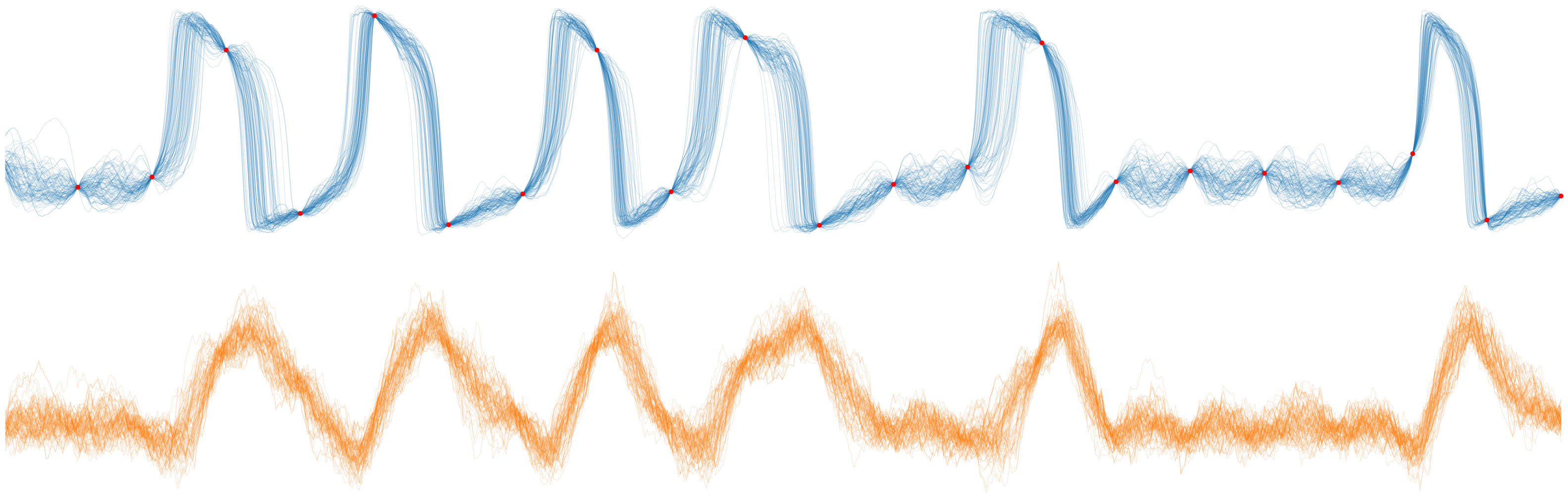
Manifold MCMC methods for inference in diffusion models



Alexandros Beskos, University College London

Joint work with Matt Graham (UCL) and Alexandre Thiery (NUS)

High-level summary

Task: infer the posterior on the parameters of a diffusion given partial observations at `T` times.

High-level summary

Task: infer the posterior on the parameters of a diffusion given partial observations at `T` times.

High-level summary

Task: infer the posterior on the parameters of a diffusion given partial observations at `T` times.

Why this is a challenging problem:

High-level summary

Task: infer the posterior on the parameters of a diffusion given partial observations at `T` times.

Why this is a challenging problem:

- Transition densities generally intractable.

High-level summary

Task: infer the posterior on the parameters of a diffusion given partial observations at `T` times.

Why this is a challenging problem:

- Transition densities generally intractable.
- Necessitates using a time-discretisation.

High-level summary

Task: infer the posterior on the parameters of a diffusion given partial observations at `T` times.

Why this is a challenging problem:

- Transition densities generally intractable.
- Necessitates using a time-discretisation.
- Resulting latent space very high-dimensional.

High-level summary

Task: infer the posterior on the parameters of a diffusion given partial observations at `T` times.

Why this is a challenging problem:

- Transition densities generally intractable.
- Necessitates using a time-discretisation.
- Resulting latent space very high-dimensional.
- Strong dependencies between variables.

Sketch of proposed approach

Sketch of proposed approach

1. Formulate generative model as a differentiable map from latent variables to observations.

Sketch of proposed approach

1. Formulate generative model as a differentiable map from latent variables to observations.
2. Recognise posterior as a distribution with known density on an embedded manifold.

Sketch of proposed approach

1. Formulate generative model as a differentiable map from latent variables to observations.
2. Recognise posterior as a distribution with known density on an embedded manifold.
3. Apply constrained Hamiltonian Monte Carlo method to sample from posterior.

Sketch of proposed approach

1. Formulate generative model as a differentiable map from latent variables to observations.
2. Recognise posterior as a distribution with known density on an embedded manifold.
3. Apply constrained Hamiltonian Monte Carlo method to sample from posterior.
4. Exploit Markovian structure of diffusions to reduce $\widetilde{\mathcal{O}}(\mathtt{T}^3)$ constrained HMC cost to $\widetilde{\mathcal{O}}(\mathtt{T})$.

Diffusions

Model defined by stochastic differential equation

$$\mathrm{d}\mathbf{x}_\tau = \boldsymbol{a}(\mathbf{x}_\tau, \mathbf{z})\,\mathrm{d}\tau + \boldsymbol{B}(\mathbf{x}_\tau, \mathbf{z})\,\mathrm{d}\mathbf{w}_\tau \quad \forall \tau \in \mathcal{T},$$

Diffusions

Model defined by stochastic differential equation

$$\mathrm{d}\mathbf{x}_\tau = \boldsymbol{a}(\mathbf{x}_\tau, \mathbf{z})\,\mathrm{d}\tau + \boldsymbol{B}(\mathbf{x}_\tau, \mathbf{z})\,\mathrm{d}\mathbf{w}_\tau \quad \forall \tau \in \mathcal{T},$$

- $\mathtt{X}$-dimensional state process $\mathbf{x}_{\mathcal{T}}$,

Diffusions

Model defined by stochastic differential equation

$$\mathrm{d}\mathbf{x}_\tau = \boldsymbol{a}(\mathbf{x}_\tau, \mathbf{z})\,\mathrm{d}\tau + \boldsymbol{B}(\mathbf{x}_\tau, \mathbf{z})\,\mathrm{d}\mathbf{w}_\tau \quad \forall \tau \in \mathcal{T},$$

- `X`-dimensional state process $\mathbf{x}_{\mathcal{T}}$,
- `W`-dimensional Wiener noise process $\mathbf{w}_{\mathcal{T}}$,

Diffusions

Model defined by stochastic differential equation

$$\mathrm{d}\mathbf{x}_\tau = \boldsymbol{a}(\mathbf{x}_\tau, \mathbf{z})\,\mathrm{d}\tau + \boldsymbol{B}(\mathbf{x}_\tau, \mathbf{z})\,\mathrm{d}\mathbf{w}_\tau \quad \forall \tau \in \mathcal{T},$$

- X-dimensional state process $\mathbf{x}_{\mathcal{T}}$,
- W-dimensional Wiener noise process $\mathbf{w}_{\mathcal{T}}$,
- Z-dimensional parameters $\mathbf{z}$.

Diffusions

Model defined by stochastic differential equation

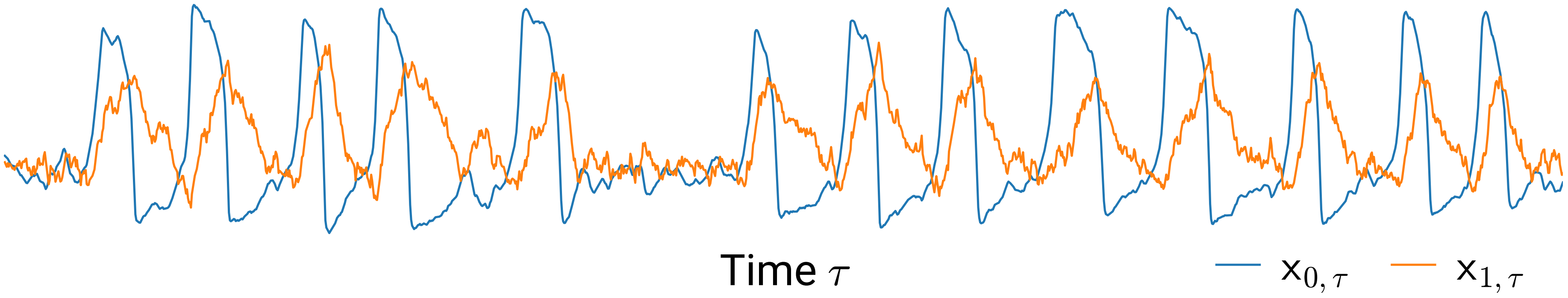
$$\mathrm{d}\mathbf{x}_\tau = \boldsymbol{a}(\mathbf{x}_\tau, \mathbf{z})\,\mathrm{d}\tau + \boldsymbol{B}(\mathbf{x}_\tau, \mathbf{z})\,\mathrm{d}\mathbf{w}_\tau \quad \forall \tau \in \mathcal{T},$$

- $\mathtt{X}$-dimensional state process $\mathbf{x}_\mathcal{T}$,
- $\mathtt{W}$-dimensional Wiener noise process $\mathbf{w}_\mathcal{T}$,
- $\mathtt{Z}$-dimensional parameters $\mathbf{z}$.

