nature research

https://doi.org/10.1038/s42256-020-00242-6

Supplementary information

Moving beyond generalization to accurate interpretation of flexible models

In the format provided by the authors and unedited

Supplementary Information for: Moving beyond generalization to accurate interpretation of flexible models

Mikhail Genkin and Tatiana A. Engel

Cold Spring Harbor Laboratory, Cold Spring Harbor, NY 11724

Corresponding author e-mail: engel@cshl.edu

September 1, 2020

Contents

1 Materials and methods

1.1 Equivalent parametrizations of Langevin dynamics

In our modelling framework, the latent Langevin dynamics are controlled by the potential function $\Phi(x)$. The same Langevin dynamics can be equivalently represented in terms of the driving force $F(x)$ or the equilibrium probability distribution $p_{\text{eq}}(x)$. These parametrizations of the Langevin dynamics are equivalent, as every potential $\Phi(x)$ has the corresponding $F(x)$ and p_{eq}, and vice

versa. These three quantities are related to each other via:

$$\begin{aligned} F(x) &= -\frac{d\Phi(x)}{dx}, \quad F(x) = \frac{1}{p_{\mathrm{eq}}}\frac{dp_{\mathrm{eq}}(x)}{dx}, \\ \Phi(x) &= -\log p_{\mathrm{eq}}(x), \\ p_{\mathrm{eq}}(x) &= \frac{\exp\left(\int_{-1}^{x} F(x')dx'\right)}{\int_{-1}^{1}\exp\left(\int_{-1}^{x} F(x')dx'\right)dx}, \quad p_{\mathrm{eq}}(x) \;\; = e^{-\Phi(x)}. \end{aligned} \tag{1}$$

In Eq. (1), the arbitrary additive constant in the potential is fixed by the normalization condition for $p_{\mathrm{eq}}(x)$: $\int_{-1}^{1} e^{-\Phi(x)}dx = 1$, which provides unique relationships between the potential, driving force, and equilibrium probability density. In this work, we perform GD optimization by updating the driving force $F(x)$. We visualize results using the potential $\Phi(x)$ that is calculated from the force by taking an antiderivative.

Note that definite integrals over the x-domain extend from -1 to 1. The range of x-domain is arbitrary, since the latent space does not correspond to any real physical quantity. We use the $[-1, 1]$ range throughout this work. The results would not change if a different range was chosen.

1.2 Likelihood calculation

Analytical derivation of the likelihood functional is briefly outlined here (see Ref.[21] for additional details). The likelihood is calculated via marginalization of the joint probability density $P(\mathcal{X}(t), Y(t)|F(x))$ of the observed spike-data $Y(t)$ and the latent trajectory $\mathcal{X}(t)$:

$$\mathscr{L}\left[Y(t)|F(x)\right] = \int \mathscr{D}\mathcal{X}(t)\; P\left(\mathcal{X}(t), Y(t)|F(x)\right). \tag{2}$$

The Markov property of the latent Langevin dynamics entails that the transition probability density over the latent space $p(x_{t_k}, t_k|x_{t_{k-1}}, t_{k-1})$ does not depend on the intermediate states and can be marginalized over all paths connecting $x_{t_{k-1}}$ and x_{t_k}. By calculating $p(x_{t_k}, t_k|x_{t_{k-1}}, t_{k-1})$, we can therefore effectively reduce the continuous path $\mathcal{X}(t)$ to a discrete set of points $X(t) = \{x_{t_0}, x_{t_1}, ..., x_{t_N}\}$. Using the assumption that spikes are independent when conditioned on the latent states, the joint probability density $P(X(t), Y(t))$ takes the form:

$$P(X(t), Y(t)) = p(x_{t_0}) \prod_{k=1}^{N} p(y_{t_i}|x_{t_i}) p(x_{t_i}|x_{t_{i-1}}). \tag{3}$$

Here $t_1, t_2, ..., t_N$ are the observed spike times and t_0 is the experiment onset time. $X(t) = \{x_{t_0}, x_{t_1}, ..., x_{t_N}\}$ is a latent trajectory at the times of spike events. $Y(t) = \{t_1, t_2, ..., t_N\}$ are the observed spike times. $p(x_{t_i}|x_{t_{i-1}})$ is the transition probability density over the latent space from $x_{t_{i-1}}$ to x_{t_i} during the time between adjacent spike observations. $p(y_{t_i}|x_{t_i})$ is the probability of spike emission given the latent state. Note that $X(t)$ is a discretized latent path sampled at $N+1$ time points. Marginalization over all intermediate time points is already performed within the transition probability densities. The likelihood is therefore calculated by marginalizing Eq. (3) over the remaining latent space variables x_{t_i}:

$$\mathscr{L} = \int_{x_{t_0}} \int_{x_{t_1}} \cdots \int_{x_{t_N}} dx_{t_0} dx_{t_1} \ldots dx_{t_N} P(X(t), Y(t)). \tag{4}$$

The transition probability density over the latent space is a solution of the generalized Fokker-Planck equation [21]:

$$\frac{\partial p(x,t)}{\partial t} = \left(-D\frac{\partial}{\partial x}F(x) + D\frac{\partial}{\partial x^2} - f(x)\right)p(x,t) \equiv -\hat{\boldsymbol{\mathcal{H}}}p(x,t). \tag{5}$$

The first two terms in the operator $\hat{\boldsymbol{\mathcal{H}}}$ are the usual drift and diffusion terms that originate from the Langevin dynamics (Eq. (1) in the main text). The last term $-f(x)$ accounts for the fact that no spikes are observed during the time intervals between each pair of adjacent spikes..

It is more convenient to work with a Hermitian form of Eq. (5) [21]. Therefore, we use the transformation $\rho(x,t) = p(x,t)\exp(\Phi(x)/2)$ and $\boldsymbol{\mathcal{H}} = \exp(\Phi(x)/2)\hat{\boldsymbol{\mathcal{H}}}\exp(-\Phi(x)/2)$, which turns Eq. (5) into [52]:

$$\frac{\partial \rho(x,t)}{\partial t} = -\boldsymbol{\mathcal{H}}\rho(x,t). \tag{6}$$

The new operator $\boldsymbol{\mathcal{H}}$ is Hermitian. We use Eq. (6) to evaluate $\rho(x,t)$ over time. The original probability density $p(x,t)$ is recovered by $p(x,t) = \rho(x,t)\exp(-\Phi(x)/2)$. We also introduce the equilibrium probability density $p_{\text{eq}}(x) = \exp(-\Phi(x))$, where we choose the arbitrary scaling of the potential $\Phi(x)$ so that $p_{\text{eq}}(x)$ is normalized. Then $p(x,t) = \rho(x,t)\sqrt{p_{\text{eq}}(x)}$.

The formal solution of Eq. (6) can be written as the operator exponential:

$$\rho(x_{t_i}|x_{t_{i-1}}) = e^{-\boldsymbol{\mathcal{H}}\Delta t_i}, \tag{7}$$

where $\Delta t_i = t_i - t_{i-1}$. To compute the operator $e^{-\boldsymbol{\mathcal{H}}\Delta t_i}$ efficiently for any Δt_i, we use the eigenfunctions $\Psi(x)$ and eigenvalues λ of the operator $\boldsymbol{\mathcal{H}}$, which satisfy

$$\boldsymbol{\mathcal{H}}\Psi(x) = \lambda\Psi(x). \tag{8}$$

We solve this eigenvalue problem numerically. To this end, we discretize the continuous domain x using the Spectral Elements Method (SEM), and solve for the eigenvalues λ_i and the eigenvectors $Q_{ij} = \Psi_j(x_i)$ of the discretized operator $\boldsymbol{\mathcal{H}}$ (see Section 2 for details). The matrix $\boldsymbol{Q}$ contains all eigenvectors of the discretized $\boldsymbol{\mathcal{H}}$. We truncate the basis $\boldsymbol{Q}$ to the 64 eigenvectors corresponding to the 64 largest eigenvalues. To confirm that this number is sufficient, we also used 128 eigenvectors and observed identical results.

In the eigenbasis $\boldsymbol{Q}$ of the operator $\boldsymbol{\mathcal{H}}$, any continuous function $g(x)$ is represented by a vector $\boldsymbol{g}$ with elements equal to the expansion coefficients in this basis:

$$g_i = \int_x g(x)\Psi_i(x)dx. \tag{9}$$

The operator exponential $e^{-\boldsymbol{\mathcal{H}}\Delta t_k}$ is represented by a transition matrix $\boldsymbol{A}_k$ with the elements:

$$A_{k,ij} = \int_x \Psi_i(x)e^{-\boldsymbol{\mathcal{H}}\boldsymbol{\Delta t_k}}\Psi_j(x)dx = \delta_{ij}e^{-\lambda_i\Delta t_k}, \tag{10}$$

where we took advantage of the eigenbasis properties.

