Received: 3 December 2019 | Accepted: 12 January 2022

DOI: 10.1111/rssb.12497

ORIGINAL ARTICLE

Manifold Markov chain Monte Carlo methods for Bayesian inference in diffusion models

Matthew M. Graham[1] | **Alexandre H. Thiery**[2] | **Alexandros Beskos**[1]

[1]University College London, London, UK

[2]National University of Singapore, Singapore

Correspondence
Matthew M. Graham, University College London, London, UK.
Email: m.graham@ucl.ac.uk

Abstract

Bayesian inference for nonlinear diffusions, observed at discrete times, is a challenging task that has prompted the development of a number of algorithms, mainly within the computational statistics community. We propose a new direction, and accompanying methodology—borrowing ideas from statistical physics and computational chemistry—for inferring the posterior distribution of latent diffusion paths and model parameters, given observations of the process. Joint configurations of the underlying process noise and of parameters, mapping onto diffusion paths consistent with observations, form an implicitly defined manifold. Then, by making use of a constrained Hamiltonian Monte Carlo algorithm on the embedded manifold, we are able to perform computationally efficient inference for a class of discretely observed diffusion models. Critically, in contrast with other approaches proposed in the literature, our methodology is *highly automated*, requiring minimal user intervention and applying *alike* in a range of settings, including: elliptic or hypo-elliptic systems; observations with or without noise; linear or non-linear observation operators. Exploiting Markovianity, we propose a variant of the method with complexity that scales linearly in the resolution of path discretisation and the number of observation times. Python code reproducing the results is available at http://doi.org/10.5281/zenodo.5796148.

KEYWORDS

constrained dynamics, Hamiltonian Monte Carlo, partially observed diffusions, stochastic differential equations

1 | INTRODUCTION

A large number of stochastic dynamical systems are modelled via the use of diffusion processes, see for example Kloeden and Platen (1992) and Oksendal (2013) and the references therein. An enormous amount of research has been dedicated to both the theoretical foundations of such processes and—as with this work—their statistical calibration. Our work lies in the context of processes observed discretely in time, under a low frequency regime, so that approximations of typically analytically intractable transition densities are assumed to be inaccurate. In this setting, *data augmentation* approaches within a Bayesian framework have delivered the prevailing methodologies, see for example Sørensen (2009) and Papaspiliopoulos et al. (2013), as they provide various model-specific algorithms for treating a number of different specifications of the structure of the diffusion process and of the observation regime. The performance of the developed algorithms can be improved via a combination of model transforms, often motivated by the *Roberts–Stramer critique* (Roberts & Stramer, 2001)—that the posterior distribution of the diffusivity parameters given a time discretisation of the process degenerates at finer resolutions—and more efficient Markov chain Monte Carlo (MCMC) kernels.

The work herein provides a natural approach for Bayesian inference over diffusion processes. Observations are treated as *constraints* placed on latent paths and parameters. This gives rise to the viewpoint that the posterior can be expressed as the prior distribution restricted to a *manifold*. We apply existing MCMC methods for sampling from distributions supported on submanifolds based on the simulation of constrained Hamiltonian dynamics (see, e.g. Brubaker et al., 2012; Hartmann & Schütte, 2005; Lelièvre et al., 2019; Rousset et al., 2010) to efficiently explore this manifold-supported posterior distribution. This class of methods relies on symplectic integrators for constrained Hamiltonian systems (Andersen, 1983; Leimkuhler & Matthews, 2016; Leimkuhler & Skeel, 1994; Reich, 1996). Critically, we leverage the Markovian structure of the diffusion process and Gaussianity of the driving noise to design a scalable inferential procedure. The main contributions of the proposed methodology can be summarised as follows:

(i) We provide a new viewpoint and accompanying algorithmic methodology for calibrating stochastic differential equation (SDE) models. The posterior is expressed as a distribution supported on a manifold embedded in a non-centred parametrization of the latent path and parameter space. We then make use of a constrained Hamiltonian Monte Carlo (HMC) scheme to explore this manifold, jointly updating both the parameters and latent path.

(ii) Unlike other algorithms that are often limited to specific model families, our approach is highly automated and remains unchanged irrespective of the choice of diffusion and observation models, including: elliptic or hypo-elliptic systems; data observed with or without noise; linear or non-linear observation operators.

(iii) We propose a novel constrained integrator that exploits the Gaussianity of the prior distribution on the pathspace. This leads to an improved scaling in sampling efficiency as the resolution of the latent path discretisation is refined.

(iv) We propose a scheme to exploit the Markovian structure of SDE models to ensure that the computational cost of the integrator for the Hamiltonian dynamics scales linearly both with the resolution of path discretisation and the number of observation times. To the best of our knowledge, the developed approach for leveraging the Markovian structure of the model is new.

(v) Our method extends the family of SDEs for which statistical calibration is now attainable. Consider the class of $\mathbb{R}^{\mathtt{X}}$-valued SDEs directly observed (without noise) via a non-linear function $\boldsymbol{h} : \mathbb{R}^{\mathtt{X}} \to \mathbb{R}^{\mathtt{Y}}$ at a finite set of times, for $\mathtt{X} \geq \mathtt{Y} \geq 1$. In such a scenario, standard data augmentation schemes fail (for non-trivial choices of $\boldsymbol{h}$) as the posterior of the latent variables given the observations does not have a density with respect to the Lebesgue measure. In contrast, our method remains applicable and unchanged.

Remark 1 (Criteria). To clarify the position of the framework put forward in this work within the wide field of statistical calibration for SDEs, we list a number of criteria met by our algorithm:

(i) It carries out full Bayesian inference for the model at hand.

(ii) It respects the Roberts–Stramer critique: the mixing times remain stable as the resolution of the path discretisation is refined.

(iii) It is applicable in scenarios where data are observed with or without noise; it is stable in the setting of diminishing noise.

(iv) It is applicable in the case of both full and partial observations. For partial observations, it accommodates both linear and non-linear observation operators.

(v) It attains the above via a unified and, in principle, automated methodology.

We have chosen the applications in Section 7 to highlight these properties. To our knowledge, the proposed method is unique in satisfying all of criteria (i)–(v).

The rest of the paper is organised as follows. Section 2 presents a generic class of SDE models relevant to our work. Section 3 recasts the inferential problem as one of exploring a posterior distribution supported on a manifold. Section 4 describes the constrained HMC method for sampling such distributions on implicitly defined manifolds. Section 5 shows how the Markovian structure of SDEs model can be exploited to design a scalable implementation of the methodology. Section 6 discusses related works. Section 7 illustrates the approach on several numerical examples, with comments on algorithmic performance and comparisons to alternative MCMC methods. Section 8 concludes with a brief summary and directions for future research.

Notation. Sans-serif symbols are used to distinguish random variables from their realisations (respectively, x and x). The set of integers from $\mathtt{A} \in \mathbb{Z}$ to $\mathtt{B} \in \mathbb{Z}$ inclusive, $\mathtt{B} \geq \mathtt{A}$, is $\mathtt{A} : \mathtt{B}$. Floor and ceiling operations are denoted $\lfloor x \rfloor$ and $\lceil x \rceil$ respectively. A symbol subscripted by a set indicates an indexed tuple, for example $x_{\mathtt{A}:\mathtt{B}} = (x_{\mathtt{s}})_{\mathtt{s}\in\mathtt{A}:\mathtt{B}}$. The set of linear maps from a vector space $\mathcal{X}$ to a vector space $\mathcal{Y}$ is $\mathfrak{L}(\mathcal{X}, \mathcal{Y})$. For $\boldsymbol{f} : \mathbb{R}^{\mathtt{M}} \to \mathbb{R}^{\mathtt{N}}$, the Jacobian of $\boldsymbol{f}$ is $\partial\boldsymbol{f} : \mathbb{R}^{\mathtt{M}} \to \mathbb{R}^{\mathtt{N}\times\mathtt{M}}$ and for $f : \mathbb{R}^{\mathtt{M}} \to \mathbb{R}$, its gradient and Hessian are $\nabla f : \mathbb{R}^{\mathtt{M}} \to \mathbb{R}^{\mathtt{M}}$ and $\nabla^2 f : \mathbb{R}^{\mathtt{M}} \to \mathbb{R}^{\mathtt{M}\times\mathtt{M}}$. For a multiple argument function $\boldsymbol{g}$, the Jacobian with respect to the $\mathtt{i}$th argument is denoted $\partial_{\mathtt{i}}\boldsymbol{g}$ and $\partial\boldsymbol{g} = (\partial_1\boldsymbol{g}, \partial_2\boldsymbol{g}, \cdots)$. The concatenation of vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ is denoted $[\boldsymbol{x};\boldsymbol{y}]$ and the concatenation of a tuple of vectors $\boldsymbol{x}_{1:\mathtt{N}}$

is $[\boldsymbol{x}_{1:\mathtt{N}}] = [\boldsymbol{x}_1; \cdots ; \boldsymbol{x}_\mathtt{N}]$ with the operation acting recursively, for example $[\boldsymbol{x}_{1:\mathtt{N}};\boldsymbol{y}] = [[\boldsymbol{x}_{1:\mathtt{N}}];\boldsymbol{y}]$. The determinant of a square matrix M is $|M|$. The $\mathtt{N}\times\mathtt{N}$ identity matrix is $\mathbb{I}_\mathtt{N}$. The block diagonal matrix with $M_{1:\mathtt{N}}$ left-to-right along its diagonal is $\operatorname{diag} M_{1:\mathtt{N}}$. The $\mathtt{N}$-dimensional Lebesgue measure is $\lambda_\mathtt{N}$. The set of Borel probability measures on a space $\mathcal{X}$ is $\mathfrak{P}(\mathcal{X})$.

2 | DIFFUSION MODEL

We consider the task of inferring the parameters of Itô-type SDEs of the form

$$\mathrm{d}\mathsf{x}(\tau) = \boldsymbol{a}(\mathbf{x}(\tau),\mathbf{z})\mathrm{d}\tau + B(\mathbf{x}(\tau),\mathbf{z})\mathrm{d}\mathsf{b}(\tau) \tag{1}$$

defined on a time interval $\mathcal{T} \subseteq \mathbb{R}_{\geq 0}$, where $\mathbf{z}$ is a $\mathcal{Z} \subseteq \mathbb{R}^\mathtt{Z}$-valued vector of model parameters, $\mathbf{x}$ a $\mathcal{X} \equiv \mathbb{R}^\mathtt{X}$-valued random process, $\mathbf{b}$ a $\mathcal{B} \equiv \mathbb{R}^\mathtt{B}$-valued standard Wiener process, $\boldsymbol{a} : \mathcal{X} \times \mathcal{Z} \to \mathcal{X}$ the drift function and $B : \mathcal{X} \times \mathcal{Z} \to \mathfrak{L}(\mathcal{B},\mathcal{X})$ the diffusion coefficient function. This time-homogeneous SDE system can be characterised by a family of Markov kernels $\kappa_\tau : \mathcal{X} \times \mathcal{Z} \to \mathfrak{P}(\mathcal{X})$ with $\kappa_{\tau'-\tau}(\boldsymbol{x},\boldsymbol{z})(\mathrm{d}\boldsymbol{x})$ the probability of $\mathbf{x}(\tau') \in \mathrm{d}\boldsymbol{x}$ given $(\mathbf{x}(\tau) = \boldsymbol{x},\ \mathbf{z} = \boldsymbol{z})$, for $(\tau,\tau',\boldsymbol{x},\boldsymbol{z}) \in \mathcal{T} \times \mathcal{T} \times \mathcal{X} \times \mathcal{Z}$. The parameter $\mathbf{z}$ is assigned a prior distribution $\mu \in \mathfrak{P}(\mathcal{Z})$ and, given $\mathbf{z}$, the initial state $\mathbf{x}_0$ is given a prior $\nu : \mathcal{Z} \to \mathfrak{P}(\mathcal{X})$.

We assume the system is observed at $\mathtt{T}$ times with a fixed inter-observation interval $\Delta > 0$ and $\mathcal{T} = [0, \mathtt{T}\Delta]$. The $\mathcal{Y} \subseteq \mathbb{R}^\mathtt{Y}$-valued observed vectors $\mathbf{y}_{1:\mathtt{T}}$ are then defined for each $\mathtt{t} \in 1\!:\!\mathtt{T}$ as $\mathbf{y}_\mathtt{t} = \boldsymbol{h}(\mathbf{x}(\mathtt{t}\Delta),\mathbf{z},\mathbf{w}_\mathtt{t})$ with $\boldsymbol{h} : \mathcal{X} \times \mathcal{Z} \times \mathcal{W} \to \mathcal{Y}$ the *observation function*, and $\mathbf{w}_\mathtt{t} \sim \eta$ the *observation noise vector* at time index $\mathtt{t}$ with distribution $\eta \in \mathfrak{P}(\mathcal{W})$ and $\mathcal{W} \subseteq \mathbb{R}^\mathtt{W}$.

Remark 2 Two common special cases of our observation model are

(i) Noiseless observations: $\boldsymbol{h}(\boldsymbol{x},\boldsymbol{z},\boldsymbol{w}) := \hbar(\boldsymbol{x})$ with $\mathtt{Y} \leq \mathtt{X}$ and $\mathtt{W} = 0$,
(ii) Additive (Gaussian) noise: $\boldsymbol{h}(\boldsymbol{x},\boldsymbol{z},\boldsymbol{w}) := \hbar(\boldsymbol{x}) + L(\boldsymbol{z})\boldsymbol{w}$ (and $\eta = \mathcal{N}(\boldsymbol{0},\mathbb{I}_\mathtt{Y})$).

In the former case the observation noise vectors $\mathbf{w}_{1:\mathtt{T}}$ can be omitted from the model. Our methodology readily extends to irregular observation times and time-varying model specification—for the SDE and the observation parts—however, for brevity of exposition, we only describe the equispaced and time-independent case here.

In general, it is neither possible to exactly sample from the Markov kernels κ_τ nor evaluate their densities with respect to the Lebesgue measure on $\mathcal{X}$. We thus adopt a data-augmentation approach (Elerian et al., 2001; Roberts & Stramer, 2001) and consider a discrete-time model formed by numerically integrating the original SDE; although this will introduce discretisation error, the error can be controlled by using a fine time resolution. We split each interobservation interval into $\mathtt{S}$ smaller time steps $\delta = \frac{\Delta}{\mathtt{S}}$. Given a time discretisation, a variety of numerical schemes for integrating SDE systems are available with varying levels of complexity and convergence properties (Kloeden & Platen, 1992). The schemes of interest in this article can be expressed as a forward operator $\boldsymbol{f}_\delta : \mathcal{Z} \times \mathcal{X} \times \mathbb{R}^\mathtt{V} \to \mathcal{X}$ defined such that, given parameters $\boldsymbol{z} \in \mathcal{Z}$, a current state $\boldsymbol{x} \in \mathcal{X}$ and a random vector $\mathbf{v} \sim \mathcal{N}(\boldsymbol{0},\mathbb{I}_\mathtt{V})$, $\boldsymbol{f}_\delta(\boldsymbol{z},\boldsymbol{x},\mathbf{v})$ is approximately distributed according to $\kappa_\delta(\boldsymbol{x},\boldsymbol{z})$ for small time steps $\delta > 0$. The simplest and most commonly used scheme is the Euler–Maruyama method, where $\mathtt{V} = \mathtt{B}$ and $\boldsymbol{f}_\delta(\boldsymbol{z},\boldsymbol{x},\boldsymbol{v}) = \boldsymbol{x} + \delta\boldsymbol{a}(\boldsymbol{x},\boldsymbol{z}) + \delta^{\frac{1}{2}}B(\boldsymbol{x},\boldsymbol{z})\boldsymbol{v}$. Importantly, the methodology developed in this article straightforwardly accommodates higher order methods, such as the Milstein scheme (Mil'shtejn, 1975).

For a particular choice of numerical scheme, given the parameters $\mathbf{z} \sim \mu$ and initial position $\mathbf{x}_0 \sim \nu(\mathbf{z})$, the states at all subsequent time steps $\mathbf{x}_{1:\mathtt{ST}}$ are iteratively generated via the forward operator $\boldsymbol{f}_\delta$ with $\mathbf{x}_\mathtt{s}$ denoting the discrete time approximation to the continuous time state $\mathbf{x}(\mathtt{s}\delta)$. The observations $\mathbf{y}_{1:\mathtt{T}}$ are computed from the discrete time state sequence $\mathbf{x}_{1:\mathtt{ST}}$ via the observation function $\boldsymbol{h}$ and observation noise vectors $\mathbf{w}_{1:\mathtt{T}}$. The overall generative model is summarised in Model 1.

