Machine Learning: Science and Technology

Simulation-based inference of single-molecule force spectroscopy

IOP Publishing Mach. Learn.: Sci. Technol. 4 (2023) 025009 https://doi.org/10.1088/2632-2153/acc8b8

MACHINE LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
6 December 2022

REVISED
16 March 2023

ACCEPTED FOR PUBLICATION
29 March 2023

PUBLISHED
18 April 2023

PAPER

Simulation-based inference of single-molecule force spectroscopy

Lars Dingeldein[1,2], **Pilar Cossio**[3,4] **and Roberto Covino**[1,2,5,*]

[1] Frankfurt Institute for Advanced Studies, Frankfurt am Main 60438 Germany
[2] Department of Physics, Goethe University Frankfurt, Frankfurt am Main 60438, Germany
[3] Center for Computational Mathematics, Flatiron Institute, New York, NY 10010, United States of America
[4] Center for Computational Biology, Flatiron Institute, New York, NY 10010, United States of America
[5] International Max Planck Research School on Cellular Biophysics, Frankfurt am Main 60438 Germany
[*] Author to whom any correspondence should be addressed.

E-mail: covino@fias.uni-frankfurt.de

Keywords: simulation-based inference, likelihood-free, derivative-free, neural posterior estimation, single-molecule biophysics, force spectroscopy

Supplementary material for this article is available online

Abstract

Single-molecule force spectroscopy (smFS) is a powerful approach to studying molecular self-organization. However, the coupling of the molecule with the ever-present experimental device introduces artifacts, that complicate the interpretation of these experiments. Performing statistical inference to learn hidden molecular properties is challenging because these measurements produce non-Markovian time series, and even minimal models lead to intractable likelihoods. To overcome these challenges, we developed a computational framework built on novel statistical methods called simulation-based inference (SBI). SBI enabled us to directly estimate the Bayesian posterior, and extract reduced quantitative models from smFS, by encoding a mechanistic model into a simulator in combination with probabilistic deep learning. Using synthetic data, we could systematically disentangle the measurement of hidden molecular properties from experimental artifacts. The integration of physical models with machine-learning density estimation is general, transparent, easy to use, and broadly applicable to other types of biophysical experiments.

1. Introduction

Single-molecule experiments provide an invaluable tool for understanding how molecules self-organize in cells and complex materials [1]. These experiments quantify the dynamics of individual molecules, capturing their heterogeneity and stochasticity. They are instrumental in understanding molecular self-assembly phenomena, like folding, the process by which proteins, nucleic acids, and other polymers form well-defined 3D structures.

Single-molecule force spectroscopy (smFS) is a powerful approach to investigating the microscopic mechanisms of folding and other structural rearrangements [2]. It can reveal folded and unfolded states, short-lived intermediates, characterize the transition paths connecting metastable states and binding and unbinding events [3–11]. Typically, a globular biomolecule will mainly populate its folded state and only rarely unfold and refold again. In smFS, two handles are attached to the biomolecule and used to apply mechanical tension to it (figure 1(a)). Normally, devices such as optical tweezers, magnetic tweezers, and atomic force microscopes are used to generate force. This tension destabilizes the folded state and promotes unfolding. In experiments at constant force, the biomolecule is in quasi-equilibrium and repeatedly unfolds and refolds. By monitoring an order parameter, e.g. the molecule's extension, we could obtain a one-dimensional time series showing hopping between the folded (low extension) and unfolded (high extension) states (figure 1(b)). We could then estimate the populations and lifetimes of each state as a function of the applied force [12–15].

However, the influence of the measuring apparatus—a pulling device attached to the small molecule via long flexible linkers (figure 1(a))—significantly affects the measurements and complicates a quantitative

interpretation of smFS. Ideally, we would directly monitor the dynamics of the molecular extension x and measure a time series $\boldsymbol{x}_t$. In practice, we have only access to the measured extension q (figure 1(b)), a combination of the molecular and linker extensions. The measuring apparatus is a mesoscopic object, much larger and slower than the molecule. The linkers are flexible and respond relatively slowly to forces. Therefore, the time series of the measured extension $\boldsymbol{q}_t$ reports only indirectly on the molecular extension. Ignoring this effect leads to significant artifacts [16–24].

Ingenious methods exist to disentangle measuring artifacts from measurements [25–28] or to reduce the apparatus's distortion [29, 30]. Yet, these methods are often challenging to apply or lack generality. Some are only valid if the diffusion coefficient of the pulling device is as fast as the molecular one, which is generally not true. Formulating a general and systematic framework to extract reduced quantitative models from smFS that recapitulate the molecular thermodynamic and kinetic properties is still an open challenge [2].

Simulation-based inference (SBI) is a powerful technique to perform Bayesian inference to connect observations to mechanistic models [31]. SBI is particularly suited for systems with an intractable likelihood, i.e. with no closed analytical form, and computationally expensive to evaluate [32]. The main idea is to encode a parametric mechanistic model of an experimental observation into a simulator. Given specific parameter values, the simulator produces synthetic data. Parameters that lead to synthetic data close to the original observation are the most plausible ones explaining it. Advances in density estimation due to neural networks and deep learning enabled a new generation of powerful SBI methods, which produce surrogate models of the likelihood or posterior using simulated data [33]. SBI is a general approach [34, 35] and a growing field with broad applications ranging from particle physics [36] to cosmology and astrophysics [37–40], nuclear fusion [41], genomics [42], and neuroscience [43, 44].

Here, we develop a computational framework that performs Bayesian inference to build quantitative models from smFS experiments at constant force. The purpose of this work is to show how SBI can be applied to smFS and demonstrate its usefulness on challenging examples. We show how we can easily extract hidden molecular properties using only the measured time series $\boldsymbol{q}_t$. We overcome the intractable likelihood problem by using neural density estimation to directly estimate the Bayesian posterior. Our approach is general, conceptually transparent, easy to use, and broadly applicable to other biophysical experiments, which require analyzing measurements with intractable likelihoods.

2. Methods

2.1. Theoretical model of smFS experiments

The harmonic-linker model is a well-established model of smFS experiments [21, 23, 45]. In this model, a two-dimensional free energy surface describes the combined system of molecule and apparatus. The molecular (hidden) extension x is defined as a distance between two amino acids in the protein to which the linkers are attached. A smFS experiment returns the measured extension q, which reports the distance between the molecular linkers connecting the molecule and the experimental apparatus. The free energy surface $G(q,x)$ is:

$$G(q,x) = G_0(x) + \frac{\kappa_l}{2}(x-q)^2. \tag{1}$$

The first term $G_0(x)$ describes the molecular free energy profile and implicitly includes the constant pulling force of the apparatus applied in smFS experiments at constant-tension. We considered several models for G_0, as explained in the next section. The second term describes the spring-like coupling of the molecule to the apparatus by the linker. The parameter κ_l describes the stiffness of the linker, and $(x-q)$ is the linker extension. We assume that the diffusion is position independent and anisotropic, $D_x \neq D_q$. Whereas D_x is a molecular property, D_q characterizes the dynamics of the mesoscopic pulling device.

We obtained trajectories describing the time evolution the system from an initial position by simulating Brownian dynamics on the free energy surface $G(q,x)$. In the simulations, the system will spend most of the time in one of the metastable states while only rarely jump between them (figure 1(b)). To mimic the situation of a smFS, we kept only the measured trajectories $\boldsymbol{q}_t$.

In practice, for the model parameters to be physically meaningful and to have a direct comparison with the experimental measurement, one should match the units of D_x, D_q, and q with the experimental one, as well as the frequency at which the data are recorded.

2.2. Models of the molecular free energy $G_0(x)$

We used various models to describe the one-dimensional molecular free-energy profile $G_0(x)$.