The probability density of a spike emission at a given latent state $p(y_{t_i}|x_{t_i})$ is given by $p(y|x) = f(x)$, just by the definition of the instantaneous firing rate of a Poisson process. The corresponding emission operator is represented by the matrix $\boldsymbol{B}$ in the basis $\boldsymbol{Q}$, which is calculated using the rules of basis transformations:

$$\boldsymbol{B} = \boldsymbol{Q}^{\boldsymbol{T}}\boldsymbol{W}\boldsymbol{F}\boldsymbol{Q}. \tag{11}$$

Here the diagonal matrix $\boldsymbol{F}$ represents the emission operator in the SEM basis, $\boldsymbol{F} = \mathrm{diag}\,(\boldsymbol{f})$, $\boldsymbol{f}$ is discretization of firing rate function $f(x)$ on SEM grid, and $\boldsymbol{W}$ is a diagonal matrix of SEM weights (see Section 2).

With the developed formalism, Eq. (4) turns into a chain of matrix multiplications, where matrix-vector products automatically account for the marginalization over the intermediate variables $x_{t_0}, x_{t_2}, ...x_{t_{N-1}}$. To marginalize over the last latent position x_{t_N}, we multiply this chain by a column vector $\boldsymbol{\beta_{N+1}}$:

$$\mathscr{L} = \boldsymbol{\alpha}_0^T \boldsymbol{A}_1 \boldsymbol{B} \boldsymbol{A}_2 \boldsymbol{B} ... \boldsymbol{A}_N \boldsymbol{B} \boldsymbol{\beta_{N+1}}. \tag{12}$$

Here $\boldsymbol{\alpha}_0 = \boldsymbol{Q}^T \boldsymbol{W} \boldsymbol{\rho_0}$ is a vector representing the initial probability distribution transformed into the basis $\boldsymbol{Q}$ (the vector $\boldsymbol{\rho_0}$ is a discretization of $\rho(x, 0)$ in the SEM basis). The initial condition is chosen to be the equilibrium probability distribution, $p(x, 0) = p_{\mathrm{eq}}(x)$, so that $\rho(x, 0) = \sqrt{p_{\mathrm{eq}}(x)} = \rho_{\mathrm{eq}}(x)$. The terminal vector $\boldsymbol{\beta_N}$ is defined by the probability normalization condition (similar to the Hidden Markov Model, HMM [24]). Since

$$\frac{p(Y(t), x_N)}{\sqrt{p_{\mathrm{eq}}(x)}} = \boldsymbol{\alpha}_0^T \boldsymbol{A}_1 \boldsymbol{B} \boldsymbol{A}_2 \boldsymbol{B} ... \boldsymbol{A}_N \boldsymbol{B}, \tag{13}$$

we see that in order to satisfy the condition $\mathscr{L} = \int_{x_N} p(Y(t), x_N) dx$, we need to set $\boldsymbol{\beta_{N+1}} = \boldsymbol{Q}^T \boldsymbol{W} \boldsymbol{\rho_{eq}}$, which follows from the normalization conditions for the probability density $p_{\mathrm{eq}}(x)$.

In summary, given the model specified by $\theta = \{\Phi(x), f(x), D\}$, the likelihood is calculated using Eq. (12), where the operator matrices are found from Eqs. (10),(11) with the eigenvectors and eigenvalues of operator $\boldsymbol{\mathcal{H}}$ obtained from Eq. (8). More details on numerical implementation are given in Section 2.

In addition to matrix-vector form, for convenience we also use Dirac bra-ket notation. In this notation, row vectors correspond to bra-states $\langle \rho |$, column vectors to ket-state $| \rho \rangle$, and matrices to operators. With a choice of basis, the states are transformed into vectors and operators into matrices, so that both formalisms are equivalent. In Dirac notation Eq. (12) can be written as a chain of operators that propagate initial state $\langle \alpha_0 |$ towards the terminal state $| \beta_{N+1} \rangle$:

$$\mathscr{L}\left[Y(t), \Phi(x)\right] = \left\langle \alpha_0 \middle| e^{-\boldsymbol{\mathcal{H}} \Delta t_1} \boldsymbol{y}_{t_1} e^{-\boldsymbol{\mathcal{H}} \Delta t_2} \boldsymbol{y}_{t_2} ... e^{-\boldsymbol{\mathcal{H}} \Delta t_N} \boldsymbol{y}_{t_N} \middle| \beta_{N+1} \right\rangle, \tag{14}$$

where $\boldsymbol{y}_{t_k}$ is the spike emission operator.

We evaluate the likelihood using the forward-backward algorithm similar to that for HMMs [24]. Starting from the initial state α_0, the states $\alpha_1, \alpha_2, ..., \alpha_N$ are calculated recursively through the forward pass:

$$\langle \alpha_n | = \langle \alpha_{n-1} | e^{-\boldsymbol{\mathcal{H}} \Delta t_n} \boldsymbol{y}_{t_n}, \qquad n = 1, 2 \ldots N. \tag{15}$$

From Eq. (14), the likelihood $\mathscr{L} = \langle \alpha_N | \beta_{N+1} \rangle$. The subsequent backward pass is needed for the likelihood gradients calculations:

$$\begin{aligned} &| \beta_{n-1} \rangle = \boldsymbol{y}_{t_{n-1}} e^{-\boldsymbol{\mathcal{H}} \Delta t_n} | \beta_n \rangle, \qquad n = 2, 3 \ldots N, \\ &| \beta_N \rangle = \boldsymbol{y}_{t_N} | \beta_{N+1} \rangle, \\ &| \beta_0 \rangle = e^{-\boldsymbol{\mathcal{H}} \Delta t_1} | \beta_1 \rangle. \end{aligned} \tag{16}$$

1.3 Variational derivatives of the likelihood functional

We derive the gradient-descent algorithm for minimizing the negative log-likelihood. In contrast to the previous work [21], which used an approximate expectation-maximization algorithm, the

gradient-descent update rule can be derived exactly. This update requires to compute the variational derivative $\delta\mathscr{L}\left[Y(t)|F(x)\right]/\delta F(x)$. The dependence of likelihood on $F(x)$ is hidden in the operator $\boldsymbol{\mathcal{H}}$, see Eq. (14). Using the product rule, the variational derivative can be written as:

$$\frac{\delta\mathscr{L}}{\delta F(x)} = \sum_{\tau=0}^{N-1}\sum_{i,j} a_i(\tau)b_j(\tau+1)\frac{\delta\langle\Psi_i|e^{-\boldsymbol{\mathcal{H}}\Delta t_\tau}|\Psi_j\rangle}{\delta F} + \frac{\delta a_i(0)}{\delta F}b_i(0) + a_i(N)\frac{\delta b_i(N+1)}{\delta F}. \tag{17}$$

Here Ψ_i are the eigenvectors of $\boldsymbol{\mathcal{H}}$, and we introduced the notation:

$$a_i(\tau) = \langle\alpha_\tau|\Psi_i\rangle\,, \qquad b_j(\tau) = \langle\Psi_j|\beta_\tau\rangle\,. \tag{18}$$

For numerical stability, it is more convenient to optimize the log-likelihood $\log\mathscr{L}$, which has the derivative

$$\frac{\delta\log\mathscr{L}}{\delta F(x)} = \frac{1}{\mathscr{L}}\frac{\delta\mathscr{L}}{\delta F(x)}. \tag{19}$$

The factor $1/\mathscr{L}$ is a scalar scaling constant.

The first term on the right-hand side of Eq. (17) can be calculated using the formula for derivative of an exponentiated operator [53]:

$$\frac{\partial}{\partial\lambda}e^{-\beta\boldsymbol{H}} = -\int_0^\beta e^{-(\beta-u)\boldsymbol{H}}\frac{\delta\boldsymbol{H}}{\delta\lambda}e^{-u\boldsymbol{H}}du. \tag{20}$$

Using Eq. (20), one can derive:

$$\begin{aligned}
\frac{\delta\langle\Psi_i|e^{-\boldsymbol{\mathcal{H}}\Delta t_\tau}|\Psi_j\rangle}{\delta F} &= -\int_0^{\Delta t_\tau}\left\langle\Psi_i\middle|e^{-(\Delta t_\tau-u)\boldsymbol{\mathcal{H}}}\frac{\delta\boldsymbol{\mathcal{H}}}{\delta F}e^{-u\boldsymbol{\mathcal{H}}}\middle|\Psi_j\right\rangle du = \\
&= -\sum_{l,m}\int_0^{\Delta t_\tau}\left\langle\Psi_i\middle|e^{-(\Delta t_\tau-u)\boldsymbol{\mathcal{H}}}\middle|\Psi_l\right\rangle\left\langle\Psi_l\middle|\frac{\delta\boldsymbol{\mathcal{H}}}{\delta F}\middle|\Psi_m\right\rangle\left\langle\Psi_m\middle|e^{-u\boldsymbol{\mathcal{H}}}\middle|\Psi_j\right\rangle du = \\
&= -\sum_{l,m}\int_0^{\Delta t_\tau}\langle\Psi_i|\Psi_l\rangle\, e^{-(\Delta t_\tau-u)\lambda_i}\left\langle\Psi_l\middle|\frac{\delta\boldsymbol{\mathcal{H}}}{\delta F}|\Psi_m\right\rangle\langle\Psi_m|\Psi_j\rangle\, e^{-u\lambda_j}du = \\
&= -\left\langle\Psi_i\middle|\frac{\delta\boldsymbol{\mathcal{H}}}{\delta F}\middle|\Psi_j\right\rangle\int_0^{\Delta t_\tau}e^{-(\Delta t_\tau-u)\lambda_i}e^{-u\lambda_j}du \equiv -\left\langle\Psi_i\middle|\frac{\partial\boldsymbol{\mathcal{H}}}{\partial F}\middle|\Psi_j\right\rangle\Gamma^\tau_{i,j}.
\end{aligned} \tag{21}$$