Model 1 Time-discretised diffusion generative model.

```
function $g_{\mathbf{x}_:,\mathbf{y}_:}(\boldsymbol{z}, \boldsymbol{x}_0, \boldsymbol{v}_{1:\mathtt{St}}, \boldsymbol{w}_{1:\mathtt{t}})$        $\mathbf{z} \sim \mu$
    for $\mathtt{s} \in 1{:}\mathtt{St}$                                   $\mathbf{x}_0 \sim \nu(\mathbf{z})$
        $\boldsymbol{x}_\mathtt{s} = \boldsymbol{f}_\delta(\boldsymbol{z}, \boldsymbol{x}_{\mathtt{s}-1}, \boldsymbol{v}_s)$                     $\mathbf{v}_\mathtt{s} \sim \mathcal{N}(\mathbf{0}, \mathbb{I}_\mathtt{V}) \ \forall \mathtt{s} \in 1{:}\mathtt{ST}$
        if $\mathtt{s} \bmod \mathtt{S} \equiv 0$                          $\mathbf{w}_\mathtt{t} \sim \eta \ \forall \mathtt{t} \in 1{:}\mathtt{T}$
            $\boldsymbol{y}_{\mathtt{s}/\mathtt{S}} = \boldsymbol{h}(\boldsymbol{x}_\mathtt{s}, \boldsymbol{z}, \boldsymbol{w}_{\mathtt{s}/\mathtt{S}})$             $\mathbf{x}_{1:\mathtt{ST}},\ \mathbf{y}_{1:\mathtt{T}} = g_{\mathbf{x}_:,\mathbf{y}_:}(\mathbf{z}, \mathbf{x}_0, \mathbf{v}_{1:\mathtt{ST}}, \mathbf{w}_{1:\mathtt{T}})$
    return $\boldsymbol{x}_{1:\mathtt{St}},\ \boldsymbol{y}_{1:\mathtt{t}}$
```

3 | INFERENTIAL OBJECTIVE ON A MANIFOLD

We are interested in computing expectations with respect to the joint posterior of $\mathbf{z}$, $\mathbf{x}_0$, $\mathbf{x}_{1:\mathtt{ST}}$, given observations $\mathbf{y}_{1:\mathtt{T}} = \boldsymbol{y}_{1:\mathtt{T}}$. However, the states at nearby time steps will be highly dependent under the prior on $\mathbf{x}_{1:\mathtt{ST}}$ for small δ. Such strong dependencies are characteristic of *centred* parametrisations of hierarchical models, and have a deleterious effect on the performance of many approximate inference algorithms (Betancourt & Girolami, 2015; Papaspiliopoulos et al., 2003, 2007).

3.1 | Non-centred parametrisation

One can instead choose to parametrise the inference problem in terms of the latent vectors $\mathbf{v}_{1:\mathtt{ST}}$ used to numerically integrate the SDE. Given values for $\mathbf{z}$, $\mathbf{x}_0$ and $\mathbf{v}_{1:\mathtt{ST}}$, the state sequence $\mathbf{x}_{1:\mathtt{ST}}$ can be deterministically computed. Such a reparametrisation has the property that, under the prior, all components of the latent vectors $\mathbf{v}_{1:\mathtt{ST}}$ are independent standard normal variables. We further assume the following.

Assumption 1 There exist functions $\mathbf{g}_\mathbf{z} : \mathbb{R}^\mathtt{U} \to \mathcal{Z}$ and $\mathbf{g}_{\mathbf{x}_0} : \mathcal{Z} \times \mathbb{R}^{\mathtt{V}_0} \to \mathcal{X}$ and corresponding distributions $\tilde{\mu} \in \mathfrak{P}(\mathbb{R}^\mathtt{U})$, $\tilde{\nu} \in \mathfrak{P}(\mathbb{R}^{\mathtt{V}_0})$ with strictly positive smooth density functions with respect to the Lebesgue measures $\lambda_\mathtt{U}$ and $\lambda_{\mathtt{V}_0}$, respectively, such that $\mathbf{g}_\mathbf{z}(\mathbf{u}) \sim \mu$ and $\mathbf{g}_{\mathbf{x}_0}(\boldsymbol{z}, \mathbf{v}_0) \sim \nu(\boldsymbol{z}) \ \forall \boldsymbol{z} \in \mathcal{Z}$ if $\mathbf{u} \sim \tilde{\mu}$ and $\mathbf{v}_0 \sim \tilde{\nu}$.

Under such parametrisation in terms of $\mathbf{q} := [\mathbf{u}; \mathbf{v}_0; \mathbf{v}_{1:\mathtt{ST}}; \mathbf{w}_{1:\mathtt{T}}]$ all of $(\mathbf{u}, \mathbf{v}_0, \mathbf{v}_{1:\mathtt{ST}}, \mathbf{w}_{1:\mathtt{T}})$ are then a-priori independent and the resulting prior distribution $\rho \in \mathfrak{P}(\mathbb{R}^\mathtt{Q})$ with $\mathtt{Q} = \mathtt{U} + \mathtt{V}_0 + \mathtt{STV} + \mathtt{TW}$, has a density with respect to the Lebesgue measure $\lambda_\mathtt{Q}$,

$$\frac{\mathrm{d}\rho}{\mathrm{d}\lambda_\mathtt{Q}}([\boldsymbol{u}; \boldsymbol{v}_0; \boldsymbol{v}_{1:\mathtt{ST}}; \boldsymbol{w}_{1:\mathtt{T}}]) \propto \frac{\mathrm{d}\tilde{\mu}}{\mathrm{d}\lambda_\mathtt{U}}(\boldsymbol{u}) \frac{\mathrm{d}\tilde{\nu}}{\mathrm{d}\lambda_{\mathtt{V}_0}}(\boldsymbol{v}_0) \prod_{\mathtt{s}=1}^{\mathtt{ST}} \exp\left(-\frac{1}{2}\boldsymbol{v}_\mathtt{s}^\mathsf{T}\boldsymbol{v}_\mathtt{s}\right) \prod_{\mathtt{t}=1}^{\mathtt{T}} \frac{\mathrm{d}\eta}{\mathrm{d}\lambda_\mathtt{W}}(\boldsymbol{w}_\mathtt{t}). \tag{2}$$

Model 2 gives the generative model under this *non-centred* parametrisation and defines a function $\mathbf{g}_{\mathbf{y}_:}$ which generates observations given values for the latent variables. The observations can be thought of as imposing a series of constraints on the possible values of the latent variables $\mathbf{q}$; under additional assumptions on the regularity of the mapping $\boldsymbol{g}_{\mathsf{y}_:}$ from latent variables to observations, the set of $\mathbf{q}$ values satisfying the constraints will form a differentiable manifold embedded in $\mathbb{R}^{\mathtt{Q}}$. The posterior distribution on $\mathbf{q}$ given $\mathbf{y}_{1:\mathtt{T}} = \boldsymbol{y}_{1:\mathtt{T}}$ will not have a density with respect to the Lebesgue measure $\lambda_{\mathtt{Q}}$ as the manifold it has support on is a $\lambda_{\mathtt{Q}}$-null set. In the following section, we show, however, that by using a different reference measure we can compute a tractable density function for the posterior.

Model 2 Non-centred parametrisation of generative model.

```
function $g_{\mathsf{y}_:}(u, v_0, v_{1:\mathtt{St}}, w_{1:\mathtt{t}})$                 $\mathsf{u} \sim \tilde{\mu}$
    $z = g_{\mathsf{z}}(u)$                                          $\mathsf{v}_0 \sim \tilde{\nu}$
    $x_0 = g_{\mathsf{x}_0}(z, v_0)$                                 $\mathsf{v}_{\mathtt{s}} \sim \mathcal{N}(\mathbf{0}, \mathbb{I}_{\mathtt{V}})\ \forall \mathtt{s} \in 1{:}\mathtt{ST}$
    $x_{1:\mathtt{St}},\ y_{1:\mathtt{t}} = g_{\mathsf{x}_:,\mathsf{y}_:}(z, x_0, v_{1:\mathtt{St}}, w_{1:\mathtt{t}})$    $\mathsf{w}_{\mathtt{t}} \sim \eta\ \forall \mathtt{t} \in 1{:}\mathtt{T}$
    return $y_{1:\mathtt{t}}$                                        $\mathsf{y}_{1:\mathtt{T}} = g_{\mathsf{y}_:}(\mathsf{u}, \mathsf{v}_0, \mathsf{v}_{1:\mathtt{ST}}, \mathsf{w}_{1:\mathtt{T}})$
```

3.2 | Target posterior on manifold

We define a *constraint function* $\boldsymbol{c} : \mathbb{R}^{\mathtt{Q}} \to \mathbb{R}^{\mathtt{C}}$ with $\mathtt{C} = \mathtt{TY} < \mathtt{Q}$ as

$$\boldsymbol{c}([\boldsymbol{u}; \boldsymbol{v}_0; \boldsymbol{v}_{1:\mathtt{ST}}; \boldsymbol{w}_{1:\mathtt{T}}]) := [\boldsymbol{g}_{\mathsf{y}_:}(\boldsymbol{u}, \boldsymbol{v}_0, \boldsymbol{v}_{1:\mathtt{ST}}, \boldsymbol{w}_{1:\mathtt{T}})] - [\boldsymbol{y}_{1:\mathtt{T}}], \tag{3}$$

with the set of values on the manifold $\mathcal{M} := \{\boldsymbol{q} \in \mathbb{R}^{\mathtt{Q}} : \boldsymbol{c}(\boldsymbol{q}) = \mathbf{0}\}$ corresponding to all inputs of $\boldsymbol{g}_{\mathsf{y}_:}$ consistent with the observations. We make the following assumption.

Assumption 2 The constraint function $\boldsymbol{c}$ is continuously differentiable and has Jacobian $\partial\boldsymbol{c}$ which is full row-rank ρ-almost surely.

The differentiability requirement will be met if $\boldsymbol{f}_\delta$, $\boldsymbol{g}_{\mathsf{z}}$, $\boldsymbol{g}_{\mathsf{x}_0}$ and $\boldsymbol{h}$ are all themselves continuously differentiable with respect to each of their arguments. The rank condition on the Jacobian requires that the observed variables do not give redundant information about the latent variables, that is no observed variable can be expressed as a deterministic function of a subset of the other observed variables. In the case of observations subject to Gaussian additive noise this will always be satisfied if the noise covariance is full-rank. In the noiseless observation case, the condition will be met if no component of the state at an observation time $\mathbf{x}_{\mathtt{St}}$ is fully determined by the state at the previous observation time $\mathbf{x}_{\mathtt{S(t-1)}}$ and parameters $\mathbf{z}$, and the function $\boldsymbol{h} : \mathcal{X} \to \mathcal{Y}$ has Jacobian with full row-rank everywhere.

Under these assumptions $\mathcal{M}$ will be a $\mathtt{D} = \mathtt{Q} - \mathtt{C}$ dimensional differentiable manifold embedded into the $\mathtt{Q}$ dimensional ambient space. The posterior distribution $\pi \in \mathfrak{P}(\mathbb{R}^{\mathtt{Q}})$ on $\mathbf{q}$ given $\boldsymbol{c}(\mathbf{q}) = \boldsymbol{0}$ (and so $\mathbf{y}_{1:\mathtt{T}} = \boldsymbol{y}_{1:\mathtt{T}}$) is supported only on $\mathcal{M}$. Note that $\mathcal{M}$ has zero Lebesgue measure, so π does not have a density with respect to $\lambda_{\mathtt{Q}}$. To define an appropriate reference measure we further assume the following.

Assumption 3 The ambient latent space $\mathbb{R}^{\mathtt{Q}}$ is equipped with a metric tensor with a fixed positive definite matrix representation M.

A possible reference measure is then the $\mathtt{D}$-dimensional Hausdorff measure $\eta^M_{\mathtt{D}}$ on the ambient space, which has the required property that π is absolutely continuous with respect to $\eta^M_{\mathtt{D}}$. For measurable subsets $\mathcal{A} \subseteq \mathcal{M}$, we have that $\sigma^M_{\mathcal{M}}(\mathcal{A}) = \eta^M_{\mathrm{D}}(\mathcal{A})$ where $\sigma^M_{\mathcal{M}}$ is the Riemannian measure on the manifold $\mathcal{M}$ with metric induced from the ambient metric (see Lemma S3.2 in Section S3 in the Supplementary Material). As later results will be more naturally stated in terms of the Riemannian measure, we will use $\sigma^M_{\mathcal{M}}$ as the reference measure here.

Proposition 1 *Under Assumptions 1–3, the posterior π has a density*

$$\frac{\mathrm{d}\pi}{\mathrm{d}\sigma^M_{\mathcal{M}}}(\boldsymbol{q}) \propto \frac{\mathrm{d}\rho}{\mathrm{d}\lambda_{\mathrm{Q}}}(\boldsymbol{q})|\partial\boldsymbol{c}(\boldsymbol{q})M^{-1}\partial\boldsymbol{c}(\boldsymbol{q})^{\mathsf{T}}|^{-\frac{1}{2}}. \tag{4}$$

A proof is given in Section S3 in the Supplementary Material. See also Rousset et al. (2010), Diaconis et al. (2013) and Graham and Storkey (2017). The negative log posterior density thus reads

$$\ell(\boldsymbol{q}) := -\log\frac{\mathrm{d}\rho}{\mathrm{d}\lambda_{\mathrm{Q}}}(\boldsymbol{q}) + \frac{1}{2}\log|G_M(\boldsymbol{q})|,$$

where the $\mathrm{C}\times\mathrm{C}$ matrix $G_M(\boldsymbol{q}) := \partial\boldsymbol{c}(\boldsymbol{q})M^{-1}\partial\boldsymbol{c}(\boldsymbol{q})^{\mathsf{T}}$ is termed the *Gram matrix*.

4 | MCMC ON IMPLICITLY DEFINED MANIFOLDS

In this section, we review MCMC methods for sampling from a distribution supported on an implicitly defined manifold. We stress at this point that we do not design a fundamentally new such MCMC method but instead rely on modifying and combining existing methodologies. In particular, we adopt a symplectic integrator for constrained Hamiltonian systems (Andersen, 1983; Leimkuhler & Matthews, 2016; Leimkuhler & Skeel, 1994) to simulate Hamiltonian dynamics trajectories on the manifold, and use this as a proposal generating mechanism within a Hamiltonian Monte Carlo (HMC) scheme (Betancourt, 2017; Duane et al., 1987; Neal, 2011). The use of constrained Hamiltonian dynamics within an HMC context has been previously proposed multiple times—see for example Hartmann and Schütte (2005), (Rousset et al. 2010 Chapter 3), Brubaker et al. (2012) and Lelièvre et al. (2019). We refer the interested reader to Arnol'd (2013) and Holm et al. (2009) for general background on constrained mechanics and to (Barp 2020 Chapter 3) for a comprehensive review of HMC on manifolds.

4.1 | Constrained Hamiltonian dynamics

To define the constrained Hamiltonian system, we first augment the latent vector $\mathbf{q}$, henceforth the *position*, with a *momentum* $\mathbf{p}$. Formally the momentum is a *co-vector*, that is a linear form in $\mathfrak{L}(\mathbb{R}^{\mathrm{Q}}, \mathbb{R})$ and the metric on the position space induces a *co-metric* on the momentum space with matrix representation M^{-1}. As a common abuse of notation, we will not distinguish between vectors and co-vectors and simply consider $\mathbf{p}$ as a vector in $\mathbb{R}^{\mathrm{Q}}$ equipped with a metric with matrix representation M^{-1}. The *Hamiltonian* function $h : \mathbb{R}^{\mathrm{Q}} \times \mathbb{R}^{\mathrm{Q}} \to \mathbb{R}$ is then defined as

$$h(\boldsymbol{q}, \boldsymbol{p}) := \ell(\boldsymbol{q}) + \frac{1}{2}\boldsymbol{p}^{\mathsf{T}}M^{-1}\boldsymbol{p}. \tag{5}$$

Thus far we have an unconstrained Hamiltonian system. To restrict $\mathbf{q}$ to $\mathcal{M}$, we introduce a Lagrange multiplier function $\lambda : \mathbb{R}^{\mathrm{Q}} \times \mathbb{R}^{\mathrm{Q}} \to \mathbb{R}^{\mathrm{C}}$ implicitly defined so that constraint $\boldsymbol{c}(\boldsymbol{q}) = \boldsymbol{0}$ is enforced, at all times, in the following dynamics. The constrained Hamiltonian dynamics associated with the Hamiltonian in Equation (5) are then described by the system of differential algebraic equations (DAE)

$$\frac{\mathrm{d}\boldsymbol{q}}{\mathrm{d}t} = M^{-1}\boldsymbol{p}, \quad \frac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}t} = -\nabla\ell(\boldsymbol{q}) - \partial\boldsymbol{c}(\boldsymbol{q})^{\mathsf{T}}\lambda(\boldsymbol{q},\boldsymbol{p}), \quad \boldsymbol{c}(\boldsymbol{q}) = \boldsymbol{0}. \tag{6}$$

The condition that the *primary constraints*, $\boldsymbol{c}(\boldsymbol{q}) = \boldsymbol{0}$, are preserved in time implies a set of *secondary constraints* of the form $\partial\boldsymbol{c}(\boldsymbol{q})M^{-1}\boldsymbol{p} = \boldsymbol{0}$.