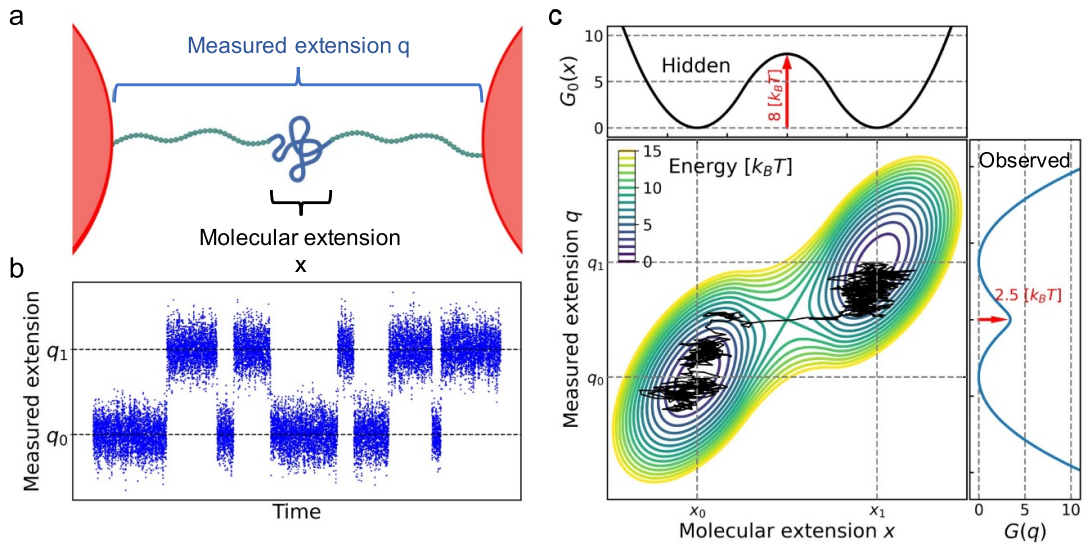

Figure 1. Schematic modelling of a smFS experiment at constant force. (a) Experimental setup. The red spheres represent the mesoscopic beads that are used in optical tweezer experiments to apply force. The molecule of interest (blue) is attached via flexible polymer linkers (green) to the pulling device. The measured extension q includes the length of the polymer linkers plus the extension of the molecule. (b) A time series $\boldsymbol{q}_t$ modelling an observation from a smFS experiment at constant force. The bi-stable trajectory shows rapid stochastic transitions between two states. (c) Example of the two-dimensional free energy surface $G(q,x)$. Isolines are drawn every 1 k_BT. The black curve shows a representative trajectory transitioning between the states. The upper panel shows the molecular free energy profile $G_0(x)$, with a high hidden barrier of $\Delta G^{\ddagger} = 8\ k_BT$. The right panel shows the observed free energy $G(q)$—the potential of mean force along q—with a projected barrier height of approximately 2.5 k_BT.

2.2.1. Symmetric-double-well

In the simplest case, the molecular free energy profile consists of a symmetric bi-stable double-well

$$G_0(x) = \Delta G^{\ddagger} \cdot f(x/x^{\ddagger}) \,, \tag{2}$$

where

$$f(x) = \begin{cases} -2x^2 & \text{for} \quad 0 \leqslant |x| \leqslant 1/2 \\ 2(|x|-1)^2 - 1 & \text{otherwise,} \end{cases} \tag{3}$$

with $\Delta G^{\ddagger}$ the energy barrier between the two minima at $x = 0$. The two meta-stable states are positioned at $x = \pm x^{\ddagger}$ and represent, for instance, the folded and unfolded states of a protein. We set $x^{\ddagger} = 1.5\,[q]$ in all simulations. For this model, the parameters that enter the prior and posterior are $\boldsymbol{\theta} = \{\Delta G^{\ddagger}, D_q/D_x, \kappa_1\}$.

2.2.2. Two- and three-well states surfaces

To follow a more general approach, we modelled the full two-dimensional free energy surface $G(q,x)$ using the negative logarithm of a linear combination of two-dimensional Gaussian functions. We did not have to explicitly define the potential of the linker molecule, which is implicitly encoded in the relative configuration of the Gaussians. We defined:

$$G(q,x) = -k_{\mathrm{B}}T\ \ln\left\{\sum_{i=0}^{K-1} \frac{\omega_i}{2\pi\,\sigma_{q,i}\sigma_{x,i}} \exp\left(-\frac{(x-x_i)^2}{2\sigma_{x,i}^2} - \frac{(q-q_i)^2}{2\sigma_{q,i}^2}\right)\right\}, \tag{4}$$

where q_i and x_i are the positions of the minima of the different states, and $\sigma_{q,i}$, $\sigma_{x,i}$ their widths, along q and x, respectively. The ω_i are the weights of the different states, with $\sum_i \omega_i = 1$.

We used three states to construct a nested model. The states where positioned along q at $q_0 = q_2 = -0.75$ and $q_1 = 0.75$. Thus, two of the states $(i = 0, 2)$ overlapped along the measured extension. We inferred the x-position of the second and third states $(i = 1, 2)$, while the first state was kept fixed at $x_0 = -1.5$. We set the weight for each state such that the projected free energy $G(q)$ is the same symmetric double-well. The weight of the second state was set to $\omega_1 = 0.5$, while the weight of the first and last state are set to $\omega_0 = \omega$ and $\omega_2 = 0.5 - \omega$. Changing ω the weight between the first and last states changes, while keeping $G(q)$ constant.

2.2.3. Rough double-well

To investigate the effect of moderate model misspecification, we produced synthetic experimental time series $\boldsymbol{q}_t$ generated from a rough (noisy) version of the symmetric double-well potential introduced in the previous section: $G_0^{\text{rough}}(x) = G_0(x) + \eta(x)$. The perturbation function $\eta(x)$ is a sum of different sinus functions, with different amplitudes a_i, frequencies b_i, and phase shifts c_i, i.e. $\eta(x) = A\sum_{i=0}^{N-1} a_i \sin(b_i x + c_i)$. We constructed random realizations of this rough potential by drawing the parameters from uniform distributions. The factor A is a scaling constant controlling the amount of noise added to the molecular free energy profile, and, therefore, the deviation from the idealized profile $G_0(x)$. We set $A = 0.7$ and $N = 15$ for all simulations.

2.2.4. Flexible spline profile

We used a cubic spline interpolation from the GNU Scientific Library to build a flexible model of the molecular free energy profile $G_0(x)$. We selected 15 points $\{(x_i, G_0(x_i))\}_{i=0}^{14}$, equally spaced out along the x-axis, and connected them pairwise with a third degree polynomial: $G_0(x) = a_i \cdot x^3 + b_i \cdot x^2 + c_i \cdot x + d_i \; \forall \, x \in [x_i, x_{i+1}]$. The first two and last two nodes had fixed values to avoid the system escaping the energy wells. The first and last node had a fixed values of $G_0(x_0) = G_0(x_{14}) = 70\ k_B T$, the second and second last $G_0(x_1) = G_0(x_{13}) = 30\ k_B T$.

The values of $G_0(x_i)$ for the inner eleven spline nodes $i \in \{2, \ldots, 12\}$ specify the details of free energy profile. Every new simulation propagates the system on a different spline $G_0(x)$. Therefore, for this system, the parameters that enter the prior and posterior are $\boldsymbol{\theta} = \{G_0(x_2), \ldots, G_0(x_{12}), D_q/D_x, \kappa_1\}$. After training, we sampled the posterior and aligned the free energy profiles, which are defined up to an additive constant.

2.3. Details of the simulator

The simulator integrated the equations of motion according to Brownian dynamics (over-damped Langevin) on a free energy surface $G(q,x)$. We used the the Euler–Maruyama integration scheme

$$q(t+\Delta t) = q(t) - \beta \partial_q G(q,x) \cdot D_q \Delta t + \sqrt{2 D_q \Delta t} \cdot R_q(t) \tag{5}$$

$$x(t+\Delta t) = x(t) - \beta \partial_x G(q,x) \cdot D_x \Delta t + \sqrt{2 D_x \Delta t} \cdot R_x(t)\ , \tag{6}$$

where D_x and D_q are the diffusion coefficients along the q and x-axis, respectively, and $R_q(t)$ and $R_x(t)$ are uncorrelated Gaussian random numbers with zero mean and a unit variance. We set the integration time step in all simulations to $D_x \Delta t = 5 \times 10^{-4}$. We decimated the raw trajectories to get time series reproducing synthetic experimental data for the measured and molecular extensions, $\boldsymbol{q}_t = \{q_{t\Delta\tau}\}_{t=1}^{M}$ and $\boldsymbol{x}_t = \{x_{t\Delta\tau}\}_{t=1}^{M}$, respectively, saving M time frames every $\Delta\tau$. For the symmetric double-well and the Langevin models we saved $M = 10^8$ frames, saving every $\Delta\tau = 100$; for the nested model $M = 2 \times 10^7$ frames, saving every $\Delta\tau = 50$; and for the cubic spline model $M = 10^6$, saving every $\Delta\tau = 100$.

For the under-damped Langevin simulations, we used the Langevin integrator in OpenMM [46]. We set the temperature to 500 K, the mass to 10^{-3} atomic units, and the timestep to 5×10^{-4} ps.

2.4. Time series featurization

Time series are structured (very) high-dimensional data, which cannot be directly used to perform neural density estimation. We must therefore project the original data $\boldsymbol{q}_t$ on a medium-dimensional set of features $\boldsymbol{y} = \boldsymbol{y}(\boldsymbol{q}_t)$. We can either use summary statistics, which might be already available in a given scientific domain, or use an additional neural network that extracts features from the data, e.g. an encoder. Here, we chose to use summary statistics.

For the double-well models, we described each time series $\boldsymbol{q}_t$ with 25 features y_i. We used the number of observed transitions between metastable states per unit time—an estimate of the microscopic rates—the first four statistical moments of the distribution of observed positions $\rho(\{q_i\})$, and of the distribution of displacements $\rho(\{\Delta q_k\})$, with $\Delta q_k = q_{i+k} - q_i$ calculated at five different lag-times $k = [1, 10, 100, 10\,000, 100\,000]$. We estimated the number of transitions based on changes in the running mean $\overline{q}_i = 1/w \sum_{j=-w/2}^{w/2} q_{i+j}$. The parameter w determines the window size and affects the estimate of the number of jumps. However, the final inference does not strongly depend on the estimated rate.