Here λ are the eigenvalues of $\boldsymbol{\mathcal{H}}$, and $\Gamma^\tau_{i,j}$ is defined as [21]:

$$\Gamma^\tau_{ij} = \int_0^{\Delta t_\tau}e^{-(\Delta t_\tau-u)\lambda_i}e^{-u\lambda_j}du = \begin{cases}\Delta t_\tau e^{-\lambda_i\Delta t_\tau}, & i=j;\\ \frac{e^{-\lambda_i\Delta t_\tau}-e^{-\lambda_j\Delta t_\tau}}{\lambda_j-\lambda_i}, & i\neq j.\end{cases} \tag{22}$$

The term $\left\langle\Psi_i\middle|\frac{\partial\boldsymbol{\mathcal{H}}}{\partial F}\middle|\Psi_j\right\rangle$ can be calculated using the Euler-Lagrange equation. Operator $\boldsymbol{\mathcal{H}}$ can be written in the following form:

$$\boldsymbol{\mathcal{H}} = -D\nabla^2 + D\frac{F'(x)}{2} + D\frac{F^2(x)}{4} - f(x). \tag{23}$$

Accordingly,

$$\begin{aligned}
\left\langle\Psi_i\middle|\frac{\delta\boldsymbol{\mathcal{H}}}{\delta F}\middle|\Psi_j\right\rangle = \frac{\delta}{\delta F}D\int\Psi_i(x)\left(\frac{F'(x)}{2}+\frac{F(x)^2}{4}\right)\Psi_j(x)dx = \\
\frac{D}{2}\left(F(x)\Psi_i(x)\Psi_j(x) - \frac{d\left(\Psi_i(x)\Psi_j(x)\right)}{dx}\right).
\end{aligned} \tag{24}$$

This expression can be simplified:

$$\left\langle \Psi_i \middle| \frac{\delta \boldsymbol{\mathcal{H}}}{\delta F} \middle| \Psi_j \right\rangle = -\frac{D}{2} p_{eq}(x) \frac{d\left(\phi_i(x)\phi_j(x)\right)}{dx}, \tag{25}$$

where $\phi(x) = \Psi(x)/\sqrt{p_{eq}(x)}$.

The last two terms on the r.h.s. of Eq. (17) can be calculated from the definitions of $\langle \alpha_0|$ and $|\beta_{N+1}\rangle$ using chain rule for functional derivatives. In the basis of $\boldsymbol{\mathcal{H}}$ states $\langle \alpha_0|$ and $|\beta_{N+1}\rangle$ are equal and correspond to the descritized version of the function $\sqrt{p_{eq}(x)} = \rho_{eq}(x)$. Introducing:

$$\mathscr{L}_{AB} = \int_{-1}^{1} dx (\beta_0(x) + \alpha_N(x)) \rho_{\mathrm{eq}}(x) dx, \tag{26}$$

we need to find $\delta \mathscr{L}_{AB}/\delta F(x)$, where the dependence of $\beta_0(x)$ and $\alpha_N(x)$ on $F(x)$ is ignored, since it was already taken into account in the first term of Eq. (17). Using the chain rule for functional derivatives [54], we obtain:

$$\frac{\delta \mathscr{L}_{\mathrm{AB}}}{\delta F(y)} = \int_{-1}^{1} ds \frac{\delta \mathscr{L}_{\mathrm{AB}}}{\delta p_{eq}[s]} \frac{\delta p_{eq}[s]}{\delta F(y)}. \tag{27}$$

Using Euler-Lagrange equation, we immediately calculate the first term:

$$\frac{\delta \mathscr{L}_{\mathrm{AB}}}{\delta p_{eq}[s]} = \frac{\beta_0(s) + \alpha_N(s)}{2\rho_{\mathrm{eq}}(s)}. \tag{28}$$

To calculate the second term, we need to express p_{eq} in terms of the force:

$$p_{\mathrm{eq}} = \frac{\exp(\int_{-1}^{x} F(x')dx')}{\int_{-1}^{1} \exp(\int_{-1}^{x} F(x')dx')dx} \equiv \frac{e^{G(x)}}{\int_{-1}^{1} e^{G(x')}dx'} = \frac{\int_{-1}^{1} e^{G(x')}\delta(x-x')dx'}{\int_{-1}^{1} e^{G(x')}dx'}. \tag{29}$$

Here $G(x) = \int_{-1}^{1} F(x')H(x-x')dx'$, and H is a Heaviside function. Using the chain rule again, we obtain:

$$\frac{\delta p_{eq}[s]}{\delta F(y)} = \int_{-1}^{1} ds' \frac{\delta p_{eq}[s]}{\delta G[s']} \frac{\delta G[s']}{\delta F(y)}. \tag{30}$$

Equating these expressions with the Euler-Lagrange rule, we get:

$$\begin{aligned} \frac{\delta p_{eq}[s]}{\delta G[s']} &= \frac{e^{G(s')}\delta(s-s')\int_{-1}^{1} e^{G(x')}dx' - e^{G(s')}e^{G(s)}}{\left(\int_{-1}^{1} e^{G(x')}dx'\right)^2}, \\ \frac{\delta G[s']}{\delta F(y)} &= H(s'-y). \end{aligned} \tag{31}$$

As a result,

$$\begin{aligned} \frac{\delta p_{eq}[s]}{\delta F(y)} &= \int_{-1}^{1} ds' \left[\frac{e^{G(s')}\delta(s-s')}{\int_{-1}^{1} e^{G(x')}dx'} - \frac{e^{G(s')}e^{G(s)}}{\left(\int_{-1}^{1} e^{G(x')}dx'\right)^2} \right] H(s'-y) = \\ &= \frac{e^{G(s)}}{\int_{-1}^{1} e^{G(x')}dx'} \left[H(s-y) - \frac{\int_{-1}^{1} e^{G(s')}H(s'-y)ds'}{\int_{-1}^{1} e^{G(x')}dx'} \right] = \\ &= p_{eq}(s) \left[H(s-y) - \frac{\int_{y}^{1} e^{G(s')ds'}}{\int_{-1}^{1} e^{G(x')}dx'} \right] = \\ &= p_{eq}(s) \left[H(s-y) - \int_{y}^{1} p_{eq}(s')ds' \right] = p_{eq}(s) \left[H(s-y) - 1 + \int_{-1}^{y} p_{eq}(s')ds' \right]. \end{aligned} \tag{32}$$

Substituting Eqs. (28),(32) into Eq. (27), we obtain:

$$\frac{\delta \mathscr{L}_{\text{AB}}}{\delta F(y)} = \int_{-1}^{1} ds \frac{\beta_0(s) + \alpha_N(s)}{2\rho_{\text{eq}}(s)} p_{eq}(s) \left[H(s-y) - 1 + \int_{-1}^{y} p_{eq}(s')ds' \right] =$$
$$= \frac{1}{2} \int_{-1}^{1} ds \left(\beta_0(s) + \alpha_N(s)\right) \rho_{\text{eq}}(s) \left[H(s-y) - 1 + \int_{-1}^{y} p_{eq}(s')ds' \right] =$$
$$= \frac{1}{2} \left[\int_{y}^{1} \left(\beta_0(s) + \alpha_N(s)\right) \rho_{\text{eq}}(s) ds + \left(\int_{-1}^{1} ds \left(\beta_0(s) + \alpha_N(s)\right) \rho_{\text{eq}}(s) \right) \left(-1 + \int_{-1}^{y} p_{eq}(s')ds' \right) \right] =$$
$$= \frac{1}{2} \left[2 - \int_{-1}^{y} \left(\beta_0(s) + \alpha_N(s)\right) \rho_{\text{eq}}(s) ds - 2 + 2 \int_{-1}^{y} p_{eq}(s) ds \right] =$$
$$= \int_{-1}^{y} \left(p_{eq}(s) - \frac{\beta_0(s) + \alpha_N(s)}{2} \rho_{\text{eq}}(s) \right) ds. \tag{33}$$

Here we used the property

$$\int_{-1}^{1} ds \beta_0(s) \rho_{\text{eq}}(s) = \int_{-1}^{1} ds \alpha_N(s) \rho_{\text{eq}}(s) = 1, \tag{34}$$

which follows from the probability normalization condition and scaling of α and β coefficients.

The final expression for the variational derivative reads:

$$\frac{\delta \mathscr{L}}{\delta F(x)} = \sum_{ij} G_{ij} \frac{D}{2} p_{eq}(x) \frac{d\left(\phi_i(x)\phi_j(x)\right)}{dx} + \int_{-1}^{x} \left(p_{eq}(s) - \frac{\beta_0(s) + \alpha_N(s)}{2} \rho_{\text{eq}}(s) \right) ds, \tag{35}$$

where we introduced:

$$G_{ij} = \sum_{\tau} \Gamma^{\tau}_{i,j} \alpha_i(\tau) \beta_j(\tau + 1). \tag{36}$$

First, the function G_{ij} is calculated during the backward pass Eq. (16), and after that Eq. (35) is evaluated. Calculation of an antiderivative in the SEM basis is described in Section 2.