Solutions define a family of Markov kernels $\kappa_\mathcal{T}$

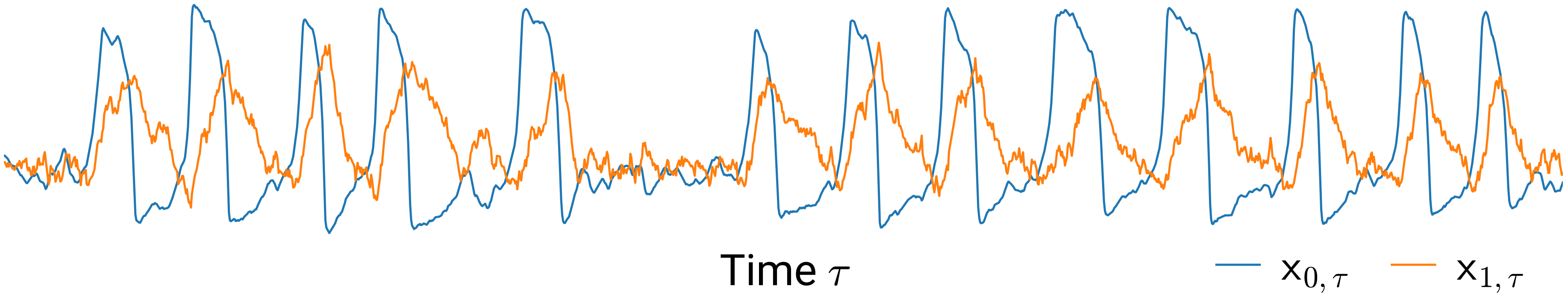
$$\mathbf{x}_\tau \mid (\mathbf{x}_0 = \boldsymbol{x}, \mathbf{z} = \boldsymbol{z}) \sim \kappa_\tau(\boldsymbol{x}, \boldsymbol{z}) \quad \forall \tau \in \mathcal{T}.$$

Example applications



Many real-world processes with noisy dynamics can be modelled as diffusions, e.g.

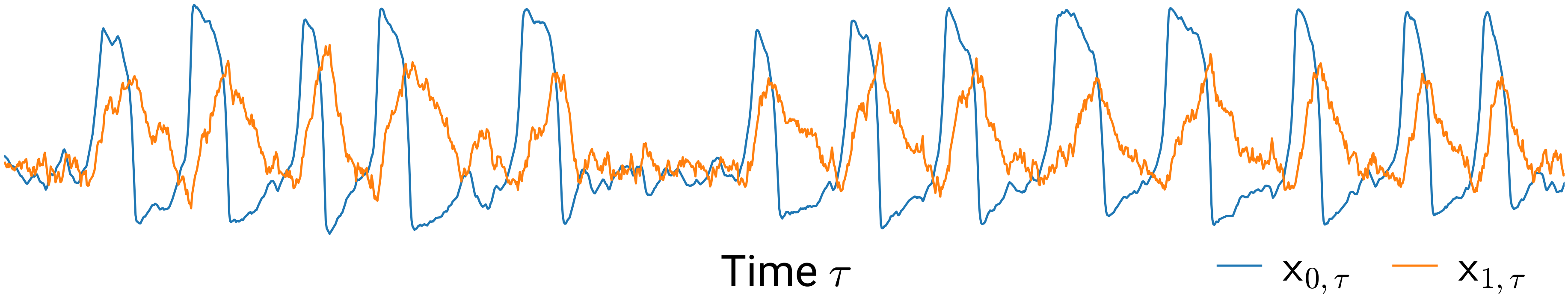
Example applications



Many real-world processes with noisy dynamics can be modelled as diffusions, e.g.

- neuronal dynamics with stochastic ion channels,

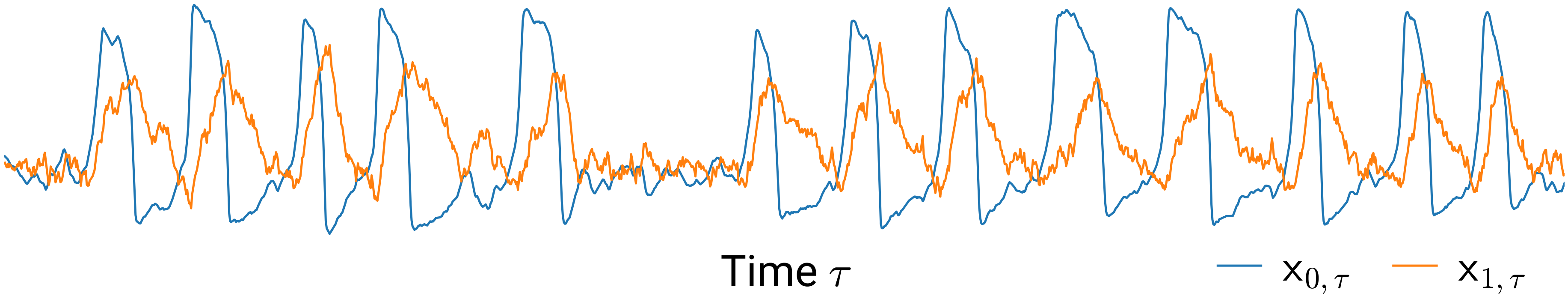
Example applications



Many real-world processes with noisy dynamics can be modelled as diffusions, e.g.

- neuronal dynamics with stochastic ion channels,
- biochemical reaction networks,

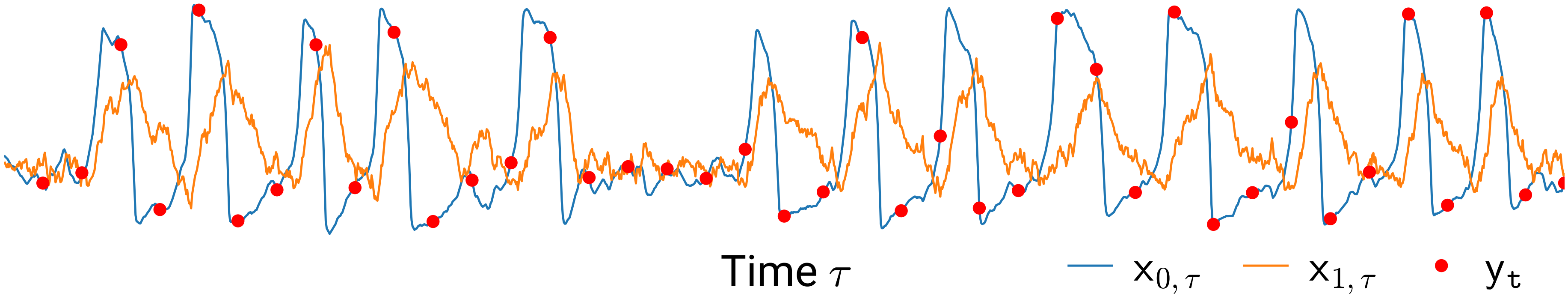
Example applications



Many real-world processes with noisy dynamics can be modelled as diffusions, e.g.

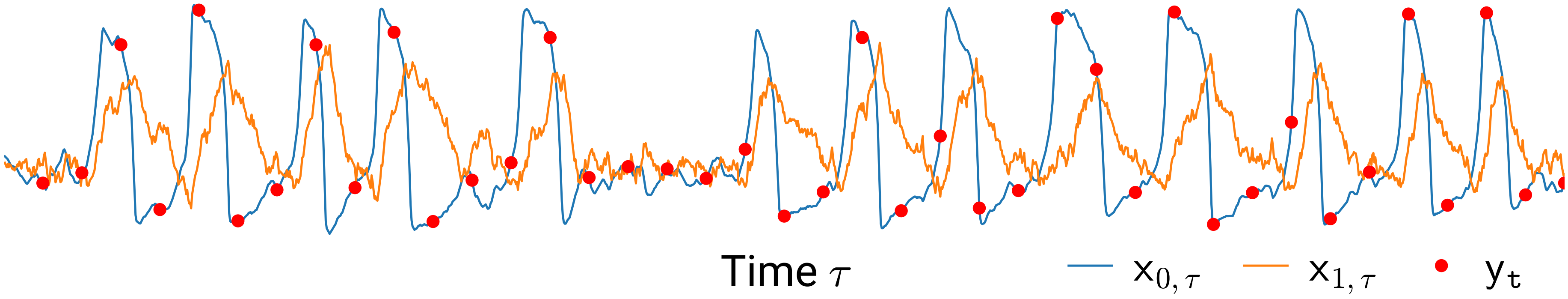
- neuronal dynamics with stochastic ion channels,
- biochemical reaction networks,
- electrical circuits subject to thermal noise.

