results of these fits are presented in Figure 17, where we see that the fits match the data quite well for each trajectory.

6.1.2 PrE nucleus

Next, we perform the same analysis for the PrE nuclear data. This dataset contains 6 trajectories, with the same time step of $\Delta t = 4.5454 \times 10^{-5}$ s and 65,535 total steps. We present the histograms for the first four trajectories in Figure 18.

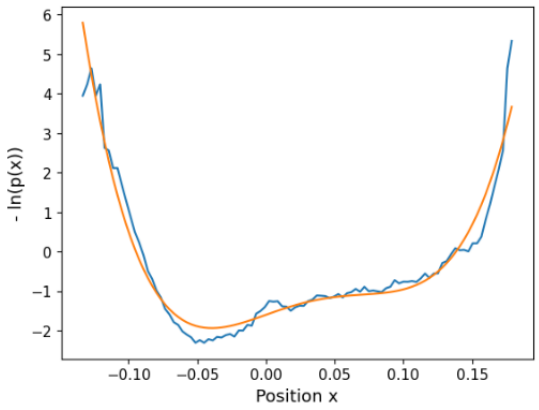
As seen in fig. 18, the histograms again do not follow a Gaussian distribution. Thus, we proceed with the same assumption as for the EPI nucleus, i.e., that the potential follows a quartic form as described by Equation 33. Consequently, the logarithm of the probability in the histogram is also given by Equation 34.

We then fit Eq. 34 to the histograms, and the fitting results are shown in fig. 19. As with the EPI data, the fits work well for all the trajectories.

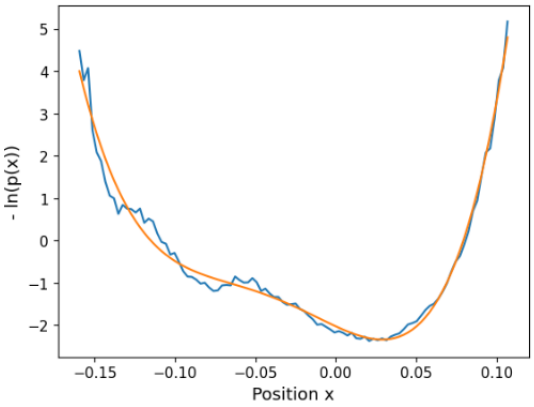
6.2 Neural Network Training

6.2.1 Classifying active noise

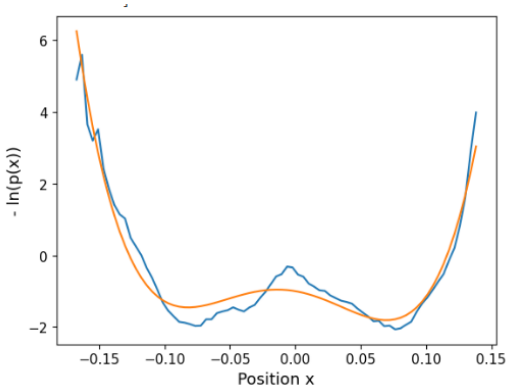
Following the approach from the previous chapter with cellular data, we train a Recurrent Neural Network (RNN) to distinguish between trajectories generated with and without



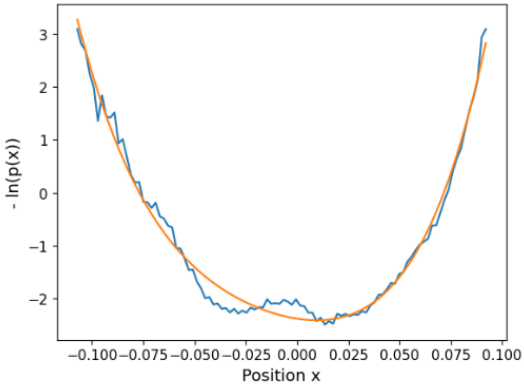
(a) Trajectory 1



(b) Trajectory 2



(c) Trajectory 3



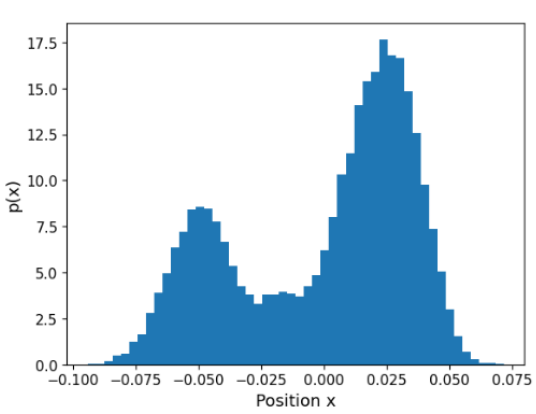
(d) Trajectory 4

Figure 17: $-\ln p(x)$ calculated directly from the data for the first four EPI trajectories (in blue) and the result of fitting Equation 34 to each trajectory (in orange).

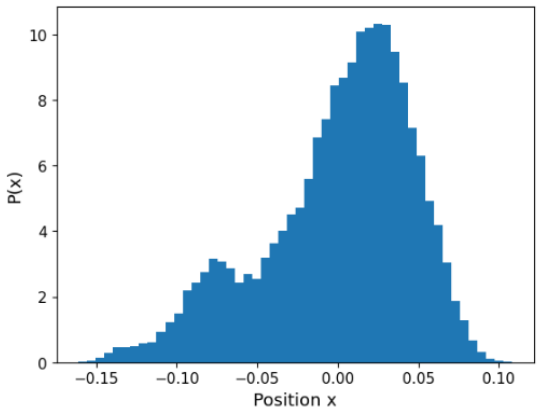
active noise.

The modeled trajectories used for training are created based on an AOUP model with a quartic potential. Each trajectory is generated by selecting random 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 characteristic time τ_a.

