Mapping spatio-temporal dynamics of single biomolecules in living cells

François Laurent, Charlotte Floderer, Cyril Favard, Delphine Muriaux, Jean-Baptiste Masson, Christian L. Vestergaard

Mapping spatio-temporal dynamics of single biomolecules in living cells

François Laurent,[1,*] Charlotte Floderer,[2] Cyril Favard,[2] Delphine Muriaux,[2] Jean-Baptiste Masson**,[1,†] and Christian L. Vestergaard**[1,‡]

[1] *Decision and Bayesian Computation, Department of Computational Biology, Department of Neuroscience, CNRS USR 3756, CNRS UMR 3571, Institut Pasteur, 25 rue du Docteur Roux, Paris, 75015, France*

[2] *Infectious Disease Research Institute of Montpellier, CNRS UMR 9004, University of Montpellier, Montpellier, France*

(Dated: January 7, 2020)

We present a Bayesian framework for inferring spatio-temporal maps of diffusivity and potential fields from recorded trajectories of single molecules inside living cells. The framework naturally lets us regularise the high-dimensional inference problem using prior distributions in order to obtain robust results. To overcome the computational complexity of inferring thousands of map parameters from large single particle tracking datasets, we developed a stochastic optimisation method based on local mini-batches and parsimonious gradient calculation. We quantified the gain in convergence speed on numerical simulations, and we demonstrated for the first time temporal regularisation and aligned values of the inferred potential fields across multiple time segments. As a proof-of-concept, we mapped the dynamics of HIV-1 Gag proteins involved in the formation of virus-like particles (VLPs) on the plasma membrane of live T cells at high spatial and temporal resolutions. We focused on transient aggregation events lasting only on tenth of the time required for full VLP formation. The framework and optimisation methods are implemented in the TRamWAy open-source software platform for analysing single biomolecule dynamics.

INTRODUCTION

Photoactivation fluorescence microscopy makes it possible to track and study the dynamics of single biomolecules in living cells and at the whole cell scale [1, 2]. Such single molecule tracking approaches have revealed rich intracellular transport dynamics [3–21]. In many cellular systems the molecules' dynamics show spatial heterogeneities which have been linked to biological function. Recent examples include neuroreceptors binding in the postsynaptic membrane [9–11, 13, 16], signaling molecules involved in the initiation of lamellopodium formation in motile cells [18], protein transport in the endoplasmic reticulum [19], and recruitment of polyproteins in the plasma membrane to form HIV-1 virus-like particles [20]. A successful approach to analyse single molecule tracking data from such systems is to assume that each tracer, of the same molecular species, experiences the same spatially dependent force/drift and diffusivity fields. This hypothesis makes it possible to infer spatial maps of the tracers' dynamics from many, typically short, recorded trajectories [7, 22–26]. Recent experimental modalities can produce datasets consisting of up to tens of millions single molecule localisations. This opens up the possibility to map not only the spatial variation of biomolecular dynamics but also study how these maps evolve in time [20].

When inferring such maps, a fundamental trade-off between spatial (and temporal) resolution and statistical precision results from the limited amount data. On the one hand, the spatial and temporal resolutions should be as high as possible in order to capture the spatial and temporal phenomena of interest. On the other hand, the effective number of parameters should be kept as low as possible in order to be able to reliably infer them from the available data.

The spatio-temporal scale of mapping is typically set by binning the recorded localisations. To apply the mapping approach to processes of a priori unknown spatial and temporal scales and mitigate the impact of arbitrarily choosing such a scale, we propose to regularise the inferred maps. This can be done in a Bayesian setting using field regularising priors [13, 24]. It increases the precision and robustness of inferred parameter maps by imposing that they must vary smoothly, and makes it possible to infer high resolution maps even from sparse data.

We here extend the Bayesian mapping framework of [7, 13, 24] to include both spatial and temporal regularisation for robust and statistically principled inference of time-varying large-scale spatial maps of the diffusivity and potential fields governing single biomolecule motion. The price to pay for the increased precision offered by regularisation is a drastic increase in the computational complexity as it couples the parameters of the map. This transforms a collection of independent low-dimensional inferences of individual map parameters into a global high-dimensional inference problem. The classic approach for parameter inference such as direct gradient based algorithms or Markov chain Monte Carlo (MCMC) are ill suited for high dimensional inferences from large datasets since they scale poorly with the size of the dataset and the number of parameters to infer.

To tackle the computational complexity of inferring large spatio-temporal maps, we instead make use of stochastic optimization [27, 28], a general yet powerful method for addressing high-dimensional optimization

* francois.laurent@pasteur.fr
† jean-baptiste.masson@pasteur.fr; ** equal contribution
‡ christian.vestergaard@pasteur.fr; ** equal contribution

problems on large datasets. Stochastic optimisation can be traced back to 1951 [27], but has more recently seen widespread application to handle the massive datasets found in modern statistical and machine learning problems [29].

In this article, we describe a stochastic optimisation procedure tailored to the mapping problem, which uses spatio-temporally local mini-batches to increase the efficiency of updates and enable multiprocessing. We validate the approach on numerical data and demonstrate the advantage of stochastic optimisation over direct optimisation of the global posterior.

Although we focus on maximum a posteriori (MAP) inference of map parameters, we also provide a procedure for sampling the posterior distribution based on stochastic gradient Langevin dynamics [30] in the Supplementary Information.

We apply the framework to infer spatio-temporal maps of the intracellular dynamics of Gag proteins, which are involved in the formation the HIV-1 capsids at the plasma membrane of T cells. In this article, we show transient aggregation events that aborted before they led to fully-formed virus-like particles (VLP).

The Bayesian framework for spatio-temporally regularised mapping of single-molecule dynamics developed here is implemented in the open-source and freely available TRamWAy project (`github.com/DecBayComp/TRamWAy`).

METHODS

Model

We model the single molecule dynamics using the heterogeneous overdamped Langevin equation (OLE):

$$\frac{d\mathbf{r}(t)}{dt} = \mathbf{a}(\mathbf{r},t) + \sqrt{2D(\mathbf{r},t)}\,\boldsymbol{\xi}(t) \ . \tag{1}$$

Here $\mathbf{a}(\mathbf{r},t)$ and $D(\mathbf{r},t)$ are the (spatially and temporally varying) deterministic drift and diffusivity of the molecules in the point $\mathbf{r}$ at time t, respectively, and $\boldsymbol{\xi}(t)$ is a continuous-time white noise process, defined as the derivative of the Wiener process.

In the applications presented here, we will assume that the dynamics are effectively potential-driven and that the fluctuation-dissipation relation is respected locally. The first assumption imposes that the drift should be proportional to the gradient of a scalar potential field, $\mathbf{a}(\mathbf{r},t) = -\nabla V(\mathbf{r},t)/\gamma(\mathbf{r},t)$. The second assumption links the proportionality constant, the drag coefficient $\gamma(\mathbf{r},t)$, to the diffusivity through the Einstein relation $\gamma(\mathbf{r},t) = k_\mathrm{B}T/D(\mathbf{r},t)$.

These assumptions are typically employed to interpret the inferred drift fields as stemming from effective potential energy landscapes [7, 9, 13, 16, 20, 26, 31, 32]. Note however that since equilibrium conditions are not guaranteed in biological systems, inferred potential maps cannot generally be interpreted as potential energies in a strict sense as they may include terms of unknown amplitude due to spatial variations in the diffusivity [33] and terms induced by non-equilibrium energy fluxes. Notwithstanding, the inferred maps may capture biologically relevant information regardless of whether the equilibrium assumption is satisfied or not [9, 13, 16, 20]. As a statistical description, a model that only assumes potential-driven dynamics, and not equilibrium, is equivalent to the equilibrium model. Note also that assumptions about equilibrium can be relaxed significantly by using the approach developed in [33] for non-regularised maps.

Bayesian inference and regularisation

Bayesian inference deals with estimating the posterior probability $p(\boldsymbol{\theta}|\boldsymbol{x})$ of model parameters $\boldsymbol{\theta}$ conditioned on data $\boldsymbol{x}$. Here, the data is a set of M single molecule trajectories, $\boldsymbol{x} \equiv \mathbf{r} = \{\{\mathbf{r}_n^m\}_{n=1}^{N_m}\}_{m=1}^{M}$, where $\mathbf{r}_n^m$ denotes the nth recorded position in the mth trajectory, and N_m the trajectory's length.

The posterior is related to the model likelihood $p(\boldsymbol{x}|\boldsymbol{\theta})$ through Bayes' theorem,

$$p(\boldsymbol{\theta}|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})}{p(\boldsymbol{x})} \ , \tag{2}$$

where $\pi(\boldsymbol{\theta})$ is the prior probability for the parameters $\boldsymbol{\theta}$ in absence of data and $p(\boldsymbol{x}) = \int p(\boldsymbol{x},\boldsymbol{\theta})d\boldsymbol{\theta}$ is a normalising constant, termed the *evidence*. The evidence $p(\boldsymbol{x})$ can be ignored for parameter inference but is central to Bayesian model selection [34].

Equation (2) provides a probabilistic framework for regularisation by employing smoothing priors instead of conjugate or noninformative priors (see below).