For the complex spline model, we used the transition matrices $T_{ij}(\Delta\tau)$ for different lag times $\Delta\tau$ as summary statistics. To compute the transitions matrix $T_{ij}(\Delta\tau)$ the trajectory $\boldsymbol{q}_t$ was binned in 20 equally spaced bins. For each lag time, we populated the transition matrix counting the transitions between bins i and j. The matrix was normalized to one along the columns. We computed the transition matrix for the lag times $\Delta\tau = [1, 10, 100, 1000, 10\,000, 100\,000]$. We did not use the rate as a feature for simulations obtained on the complex spline molecular free energy profile.

2.5. Details of SBI

We used SBI to perform Bayesian inference with an intractable likelihood [31]. In particular, we used neural posterior estimation (NPE) [47], where we approximate the posterior $p(\boldsymbol{\theta}|\boldsymbol{y}(\boldsymbol{q}_t))$ from simulated data with a neural network-based conditional density estimators $f_\phi(\boldsymbol{\theta}|\boldsymbol{y}(\boldsymbol{q}_t))$. The neural network model can vary depending on the specific problem. We used mixture density networks (MDNs) and normalizing flows.

Relatively simple posterior distributions can be approximated with MDNs [48]. MDNs are a general framework to approximate conditional densities with a superposition of K Gaussians,

$$f_\phi(\boldsymbol{\theta}|\boldsymbol{y}) = \sum_{k=1}^{K} \alpha_k \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{m}_k, \boldsymbol{S}_k), \tag{7}$$

where the means $\boldsymbol{m}_k$, covariance matrices $\boldsymbol{S}_k$ and mixing coefficients $\{\alpha_k\}$ are all non-linear functions of the observation $\boldsymbol{y}$, approximated by a neural network of parameters ϕ. To train the network, we maximised the average log probability $\frac{1}{N}\sum_{i=1}^{N} \log f_\phi(\boldsymbol{\theta}^{(i)}|\boldsymbol{y}^{(i)}(\boldsymbol{q}_t^{(i)}))$ w.r.t. ϕ on the training set $\mathcal{D} = \{(\boldsymbol{\theta}^{(i)}, \boldsymbol{y}^{(i)}(\boldsymbol{q}_t^{(i)})\}_{i=1}^{N}$.

For more complex models, we instead used normalizing flows, an alternative approach to estimating conditional densities that offer more flexibility [43, 49]. A normalizing flow is a series of invertible mappings to transform a simple base distribution into a complex target distribution [50]. We used the neural spline flow, which uses cubic splines parameterized by neural networks to model f_ϕ [51]. We used the implementation of both MDNs and neural spline flows available at the SBI package [47].

We trained both the MDN and the neural spline flow using the Adam optimizer. We adjusted the specific training settings and hyper-parameters for each problem separately (appendix). We terminated the training after the validation loss did not improve for a given number of epochs.

2.6. Code

We generated, analysed, and visualized the data with custom code based on NumPy [52], SciPy [53], Numba [54], Cython [55], Pytorch [56] and Matplotlib [57]. We performed the spline interpolation using the implementation of the GNU Scientific library [58]. We used the SBI algorithm NPE and the implementation of the MDN and the neural spline flow from the SBI-Toolkit [47].

3. Results

3.1. Diffusive models of smFS with hidden degrees of freedom

We aim to extract reduced quantitative models from smFS experiments based on a diffusive Brownian dynamics on a two-dimensional free energy landscape [59]. The experiment will measure the one-dimensional time series of the measured extension $\boldsymbol{q}_t = \{q_{t\Delta\tau}\}_{t=1}^{M}$, recorded with a lag-time $\Delta\tau$ and containing M data points. The measured extension will indirectly report on the molecular extension, described by the time series $\boldsymbol{x}_t = \{x_{t\Delta\tau}\}_{t=1}^{M}$. Ideally, from $\boldsymbol{x}_t$ we could estimate a probability distribution $P(x)$ at a given force and then get the force-dependent free energy profile $G(x) = -k_\mathrm{B}T\log P(x)$, where T is the absolute temperature and k_B is Boltzmann's constant. We could also estimate the diffusion coefficient D_x from the fluctuations of x [59]. In practice, all these quantities are hidden by the compounded dynamics of the measuring apparatus.

We consider here a well-established minimal model to describe the joint dynamics of the molecule and apparatus, first introduced by Hummer and Szabo [21, 23, 45], where $\boldsymbol{q}_t$ and $\boldsymbol{x}_t$ are diffusive processes on the two-dimensional free energy surface $G(q,x) = G_0(x) + \frac{\kappa_1}{2}(x-q)^2$ (figure 1(c)). The molecule's extension x diffuses with diffusion coefficient D_x on the molecular free energy profile $G_0(x)$. The measured extension q is coupled to x by an harmonic linker term, with stiffness κ_1, and diffuses with D_q. Both parameters describe the pulling device's properties. Let us first consider the simple case where $G_0(x)$ is an ideal symmetric double-well, with a barrier of height $\Delta G^\ddagger$ separating folded and unfolded states. The challenge becomes to estimate the parameters $\boldsymbol{\theta} = \{\Delta G^\ddagger, D_q/D_x, \kappa_1\}$ by using only the measured extension $\boldsymbol{q}_t$. By naively estimating the free energy profile $G(q)$ from $\boldsymbol{q}_t$, we would obtain a significantly biased value of the free energy barrier (figure 1(c)).

Bayesian inference provides a general framework to estimate the hidden molecular parameters from the measured extension. The result of the inference is the posterior $p(\boldsymbol{\theta}|\boldsymbol{q}_t)$, a probability distribution that quantifies how much the parameter values are compatible with the observed trajectory $\boldsymbol{q}_t$. The posterior distribution is given by Bayes' theorem

$$p(\boldsymbol{\theta}|\boldsymbol{q}_t) = \frac{p(\boldsymbol{q}_t|\boldsymbol{\theta})\cdot p(\boldsymbol{\theta})}{\int p(\boldsymbol{q}_t|\boldsymbol{\theta}')\cdot p(\boldsymbol{\theta}')\mathrm{d}\boldsymbol{\theta}'}, \tag{8}$$