1.4 Variational derivatives of regularized likelihood

In this section we derive gradients of the regularized cost function w.r.t. driving force $F(X)$. As a regularizer, we use the trajectory entropy functional [33]:

$$S[\Phi(x)] = -\int \left(\frac{d\Phi}{dx} \right)^2 e^{-\Phi} dx. \tag{37}$$

For optimization with the gradient descent, we need to calculate the variational derivative of the regularizer w.r.t. the driving force $\delta S / \delta F$. Since $F(x) = p'_{\text{eq}}(x)/p_{\text{eq}}(x)$, Eq. (37) can be written as:

$$S[\Phi(x)] = -\int_{-1}^{1} \frac{p'^2_{\text{eq}}(x)}{p_{\text{eq}}(x)} dx. \tag{38}$$

Using variational chain rule [54]:

$$\frac{\delta S}{\delta F(y)} = \int_{-1}^{1} \frac{\delta S}{\delta p_{\text{eq}}(s)} \frac{\delta p_{\text{eq}}(s)}{\delta F(y)} ds. \tag{39}$$

The first term is calculated from Eq. (38) using the Euler-Lagrange equation:

$$\frac{\delta S}{\delta p_{\mathrm{eq}}(s)} = -\frac{p'^2_{\mathrm{eq}}(s)}{p^2_{\mathrm{eq}}(s)} + 2\frac{p''_{\mathrm{eq}}(s)}{p_{\mathrm{eq}}(s)}. \tag{40}$$

The second term is already calculated, see Eq. (32). Substituting these results, we obtain:

$$\begin{aligned}\frac{\delta S}{\delta F(y)} &= -\int_{-1}^{1}\left(\frac{p'^2_{\mathrm{eq}}(s)}{p_{\mathrm{eq}}(s)} - 2p''_{\mathrm{eq}}(s)\right)\left(H(s-y) - 1 + \int_{-1}^{y} p_{\mathrm{eq}}(s')ds'\right)ds = \\ &= -\int_{-1}^{1}\left(\frac{p'^2_{\mathrm{eq}}(s)}{p_{\mathrm{eq}}(s)} - 2p''_{\mathrm{eq}}(s)\right)\left(-H(y-s) + \int_{-1}^{y} p_{\mathrm{eq}}(s')ds'\right)ds = \\ &-\int_{-1}^{y}\left(-\frac{p'^2_{\mathrm{eq}}(s)}{p_{\mathrm{eq}}(s)} + 2p''_{\mathrm{eq}}(s) + p_{\mathrm{eq}}(s)\left[\int_{-1}^{1}\left(\frac{p'^2_{\mathrm{eq}}(s')}{p_{\mathrm{eq}}(s')} - 2p''_{\mathrm{eq}}(s')\right)ds'\right]\right)ds.\end{aligned} \tag{41}$$

1.5 Extensions of the modelling framework

Our modelling framework can be generalized for the simultaneous inference of $\Phi(x)$, $f(x)$, and D, including multiple neurons with diverse firing rate functions $f_1(x), f_2(x), \ldots$ coupled through the same latent dynamics. All of these quantities can be updated using gradient-descent of the negative log-likelihood:

$$\begin{aligned}D_n &= D_{n-1} + \gamma_D \frac{\partial \log \mathscr{L}}{\partial D}, \\ f_{i,n}(x) &= f_{i,n-1}(x) + \gamma_f \frac{\delta \log \mathscr{L}}{\delta f_i}.\end{aligned} \tag{42}$$

The analytical expressions for the likelihood derivatives w.r.t. D and $f_i(x)$ can be calculated following the exact same steps as for the derivative w.r.t. force (see Section 1.3). An example of simultaneous inference of $\Phi(x)$ and D is shown in Supplementary Fig. 1. It is also possible to extend the framework to multiple latent dimensions. For example, in the case of two-dimensions the expressions for the likelihood and its derivatives are similar, while the dimensionality of matrices and vectors become $N^2 \times N^2$ and N^2 correspondingly (compared to $N \times N$ and N in the case of one dimension).

1.6 Simulation parameters

An overview of simulation parameters is provided in Table 1. Here we list specific parameter values for all simulations. For synthetic data, we used $D = 2$ for a single-well (Tables 2 and 3), $D = 10$ for a double-well (Figs. 3, 4, 5, Extended Data Fig. 4, Tables 2 and 3), and $D = 7$ for a triple-well (Extended Data Fig. 3, Tables 2 and 3) ground-truth potentials with intermediate noise level. For low and high noise levels, we used four times smaller and four times larger noise magnitude D, respectively, for each of the ground-truth potentials. We used $D = 0.5$ for a four-well (Extended Data Fig. 5) potential model. For neurophysiological data, we used $D = 10$ (Fig. 6, Extended Data Figs. 6, 7). For data from the balanced E-I network, we used $D = 10$ (Extended Data Fig. 8). The parameter f_0 in the linear firing function was $f_0 = 100$ Hz for all synthetic data, $f_0 = 200$ Hz for neurophysiological data (Fig. 6, Extended Data Figs. 6, 7), and $f_0 = 300$ Hz for data from the balanced E-I network (Extended Data Fig. 8). We used $D_{\mathrm{KL}}^{\mathrm{thresh}} = 0.01$ in all simulations (except in Extended Data Fig. 5d where we show the effect of changing $D_{\mathrm{KL}}^{\mathrm{thresh}}$). In Figs. 3, 4, 5, each data sample contained roughly 20,000 spikes, which was split in two halves for training and validation or for feature consistency analysis. The test set in Fig. 4b contained 2,000

spikes and was separate from all training and validation sets. For multi-fold cross-validation in Fig. 4d, we split each data sample in 80% training, 10% validation and 10% test sets. In the neurophysiological data (Fig. 6 and Extended Data Fig. 6), the size of data samples $\mathscr{D}_1$ and $\mathscr{D}_2$ was about 6,000 spikes for each channel.

Parameter	Value	Description
Optimization		
γ	$5 \cdot 10^{-2}$ — $5 \cdot 10^{-4}$	GD learning rate
$n_{\max}$	100,000	maximum number of iterations
η	0 — 0.1	regularization strength
$\Phi_0(x)$	$-\log\left(\cos^2 \pi x/2\right), -\log\left(1/2\right)$	initialization of potential
D	0.5 — 40	noise magnitude
$f(x)$	$f_0(x+1)$	firing rate function, Hz
Spectral Elements Method		
x_{begin}	-1	left boundary of the latent space
x_{end}	1	right boundary of the latent space
Nv	64	number of eigenfunctions of $\boldsymbol{\mathcal{H}}$
bnd_cnd	Neumann	boundary conditions
N_e	256	number of elements
N_p	8	number of grid points in each element
Synthetic data generation		
T	10 —1,000	trial duration, s
Δt	0.001 — 0.0001	time bin for latent trajectory generation, s
Hyperparametrs for model selection based on feature consistency		
$D_{\text{KL}}^{\text{thresh}}$	0.01, 0.02	KL threshold for feature consistency
R	1 — 20	feature complexity radius

Table 1. Simulation parameters

1.7 Good fit, overfitting and underfitting for different dynamics and data amount

We performed multiple simulations to test how the probabilities of a good fit, overfitting and underfitting depend on the data amount, the complexity of the ground-truth dynamics, and noise magnitude D. Among models produced by the gradient descent, the best fitting model is the model most similar to the ground truth. However, a model selection procedure is not guaranteed to identify the best fitting model. We define a good fit as an outcome when the selected model is close to the best fitting model. Underfitting or overfitting occur when the selected model contains, respectively, fewer or more features than a good fitting model. For practical purposes, we developed an algorithmic procedure to distinguish a good fit, underfitting and overfitting. Specifically, we compute symmetrized Kullback-Leibler (KL) divergence D_{KL} between the ground-truth model and all models Φ_n produced by the GD on iterations $n = 1, 2 \ldots n_{\max}$ (see Eq. (16) in the main text). The $D_{\text{KL}}(n)$ curve is typically U-shaped and exhibits a minimum at the iteration $n_{\min}$ where the model is closest to the ground truth, i.e. the best fitting model (Supplementary Fig. 4). A model selection procedure selects a model at iteration n^*. The selected model is defined a good fit if $D_{\text{KL}}(n^*)$ is within a distance $\bar{D}_{\text{KL}} = 0.02$ from the minimum $D_{\text{KL}}(n_{\min})$. If $D_{\text{KL}}(n^*)$

is outside this range, the selected model is defined underfitted or overfitted when n^* is an earlier or later iteration than $n_{\min}$, respectively:

$$\begin{cases} \text{accurate fit}, & \text{if } |D_{\mathrm{KL}}(n^*) - D_{\mathrm{KL}}(n_{\min})| < \bar{D}_{\mathrm{KL}}; \\ \text{overfitting}, & \text{if } |D_{\mathrm{KL}}(n^*) - D_{\mathrm{KL}}(n_{\min})| \geqslant \bar{D}_{\mathrm{KL}},\ n^* > n_{\min}; \\ \text{underfitting}, & \text{if } |D_{\mathrm{KL}}(n^*) - D_{\mathrm{KL}}(n_{\min})| \geqslant \bar{D}_{\mathrm{KL}},\ n^* < n_{\min}. \end{cases} \tag{43}$$

Examples of a good fit, underfitting and overfitting are shown in Supplementary Fig. 4.