Definition 1 The set of momenta satisfying the secondary constraints at a position $\boldsymbol{q}$ coincides with the *co-tangent space* of the manifold $\mathcal{M}$ at $\boldsymbol{q}$, denoted

$$\mathsf{T}^*_{\boldsymbol{q}}\mathcal{M} := \left\{\boldsymbol{p} \in \mathbb{R}^Q : \partial\boldsymbol{c}(\boldsymbol{q})M^{-1}\boldsymbol{p} = \boldsymbol{0}\right\}.$$

Definition 2 The set of positions and momenta in the manifold and corresponding co-tangent spaces, respectively, are termed the *co-tangent bundle*, denoted

$$\mathsf{T}^*\mathcal{M} := \left\{\boldsymbol{q} \in \mathcal{M}, \boldsymbol{p} \in \mathsf{T}^*_{\boldsymbol{q}}\mathcal{M}\right\} = \left\{\boldsymbol{q} \in \mathbb{R}^Q, \boldsymbol{p} \in \mathbb{R}^Q : \boldsymbol{c}(\boldsymbol{q}) = \boldsymbol{0}, \partial\boldsymbol{c}(\boldsymbol{q})M^{-1}\boldsymbol{p} = \boldsymbol{0}\right\}.$$

Remark 3 $\mathsf{T}^*\mathcal{M}$ is a *symplectic manifold* with a *symplectic form* given by the restriction of the symplectic form on $\mathbb{R}^Q \times \mathbb{R}^Q$ to $\mathsf{T}^*\mathcal{M}$, which under Assumptions 2 and 3 is almost surely non-degenerate.

Definition 3 The symplectic form on $\mathsf{T}^*\mathcal{M}$ induces a volume form and corresponding *Liouville measure* denoted $\sigma_{\mathsf{T}^*\mathcal{M}}$, which can be decomposed as

$$\sigma_{\mathsf{T}^*\mathcal{M}}(\mathrm{d}\boldsymbol{q},\mathrm{d}\boldsymbol{p}) = \sigma^M_{\mathcal{M}}(\mathrm{d}\boldsymbol{q})\,\sigma^{M^{-1}}_{\mathsf{T}^*_{\boldsymbol{q}}\mathcal{M}}(\mathrm{d}\boldsymbol{p}), \tag{7}$$

which is independent of the choice of M (Rousset et al., 2010, proposition 3.40).

The *flow map* associated with the solution to the DAEs in Equation (6) is $\Phi^{h_c}_t : \mathsf{T}^*\mathcal{M} \to \mathsf{T}^*\mathcal{M}$, such that for $(\boldsymbol{q}(0),\boldsymbol{p}(0)) \in \mathsf{T}^*\mathcal{M}$ and $t \geq 0$ we have $(\boldsymbol{q}(t),\boldsymbol{p}(t)) = \Phi^{h_c}_t(\boldsymbol{q}(0),\boldsymbol{p}(0))$. Fundamental properties of $\Phi^{h_c}_t$ are that it is energy conserving and symplectic.

Proposition 2 *The Hamiltonian in Equation (5) is conserved under the flow map* $\Phi^{h_c}_t$.

Proposition 3 *The flow map* $\Phi^{h_c}_t$ *preserves the symplectic form on* $\mathsf{T}^*\mathcal{M}$.

See for example Leimkuhler and Reich (2004 Chapter 7). Proofs are also given in Sections S5 and S6 in the Supplementary Material. Together these properties mean the flow map $\Phi^{h_c}_t$ has an invariant measure on $\mathsf{T}^*\mathcal{M}$.

Corollary 1 *The conservation properties in Propositions 2 and 3 imply that the measure* $\zeta(\mathrm{d}\boldsymbol{q},\mathrm{d}\boldsymbol{p}) \propto \exp(-h(\boldsymbol{q},\boldsymbol{p}))\sigma_{\mathsf{T}^*\mathcal{M}}(\mathrm{d}\boldsymbol{q},\mathrm{d}\boldsymbol{p})$ *is invariant under the flow map* $\Phi^{h_c}_t$ *corresponding to the constrained dynamics in Equation (6).*

Using the definitions in Equations (5) and (7), it readily follows that the target posterior $\pi(\mathrm{d}\boldsymbol{q}) \propto \exp(-\ell(\boldsymbol{q}))\sigma^M_{\mathcal{M}}(\mathrm{d}\boldsymbol{q})$ is the marginal distribution on the position under the invariant measure ζ. Thus, the flow map $\Phi^{h_c}_t$ can be used to construct a family of Markov kernels which marginally leave π invariant.

4.2 | Momentum resampling

As the dynamics remain confined to a level-set of the Hamiltonian in Equation (5), a Markov chain constructed by iterating $\Phi_t^{h_c}$ will not be ergodic. By resampling the momentum between $\Phi_t^{h_c}$ applications we can however move between Hamiltonian level-sets.

To orthogonally (with respect to the co-metric) project a momentum onto $\mathsf{T}^*_{\boldsymbol{q}}\mathcal{M}$, the co-tangent space at $\boldsymbol{q}$, we apply the projector $P_M(\boldsymbol{q})$, defined as

$$P_M(\boldsymbol{q}) := \mathbb{I}_{\mathrm{Q}} - \partial\boldsymbol{c}(\boldsymbol{q})^{\mathsf{T}} G_M(\boldsymbol{q})^{-1} \partial\boldsymbol{c}(\boldsymbol{q}) M^{-1}. \tag{8}$$

Using P_M, we can independently sample a momentum from its conditional distribution given the position under the measure ζ by projecting a sample from $\mathcal{N}(\boldsymbol{0}, M)$.

Proposition 4 *If* $\tilde{\mathsf{p}} \sim \mathcal{N}(\boldsymbol{0}, M)$ *then* $\mathsf{p} = P_M(\boldsymbol{q})\tilde{\mathsf{p}}$ *is distributed with density* $\exp(-\boldsymbol{p}^{\mathsf{T}} M^{-1} \boldsymbol{p}/2)$ *with respect to* $\sigma^{M^{-1}}_{\mathsf{T}^*_{\boldsymbol{q}}\mathcal{M}}$, *the distribution of* $\mathsf{p} \mid \mathsf{q} = \boldsymbol{q}$ for $\mathsf{q}, \mathsf{p} \sim \zeta$.

See Section S7 in the Supplementary Material for a proof.

4.3 | Numerical discretisation

In general, the system of DAEs in Equation (6) will not have an analytic solution, and we are required to use a time discretisation to approximate the exact flow map $\Phi_t^{h_c}$. We will first introduce a class of symplectic integrators for the unconstrained Hamiltonian system before showing how they can be used to construct a symplectic integrator for the constrained Hamiltonian system.

4.3.1 | Unconstrained integrator: Störmer–Verlet and Gaussian splittings

A standard approach for defining symplectic integrators for Hamiltonian systems is to *split* the Hamiltonian into a sum of components for which the exact corresponding flow map can be computed, with a splitting of the form $h(\boldsymbol{q}, \boldsymbol{p}) = h_1(\boldsymbol{q}) + h_2(\boldsymbol{q}, \boldsymbol{p})$ particularly common. If $\Phi_t^{h_1}$ and $\Phi_t^{h_2}$ denote the flow maps associated with the dynamics for Hamiltonians h_1 and h_2, respectively, then the symmetric composition $\Psi_t = \Phi_{t/2}^{h_1} \circ \Phi_t^{h_2} \circ \Phi_{t/2}^{h_1}$ is a symplectic and second-order accurate integrator for the Hamiltonian system (Leimkuhler & Reich, 2004). Furthermore, as both $\Phi_t^{h_1}$ and $\Phi_t^{h_2}$ are time-reversible, Ψ_t is also time-reversible.

Various choices can be made for splitting the Hamiltonian of interest in Equation (5) between h_1 and h_2, subject to the requirement that the flow map $\Phi_t^{h_2}$ can be computed, with $\Phi_t^{h_1}(\boldsymbol{q}, \boldsymbol{p}) = (\boldsymbol{q}, \boldsymbol{p} - t\nabla h_1(\boldsymbol{q}))$ always trivial to compute. An obvious splitting is $h_1(\boldsymbol{q}) = \ell(\boldsymbol{q})$ and $h_2(\boldsymbol{q}, \boldsymbol{p}) = \frac{1}{2}\boldsymbol{p}^{\mathsf{T}} M^{-1} \boldsymbol{p}$; in this case $\Phi_t^{h_2}(\boldsymbol{q}, \boldsymbol{p}) = (\boldsymbol{q} + tM^{-1}\boldsymbol{p}, \boldsymbol{p})$. The composition then corresponds to the Störmer–Verlet integrator (Verlet, 1967).

In our setting, the log prior density on the ambient space $\log \mathrm{d}\rho/\mathrm{d}\lambda_{\mathrm{Q}}$ is quadratic in the components of the position q corresponding to $\mathsf{v}_{1:\mathrm{ST}}$ due to their standard normal prior distribution. It will typically also be possible to choose an appropriate parametrization such that the prior densities $\mathrm{d}\tilde{\mu}/\mathrm{d}\lambda_{\mathrm{Z}}$, $\mathrm{d}\tilde{\nu}/\mathrm{d}\lambda_{\mathrm{X}}$ and $\mathrm{d}\eta/\mathrm{d}\lambda_{\mathrm{W}}$ are equal to or well approximated by standard normal densities. An alternative splitting, which can be useful in this setting, is then $h_1(\boldsymbol{q}) = \ell(\boldsymbol{q}) - \frac{1}{2}\boldsymbol{q}^{\mathsf{T}}\boldsymbol{q}$, $h_2(\boldsymbol{q}, \boldsymbol{p}) = \frac{1}{2}\boldsymbol{q}^{\mathsf{T}}\boldsymbol{q} + \frac{1}{2}\boldsymbol{p}^{\mathsf{T}} M^{-1} \boldsymbol{p}$, with the simplification $h_1(\boldsymbol{q}) = \frac{1}{2}\log|G_M(\boldsymbol{q})|$ when $\log \mathrm{d}\rho/\mathrm{d}\lambda_{\mathrm{Q}}(\boldsymbol{q}) = -\frac{1}{2}\boldsymbol{q}^{\mathsf{T}}\boldsymbol{q}$.

The quadratic form of h_2 and corresponding linear derivatives mean the corresponding flow map is still exactly computable. If we let R be an orthonormal matrix with the normalised eigenvectors of M^{-1} as columns and Ω a diagonal matrix of the square-roots of the eigenvalues such that $M^{-1} = R\Omega^2 R^\mathsf{T}$ then we have that

$$\Phi_t^{h_2}(\boldsymbol{q},\boldsymbol{p}) = \left(R\cos(\Omega t)R^\mathsf{T}\boldsymbol{q} + R\Omega\sin(\Omega t)R^\mathsf{T}\boldsymbol{p}, R\cos(\Omega t)R^\mathsf{T}\boldsymbol{p} - R\Omega^{-1}\sin(\Omega t)R^\mathsf{T}\boldsymbol{q}\right).$$

This splitting and corresponding integrator has been used previously in various settings (Beskos et al., 2011; Beskos et al., 2013a; Neal, 2011; Shahbaba et al., 2014). Importantly as the flow-map $\Phi_t^{h_2}$ exactly preserves Gaussian prior measures, under certain assumptions the change in Hamiltonian over a trajectory generated using the integrator does not grow as the dimension $\mathtt{Q}$ of the space is increased, so the probability of accepting a proposed move from the trajectory remains independent of dimension for a fixed step size. This in contrast to the Störmer–Verlet integrator for which for a fixed step size the accept probability will tend to zero as the dimension becomes large (Beskos et al., 2011).

In the context here of inference in partially observed diffusion models, as the time step δ of the discretisation of the diffusion is decreased (or equivalently $\mathtt{S}$ increased), the dimension of set of latent noise vectors $\mathbf{v}_{1:\mathtt{ST}}$ and so $\mathbf{q}$ will increase, with the prior distribution on $\mathbf{q}$ tending to a distribution with a density with respect to an infinite-dimensional Gaussian measure in the limit $\delta \to 0$. As here the target *posterior* distribution has support only on a submanifold of the ambient space, the results of Beskos et al. (2011) do not directly carry over, however, empirically we have found that a constrained integrator based on this *Gaussian splitting* gives an improved scaling in sampling efficiency with $\mathtt{S}$ compared to the *Störmer–Verlet splitting* as we illustrate in our numerical experiments in Section 7.

4.3.2 | Constrained integrator

We now show how a constraint-preserving symplectic integrator can be formed from the unconstrained integrator Ψ_t. In (Reich 1996 section 3.1) it is observed that the map defined by $\Pi_\lambda(\boldsymbol{q},\boldsymbol{p}) = (\boldsymbol{q}, \boldsymbol{p} - \partial\boldsymbol{c}(\boldsymbol{q})^\mathsf{T}\lambda(\boldsymbol{q},\boldsymbol{p}))$, with $\boldsymbol{q} \in \mathcal{M}$, is symplectic for any function λ that is sufficiently regular (e.g. continuously differentiable). Reich (1996) then shows that if a second-order accurate symplectic integrator for an unconstrained system with Hamiltonian as in Equation (5) is defined by the map Ψ_t, then the integrator with step defined by the composition $(\boldsymbol{q}', \boldsymbol{p}') = \Pi_{\lambda'} \circ \Psi_t \circ \Pi_\lambda(\boldsymbol{q},\boldsymbol{p})$, with λ implicitly defined by solving for the primary constraints, $\boldsymbol{c}(\boldsymbol{q}') = \boldsymbol{0}$, and λ' by solving for the secondary constraints, $\partial\boldsymbol{c}(\boldsymbol{q}')M^{-1}\boldsymbol{p}' = \boldsymbol{0}$, is a second-order accurate symplectic integrator for the corresponding constrained system.

Rather than composing instances of Π_λ with the overall map Ψ_t as proposed by Reich (1996), we can instead consider composing Π_λ with the component maps which make up Ψ_t to enforce the constraints within each 'sub-step'. This was proposed for the specific case of Ψ_t corresponding to a Störmer–Verlet integrator in the *geodesic integration* algorithm of Leimkuhler and Matthews (2016).

For a general quadratic h_2 (covering both the Störmer–Verlet and Gaussian splittings introduced above), the associated component flow-maps $\Phi_t^{h_1}$ and $\Phi_t^{h_2}$ can be expressed for suitable choices of matrices $(\Gamma_t^{q,q}, \Gamma_t^{q,p}, \Gamma_t^{p,q}, \Gamma_t^{p,p})$ as

$$\Phi_t^{h_1}(\boldsymbol{q},\boldsymbol{p}) := (\boldsymbol{q}, \boldsymbol{p} - t\nabla h_1(\boldsymbol{q})), \quad \Phi_t^{h_2}(\boldsymbol{q},\boldsymbol{p}) := \left(\Gamma_t^{q,q}\boldsymbol{q} + \Gamma_t^{q,p}\boldsymbol{p}, \Gamma_t^{p,q}\boldsymbol{q} + \Gamma_t^{p,p}\boldsymbol{p}\right). \tag{9}$$

First considering the flow-map $\Phi_t^{h_1}$, we define the constraint-preserving composition

$$\Xi_t^{h_1}(\boldsymbol{q},\boldsymbol{p}) := \Pi_\lambda \circ \Phi_t^{h_1}(\boldsymbol{q},\boldsymbol{p}) = \left(\boldsymbol{q}, \boldsymbol{p} - t\nabla h_1(\boldsymbol{q}) - \partial \boldsymbol{c}(\boldsymbol{q})^\mathsf{T}\lambda\right), \tag{10}$$

with λ implicitly defined by the condition $\Xi_t^{h_1}(\boldsymbol{q},\boldsymbol{p}) \in \mathsf{T}^*\mathcal{M}\ \forall(\boldsymbol{q},\boldsymbol{p}) \in \mathsf{T}^*\mathcal{M}$. Solving for λ yields the explicit definition $\Xi_t^{h_1}(\boldsymbol{q},\boldsymbol{p}) := (\boldsymbol{q}, P_M(\boldsymbol{q})(\boldsymbol{p} - t\nabla h_1(\boldsymbol{q})))$. As $\Xi_{-t}^{h_1} \circ \Xi_t^{h_1}(\boldsymbol{q},\boldsymbol{p}) = (\boldsymbol{q},\boldsymbol{p})$ for all $(\boldsymbol{q},\boldsymbol{p}) \in \mathsf{T}^*\mathcal{M}$, the mapping $\Xi_t^{h_1}$ is time reversible. Now considering the $\Phi_t^{h_2}$ map, we first consider the composition

$$\Phi_t^{h_2} \circ \Pi_\lambda(\boldsymbol{q},\boldsymbol{p}) = \left(\Gamma_t^{q,q}\boldsymbol{q} + \Gamma_t^{q,p}\left(\boldsymbol{p} - \partial\boldsymbol{c}(\boldsymbol{q})^\mathsf{T}\lambda\right), \Gamma_t^{p,q}\boldsymbol{q} + \Gamma_t^{p,p}\left(\boldsymbol{p} - \partial\boldsymbol{c}(\boldsymbol{q})^\mathsf{T}\lambda\right)\right), \tag{11}$$

with λ implicitly defined by requiring the following to hold for any $(\boldsymbol{q},\boldsymbol{p}) \in \mathsf{T}^*\mathcal{M}$,

$$\boldsymbol{c}\left(\Gamma_t^{q,q}\boldsymbol{q} + \Gamma_t^{q,p}\boldsymbol{p} - \Gamma_t^{q,p}\partial\boldsymbol{c}(\boldsymbol{q})^\mathsf{T}\lambda\right) = \boldsymbol{0}. \tag{12}$$