Maximising $p(\boldsymbol{\theta}|\boldsymbol{x})$ w.r.t. $\boldsymbol{\theta}$ leads to maximum a posteriori (MAP) estimation. In this article, we will focus on MAP estimation, but we also provide a procedure for sampling the posterior around the MAP in the Supplementary Information.

Spatio-temporal meshing and approximate likelihood

The likelihood for our model can be obtained as the fundamental solution (the *Green's function*) of the Fokker-Planck equation corresponding to Eq. (1). However, in general such a solution is analytically intractable and numerically too time-consuming to be considered for inference.

Instead, we follow and extend the approach developed in [7, 24] and approximate the fields $V(\mathbf{r},t)$ and $D(\mathbf{r},t)$ as piecewise constant in both t and $\mathbf{r}$. In practice, we tessellate/segment space and time into spatio-temporal domains and consider instead the sets of average values of $V(\mathbf{r},t)$ and $D(\mathbf{r},t)$ in each domain, $\mathbf{V} = \{V_{\alpha,\tau}\}$ and $\mathbf{D} = \{D_{\alpha,\tau}\}$, where $\tau = 1, 2, \ldots, T$ indexes the time segments, and $\alpha = 1, 2, \ldots, \Omega$ identifies the space domains.

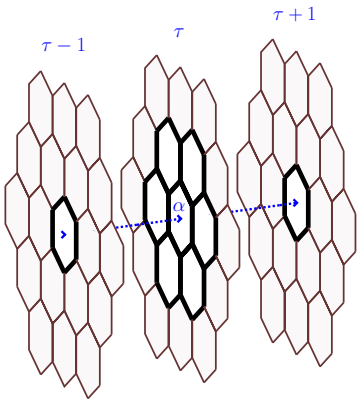

FIG. 1. Illustration of the spatio-temporal tessellation. We used regular meshes consisting of equal-sized hexagons in space and equal-length time segments. The highlighted domains (mesh domains with thick black edges) show the spatio-temporal neighbourhood $\mathcal{B}_{\alpha,\tau}$ of the central domain (α,τ), with (α,τ) itself included. We refer to the spatial neighbourhood of a spatial domain α (disregarding time) as $\mathcal{R}_\alpha$, and to the temporal neighbourhood of a temporal segment τ as $\mathcal{T}_\tau = \{\tau-1,\tau,\tau+1\}$. Temporal neighbours share the same spatial position and spatial neighbours are considered at fixed time.

Figure 1 illustrates the spatio-temporal tessellation we employ here, which consists in a regular hexagonal tessellation combined with a sliding time window. It also illustrates the important concepts of the spatial neighbourhood $\mathcal{R}_\alpha$ of a space domain α, containing α itself and the space domains in contact with α, the temporal neighbourhood $\mathcal{T}_\tau = \{\tau-1,\tau,\tau+1\}$ of a time segment τ, and the spatio-temporal neighbourhood $\mathcal{B}_{\alpha,\tau} = \{(\alpha',\tau)\}_{\alpha'\in\mathcal{R}_\alpha} \cup \{(\alpha,\tau')\}_{\tau'\in\mathcal{T}_\tau}$ of spatio-temporal domain (α,τ). Note that the domains can overlap in space and time (see *Data sampling* below).

Many other geometries are possible for the domains, which need not be regular and can be adapted to the density of localisations. We here consider only regular hexagonal tessellations, which have several attractive properties for spatio-temporal mapping. Using a mesh that is constant over time makes it straightforward to implement temporal regularisation as one does not have to deal with partially overlapping domains (Fig. 1). Similarly, the hexagonal spatial meshing avoids having to arbitrarily choose how to make regularisation of a domain depend on the distances to the centers of neighbouring domains and on their shared perimeters since these are all equal. In the case of a square meshing this problem is especially egregious as neighbours along the diagonal meet in a single point only. Instead of using an adaptive tessellation, we rely on regularisation to handle the domains with little data. Indeed, regularising makes the inference more robust to noisy data and allows to consider higher resolutions for the entire maps without having to sacrifice either statistical precision in low density regions or spatio-temporal resolution in high density regions.

Under the piecewise constant approximation, the fundamental solution to the Fokker-Planck equation is a Gaussian distribution, and the probability for a tracer located in the spatio-temporal domain (α,τ) to perform a displacement of $\Delta\mathbf{r}_i$ is then [7, 24]:

$$p(\Delta\mathbf{r}_i|\boldsymbol{\theta}_{\mathcal{R}_\alpha,\tau}) = \frac{1}{4\pi D_{\alpha,\tau}\Delta t_i}\exp\left(-\frac{\left||\Delta\mathbf{r}_i + \frac{D_{\alpha,\tau}\Delta t_i}{k_\mathrm{B}T}\nabla V_{\alpha,\tau}\right|^2}{4D_{\alpha,\tau}\Delta t_i}\right) \tag{3}$$

Here, the potential gradient $\nabla V_{\alpha,\tau}$ is evaluated numerically, in practice using a finite differences scheme. It thus depends not only on the local parameters $\boldsymbol{\theta}_{\alpha,\tau} = (D_{\alpha,\tau}, V_{\alpha,\tau})$ in the domain, but on the values of V in the whole spatial neighbourhood of (α,τ): $\boldsymbol{\theta}_{\mathcal{R}_\alpha,\tau} = \{\boldsymbol{\theta}_{\alpha',\tau}\}_{\alpha'\in\mathcal{R}_\alpha}$.

The total likelihood of all recorded trajectories can be written in a factorised form as a product of local likelihoods [Eq. (3)]. If we let $\Delta\mathbf{r}_{\alpha,\tau} = \{\Delta\mathbf{r}_{\alpha,\tau,i}\}_i$ denote the set of all displacements recorded inside the domain $\{\alpha,\tau\}$, we can write the total likelihood for all the recorded data as:

$$p(\Delta\mathbf{r}|\boldsymbol{\theta}) = \prod_{\alpha,\tau} p(\Delta\mathbf{r}_{\alpha,\tau}|\boldsymbol{\theta}_{\mathcal{R}_\alpha,\tau}) \ , \tag{4}$$

where $\boldsymbol{\theta} = \{\boldsymbol{\theta}_{\alpha,\tau}\}$ is the set of all local map parameters and $\Delta\mathbf{r} = \{\Delta\mathbf{r}_{\alpha,\tau}\}$ is the set of all recorded displacements, with $\alpha = 1,2,\ldots,\Omega$ and $\tau = 1,2,\ldots,T$. Note that for the factorised likelihood [Eq. (4)], $\Delta\mathbf{r}$ is equivalent to the set of trajectories $\mathbf{r} = \{\{\mathbf{r}_n^m\}_{n=1}^{N_m}\}_{m=1}^{M}$.

Spatio-temporal regularisation

To regularise the inferred maps, we impose a field regularising (Sobolev) prior $\pi(D,V)$ that penalizes variations in D and V between neighbouring domains in space and time [35, 36]. This prevents physically-unrealistic patterns of field parameters such as spurious minima or maxima. The regularisation prior factorizes as $\pi(D,V) = \pi(D)\pi(V)$, with

$$\pi(D) \propto e^{-\int\left(\mu_\mathbf{r}||\nabla D||^2 + \mu_t\dot{D}^2\right)dt\,d\mathbf{r}} \ , \tag{5}$$

$$\pi(V) \propto e^{-\int\left(\lambda_\mathbf{r}||\nabla V||^2 + \lambda_t\dot{V}^2\right)dt\,d\mathbf{r}} \ . \tag{6}$$

In practice, since the maps are modeled as being piecewise constant, the integrals above reduce to sums:

$$\pi(\mathbf{D}) \propto e^{-\sum_{\alpha,\tau}\delta_\tau\mathcal{A}_\alpha\left(\mu_\mathbf{r}||\nabla D_{\alpha,\tau}||^2 + \mu_t\dot{D}^2_{\alpha,\tau}\right)} \ , \tag{7}$$

$$\pi(\mathbf{V}) \propto e^{-\sum_{\alpha,\tau}\delta_\tau\mathcal{A}_\alpha\left(\mu_\mathbf{r}||\nabla V_{\alpha,\tau}||^2 + \mu_t\dot{V}^2_{\alpha,\tau}\right)} \ . \tag{8}$$

where δ_τ is the temporal duration and $\mathcal{A}_\alpha$ is the area (in 2D) or volume (in 3D) of domain (α, τ). For regular maps, $\delta_\tau = \delta$ and $\mathcal{A}_\alpha = \mathcal{A}$ are constant, but we allow them to vary in general for applications with non-regular meshing.

In $\pi(\mathbf{D})$ and $\pi(\mathbf{V})$, the differential operators ∇ and $\dot{}$ involve calculating finite differences [see *Stochastic optimisation*, Eqs. (12) and (13), below].

Including the regularising priors, the posterior reads:

$$p(\boldsymbol{\theta}|\Delta\mathbf{r}) \propto p(\Delta\mathbf{r}|\boldsymbol{\theta})\pi(\mathbf{D})\pi(\mathbf{V}) \ . \tag{9}$$

As seen from Eq. (4), (7), and (8) neither the likelihood nor the priors factorise into independent local likelihoods. This couples the inferences of map parameters of the different domains, both spatially and temporally, which in turn renders the optimisation of the posterior computationally hard. Indeed, we cannot simply optimise the posterior $p(\boldsymbol{\theta}|\Delta\mathbf{r})$ w.r.t. the parameters in each map domain independently, but instead need to optimise w.r.t. all parameters simultaneously.