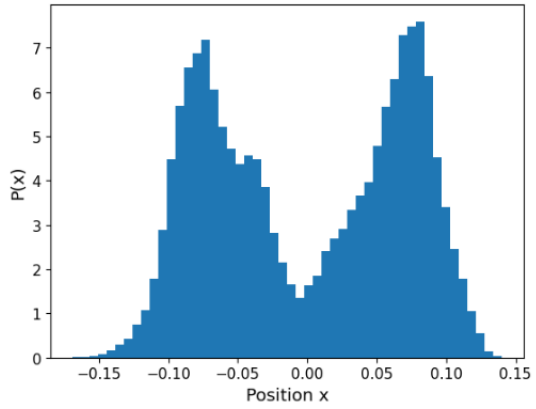
In total, we generate $5,000$ trajectories without active noise (where $D_a = 0$, and the other parameters are randomly selected) and $5,000$ trajectories with active noise (where D_a is a random value for each trajectory). This setup results in a binary classification task where the RNN must determine, for each trajectory, whether it contains active noise. The RNN outputs a probability between 0 and 1, representing the likelihood that a given



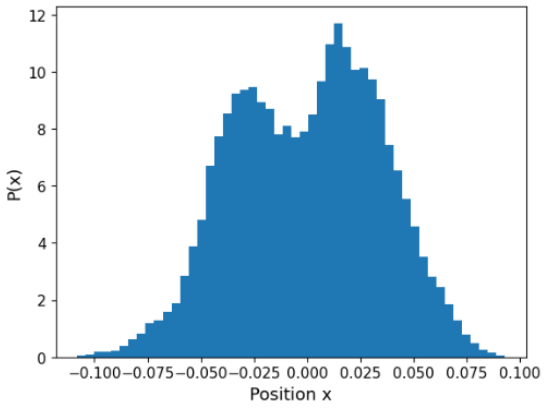
(a) Trajectory 1



(b) Trajectory 2



(c) Trajectory 3



(d) Trajectory 4

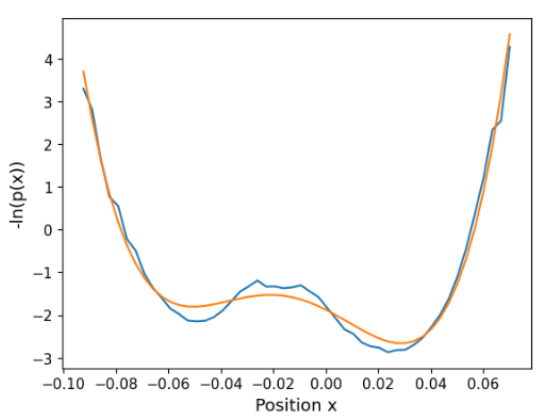
Figure 18: Histograms for four of the PrE nucleus trajectories.

trajectory exhibits active noise.

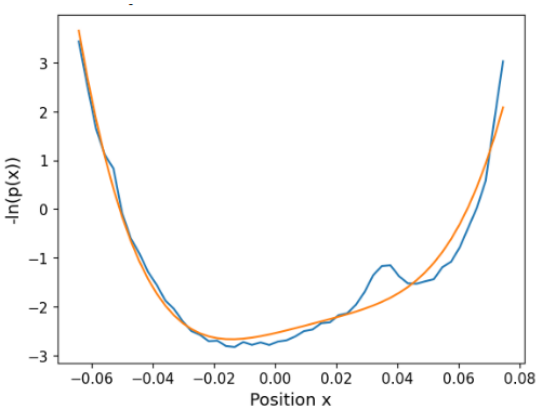
After training the RNN on the generated data, we evaluate its performance on unseen test data. The results are shown in Fig. 20. The mean probability for trajectories without active noise was 0.267, while for trajectories with active noise, it was 0.6727. Of the 2,000 test trajectories, 77.6% were correctly classified, with the RNN assigning probabilities greater than 0.5 for trajectories with active noise and less than 0.5 for those without.

6.2.2 Predicting D and D_a

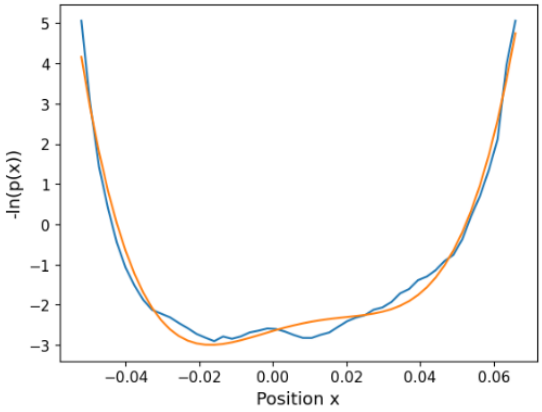
In addition to classifying active noise, we also trained two separate RNNs to predict the values of D and D_a. Each RNN was trained on trajectories generated as described in the previous section, with random values for the parameters k_1/γ, k_2/γ, k_3/γ, k_4/γ, and D_a (where D_a is always positive).



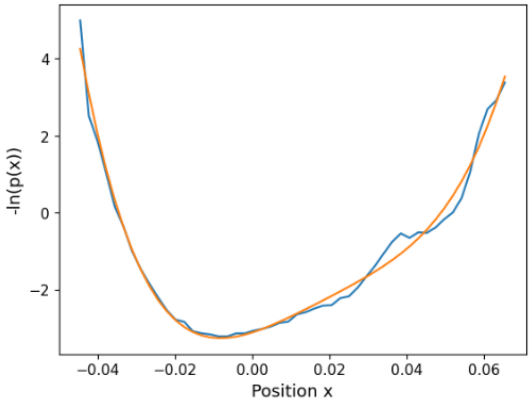
(a) Trajectory 1



(b) Trajectory 2



(c) Trajectory 3



(d) Trajectory 4

Figure 19: $-\ln p(x)$ calculated directly from the data for the first four PrE trajectories (in blue) and the result of fitting Equation 34 to each trajectory (in orange).

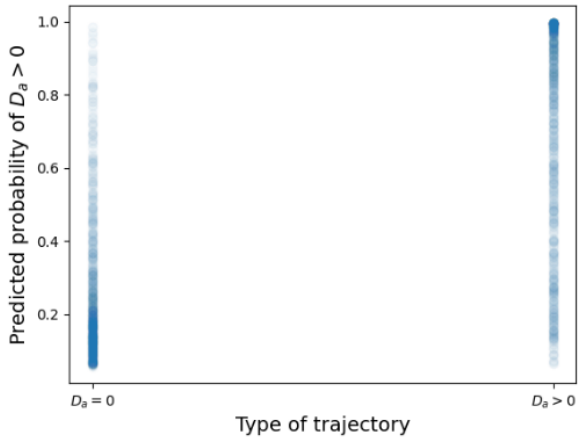
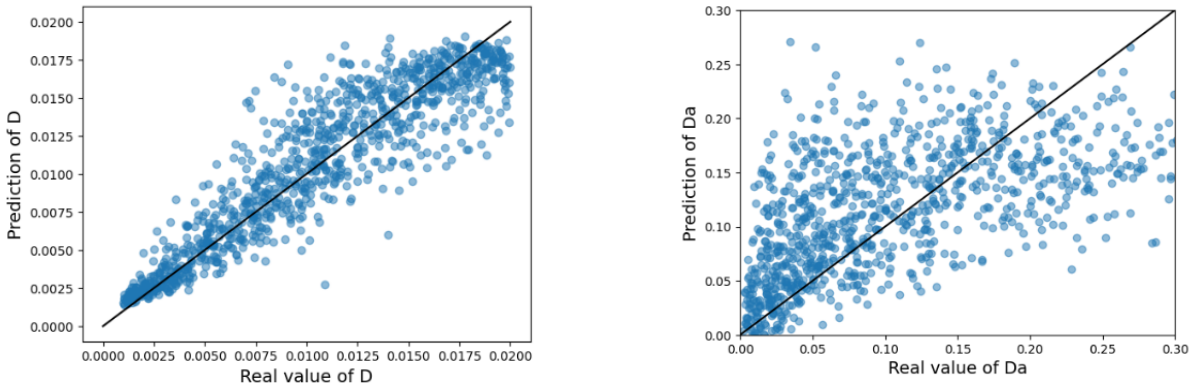