For general constraint functions $\boldsymbol{c}$, this is a non-linear system of equations in λ that needs to be solved using an iterative method. Newton's method gives the update

$$\begin{aligned}(\boldsymbol{q}_\mathsf{j}, \boldsymbol{p}_\mathsf{j}) &= \Phi_t^{h_2}\left(\boldsymbol{q}, \boldsymbol{p} - \partial\boldsymbol{c}(\boldsymbol{q})^\mathsf{T}\lambda_\mathsf{j}\right),\\ \lambda_{\mathsf{j}+1} &= \lambda_\mathsf{j} + \left(\partial\boldsymbol{c}(\boldsymbol{q}_\mathsf{j})\left(\Gamma_t^{q,p}\right)^{-1}\partial\boldsymbol{c}(\boldsymbol{q})^\mathsf{T}\right)^{-1}\boldsymbol{c}(\boldsymbol{q}_\mathsf{j}) \quad \text{with } \lambda_0 = \boldsymbol{0}.\end{aligned} \tag{13}$$

Assuming for now the iterative solver can find a value for λ to satisfy Equation (12), the composition in Equation (11) preserves the primary constraints, but not the secondary constraints in general. The secondary constraints can be enforced by composing with a further instance of the map $\Pi_{\lambda'}$ resulting in the overall composition $\Xi_t^{h_2}(\boldsymbol{q},\boldsymbol{p}) := \Pi_{\lambda'} \circ \Phi_t^{h_2} \circ \Pi_\lambda(\boldsymbol{q},\boldsymbol{p})$ with λ' implicitly defined by the condition $\Xi_t^{h_2}(\boldsymbol{q},\boldsymbol{p}) \in \mathsf{T}^*\mathcal{M}\ \forall(\boldsymbol{q},\boldsymbol{p}) \in \mathsf{T}^*\mathcal{M}$. This can be explicitly solved for λ' to give $\Xi_t^{h_2}(\boldsymbol{q},\boldsymbol{p}) = (\overline{\boldsymbol{q}}, P_M(\overline{\boldsymbol{q}})\overline{\boldsymbol{p}})$ with $(\overline{\boldsymbol{q}},\overline{\boldsymbol{p}}) = \Phi_t^{h_2} \circ \Pi_\lambda(\boldsymbol{q},\boldsymbol{p})$ as defined in Equations (11) and (12).

For sufficiently small t and sufficiently smooth constraint functions, it can be shown that there exists a locally unique solution to Equation (12) (Brubaker et al., 2012; Lelièvre et al., 2019). In general, though, there may be multiple or no solutions, and even if there is a unique solution the iterative solver may fail to converge. This lack of a guarantee of converging to a unique solution presents a challenge in terms of maintaining the time-reversibility of the $\Xi_t^{h_2}$ step and so the overall integrator.

To enforce reversibility on $\Xi_t^{h_2}$, we apply a *reversibility-check transform* R defined such that $R(\Xi_t^{h_2})(\boldsymbol{q},\boldsymbol{p}) = \Xi_t^{h_2}(\boldsymbol{q},\boldsymbol{p})$ for all $(\boldsymbol{q},\boldsymbol{p})$ where $\Xi_{-t}^{h_2} \circ \Xi_t^{h_2}(\boldsymbol{q},\boldsymbol{p}) = (\boldsymbol{q},\boldsymbol{p})$, with $(\boldsymbol{q},\boldsymbol{p})$ values for which the condition is not met causing evaluation of $R(\Xi_t^{h_2})(\boldsymbol{q},\boldsymbol{p})$ to raise an error. Similarly if the iterative solves in the evaluation of either the forward $\Xi_t^{h_2}$ or time-reversed $\Xi_t^{h_2}$ maps fail to converge an error is also raised. The map $R(\Xi_t^{h_2})$ is then by construction reversible unless an error is raised that can be suitably handled downstream by the HMC implementation. The approach of using an explicit reversibility check in MCMC methods using an iterative solver was first proposed by Zappa et al. (2018) with subsequent application within the context of constrained HMC in Graham and Storkey (2017) and Lelièvre et al. (2019).

With the reversibility check, the map $R(\Xi_t^{h_2})$ is guaranteed to be time-reversible if it does not raise an error. As $\Xi_t^{h_1}$ is also time reversible and both maps are symplectic, the integrator $\Xi_{t/2}^{h_1} \circ R(\Xi_t^{h_2}) \circ \Xi_{t/2}^{h_1}$ defines a time-reversible symplectic map on $\mathsf{T}^*\mathcal{M}$ whenever an error is not raised. In practice, the equality conditions indicating whether the iterative solver has converged and the reversibility check is satisfied, are both relaxed to an error norm being less than tolerances θ_c and θ_q, respectively. Further details of the implementation of the integrator and its relation to previous work are given in Section S8 in the Supplementary Material.

4.4 | Choice of metric matrix representation M

We recommend choosing $M = \operatorname{cov}(\mathbf{q})^{-1}$ for $\mathbf{q} \sim \rho$, that is the precision matrix under the prior ρ; this requires that ρ has finite second-order central moments. While there is no fundamental requirement for M to match the prior precision matrix and so a different choice of M could be used when $\operatorname{cov}(\mathbf{q})^{-1}$ is not defined, heuristically we find that the performance of the proposed methodology is improved when ρ is exactly or 'close to' Gaussian in all components, and this can usually be arranged by transforms of $\mathbf{u}$, $\mathbf{v}_0$ and $\mathbf{w}_{1:\mathtt{T}}$ and corresponding reparametrisations of $(\tilde{\mu}, \mathbf{g}_{\mathtt{Z}})$, $(\tilde{\nu}, \mathbf{g}_{\mathtt{X}_0})$ and $(\eta, \boldsymbol{h})$. The constrained Hamiltonian dynamics in Equation (6) with $M = \operatorname{cov}(\mathbf{q})^{-1}$ are equivalent to the dynamics under a linear transform $\mathbf{q}' = L^{\mathsf{T}}\mathbf{q}$ with $LL^{\mathsf{T}} = M$ for which $\operatorname{cov}(\mathbf{q}') = \mathbb{I}_{\mathtt{Q}}$ and normalising for the prior scales and correlations in this manner appears to improve the robustness and efficiency of the algorithm.

4.5 | Overall algorithm

Pseudo-code corresponding to applying the reversible, constraint-preserving and symplectic integrator with step $\Xi^{h_1}_{t/2} \circ R(\Xi^{h_2}_{t}) \circ \Xi^{h_1}_{t/2}$ within a HMC algorithm is summarised in Algorithm 1. Any errors raised when integrating the trajectory by iteratively applying the ConstrStep function are handled by terminating the trajectory and the HMC transition returning the initial state, that is a 'rejection'. Although for simplicity we have described in Algorithm 1 the use of a constraint-preserving integrator within a Metropolis-adjusted HMC algorithm with a static integration time $\mathtt{I}t$ per chain iteration, in practice we use a HMC algorithm which dynamically adapts the integration time $\mathtt{I}t$, in particular the dynamic multinomial HMC algorithm described in the appendix of Betancourt (2017), an extension of the *No-U-Turn-Sampler* algorithm (Hoffman & Gelman, 2014). We also use the dual-averaging scheme of Hoffman and Gelman (2014) to adaptively tune the integrator step-size t in a *warm-up* sampling phase to target an acceptance statistic of 0.8. A general purpose implementation of the full algorithm is provided in Python package *Mici* (Graham, 2019), which we use in the numerical experiments in Section 7.

We have found the suggested defaults values for the various algorithmic parameters work well in practice for a range of different models. This therefore results in an automated methodology with a practitioner only needing to specify functions to evaluate the log prior density $\log \mathrm{d}\rho/\mathrm{d}\lambda_{\mathtt{Q}}$ and constraint function $\boldsymbol{c}$ for the diffusion model in question, with the required derivatives of these functions being able to be constructed algorithmically (Griewank & Walther, 2008).

5 | COMPUTATIONAL COST

We can apply Algorithm 1 to perform inference in partially observed diffusion models by targeting the manifold-supported posterior distribution in the non-centred parametrisation of the time-discretised model described in Section 3.2. While this approach allows significant generality in the choice of the elements of the diffusion and observation model, it can be computationally

Algorithm 1 Hamiltonian Monte Carlo with a constrained symplectic integrator.

Inputs: (reasonable default values for parameters are given in parenthesis)

$\boldsymbol{q}$: current state with $\|\boldsymbol{c}(\boldsymbol{q})\| < \theta_c$, t : integrator time step,
θ_c : constraint tolerance (10^{-9}), $\mathtt{I}$: number of integrator steps / sample,
θ_q: position change tolerance (10^{-8}), $\mathtt{J}$: maximum Newton iterations (50).

Outputs:

$\boldsymbol{q}'$: next state with $\|\boldsymbol{c}(\boldsymbol{q}')\| < \theta_c$ and if $\boldsymbol{q} \sim \pi \implies \boldsymbol{q}' \sim \pi$.

```
function $\Xi^{h_1}(\boldsymbol{q}, \boldsymbol{p}, t)$
  return $(\boldsymbol{q}, P_M(\boldsymbol{q})(\boldsymbol{p} - t\,\nabla h_1(\boldsymbol{q})))$
function $\Xi^{h_2}(\boldsymbol{q}, \boldsymbol{p}, t)$
  $(\boldsymbol{\lambda}, \bar{\boldsymbol{q}}) = (\boldsymbol{0}, \boldsymbol{q})$
  for $\mathtt{j} \in 1{:}\mathtt{J}$
    $(\boldsymbol{q}', \boldsymbol{p}') = \Phi_t^{h_2}(\boldsymbol{q}, \boldsymbol{p} - \partial\boldsymbol{c}(\boldsymbol{q})^{\mathsf{T}}\boldsymbol{\lambda})$
    $\boldsymbol{e} = \boldsymbol{c}(\boldsymbol{q}')$
    if $\|\boldsymbol{e}\| < \theta_c$ and $\|\boldsymbol{q}' - \bar{\boldsymbol{q}}\| < \theta_q$
      return $(\boldsymbol{q}', P_M(\boldsymbol{q}')\boldsymbol{p}')$
    $\boldsymbol{\lambda} = \boldsymbol{\lambda} + (\partial\boldsymbol{c}(\boldsymbol{q}')(\Gamma_t^{q,p})^{-1}\,\partial\boldsymbol{c}(\boldsymbol{q})^{\mathsf{T}})^{-1}\boldsymbol{e}$
    $\bar{\boldsymbol{q}} = \boldsymbol{q}'$
  throw IntegratorError
function Reversible$\Xi^{h_2}(\boldsymbol{q}, \boldsymbol{p}, t)$
  $(\boldsymbol{q}', \boldsymbol{p}') = \Xi^{h_2}(\boldsymbol{q}, \boldsymbol{p}, t)$
  $(\boldsymbol{q}_r, \boldsymbol{p}_r) = \Xi^{h_2}(\boldsymbol{q}', \boldsymbol{p}', -t)$
  if $\|\boldsymbol{q} - \boldsymbol{q}_r\|_\infty > 2\theta_q$
    throw IntegratorError
  return $(\boldsymbol{q}', \boldsymbol{p}')$
function ConstrStep$(\boldsymbol{q}, \boldsymbol{p}, t)$
  $(\boldsymbol{q}, \boldsymbol{p}) = \Xi^{h_1}(\boldsymbol{q}, \boldsymbol{p}, t/2)$
  $(\boldsymbol{q}, \boldsymbol{p}) = $ Reversible$\Xi^{h_2}(\boldsymbol{q}, \boldsymbol{p}, t)$
  return $\Xi^{h_1}(\boldsymbol{q}, \boldsymbol{p}, t/2)$
$\tilde{\boldsymbol{p}} \sim \mathcal{N}(\boldsymbol{0}, M)$
$\boldsymbol{p} = P_M(\boldsymbol{q})\tilde{\boldsymbol{p}}$
$(\boldsymbol{q}', \boldsymbol{p}') = (\boldsymbol{q}, \boldsymbol{p})$
try
  for $\mathtt{i} \in 1{:}\mathtt{I}$
    $(\boldsymbol{q}', \boldsymbol{p}') = $ ConstrStep$(\boldsymbol{q}', \boldsymbol{p}', t)$
  $u \sim \mathcal{U}(0, 1)$
  if $u > \exp(h(\boldsymbol{q}, \boldsymbol{p}) - h(\boldsymbol{q}', \boldsymbol{p}'))$
    $(\boldsymbol{q}', \boldsymbol{p}') = (\boldsymbol{q}, -\boldsymbol{p})$
catch IntegratorError
  $(\boldsymbol{q}', \boldsymbol{p}') = (\boldsymbol{q}, -\boldsymbol{p})$
```

expensive to run. To analyse the cost of Algorithm 1 in this setting, we make the following simplifying assumption.

Assumption 4 The Newton iteration to solve (12) converges within $\mathtt{J}$ iterations for some $\mathtt{J} > 0$ that does not depend on $\mathtt{S}$ and $\mathtt{T}$, for fixed t, θ_c and θ_q.

This assumption appears to hold in practice, and we provide numerical evidence to this effect in the numerical experiments in Section 7. We then have the following.

Proposition 5 *Under Assumption 4, the computational cost of a single constrained integrator step in Algorithm* 1 *when directly applied to the posterior density* (4) *of the generative model in Model* 2 *is* $\mathcal{O}(\mathtt{S}\mathtt{T}^3)$.

A proof is given in Section S1 in the Supplementary Material. The cost of Algorithm 1 when applied directly to the posterior distribution with density in Equation (4) therefore scales linearly with the number of discrete time steps per observation $\mathtt{S}$ but cubically with the number of observation times $\mathtt{T}$.

5.1 | Exploiting Markovianity for scalability

While we have so far considered only sampling from the posterior distribution on latent variables $(\mathbf{u}, \mathbf{v}_0, \mathbf{v}_{1:\mathtt{ST}}, \mathbf{w}_{1:\mathtt{T}})$ given observations $\mathbf{y}_{1:\mathtt{T}}$, the constrained HMC approach we have described can be applied to sampling from any conditional distribution of the joint distribution on observations and latent variables under the generative model, of which the target posterior distribution is just one example.

One way to improve the scaling of the computational cost with respect to the number of observation times is therefore to restrict the information flow through the state sequence $\mathbf{x}_{1:\mathtt{ST}}$ by conditioning on a set of intermediate states in the sequence. Due to the Markovian nature of the state dynamics, the state sequences $\mathbf{x}_{0:\mathtt{s}-1}$ and $\mathbf{x}_{\mathtt{s}+1:\mathtt{ST}}$ are conditionally independent given the state $\mathbf{x}_\mathtt{s}$ and the parameters $\mathbf{z}$ for any $\mathtt{s} \in 1:\mathtt{ST}$. As a consequence under the non-centred parametrisation of the generative model in Model 2, we have that if we condition on the intermediate state $\mathbf{x}_{\mathtt{St}}$ at the $\mathtt{t}^{\text{th}}$ observation time then we can independently generate the observation sequences $\mathbf{y}_{1:\mathtt{t}}$ and $\mathbf{y}_{\mathtt{t}+1:\mathtt{T}}$ from respectively $(\mathbf{u}, \mathbf{v}_{0:\mathtt{St}}, \mathbf{w}_{1:\mathtt{t}})$ and $(\mathbf{u}, \mathbf{v}_{\mathtt{St}+1:\mathtt{ST}}, \mathbf{w}_{\mathtt{t}+1:\mathtt{T}})$.

We can extend this idea to conditioning on multiple intermediate states in the sequence. If at $\mathtt{B}-1$ observation time indices $\mathtt{t}_{1:\mathtt{B}-1} \subseteq 1:\mathtt{T}$, the full state is conditioned on, $\mathbf{x}_{\mathtt{St}_\mathtt{b}} = \boldsymbol{x}_{\mathtt{St}_\mathtt{b}} \forall \mathtt{b} \in 1:\mathtt{B}-1$, then we have that the observation subsequence $\mathbf{y}_{\mathtt{t}_{\mathtt{b}-1}+1:\mathtt{t}_\mathtt{b}}$ and conditioned state $\mathbf{x}_{\mathtt{St}_\mathtt{b}}$ depend only on the latent variables $(\mathbf{u}, \mathbf{v}_{\mathtt{St}_{\mathtt{b}-1}+1:\mathtt{St}_\mathtt{b}}, \mathbf{w}_{\mathtt{t}_{\mathtt{b}-1}+1:\mathtt{t}_\mathtt{b}})$ for each $\mathtt{b} \in 1:\mathtt{B}-1$ (with $\mathtt{t}_0 = 0$) and the final observation subsequence $\mathbf{y}_{\mathtt{t}_{\mathtt{B}-1}+1:\mathtt{T}}$ depends only on the latent variables $(\mathbf{u}, \mathbf{v}_{\mathtt{St}_{\mathtt{B}-1}+1:\mathtt{ST}}, \mathbf{w}_{\mathtt{t}_{\mathtt{B}-1}+1:\mathtt{T}})$.