Stochastic optimisation

Here, we propose to rely on stochastic optimisation [27, 28] to solve our inference problem. In stochastic optimisation, the cost function—here the negative log posterior $f(\boldsymbol{\theta}) = -\log p(\boldsymbol{\theta}|\Delta\mathbf{r})$—is optimised by following a set of noisy estimates of its gradient based on minibatches of data.

In its general form, stochastic optimisation randomly selects a subset of $n \ll N$ points, out of the total dataset of N points, to calculate the gradient of the cost function at each optimisation step. This can dramatically decrease the computational burden of high dimensional inferences from massive datasets.

We here describe our algorithm briefly and refer to the Supplementary Information for technical details and implementation procedures. We devised a stochastic optimisation scheme to take advantage of the fact that the parameters of each domain only couple to spatio-temporally neighbouring domains. This lets us decompose the cost function f as a sum of local terms,

$$f(\boldsymbol{\theta}) = \sum_{\alpha,\tau} f_{\alpha,\tau}(\boldsymbol{\theta}_{\mathcal{B}_{\alpha,\tau}}) \ , \tag{10}$$

where each local term depends on the neighbourhood $\mathcal{B}_{\alpha,\tau} = \{(\alpha', \tau)\}_{\alpha' \in \mathcal{R}_\alpha} \cup \{(\alpha, \tau')\}_{\tau' \in \mathcal{T}_\tau}$, comprising (α, τ) as well as spatio-temporally neighbouring domains (see Figure 1). Each local cost term is given by:

$$\begin{aligned} f_{\alpha,\tau}(\boldsymbol{\theta}_{\mathcal{B}_{\alpha,\tau}}) = &-\log p(\Delta\mathbf{r}_{\alpha,\tau}|\boldsymbol{\theta}_{\mathcal{R}_\alpha,\tau}) \\ &+\mu_\mathbf{r}\, q_\alpha(\mathbf{D}_{\mathcal{R}_\alpha,\tau}) + \mu_t\, q_\tau(\mathbf{D}_{\alpha,\mathcal{T}_\tau}) \\ &+\lambda_\mathbf{r}\, q_\alpha(\mathbf{V}_{\mathcal{R}_\alpha,\tau}) + \lambda_t\, q_\tau(\mathbf{V}_{\alpha,\mathcal{T}_\tau}) \ . \end{aligned} \tag{11}$$

Here, the local regularisation terms q_α and q_τ are given by:

$$q_\alpha(\boldsymbol{\phi}_{\mathcal{R}_\alpha,\tau}) = \frac{1}{2}\frac{\delta_\tau \mathcal{A}_\alpha}{|\mathcal{R}_\alpha| - 1} \sum_{\alpha' \in \mathcal{R}_\alpha \setminus \{\alpha\}} \frac{(\phi_{\alpha,\tau} - \phi_{\alpha',\tau})^2}{||\mathbf{r}_\alpha - \mathbf{r}'_\alpha||^2} \ , \tag{12}$$

where ϕ is a placeholder for either D or V and $\mathbf{r}_\alpha$ and $\mathbf{r}'_\alpha$ are the centers of the space domains α and α', respectively, and

$$q_\tau(\boldsymbol{\phi}_{\alpha,\mathcal{T}_\tau}) = \frac{1}{2}\frac{\delta_\tau \mathcal{A}_\alpha}{|\mathcal{T}_\tau| - 1} \sum_{\tau' \in \mathcal{T}_\tau \setminus \{\tau\}} \frac{(\phi_{\alpha,\tau} - \phi_{\alpha,\tau'})^2}{\Delta T^2} \ . \tag{13}$$

The factor $1/2$ in Eqs. (12) and (13) accounts for the fact that each individual penalty term is counted twice [e.g. $(\phi_{\alpha,\tau} - \phi_{\alpha,\tau'})^2/\Delta T^2$ is counted in both $f_{\alpha,\tau}$ and $f_{\alpha,\tau'}$].

The local cost functions $f_{\alpha,\tau}$ are not independent, but the decomposition [Eq. (10)] enables partial evaluation of f for gradient calculation and line search in parameter subspaces. We devised a stochastic algorithm, based on the Broyden-Fletcher-Goldfarb-Shanno (BFGS) optimization method [37], that only requires such partial evaluations of f.

This algorithm updates a subset of the parameters based on a local minibatch of data at each iteration. Rather than randomly sampling data minibatches, the algorithm selects data that are directly related to a set of local physical parameters. This selection is thus only local and directly maps onto the mesh domains associated with the parameters. The locality of the minibatches also makes the algorithm amenable to parallelisation and multiprocessing (see Supplementary Information for details on multi-process implementation).

The optimisation algorithm repeats the following steps until convergence: For each iteration k,

1. select a local domain (α, τ) to update;
2. sample a minibatch $\Delta\mathbf{r}_{\mathcal{B}_{\alpha,\tau}} = \cup_{(\alpha',\tau') \in \mathcal{B}_{\alpha,\tau}} \Delta\mathbf{r}_{\alpha',\tau'}$ of displacements in (α, τ)'s neighbourhood;
3. update the local parameters following $\boldsymbol{\theta}^{(k)}_{\alpha,\tau} = \boldsymbol{\theta}^{(k-1)}_{\alpha,\tau} + \Delta\boldsymbol{\theta}(\Delta\mathbf{r}_{\mathcal{B}_{\alpha,\tau}}, \boldsymbol{\theta}^{(k-1)}_{\mathcal{S}_{\alpha,\tau}})$, where $\Delta\boldsymbol{\theta}$ is an implicit function of the local data $\Delta\mathbf{r}_{\mathcal{B}_{\alpha,\tau}}$ and the parameters in the extended neighbourhood $\mathcal{S}_{\alpha,\tau} = \cup_{\alpha',\tau' \in \mathcal{B}_{\alpha,\tau}} \mathcal{B}_{\alpha',\tau'}$ (see *Gradient descent in a subspace* below).

The order in which the domains are updated in (1) should be randomised in order to avoid artefacts in the inferred maps. We do so by looping over all domains in an order that is randomised in each epoch. This ensures that all domains are updated while avoiding an arbitrary ordering of the updates. We here select the whole set $\Delta\mathbf{r}_{\mathcal{B}_{\alpha,\tau}}$ as minibatch in (2), but it is possible to subsample it instead for large datasets. The update $\Delta\boldsymbol{\theta}$ of $\boldsymbol{\theta}_{\alpha,\tau}$ in (3) only depends on the local cost terms $f_{\alpha,\tau}$ in the neighbourhood $\mathcal{B}_{\alpha,\tau}$. Since each local cost function $f_{\alpha',\tau'}$ depends on parameters of the domains in its neighbourhood

$\mathcal{B}_{\alpha',\tau'}$, the update $\Delta\boldsymbol{\theta}$ ultimately depends on the parameters of the extended neighbourhood $\mathcal{S}_{\alpha,\tau}$ that includes all neighbours of the domains in $\mathcal{B}_{\alpha,\tau}$, but not any domains further away.

Gradient descent in a subspace

A detailed description of how $\Delta\boldsymbol{\theta}$ is calculated is given in the Supplementary Information. Briefly, for each iteration k we obtain $\Delta\boldsymbol{\theta}$ from the implicit relation $\Delta\boldsymbol{\theta}(\Delta\mathbf{r}_{\mathcal{B}_{\alpha,\tau}}, \boldsymbol{\theta}^{(k-1)}_{\mathcal{S}_{\alpha,\tau}}) = s^{(k)}\mathbf{p}^{(k)}$. Here $s^{(k)}$ is the step size and $\mathbf{p}^{(k)} = -\mathcal{H}^{(k-1)}\nabla_{\alpha,\tau}f(\boldsymbol{\theta}^{(k-1)})$ is the descent direction, with $\nabla_{\alpha,\tau}f$ the gradient of f w.r.t the local parameters $\boldsymbol{\theta}_{\alpha,\tau}$ and $\mathcal{H}$ an approximation to the inverse Hessian matrix. We relied on approximate line search [37] to estimate $s^{(k)}$. Interestingly, the same partial evaluation as for gradient calculation stands, which makes line search operate fast, in a 2-dimensional parameter subspace.

Data sampling

This study showcases three datasets of simulated trajectories and one experimental dataset consisting of trajectories of wild-type Gag proteins (WT Gag) recorded in the plasma membrane using sptPALM. In this section, we briefly describe how these datasets were binned and how molecule locations were assigned to the resulting spatio-temporal domains.

For each dataset, a regular grid of non-overlapping hexagons was adjusted to the bounding box of all the molecule locations. The minimal diameter (the diameter of the inscribed circle) of the hexagons was set to 100 nm for the first two simulations, 68.5 nm for the third simulation, and 20 nm for the WT Gag dataset. This diameter defines the spatial resolution of the analysis.

Similarly, the total recording (or simulation) time was divided into regular intervals. The first simulated dataset did not feature time-evolving dynamics, so only a single time-slice was used. The second and third simulated datasets feature 18 time segments of 1 s and 5 s respectively, while the experimental WT Gag dataset features 12 1-min time segments. The experimental modalities of the WT Gag dataset and statistical analysis is described in more detail in the following section (*Experimental data*).