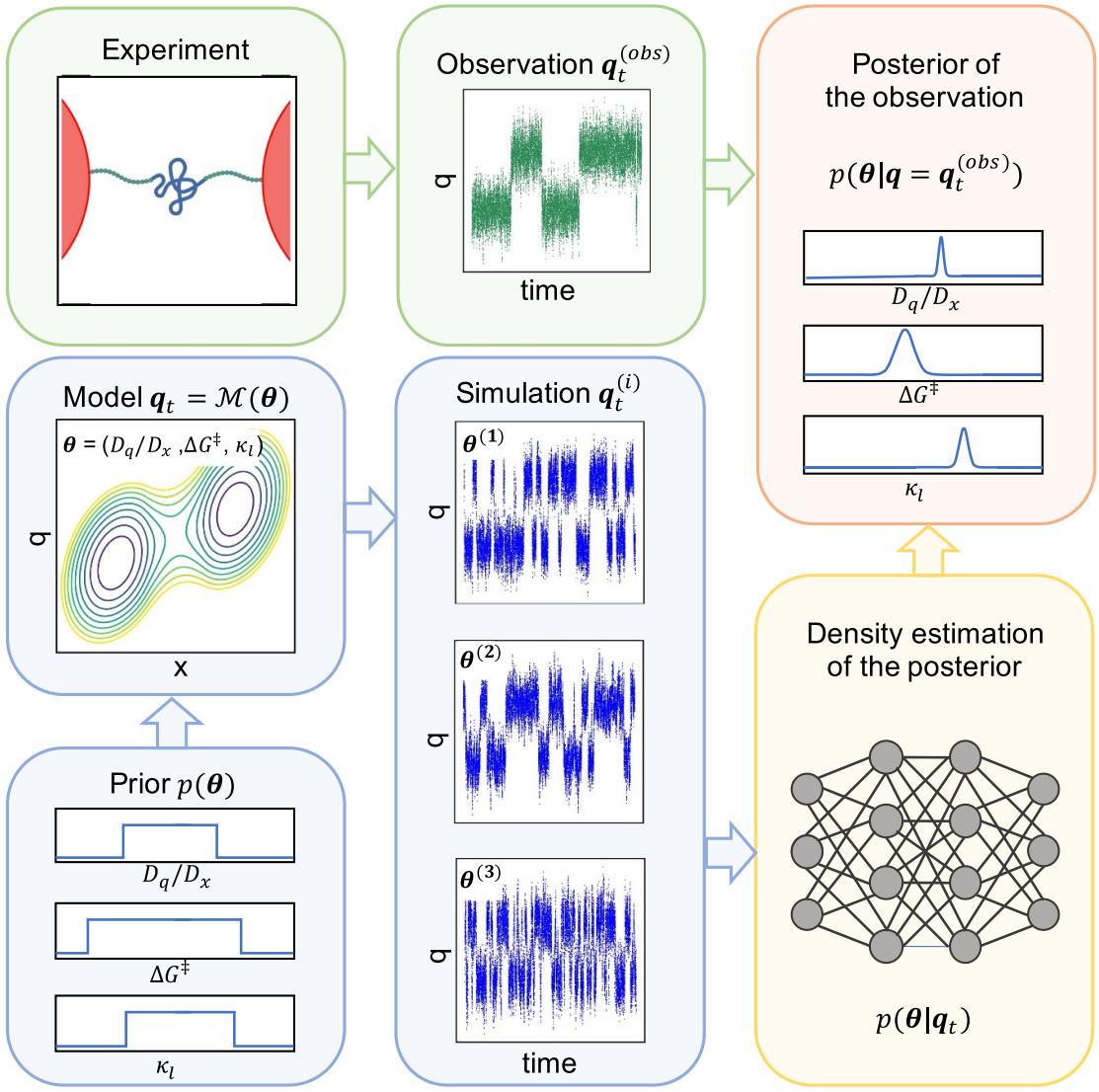

Figure 2. Workflow of simulation-based inference of smFS experiments. Given an experimental observation $\boldsymbol{q}_t^{(\mathrm{obs})}$, we want to explain it with a parametric mechanistic model $\mathcal{M}(\boldsymbol{\theta})$, encoded in a simulator. The parameters $\boldsymbol{\theta}^{(\mathrm{i})}$ are drawn from the prior distribution $p(\boldsymbol{\theta})$. The simulator takes parameters drawn from the prior and generates synthetic observations $\boldsymbol{q}_t^{(\mathrm{i})}$. The synthetic observations and corresponding parameters are used to train a conditional density estimator to approximate the posterior $p(\boldsymbol{\theta}|\boldsymbol{q}_t)$. Evaluating the posterior, $p(\boldsymbol{\theta}|\boldsymbol{q}_t = \boldsymbol{q}_t^{(\mathrm{obs})})$, we obtain the most plausible parameters explaining a given observation.

where $p(\boldsymbol{q}_t|\boldsymbol{\theta})$ is the likelihood of the data given the model, and $p(\boldsymbol{\theta})$ is the prior, which encodes all previous knowledge of $\boldsymbol{\theta}$. The normalization at the denominator is the model's evidence.

Even though often parameter inference relies on likelihood optimization, the likelihood is intractable in many cases of practical interest, even for minimal models. Here, the likelihood for $\boldsymbol{q}_t$ is a marginalization (projection) of the full likelihood, $p(\boldsymbol{q}_t|\boldsymbol{\theta}) = \int \mathcal{D}\boldsymbol{x}_t p(\boldsymbol{q}_t, \boldsymbol{x}_t|\boldsymbol{\theta})$, which is a path-integral over all possible hidden trajectories $\boldsymbol{x}_t$. It is, in general, analytically intractable and computationally costly. This significantly hinders conventional approaches that require many repeated evaluations of the likelihood or its gradient. SBI is a powerful way to perform Bayesian inference avoiding the evaluation of intractable likelihoods [31]. NPE, a specific SBI algorithm, aims to directly estimate the posterior from the data [33, 43, 49].

3.2. NPE of smFS

The main ingredients of SBI are an experimental observation, a simulator, a prior, and a suitable density estimator (figure 2). The simulator $\mathcal{M}(\boldsymbol{\theta})$ is a computer program encoding a parametric model that should explain the observed data. The simulator implicitly encodes the model's likelihood—even if intractable. For any parameter choice $\boldsymbol{\theta}^{(i)}$, the simulator samples the implicit likelihood producing synthetic data $\boldsymbol{q}_t^{(i)} \sim \mathcal{M}(\boldsymbol{\theta}^{(i)})$, which, ideally, should reproduce the experimental observation $\boldsymbol{q}_t^{(\mathrm{obs})}$. Drawing N parameter

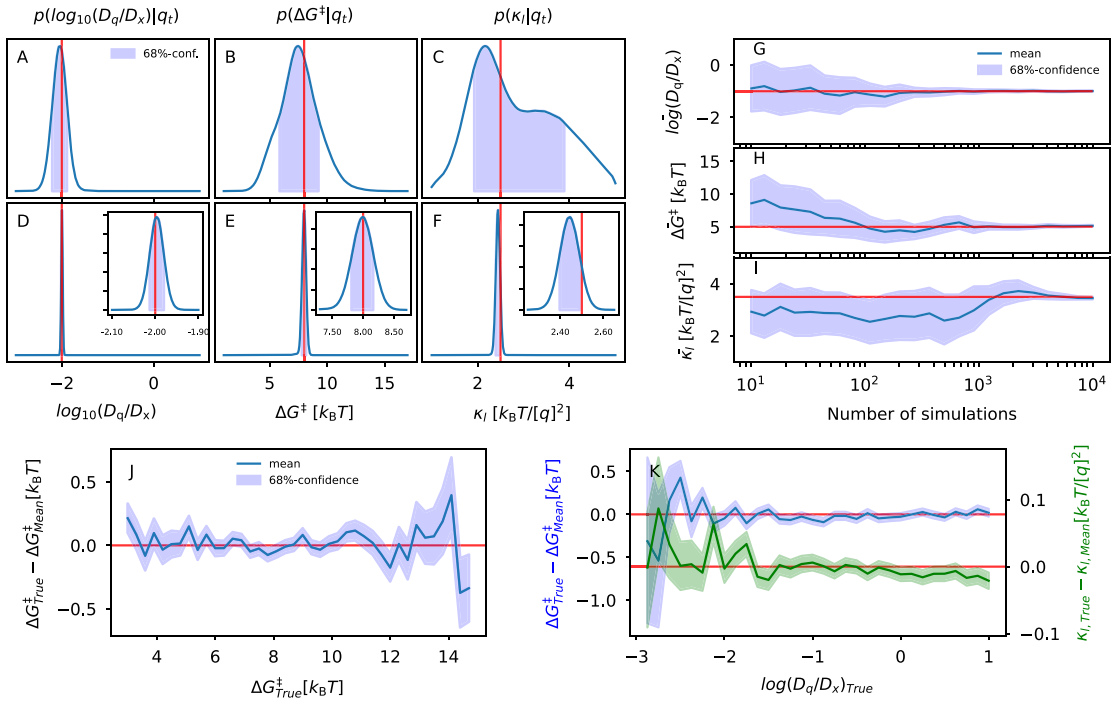

Figure 3. Neural posterior estimation of smFS at constant force. (a)–(f) Posterior marginal distributions trained on 600 (a)–(c) and 6000 (d)–(f) simulations. All posteriors are evaluated on the same observation, which was computed with the true parameters indicated by the red vertical lines. The blue shaded area represents 68% of the marginal density, corresponding to a 1σ confidence interval. The insets show a zoom-in of the posterior marginals. (g)–(i) Posterior marginals as a function of increasing numbers of simulations. For each number of simulations, we trained ten independent posteriors using different training data. The blue line represents the average of the mean of the posteriors, while the blue shaded area is the average of the 1σ confidence intervals. The observation was generated with the true parameters indicated with the horizontal red line. (j) Difference between best estimate and true molecular barrier height as a function of increasing true barrier height. The synthetic observations varied only in the barrier height, while we kept $\log(D_q/D_x) = -1$ and $\kappa_\mathrm{l} = 2\ k_\mathrm{B}T[q]^{-2}$ fixed. We used the mean of the posterior as the best estimate (blue line). (k) Difference between the best estimate and true molecular barrier height $\Delta G^\ddagger = 7\ k_\mathrm{B}T$ as a function of the ratio of diffusion coefficients (blue line). Difference between the best estimate and true linker stiffness $\kappa_\mathrm{l} = 2\ k_\mathrm{B}T[q]^{-2}$ as a function of the ratio of diffusion coefficients (green line). We estimated the 1σ confidence interval as the 68% of the posterior marginal density.

samples from the prior $p(\boldsymbol{\theta})$, the simulator produces a data-set $\mathcal{D} = \{(\boldsymbol{\theta}^{(i)}, \boldsymbol{q}_t^{(i)}\}_{i=1}^N$. In NPE, we use a neural network of parameters ϕ to model a conditional density estimator $f_\phi(\boldsymbol{\theta}|\boldsymbol{q}_t)$ and train it on $\mathcal{D}$. The trained network is a surrogate of the posterior. It allows us to perform inference for any given observation, $p(\boldsymbol{\theta}|\boldsymbol{q}_t = \boldsymbol{q}_t^{(\mathrm{obs})}) \approx f_\phi(\boldsymbol{\theta}|\boldsymbol{q}_t = \boldsymbol{q}_t^{(\mathrm{obs})})$, at a negligible computational cost.