Parameter inference



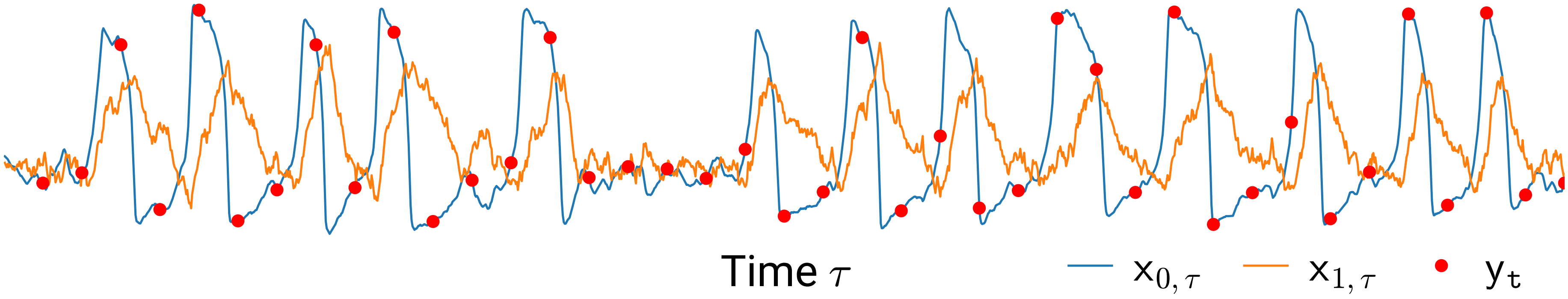
A common task is given partial observations $\mathbf{y}_{1:T}$ of the process $\mathbf{x}_{\mathcal{T}}$ at discrete times to infer the posterior distribution of the model parameters $\mathbf{z}$.

Parameter inference



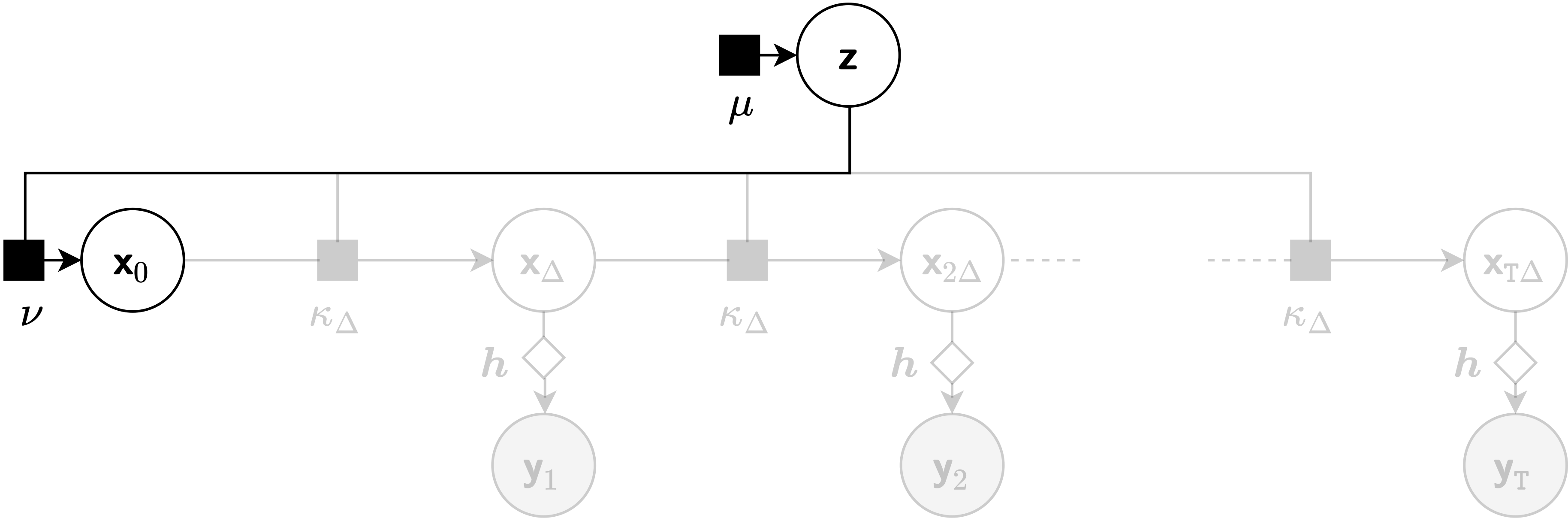
Concentrate on case where $\mathbf{y}_{\mathtt{t}} = \boldsymbol{h}(\mathbf{x}_{\Delta\mathtt{t}})\ \forall \mathtt{t} \in 1{:}\mathtt{T}$ with $\boldsymbol{h} : \mathbb{R}^{\mathtt{X}} \to \mathbb{R}^{\mathtt{Y}}$ potentially non-linear and $\mathtt{Y} < \mathtt{X}$.

Parameter inference



Simple to extend to noisy observations. Manifold MCMC methods particularly advantageous in small noise regime (Au, Graham & Thiery, 2020).