Figure 20: RNN predictions on simulated data of an active Ornstein-Uhlenbeck particle.

After training, we evaluated the RNNs on unseen test trajectories, comparing the predicted values of D and D_a to the true values used to generate the data. As observed with the cellular data, the predictions for D were significantly more accurate than those for D_a.



(a) Results for D

(b) Results for D_a

Figure 21: Comparison of predicted and true values of D and D_a for test data.

6.3 Predictions on real data

We now apply the trained RNNs for binary classification and for predicting D and D_a to both EPI and PRE data.

6.3.1 EPI nuclei

First, we use the RNN trained for binary classification to determine whether the 7 trajectories of EPI nuclei exhibit active or inactive behavior. Each trajectory is split into 80 sections, and the RNN is applied to each section individually. The results are shown in Fig. 22.
As seen in Fig. 22, the RNN does not classify the trajectories with high confidence, as it outputs probabilities close to 0.5 for most of them, with trajectory 7 having a slightly higher probability of 0.63.

Next, we use the RNNs to predict the values of D and D_a for each of the EPI trajectories, as shown in Fig. 23.

6.3.2 PRE nuclei

We repeat the same procedure for the 6 trajectories of PRE nuclei. We begin with the classifier RNN to determine whether the trajectories are active or inactive. Again, the

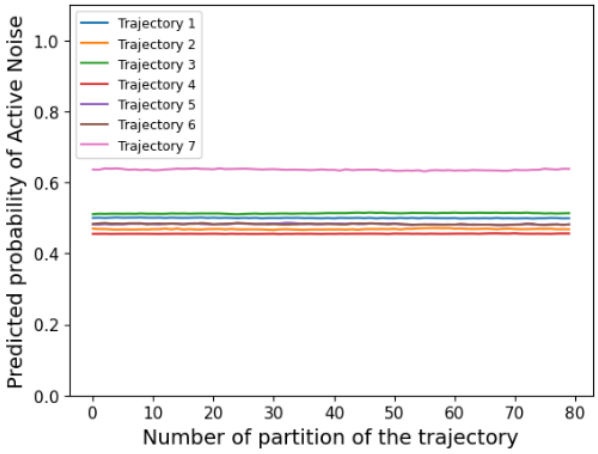
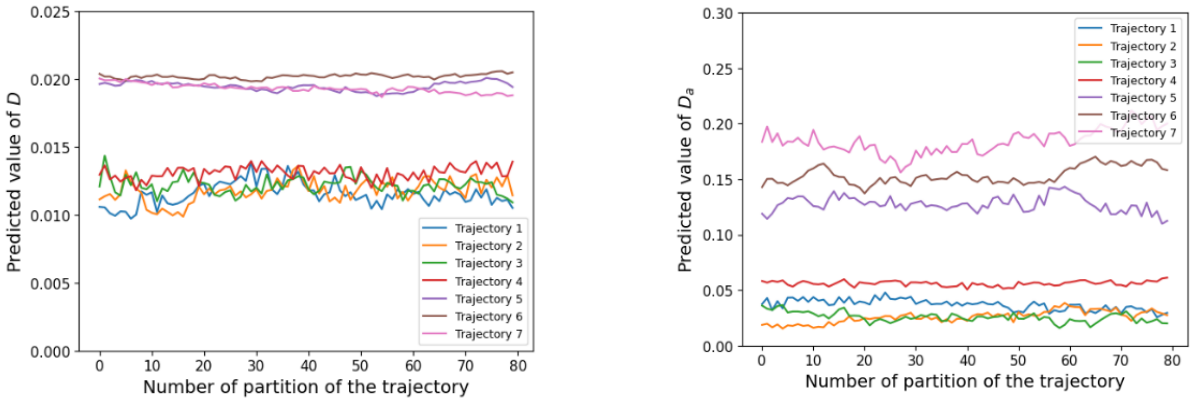


Figure 22: Results of applying the trained RNN to classify EPI nuclei trajectories as active or inactive. Each trajectory is split into 80 sections, and the RNN is applied to each one.



(a) Results for D

(b) Results for D_a

Figure 23: Predicted values for D and D_a on EPI nuclei trajectories.

trajectories are split into 80 sections, and the RNN is applied to each section separately. The results are shown in Fig. 24.
As can be seen in fig. 22, once again, the RNN doesn't confidently classify the trajectories as active or inactive. We also apply the RNNs to predict D and D_a to each of the trajectories, as shown in fig. 25.