We applied NPE on synthetic constant-force smFS experiments to extract quantitative models and studied how well the inference matched the ground truth parameters. In every numerical experiment, we ran Brownian dynamics on $G(q,x)$ with a given set of true parameters $\boldsymbol{\theta}^{(o)}$ to obtain a synthetic observation $\boldsymbol{q}_t^{(\mathrm{obs})}$. We discarded the corresponding hidden time series $\boldsymbol{x}_t$ and projected all time series $\boldsymbol{q}_t$ on a medium-dimensional feature space (see Methods for more details). We used a uniform prior $p(\boldsymbol{\theta})$ defined in a reasonable range of values.

NPE extracts hidden parameters from incomplete observations with high accuracy and precision. In the first computational experiment, we trained the posterior on only 600 Brownian simulations (figures 3(a)–(c)). To visualize the inference's quality, we plot the marginal posterior distributions of every single parameter θ_i, having integrated out all the remaining ones, $p(\theta_i|\boldsymbol{q}_t = \boldsymbol{q}_t^{(\mathrm{obs})}) = \int p(\boldsymbol{\theta}|\boldsymbol{q}_t = \boldsymbol{q}_t^{(\mathrm{obs})}) \cdot \prod_{j\neq i} \mathrm{d}\theta_j$. The inference is remarkably good, especially for D_q/D_x and $\Delta G^\ddagger$. The posterior's peak is close to the true values for all three parameters. The posterior's spread provides the uncertainty of the inference. While this is reasonably precise for $\Delta G^\ddagger$ and D_q/D_x, it is not for κ_l. Reducing the uncertainty requires more simulated data. Training over 6000 Brownian trajectories led to an exceptional inference (figures 3(d)–(f)).

Obtaining high-quality inference is computationally efficient. We studied the inference quality as a function of the number of simulations (figures 3(g)–(i)). A few thousand simulations are sufficient to provide very good estimates of all three parameters. D_q/D_x is the most accessible parameter to extract, probably because contained in the statistics of local fluctuations. The stiffness κ_l is the most challenging parameter to extract, and its uncertainty decreases significantly only after approx. 1000 simulations.

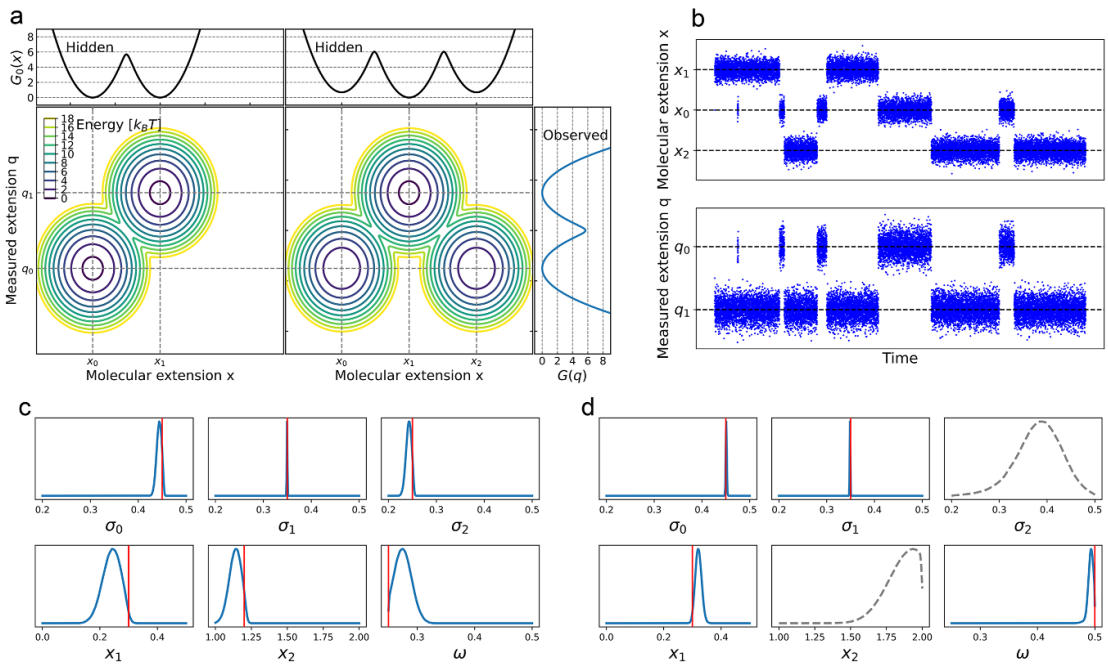

Figure 4. Model comparison. (a) We compare the inference of an observation using a two- or three-state models. In the latter, one state is hidden due to the projection on q. The molecular free energy profile $G(x)$ shows two and three states, whereas $G(q)$ is the same and only exhibits two states. (b) Brownian simulations in the three-state model show jumps between three states in x, but only two states in q. (c), (d) Posterior marginals obtained by NPE trained on simulations of a nested model. In general, the model contains three Gaussian states. The parameter ω determines their relative weight. For $\omega = 0.5$, the model contains only two Gaussian states. (c) Marginals of the posterior $p(\boldsymbol{\theta}|\boldsymbol{q}_t = \boldsymbol{q}_t^{(3)})$, evaluated on a synthetic observation produced by a three-state model. $p(\omega)$ favours a three-state model to explain this observation. (d) Marginals of the posterior $p(\boldsymbol{\theta}|\boldsymbol{q}_t = \boldsymbol{q}_t^{(2)})$, evaluated on a synthetic observation produced by a two-state model. $p(\sigma_2)$ and $p(x_2)$ are dashed because not necessary.

3.3. Amortized inference

Having trained a neural network to approximate the conditional posterior, we can perform inference for new observations without running any additional simulations. The inference is amortized. Further posterior evaluations only require a forward pass of the trained network, which takes milliseconds.

We used the posterior trained over 60 000 simulations and systematically investigated the inference's quality. The estimate of the hidden barrier height is excellent for values between 4 and 13 k_BT (figure 3(j)). Lower barriers do not produce clear transitions between the two states. Larger barriers cause poor transition statistics in the training set. Yet, the most significant error is only a fraction of k_BT. We could obtain excellent estimates of the barrier height and linker stiffness varying D_q/D_x over four orders of magnitude (figure 3(k)). The quality degrades for very small values of D_q. These data show that SBI allows us to extract accurate diffusive models from synthetic data of smFS experiments over a broad range of parameters.

3.4. Hidden states and model comparison

Bayesian inference enables model comparison, quantifying how well two alternative models $\mathcal{M}_1$ and $\mathcal{M}_2$ explain the observed data. Usually, this comparison relies on the calculation of the Bayes factor, defined as the ratio of the models' evidences introduced in equation (8). Intractable likelihoods—like in our case—hinder calculating Bayes factors. Here, we performed model comparison by formulating nested models, where $\mathcal{M}_1$ is a particular case of the more general model $\mathcal{M}_2$.

An interesting question in the context of smFS is whether we can detect a hidden metastable state. By measuring a single quantity—the observed extensions q—we project an inherently high-dimensional dynamical system on a one-dimensional coordinate. If the projection of two states leads to similar values of q, we might not resolve one of them.

To investigate this problem in a simplified setting, we considered two alternative models, $\mathcal{M}_1$ and $\mathcal{M}_2$. $\mathcal{M}_1$ is defined by free energy surface $G_1(q,x)$, which contains two states (figure 4(a)). The first state is centered at (q_0,x_0), while the second is at (q_1,x_1). This model is very similar to the harmonic-linker model studied in the previous section, with the linker implicitly modeled by the relative position and the width of the two states. We then considered a model $\mathcal{M}_2$ with the surface $G_2(q,x)$, defined by three states. The two bottom states are centered at the coordinates (q_0,x_0) and (q_0,x_2); and the top state is centered at (q_1,x_1)

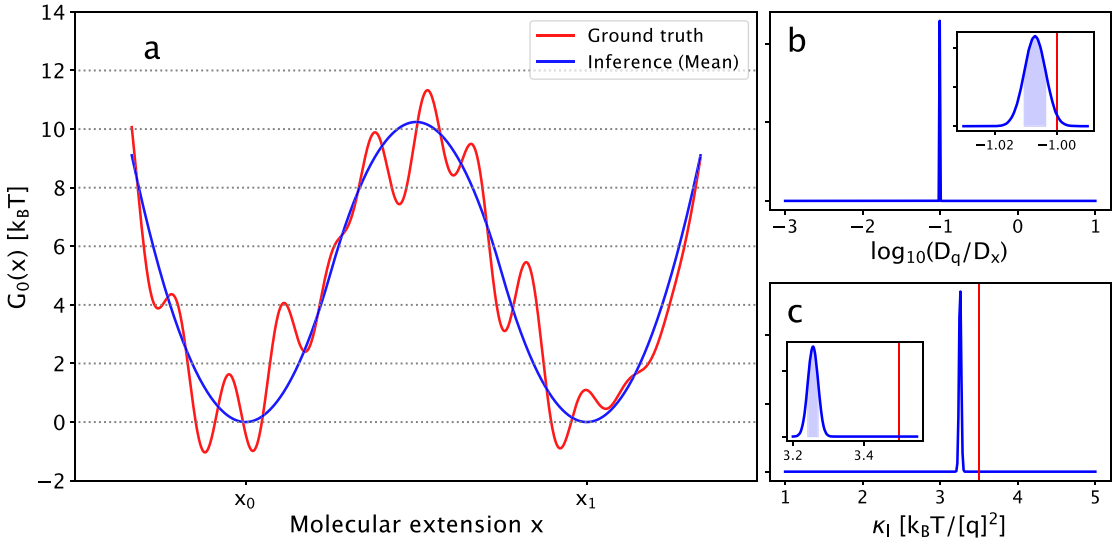

Figure 5. Model misspecification. (a) Best inference assuming an idealized symmetric double-well (blue line), evaluated on a synthetic observation produced by a rough molecular profile $G_0^r(x)$ (red line). (b) and (c) Marginal posterior distributions (blue lines) for the ratio of diffusion coefficients and linker stiffness, respectively. Red lines indicate the parameter true values, while blue shaded areas indicate a 68% confidence interval.