We used the same sets of synthetic data to compare two model selection procedures: (i) selecting models with the best generalization at the minimum of the validated negative log-likelihood, and (ii) our model selection procedure based on feature consistency (see Methods in the main text). We used 27 simulation settings, which correspond to all combinations of three levels of the ground-truth model complexity (single, double and triple well potentials), three data amounts (with roughly 2,000, 20,000 and 200,000 spikes), and three levels of noise (low, intermediate and high). For each setting, we performed 10 simulations with 10 independent data samples generated from the same ground-truth model. Each data sample was split in two halves, which were used as training and validation sets for the generalization-based model selection. The same data split was used by our feature-consistency method comparing models discovered on two halves of the data. In each of the 270 independent simulations (27 settings $\times$ 10 realizations), we performed optimization and model selection and classified the outcomes according to Eq. (43).

The summary of results is presented in Table 2 for generalization-based model selection and in Table 3 for our feature-consistency method. For large data amounts (200,000 spikes), both model selection methods produce good fitting models with high probability. However, the two model selection methods behave differently for moderate (20,000 spikes) and small (2,000 spikes) data amounts, that are realistic for neurophysiological experiments. For moderate data amounts, our feature-consistency method reliably produces good fits of simpler dynamics (single and double well) and tends to underfit more complex dynamics (triple well). For small data amounts, underfitting becomes more frequent even for simpler dynamics. This behaviour is consistent with the Occam's principle: with insufficient data a model selection procedure is expected to underfit, i.e. favour simpler explanations of the data. In contrast, the generalization-based model selection apparently violates the Occam's principle and can still overfit for moderate and small data amounts. Moreover, the probabilities of a good fit and overfitting do not depend on the complexity of the ground-truth dynamics (cf. double to triple well for 20,000 spikes in Table 2), the data amount (cf. 2,000 to 20,000 spikes for double well in Table 2), or noise magnitude, which indicates that these outcomes arise largely by chance. This counter-intuitive behaviour of the generalization-based model selection is due to the fact that the bias-variance trade-off does not always hold in the regime where overfitted models generalize well. Therefore, generalization-based model selection can overfit for low data amounts and does not conform to the expected dependence on the ground-truth complexity and data amount.

We used a relatively conservative value $D^{\mathrm{KL}}_{\mathrm{thres}} = 0.01$, so that the feature-consistency method rarely overfits (only 1 outcome is overfitted, while 80 outcomes are underfitted out of 270 simulations in Table 3). Increasing $D^{\mathrm{KL}}_{\mathrm{thres}}$ would reduce the number of underfittings and increase the number of good fits (Extended Data Fig. 5c,d) but also increase the number of overfittings for simple ground-truth dynamics in Table 3. We recommend using more conservative values of $D^{\mathrm{KL}}_{\mathrm{thres}}$ to produce a behaviour consistent with the Occam's principle.

Ground-truth model	Number of spikes in the data		
Low noise			
	$2,000$	$20,000$	$200,000$
Single well	2 / 5 / 3	8 / 2 / 0	9 / 1 / 0
Double well	6 / 4 / 0	6 / 4 / 0	9 / 1 / 0
Triple well	4 / 6 / 0	9 / 1 / 0	9 / 1 / 0
Intermediate noise			
	$2,000$	$20,000$	$200,000$
Single well	5 / 3 / 2	6 / 4 / 0	10 / 0 / 0
Double well	7 / 3 / 0	7 / 3 / 0	10 / 0 / 0
Triple well	1 / 3 / 6	7 / 2 / 1	10 / 0 / 0
High noise			
	$2,000$	$20,000$	$200,000$
Single well	3 / 5 / 2	7 / 2 / 0	10 / 0 / 0
Double well	7 / 3 / 0	7 / 3 / 0	9 / 1 / 0
Triple well	2 / 5 / 3	3 / 3 / 4	6 / 1 / 3

Table 2. Summary of outcomes for generalization-based model selection. For each of the 27 simulation settings (3 levels of the ground-truth model complexity $\times$ 3 data amounts $\times$ 3 levels of noise D), the numbers show the counts of good fits / overfitting / underfitting out of 10 simulations on independent data samples. For example, 2/5/3 in the first cell means 2 good fits, 5 overfitting and 3 underfitting outcomes out of 10 simulations. Simulations for the cells highlighted in grey are shown in Extended Data Figs. 3 and 4.

Ground-truth model	Number of spikes in the data		
Low noise			
	$2,000$	$20,000$	$200,000$
Single well	6 / 0 / 4	10 / 0 / 0	10 / 0 / 0
Double well	0 / 0 / 10	9 / 0 / 1	10 / 0 / 0
Triple well	0 / 0 / 10	6 / 0 / 4	10 / 0 / 0
Intermediate noise			
	$2,000$	$20,000$	$200,000$
Single well	5 / 1 / 4	9 / 0 / 1	10 / 0 / 0
Double well	4 / 0 / 6	10 / 0 / 0	10 / 0 / 0
Triple well	1 / 0 / 9	4 / 0 / 6	10 / 0 / 0
High noise			
	$2,000$	$20,000$	$200,000$
Single well	5 / 0 / 5	10 / 0 / 0	10 / 0 / 0
Double well	6 / 0 / 4	10 / 0 / 0	10 / 0 / 0
Triple well	2 / 0 / 8	2 / 0 / 8	10 / 0 / 0

Table 3. Summary of outcomes for model selection based on feature consistency. Same format as in Table 2.

2 Materials and methods: Additional details

2.1 Eigenvector-eigenvalue expansion of the Fokker-Planck operator

Consider Eq. (5) and let $\hat{\boldsymbol{\mathcal{H}}} = \hat{\boldsymbol{\mathcal{H}}}_{\mathbf{0}} + \hat{\boldsymbol{\mathcal{H}}}_{\boldsymbol{I}}$, where:

$$\hat{\boldsymbol{\mathcal{H}}}_{\mathbf{0}} = \left(D\frac{\partial}{\partial x}F(x) - D\frac{\partial^2}{\partial x^2} \right), \quad \hat{\boldsymbol{\mathcal{H}}}_{\boldsymbol{I}} = f(x). \tag{44}$$

Here $\hat{\boldsymbol{\mathcal{H}}}_{\mathbf{0}}$ is the original Fokker-Planck operator that propagates probability density $p(x,t)$ according to the Langevin dynamics, and the additional contribution $\hat{\boldsymbol{\mathcal{H}}}_{\boldsymbol{I}}$ accounts for the fact that no spikes were observed during a time period between any two adjacent spikes.

Operator $\hat{\boldsymbol{\mathcal{H}}}_{\mathbf{0}}$ can be written in terms of potential:

$$\hat{\boldsymbol{\mathcal{H}}}_{\mathbf{0}} = -\frac{\partial}{\partial x} D e^{-\Phi(x)} \frac{\partial}{\partial x} e^{\Phi(x)}. \tag{45}$$

To take advantage of Hermitian operators, we multiply Eq. (5) by $\exp(\Phi(x)/2)$:

$$e^{\Phi(x)/2}\frac{\partial p}{\partial t} = -\underbrace{e^{\Phi(x)/2}\hat{\boldsymbol{\mathcal{H}}}e^{-\Phi(x)/2}}_{\mathcal{H}}\underbrace{e^{\Phi(x)/2}p}_{\rho}. \tag{46}$$

Here we introduced a new Hermitian operator $\boldsymbol{\mathcal{H}} = \exp(\Phi(x)/2)\hat{\boldsymbol{\mathcal{H}}}\exp(-\Phi(x)/2)$. This operator acts on $\rho(x,t) = p(x,t)/\rho_{\mathrm{eq}}(x)$, where $\rho_{\mathrm{eq}}(x) = \exp(-\Phi(x)/2) = \sqrt{p_{\mathrm{eq}}(x)}$. Similar to the original operator, we decompose $\boldsymbol{\mathcal{H}} = \boldsymbol{\mathcal{H}}_{\mathbf{0}} + \boldsymbol{\mathcal{H}}_{\boldsymbol{I}}$, where

$$\boldsymbol{\mathcal{H}}_{\mathbf{0}} = e^{\Phi(x)/2}\hat{\boldsymbol{\mathcal{H}}}_{\mathbf{0}}e^{-\Phi(x)/2}, \qquad \hat{\boldsymbol{\mathcal{H}}}_{\boldsymbol{I}} = \boldsymbol{\mathcal{H}}_{\boldsymbol{I}}. \tag{47}$$

Following Ref. [21], we first solve the eigenvalue-eigenvector problem for the operator $\boldsymbol{\mathcal{H}}_{\mathbf{0}}$:

$$\boldsymbol{\mathcal{H}}_{\mathbf{0}}\Psi_0(x) = -e^{\Phi(x)/2}\frac{\partial}{\partial x} D e^{-\Phi(x)} \frac{\partial}{\partial x} e^{\Phi(x)/2}\Psi_0(x) = \lambda_0\Psi_0(x). \tag{48}$$