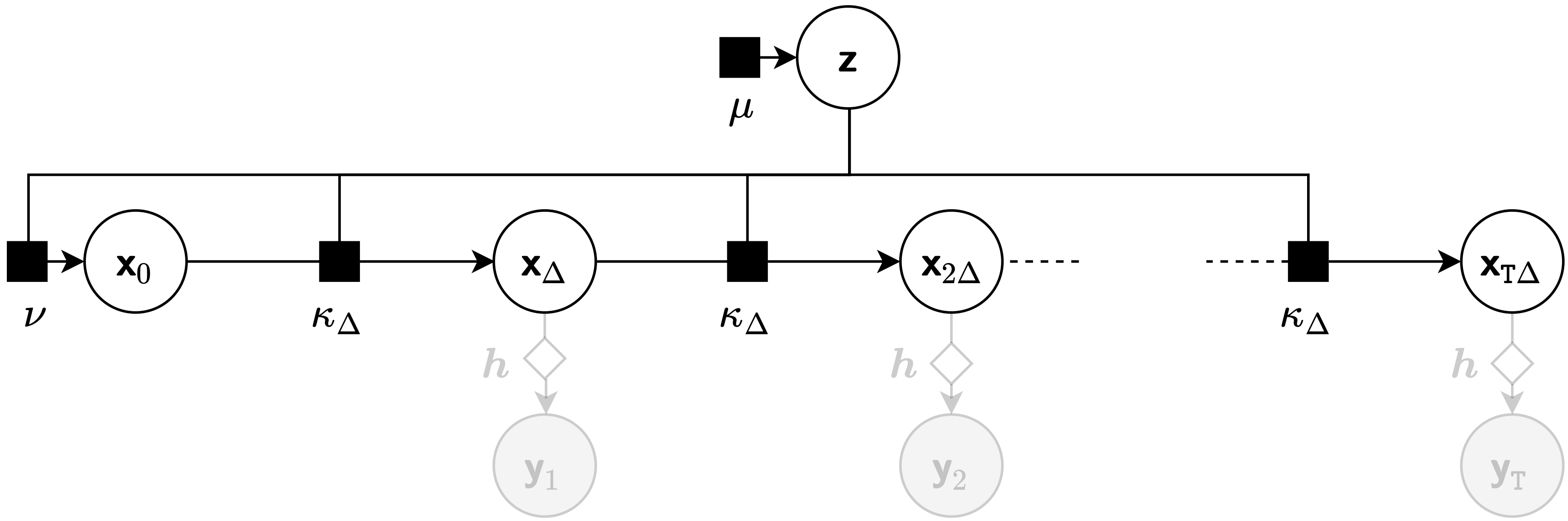
Prior generative model



Parameters $\mathbf{z}$ and initial state $\mathbf{x}_0$ given priors

$$\mathbf{z} \sim \mu, \quad \mathbf{x}_0 \sim \nu(\mathbf{z}).$$

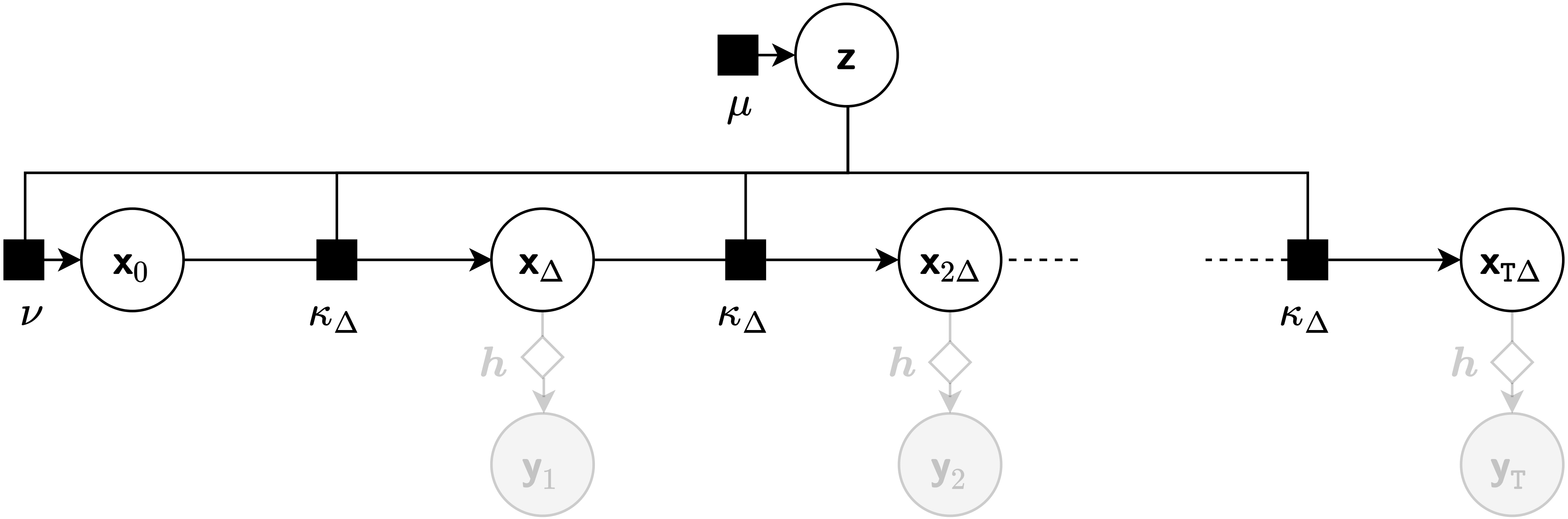
Prior generative model



State observed at $\mathtt{T}$ equispaced times $\tau_{\mathtt{t}} = \mathtt{t}\Delta$

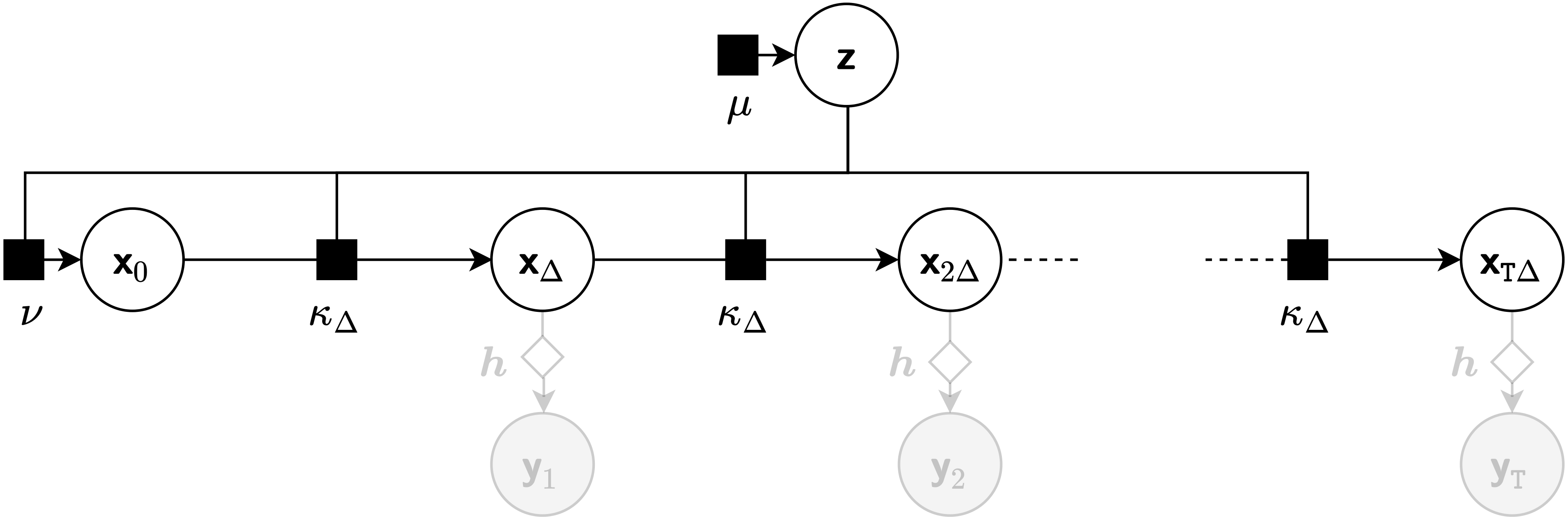
$$\mathbf{x}_{\mathtt{t}\Delta} \sim \kappa_{\Delta}\left(\mathbf{x}_{(\mathtt{t}-1)\Delta}, \mathbf{z}\right) \quad \forall \mathtt{t} \in 1{:}\mathtt{T}$$

Prior generative model



$$\bar{\pi}_0(\mathrm{d}\boldsymbol{z}, \mathrm{d}\boldsymbol{x}_0, \mathrm{d}\boldsymbol{x}_{(1:\mathtt{T})\Delta}) = \mu(\mathrm{d}\boldsymbol{z})\nu(\mathrm{d}\boldsymbol{x}_0 \mid \boldsymbol{z}) \prod_{\mathtt{t}=1}^{\mathtt{T}} \kappa_\delta(\mathrm{d}\boldsymbol{x}_{\mathtt{t}\Delta} \mid \boldsymbol{x}_{(\mathtt{t}-1)\Delta})$$

Prior generative model



However typically we can neither exactly sample from κ_Δ nor evaluate its density.

Data augmentation

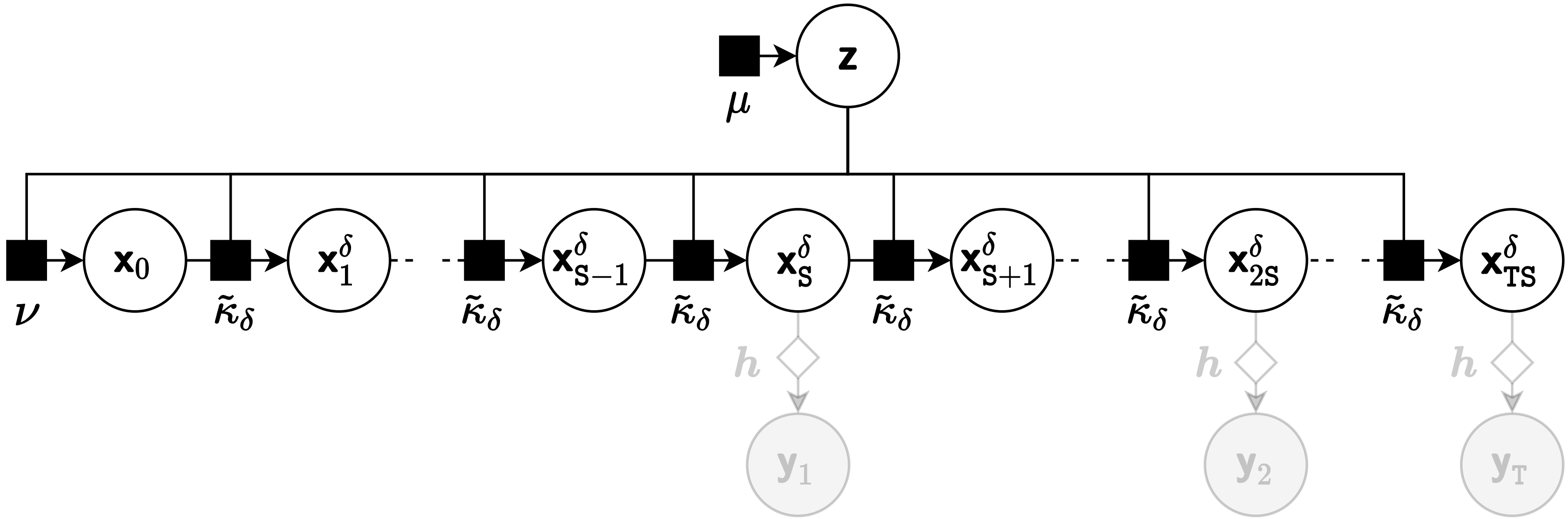
(Roberts & Stramer, 2001; Elerian, Chib + Shepard, 2001)

We instead use a numerical integration scheme - defines a kernel $\tilde{\kappa}_\delta \approx \kappa_\delta$ for small time step δ.