(figure 4(a)). In the two bottom states, the molecular extension takes different values x_0 and x_2, but both are projected on the same value q_0 (figure 4(b)). The time series of $\boldsymbol{q}_t$ produced by both models $\mathcal{M}_1$ and $\mathcal{M}_2$ describe a hopping process between only two states. Can an inference tell us whether a two-state model is enough to explain the observed data or if we need a three-state one?

The posterior trained on a nested model enables us to choose between models of different complexity. We trained a conditional posterior $p(\boldsymbol{\theta}|\boldsymbol{q}_t)$ using Brownian simulations on a free energy surface $G(q,x)$ defined by a linear combination of three Gaussian distributions (equation (4)). The model's parameters are $\{x_i, \sigma_i\}$ for $i = 1, 2$, which describe location and width, and the mixing coefficient ω, which determines the relative weight between the states. This model contains, in general, three states, like $\mathcal{M}_2$, but for $\omega = 1/2$ it reduces to the simpler two-state model $\mathcal{M}_1$. We compared the posterior of a synthetic observed time series produced from the three-state model, $p(\boldsymbol{\theta}|\boldsymbol{q}_t = \boldsymbol{q}_t^{(3)})$ (figure 4(c)), and one produced with a two-state model, $p(\boldsymbol{\theta}|\boldsymbol{q}_t = \boldsymbol{q}_t^{(2)})$ (figure 4(d)). In the first case, the marginal posteriors peak around the true values for all parameters of the three Gaussians. Also, the marginal of ω indicates that we need a three-state model to make an inference on the $\boldsymbol{q}_t^{(3)}$ observation. In the second case, instead, the posterior is sharply peaked around $\omega = 1/2$, indicating that a two-state model is sufficient.

3.5. Robustness to model misspecification

SBI performs an excellent inference if the synthetic observed data are produced by the same model that we encoded in the simulator. But what happens if this is not true? In reality, our model will only be an approximation of the process that produced the experimental observation.

A moderate model mismatch slightly degrades the estimate's accuracy. If a rough two-state curve $G_0^{\text{rough}}(x)$ generated $\boldsymbol{q}_t^{\text{obs}}$, performing inference with the posterior trained assuming a smooth symmetric double-well model will return the best fit to the true curve (figure 5(a)). The estimated D_q/D_x is very accurate since this quantity depends only on local fluctuations (figure 5(b)). The estimated κ_l is close to the true value (figure 5(c)). However, the posterior severely underestimate the uncertainties. It is too narrow and does not include the true value of κ_l. We also considered misspecification of the system's dynamics. Making an inference on synthetic observations produced with inertial dynamics (under-damped Langevin), while assuming a diffusive one, leads to good results in the limit of high friction (supplementary data figures 7(a)–(c)), but breaks down for low friction (supplementary data figures 7(d)–(f)).

SBI provides tools to diagnose model misspecification. The posterior predictive check reveals whether the experimental observation is 'unusual' compared to the simulated data (supplementary data figures 8 and 9). If so, the inference should not be trusted. Synthetic observations produced with the rough double-well or an inertial dynamics are atypical compared to simulations performed in the smooth symmetric double-well with Brownian dynamics (supplementary data figures 10 and 11).

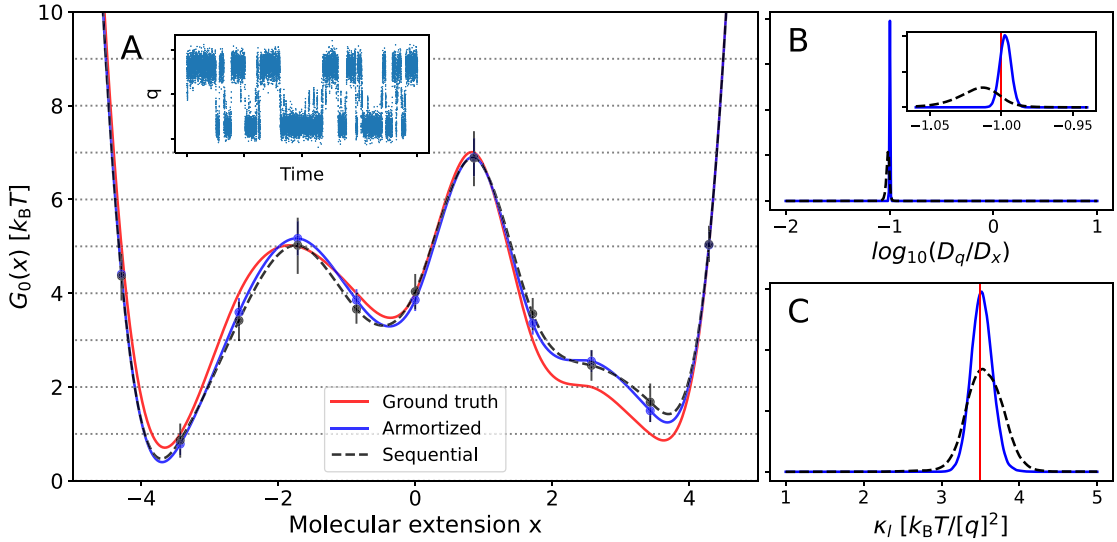

Figure 6. Amortized and sequential SBI of complex free energy profiles. (a) Best inference of a complex hidden molecular free energy profile $G_0(x)$. The true hidden profile is indicated in red, while the best estimate obtained from an amortized and sequential inference in blue and dashed black, respectively. Best estimate and error bars are the mean and the 68% marginal density of the posterior, respectively. The inset shows a representative trajectory $\mathbf{q}_t$. (b) Posterior marginal as a function of the ratio of diffusion coefficients, and (c) the linker stiffness. The inset in (b) shows a zoom-in.

3.6. Inference of complex free energy landscapes

Having proven the potential of the SBI approach, we aimed to making inferences of more realistic free energy profiles. Describing folding and other conformational rearrangements generally requires profiles presenting several long-lived intermediates and barriers of different heights. We considered a new class of models for $G_0(x)$ using polynomial splines. These allow for greater flexibility than the models considered so far, albeit at the cost of increasing the parameter space, requiring 11 parameters corresponding to the height of the nodes. The total dimension of $\boldsymbol{\theta}$ is now 13.

Despite the increased complexity, SBI successfully extracts complex hidden molecular free energy profiles. We trained a posterior on simulations performed with the flexible spline model of $G_0(x)$, and extracted profiles with an average error smaller than a k_BT (figure 6(a)). The estimate is also very good for the diffusion coefficients of the linker stiffness (figures 6(b) and (c)). The inference is amortized, and allows us to show without further simulations that we can extract $G_0(x)$ over a broad range of parameters (supplementary data figure 12). Notably, the spline model does not require defining the number and location of states and barriers. Both are automatically extracted from the observed time series. We obtained this result training on 1.5 million simulations. Whereas for our minimal model this required only a few days of simulations, the same might not be possible for more complex simulators.

Sequential SBI is a powerful alternative for computationally expensive simulators, or if we are interested in making inference on a single observation. In this approach, we iterate between running small batches of simulations, training a posterior, and using this posterior as a proposal distribution to initiate new simulations. This is a form of active learning: the algorithm autonomously learns where it should run simulations in the parameter space. In 20 iterations, the sequential approach provided an excellent inference (figure 6), and used only 30 000 simulations—2% of the simulations used for the amortized posterior (supplementary data figure 13).

4. Discussion

Despite their great success, the challenge of extracting quantitative models from partial observations hampers the full potential of smFS experiments. This challenge is a fundamental inverse problem. We lose information by projecting a high-dimensional system on a single quantity. Additionally, we measure this quantity via the mediation of an ever-present experimental apparatus that further distort the measurement.

In this work, we showcased how machine-learning-empowered SBI is a general, conceptually simple, and powerful technique for addressing these challenges for smFS experiments at constant force. Using synthetic data, we could extract compact interpretable models of increasing complexity that accurately described hidden physical processes over a broad range of parameters.

Model misspecification remains an outstanding problem for every practical inference. One of the clear advantages of SBI is that it provides a self-contained quality control. Posterior predictive checks can reveal that the model encoded in the simulator is unsuitable for reproducing the observed data and that the inference should not be trusted.