6.4 Predicting the Potential

In this section, we construct an additional RNN to predict the potential $U(x)$ of a trajectory. Specifically, the RNN is designed to predict the values of k_1/γ, k_2/γ, k_3/γ, and k_4/γ for a given trajectory. These predictions are then combined with the predicted value

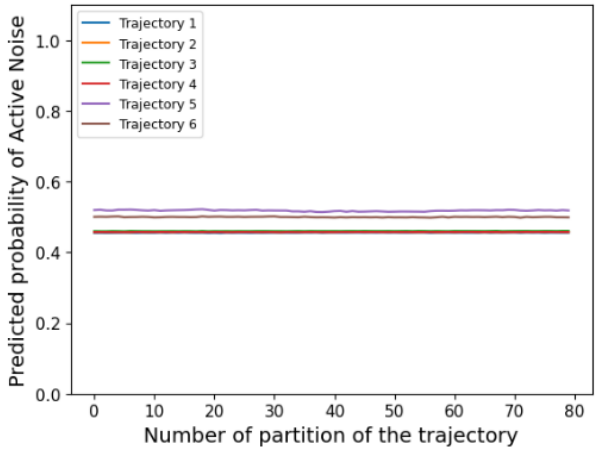
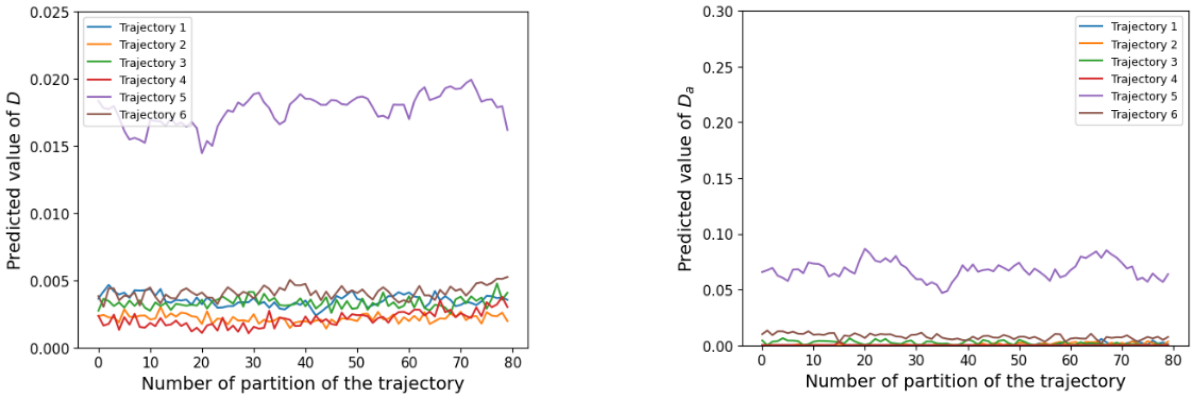


Figure 24: Results of applying the trained RNN to classify PRE nuclei trajectories as active or inactive. Each trajectory is split into 80 sections, and the RNN is applied to each one.



(a) Results for D

(b) Results for D_a

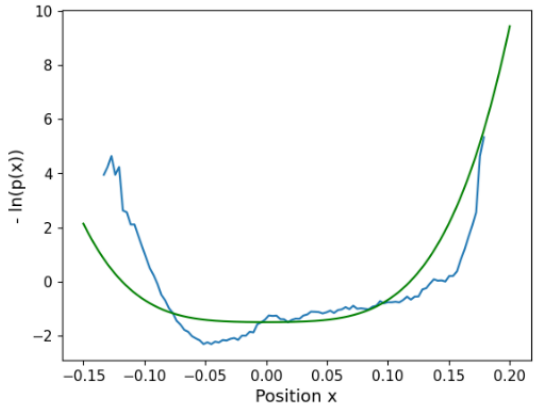
Figure 25: Predicted values for D and D_a on PRE nuclei trajectories.

of D to calculate the logarithm of the potential, as given in Eq. 34. We compare this predicted value of $-\ln p(x)$ with the true value, which is derived from the histogram of the trajectories. This comparison provides evidence that the RNNs are at least self-consistent.

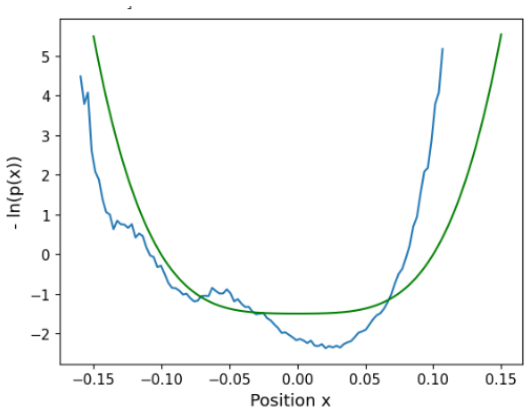
The trajectories are generated with random values for k_1/γ, k_2/γ, k_3/γ, k_4/γ, D, and D_a, and an RNN is trained to predict k_1/γ, k_2/γ, k_3/γ, and k_4/γ from the trajectory. The trained RNN is then applied to the real nuclei trajectories. The predictions for k_1/γ, k_2/γ, k_3/γ, and k_4/γ, along with the predicted value of D, are used to calculate $-\ln p(x)$ using Eq. 34.

In fig. 26 we show the results of $-\ln p(x)$ calculated using this RNN along with the graph of $-\ln p(x)$ obtained directly from the histogram data of the trajectories, for the first 4

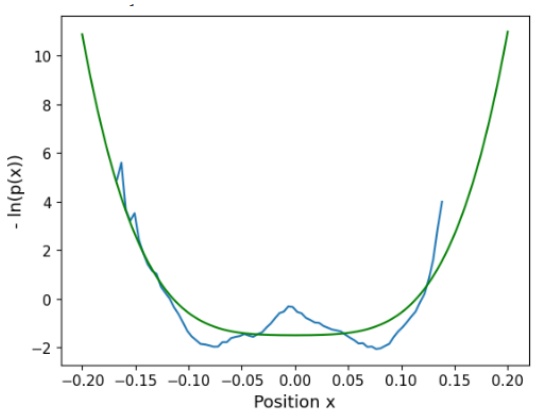
trajectories of EPI nuclei. Fig. 27 shows the same but for the PrE nuclei. We can see that the predictions using RNNs closely give the general shape of $-\ln p(x)$, failing only in minor details.