Data augmentation

(Roberts & Stramer, 2001; Elerian, Chib + Shepard, 2001)

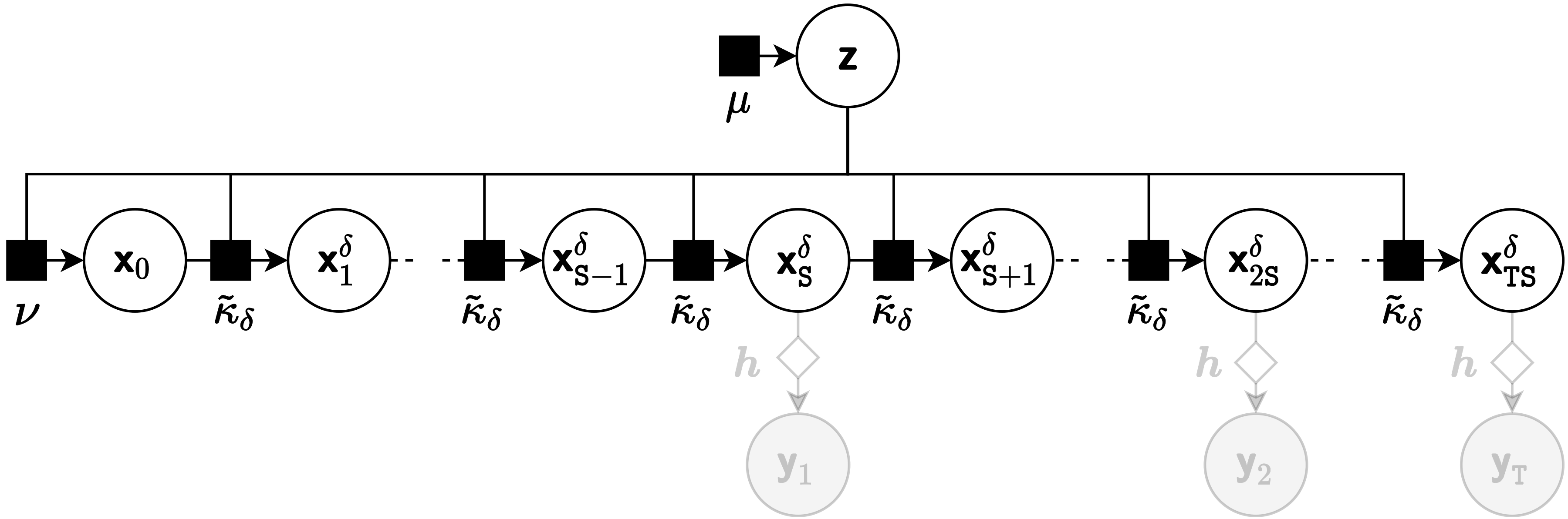
Split each inter-observation interval into $\mathtt{S}$ steps $\delta = \frac{\Delta}{\mathtt{S}}$ with approximation error $\to 0$ as $\mathtt{S} \to \infty$.



Data augmentation

(Roberts & Stramer, 2001; Elerian, Chib + Shepard, 2001)

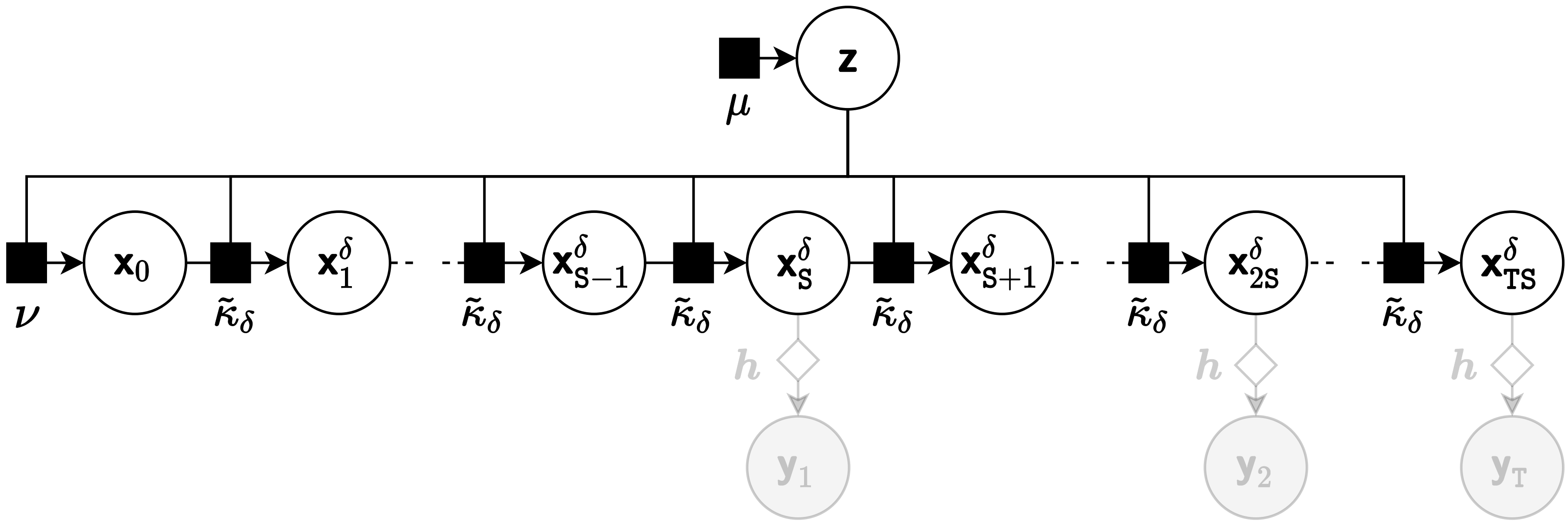
For small δ states and parameters highly correlated $\implies$ challenging for MCMC.



Data augmentation

(Roberts & Stramer, 2001; Elerian, Chib + Shepard, 2001)

Further $\tilde{\kappa}_\delta$ may not have a tractable density function in some cases.



Noise parameterisation

(Chib, Pitt & Shepard, 2004)

Typically $\tilde{\kappa}_\delta$ defined via a generative process

$$\mathsf{v} \sim \mathcal{N}(\mathbf{0}, \mathbb{I}_{\mathrm{V}}), \quad \mathsf{x} = \boldsymbol{f}_\delta(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{v}) \implies \mathsf{x} \sim \tilde{\kappa}_\delta(\boldsymbol{x}, \boldsymbol{z}).$$

Noise parameterisation

(Chib, Pitt & Shepard, 2004)

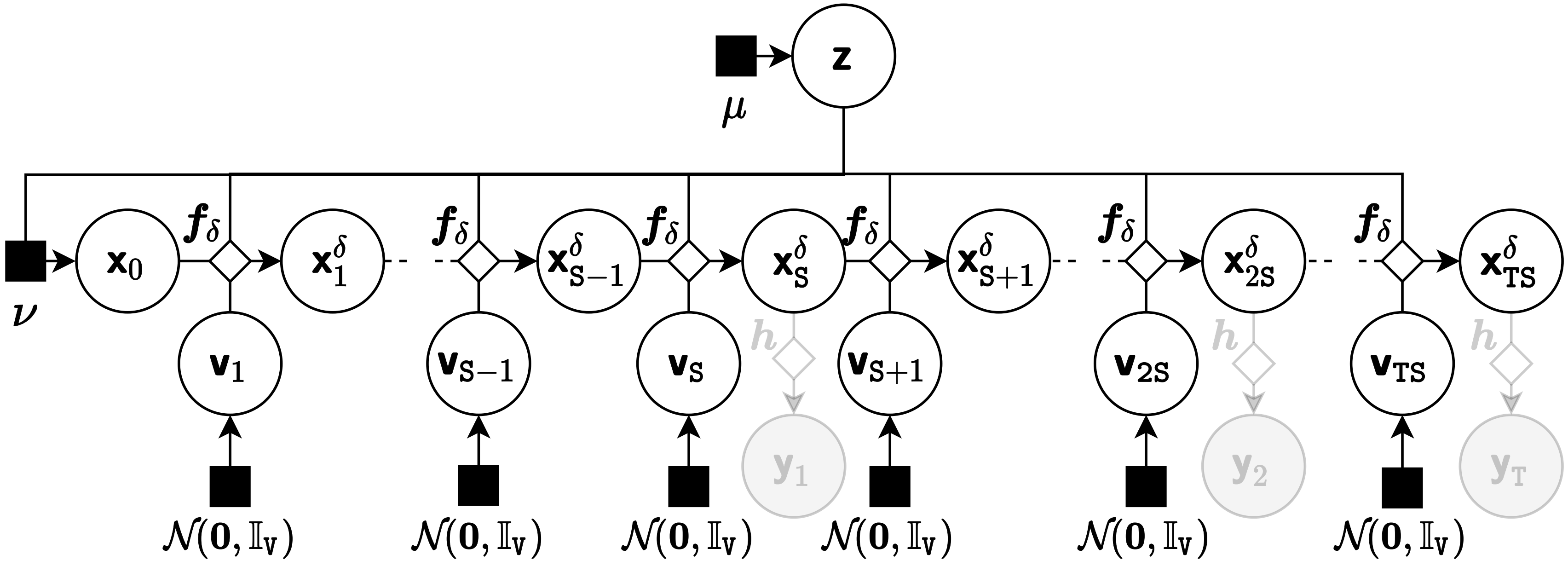
For example for the Euler-Maruyama method