Due to these conditional independencies introduced when conditioning on the values of $(\mathbf{x}_{\mathtt{St}_\mathtt{b}})_{\mathtt{b}=1}^{\mathtt{B}-1}$, we can 'split' the generation of the state sequence in to $\mathtt{B}$ independent calls to a function which given a conditioned state $\boldsymbol{x}_{\mathtt{St}_{\mathtt{b}-1}}$ generates the subsequence of states for step indices $\mathtt{St}_{\mathtt{b}-1}+1:\mathtt{St}_\mathtt{b}$ and outputs the observations $\mathbf{y}_{\mathtt{t}_{\mathtt{b}-1}+1:\mathtt{t}_\mathtt{b}}$ and final state $\mathbf{x}_{\mathtt{St}_\mathtt{b}}$ of the subsequence (or just observations $\mathbf{y}_{\mathtt{t}_{\mathtt{B}-1}+1:\mathtt{T}}$ for the final subsequence). For noiseless observations, $\mathbf{y}_{\mathtt{t}_\mathtt{b}}$ is completely determined by $\mathbf{x}_{\mathtt{St}_\mathtt{b}}$, and so only $\mathbf{y}_{\mathtt{t}_{\mathtt{b}-1}+1:\mathtt{t}_\mathtt{b}-1}$ and $\mathbf{x}_{\mathtt{St}_\mathtt{b}}$ should be returned for the non-final subsequences. The resulting conditioned generative model is summarised in Model 3.

Model 3 Generative model conditioning on intermediate states $(\boldsymbol{x}_{\mathtt{St}_\mathtt{b}})_{\mathtt{b}=1}^{\mathtt{B}-1}$.

```
function $g_{\bar{\mathbf{y}}}(\boldsymbol{u},\ \boldsymbol{v}_0,\ \boldsymbol{v}_{1:\mathtt{St}},\ \boldsymbol{w}_{1:\mathtt{t}},\ \mathtt{b})$
    $\boldsymbol{z} = g_{\mathbf{z}}(\boldsymbol{u})$
    $\boldsymbol{x}_0 = g_{\mathbf{x}_0}(\boldsymbol{z}, \boldsymbol{v}_0)$ if $\mathtt{b} \equiv 1$ else $\boldsymbol{v}_0$
    $\boldsymbol{x}_{1:\mathtt{St}},\ \boldsymbol{y}_{1:\mathtt{t}} = g_{\mathbf{x}_:,\mathbf{y}_:}(\boldsymbol{z}, \boldsymbol{x}_0, \boldsymbol{v}_{1:\mathtt{St}}, \boldsymbol{w}_{1:\mathtt{t}})$
    if $\mathtt{b} \neq \mathtt{B}$
        return $[\boldsymbol{y}_{1:\mathtt{t}}; \boldsymbol{x}_{\mathtt{St}}]$
    else
        return $[\boldsymbol{y}_{1:\mathtt{t}}]$

$\mathbf{u} \sim \tilde{\mu}$
$\mathbf{v}_0 \sim \tilde{\nu}$
$\mathbf{v}_\mathtt{s} \sim \mathcal{N}(\mathbf{0}, \mathbb{I}_\mathtt{V})\ \forall \mathtt{s} \in 1{:}\mathtt{ST}$
$\mathbf{w}_\mathtt{t} \sim \eta\ \forall \mathtt{t} \in 1{:}\mathtt{T}$
$\bar{\mathbf{y}}_1 = g_{\bar{\mathbf{y}}}(\mathbf{u}, \mathbf{v}_0, \mathbf{v}_{1:\mathtt{St}_1}, \mathbf{w}_{1:\mathtt{t}_1}, 1)$
for $\mathtt{b} \in 2{:}\mathtt{B}$                                # $\mathtt{t}_\mathtt{B} \equiv \mathtt{T}$
    $\mathbf{v}_:, \mathbf{w}_: = \mathbf{v}_{\mathtt{St}_{\mathtt{b}-1}+1:\mathtt{St}_\mathtt{b}}, \mathbf{w}_{\mathtt{t}_{\mathtt{b}-1}+1:\mathtt{t}_\mathtt{b}}$
    $\bar{\mathbf{y}}_\mathtt{b} = g_{\bar{\mathbf{y}}}(\mathbf{u}, \boldsymbol{x}_{\mathtt{St}_{\mathtt{b}-1}}, \mathbf{v}_:, \mathbf{w}_:, \mathtt{b})$
```

Using $\mathbf{g}_{\bar{\mathbf{y}}}$ from Model 3, we can then define *partial* constraint functions

$$\begin{aligned}
&\boldsymbol{c}_1(\boldsymbol{u}, [\boldsymbol{v}_0; \boldsymbol{v}_{1:\mathtt{St}_1}], [\boldsymbol{w}_{1:\mathtt{t}_1}]) := \mathbf{g}_{\bar{\mathbf{y}}}(\boldsymbol{u}, \boldsymbol{v}_0, \boldsymbol{v}_{1:\mathtt{St}_1}, \boldsymbol{w}_{1:\mathtt{t}_1}, 1) - \bar{\boldsymbol{y}}_1, \text{ and } \forall\, \mathtt{b} \in 2:\mathtt{B} \\
&\boldsymbol{c}_\mathtt{b}(\boldsymbol{u}, [\boldsymbol{v}_{\mathtt{St}_{\mathtt{b}-1}+1:\mathtt{St}_\mathtt{b}}], [\boldsymbol{w}_{\mathtt{t}_{\mathtt{b}-1}+1:\mathtt{t}_\mathtt{b}}]) := \mathbf{g}_{\bar{\mathbf{y}}}(\boldsymbol{u}, \boldsymbol{x}_{\mathtt{St}_{\mathtt{b}-1}}, \boldsymbol{v}_{\mathtt{St}_{\mathtt{b}-1}+1:\mathtt{St}_\mathtt{b}}, \boldsymbol{w}_{\mathtt{t}_{\mathtt{b}-1}+1:\mathtt{t}_\mathtt{b}}, \mathtt{b}) - \bar{\boldsymbol{y}}_\mathtt{b}
\end{aligned}$$

with $\bar{\boldsymbol{y}}_\mathtt{b} = [\boldsymbol{y}_{\mathtt{t}_{\mathtt{b}-1}+1:\mathtt{t}_\mathtt{b}}; \boldsymbol{x}_{\mathtt{St}_\mathtt{b}}]\ \forall\, \mathtt{b} \in 1:\mathtt{B}-1$ and $\bar{\boldsymbol{y}}_\mathtt{B} = [\boldsymbol{y}_{\mathtt{t}_{\mathtt{B}-1}+1:\mathtt{T}}]$. We then define, respectively, *partitioned* and *full* constraint functions $\bar{\boldsymbol{c}} : \mathbb{R}^\mathtt{U} \times \mathbb{R}^{\mathtt{V}_0+\mathtt{STV}} \times \mathbb{R}^{\mathtt{TW}} \to \mathbb{R}^\mathtt{C}$ and $\boldsymbol{c} : \mathbb{R}^\mathtt{Q} \to \mathbb{R}^\mathtt{C}$, with $\mathtt{C} = (\mathtt{B}-1)\mathtt{X} + \mathtt{TY}$ the number of constraints, as

$$\bar{\boldsymbol{c}}\left(\boldsymbol{u}, [\bar{\boldsymbol{v}}_{1:\mathtt{B}}], [\bar{\boldsymbol{w}}_{1:\mathtt{B}}]\right) := \boldsymbol{c}\left([\boldsymbol{u}; \bar{\boldsymbol{v}}_{1:\mathtt{B}}; \bar{\boldsymbol{w}}_{1:\mathtt{B}}]\right) = \left[\left(\boldsymbol{c}_\mathtt{b}(\boldsymbol{u}, \bar{\boldsymbol{v}}_\mathtt{b}, \bar{\boldsymbol{w}}_\mathtt{b})\right)_{\mathtt{b}\in 1:\mathtt{B}}\right]. \tag{14}$$

The Jacobian of the full constraint function will then have the block structure $\partial\boldsymbol{c}([\boldsymbol{u};\bar{\boldsymbol{v}};\bar{\boldsymbol{w}}]) = \begin{bmatrix}\partial_1\bar{\boldsymbol{c}}(\boldsymbol{u},\bar{\boldsymbol{v}},\bar{\boldsymbol{w}}) & \partial_2\bar{\boldsymbol{c}}(\boldsymbol{u},\bar{\boldsymbol{v}},\bar{\boldsymbol{w}}) & \partial_3\bar{\boldsymbol{c}}(\boldsymbol{u},\bar{\boldsymbol{v}},\bar{\boldsymbol{w}})\end{bmatrix}$ with $\partial_1\bar{\boldsymbol{c}}(\boldsymbol{u},\bar{\boldsymbol{v}},\bar{\boldsymbol{w}})$ a dense $\mathtt{C}\times\mathtt{U}$ matrix, and $\partial_{\mathtt{i}}\bar{\boldsymbol{c}}(\boldsymbol{u},\bar{\boldsymbol{v}},\bar{\boldsymbol{w}})$ for $\mathtt{i}\in\{2,3\}$ block diagonal $\mathtt{C}\times(\mathtt{V}_0+\mathtt{STV})$ $(\mathtt{i}=2)$ and $\mathtt{C}\times\mathtt{TW}$ $(\mathtt{i}=3)$ matrices with $\partial_{\mathtt{i}}\bar{\boldsymbol{c}}(\boldsymbol{u},[\bar{\boldsymbol{v}}_{1:\mathtt{B}}],[\bar{\boldsymbol{w}}_{1:\mathtt{B}}]) = \mathrm{diag}(\partial_{\mathtt{i}}\boldsymbol{c}_{\mathtt{b}}(\boldsymbol{u},\bar{\boldsymbol{v}}_{\mathtt{b}},\bar{\boldsymbol{w}}_{\mathtt{b}}))_{\mathtt{b}\in 1:\mathtt{B}}$.

As $\mathbf{u}$, $[\mathbf{v}_{0:\mathtt{ST}}]$ and $[\mathbf{w}_{1:\mathtt{T}}]$ are independent under the prior ρ, under the recommendation in Section 4.4 the metric matrix is $M=\mathrm{diag}(M_u,M_v,M_w)$ with M_u a $\mathtt{U}\times\mathtt{U}$ matrix, M_v a $(\mathtt{V}_0+\mathtt{STV})\times(\mathtt{V}_0+\mathtt{STV})$ block-diagonal matrix and M_w a $\mathtt{TW}\times\mathtt{TW}$ block-diagonal matrix. The Gram matrix can then be decomposed as

$$\begin{aligned}G_M([\boldsymbol{u};\bar{\boldsymbol{v}};\bar{\boldsymbol{w}}]) &= \partial_1\bar{\boldsymbol{c}}(\boldsymbol{u},\bar{\boldsymbol{v}},\bar{\boldsymbol{w}})M_u^{-1}\partial_1\bar{\boldsymbol{c}}(\boldsymbol{u},\bar{\boldsymbol{v}},\bar{\boldsymbol{w}})^{\mathsf{T}} + D([\boldsymbol{u};\bar{\boldsymbol{v}};\bar{\boldsymbol{w}}]) \text{ with}\\ D([\boldsymbol{u};\bar{\boldsymbol{v}};\bar{\boldsymbol{w}}]) &:= \partial_2\bar{\boldsymbol{c}}(\boldsymbol{u},\bar{\boldsymbol{v}},\bar{\boldsymbol{w}})M_v^{-1}\partial_2\bar{\boldsymbol{c}}(\boldsymbol{u},\bar{\boldsymbol{v}},\bar{\boldsymbol{w}})^{\mathsf{T}} + \partial_3\bar{\boldsymbol{c}}(\boldsymbol{u},\bar{\boldsymbol{v}},\bar{\boldsymbol{w}})M_w^{-1}\partial_3\bar{\boldsymbol{c}}(\boldsymbol{u},\bar{\boldsymbol{v}},\bar{\boldsymbol{w}})^{\mathsf{T}},\end{aligned} \tag{15}$$

corresponding to a rank $\mathtt{U}$ correction of a block-diagonal matrix $D([\boldsymbol{u};\bar{\boldsymbol{v}};\bar{\boldsymbol{w}}])$.

Using the matrix determinant lemma, we then have that

$$\log|G_M(\boldsymbol{q})| = \log|C(\boldsymbol{q})| + \log|D(\boldsymbol{q})| - \log|M_u|, \tag{16}$$

with $C([\boldsymbol{u};\bar{\boldsymbol{v}};\bar{\boldsymbol{w}}]) := M_u + \partial_1\bar{\boldsymbol{c}}(\boldsymbol{u},\bar{\boldsymbol{v}},\bar{\boldsymbol{w}})^{\mathsf{T}}D([\boldsymbol{u};\bar{\boldsymbol{v}};\bar{\boldsymbol{w}}])^{-1}\partial_1\bar{\boldsymbol{c}}(\boldsymbol{u},\bar{\boldsymbol{v}},\bar{\boldsymbol{w}})$. Similarly, the Woodbury matrix identity yields, for a vector $\boldsymbol{r}\in\mathbb{R}^{\mathtt{C}}$ and $\boldsymbol{q}=[\boldsymbol{u};\bar{\boldsymbol{v}};\bar{\boldsymbol{w}}]$, that

$$G_M(\boldsymbol{q})^{-1}\boldsymbol{r} = D(\boldsymbol{q})^{-1}\left(\boldsymbol{r} - \partial_1\bar{\boldsymbol{c}}(\boldsymbol{u},\bar{\boldsymbol{v}},\bar{\boldsymbol{w}})C(\boldsymbol{q})^{-1}\partial_1\bar{\boldsymbol{c}}(\boldsymbol{u},\bar{\boldsymbol{v}},\bar{\boldsymbol{w}})^{\mathsf{T}}D(\boldsymbol{q})^{-1}\boldsymbol{r}\right). \tag{17}$$

By applying a sequence of constrained HMC Markov kernels, each conditioning on intermediate states $(\mathbf{x}_{\mathtt{St}_{\mathtt{b}}})_{\mathtt{b}=1}^{\mathtt{B}-1}$ at a different set of observation time indices $\mathtt{t}_{1:\mathtt{B}-1}$ as well as the observations $\mathbf{y}_{1:\mathtt{T}}$ we can construct a MCMC method which asymptotically samples from the target posterior distribution at a substantially reduced computational cost compared to the case of conditioning only on the observations $\mathbf{y}_{1:\mathtt{T}}$. To analyse the computational cost of applying the constrained HMC implementation in Algorithm 1 to the conditioned generative model, we assume the following.

Assumption 5 $\mathtt{T}=\mathtt{BR}$ and $\mathtt{t}_{\mathtt{b}}=\mathtt{bR}\,\forall\,\mathtt{b}\in 1:\mathtt{B}-1$, that is, that the observations are split in to $\mathtt{B}$ equally sized subsequences of $\mathtt{R}$ observation times.

Assumption 6 The Newton iteration to solve (12) converges within $\mathtt{J}$ iterations for $\mathtt{J}>0$ that does not depend on $\mathtt{R}$, $\mathtt{S}$ and $\mathtt{T}$, for fixed t, θ_c and θ_q.

In practice, we will need to alternate with conditioning on a different set of observation times to allow the Markov chain to be ergodic, for example, $\mathtt{t}'_{\mathtt{b}} = \lfloor\frac{(2\mathtt{b}-1)\mathtt{R}}{2}\rfloor\,\forall\,\mathtt{b}\in 1:\mathtt{B}$, with in this case the observation times split in to $\mathtt{B}-1$ subsequences of $\mathtt{R}$ observations times and two subsequences of $\lfloor\frac{\mathtt{R}}{2}\rfloor$ and $\lceil\frac{\mathtt{R}}{2}\rceil$ observation times. This alternative splitting will result in only minor difference in operation cost compared to the equispaced partition hence we consider only the former case in our analysis. Assumption 6 is motivated by our observation that the average number of Newton iterations needed for convergence appears to be independent of $\mathtt{R}$, $\mathtt{S}$ and $\mathtt{T}$.

Proposition 6 *Under Assumptions 5 and 6, the computational cost of a single constrained integrator step in Algorithm 1 when applied to the posterior of the generative model conditioning additionally on the values of the states at observation time indices $\mathtt{t}_{1:\mathtt{B}-1}$ as in Model 3 is $\mathcal{O}(\mathtt{R}^2\mathtt{ST})$ operations.*

A proof is given in Section S2 in the Supplementary Material. If the number of observations per subsequence $\mathtt{R}$ is kept fixed, the computational cost of each constrained integrator step therefore scales linearly with in both the number of time steps per observation $\mathtt{S}$ and the number of observation times $\mathtt{T}$.