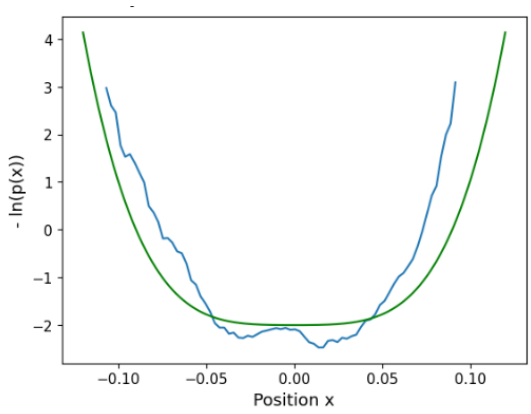
(a) Trajectory 1



(b) Trajectory 2



(c) Trajectory 3

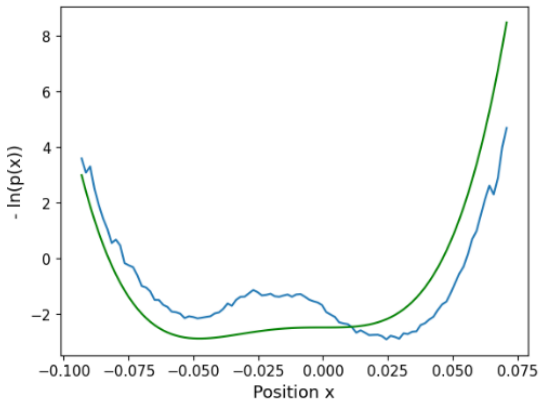


(d) Trajectory 4

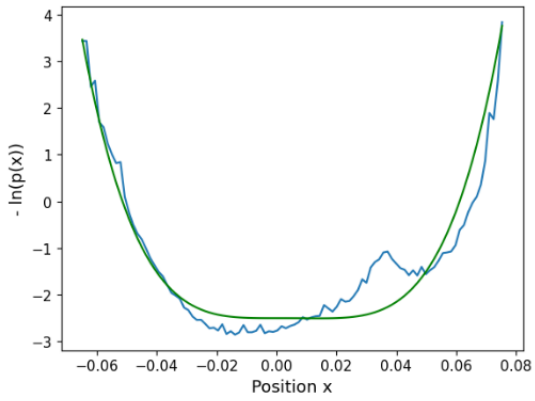
Figure 26: Predictions of $-\ln p(x)$ (in green) using Eq. 34 and an rnn to predict $k_1/\gamma, k_2/\gamma, k_3/\gamma, k_4/\gamma$, along with the prediction of D for the trajectories of EPI nuclei. We also plot the actual graph of $-\ln p(x)$ obtained directly from the histograms of the trajectories (in blue).

6.5 Power Spectrum Analysis

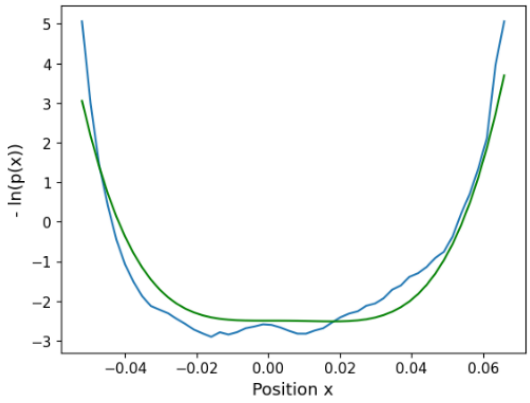
In contrast to the cellular data discussed in the previous chapter, we cannot apply the power spectrum analysis and the results from [24] to the nuclear data. Recall that in the power spectrum analysis, we compute the finite-time Fourier transform of a trajectory,



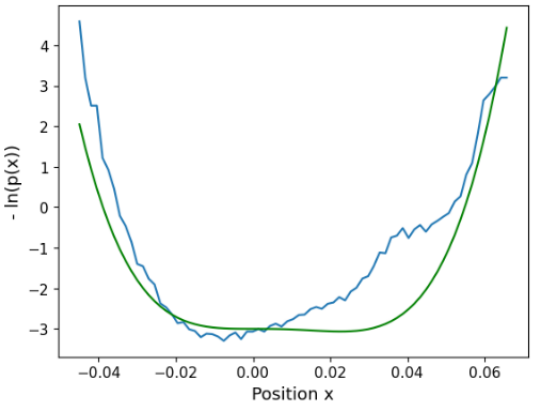
(a) Trajectory 1



(b) Trajectory 2



(c) Trajectory 3



(d) Trajectory 4

Figure 27: Predictions of $-\ln p(x)$ (in green) using Eq. 34 and an rnn to predict $k_1/\gamma, k_2/\gamma, k_3/\gamma, k_4/\gamma$, along with the prediction of D for the trajectories of PRE nuclei. We also plot the actual graph of $-\ln p(x)$ obtained directly from the histograms of the trajectories (in blue).

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. 15), 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. The problem is that for this data, Δt is not small enough to allow us to reach large enough values of ω so that we can have a reasonable analysis of $\omega^2 S_\omega$ and know if it has reached a steady value after growing monotonously or not.

7 Discussion

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 $t + \Delta t$:

$$\int_t^{t+\Delta t} \dot{x}(s)\,ds = \int_t^{t+\Delta t} f(x(s))\,ds + \sqrt{2D}\int_t^{t+\Delta t} \xi(s)\,ds. \tag{35}$$

The left hand side of Eq. 35 is

$$\int_t^{t+\Delta t} \dot{x}(s)\,ds = x(t+\Delta t) - x(t) := \Delta x. \tag{36}$$

We can integrate the first term of the right hand side of Eq. 35 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. 35, 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. 35, we get:

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

and therefore:

$$x(t+\Delta t) = x(t) + f(x(t))\Delta t + \Delta W(t). \tag{37}$$

Eq. 37 tells us how to numerically solve Langevin's equation. We first divide the time range $[0, \tau]$ on which we want to solve the equation 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. 37 repeatedly starting from $x(0) = x_0$.

The only remaining thing needed to use Eq. 37 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 $\langle \Delta W^2 \rangle = \Delta t$, so that 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: Theoretical path integral 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 $x(t)$ given an initial position x_0. For this derivation, we follow what is done in [42]. 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:

$$P(\xi_{\Delta t}) = \frac{1}{\sqrt{2\pi\Delta t}} e^{-\xi^2_{\Delta t}/2\Delta t}. \tag{38}$$

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:

$$P_\xi(\underline{x}|x_0)\mathcal{D}[\xi] = \left(\frac{1}{\sqrt{2\pi\Delta t}}\right)^N \exp\left(-\sum_{n=0}^{N-1}\frac{\xi_{\Delta t}(t_n)^2}{2\Delta t}\right)\mathcal{D}[\xi]$$
$$= \left(\frac{1}{\sqrt{2\pi\Delta t}}\right)^N \exp\left(-\sum_{n=0}^{N-1}\frac{[(\dot{x}(t_n)-a(x_n,t_n))\Delta t]^2}{2b^2\Delta t}\right)\mathcal{D}[\xi], \qquad (39)$$

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 $x'_n := \frac{x_n + x_{n+1}}{2}$. Using this, the time-evolution is expressed as:

$$x_{n+1} = x_n + a(x'_n, t_n)\Delta t + b\xi_{\Delta t}(t_n). \qquad (40)$$

We want to transform Eq. 39 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. 40 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 $\frac{\partial \xi_{\Delta t}(t_n)}{\partial x_{n+1}}$.
Therefore, continuing with Eq. 39, 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:

$$P(\underline{x}|x_0)d\underline{x} = C_0\exp\left(-\int_0^\tau dt\left[\frac{[(\dot{x}(t_n)-a(x_n,t_n))]^2}{2b^2} + \frac{1}{2}\frac{\partial}{\partial x'_n}a(x'_n,t_n)\right]\right), \qquad (41)$$

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 of observing a given trajectory $\underline{x}$ starting from an initial position x_0.