$$\boldsymbol{f}_\delta(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{v}) = \boldsymbol{x} + \delta \boldsymbol{a}(\boldsymbol{x}, \boldsymbol{z}) + \delta^{\frac{1}{2}} \boldsymbol{B}(\boldsymbol{x}, \boldsymbol{z}) \boldsymbol{v}.$$

Noise parameterisation

(Chib, Pitt & Shepard, 2004)

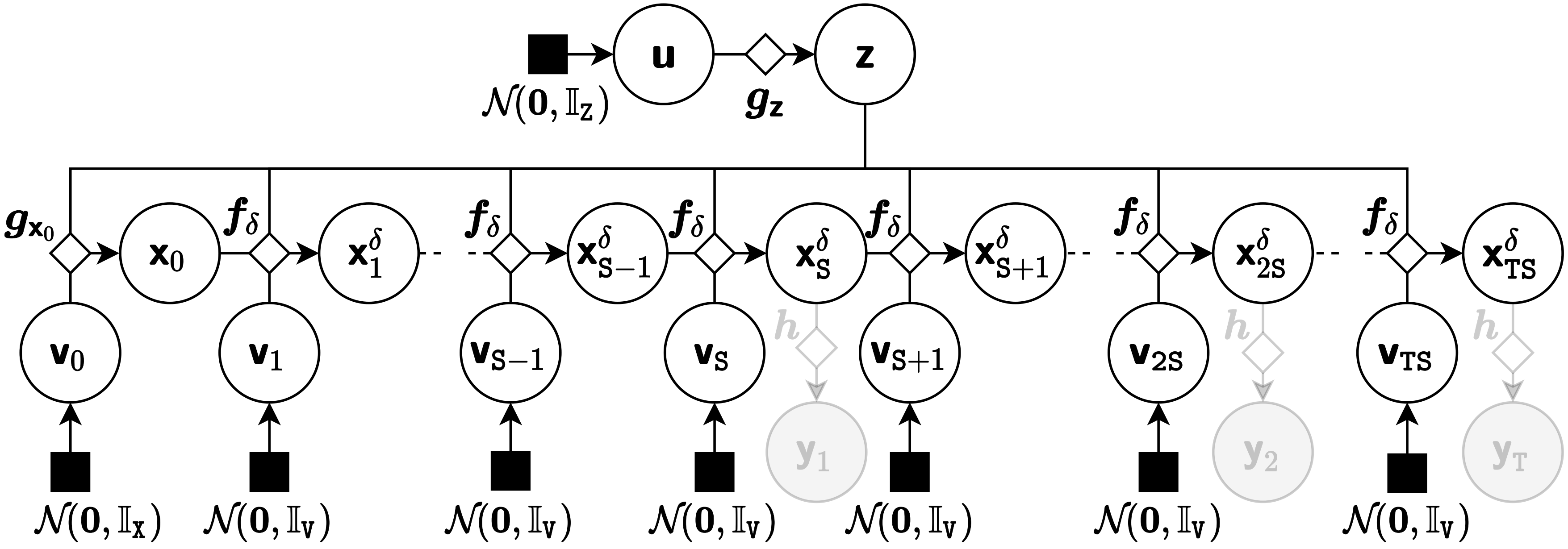
We can reparameterise the model in terms of the random vectors $\mathbf{v}_{1:TS}$ used to generate $\mathbf{x}^{\delta}_{1:TS}$.



Non-centred reparametrisation

(Papaspiliopoulos, Roberts + Sköld, 2003)

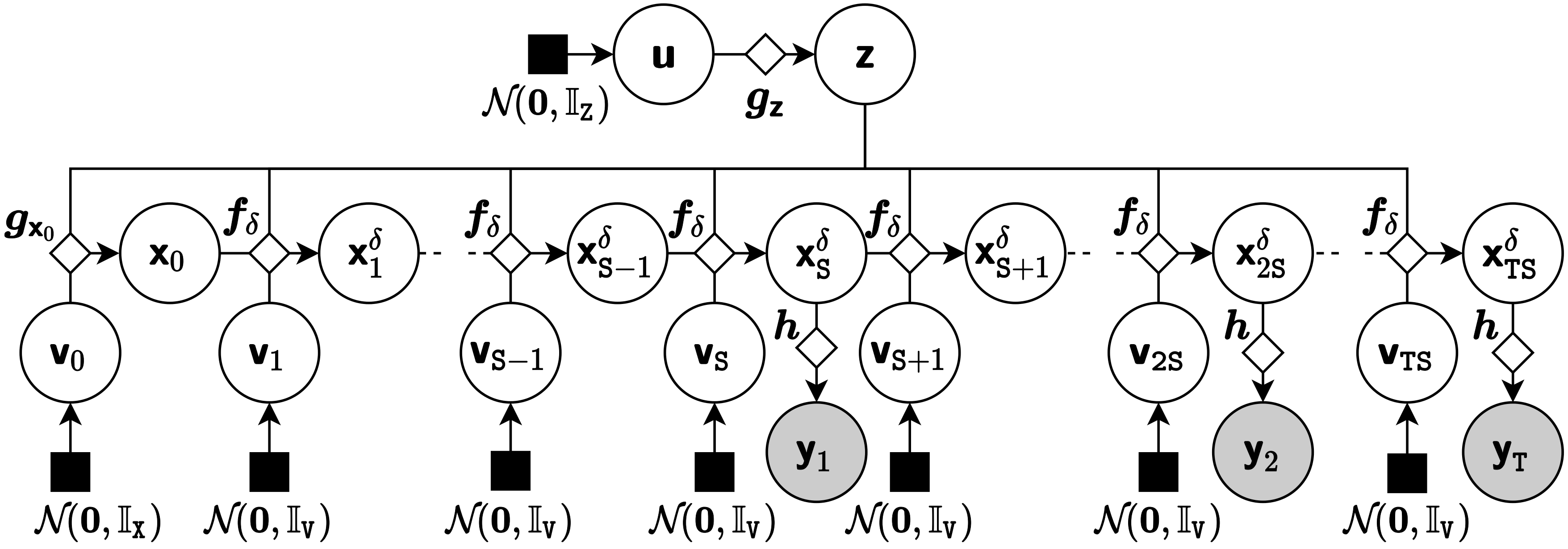
Assume that $\mathbf{x}_0$ and $\mathbf{z}$ can also be reparametrised in terms of standard normal vectors $\mathbf{v}_0$ and $\mathbf{u}$.



Non-centred reparametrisation

(Papaspiliopoulos, Roberts + Sköld, 2003)

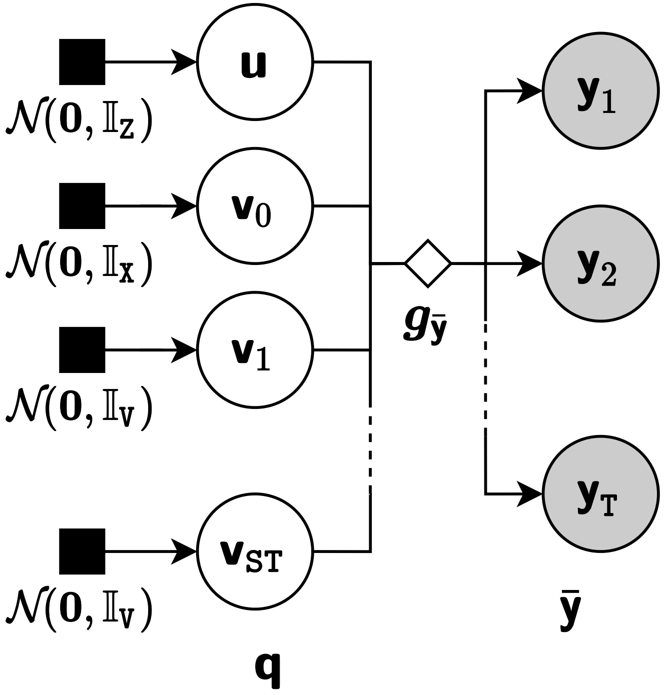
Prior distribution now product of independent normal factors. *However*: how to form posterior?