6 | RELATED WORK

Our approach follows the general framework described in Graham and Storkey (2017) for performing inference in generative models where the simulated observations can be computed as a differentiable function of a random vector with a known prior distribution. As in this work, Graham and Storkey (2017) suggest using a constrained HMC algorithm to target the manifold-supported posterior distribution arising in such a setting, and consider a diffusion model with high-frequency noiseless observations of the full state as an example. In this setting with $\mathtt{S} = 1$, the constraint Jacobian was observed to have a structure allowing a $\mathcal{O}(\mathtt{T}^2)$ cost implementation of the operations required for each constrained integrator step.

Here we make several important extensions to the framework of Graham and Storkey (2017), with the scheme proposed in Section 5.1 allowing efficient $\mathcal{O}(\mathtt{ST})$ cost constrained integrator steps irrespective of the observation regime (high- or low-frequency, partial or full, with or without noise) and the use of a constrained integrator based on a Gaussian splitting as proposed in Section 4.3 giving improved mixing performance as the time-discretisation is refined ($\mathtt{S}$ increased). Further by integrating the constrained integrator into an adaptive HMC algorithm (Hoffman & Gelman, 2014) we eliminate the need to tune the integrator step size and number of integrator steps per trajectory, giving a more automated inference procedure.

The non-centred parametrisation of the diffusion generative model described in Section 3.1 has similarities to the *innovation scheme* of Chib et al. (2004), and its later extension in Golightly and Wilkinson (2008), which recognises for the specific case of an Euler–Maruyama discretisation, that the state sequence $\mathbf{x}_{1:\mathtt{ST}}$ can be computed as a function of the model parameter $\mathbf{z}$, initial state $\mathbf{x}_0$ and increments of the driving Brownian motion process. This relationship can be inverted to compute the increments given $\mathbf{x}_{0:\mathtt{ST}}$ and $\mathbf{z}$. By performing a Metropolis-within-Gibbs update to $\mathbf{z}$ conditioning on the increments and observations, the degeneracy in the conditional distribution of parameters of the drift coefficient when instead conditioning on $\mathbf{x}_{1:\mathtt{ST}}$ as $\mathtt{S} \to \infty$ is avoided, thus producing an algorithm respecting the Roberts–Stramer critique. Our approach generalizes this idea beyond the Euler–Maruyama case by allowing for a generic forward operator $\boldsymbol{f}_\delta$, and jointly updated all latent variables under this reparametrisation rather than using it to only update the parameters.

The conditioning scheme proposed in Section 5.1 is similar in spirit to *blocking* schemes proposed previously in MCMC methods for inference in partially observed time series models, see for example Shephard and Pitt (1997), Golightly and Wilkinson (2008) and Mider et al. (2020); however, the implementation and motivation of the approach here both differ. In the blocking schemes, conditioning on intermediate states introduces conditional independencies allowing proposing updates to blocks of the latent path given fixed parameters in a Metropolis-within-Gibbs type update, with a separate update to the parameters. Here we jointly update the parameters and latent path, and use the conditioning to induce structure in the constraint Jacobian which can be used to reduce the cost of the constrained integrator.

Hypoelliptic diffusions have a rank-deficient diffusion matrix $B(\mathbf{x}, \mathbf{z})B(\mathbf{x}, \mathbf{z})^\mathsf{T}$, but still have transition kernels κ_τ with smooth densities with respect to the Lebesgue measure due to the propagation of noise to all state components via the drift function. Prior work on the calibration of such models has often adopted a maximum likelihood approach, in the setting of high-frequency observations, see for example Ditlevsen and Samson (2019) and the references therein. The singularity of the Wiener noise increment covariance matrix when discretising using an Euler–Maruyama scheme can be avoided via the use of a higher-order discretisation scheme: Ditlevsen and Samson (2019) use a strong order 1.5 Taylor scheme to obtain consistency in the estimation of parameters in both the drift and diffusion coefficient functions.

In terms of our criteria, in Remark 1, to our knowledge there is currently no alternative algorithm that satisfies them all for noiselessly observed hypoelliptic systems. The *guided proposals* framework (Bierkens et al., 2020; van der Meulen & Schauer, 2018; Mider et al., 2021) comes close, as it allows for Bayesian inference in both elliptic and hypoelliptic systems, fully or partially observed with noise or with noiseless observations and a linear observation function $\hbar(\boldsymbol{x}) = L\boldsymbol{x}$, and respects the Roberts–Stramer critique. The approach however does not allow for non-linear noiseless observations, and the methodology requires choosing a tractable auxiliary process used to construct the proposed updates to the latent path given observations and parameters, with the original and auxiliary processes needing to satisfy *matching conditions* on their drift and diffusion coefficients, which can be non-trivial—for example, when the diffusion coefficient is state dependent—hindering the automation of the methodology. In contrast, our method can be applied without change to both hypoelliptic and elliptic diffusions.

A long line of previous work has considered MCMC methods for performing inference in distributions on non-Euclidean spaces, particularly prominent being the influential paper Girolami and Calderhead (2011) where the latent space is equipped with a user-defined Riemannian metric which facilitates local rescaling of the posterior distribution across different directions. Related algorithms have also been proposed based for finite-dimensional projections of distributions with densities with respect to Gaussian measures on Hilbert spaces (Beskos, 2014; Beskos et al., 2017).

In our case, the manifold structure arises directly from the observational constraints placed on the latent space of a generative model and the manifold is extrinsically defined by its embedding in an ambient latent space. Rather than the non-trivial task of selecting a positive-definite matrix valued function to define a Riemannian metric on the latent space, our method only requires the user to choose a matrix representing the fixed metric on the ambient space. As discussed in Section 4.4, we find the prior precision matrix to be a good default in practice.

7 | NUMERICAL EXAMPLES

To demonstrate the flexibility and efficiency of our proposed approach we now present the results of numerical experiments in a range of different settings: hypoelliptic and elliptic systems, simulated and real data, noiseless and noisy observations. In all cases, we use the same methodology, as described in the preceding sections, for performing inference, and where possible we compare to alternative approaches.

7.1 | FitzHugh–Nagumo model with noiseless observations

As a first example, we consider a stochastic-variant of the FitzHugh–Nagumo model (FitzHugh, 1961; Nagumo et al., 1962), a simplified description of the dynamics of action potential generation within an neuronal axon. Following Ditlevsen and Samson (2019) we formulate the model as a $\mathtt{X} = 2$ dimensional hypoelliptic diffusion process $\mathbf{x}$ with drift function $\boldsymbol{a}(\boldsymbol{x}, \boldsymbol{z}) = [\frac{1}{\epsilon}(x_1 - x_1^3 - x_2); \gamma x_1 - x_2 + \beta]$ and diffusion coefficient operator $B(\boldsymbol{x}, \boldsymbol{z}) = [0; \sigma]$ where the components of the $\mathtt{Z} = 4$ dimensional parameter vector are $\mathbf{z} = [\sigma; \epsilon; \gamma; \beta]$ and the Wiener process $\mathbf{b}$ has dimension $\mathtt{B} = 1$. We initially assume the $\mathtt{Y} = 1$ dimensional observations $\mathbf{y}_{1:\mathtt{T}}$ correspond to noiseless observation of the first state component, that is $\hbar(\boldsymbol{x}) = x_1$, with an interobservation time interval $\Delta = \frac{1}{5}$. Further details of the discretisation and priors used are given in the Section S10 in the Supplementary Material.

To measure sampling efficiency in the experiments we use two complementary metrics: the average computation time per constrained integrator step $\hat{\tau}_{\text{step}}$ and the estimated computation time per effective sample $\hat{\tau}_{\text{eff}}$, that is the total chain computation time divided by the estimated effective sample size (ESS) for the chain for each parameter. Proposition 6 closely relates to $\hat{\tau}_{\text{step}}$, and so by estimating how this quantity varies with $\mathtt{R}$, $\mathtt{S}$ and $\mathtt{T}$ we can empirically test whether the proposed scaling holds in practice. While our analysis only considers the computational cost of the constrained integrator, ultimately we are interested in the overall sampling efficiency, which also depends on the mixing performance of the chains; by measuring $\hat{\tau}_{\text{eff}}$ we therefore also gain insight into how our approach performs on this metric. In order to empirically verify that Assumption 6 holds in practice we additionally record the average number of Newton iterations per constrained integrator step (averaged over both forward and time-reversed Ξ^{h_2} calls) which we denote $\overline{\mathtt{n}}$.

The ESS estimates were computed using the Python package *ArviZ* (Kumar et al., 2019) using the rank-normalisation approach proposed by Vehtari et al. (2019). The chain computation times were measured by counting the calls of the key expensive operations in Algorithm 1 and separately timing the execution of these operations—details are given in Section S13 in the Supplementary Material. The Python package *JAX* (Bradbury et al., 2018) was used to allow automatic computation of the derivatives of model functions and all plots were produced using the Python package *Matplotlib* (Hunter, 2007).

For all experiments, we use chains which alternate between Markov transitions which condition on the states at observation time indices $\{\mathtt{t}_\mathtt{b} : \mathtt{bR}\ \forall \mathtt{b} \in 1{:}\mathtt{B}\}$ and $\{\mathtt{t}_\mathtt{b} : \lfloor\frac{(2\mathtt{b}-1)\mathtt{R}}{2}\rfloor\ \ \forall \mathtt{b} \in 1{:}\mathtt{B}\} \cup \{\mathtt{T}\}$ with $\mathtt{B} = \mathtt{T}/\mathtt{R}$. For the experiments in this subsection, we ran all chains with constrained integrators using both the Gaussian and Störmer–Verlet splittings to allow comparison of their relative performance. We use the parameter values $\sigma = 0.3$, $\epsilon = 0.1$, $\gamma = 1.5$ and $\beta = 0.8$ and initial state $\boldsymbol{x}_0 = [-0.5; 0.2]$ to generate the simulated data $\boldsymbol{y}_{1:\mathtt{T}}$ for all experiments.

To allow measuring how performance of our approach varies with $\mathtt{R}$, $\mathtt{S}$ and $\mathtt{T}$, we ran experiments over a grid values for each of these parameters with the other two kept fixed, specifically: $\mathtt{R} \in \{2, 5, 10, 20, 50, 100\}$ with $\mathtt{S} = 25$ and $\mathtt{T} = 100$, $\mathtt{S} \in \{25, 50, 100, 200, 400\}$ with $\mathtt{R} = 5$ and $\mathtt{T} = 100$, $\mathtt{T} \in \{25, 50, 100, 200, 400\}$ with $\mathtt{R} = 5$ and $\mathtt{S} = 25$. For all $(\mathtt{R}, \mathtt{S}, \mathtt{T})$ values and splittings tested we ran three sets of four chains of 1250 iterations each with independent initialisations (details of the initializations are given in Section S12 in the Supplementary Material), with the first 250 iterations of each set of four chains an adaptive warm-up phase used to tune the integrator step-size t, with the samples from these warm-up iterations omitted from the ESS estimates but included in the computation time estimates. For all sets of chains, the split-$\hat{R}$ convergence diagnostic values computed from the (non-warm-up iterations of the) four chains for all parameter values using rank-normalisation and folding were less than 1.01 as recommended in Vehtari et al. (2019). A dynamic HMC implementation (Betancourt, 2017) was used to set the number of integrator steps per trajectory in each transition. A summary of all the algorithmic parameter values used in the numerical experiments is given in Section S14 in the Supplementary Material.

The top panels in Figure 1 show how the number of Newton iterations required to solve (12) in each of the forward and reverse Ξ^{h_2} steps, averaged across the chains, varies with $\mathtt{R}$, $\mathtt{S}$ and $\mathtt{T}$

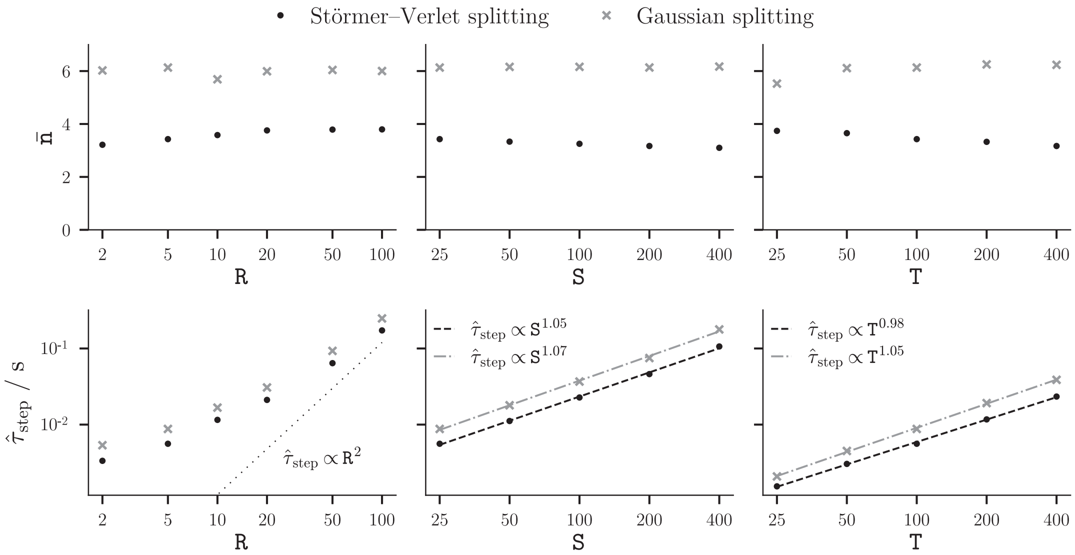

FIGURE 1 *FitzHugh–Nagumo model (noiseless observations)*: Average number of Newton iterations per integrator step $\bar{n}$ (top) and computation time per integrator step $\hat{\tau}_{step}$ in seconds (bottom) for varying R, S and T. The markers show the median across three independent runs. The dashed lines in the bottom right two plots show a log-domain least squares fit to the medians for each splitting

and for the two different Hamiltonian splittings. We see that for a given splitting, the number of Newton iterations is close to constant in all cases, providing empirical support for Assumptions 4 and 6. The bottom panels in Figure 1 instead show the average time per integrator step $\hat{\tau}_{step}$ varies with R, S and T for both splittings. We see that the log-domain least-square fits show a very close to linear scaling of $\hat{\tau}_{step}$ with both S and T, verifying these aspects of the $\mathcal{O}(R^2ST)$ scaling claimed in Proposition 6. The growth of $\hat{\tau}_{step}$ with R over the range here is sub-quadratic (the dotted line shows a quadratic trend for reference), however there is visible acceleration in the growth. An inspection of a breakdown of the total computation time spent on different individual operations revealed that for smaller R the $\mathcal{O}(RST)$ computation of the constraint Jacobian is dominating, with the $\mathcal{O}(R^2ST)$ linear algebra operations only becoming significant for larger R.

Figure 2 shows how the estimated computation time per effective sample $\hat{\tau}_{eff}$ varies with each of R, S and T, for each of the four model parameters and for each of the two Hamiltonian splittings. First considering the results for varying number of observations per subsequence R we see the efficiency is maximised ($\hat{\tau}_{eff}$ minimised) for both splittings for an intermediate value of $R \approx 5$, with a small drop-off in efficiency for $R = 2$ and a larger decrease in efficiency as R is increased beyond 5. This reflects the competing effects of the reduced cost of each constrained integrator step as R is made smaller versus the reduced chain mixing performance in each transition for smaller R due to the extra states being (artificially) conditioned on. Importantly we see, however, that the latter effect is less significant (in this model at least), meaning that performance is still close to optimal for $R = 2$, suggesting performance will not be too adversely effected if a too small R value is chosen.