Multiplying both sides by $\exp(-\Phi(x)/2)$ from the left and letting $\phi_0(x) = \Psi_0(x)\exp(\Phi(x)/2) = \Psi_0(x)/\sqrt{p_{\mathrm{eq}}(x)}$, we obtain:

$$-\frac{\partial}{\partial x} D e^{-\Phi(x)} \frac{\partial}{\partial x}\phi_0(x) = \lambda_0 p_{\mathrm{eq}}(x)\phi_0(x). \tag{49}$$

This problem is discretized in the spectral elements basis (see Supplementary Note 2.2). Since Eq. (49) is of the second order, one has to impose two boundary conditions. Assuming reflecting boundaries for the original probability distribution $p(x,t)$ (zero probability flux), it can be shown [55] that our problem satisfies backward Fokker-Planck equation and obeys Neumann (zero-derivative) boundary conditions. Eigenvalues $\boldsymbol{\lambda}_{\mathbf{0}}$ and eigenvectors $\phi_{0,i}(x)$ of the discretized problem are obtained with Python solver *scipy.linalg.eigh*. This basis is truncated to the first 64 basis functions. After that the eigenvectors of $\boldsymbol{\mathcal{H}}_{\mathbf{0}}$ are obtained by scaling back $\Psi_0(x) = \phi_0(x)\rho_{\mathrm{eq}}(x)$. Next the eigenvalue problem for the full operator $\boldsymbol{\mathcal{H}}$ is solved in the Ψ_0 basis. The matrix form of this problem is $\boldsymbol{K}\hat{\Psi} = \lambda\hat{\Psi}$, where:

$$K_{ij} = \langle\Psi_{0,i}|\boldsymbol{\mathcal{H}}_{\mathbf{0}} + \boldsymbol{\mathcal{H}}_{\boldsymbol{I}}|\Psi_{0,j}\rangle = \lambda_{0,i}\delta_{ij} + \langle\Psi_{0,i}|f(x)|\Psi_{0,j}\rangle = \lambda_{0,i}\boldsymbol{I} + \boldsymbol{Q}_0^T\boldsymbol{W}f(x)\boldsymbol{Q}_0. \tag{50}$$

Eigenvectors of the operator $\boldsymbol{\mathcal{H}}$ in the original spectral elements basis are then obtained by the rules of basis transformation: $\boldsymbol{Q} = \boldsymbol{Q}_0\hat{\boldsymbol{Q}}$, where $\boldsymbol{Q}_0$ and $\hat{\boldsymbol{Q}}$ are the eigenvector matrices of $\boldsymbol{\mathcal{H}}_{\mathbf{0}}$ and $\boldsymbol{K}$, respectively.

2.2 Spectral elements method

As in Ref. [21], we use the Spectral Elements Method (SEM) for discretization of the eigenvalue problem Eq. (49). This method is numerically efficient due to sparse discretization and has exponential precision as other spectral methods. SEM is briefly outlined here, while the details can be found in Ref. [56].

Consider the following Sturm-Liouville problem for the function $u(x)$:

$$\begin{aligned} -\frac{d}{dx}\left(p(x)\frac{du(x)}{dx}\right) + q(x)u(x) = \lambda w(x)u(x), \qquad a < x < b, \\ A_1 u(a) + A_2 u'(a) = 0, \quad B_1 u(b) + B_2 u'(b) = 0. \end{aligned} \tag{51}$$

Using the weak formulation, we multiply both sides of Eq. (51) by a test function $\Xi_i(x)$ and integrate:

$$\int_a^b \left(-\frac{d}{dx}\left(p(x)\frac{du(x)}{dx}\right)\right)\Xi_i(x)dx + \int_a^b q(x)u(x)\Xi_i(x)dx = \int_a^b \lambda w(x)u(x)\Xi_i(x)dx. \tag{52}$$

The first term can be integrated by parts:

$$\left[-p(x)\frac{du(x)}{dx}\Xi_i(x)\right]_a^b + \int_a^b p(x)\frac{du(x)}{dx}\frac{d\Xi_i(x)}{dx}dx + \int_a^b q(x)u(x)\Xi_i(x)dx = \int_a^b \lambda w(x)u(x)\Xi_i(x)dx. \tag{53}$$

Next, we assume a finite number of collocation points x_i, $i = 1, 2, ..., N$ and use (yet unspecified) basis that consists of N basis functions $\Psi_i(x)$. Expanding $u(x)$ with this basis, we get:

$$u(x) = \sum_{i=1}^{N} u_i \Psi_i(x), \qquad \frac{du(x)}{dx} = \sum_{i=1}^{N} u_i \frac{d\Psi_i(x)}{dx}. \tag{54}$$

By using the basis function instead of test functions and substituting the above expansions into Eq. (53), we get:

$$\begin{aligned} \left[-p(x)\frac{du(x)}{dx}\Psi_i(x)\right]_a^b + \sum_j \int_a^b p(x)u_j\frac{d\Psi_j(x)}{dx}\frac{d\Psi_i(x)}{dx}dx + \sum_j \int_a^b q(x)u_j\Psi_j(x)\Psi_i(x)dx = \\ = \sum_j \int_a^b \lambda w(x)u_j\Psi_j(x)\Psi_i(x)dx. \end{aligned} \tag{55}$$

Using our collocation points and replacing integrals with sums (and weights ρ_k), we obtain:

$$\begin{aligned} \left[-p(x)\frac{du(x)}{dx}\Psi_i(x)\right]_a^b + \sum_j \sum_k \rho_k p(x_k)u_j\frac{d\Psi_j(x_k)}{dx}\frac{d\Psi_i(x_k)}{dx} + \sum_j \sum_k \rho_k q(x_k)u_j\Psi_j(x_k)\Psi_i(x_k) = \\ = \lambda \sum_j \sum_k \rho_k w(x_k)u_j\Psi_j(x_k)\Psi_i(x_k). \end{aligned} \tag{56}$$

At this point we need to specify the basis functions and collocation points. We use Gauss-Legendre-Lobatto (GLL) grid and Lagrange interpolation polynomials as basis functions [56]. The GLL grid specifies the collocation points x_k and integration weights ρ_k:

$$\begin{aligned} x_k: \quad & x_1 = -1, \quad x_N = 1, \text{ and all zeros of } L'_{N-1}(x), \\ \rho_k = & \frac{2}{N(N-1)}\frac{1}{[L_{N-1}(x_k)]^2}. \end{aligned} \tag{57}$$

Lagrange interpolation polynomials are defined as:

$$\Psi_j(x) = \prod_{\substack{1<m<N \\ m\neq j}} \frac{x - x_m}{x_j - x_m}. \tag{58}$$

They have a useful property $\Psi_j(x_i) = \delta_{ij}$, hence the expansion coefficients in Eq. (54) are trivial: $u_i = u(x_i)$. By introducing a differentiation matrix $D_{ij} = d\Psi_j(x_i)/dx$, Eq. (56) can be written as:

$$\begin{aligned}\left[-p(x)\frac{du(x)}{dx}\Psi_i(x)\right]_a^b + \sum_j \left(\sum_k \rho_k p(x_k) D_{kj} D_{ki}\right) u(x_j) + \sum_j \left(\rho_i q(x_i)\delta_{ij}\right) u(x_j) = \\ = \lambda \sum_j \left(\rho_i w(x_i)\delta_{ij}\right) u(x_j).\end{aligned} \tag{59}$$

This equation can be written in the matrix-vector form:

$$\left[-p(x)\frac{du(x)}{dx}\Psi_i(x)\right]_a^b + \mathbf{H}\tilde{\mathbf{u}} = \lambda \mathbf{M}\tilde{\mathbf{u}}. \tag{60}$$

Here the stiffness $\boldsymbol{H}$ and mass $\boldsymbol{M}$ matrices are defined as:

$$H_{ij} = \sum_k \left(D_{ik}^T \rho_k p(x_k) D_{kj}\right) + \rho_i q(x_i)\delta_{ij}, \qquad M_{ij} = \rho_i w(x_i)\delta_{ij}. \tag{61}$$

For the chosen grid and basis functions, the differentiation matrix is known [56]:

$$D_{ij} = \frac{d\Psi_j(x_i)}{dx} = \begin{cases} \frac{L_{N-1}(x_i)}{L_{N-1}(x_j)} \frac{1}{x_i - x_j}, & i \neq j, \\ -\frac{N(N-1)}{4}, & i = j = 1, \\ \frac{N(N-1)}{4}, & i = j = N, \\ 0, & \text{otherwise.} \end{cases} \tag{62}$$

By comparing Eq. (51) with our problem Eq. (49), we see that in Sturm-Liouville notations:

$$p(x) = De^{-\Phi(x)} = Dp_{\text{eq}}(x), \qquad q(x) = 0, \qquad w(x) = p_{\text{eq}}(x). \tag{63}$$

Our goal is to discretize Eq. (60) with Neumann boundary conditions. These conditions imply that the boundary term in Eq. (60) is zero. To solve Eq. (60), we split the x-domain into N_e equal elements, each of which contains N_p points, including the boundary points, such that each of the adjacent elements have only one point in common. Next, a single element is mapped into an interval $[-1; 1]$ using affine mapping. After that the local stiffness and mass matrices are calculated on a single element using Eq. (61). To construct the global $\boldsymbol{H}$ and $\boldsymbol{M}$ for the entire x-domain, we map the local matrices back to the x-domain and patch the pieces together (see Ref. [56] for details). The resulting matrices have block-diagonal structure, where the width of the block is equal to the number of collocation points on a single element, N_p. After that, the discretized eigenvector-eigenvalue problem was solved with Python solver *scipy.linalg.eigh*. All parameter values are provided in Table 1.