To further ensure the reliability of the inference procedure for sparse data, we incorporated different oversampling modalities that consider overlapping domains and may consequently assign localisations to multiple domains. Namely, (i) considering any initial tessellation, the spatial domains were enlarged as overlapping circular regions of constant radius; (ii) the temporal extent of the domains was defined using a sliding time window that introduced some constant overlap between successive time segments; and (iii) the domains that did not include a given minimum number of localisations were individually and symmetrically enlarged in time until this minimum number was reached. While the oversampling has a smoothing effect, the main purpose is to ensure the reliability of the optimisation and inference procedures. For visualisation, the non-overlapping shapes from the original tessellation were shown. In all three oversampling modalities, the spatial and temporal integration in q_α and q_τ (respectively) considered the non-overlapping extent of the domains, e.g. for q_α in modality (i) the original non-overlapping shapes of the spatial domains.

No oversampling was performed for the first simulated dataset. For the second simulated dataset, following modality (iii) spatio-temporal domains with less than 20 molecule locations were individually enlarged in time until 20 locations could be assigned to the domain. No oversampling was performed for the third simulated dataset in order to isolate the effect of the smoothing priors. For the experimental dataset, regular spatial (i) and temporal (ii) windowing were applied. Since the dataset is very sparse and our focus is to detect short-lived events, we used relatively large spatial overlap (130 nm for 20 nm spatial resolution) and little temporal overlap (2 min sliding windows for a 1 min temporal resolution), and we did not apply adaptive temporal windowing.

Note that while the regularizing priors in principle suffice to handle domains with few datapoints, it is in practice preferable to exclude domains not containing a minimal number of localisations and/or artificially increasing the number of points by oversampling in order to ensure that inferences are stable and the posterior is integrable. This is because each spatio-temporal domain contributes equally to the regularisation term in the posterior probability no matter how many points it contains. Thus, if many domains contain few or no points (as is the case in the experimental dataset), these will contribute disproportionally to the posterior, and in a manner that is overly sensitive to initialisation, possibly rendering the inference procedure unstable.

Experimental data

We analyse a previously unpublished set of trajectories of wild-type (WT) mEOS2-tagged HIV-1 Gag proteins recorded at 50 Hz in the plasma membrane of a single $CD4^+$ T cell for a previous study [20]. The experimental procedure, from protein production (using the pGag-(i)mEOS2 WT plasmid) to imaging and tracking, is described in [20] (condition: *WT Gag*).

To map the single Gag dynamics, here the spatial 2D domains were regularly laid across a 3 μm $\times$ 3 μm area, with centers arranged along a hexagonal grid, spaced by 20 nm and sized as 130 nm-radius discs. This large radius—as large as observed virus-like particles (VLPs) [20]—was necessary to collect enough molecule locations so that the diffusivity and potential energy could be estimated over dense and large spatio-temporal patches that

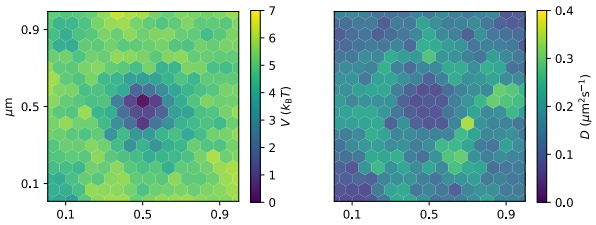

FIG. 2. Inferred potential energy and diffusivity landscapes in the static, low-dimensional example (558 parameters in total).

cover not only patterns of interest, but also the close environment of these patterns. To emphasize the benefit of temporal regularisation, we used a 2 min sliding temporal windows and limited the overlap between successive time segments to 50%, resulting in one time segment every minute. Spatio-temporal domains containing no localisations were excluded from the analysis.

The effects of localization uncertainty and motion blur (also known as *static* and *dynamic* errors, respectively [38]) were accounted for by replacing $D(\mathbf{r},t)$ by $D^{\text{eff}}(\mathbf{r},t) = (1-2R)D(\mathbf{r},t)+\sigma^2/\Delta t$ in Eq. (3) [39]. Here σ^2 is the variance of the localization uncertainty, assumed zero mean and time-uncorrelated, and R is a coefficient quantifying the motion blur [40, 41], where $R = 1/6$ corresponds to continuously open camera shutter and $R = 0$ may be obtained by leaving the camera shutter open for only a vanishing part of the time-lapse between measurements. For the present measurements, the localisation uncertainty was estimated to be $\sigma = 30$ nm and the motion blur coefficient was $R = 1/6$.

RESULTS

Stochastic optimisation accelerates inference of spatio-temporal maps

We first demonstrate the stochastic optimisation approach on simulated single particle tracking data, and we compare it to the deterministic global optimisation. We namely consider a low-dimensional example of inference of a static, spatially varying potential field (558 parameters to infer and 2 682 particle localisations) and a higher-dimensional intermediary example of a spatio-temporally varying field (4 648 parameters, 22 589 particle localisations). Both datasets exhibit constant diffusivity and a potential energy sink in space, which is constant in time for the first dataset (Fig. 2) and varies over time for the second (Fig. 3). The first dataset was introduced to quantify the convergence of several optimisation methods and does not involve time windowing. The second dataset introduces time regularisation and emphasizes the additional benefit of stochastic optimization in this context. It is divided into 18 non-overlapping time segments as illustrated in Fig. 3.

Convergence is measured by the correlation of the partial estimates $\boldsymbol{\phi}^{(k)}$ (where $\boldsymbol{\phi} \in \{\mathbf{D}, \mathbf{V}\}$) of each method with the final estimate $\boldsymbol{\phi}^*$ obtained using the global deterministic optimisation of the full cost function. This way we compare not only the speed of convergence of the methods, but also that they converge towards the same optimum.

Fitting $\mathbf{V}$ took more local cost function calls than fitting $\mathbf{D}$ did (Figs. 4 and 5). This is generally the case even when $\mathbf{D}$ exhibits equivalent patterns. We will therefore focus on convergence for $\mathbf{V}$.

Spatial map

The lower computational complexity of all the improved methods compared to the naive gradient calculation is illustrated in Figure 4. Direct gradient descent converges much slower than its quasi-Newton counterpart, although this variant can also run in multi-processing mode and consequently be faster (in absolute processing time) than the non-stochastic variant. This validates the approximation made in estimating the inverse Hessian matrix as a diagonal 2x2-block matrix (see Supplementary Information).

All optimisation methods converge relatively fast for the inference of the low-dimensional spatial map. The stochastic variant initially converges fastest until some precision is reached (for the potential energy $\mathbf{V}$ at a correlation of 0.9997 with the optimal parameter values). Above this level, the non-stochastic variant converges at a constant pace whereas the stochastic variant slows down, perhaps due to many local updates being redundant when most parameters are close to their optimal values. This is mitigated by the parallelisability of stochastic optimisation. Since multiprocessing does not increase the number of local cost function calls (Figure 4), this allows to divide the elapsed time for optimisation by the number of workers (about 20 on modern computers) using multi processing as compared to single processing.

Spatio-temporal map

The second dataset involves a transient potential energy sink as illustrated in Figure 3. The spatial-temporal map we aim to infer contains 18 individual time slices, so the number of parameters is 18 times higher than for the static map in the low-dimensional example above.

We compared stochastic and non-stochastic quasi-Newton optimisation (Figure 5). For the regularised spatio-temporal inference, the stochastic algorithm converges much faster than the non-stochastic one—about an order of magnitude for the present example—and the parallelisability of the stochastic algorithm compounds this gap by another order of magnitude.

Note that the number of parameters inferred in this example (4 648) is much smaller than the number of

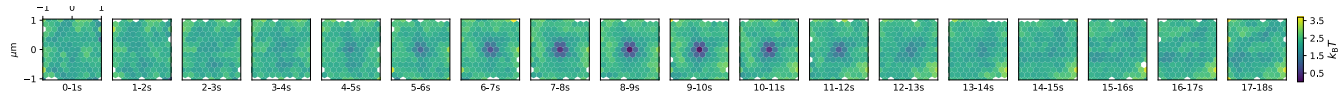

FIG. 3. Inferred potential energy landscapes at successive time segments in the intermediary spatio-temporally varying example (4 648 parameters in total).

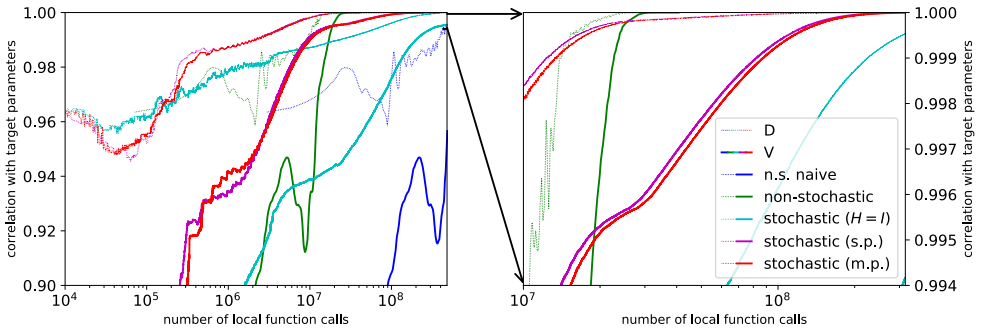

FIG. 4. Convergence as function of the number of calls to local cost functions $f_{\alpha,\tau}$ for the low-dimensional spatial inference (558 parameters to infer). *n.s. naive* stands for non-stochastic with naive gradient calculation. $H = I$ means that the inverse Hessian matrix is not approximated using the BFGS update rule, but set to the identity matrix instead; this is the direct gradient descent, in contrast to the quasi-Newton approach. *s.p.* and *m.p.* stand for single-process and multi-process, respectively.