7.1 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 overdamped system 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. 41 (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

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

7.2 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. 41 but now with these two variables instead of only one, so that it becomes:

$$P(\underline{x}, \underline{\eta} \mid x_0, \eta_0) = C_1 \exp\left(- \int_0^\tau dt \left[\frac{(\dot{x}_t - g_t - \sqrt{2D_a}\eta_t)^2}{4D} + \frac{(\tau_a \dot{\eta}_t + \eta_t)^2}{2} + \frac{1}{2}\frac{\partial g_t}{\partial x} \right] \right), \tag{42}$$

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 coefficient comes from having two of the factors $\dfrac{1}{\sqrt{2\pi\Delta t}}$ from the Gaussian distribution of both noises. The factor $\dfrac{1}{\sqrt{2D}}$ 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 $\exp\left(- \displaystyle\int_0^\tau \frac{1}{2}\frac{\partial}{\partial \eta}\left(-\frac{1}{\tau_a}\eta \right) dt \right)$, 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 [15]. From the law of total probability, we know that we can integrate out η by doing:

$$p(\underline{x}|x_0) = \int \mathcal{D}\underline{\eta}\, d\eta_0\, p(\underline{x}, \underline{\eta}|x_0, \eta_0) p_0(\eta_0|x_0). \tag{43}$$

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:

$$p_0(\eta_0|x_0) = p_s(\eta_0) = \sqrt{\frac{\tau_a}{\pi}} e^{-\tau_a \eta_0^2}. \tag{44}$$

The distribution for η_0 is a Gaussian with mean 0 and standard deviation of $\dfrac{1}{\sqrt{2\tau_a}}$, as shown in Eq. 55. 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. 44 and Eq.42 into Eq. 43, (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:

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

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 [15]. 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 $\frac{\sqrt{2D_a}}{2D}(\dot{x}_t - g_t)$. 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:

$$p(\underline{x}|x_0) = C_1' K \exp\left(\frac{1}{2}\int_0^\tau dt \int_0^\tau dt'\ w_t^T \Gamma_\tau(t,t') w_{t'}\right) \int \mathcal{D}\overline{\eta} \exp\left(-\frac{1}{2}\int_0^\tau dt \int_0^\tau dt'\ (\eta_t + \epsilon_t)^T V_\tau(t,t')(\eta_{t'} + \epsilon_{t'})\right)$$
$$= C_1' K B \exp\left(\frac{1}{2}\int_0^\tau dt \int_0^\tau dt'\ w_t^T \Gamma_\tau(t,t') w_{t'}\right) \tag{45}$$

where we define B as:

$$B := \int \mathcal{D}\overline{\eta} \exp\left(-\frac{1}{2}\int_0^\tau dt \int_0^\tau dt'\ (\eta_t + \epsilon_t)^T V_\tau(t,t')(\eta_{t'} + \epsilon_{t'})\right) \tag{46}$$

7.2.1 Calculation of B

We still need to calculate the factor B of Eq.46 so that we can get the complete normalization factor. 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:

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

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

$$B = \int \mathcal{D}\overline{\eta} \exp\Big[-\frac{1}{2}\int_0^\tau \int_0^\tau dt dt'\ \delta(t-t')\big[-\tau_a^2\eta(t)\ddot{\eta}(t') + L\eta(t)\eta(t') +$$
$$+ \delta(t)(\tau_a\eta(t)\eta(t') - \tau_a^2\eta(t)\dot{\eta}(t')) + \delta(\tau)(\tau_a\eta(t)\eta(t') + \tau_a^2\eta(t)\dot{\eta}(t'))\big]\Big].$$

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:

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

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,

$$B = \int \mathcal{D}\overline{\eta} \exp\left[-\frac{1}{2}\int_0^\tau dt\ (\tau_a\dot{\eta}(t) + \sqrt{L}\eta(t))^2 + \frac{1}{2}BT''\right], \tag{47}$$

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. 47 as having the shape of an Onsager-Machlup integral for a system defined by the Langevin equation

$$\dot{y}(t) = -\frac{\sqrt{L}}{\tau_a}y(t) + \frac{1}{\tau_a}\xi(t). \tag{48}$$

For such a system, the probability of a given trajectory $\underline{y}$ is given by the Onsager-Machlup integral according to Eq. 41:

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

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. 47 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:

$$\sqrt{\frac{\tau_a}{\pi}}L^{1/4}\int \mathcal{D}\overline{y}\; p(\underline{y}|y_0)e^{-\tau_a\sqrt{L}y(0)^2} = 1. \tag{49}$$

We notice that this result is the same as we have for the η process in Eq. 47, but with a term $\dfrac{\sqrt{L}\tau}{2\tau_a} - \tau_a\sqrt{L}\eta(0)^2$, and without the $\dfrac{1}{2}BT''$ 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. 47 as:

$$\begin{aligned}
B &= \int \mathcal{D}\overline{\eta}\exp\left[-\frac{1}{2}\int_0^\tau dt\;(\tau_a\dot{\eta}(t)+\sqrt{L}\eta(t))^2 + \frac{1}{2}BT''\right] \\
&= \int \mathcal{D}\overline{\eta}\exp\left[-\frac{1}{2}\int_0^\tau dt\;(\tau_a\dot{\eta}(t)+\sqrt{L}\eta(t))^2 \;+\frac{\sqrt{L}\tau}{2\tau_a} - \tau_a\sqrt{L}\eta(0)^2 + \frac{1}{2}BT'' - \frac{\sqrt{L}\tau}{2\tau_a} + \tau_a\sqrt{L}\eta(0)^2\right] \\
&= \int \mathcal{D}\overline{\eta}\exp\left[-\frac{1}{2}\int_0^\tau dt\;(\tau_a\dot{\eta}(t)+\sqrt{L}\eta(t))^2 \;+\frac{\sqrt{L}\tau}{2\tau_a} - \tau_a\sqrt{L}\eta(0)^2\right]\exp\left[\frac{1}{2}BT'' - \frac{\sqrt{L}\tau}{2\tau_a} + \tau_a\sqrt{L}\eta(0)^2\right]
\end{aligned} \tag{50}$$