2.3 Antiderivatives using Lagrange polynomials

In this work we use the Lagrange-polynomial basis, $f(x) = \sum_i f(x_i)\Psi_i(x)$, where $\Psi_i(x)$ is given by Eq. (58). To calculate the antiderivative of any function $F(x) = \int_{-1}^{x} f(x)dx$, we use our Lagrange

expansion of $f(x)$ and integrate each of the Lagrange polynomials analytically. The integration matrix is thus:

$$B_{lk} = \int_{-1}^{x_l} \Psi_k(x')dx'.$$

With this notation, the antiderivative is calculated with a matrix-vector multiplication:

$$F(x_l) = \int_{-1}^{x_l} f(x')dx' = \sum_{k=1}^{N} f(x_k) \int_{-1}^{x_l} \Psi_k(x')dx' = (\boldsymbol{B}\boldsymbol{f})_l \,.$$

The integration matrix can be calculated from the definition of Lagrange polynomials Eq. (58):

$$\boldsymbol{B}_{lk} = \frac{1}{D_k} \int_{-1}^{x_l} \prod_{i \neq k} (x' - x_i)dx', \qquad D_k = \prod_{i \neq k} (x_k - x_i).$$

Expanding the parentheses under the integral, we obtain a polynomial of degree $(N-1)$, so that the result will be a polynomial of degree N. The coefficients near each of x^k will be the sum of all possible k-wise products. After some algebra the final result is given by:

$$\boldsymbol{B}_{lk} = \frac{1}{D_k} \sum_{n=1}^{N} C_{nk} x_l^n + C_{0k}, \qquad C_{nk} = \frac{(-1)^{\mathrm{mod(N-n,2)}}}{n} \mathrm{Com(N-n, \{X \setminus x_k\})}. \tag{64}$$

Here mod is the reminder after division, $\{X \setminus x_k\}$ is a set of all the grid points except for x_k, and $\mathrm{Com(n, \{X \setminus x_k\})}$ denotes the sum of all possible n-wise products of a given set, for example:

$$\begin{aligned}
&\mathrm{Com}(0, \{x_1, x_2, x_3, x_4\}) = 1 \\
&\mathrm{Com}(1, \{x_1, x_2, x_3, x_4\}) = x_1 + x_2 + x_3 + x_4 \\
&\mathrm{Com}(2, \{x_1, x_2, x_3, x_4\}) = x_1x_2 + x_1x_3 + x_1x_4 + x_2x_3 + x_2x_4 + x_3x_4 \\
&\mathrm{Com}(3, \{x_1, x_2, x_3, x_4\}) = x_1x_2x_3 + x_1x_2x_4 + x_1x_3x_4 + x_2x_3x_4 \\
&\mathrm{Com}(4, \{x_1, x_2, x_3, x_4\}) = x_1x_2x_3x_4.
\end{aligned} \tag{65}$$

Finally, C_{0k} is an arbitrary integration constant. We set $F(-1) = 0$, which can be implemented by setting the first row of integration matrix to zero.

To adapt this method for finite element grid, we calculated the integration matrix on a single element, scaled it back on a full domain and patched the global integration matrix by stacking the local ones on the main diagonal. To account for an additive property of integration (accumulation of the integration results when moving from the left to the right), we added a bias vector $\boldsymbol{u}$. This vector accumulates the result of integration from the previous elements (that are located to the left of the current element). The final result is:

$$\begin{aligned}
&\tilde{F} = \boldsymbol{B}\boldsymbol{f}, \\
&u = \{\underbrace{0, 0, ..., 0}_{N_p}, \underbrace{\tilde{F}_{N_p-1}, \tilde{F}_{N_p-1}, ...\tilde{F}_{N_p-1}}_{N_p-1}, \underbrace{\tilde{F}_{N_p-1} + \tilde{F}_{2(N_p-1)}, ..., \tilde{F}_{N_p-1} + \tilde{F}_{2(N_p-1)}}_{N_p-1}, ...\}, \\
&F = \boldsymbol{B}\boldsymbol{f} + \boldsymbol{u}.
\end{aligned} \tag{66}$$

Alternatively, the bias can be absorbed into the differentiation matrix, which would make the differentiation matrix lower triangular instead of sparse.

References

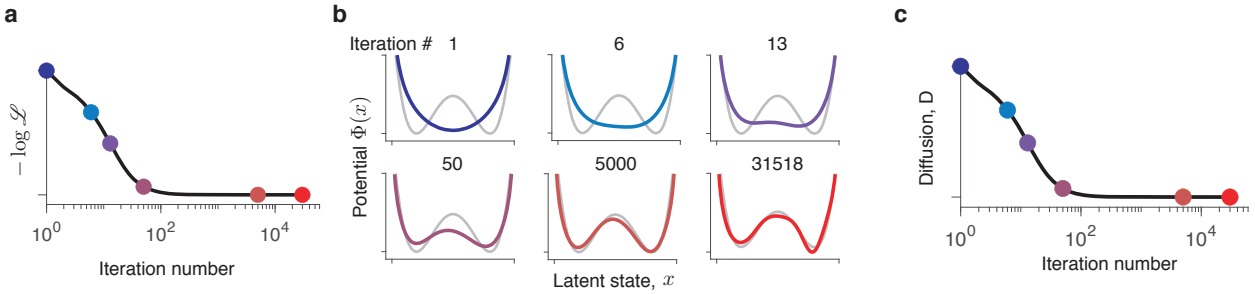
[53] R. M. Wilcox, *J Math Phys* **8**, 962 (1967).

[54] W. Greiner, J. Reinhardt, *Field Quantization* (Springer Science & Business Media, 2013).

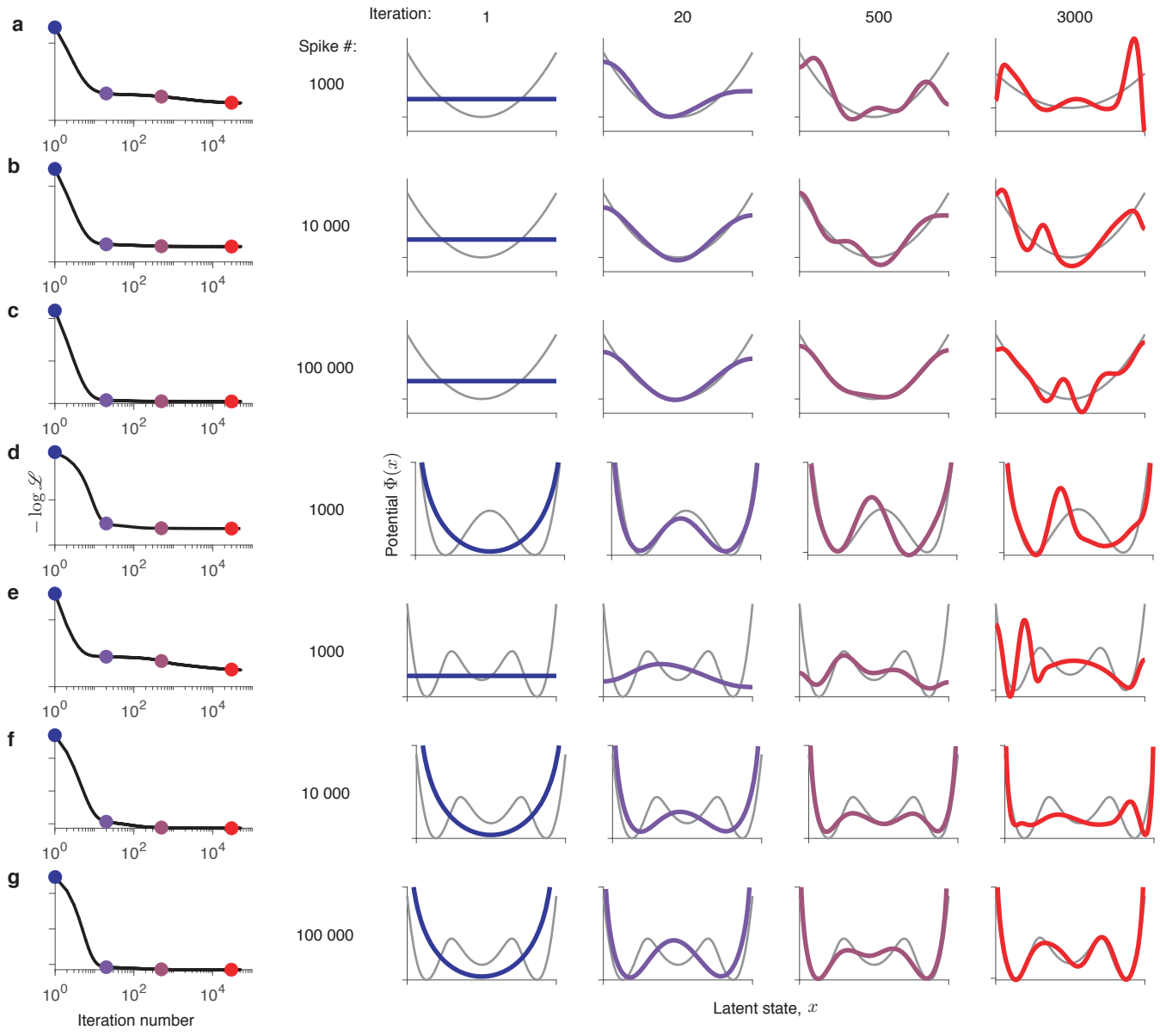
[55] C. W. Gardiner, *Handbook of Stochastic Methods*, vol. 3 (Springer Berlin, 1985).