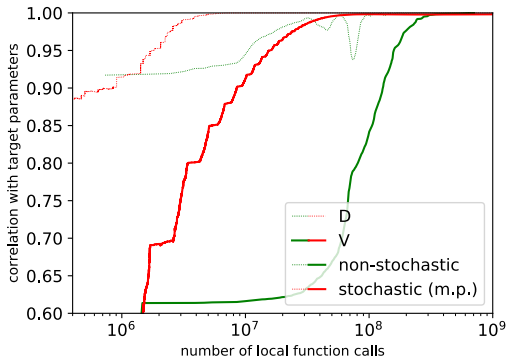

FIG. 5. Convergence as a function of the number of calls to local cost functions $f_{\alpha,\tau}$ for the spatio-temporal inference (4 648 parameters to infer). *m.p.* stand for multi-process.

parameters one may meet in empirical examples. The map inferred in the experimental example presented below for example contains 127 578 parameters. For such high-dimensional inferences, the non-stochastic algorithm does not converge.

Computational complexity

We let P denote the number of parameters. P is proportional to the number of different domains (α, τ) in the mesh, so the full gradient calculation of non-stochastic algorithms has a computational complexity of $\mathcal{O}(P^2)$ per iteration. In comparison the partial gradient evaluation of our stochastic algorithm has a complexity of $\mathcal{O}(P\,B)$, where $B = |\mathcal{B}_{\alpha,\tau}|$ is the size of a local neighbourhood. In terms of the size of the parameter space, P, this reduces to $\mathcal{O}(P)$ since B is always bounded by a low value in practice: here $B = 9$ and typical values in 2D are $B < 10$. For reference, $P = 127\,578$ in the experimental example below.

Strategies for choosing regularisation hyperparameters

Figure 6 illustrates how the value of the hyperparameter λ_τ controlling the strength of temporal regularisation of the potential field $\mathbf{V}$ influences the inferred field. The dataset involved is slightly larger then the previous one. It features 9 992 parameters and 126 022 localisations, and it displays three potential wells (Fig. 6, top rows).

The optimal choice of regularisation hyperparameters $\mu_\mathbf{r}$, μ_t, $\lambda_\mathbf{r}$ and λ_t depends on the overall study, and there is no general be-all and end-ell strategy for choosing them. The application of mapping to biomolecule dynamics in living cells may involve the characterisation of specific features of the inferred maps, such as the size, depth and duration of potential wells. Some studies then focus on differences between various biological conditions rather than on absolute measurements. More generally, inferring maps is often a preliminary analysis step, and the further analysis steps may all bring their specific requirements, which can guide the selection of regularising hyperparameters.

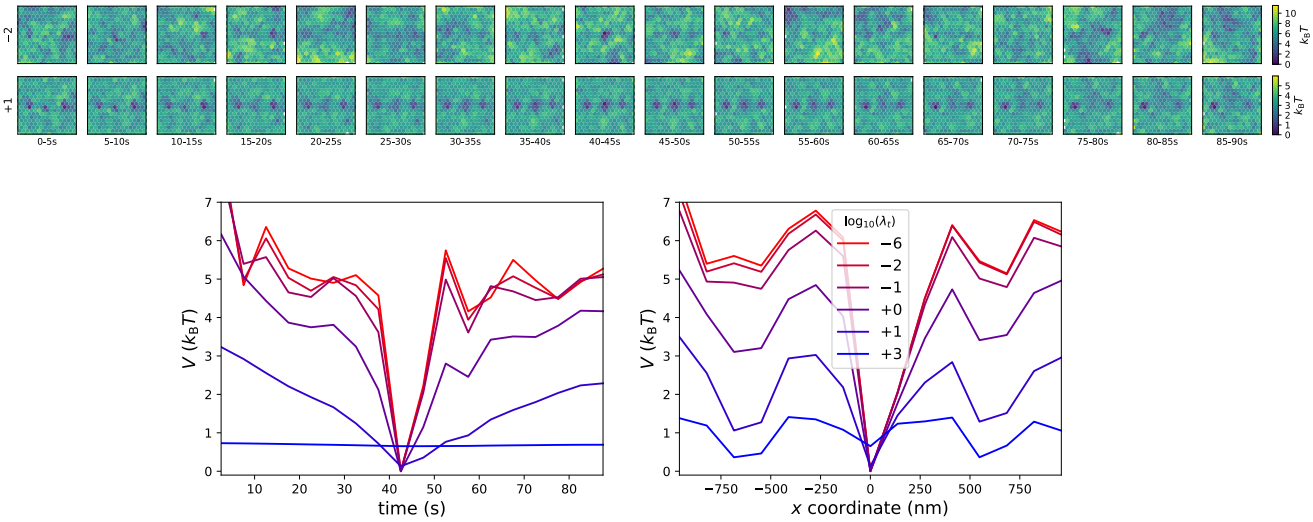

FIG. 6. Effects of temporal regularisation on inferred maps for an example consisting in two static and one temporally varying potential wells. The centers of the wells are all placed on the central horizontal line of the maps, with the temporally varying well in the center and the static wells to each side (see top rows). (Top rows) Two examples of inferred spatio-temporal maps for different strengths of the temporal regularisation ($\log_{10}\lambda_t = -2$ and $\log_{10}\lambda_t = +1$, respectively). (Bottom row) Temporal (left) and spatial (right) profiles of the inferred potential $\mathbf{V}$ for a selection of different values of the hyperparameter λ_t controlling temporal regularisation (logarithmic scale). (Bottom, left) The temporal profile shows the value in the central space domain (see top rows) as function of time. (Bottom, right) The spatial profile shows the values in the spatial domains lying on the horizontal line going through the central domain at the time when the depth of the potential well is maximal.

A common criterion is to avoid false detections of differences between the feature values corresponding to different biological conditions. False positives may notably be caused by unphysical outliers in the distributions of values of map features as statistical tests used for comparison often are sensitive to outliers. A simple approach to correct this consists in tuning the hyperparameters to ensure that the resulting distributions do not exhibit long tails, as measured for example using a Kolomogorov-Smirnov test to compare the empirical distribution with a Gaussian fit [20]. As an undesired consequence of using smoothing priors, the inferred parameter maps may exhibit biased (flattened) values.

Conversely, if we want to detect sharp patterns or extreme values, the hyperparameters should be tuned to the smallest values that still let the inference converge and make the inferred maps exhibit as few holes (undefined parameter values) as possible within the regions of interest.

Another approach, applied in [13] (Supplementary Information), tunes the spatial regularisation hyperparameters using large-scale numerical simulations based on realistic parameter landscapes. Numerical trajectories are generated so that they mimick experimental trajectories, and a grid search is performed to find the hyperparameter values that minimise the discrepancy (*reconstruction error*) between the inferred and expected maps.

Multiple procedures for tuning hyper-parameters can be envisioned depending on the nature of the experiments and the expected results. While we recommend simulation-based exploration of the effects of hyperparameters, increasing the robustness of statistical tests is often taken as a priority in the analysis of biological data. Generally the two procedures are not mutually exclusive, but simulation-based tuning can be computer-intensive. Furthermore, if the regions of biological interest are highly local in space and/or time, minimising the reconstruction error will tend to put most weight on domains that are of lesser interest because there are more of these.

The experimental dataset introduced in this article is an example of such a complex inference with sparse regions of interest. The hyperparameters were set to the following values: $\mu_{\mathbf{r}} = 1$, $\mu_t = 10$, $\lambda_{\mathbf{r}} = 0.1$ and $\lambda_t = 1$.

Spatio-temporal mapping of Gag protein dynamics

We illustrate our inference framework by inferring a spatio-temporal map of the dynamics wild-type mEOS2-tagged HIV-1 Gag proteins in a single CD4$^+$ T cell. The Gag protein is involved in the budding of HIV-1 virus particles at the plasma membrane and, when expressed alone, is sufficient to form inactive virus-like particles (VLPs) [20].