The main advantage of our approach is that the inference is amortized. The heavy computational part—running the simulator with many different choices of parameters—is necessary to train the posterior but has to be done only once for any given choice of model and priors. Once trained, any new inference requires only plugging the observation in the trained posterior and performing a forward-pass of the underlying neural network. Sequential inference offers a powerful alternative in situations where amortized inference is not feasible or desirable. If prior measurements are available, for example, for the linker stiffness and its uncertainty, one can encode them in the prior and reduce the parameter search space.

As increasingly more challenging smFS experiments are established, approaches like the one we explored here, together with non-parametric Bayesian techniques [60], will become more and more crucial. Identifying mechanistic heterogeneity requires monitoring subtle differences in the transition paths, i.e. how molecules re-organize between alternative meta-stable states. Transition paths are particularly affected by kinetic artifacts from measuring devices [22]. With more and more available computational power, it will soon be possible to use quasi-atomistic molecular dynamics with SBI. The simulator will then explicitly map trajectories containing molecular structures to measured extensions, keeping into account the specific position of the linkers, beads, and their physics. Moreover, our SBI formalism could be generalized to account for incomplete observations from multiple rebinding events [61], force-rupture [45], or other types of single-molecule experiments.

The main challenge of applying inference schemes to actual experimental data is to model the noise correctly. Whereas many approaches are often limited to idealized noise models—e.g. Gaussian distributed—real noise is generally complex. A simulation-based approach like ours can consider any noise model that can be encoded in a simulator. This includes not only known functions that would frustrate analytical treatments but also data-driven models of noise obtained, for instance, using machine learning approaches.

The combination of amortized and sequential inference enables the establishment of so-called foundation models. These would consist of amortized posteriors trained on complex, realistic simulations of biophysical experiments, which might require months to simulate and train. Once trained, they can be made available to the community and serve as the proposal distribution of an inexpensive sequential approach to fine-tune the posterior to specific experiments or observations.

Intractable likelihoods are very common for many important problems and hinder analytical investigations. On the other hand, generating synthetic data with high-fidelity simulators is very often straightforward. By integrating physics-based parametric models with machine learning density estimate, SBI enables accurate Bayesian inference for models with an intractable likelihood. In this way, it enables us to consider more complex and realistic models that would be otherwise ruled out due to their mathematical intractability, with great potential for applications in the quantitative biomolecular sciences.

Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: https://zenodo.org/record/7717522.

All the code and the data required to entirely reproduce the findings of this study are available on a public repository [62]. We also provide a self-contained Colab notebook tutorial: https://github.com/covinolab/sbi_for_smfe_tutorial/blob/main/sbi_smfe_example.ipynb.

Acknowledgments

We thank Drs Attila Szabo and Jürgen Köfinger for useful discussions and feedback. L D and R C acknowledge the support of the Frankfurt Institute of Advanced Studies, the LOEWE Center for Multiscale Modelling in Life Sciences of the state of Hesse, and the CRC 1507: Membrane-associated Protein Assemblies, Machineries, and Supercomplexes. R C acknowledges the support of the International Max Planck Research School on Cellular Biophysics. L D acknowledges the support of the GRADE center iQbio of the Goethe University. P C acknowledges the Simons Foundation. We thank the Center for Scientific Computing of the Goethe University and the Jülich Supercomputing Centre for computational resources and support. R C dedicates this paper in loving memory of Piero Angela, chief Italian science journalist, who, in more than 50 years of incessant activity, has narrated the beauty of science to many generations.

Appendix. Priors and hyperparameters

Symmetric double-well

Prior

We chose uniform distributions covering reasonable values based on previous publications [21, 23]:

$$\log_{10}(D_q/D_x) \sim \mathcal{U}(-3,1),\ \Delta G^{\ddagger} \sim \mathcal{U}(3\ k_\mathrm{B}T, 17\ k_\mathrm{B}T),\ \kappa_\mathrm{l} \sim \mathcal{U}(1\ k_\mathrm{B}T/[q]^{-2}, 5\ k_\mathrm{B}T/[q]^{-2}).$$

Hyperparameters

We used an MDN as the density estimator, with $K = 50$ and a feed-forward neural network with three layers. Each layer had 80 hidden nodes (sometimes called features in the SBI literature) and used the ReLU activation function. The output from the third layer yielded the parameters of the Gaussians. We kept 25% of the simulation data for the validation. The batch size was the default value of 50. We stopped training after the validation loss did not improve for over 30 epochs. The training took 267 s on one Intel Core i9-12900K processor.

Nested model

Prior

We chose the priors to ensure that the configuration of the three Gaussian distributions would reproduce the essential features of a smFS experiment: $\sigma_1 \sim \mathcal{U}(0.2, 0.5), \sigma_2 \sim \mathcal{U}(0.2, 0.5), \sigma_3 \sim \mathcal{U}(0.2, 0.5)$, $x_1 \sim \mathcal{U}(0.0, 0.5), x_2 \sim \mathcal{U}(1, 2), \omega \sim \mathcal{U}(1/4, 1/2)$.

Hyperparameters

We used an MDN as the density estimator with $K = 50$, with the same neural network topology as in appendix 'Nested model'. We only increased the number of hidden features per layer to 150. We kept 15% of the simulated data for validation. The batch size was set to 500. We stopped the training after the validation loss did not improve for over 20 epochs. The training took 446 s on one Intel Core i9-12900K processor.

Cubic spline

Prior

For the diffusion coefficients and linkers we used the similar priors as for the symmetric double-well $\log_{10}(D_q/D_x) \sim \mathcal{U}(-2,1),\ \kappa_\mathrm{l} \sim \mathcal{U}(1\ k_\mathrm{B}T/[q]^{-2}, 5\ k_\mathrm{B}T/[q]^{-2})$. The prior for the internal spline values were instead $G_{0,i} \sim \mathcal{U}(0\ k_\mathrm{B}T, 10\ k_\mathrm{B}T)\ \forall i \in \{2, \ldots, 12\}$.

Hyperparameters

We used a neural spline flow as the density estimator as implemented in the SBI-toolkit [47]. We used five transformations with 100 hidden features. We augmented the neural spline flow with an convolutional layer with six input channels and six output channels. The kernel had a size of 6x6 and a stride of 2. The convolutional layer used an ReLu activation function. We kept 15% from the simulation data for validation. The batch size was set to 1500. The training stopped after the validation loss did not improve for more than 20 epochs. The training took 27 778 s on one Xeon Skylake Gold 6148 Processor.

For the sequential approximation of the posterior, we iteratively ran new simulations with parameters from a prior or the previous posterior. The new simulations were added to the training set and the approximate posterior was further trained. We used the SNPE-C version of the sequential posterior estimation from the SBI-Toolbox [47]. In total, we performed 20 rounds of approximation each adding 1500 new simulations to the data set. We used the same hyperparameters for the training and the density estimator as for the amortized case. Just the batch size was set to 50. The cumulative training time for all 20 sequential runs was 24 619 s on one Intel Core i9-12900K processor.

ORCID iD

Roberto Covino https://orcid.org/0000-0003-0884-0402

References

[1] Plitzko J M, Schuler B and Selenko P 2017 Structural biology outside the box-inside the cell *Curr. Opin. Struct. Biol.* **46** 110–21
[2] Petrosyan R, Narayan A and Woodside M T 2021 Single-molecule force spectroscopy of protein folding *J. Mol. Biol.* **433** 167207
[3] Wang H, Chen G and Li H 2022 Templated folding of the RTX domain of the bacterial toxin adenylate cyclase revealed by single molecule force spectroscopy *Nat. Commun.* **13** 2784