Differentiable generative model

(Graham & Storkey, 2017)

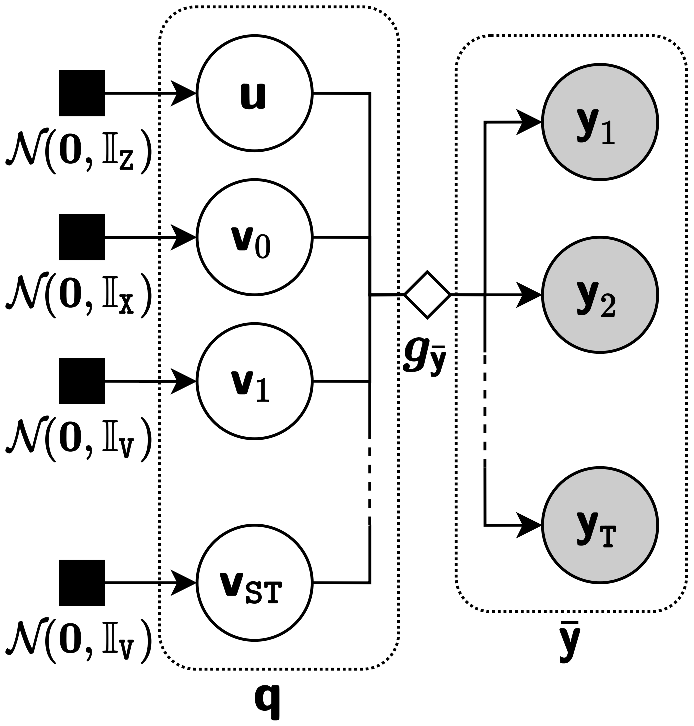
Observations are computed as a deterministic function of latent inputs with tractable prior density



Differentiable generative model

(Graham & Storkey, 2017)

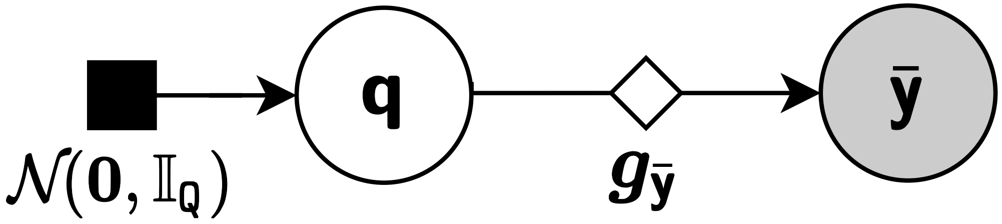
Observations are computed as a deterministic function of latent inputs with tractable prior density



Differentiable generative model

(Graham & Storkey, 2017)

Observations are computed as a deterministic function of latent inputs with tractable prior density



$$\mathtt{Q} = \mathtt{Z} + \mathtt{X} + \mathtt{STV} \quad \text{and} \quad \bar{\mathtt{Y}} = \mathtt{TY}.$$

Assume that $\boldsymbol{g}_{\bar{\mathbf{y}}} : \mathbb{R}^{\mathtt{Q}} \to \mathbb{R}^{\bar{\mathtt{Y}}}$ is differentiable and has a surjective differential almost everywhere.

Posterior on a manifold (Diaconis+, 2011)

Posterior π on $\mathbf{q} \mid \bar{\mathbf{y}} = \bar{\boldsymbol{y}}$ supported on implicitly defined manifold $\boldsymbol{g}_{\bar{\mathbf{y}}}^{-1}(\bar{\boldsymbol{y}}) = \{\boldsymbol{q} \in \mathbb{R}^{\mathtt{Q}} : \boldsymbol{g}_{\bar{\mathbf{y}}}(\boldsymbol{q}) = \bar{\boldsymbol{y}}\}$.

Posterior on a manifold (Diaconis+, 2011)

Posterior π on $\mathbf{q} \mid \bar{\mathbf{y}} = \bar{y}$ supported on implicitly defined manifold $g_{\bar{\mathbf{y}}}^{-1}(\bar{y}) = \{q \in \mathbb{R}^{Q} : g_{\bar{\mathbf{y}}}(q) = \bar{y}\}$.

$g_{\bar{\mathbf{y}}}^{-1}(\bar{y})$ has zero Lebesgue measure $\implies$ π has no density with respect to Lebesgue measure on $\mathbb{R}^{Q}$.

Posterior on a manifold (Diaconis+, 2011)

Posterior π on $\mathbf{q} \mid \bar{\mathbf{y}} = \bar{y}$ supported on implicitly defined manifold $g_{\bar{\mathbf{y}}}^{-1}(\bar{y}) = \{q \in \mathbb{R}^{\mathtt{Q}} : g_{\bar{\mathbf{y}}}(q) = \bar{y}\}$.

However π has a density with respect to $\eta_{\mathtt{Q}}^{\mathtt{D}}$, the $\mathtt{D} = \mathtt{Q} - \bar{\mathtt{Y}}$ dimensional Hausdorff measure on $\mathbb{R}^{\mathtt{Q}}$

$$\frac{\mathrm{d}\pi}{\mathrm{d}\eta_{\mathtt{Q}}^{\mathtt{D}}}(\boldsymbol{q}) \propto \exp(-\phi(\boldsymbol{q}))\, \mathbb{1}_{\boldsymbol{g}_{\bar{\mathbf{y}}}^{-1}(\bar{\boldsymbol{y}})}(\boldsymbol{q}),$$

Posterior on a manifold (Diaconis+, 2011)

Posterior π on $\mathbf{q} \mid \bar{\mathbf{y}} = \bar{y}$ supported on implicitly defined manifold $g_{\bar{\mathbf{y}}}^{-1}(\bar{y}) = \{q \in \mathbb{R}^{\mathtt{Q}} : g_{\bar{\mathbf{y}}}(q) = \bar{y}\}$.

However π has a density with respect to $\eta_{\mathtt{Q}}^{\mathtt{D}}$, the $\mathtt{D} = \mathtt{Q} - \bar{\mathtt{Y}}$ dimensional Hausdorff measure on $\mathbb{R}^{\mathtt{Q}}$

$$\frac{\mathrm{d}\pi}{\mathrm{d}\eta_{\mathtt{Q}}^{\mathtt{D}}}(\boldsymbol{q}) \propto \exp(-\phi(\boldsymbol{q}))\, \mathbb{1}_{\boldsymbol{g}_{\bar{\mathbf{y}}}^{-1}(\bar{\boldsymbol{y}})}(\boldsymbol{q}),$$

$$\phi(\boldsymbol{q}) = \frac{1}{2}\boldsymbol{q}^{\mathsf{T}}\boldsymbol{q} + \frac{1}{2}\log\left|\,\partial\boldsymbol{g}_{\bar{\mathbf{y}}}(\boldsymbol{q})\,\partial\boldsymbol{g}_{\bar{\mathbf{y}}}(\boldsymbol{q})^{\mathsf{T}}\right|.$$

Constrained Hamiltonian Monte Carlo

(Hartmann & Schutte, 2005; Brubaker+, 2012; Lelièvre+, 2019)

Constrained Hamiltonian Monte Carlo

(Hartmann & Schutte, 2005; Brubaker+, 2012; Lelièvre+, 2019)

MCMC method based on simulating a constrained Hamiltonian dynamic defined by DAEs

$$\dot{\boldsymbol{q}} = \boldsymbol{p},\ \dot{\boldsymbol{p}} = -\nabla\phi(\boldsymbol{q})^{\mathsf{T}} + \partial \boldsymbol{g}_{\bar{\mathbf{y}}}(\boldsymbol{q})^{\mathsf{T}}\boldsymbol{\lambda},\ \boldsymbol{g}_{\bar{\mathbf{y}}}(\boldsymbol{q}) = \bar{\boldsymbol{y}},$$

Constrained Hamiltonian Monte Carlo

(Hartmann & Schutte, 2005; Brubaker+, 2012; Lelièvre+, 2019)

MCMC method based on simulating a constrained Hamiltonian dynamic defined by DAEs

$$\dot{\boldsymbol{q}} = \boldsymbol{p},\ \dot{\boldsymbol{p}} = -\nabla\phi(\boldsymbol{q})^{\mathsf{T}} + \partial \boldsymbol{g}_{\bar{\mathbf{y}}}(\boldsymbol{q})^{\mathsf{T}}\boldsymbol{\lambda},\ \boldsymbol{g}_{\bar{\mathbf{y}}}(\boldsymbol{q}) = \bar{\boldsymbol{y}},$$

Simulate using a constraint-preserving symplectic integrator such as RATTLE (Andersen, 1983).