We mapped the dynamics in a 3 μm $\times$ 3 μm square of the plasma membrane over 15 min at spatial and temporal resolutions of 20 nm and 1 min (see *Methods* for a detailed description of the dataset and analysis). To observe transient interactions with possibly shorter du-

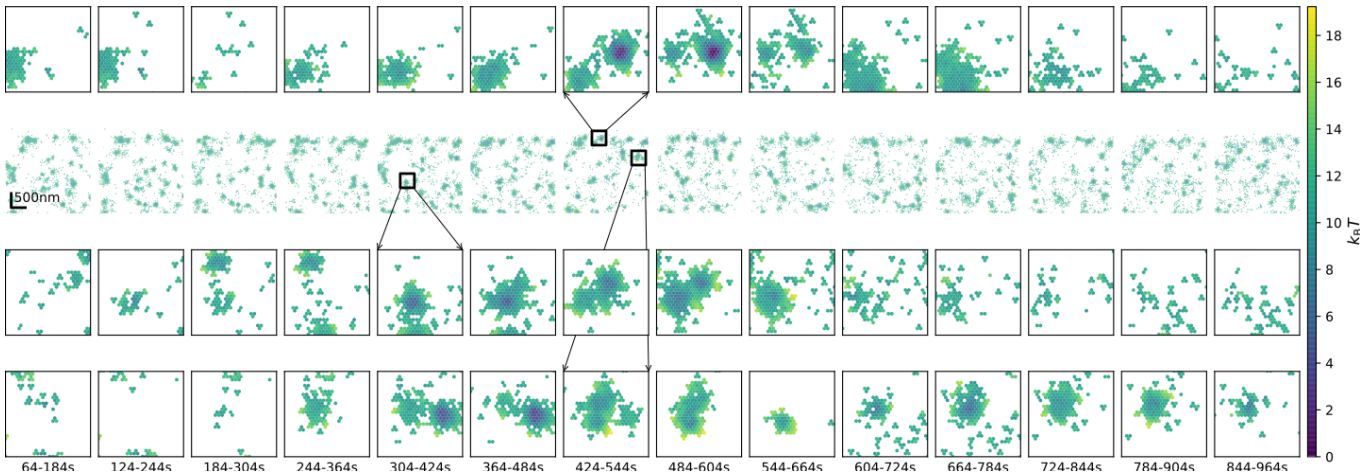

FIG. 7. Potential energy maps of Gag proteins in part of a T cell (127 578 parameters in total and 126 022 recorded localisations). The second row shows the full inferred map. The rows above and below show magnifications of spatial regions of interest. White domains were not visited by any molecule and not represented (no associated parameter).

rations than the classically observed 20-minute particle budding, we segmented time using a 2-minute sliding time window and a 1-minute resolution. This is to be compared to previous analyses that ran a 4-minute time window [20].

The dataset exhibited low density—white areas in Fig. 7 were not visited by any molecule during the corresponding time segment. Clear patterns could still be identified, such as the $\sim$100 nm energy trap illustrated on the top row. This trap lasted less than 2 minutes, which is indicative of transient assembly that did not lead to a fully assembled VLP [20]. It is worth noting that these short events cannot be detected based on the local density of Gag and that information still lacks on VLP that did not assemble. Other weaker traps were also identified, such the ones illustrated on the third and fourth rows.

A key benefit of regularising in time is the homogeneity of inferred potential energies between distant locations and distinct time segments. This is not the case when individually inferring spatial maps for different time segments since potential energies are only determined up to an additive constant. The temporal regularisation made these estimated energies directly comparable between one another.

In spite of the 127 578 parameters, the inference took less than 24 hours on a desktop computer with an Intel© Xeon© E5-2687W central processing unit.

CONCLUSIONS

Here, we introduced a Bayesian framework for fully regularised inference of spatio-temporal maps of intracellular biomolecule dynamics from their recorded trajectories. To solve the computational optimisation problem of inferring the most probable (*maximum a posteriori* – MAP) values of the thousands of map parameters we relied on stochastic optimisation – a general class of optimisation methods well suited to high-dimensional inferences and large datasets. We exploited the particular physical structure of the mapping problem by adapting the quasi-Newton BFGS algorithm to make use of spatio-temporally local minibatches, which provide particularly efficient updates as they involve only small parameter subspaces while separating experimental localisations that are linked to a limited set of parameters. While our focus was on MAP estimation, we additionally provided a complementary procedure based on a modern technique from Bayesian statistics and machine learning to access the full posterior distribution: sampling of the posterior around the MAP using stochastic Gradient Langevin dynamics. We demonstrated the advantage of stochastic optimisation over direct optimisation of the full posterior, and, for the first time, we mapped the spatio-temporal dynamics of HIV-1 Gag proteins on large areas with temporal regularisation.

[1] S. Manley, J. M. Gillette, G. H. Patterson, H. Shroff, H. F. Hess, E. Betzig, and J. Lippincott-Schwartz, Nature Methods **5**, 155 (2008).
[2] G. Giannone, E. Hosy, F. Levet, A. Constals, K. Schulze, A. I. Sobolevsky, M. P. Rosconi, E. Gouaux, R. Tamp, D. Choquet, and L. Cognet, Biophysical Journal **99**, 1303 (2010).
[3] J. Meier, C. Vannier, A. Serg, A. Triller, and D. Choquet, Nature Neuroscience **4**, 253 (2001).
[4] J. Elf, G.-W. Li, and X. S. Xie, Science **316**, 1191 (2007).
[5] R. Das, C. W. Cairo, and D. Coombs, PLoS Computational Biology **5**, e1000556 (2009).

[6] P. Pierobon, S. Achouri, S. Courty, A. R. Dunn, J. A. Spudich, M. Dahan, and G. Cappello, Biophysical Journal **96**, 4268 (2009).

[7] J.-B. Masson, D. Casanova, S. Türkcan, G. Voisinne, M.-R. Popoff, M. Vergassola, and A. Alexandrou, Physical Review Letters **102**, 048103 (2009).

[8] A. V. Weigel, B. Simon, M. M. Tamkun, and D. Krapf, Proceedings of the National Academy of Sciences of the United States of America **108**, 6438 (2011).

[9] N. Hoze, D. Nair, E. Hosy, C. Sieben, S. Manley, A. Herrmann, J.-B. Sibarita, D. Choquet, and D. Holcman, Proceedings of the National Academy of Sciences of the United States of America **109**, 17052 (2012).

[10] D. Nair, E. Hosy, J. D. Petersen, A. Constals, G. Giannone, D. Choquet, and J.-B. Sibarita, The Journal of Neuroscience **33**, 13204 (2013).

[11] C. G. Specht, I. Izeddin, P. C. Rodriguez, M. El Beheiry, P. Rostaing, X. Darzacq, M. Dahan, and A. Triller, Neuron **79**, 308 (2013).

[12] F. Persson, M. Lindén, C. Unoson, and J. Elf, Nature Methods **10**, 265 (2013).

[13] J.-B. Masson, P. Dionne, C. Salvatico, M. Renner, C. G. Specht, A. Triller, and M. Dahan, Biophysical Journal **106**, 74 (2014).

[14] C. Manzo and M. F. Garcia-Parajo, Reports on Progress in Physics **78**, 124601 (2015).

[15] N. Monnier, Z. Barry, H. Y. Park, K.-C. Su, Z. Katz, B. P. English, A. Dey, K. Pan, I. M. Cheeseman, R. H. Singer, *et al.*, Nature Methods **12**, 838 (2015).

[16] E. J. Akin, L. Solé, B. Johnson, M. el Beheiry, J.-B. Masson, D. Krapf, and M. M. Tamkun, Biophysical Journal **111**, 1235 (2016).

[17] T. Sungkaworn, M.-L. Jobin, K. Burnecki, A. Weron, M. J. Lohse, and D. Calebiro, Nature **550**, 543 (2017).

[18] A. Remorino, S. De Beco, F. Cayrac, F. Di Federico, G. Cornilleau, A. Gautreau, M. C. Parrini, J.-B. Masson, M. Dahan, and M. Coppey, Cell Reports **21**, 1922 (2017).

[19] D. Holcman, P. Parutto, J. E. Chambers, M. Fantham, L. J. Young, S. J. Marciniak, C. F. Kaminski, D. Ron, and E. Avezov, Nature Cell Biology **20**, 1118 (2018).

[20] C. Floderer, J.-B. Masson, E. Boilley, S. Georgeault, P. Merida, M. El Beheiry, M. Dahan, P. Roingeard, J.-B. Sibarita, C. Favard, *et al.*, Scientific Reports **8**, 16283 (2018).

[21] I. Sgouralis and S. Presse, Biophysical Journal **112**, 2021 (2017).

[22] N. Hoze and D. Holcman, Physical Review E **92**, 052109 (2015).

[23] D. Holcman, N. Hoze, and Z. Schuss, Biophysical Journal **109**, 1761 (2015).

[24] M. El Beheiry, M. Dahan, and J.-B. Masson, Nature Methods **12**, 594 (2015).

[25] E. F. Koslover, C. K. Chan, and J. A. Theriot, Biophysical Journal **110**, 700 (2016).

[26] A. Frishman and P. Ronceray, "Learning force fields from stochastic trajectories," (2018), arXiv:1809.09650v2.

[27] H. Robbins and S. Monro, The Annals of Mathematical Statistics **22**, 400 (1951).

[28] J. C. Spall, *Introduction to Stochastic Search and Optimization* (Wiley-Interscience, 2003).

[29] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," (2013), arXiv:1312.6114.

[30] M. Welling and Y. W. Teh, in *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11 (Omnipress, USA, 2011) pp. 681–688.

[31] J. C. Chang, P.-W. Fok, and T. Chou, Biophysical Journal **109**, 966 (2015).

[32] G. Hummer and I. G. Kevrekidis, Journal of Chemical Physics **118**, 10762 (2003).