Now turning our attention to the plots of $\hat{\tau}_{eff}$ versus the number of discrete time steps per inter-observation interval S, we see that there is a clear difference in the scaling of $\hat{\tau}_{eff}$ with S for the two Hamiltonian splittings, with the Gaussian splitting giving a only slightly above linear scaling across all four parameters with exponents in the range 1.06–1.10 compared to 1.14–1.44

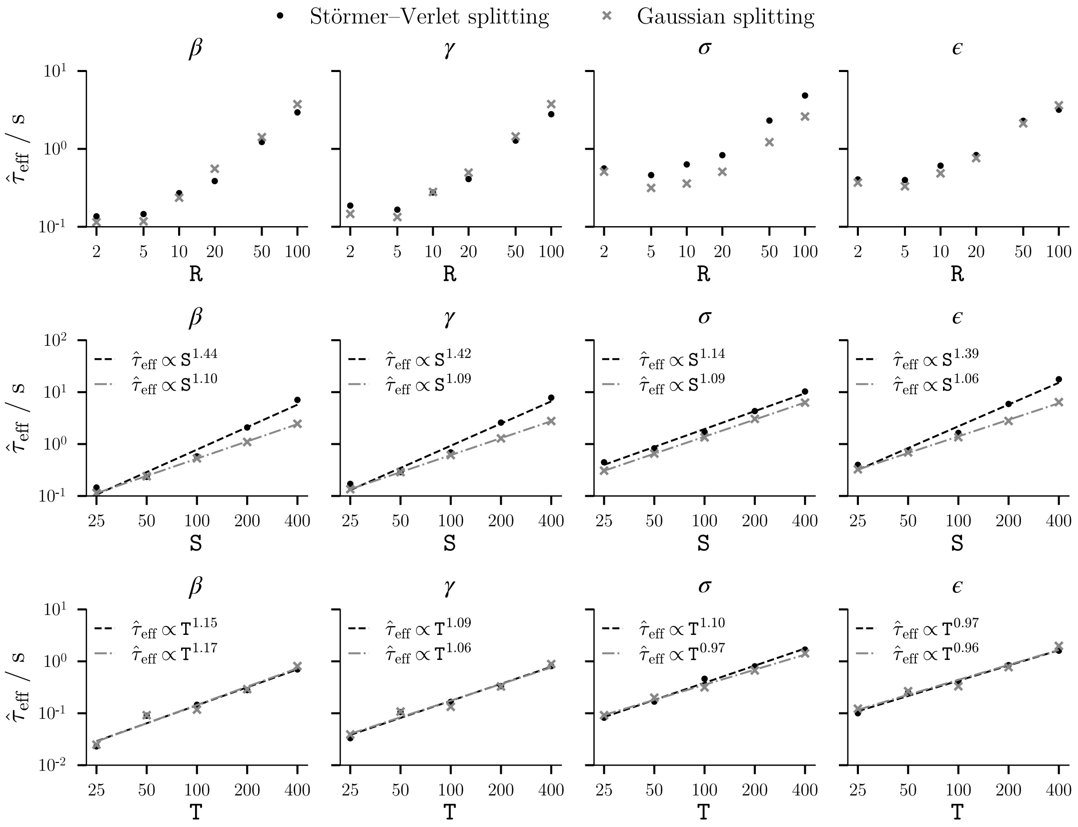

FIGURE 2 *FitzHugh–Nagumo model (noiseless observations):* Computation time per effective sample $\hat{\tau}_{\text{eff}}$ in seconds for varying R, S and T for each model parameter, in all cases on a log-log scale. The markers shows the median across three independent runs. The dashed lines in the plots in bottom two rows show log-domain least-square fits to the medians for each splitting

for the Störmer–Verlet splitting. Inspection of the integrator step sizes t (not shown), which were adaptively tuned in a warm-up phase to control the average acceptance statistic for the chains to be fixed, reveals that for the Gaussian splitting the step size t is in the range $t = 0.29 \pm 0.01$ for all S while for the Störmer–Verlet splitting shows a decrease from $t = 0.20$ for S = 25 to $t = 0.10$ for S = 400, consistent with results suggesting that the step size of the Störmer–Verlet integrator needs to scale with $\mathtt{Q}^{-1/4}$ to maintain a constant accept probability of a static integration time HMC algorithm in the unconstrained setting (Beskos et al., 2013b; Neal, 2011), compared to a dimension-free dependence of the acceptance probability on t for integrators using the Gaussian splitting in appropriate targets (Beskos et al., 2011). While we have emphasised here the superior performance of the Gaussian splitting, we note that the growth of $\hat{\tau}_{\text{eff}}$ with S for both methods is very favourable, and shows our approach is able to remain efficient for fine time discretisations of the continuous time model.

Finally, we consider the bottom row of plots in Figure 2, showing how $\hat{\tau}_{\text{eff}}$ varies with the number of observation times T for each model parameter. We see that in this case both splittings give very similar scalings, with a close to linear growth in $\hat{\tau}_{\text{eff}}$ with T for all four parameters. The (infinite-dimensional) target posterior being approximated for each T value differs here unlike the case for varying R and S), in particular becoming more concentrated as T increases. The increase in $\hat{\tau}_{\text{eff}}$ with $\mathtt{T}$ seems to be largely attributable to the increase in the computational cost of each constrained integrator step with $\mathtt{T}$, and so the mixing performance of the chains seems to be largely independent of $\mathtt{T}$. This suggests that the constrained HMC algorithm is able to efficiently explore posterior distributions with varying geometries. While here the concentration of the posterior is due to an increasing number of observations, in the following section we will see that our approach is also robust to varying informativeness of the individual observations.

7.2 | FitzHugh–Nagumo model with additive observation noise

As a second example we consider the same hypoelliptic diffusion model as in the preceding section, but now with observations subject to additive Gaussian noise of standard deviation σ_y, that is $\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{z}, w) = x_1 + \sigma_y w$ and $\eta = \mathcal{N}(0, 1)$. The presence of additive observation noise means that the posterior on $(\mathbf{u}, \mathbf{v}_{0:\mathtt{ST}})$ given $\mathbf{y}_{1:\mathtt{T}} = \boldsymbol{y}_{1:\mathtt{T}}$ has a tractable Lebesgue density. We therefore compare our constrained HMC approach to running a standard (unconstrained) HMC algorithm targeting the posterior on $(\mathbf{u}, \mathbf{v}_{0:\mathtt{ST}})$ with details of the posterior density and HMC algorithm used given in Section S9 in the Supplementary Material. As a further baseline, we also compare to the approach of Mider et al., 2021), which uses a Metropolis-within-Gibbs scheme alternating *guided proposal* Metropolis–Hastings updates to the latent path $\mathbf{x}_{0:\mathtt{ST}}$ given parameters $\mathbf{z}$, with random-walk Metropolis (RWM) updates to the parameters $\mathbf{z}$ given the path $\mathbf{x}_{0:\mathtt{ST}}$. An application of this approach to the FitzHugh–Nagumo model considered here is described in van der Meulen et al. (2020), and we use the Julia code accompanying that article to run the experiments.

We use the same priors and time discretisation as in the previous section, and fix $\mathtt{S} = 40$ and $\mathtt{T} = 100$. Simulated observed sequences $\boldsymbol{y}_{1:\mathtt{T}}$ were generated for each of the observation noise variances $\sigma_y^2 \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$. In all cases, $\boldsymbol{y}_{1:\mathtt{T}}$ was generated using the same parameters and initial state as in the previous section and sharing the same values for $\mathbf{v}_{1:\mathtt{ST}}$ and $\mathbf{w}_{1:\mathtt{T}}$ (sampled from their standard normal priors). Chains targeting the resulting posteriors were run for each σ_y^2 value and for each of the three MCMC methods being considered.

For our constrained HMC algorithm, we used $\mathtt{R} = 5$ and ran chains using a constrained integrator based on the Störmer–Verlet splitting, with results instead using the Gaussian splitting showing a similar pattern of performance and hence omitted here to avoid duplication. For the standard HMC algorithm, a diagonal metric matrix representation M was adaptively tuned in the warm-up iterations with this found to uniformly outperform using a fixed identity matrix for all σ_y values tested here. For both the standard and constrained HMC algorithms, we run four chains of 3000 iterations with the first 500 iterations an adaptive warm-up phase used to tune the integrator step-size t (and M for the standard HMC case). For the guided proposals/RWM case, we ran four chains of 3×10^5 iterations, with the first 5×10^4 iterations an adaptive warm-up phase where the persistence parameter of the guided proposals update to $\mathbf{x}_{1:\mathtt{ST}} \mid \mathbf{z}$ and step sizes of the random-walk proposals for the update to $\mathbf{z} \mid \mathbf{x}_{1:\mathtt{ST}}$ were adapted as described in Mider et al., 2021).

Estimated computation time per effective sample $\hat{\tau}_{\text{eff}}$ values were calculated for the chains of each of the three MCMC methods and each of the observation noise variance values σ_y^2. The ESS estimates were calculated as described in the preceding section, however, the true total wall-clock run times were used for the chain computation times here due the difficulty in ensuring a consistent treatment of different MCMC algorithms in the approach used in the previous section. To ensure as fair a comparison as possible all chains were run on the same computer and limited to

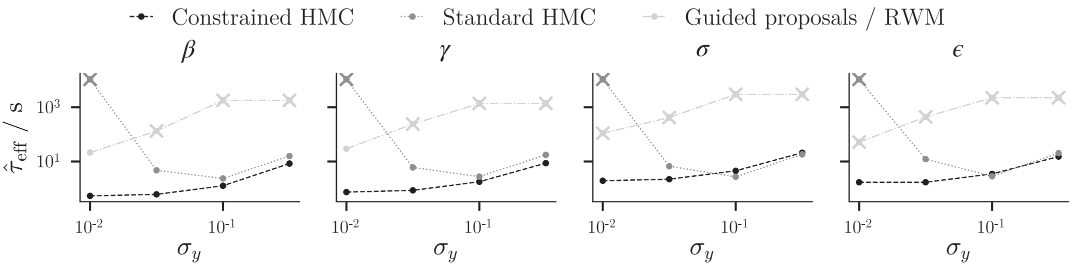

FIGURE 3 *FitzHugh–Nagumo model (noisy observations)*: Computation time per effective sample $\hat{\tau}_{\text{eff}}$ for varying observation noise standard deviation σ_y for each model parameter, in all cases on a log–log scale. Points with cross markers indicate chains with an estimated split- $\hat{R}$ value of greater than 1.01, indicative of non-convergence

a single processor core to avoid differences due to varying exploitation of parallel computation in the implementations.

Figure 3 summarises the results. We first note that despite the large number of iterations used for the guided proposals/RWM chains (3×10^5), the per-parameter split- $\hat{R}$ diagnostics (Vehtari et al., 2019) for the chains indicated non-convergence of the chains in nearly all cases, with only the chains for $\sigma_y = 10^{-2}$ appearing to be approaching convergence with $\hat{R}$ values in the range [1.01, 1.05] compared to $\hat{R}$ values in the range [1.39, 1.87] for $\sigma_y = 10^{-0.5}$. The poor convergence here seems to be at least in part due to the difficulty in finding globally appropriate values of the RWM step sizes, with the step sizes still changing significantly in the final iterations of the warm-up phase and the final adapted values differing significantly across chains for the same σ_y.

Given the poor convergence the estimated ESS values must be treated with some caution, however even for the $\sigma_y = 10^{-2}$ case where the chains appeared to be closest to convergence the estimated $\hat{\tau}_{\text{eff}}$ values for the guided proposals/RWM chains are between 30 and 80 times larger than the corresponding values for the constrained HMC chains. As Julia implementations of numerical algorithms generally significantly outperform Python equivalents (Bezanson et al., 2017), the superior sampling performance of the (Python) constrained HMC implementation compared to the (Julia) guided proposals/RWM implementation here seems unlikely to be just due to differences in the efficiency of the implementations, but rather reflects significantly improved mixing of the joint gradient-informed updates to the latent variables by the constrained HMC algorithm, compared to the non-gradient-informed Metropolis-within-Gibbs updates of the guided proposals/RWM algorithm.

Comparing now the results for the standard and constrained HMC algorithms, we see that while both algorithms perform similarly for larger σ_y values (i.e. less informative observations), the constrained HMC algorithm provides significantly better sampling efficiency for smaller σ_y values. Inspecting the integrator step size t set at the end of the adaptive warm-up phase for each of σ_y values reveals that, while for constrained HMC all step sizes t fall in the range 0.17–0.18 and so seem invariant to σ_y, for the standard HMC chains, t ranges from 5.1×10^{-4} for $\sigma_y = 10^{-2}$ to 9.1×10^{-3} for $\sigma_y = 10^{-1}$, resulting in a need to take more integrator steps per transition to make moves of the same distance in the latent space and hence a decreasing sampling efficiency as σ_y becomes smaller.

The results for $\sigma_y = 10^{-0.5}$ break the trend of increasing σ_y leading to increased efficiency for the standard HMC chains, with a significant increase in $\hat{\tau}_{\text{eff}}$ compared to $\sigma_y = 10^{-1}$. This seems to be due to a roughly halving of the integrator step size t set in the adaptive warm-up phase to 5.2×10^{-3} for $\sigma_y = 10^{-0.5}$, which, combined with the more diffuse posterior for the larger σ_y value, led to a significant increase in the average number of integrator steps per transition and so computational cost per effective sample. A potential explanation for the decrease in the adapted step size is that the more diffuse posterior extends to regions where the posterior density has higher curvature necessitating a smaller step size to control the Hamiltonian error. In contrast, for the constrained HMC chains, the Hamiltonian error is controlled with a close to constant step size for all σ_y; however, there is a drop in efficiency as σ_y becomes larger, which seems to be due to the more diffuse posterior requiring a greater number of integrator steps to explore and so higher computational cost per effective sample on average.

7.3 | Susceptible-infected-recovered model with additive observation noise

As a final example, we perform inference in an epidemiological compartmental model given real observations of the time course of the number of infected patients in an influenza outbreak in a boarding school (Anonymous, 1978). Specifically we consider a diffusion approximation of a susceptible-infected-recovered (SIR) model (see e.g. the derivation in Fuchs, 2013 section 5.1.3), with a time-varying contact rate parameter itself modelled as a diffusion process as proposed in Ryder et al. (2018), resulting in the following three-dimensional elliptic SDE system

$$\begin{bmatrix} \mathsf{ds} \\ \mathsf{di} \\ \mathsf{dc} \end{bmatrix} = \begin{bmatrix} -N^{-1}\mathsf{csi} \\ N^{-1}\mathsf{csi} - \gamma\mathsf{i} \\ \left(\alpha(\beta - \log \mathsf{c}) + \frac{\sigma^2}{2}\right)\mathsf{c} \end{bmatrix} \mathrm{d}\tau + \begin{bmatrix} \sqrt{N^{-1}\mathsf{csi}} & 0 & 0 \\ -\sqrt{N^{-1}\mathsf{csi}} & \sqrt{\gamma\mathsf{i}} & 0 \\ 0 & 0 & \sigma\mathsf{c} \end{bmatrix} \begin{bmatrix} \mathrm{d}\mathbf{w}_1 \\ \mathrm{d}\mathbf{w}_2 \\ \mathrm{d}\mathbf{w}_3 \end{bmatrix}$$

where τ is the time in days, s and i are the number of susceptible and infected individuals respectively, c the contact rate, N the population size and γ the recovery rate parameter. The SDE for c arises from $\log \mathsf{c}$ following an Ornstein–Uhlenbeck process with reversion rate α, long term mean β and instantaneous volatility σ.

As each of s, i and c represent positive-valued quantities, the diffusion state is defined to be $\mathbf{x} = [\log \mathsf{s}; \log \mathsf{i}; \log \mathsf{c}] \in \mathbb{R}^3$ with drift $\boldsymbol{a}$ and diffusion coefficient B functions derived from the above SDEs via Itô's lemma. By computing the time-discretisation in this log-transformed space, the positivity of s, i and c is enforced and the numerical issues arising when evaluating the square-root terms in the diffusion coefficient for negative s, i or c are avoided. The observed data $\boldsymbol{y}_{1:\mathtt{T}}$ corresponds to measurements of the number of infected individuals $\mathsf{i} = \exp(\mathbf{x}_2)$ at daily intervals, i.e. $\Delta = 1$, over a period of $\mathtt{T} = 14$ days, with the observations assumed to be subject to additive noise of unknown standard deviation σ_y, that is $\mathbf{y}_\mathtt{t} = \exp(\mathbf{x}_2(\mathtt{t})) + \sigma_y \mathbf{w}_\mathtt{t}$. The $\mathtt{Z} = 5$ dimensional parameter vector is then $\mathbf{z} = [\gamma; \alpha; \beta; \sigma; \sigma_y]$. Details of the priors and discretisation used are given in Section S11 in the Supplementary Material.

We compare the performance of our proposed constrained HMC approach to a standard HMC algorithm, with the noise in the observations meaning that the posterior on $\mathbf{u}$ and $\mathbf{v}_{0:\mathtt{ST}}$ admits a Lebesgue density. For each algorithm, we run four chains of 3000 iterations with the first 500 iterations an adaptive warm-up phase. For our constrained HMC algorithm due to the small number of, and high correlations between, the observations we do not introduce any artificial conditioning on intermediate states, that is $\mathtt{R} = \mathtt{T} = 14$. Chains using constrained integrators based on both the Störmer–Verlet and Gaussian splitting show very similar performance here so we show only

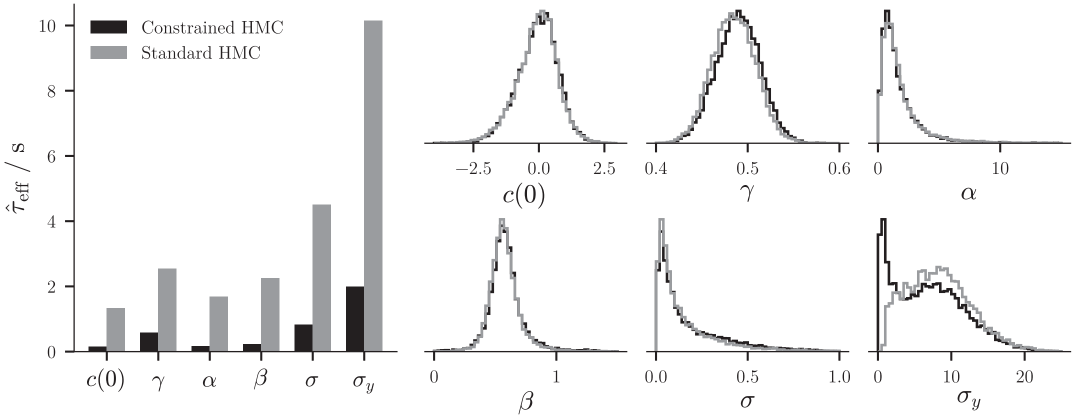

FIGURE 4 *SIR model*: Computation time per effective sample $\hat{\tau}_{\text{eff}}$ (leftmost panel) and estimated posterior marginals (right panels) for each model parameter computed using constrained (black) and standard (dark gray) Hamiltonian Monte Carlo chain samples

the Störmer–Verlet case to avoid duplication. For the standard HMC algorithm, a diagonal metric matrix representation M was adaptively tuned in the warm-up iterations. The estimated computation time per effective sample $\hat{\tau}_{\text{eff}}$ values were calculated using the total wall-clock run times for the chains.