[4] Kramm K et al 2020 DNA origami-based single-molecule force spectroscopy elucidates RNA Polymerase III pre-initiation complex stability Nat. Commun. 11 2828
[5] Pelz B, Žoldák G, Zeller F, Zacharias M and Rief M 2016 Subnanometre enzyme mechanics probed by single-molecule force spectroscopy Nat. Commun. 7 10848
[6] Bustamante C J, Chemla Y R, Liu S and Wang M D 2021 Optical tweezers in single-molecule biophysics Nat. Rev. Methods Primers 1 1–29
[7] Petrosyan R, Patra S, Rezajooei N, Garen C R and Woodside M T 2021 Unfolded and intermediate states of PrP play a key role in the mechanism of action of an antiprion chaperone Proc. Natl Acad. Sci. 118 e2010213118
[8] Chen H, Yuan G, Winardhi R S, Yao M, Popa I, Fernandez J M and Yan J 2015 Dynamics of equilibrium folding and unfolding transitions of titin immunoglobulin domain under constant forces J. Am. Chem. Soc. 137 3540–6
[9] Yao M et al 2014 Force-dependent conformational switch of α-catenin controls vinculin binding Nat. Commun. 5 4525
[10] Pang S M, Le S and Yan J 2018 Mechanical responses of the mechanosensitive unstructured domains in cardiac titin Biol. Cell 110 65–76
[11] Liu J, Feng W and Zhang W 2020 A single-molecule study reveals novel rod-like structures formed by a thrombin aptamer repeat sequence Nanoscale 12 4159–66
[12] Evans E and Ritchie K 1997 Dynamic strength of molecular adhesion bonds Biophys. J. 72 1541–55
[13] Hummer G and Szabo A 2003 Kinetics from nonequilibrium single-molecule pulling experiments Biophys. J. 85 5–15
[14] Dudko O, Hummer G and Szabo A 2006 Intrinsic rates and activation free energies from single-molecule pulling experiments Phys. Rev. Lett. 96 108101
[15] Cossio P, Hummer G and Szabo A 2016 Kinetic ductility and force-spike resistance of proteins from single-molecule force spectroscopy Biophys. J. 111 832–40
[16] Maitra A and Arya G 2011 Influence of pulling handles and device stiffness in single-molecule force spectroscopy Phys. Chem. Chem. Phys. 13 1836–42
[17] Friddle R W, Noy A and De Yoreo J J 2012 Interpreting the widespread nonlinear force spectra of intermolecular bonds Proc. Natl Acad. Sci. 109 13573–8
[18] Pierse C A and Dudko O K 2013 Kinetics and energetics of biomolecular folding and binding Biophys. J. 105 L19–22
[19] Makarov D E 2014 Communication: does force spectroscopy of biomolecules probe their intrinsic dynamic properties? J. Chem. Phys. 141 241103
[20] Hinczewski M, Gebhardt J C M, Rief M and Thirumalai D 2013 From mechanical folding trajectories to intrinsic energy landscapes of biopolymers Proc. Natl Acad. Sci. 110 4500–5
[21] Cossio P, Hummer G and Szabo A 2015 On artifacts in single-molecule force spectroscopy Proc. Natl Acad. Sci. 112 14248–53
[22] Cossio P, Hummer G and Szabo A 2018 Transition paths in single-molecule force spectroscopy J. Chem. Phys. 148 123309
[23] Covino R, Woodside M T, Hummer G, Szabo A and Cossio P 2019 Molecular free energy profiles from force spectroscopy experiments by inversion of observed committors J. Chem. Phys. 151 154115
[24] Satija R, Berezhkovskii A M and Makarov D E 2020 Broad distributions of transition-path times are fingerprints of multidimensionality of the underlying free energy landscapes Proc. Natl Acad. Sci. 117 27116–23
[25] Stigler J, Ziegler F, Gieseke A, Gebhardt J C M and Rief M 2011 The complex folding network of single calmodulin molecules Science 334 512–6
[26] Türkcan S, Alexandrou A and Masson J-B 2012 A Bayesian inference scheme to extract diffusivity and potential fields from confined single-molecule trajectories Biophys. J. 102 2288–98
[27] Bryan J S, Sgouralis I and Pressé S 2020 Inferring effective forces for Langevin dynamics using Gaussian processes J. Chem. Phys. 152 124106
[28] Bryan J S, Basak P, Bechhoefer J and Pressé S 2022 Inferring potential landscapes from noisy trajectories of particles within an optical feedback trap iScience 25 104731
[29] Yu H, Siewny M G W, Edwards D T, Sanders A W and Perkins T T 2017 Hidden dynamics in the unfolding of individual bacteriorhodopsin proteins Science 355 945–50
[30] Woodside M T, Behnke-Parks W M, Larizadeh K, Travers K, Herschlag D and Block S M 2006 Nanomechanical measurements of the sequence-dependent folding landscapes of single nucleic acid hairpins Proc. Natl Acad. Sci. 103 6190–5
[31] Cranmer K, Brehmer J and Louppe G 2020 The frontier of simulation-based inference Proc. Natl Acad. Sci. USA 117 30055–62
[32] Lueckmann J-M, Boelts J, Greenberg D, Goncalves P and Macke J 2021 Benchmarking simulation-based inference Int. Conf. on Artificial Intelligence and Statistics (PMLR) pp 343–51
[33] Papamakarios G and Murray I 2016 Fast ε-free inference of simulation models with Bayesian conditional density estimation Advances in Neural Information Processing Systems pp 1036–1044 (arXiv:1605.06376)
[34] Zeraati R, Engel T A and Levina A 2022 A flexible Bayesian framework for unbiased estimation of timescales Nat. Comput. Sci. 2 193–204
[35] Barrett R, Ansari M, Ghoshal G and White A D 2022 Simulation-based inference with approximately correct parameters via maximum entropy Mach. Learn.: Sci. Technol. 3 025006
[36] Brehmer J 2021 Simulation-based inference in particle physics Nat. Rev. Phys. 3 305–305
[37] Dax M, Green S R, Gair J, Macke J H, Buonanno A and Schölkopf B 2021 Real-time gravitational wave science with neural posterior estimation Phys. Rev. Lett. 127 241103
[38] Green S R and Gair J 2021 Complete parameter inference for gw150914 using deep learning Mach. Learn.: Sci. Technol. 2 03LT01
[39] Lemos P, Cranmer M, Abidi M, C, Eickenberg M, Massara E, Yallup D, Ho S 2022 Robust simulation-based inference in cosmology with Bayesian neural networks (arXiv:2207.08435)
[40] Khullar G, Nord B, Ciprijanovic A, Poh J and Xu F 2022 Digs: deep inference of galaxy spectra with neural posterior estimation Mach. Learn.: Sci. Technol. 3 04LT04
[41] Furia C S and Churchill R M 2022 Normalizing flows for likelihood-free inference with fusion simulations Plasma Phys. Control. Fusion 64 104003
[42] Bernstein D B, Sulheim S, Almaas E and Segrè D 2021 Addressing uncertainty in genome-scale metabolic model reconstruction and analysis Genome Biol. 22 64
[43] Lueckmann J M, Gonçalves P J, Bassetto G, öcal K, Nonnenmacher M and Macke J H 2017 Flexible statistical inference for mechanistic models of neural dynamics Advances in Neural Information Processing Systems pp 1290–300
[44] Gonçalves P J et al 2020 Training deep neural density estimators to identify mechanistic models of neural dynamics eLife 9 1–45
[45] Hummer G and Szabo A 2010 Free energy profiles from single-molecule pulling experiments Proc. Natl Acad. Sci. 107 21441–6

[46] Eastman P et al 2017 OpenMM 7: rapid development of high performance algorithms for molecular dynamics PLoS Comput. Biol. 13 e1005659
[47] Tejero-Cantero A, Boelts J, Deistler M, Lueckmann J-M, Durkan C, Gonçalves P, Greenberg D and Macke J 2020 SBI: a toolkit for simulation-based inference J. Open Source Softw. 5 2505
[48] Bishop C M 1994 Mixture density networks. Technical Report NCRG/94/004
[49] Greenberg D, Nonnenmacher M and Macke J 2019 Automatic posterior transformation for likelihood-free inference Int. Conf. on Machine Learning (PMLR) pp 2404–14
[50] Kobyzev I, Prince S J and Brubaker M A 2021 Normalizing flows: an introduction and review of current methods IEEE Trans. Pattern Anal. Mach. Intell. 43 3964–79
[51] Durkan C, Bekasov A, Murray I and Papamakarios G 2019 Neural spline flows Advances in Neural Information Processing Systems vol 32
[52] Harris C R et al 2020 Array programming with NumPy Nature 585 357–62
[53] Virtanen P et al 2020 SciPy 1.0: fundamental algorithms for scientific computing in Python Nat. Methods 17 261–72
[54] Lam S K, Pitrou A and Seibert S 2015 Numba: a LLVM-based python JIT compiler Proc. 2nd Workshop on the LLVM Compiler Infrastructure in HPC pp 1–6
[55] Behnel S, Bradshaw R, Citro C, Dalcin L, Seljebotn D S and Smith K 2010 Cython: the best of both worlds Comput. Sci. Eng. 13 31–39
[56] Paszke A et al 2019 Pytorch: an imperative style, high-performance deep learning library Advances in Neural Information Processing Systems vol 32
[57] Hunter J D 2007 Matplotlib: a 2D graphics environment Comput. Sci. Eng. 9 90–95
[58] Galassi M et al 2002 GNU Scientific Library (Godalming: Network Theory Limited) (available at: https://godalming.cylex-uk.co.uk/company/network-theory-limited-21826528.html)
[59] Hummer G 2005 Position-dependent diffusion coefficients and free energies from Bayesian analysis of equilibrium and replica molecular dynamics simulations New J. Phys. 7 34
[60] Bryan ISathyV J S, Sgouralis I and Pressé S 2022 Diffraction-limited molecular cluster quantification with Bayesian nonparametrics Nat. Comput. Sci. 2 102–11
[61] Bullerjahn J T and Hummer G 2022 Reversible bond kinetics from single-molecule force spectroscopy experiments close to equilibrium Phys. Rev. Res. 4 033097
[62] Dingeldein L, Cossio P and Covino R 2022 Simulation-based inference for single-molecule force-spectroscopy: code and data Zenodo (https://doi.org/10.5281/zenodo.7717522)