[33] A. S. Serov, F. Laurent, C. Floderer, K. Perronet, C. Favard, D. Muriaux, C. L. Vestergaard, and J.-B. Masson, "Robust inference of forces in heterogeneous environments," (2019), arXiv:1903.03048.

[34] R. E. Kass, A. E. Raftery, S. Association, and N. Jun, J. Am. Stat. Assoc. **90**, 773 (1995).

[35] G. Peyré, *Mathematical Foundations of Data Sciences*.

[36] H. W. Engl, M. Hanke, and A. Neubauer, *Regularization of inverse problems*, Vol. 375 (Springer Science & Business Media, 1996).

[37] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical recipes 3rd edition: The art of scientific computing* (Cambridge university press, 2007).

[38] T. Savin and P. S. Doyle, Biophysical Journal **88**, 623 (2005).

[39] C. L. Vestergaard, P. C. Blainey, and H. Flyvbjerg, Physical Review E, **89**, 022726 (2014).

[40] A. J. Berglund, Physical Review E **82**, 011917 (2010).

[41] C. L. Vestergaard, Physical Review E **94**, 022401 (2016).

SUPPLEMENTARY INFORMATION

SAMPLING THE POSTERIOR DISTRIBUTION

Here, we relied on stochastic optimisation to access the maximum *a posteriori* (MAP) values of the map parameters. Sampling of the posterior distribution around the MAP can be performed using a similar procedure, *i.e.* stochastic gradient Langevin dynamics [30]. It uses the following update rule:

$$\boldsymbol{\theta}^{(k)} = \boldsymbol{\theta}^{(k-1)} + \frac{\epsilon}{2}\nabla P\left(\boldsymbol{\theta}|\{\Delta\mathbf{r}\}\right) + \sqrt{\epsilon}\,\boldsymbol{\eta}_k \ , \tag{14}$$

where η_k follows a standard Gaussian distribution, $\eta_k \sim \mathcal{N}(0,1)$. Equation (14) can be recognised as the Euler approximation to an overdamped Langevin equation with the posterior distribution acting as an effective potential and ϵ as a diffusivity. This procedure was shown to allow efficient sampling of the posterior distribution, and it converges quickly when initialised at the MAP values of the parameters. Note that since we here initialise the procedure at the MAP values for $\boldsymbol{\theta}$, we do not need to make ϵ vary with time as in the original publication [30] where the scheme was also used to optimise the posterior before sampling around the MAP.

STOCHASTIC OPTIMISATION

We here give a detailed description of the proposed stochastic optimisation algorithm.

Each physical parameter ϕ (indexed by ℓ) in the set of model parameters $\boldsymbol{\theta}$, namely the diffusivity D and the potential energy V, is represented by as many model parameters as spatial subdomains (indexed by α) times temporal segments (indexed by τ). We may write the set of parameters formally as: $\boldsymbol{\theta} = \cup_\ell \boldsymbol{\theta}_\ell = \cup_{\alpha,\tau}\cup_\ell \{\theta_{\ell,\alpha,\tau}\} = \cup_{\alpha,\tau}\{D_{\alpha,\tau}, V_{\alpha,\tau}\}$.

To show the generality of the stochastic approach, we here consider a more general formulation that in the main text by letting $\mathcal{B}_{\alpha,\tau}$ be any neighbourhood of (α,τ), instead of just being comprised of nearest neighbours, and by letting $\boldsymbol{\theta}_{\alpha,\tau}$ be any collection of local parameters, e.g. mobility and drift $\boldsymbol{\theta}_{\alpha,\tau} = (\psi_{\alpha,\tau}, \mathbf{a}_{\alpha,\tau})$ or drag and force $\boldsymbol{\theta}_{\alpha,\tau} = (\gamma_{\alpha,\tau}, \mathbf{f}_{\alpha,\tau})$. In this more general case, the local cost can be written as

$$\begin{aligned} f_{\alpha,\tau}(\boldsymbol{\theta}_{\mathcal{B}_{\alpha,\tau}}) = &-\log(P(\boldsymbol{\theta}_{\mathcal{R}_{\alpha,\tau}}|\{\Delta\mathbf{r}\}_{\alpha,\tau})) \\ &+\sum_\ell \left[\mu_{\ell,\mathbf{r}} q_\alpha(\boldsymbol{\theta}_{\ell,\mathcal{R}_\alpha,\tau}) + \mu_{\ell,t} q_\tau(\boldsymbol{\theta}_{\ell,\alpha,\mathcal{T}_\tau})\right] \ . \end{aligned} \tag{15}$$

Parsimonious gradient calculation

The cost function f being a sum of local components makes partial evaluations of the gradient possible. For example, considering any increment $\epsilon > 0$ along a single scalar parameter $\boldsymbol{\theta}^+ = \boldsymbol{\theta} + \epsilon\,\mathbb{1}_{\ell,\alpha,\tau}$ and $\boldsymbol{\theta}^- = \boldsymbol{\theta} - \epsilon\,\mathbb{1}_{\ell,\alpha,\tau}$, we have:

$$\begin{aligned} f(\boldsymbol{\theta}^+) - f(\boldsymbol{\theta}^-) = &\sum_{\alpha'\in\mathcal{R}_\alpha}\left[f_{\alpha',\tau}\left(\boldsymbol{\theta}^+_{\mathcal{B}_{\alpha',\tau}}\right) - f_{\alpha',\tau}\left(\boldsymbol{\theta}^-_{\mathcal{B}_{\alpha',\tau}}\right)\right] \\ &+\sum_{\tau'\in\mathcal{T}_\tau}\left[f_{\alpha,\tau'}\left(\boldsymbol{\theta}^+_{\mathcal{B}_{\alpha,\tau'}}\right) - f_{\alpha,\tau'}\left(\boldsymbol{\theta}^-_{\mathcal{B}_{\alpha,\tau'}}\right)\right] \ . \end{aligned} \tag{16}$$

Only a few local functions $\{f_{\alpha,\tau}\}$ are evaluated above.

The above calculus is the basis for gradient calculation. Indeed, the component (ℓ,α,τ) of the gradient ∇f is calculated as: $\nabla f_{\ell,\alpha,\tau}(\boldsymbol{\theta}) = \frac{f(\boldsymbol{\theta}^+) - f(\boldsymbol{\theta}^-)}{\epsilon}\,\mathbb{1}_{\ell,\alpha,\tau}$ for given ϵ, chosen as described in [37]. In comparison, a default implementation would evaluate the full cost function for each component of the gradient.

From decomposability to stochasticity

At each iteration k of the optimisation algorithm, the parameters are updated in an approximate fashion: $\boldsymbol{\theta}^{(k)} = \boldsymbol{\theta}^{(k-1)} + \Delta\boldsymbol{\theta}(\{\Delta\mathbf{r}\}, \boldsymbol{\theta}^{(k-1)})$. The decomposition of f makes it possible minimize the cost function w.r.t. local parameters $\boldsymbol{\theta}_{\alpha,\tau}$ only at each iteration, and visit the parameters sequentially instead of jointly according to $\boldsymbol{\theta}^{(k)}_{\alpha,\tau} = \boldsymbol{\theta}^{(k-1)}_{\alpha,\tau} + \Delta\boldsymbol{\theta}(\Delta\mathbf{r}_{\mathcal{B}_{\alpha,\tau}}, \boldsymbol{\theta}^{(k-1)}_{\mathcal{S}_{\alpha,\tau}})$. Calculating the update $\Delta\boldsymbol{\theta}$ requires only subsets $\mathcal{B}_{\alpha,\tau}$ and $\mathcal{S}_{\alpha,\tau}$ of the full data and parameter sets, respectively.

If only gradient calculation and approximate line search are required, the smallest possible minibatch of data $\Delta\mathbf{r}_{\mathcal{B}_{\alpha,\tau}} = \{\Delta\mathbf{r}_{\alpha',\tau'}\}_{(\alpha',\tau')\in\mathcal{B}_{\alpha,\tau}}$ is such that: $\mathcal{B}_{\alpha,\tau} = (\mathcal{R}_\alpha\times\{\tau\})\cup(\{\alpha\}\times\mathcal{T}_\tau)$, which is what we used in the main text. Here the notations $\{k\}$ and $\{t\}$ represent singletons and $\times$ the Cartesian (outer) product.

Similarly, the smallest possible subset of the parameters (*subspace*) that is accessed due to the regularizing priors is $\mathcal{S}_{\alpha,\tau} = \cup_{\alpha',\tau'\in\mathcal{B}_{\alpha,\tau}}\mathcal{B}_{\alpha',\tau'}$.

Altogether, we consider a minibatch sampling function:

$$\mathcal{B} : k \longrightarrow (\alpha,\tau) \longrightarrow \{\Delta\mathbf{r}_{\alpha',\tau'}\}_{(\alpha',\tau')\in\mathcal{B}_{\alpha,\tau}}$$

This implies several possible optimization paths. If either the $k \longrightarrow (\alpha,\tau)$ or the $(\alpha,\tau) \longrightarrow \{\Delta\mathbf{r}_{\alpha',\tau'}\}_{(\alpha',\tau')\in\mathcal{B}_{\alpha,\tau}}$ parts acts randomly, we thus define a stochastic approach.