The results are summarised in Figure 4. The left plot shows the estimated per-parameter $\hat{\tau}_{\text{eff}}$ values for each of the two MCMC methods: we see that the constrained HMC algorithm is able to give significantly improved sampling efficiency over standard HMC here. More importantly, however, it appears that the standard HMC algorithm is in fact failing to explore the full posterior. The grid of six plots in the right of Figure 4 shows the estimated posterior marginals for each parameter computed from either the constrained or standard HMC chain samples. There is a clear discrepancy in the estimated posterior marginal of σ_y between the two methods, with the standard HMC chains having many fewer samples at smaller σ_y values compared to the constrained HMC chains. Figure S2 in the Supplementary Material shows the corresponding estimated pairwise marginals with σ_y on a log-scale, with the poorer coverage of small σ_y values by the standard HMC chains more apparent.

While the per-parameter $\hat{R}$ diagnostics for the standard HMC chains are all below 1.01, some hint of the underlying issue being encountered here is given by the high number of iterations in which the integration of the Hamiltonian dynamics diverged for the standard HMC chains–roughly 4% of the non-warm-up iterations for each chain. Such divergences are indicative of the presence of regions of high curvature in the posterior distribution that result in the numerical simulation of the Hamiltonian trajectories becoming unstable, and in some cases may be ameliorated by use of a smaller integrator step size t (Betancourt, 2017).

Here specifically the adaptive tuning of the step size t in the warm-up phase has led to a step size which is too large for exploring the regions of the posterior in which σ_y is small. Although by setting a higher target acceptance statistics for the step size adaptation algorithm or hand tuning t to a smaller value we could potentially fix this issue, this would be at the cost of an associated decrease in sampling efficiency, leading to even poorer performance relative to the constrained HMC chains. As seen in the results for the FitzHugh–Nagumo model in Section 7.2, if σ_y is fixed the integrator step size t for the standard HMC algorithm needs to be decreased as σ_y is decreased to control the acceptance rate resulting in a higher computation cost per effective sample—Figure S1 in the Supplementary Material illustrates this directly for the SIR model. When σ_y instead is unknown as here, standard HMC needs to use a step size appropriate for the smallest σ_y value 'typical' under the posterior, which if σ_y is poorly informed by the data (as is the case for this model) can require using very small integrator step sizes t. In contrast as the constrained HMC algorithm is able to use an integrator step size t which is independent of σ_y, the sampling efficiency of the chains is not limited by the need to use a small step size t to explore regions of the posterior in which σ_y is small.

8 | CONCLUSIONS AND FURTHER DIRECTIONS

We have introduced a methodology for calibrating a wide class of diffusion-driven models. Our approach is based on recasting the inferential problem as one of exploring a posterior supported on a manifold, the structure of the latter determined by the observational constraints on the generative model. Once this viewpoint is adopted, available techniques from the literature on constrained HMC can be called upon to allow for effective traversing of the high-dimensional latent space. We have further shown that the Markovian structure of the model can be exploited to design a methodology with computational complexity that scales linearly with both the resolution of the time-discretisation and the number of observation times.

A critical argument put forward via the methodology developed in this work is that practitioners working with SDE models are now provided with the option to refer to a *single* and *highly automated*, algorithmic framework for Bayesian calibration of their models. This algorithmic framework employs efficient Hamiltonian dynamics and adheres to all sought out criteria listed in Remark 1.

When exploring distributions with rapidly varying curvatures, standard HMC methods with a fixed step size can yield trajectories that either require too small of a step size (as in the FitzHugh–Nagumo model with noise in Section 7.2), or become unstable and diverge if the step size is not small enough in areas of high curvature of the posterior on the latent space (as with our SIR example in Section 7.3 where variations in the scale parameter σ_y have strong effect on the curvature). In both cases, particularly strong effects can render standard HMC non-operational (as in the SIR case). Although the methodology presented in Girolami and Calderhead (2011) can in principle be helpful in such contexts, this class of algorithms is intrinsically constructed to induce good performances in the centre of the target distribution as it involves an expectation over the data, and not the *given* data, for the specification of the employed Riemannian metric. Constrained HMC dynamics can provide a more appropriate approach for dealing with rapidly varying curvature across the whole of the support of the target distribution. When combined with efficient discretisations of the dynamics—as in the case of the class of diffusion models we have studied in this work—they can provide statistically efficient methods.

The viewpoint adopted in this paper is potentially relevant to a larger class of stochastic models for time series (e.g. random ordinary differential equations), as well as other Markovian model classes (e.g. Markov networks). Some of the authors are currently involved in applying such MCMC methods to Bayesian inverse problems; manifold structures naturally appear in the low noise regime (Beskos et al., 2018). In general, we believe that the approach presented in this paper warrants further investigation, with a corresponding study of critical algorithmic aspects, for example computational complexity and mixing properties.

ACKNOWLEDGEMENTS

MMG acknowledges support from the Lloyd's Register Foundation programme on data-centric engineering at the Alan Turing Institute. AHT and MMG acknowledge support from the Singapore Ministry of Education Tier 2 (MOE2016-T2-2-135) and a Young Investigator Award Grant (NUSYIA FY16 P16; R-155-000-180-133). AB acknowledges support from the Leverhulme Trust Prize. This research made use of the Rocket High Performance Computing service at Newcastle University, UK.

ORCID

Matthew M. Graham http://orcid.org/0000-0001-9104-7960

REFERENCES

Andersen, H.C. (1983) RATTLE: a 'velocity' version of the SHAKE algorithm for molecular dynamics calculations. *Journal of Computational Physics*, 52, 24–34.

Anonymous (1978) News and Notes: influenza in a boarding school. *British Medical Journal*, 1, 586–590. Available from: https://www.bmj.com/content/1/6112/586

Arnol'd, V.I. (2013) *Mathematical methods of classical mechanics*, vol. 60. Berlin: Springer Science & Business Media.

Barp, A.A. (2020) The bracket geometry of statistics. Ph.D. thesis.

Beskos, A. (2014) A stable manifold MCMC method for high dimensions. *Statistics & Probability Letters*, 90, 46–52.

Beskos, A., Pinski, F.J., Sanz-Serna, J.M. & Stuart, A.M. (2011) Hybrid Monte Carlo on Hilbert spaces. *Stochastic Processes and their Applications*, 121, 2201–2230.

Beskos, A., Kalogeropoulos, K. & Pazos, E. (2013a) Advanced MCMC methods for sampling on diffusion pathspace. *Stochastic Processes and their Applications*, 123, 1415–1453.

Beskos, A., Pillai, N., Roberts, G., Sanz-Serna, J.-M. & Stuart, A. (2013b) Optimal tuning of the hybrid Monte Carlo algorithm. *Bernoulli*, 19, 1501–1534.

Beskos, A., Girolami, M., Lan, S., Farrell, P.E. & Stuart, A.M. (2017) Geometric MCMC for infinite-dimensional inverse problems. *Journal of Computational Physics*, 335, 327–351.

Beskos, A., Roberts, G., Thiery, A. & Pillai, N. (2018) Asymptotic analysis of the random walk Metropolis algorithm on ridged densities. *The Annals of Applied Probability*, 28, 2966–3001.

Betancourt, M. (2017) A conceptual introduction to Hamiltonian Monte Carlo*arXiv preprint 1701.02434*.

Betancourt, M. & Girolami, M. (2015) Hamiltonian Monte Carlo for hierarchical models. *Current trends in Bayesian methodology with applications*, 79, 30.

Bezanson, J., Edelman, A., Karpinski, S. & Shah, V.B. (2017) Julia: a fresh approach to numerical computing. *SIAM review*, 59, 65–98.

Bierkens, J., van der Meulen, F. & Schauer, M. (2020) Simulation of elliptic and hypo-elliptic conditional diffusions. *Advances in Applied Probability*, 52, 173–212.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M.J., Leary, C., Maclaurin, D.. & Wanderman-Milne, S. (2018) JAX: composable transformations of Python+NumPy programs. Available from: http://github.com/google/jax

Brubaker, M., Salzmann, M. & Urtasun, R. (2012) A family of MCMC methods on implicitly defined manifolds. In: *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, 161–172. Proceedings of Machine Learning Research.

Chib, S., Pitt, M.K. & Shephard, N. (2004) Likelihood based inference for diffusion driven models. *Economics Working Papers W20*, Oxford University.

Diaconis, P., Holmes, S. & Shahshahani, M. (2013) Sampling from a manifold. In: *Advances in modern statistical theory and applications: a Festschrift in honor of Morris L. Eaton*, 102–125. Institute of Mathematical Statistics.

Ditlevsen, S. & Samson, A. (2019) Hypoelliptic diffusions: filtering and inference from complete and partial observations. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 81, 361–384.

Duane, S., Kennedy, A.D., Pendleton, B.J. & Roweth, D. (1987) Hybrid Monte Carlo. *Physics Letters B*, 195, 216–222.

Elerian, O., Chib, S. & Shephard, N. (2001) Likelihood inference for discretely observed nonlinear diffusions. *Econometrica*, 69, 959–993.

FitzHugh, R. (1961) Impulses and physiological states in theoretical models of nerve membrane. *Biophysical Journal*, 1, 445–466.

Fuchs, C. (2013) *Inference for diffusion processes: with applications in life sciences*. Berlin: Springer Science & Business Media.

Girolami, M. & Calderhead, B. (2011) Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73, 123–214.

Golightly, A. & Wilkinson, D.J. (2008) Bayesian inference for nonlinear multivariate diffusion models observed with error. *Computational Statistics & Data Analysis*, 52, 1674–1693.

Graham, M.M. (2019) Mici: Python implementations of manifold MCMC methods. Available from: https://doi.org/10.5281/zenodo.3517301

Graham, M.M. & Storkey, A.J. (2017) Asymptotically exact inference in differentiable generative models. *Electronic Journal of Statistics*, 11, 5105–5164.

Griewank, A. & Walther, A. (2008) *Evaluating derivatives: principles and techniques of algorithmic differentiation*, vol. 105. Philadelphia: Society for Industrial and Applied Mathematics.

Hartmann, C. & Schütte, C. (2005) A constrained hybrid Monte-Carlo algorithm and the problem of calculating the free energy in several variables. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik: Applied Mathematics and Mechanics*, 85, 700–710.

Hoffman, M.D. & Gelman, A. (2014) The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15, 1593–1623.

Holm, D.D., Schmah, T. & Stoica, C. (2009) *Geometric mechanics and symmetry: from finite to infinite dimensions*, vol. 12. Oxford: Oxford University Press.

Hunter, J.D. (2007) Matplotlib: a 2D graphics environment. *Computing in Science & Engineering*, 9, 90–95.

Kloeden, P.E. & Platen, E. (1992) *Numerical Solution of Stochastic Differential Equations, vol. 23 of Stochastic Modelling and Applied Probability*. Berlin Heidelberg: Springer.

Kumar R., Carroll C., Hartikainen A. & Martin O. (2019) ArviZ a unified library for exploratory analysis of Bayesian models in Python. *Journal of Open Source Software*, 4, 1143. https://doi.org/10.21105/joss.01143

Leimkuhler, B. & Matthews, C. (2016) Efficient molecular dynamics using geodesic integration and solvent–solute splitting. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 472, 20160138.

Leimkuhler, B. & Reich, S. (2004) *Simulating Hamiltonian dynamics, vol. 14 of Cambridge monographs on applied and computational mathematics*. Cambridge: Cambridge University Press.

Leimkuhler, B.J. & Skeel, R.D. (1994) Symplectic numerical integrators in constrained Hamiltonian systems. *Journal of Computational Physics*, 112, 117–125.

Lelièvre, T., Rousset, M. & Stoltz, G. (2019) Hybrid Monte Carlo methods for sampling probability measures on submanifolds. *Numerische Mathematik*, 143, 379–421.

van der Meulen, F. & Schauer, M. (2018) Bayesian estimation of incompletely observed diffusions. *Stochastics*, 90, 641–662.

van der Meulen, F., Schauer, M., Grazzi, S., Danisch, S. & Mider, M. (2020) Bayesian inference for SDE models: a case study for an excitable stochastic-dynamical model. Available from: https://nextjournal.com/Lobatto/FitzHugh-Nagumo

Mider, M., Jenkins, P.A., Pollock, M. & Roberts, G.O. (2020) The computational cost of blocking for sampling discretely observed diffusions. *arXiv preprint arXiv:2009.10440*.

Mider M., Schauer M. & van der Meulen F. (2021) Continuous-discrete smoothing of diffusions. *Electronic Journal of Statistics*, 15, https://doi.org/10.1214/21-ejs1894

Mil'shtejn, G. (1975) Approximate integration of stochastic differential equations. *Theory of Probability & Its Applications*, 19, 557–562.

Nagumo, J., Arimoto, S. & Yoshizawa, S. (1962) An active pulse transmission line simulating nerve axon. *Proceedings of the IRE*, 50, 2061–2070.

Neal, R.M. (2011) MCMC using Hamiltonian dynamics. In: *Handbook of Markov Chain Monte Carlo*, London: Chapman and Hall/CRC, pp. 139–188.

Oksendal, B. (2013) *Stochastic differential equations: an introduction with applications*. Berlin: Springer Science & Business Media.

Papaspiliopoulos, O., Roberts, G.O. & Sköld, M. (2003) Non-centered parameterisations for hierarchical models and data augmentation. In: *Bayesian Statistics 7: Proceedings of the Seventh Valencia International Meeting*, 307–326. Oxford University Press, USA.

Papaspiliopoulos O., Roberts G.O. & Sköld M. (2007) A General Framework for the Parametrization of Hierarchical Models. *Statistical Science*, 22, 59–73. https://doi.org/10.1214/088342307000000014

Papaspiliopoulos, O., Roberts, G.O. & Stramer, O. (2013) Data augmentation for diffusions. *Journal of Computational and Graphical Statistics*, 22, 665–688.

Reich, S. (1996) Symplectic integration of constrained Hamiltonian systems by composition methods. *SIAM Journal on Numerical Analysis*, 33, 475–491.

Roberts, G.O. & Stramer, O. (2001) On inference for partially observed nonlinear diffusion models using the Metropolis–Hastings algorithm. *Biometrika*, 88, 603–621.

Rousset, M., Stoltz, G. & Lelièvre, T. (2010) *Free Energy Computations: A Mathematical Perspective*. London: Imperial College Press.

Ryder, T., Golightly, A., McGough, A.S. & Prangle, D. (2018) Black-box variational inference for stochastic differential equations. In: Dy, J. & Krause, A. (Eds.) International Conference on Machine Learning *vol. 80 of* Proceedings of Machine Learning Research, pp. 4423–4432.

Sørensen, M. (2009) Parametric inference for discretely sampled stochastic differential equations. In: *Handbook of financial time series*, Berlin: Springer, 531–553.

Shahbaba, B., Lan, S., Johnson, W.O. & Neal, R.M. (2014) Split Hamiltonian Monte Carlo. *Statistics and Computing*, 24, 339–349.

Shephard, N. & Pitt, M.K. (1997) Likelihood analysis of non-Gaussian measurement time series. *Biometrika*, 84, 653–667.

Vehtari, A., Gelman, A., Simpson, D., Carpenter, B. & Bürkner, P.-C. (2019) Rank-normalization, folding, and localization: An improved $\hat{R}$ for assessing convergence of MCMC. *arXiv preprint 1903.08008*.

Verlet, L. (1967) Computer 'experiment' on classical fluids. i. thermodynamical properties of Lennard–Jones molecules. *Physical Review*, 159, 98.

Zappa, E., Holmes-Cerfon, M. & Goodman, J. (2018) Monte Carlo on manifolds: sampling densities and integrating functions. *Communications on Pure and Applied Mathematics*, 71, 2609–2647.

SUPPORTING INFORMATION

Additional supporting information may be found in the online version of the article at the publisher's website.

How to cite this article: Graham, M.M., Thiery, A.H. & Beskos, A. (2022) Manifold Markov chain Monte Carlo methods for Bayesian inference in diffusion models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 1–28. Available from: https://doi.org/10.1111/rssb.12497