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 $\sqrt{\dfrac{\tau_a}{\pi}}L^{1/4}$ factor in Eq. 49, the result of this integral is $\dfrac{1}{C_0'}\sqrt{\dfrac{\pi}{\tau_a}}L^{-1/4}$. Therefore, we have:

$$B = \sqrt{\frac{\pi}{\tau_a}}L^{-1/4}\;\frac{\langle e^{BT''/2\,+\,\tau_a\sqrt{L}\eta(0)^2}\rangle e^{-\tau\sqrt{L}/(2\tau_a)}}{C_0'}. \tag{51}$$

7.2.2 Putting the Results Together

Now that we have the result for B, we can put everything together into equation 45 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:

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

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

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

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 48 and not the equation we had originally for η. Therefore, the statistics of η need to be obtained from equation 48 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:

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

$$= \frac{\sqrt{L}\tau_a}{\pi\sqrt{1-\rho^2}} \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \exp\left\{ -\frac{1}{2}\left[\left(\frac{2\sqrt{L}\tau_a}{1-\rho^2} - 2q\right)\eta_0^2 - \frac{4\sqrt{L}\tau_a\rho}{1-\rho^2}\eta_0\eta_\tau + \left(\frac{2\sqrt{L}\tau_a}{1-\rho^2} - 2q\right)\eta_{\tau^2}\right]\right.$$

$$= \frac{\sqrt{L}\tau_a}{\pi\sqrt{1-\rho^2}} \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \exp\left\{ -\frac{1}{2}\begin{pmatrix}\eta_0 & \eta_\tau\end{pmatrix}\begin{pmatrix}\frac{2\sqrt{L}\tau_a}{1-\rho^2} - 2q & -\frac{2\sqrt{L}\tau_a\rho}{1-\rho^2} \\ -\frac{2\sqrt{L}\tau_a\rho}{1-\rho^2} & \frac{2\sqrt{L}\tau_a}{1-\rho^2} - 2q\end{pmatrix}\begin{pmatrix}\eta_0 \\ \eta_\tau\end{pmatrix}\right\} d\eta_0 d\eta_\tau$$

The result of this Gaussian integral is $\frac{2\pi}{\sqrt{\det A}}$, 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:

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

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 $\rho = e^{-\sqrt{L}\tau/\tau_a}$.

7.2.3 Final Result

Therefore, the final result is:

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

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

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

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

where 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 $\zeta(t)$.

7.3 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 $\langle\eta(t)\eta(t')\rangle$. First we formally solve the equation for $\eta(t)$ by introducing an integrating factor e^{t/τ_a}:

$$\begin{aligned}
&\dot{\eta}(t) + \frac{1}{\tau_a}\eta(t) = \frac{1}{\tau_a}\zeta(t) \\
&\Rightarrow\ \dot{\eta}(t)e^{t/\tau_a} + \frac{1}{\tau_a}\eta(t)e^{t/\tau_a} = \frac{1}{\tau_a}\zeta(t)e^{t/\tau_a} \\
&\Rightarrow\ \frac{d}{dt}\left(\eta(t)e^{t/\tau_a}\right) = \frac{1}{\tau_a}\zeta(t)e^{t/\tau_a} \\
&\Rightarrow\ \int_0^t \frac{d}{ds}\left(\eta(s)e^{s/\tau_a}\right)ds = \int_0^t \frac{1}{\tau_a}\zeta(s)e^{s/\tau_a}ds \\
&\Rightarrow\ \eta(t)e^{t/\tau_a} - \eta(0) = \int_0^t \frac{1}{\tau_a}\zeta(s)e^{s/\tau_a}ds \\
&\Rightarrow \eta(t) = \eta(0)e^{-t/\tau_a} + \frac{1}{\tau_a}\int_0^t \zeta(s)e^{(s-t)/\tau_a}ds. \qquad (52)
\end{aligned}$$

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

$$\langle\eta(t)\rangle = \langle\eta(0)\rangle e^{-t/\tau_a}. \qquad (53)$$

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. 52 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:

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

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

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

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

$$\langle\eta(t)\eta(t')\rangle = e^{-(t+t')/\tau_a}\left(\langle\eta(0)^2\rangle - \frac{1}{2\tau_a}\right) + \frac{1}{2\tau_a}e^{-|t'-t|/\tau_a}. \tag{54}$$

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

$$\langle\eta(t)^2\rangle = e^{-2t/\tau_a}\left(\langle\eta(0)^2\rangle - \frac{1}{2\tau_a}\right) + \frac{1}{2\tau_a}. \tag{55}$$

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:

$$\langle\eta(t)^2\rangle = \frac{1}{2\tau_a}. \tag{56}$$

Therefore, if the system begins in steady state, the term $\langle \eta(0)^2 \rangle$ in Eq. 54 is equal to $\frac{1}{2\tau_a}$ and so the expression for the correlation in steady state reduces to:

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

7.4 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 $x(t)$:

$$\begin{aligned}
&\dot{x}(t) + kx(t) = \sqrt{2D_a}\eta(t) + \sqrt{2D}\xi(t) \\
&\Rightarrow \dot{x}(t)e^{kt} + kx(t)e^{kt} = \sqrt{2D_a}\eta(t)e^{kt} + \sqrt{2D}\xi(t)e^{kt} \\
&\Rightarrow \frac{d}{dt}\left(x(t)e^{kt}\right) = \sqrt{2D_a}\eta(t)e^{kt} + \sqrt{2D}\xi(t)e^{kt} \\
&\Rightarrow \int_0^t \frac{d}{ds}\left(x(s)e^{ks}\right) ds = \sqrt{2D_a}\int_0^t \eta(s)e^{ks}ds + \sqrt{2D}\int_0^t \xi(s)e^{ks}ds \\
&\Rightarrow x(t)e^{kt} = x(0) + \sqrt{2D_a}\int_0^t \eta(s)e^{ks}ds + \sqrt{2D}\int_0^t \xi(s)e^{ks}ds \\
&\Rightarrow\ x(t) = x(0)e^{-kt} + \sqrt{2D_a}e^{-kt}\int_0^t \eta(s)e^{ks}ds + \sqrt{2D}e^{-kt}\int_0^t \xi(s)e^{ks}ds.
\end{aligned} \tag{58}$$

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:

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

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. 58:

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

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:

$$\begin{aligned}
\langle x(t)x(t') \rangle &= \langle x(0)^2 \rangle e^{-k(t+t')} \; + \; 2D_a e^{-k(t+t')} \int_0^t \int_0^{t'} \langle \eta(s)\eta(s') \rangle e^{k(s+s')} ds' ds \\
&\quad + \; 2De^{-k(t+t')} \int_0^t \int_0^{t'} \langle \xi(s)\xi(s') \rangle e^{k(s+s')} ds' ds \qquad (59) \\
&= \langle x(0)^2 \rangle e^{-k(t+t')} + \frac{D_a}{\tau_a} e^{-k(t+t')} \int_0^t \int_0^{t'} e^{k(s+s')} e^{-|s'-s|/\tau_a} ds' ds \qquad (60) \\
&\quad + 2De^{-k(t+t')} \int_0^t \int_0^{t'} \delta(s-s') e^{k(s+s')} ds' ds. \qquad (61)
\end{aligned}$$

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 $[t, t']$.

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

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

$$\begin{aligned}
&= \int_0^t \left[\int_0^t e^{k(s+s')} e^{-|s'-s|/\tau_a} ds' + e^{\Lambda_+ s} \frac{1}{\Lambda_-} \left(e^{\Lambda_- t'} - e^{\Lambda_- t} \right) \right] ds \\
&= \int_0^t \int_0^t e^{k(s+s')} e^{-|s'-s|/\tau_a} ds' ds + \frac{1}{\Lambda_+} \left(e^{\Lambda_+ t} - 1 \right) \frac{1}{\Lambda_-} \left(e^{\Lambda_- t'} - e^{\Lambda_- t} \right) \\
&= \int_0^t \int_0^t e^{k(s+s')} e^{-|s'-s|/\tau_a} ds' ds \; + \; \frac{1}{\Lambda_+ \Lambda_-} \left(e^{\Lambda_+ t} - 1 \right) \left(e^{\Lambda_- t'} - e^{\Lambda_- t} \right). \qquad (62)
\end{aligned}$$

To finish the calculation, we still need to do the missing integral in Eq. 62:

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

Therefore, the complete result of the first integral in Eq. 61 is:

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

Second Integral
The second integral in Eq. 61 can be calculated directly:

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

Complete Result
Now that we have the results for both integrals in Eq. 61, we can substitute them to get the correlation:

$$\begin{aligned}
\langle x(t) x(t') \rangle = \langle x(0)^2 \rangle e^{-k(t+t')} &+ \frac{D}{k} e^{-k(t+t')} \left(e^{2kt} - 1 \right) \\
&+ \frac{D_a}{\tau_a} e^{-k(t+t')} \left\{ \frac{1}{\Lambda_+} \left[\frac{1}{2k} \left(e^{2kt} - 1 \right) - \frac{1}{\Lambda_-} \left(e^{\Lambda_- t} - 1 \right) \right] + \frac{1}{\Lambda_-} \left[\frac{e^{\Lambda_- t}}{\Lambda_+} \left(e^{\Lambda_+ t} - 1 \right) - \frac{1}{2k} \left(e^{2kt} - 1 \right) \right] \right. \\
&\left. + \frac{1}{\Lambda_+ \Lambda_-} \left(e^{\Lambda_+ t} - 1 \right) \left(e^{\Lambda_- t'} - e^{\Lambda_- t} \right) \right\}.
\end{aligned}$$

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

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

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

$$\begin{aligned}\langle x(t)x(t')\rangle &= e^{-k(t+t')}\left(\langle x(0)^2\rangle - \frac{D}{k} - \frac{D_a}{2k\tau_a\Lambda_+} + \frac{D_a}{\tau_a\Lambda_+\Lambda_-} + \frac{D_a}{2k\Lambda_-\tau_a}\right)\\ &+ e^{-k(t'-t)}\left(\frac{D}{k} + \frac{D_a}{2k\tau_a\Lambda_+} + \frac{D_a}{\Lambda_-\Lambda_+\tau_a} - \frac{D_a}{2k\Lambda_-\tau_a} - \frac{D_a}{\tau_a\Lambda_+\Lambda_-}\right)\\ &+ e^{-t/\tau_a - kt'}\left(-\frac{D_a}{\tau_a\Lambda_+\Lambda_-} - \frac{D_a}{\Lambda_-\Lambda_+\tau_a} + \frac{D_a}{\tau_a\Lambda_+\Lambda_-}\right) - e^{-t'/\tau_a - kt}\left(\frac{D_a}{\tau_a\Lambda_+\Lambda_-}\right) + e^{-1/\tau_a(t'-t)}\left(\frac{D_a}{\tau_a\Lambda_+\Lambda_-}\right)\\ &= \left(\langle x(0)^2\rangle + \frac{D - Dk\tau_a + D_a}{k(-1+k\tau_a)}\right)e^{-k(t+t')} + \left(\frac{D}{k} + \frac{D_a}{k(1-k^2\tau_a^2)}\right)e^{-k(t'-t)} + \frac{D_a\tau_a}{1-k^2\tau_a^2}\,e^{-t/\tau_a - kt'}\\ &+ \frac{D_a\tau_a}{1-k^2\tau_a^2}\,e^{-t'/\tau_a - kt} - \frac{D_a\tau_a}{1-k^2\tau_a^2}\,e^{-1/\tau_a(t'-t)} \qquad (63)\end{aligned}$$

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 $t' = t$:

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

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

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

Therefore, if we start from steady state, we have that $\langle x^2(0)\rangle = \frac{D + Dk\tau_a + D_a}{k(1+k\tau_a)}$. Substituting this into Eq. 63, and disregarding the terms that become 0 if t and t' are large enough, we get the steady state correlation:

$$\langle x(t)x(t')\rangle = \left(\frac{D}{k} + \frac{D_a}{k(1-k^2\tau_a^2)}\right)e^{-k(t'-t)} - \frac{D_a\tau_a}{1-k^2\tau_a^2}\,e^{-1/\tau_a(t'-t)} \qquad (64)$$

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*, 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] 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.

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

[14] 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.

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

[16] 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.

[17] 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.

[18] 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.

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

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

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

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

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

[24] 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.

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

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

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

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

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

[30] 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.

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

[32] 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.

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

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

[35] 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.

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

[37] Robin Schmidt. Recurrent neural networks (rnns): A gentle introduction and overview, 11 2019.

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

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

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

[41] 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.

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

[43] 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.

[44] 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.

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

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

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

[48] 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.

[49] 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.