[56] M. O. Deville, P. F. Fischer, E. H. Mund, *High-order Methods for Incompressible Fluid Flow*, vol. 9 (Cambridge University Press, 2002).

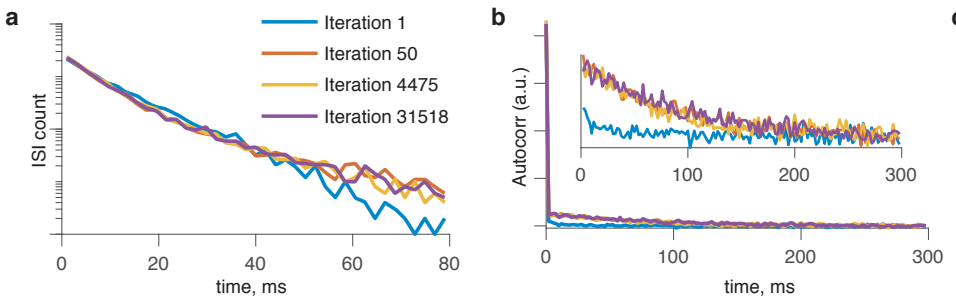
3 Supplementary figures



Supplementary Fig. 1. Simultaneous inference of potential $\Phi(x)$ and noise magnitude D. (**a**) Negative log-likelihood monotonically decreases during gradient-descent optimization. (**b**) Fitted potentials $\Phi_n(x)$ at selected iterations of the gradient-descent (colours correspond to dots in a) and the ground-truth potential (grey) from which the data was generated. (**c**) Noise magnitude D over iterations of the gradient-descent. The ground-truth value of D is shown with a dashed grey line. Coloured dots mark the same iterations as in a.

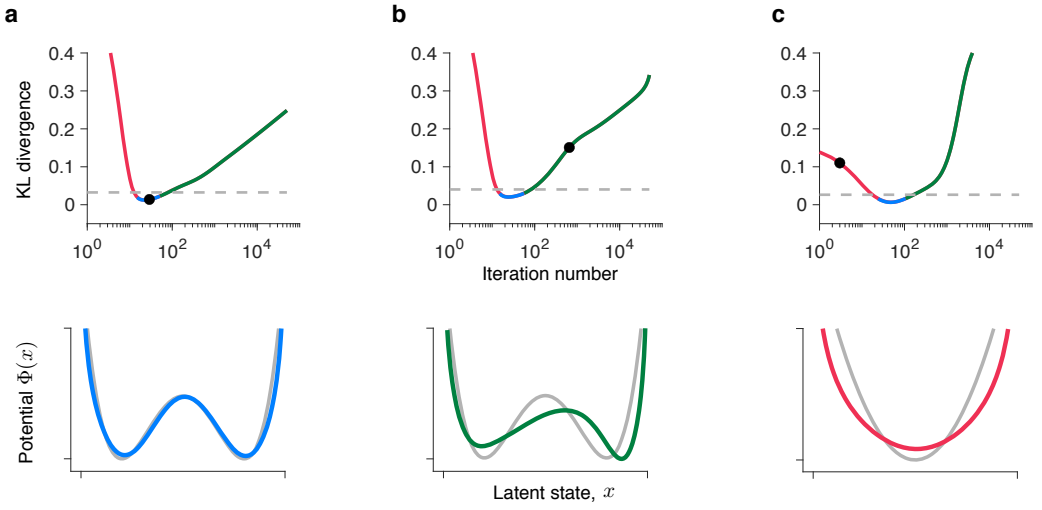


Supplementary Fig. 2. Overfitting is universally observed for unregularized optimization with different dynamics, data amount and hyperparameters. Negative log-likelihood $-\log \mathscr{L}$ over iterations of the gradient descent (left), and fitted potentials on selected iterations (right, colours correspond to the dots on the negative log-likelihood plot). The rows show simulations with different ground-truth dynamics, size of the data sample and initial condition for optimization $\Phi_0(x)$. (**a**) Single-well potential, 1,000 spikes, and $\Phi_0(x) = \text{const}$. (**b**) Single-well potential, 10,000 spikes, and $\Phi_0(x) = \text{const}$. (**c**) Single-well potential, 100,000 spikes, and $\Phi_0(x) = \text{const}$. (**d**) Double-well potential, 1,000 spikes, and $\Phi_0(x) = -\log(\cos^2(\pi x/2))$. (**e**) Triple-well potential, 1,000 spikes, and $\Phi_0(x) = \text{const}$. (**f**) Triple-well potential, 10,000 spikes, and $\Phi_0(x) = -\log(\cos^2(\pi x/2))$. (**g**) Triple-well potential, 100,000 spikes, and $\Phi_0(x) = -\log(\cos^2(\pi x/2))$.

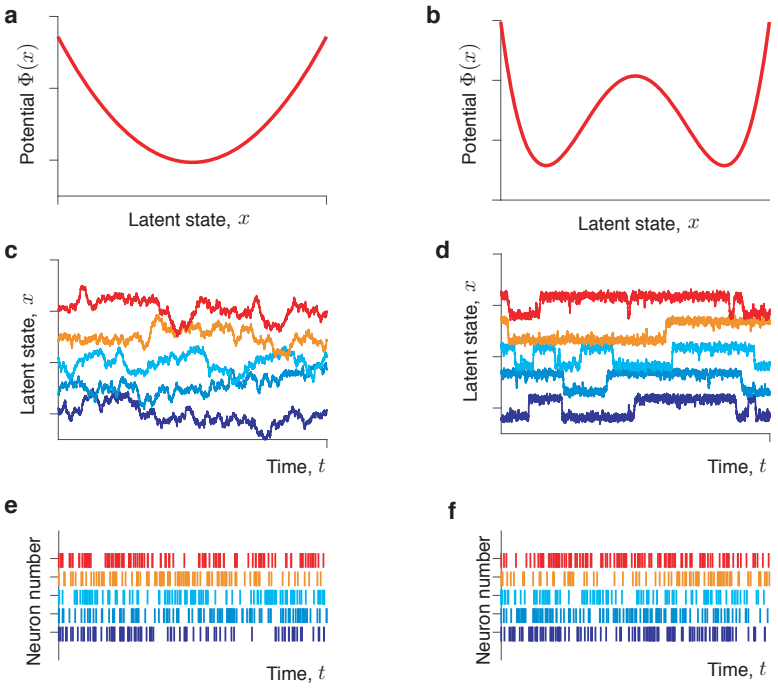


Iteration #	Mean, ms	Std, ms
0	9.7	10.2
50	10.2	14.0
4475	9.7	12.7
31518	10.0	13.4

Supplementary Fig. 3. Statistics of spikes in overfitted models. Statistics of spikes are computed for the simulation in Fig. 3b in the main text. Synthetic spike data was generated from the fitted potentials on four selected iterations (1, 50, 4475 and 31518). Interspike interval (ISI) distributions and autocorrelation functions were computed for these data. (**a**) ISIs distributions. (**b**) Autocorrelation functions (inset: zoom on vertical axis). The spike statistics are very similar for the fitted potentials obtained on iterations 50, 4475 and 31518, despite these potentials exhibit different features (see Fig. 3b).



Supplementary Fig. 4. Classification of model selection outcomes. KL divergence (upper panels) between the ground-truth model and models $\Phi_n(x)$ produced over iterations of the gradient descent. Three example simulations are shown illustrating a good fit (**a**), overfitting (**b**), and underfitting (**c**) as reported in Tables 2 and 3. The KL-divergence curves are U-shaped, where the minimum corresponds to the best fitting model (most similar to the ground truth). Along each curve, the iterations corresponding to underfitting (red), good fit (blue), and overfitting (green) are separated by the threshold $\bar{D}_{\mathrm{KL}}$ (dashed grey line), see Eq. (43). Iterations corresponding to the minimum of the validated negative log-likelihood (as in Table 2) are marked with black dots. Potentials selected at these iterations (lower panels, coloured lines) are shown along with the ground-truth potentials (grey). Underfitted models miss some of the true model features. Overfitted models exhibit spurious features. According to the classical bias-variance trade-off, overfitted models are expected to generalize poorly. However, our results and other recent work [20] show that overfitted high-capacity models can generalize well.



Supplementary Fig. 5. Spike trains generated from models with a double-well and single-well potentials appear similar. (**a**) Single-well potential. (**b**) Double-well potential. The potentials are chosen so that the second order spike-statistics are similar between the double-well and the single-well potential in a. (**c**) Five independent realizations of the latent trajectories generated from a single-well potential in a. (**d**) Five independent realizations of the latent trajectories generated from a double-well potential in b. (**e**) Five spike trains generated from the latent trajectories in c. (**f**) Five spike trains generated from the latent trajectories in d.