Line search for descending the gradient in a subspace

As described in the *Methods* section, the update at iteration k takes the following form: $\Delta\boldsymbol{\theta}(\Delta\mathbf{r}_{\mathcal{B}_{\alpha,\tau}}, \boldsymbol{\theta}^{(k-1)}_{\mathcal{S}_{\alpha,\tau}}) = s^{(k)}\mathbf{p}^{(k)}$, with descent direction given by $\mathbf{p}^{(k)} = -\mathcal{H}^{(k-1)}\nabla f_{\alpha,\tau}(\boldsymbol{\theta}^{(k-1)})$.

The step size $s^{(k)}$ can be estimated in various ways. Here, in the context of minimizing f only w.r.t. the parameters $\boldsymbol{\theta}_{\alpha,\tau}$ (i.e. in the subspace $\mathcal{D}_{\alpha,\tau} = \Phi \times \{\alpha\} \times \{\tau\}$, where $\Phi = \{D, V\}$), $s^{(k)}$ is best estimated using an approximate line search. Indeed, for each cell (α, τ), a line search operates in a subspace of small dimensionality, $|\mathcal{D}_{\alpha,\tau}|$ (here two), and interestingly the same partial evaluation as for gradient calculation stands. This simplification comes from the fact that an approximate line search selects a step on basis of differences in f along the descent direction (Armijo's rule), i.e. with only the parameters in $\mathcal{D}$ that vary, and optionally of differences in ∇f (Wolfe's rule) in $\mathcal{D}$ as well.

Consequently, such an approximate line search is efficient and there is no need to determine $s^{(k)}$ using an adaptive rule, which is another departure of our algorithm from most of the other stochastic optimization procedures.

Inverse Hessian approximation

The inverse Hessian $\mathcal{H}$ was updated using the BFGS rule [37]. Estimating the gradient and expressing the updates in small subspaces $\mathcal{D}_{\cdot}$ makes the updates to the inverse Hessian matrix similarly small ($\mathcal{D}_{\cdot} \times \mathcal{D}_{\cdot}$). This leads to the approximation for almost all parameter pairs that $\frac{\partial f}{\partial \theta_i \partial \theta_j}(\theta) \approx 0$. In our system this approximation is supported by the fact that separated regions in space and time have limited interactions regarding their parameters.

For example, the inference showcased in this article exhibits a block diagonal inverse Hessian matrix composed of 2×2 blocks in its stochastic version. We observed that the BFGS update rule for such a sparse inverse Hessian allowed faster convergence than a basic gradient descent approach, which corresponds to having the identity matrix replace $\mathcal{H}$. This is illustrated Figure 4.

In complex cases, or when some domains contain limited numbers of points, $\mathcal{H}$ may not be definite positive. In this case we replace it by the identity matrix.

Correcting spatial oscillations in V

As Figure S4 in [13] shows, direct optimisation of the posterior distribution can lead to a potential energy landscape exhibiting spatial oscillations (of amplitude $\approx 0.15\, k_{\mathrm{B}}T$ for a diffusivity of $D \approx 0.1\, \mu m^2 s^{-1}$) even in the absence of forces.

The posterior probability [Eq. (9)] only depends on the potential energy V through its spatial gradient ∇V. A gradient is insensitive to spatial oscillations on a regular mesh at the Nyquist frequency – ∇V is always evaluated on top of a hill or at the bottom of a trough and the variation in-between is just not sampled. This undesirable phenomenon, limited in full optimization, was amplified when locally and sequentially updating the parameters since the approximate line search generates noisy updates that are even noisier when the step sizes are driven by fewer parameters.

Regularising priors lessen the problem as they add a damping factor. However, when attempting to detect subtle effects and features in the map, it is not desirable to increase regularisation beyond what is needed to attenuate statistical errors. We instead introduced two modifications of the approximate line search, which also make the algorithm more robust in general:

- setting an upper bound on individual updates,
- adapting Wolfe's rule in the approximate line search to prevent gradient reversal.

An upper bound on the infinite-norm of the update allows to scale the initial update considered in the line search (for our stochastic algorithm, the infinite norm is equal to the maximum of the absolute values of $D_{\alpha,\tau}$ and $V_{\alpha,\tau}$). This mostly makes the search faster and better uses a limited number of iterations (default is 10), but can also prevent large updates. We set this upper bound equal to $2k_{\mathrm{B}}T$ for V and $2\mu\mathrm{m}^2\mathrm{s}^{-1}$ for D based on physical considerations, though the precise value of the bound is not important as its main purpose is to avoid diverging updates.

The standard strong Wolfe's rule [37] aims at making the norm of the projected gradient decrease (with iteration index k shown as subscript for convenience):

$$c\mathbf{p}_k^T \nabla f(\boldsymbol{\theta}_{k-1}) \leq -\mathbf{p}_k^T \nabla f(\boldsymbol{\theta}_{k-1} + s_k \mathbf{p}_k) \leq -c\mathbf{p}_k^T \nabla f(\boldsymbol{\theta}_{k-1}) \ , \tag{17}$$

with $c = 0.9$. We propose instead:

$$c\mathbf{p}_k^T \nabla f(\boldsymbol{\theta}_{k-1}) \leq -\mathbf{p}_k^T \nabla f(\boldsymbol{\theta}_{k-1} + s_k \mathbf{p}_k) \ , \tag{18}$$

with $c = 0.5$ (or lower).

Our first motivation here is to bound more tightly the projected gradient $\mathbf{p}_k^T \nabla f(\boldsymbol{\theta}_{k-1} + s_k \mathbf{p}_k)$ when it points opposite to $\mathbf{p}_k^T \nabla f(\boldsymbol{\theta}_{k-1})$. Indeed, a gradient reversal may be the signature of a large step and cause oscillations around the optimal parameter value, especially with local updates. With $c = 0.5$, a reversed gradient is admitted only if it is half the size of the initial gradient, along the descent direction.

A second motivation in dropping the weak part of Wolfe's rule deals with non-monotonic properties in the descent subspaces. Empirically, situations revealed cases where the projected gradient was only increasing along the descent direction, while the candidate updates complied with the Armijo rule [37] (decrease in f). As long as the Armijo rule is ensured, if the gradient keeps on pointing in the same direction, we only need to carry out more steps. Especially, if the projected gradient increases in absolute value, then a small step is preferable in order to stay in smooth conditions, so that the inverse Hessian matrix can be properly updated.

Parallelisation of the optimisation

The main constraint in parallelising the computation performed on minibatches is to ensure non-conflicting access to resources during execution of the various tasks. We implemented the parallel procedure the following way.

A scheduler process operates the $k \longrightarrow (\alpha, \tau)$ sampling function and dispatches single iterations (tasks) to a pool of workers. Each worker will process an iteration and consequently update the parameters of a given spatio-temporal domain, while another worker will simultaneously process another iteration. Parameters involved in the computations run by a worker are not modified during computation by any other workers.

The parameters involved at iteration k are identified by the following function: $\mathcal{S} : k \longrightarrow (\alpha, \tau) \longrightarrow \boldsymbol{\theta}_{\mathcal{S}_{\alpha,\tau}}$, where $\boldsymbol{\theta}_{\mathcal{S}_{\alpha,\tau}} = \{\boldsymbol{\theta}_{\alpha',\tau'}\}_{(\alpha',\tau')\in\mathcal{S}_{\alpha,\tau}}$. Each time the scheduler assigns an iteration to a domain (α, τ), it can lock the corresponding $\mathcal{S}_{\alpha,\tau}$ parameters. The later iterations that require access to locked parameters will be postponed until the locked parameters are released. Finally, note that to allow for more parallelisation on small maps, we locked $\mathcal{B}_{\alpha,\tau}$ instead of $\mathcal{S}_{\alpha,\tau}$.

More precisely, each worker maintains its own copy of the parameter vector $\boldsymbol{\theta}$. The workers synchronise together, exchanging local parameter updates $\boldsymbol{\theta}^{(k)}_{\alpha,\tau}$ and local blocks $\mathcal{H}^{(k)}_{\alpha,\tau}$ of the inverse Hessian matrix. All workers send status information to the scheduler, this merely includes the decrease in f: $\Delta f^{(k)} = f\left(\boldsymbol{\theta}^{(k-1)}\right) - f\left(\boldsymbol{\theta}^{(k)}\right)$. Convergence is stated once Δf drops below some small tolerance for 90% of the iterations in an epoch of ΩT iterations, where ΩT is the number of domains in the map.

To ensure that all domains are updated, they are each sampled once, in random order, during each epoch. To avoid potential latencies between epochs, the next epoch begins as soon as all domains have been sampled; the scheduler does not wait for the workers to complete the corresponding updates.

TIME REGULARISATION AND INITIAL VALUES FOR V

In the context of inference of a potential energy field, time regularization is especially useful in making potential energies comparable between different time segments. Nothing in the underlying physics fixes the average value of the effective potential, creating complexities when evaluating the effective potential at different time points.

In InferenceMAP [24], the potential fields were initialized to match the corresponding equilibrium distribution to the recorded point density. When temporal regularisation is imposed, direct optimisation does not necessarily constrain the effective potential (in non interacting space domains) for small values of λ_t. To make temporal smoothing less sensitive to the value of λ_t and to ensure similar levels between time segments, we initialised V to 0 instead, for inferences with temporal regularisation.