Constrained Hamiltonian Monte Carlo

(Hartmann & Schutte, 2005; Brubaker+, 2012; Lelièvre+, 2019)

MCMC method based on simulating a constrained Hamiltonian dynamic defined by DAEs

$$\dot{\boldsymbol{q}} = \boldsymbol{p},\ \dot{\boldsymbol{p}} = -\nabla\phi(\boldsymbol{q})^{\mathsf{T}} + \partial \boldsymbol{g}_{\bar{\mathrm{y}}}(\boldsymbol{q})^{\mathsf{T}}\boldsymbol{\lambda},\ \boldsymbol{g}_{\bar{\mathrm{y}}}(\boldsymbol{q}) = \bar{\boldsymbol{y}},$$

To enforce constraints in each step solve $\bar{\mathtt{Y}}$ non-linear equations to project $\boldsymbol{q}$ on to manifold and $\bar{\mathtt{Y}}$ linear equations to project $\boldsymbol{p}$ on to cotangent space.

Constrained HMC implementation

Constrained HMC implementation

mici

Manifold MCMC methods in Python

Available on Github at git.io/mici.py or

```
pip install mici
```

Constrained HMC computational cost

Dominant costs are evaluating $\mathcal{O}(\mathtt{T}) \times \mathcal{O}(\mathtt{ST})$ Jacobian $\partial \boldsymbol{g}_{\bar{\mathbf{y}}}(\boldsymbol{q})$ and Gram matrix $\partial \boldsymbol{g}_{\bar{\mathbf{y}}}(\boldsymbol{q})\, \partial \boldsymbol{g}_{\bar{\mathbf{y}}}(\boldsymbol{q})^{\mathsf{T}}$.

Constrained HMC computational cost

Dominant costs are evaluating $\mathcal{O}(\mathtt{T}) \times \mathcal{O}(\mathtt{ST})$ Jacobian $\partial \boldsymbol{g}_{\bar{\mathbf{y}}}(\boldsymbol{q})$ and Gram matrix $\partial \boldsymbol{g}_{\bar{\mathbf{y}}}(\boldsymbol{q})\, \partial \boldsymbol{g}_{\bar{\mathbf{y}}}(\boldsymbol{q})^{\mathsf{T}}$.

Using reverse-mode algorithmic differentiation evaluating $\partial \boldsymbol{g}_{\bar{\mathbf{y}}}(\boldsymbol{q})$ costs $\mathcal{O}(\mathtt{T})$ evaluations of $\boldsymbol{g}_{\bar{\mathbf{y}}}(\boldsymbol{q})$.

Constrained HMC computational cost

Dominant costs are evaluating $\mathcal{O}(\mathtt{T}) \times \mathcal{O}(\mathtt{ST})$ Jacobian $\partial \boldsymbol{g}_{\bar{\mathbf{y}}}(\boldsymbol{q})$ and Gram matrix $\partial \boldsymbol{g}_{\bar{\mathbf{y}}}(\boldsymbol{q})\, \partial \boldsymbol{g}_{\bar{\mathbf{y}}}(\boldsymbol{q})^{\mathsf{T}}$.

Using reverse-mode algorithmic differentiation evaluating $\partial \boldsymbol{g}_{\bar{\mathbf{y}}}(\boldsymbol{q})$ costs $\mathcal{O}(\mathtt{T})$ evaluations of $\boldsymbol{g}_{\bar{\mathbf{y}}}(\boldsymbol{q})$.

Cost of evaluating $\boldsymbol{g}_{\bar{\mathbf{y}}}(\boldsymbol{q})$ i.e. forward simulating from model is $\mathcal{O}(\mathtt{ST})$ $\therefore$ $\partial \boldsymbol{g}_{\bar{\mathbf{y}}}(\boldsymbol{q})$ has $\mathcal{O}(\mathtt{ST}^2)$ cost.

Constrained HMC computational cost

Dominant costs are evaluating $\mathcal{O}(\mathtt{T}) \times \mathcal{O}(\mathtt{ST})$ Jacobian $\partial \boldsymbol{g}_{\bar{\mathrm{y}}}(\boldsymbol{q})$ and Gram matrix $\partial \boldsymbol{g}_{\bar{\mathrm{y}}}(\boldsymbol{q})\, \partial \boldsymbol{g}_{\bar{\mathrm{y}}}(\boldsymbol{q})^{\mathsf{T}}$.

Using reverse-mode algorithmic differentiation evaluating $\partial \boldsymbol{g}_{\bar{\mathrm{y}}}(\boldsymbol{q})$ costs $\mathcal{O}(\mathtt{T})$ evaluations of $\boldsymbol{g}_{\bar{\mathrm{y}}}(\boldsymbol{q})$.

Cost of evaluating $\boldsymbol{g}_{\bar{\mathrm{y}}}(\boldsymbol{q})$ i.e. forward simulating from model is $\mathcal{O}(\mathtt{ST})$ $\therefore$ $\partial \boldsymbol{g}_{\bar{\mathrm{y}}}(\boldsymbol{q})$ has $\mathcal{O}(\mathtt{ST}^2)$ cost.

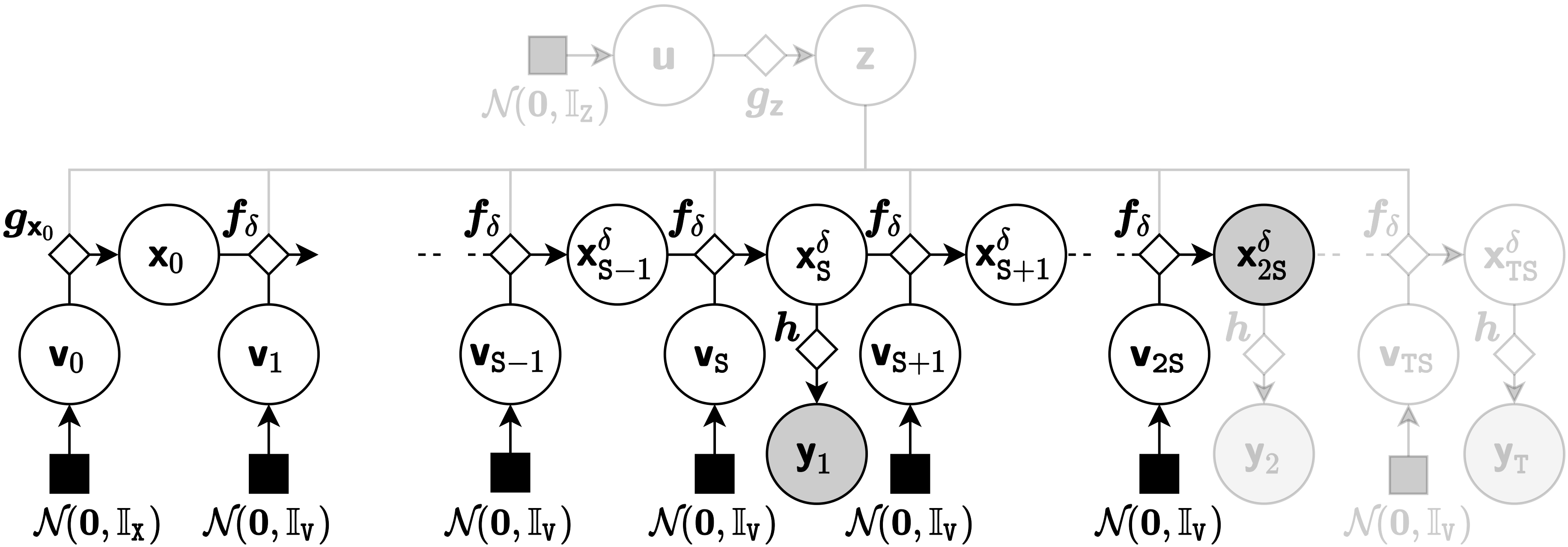
As $\partial \boldsymbol{g}_{\bar{\mathrm{y}}}(\boldsymbol{q})$ has limited sparsity, evaluating $\partial \boldsymbol{g}_{\bar{\mathrm{y}}}(\boldsymbol{q})\, \partial \boldsymbol{g}_{\bar{\mathrm{y}}}(\boldsymbol{q})^{\mathsf{T}}$ is $\mathcal{O}(\mathtt{ST}^3)$.

Blocking scheme

However by exploiting Markovianity can reduce complexity to linear in `S` and `T`.

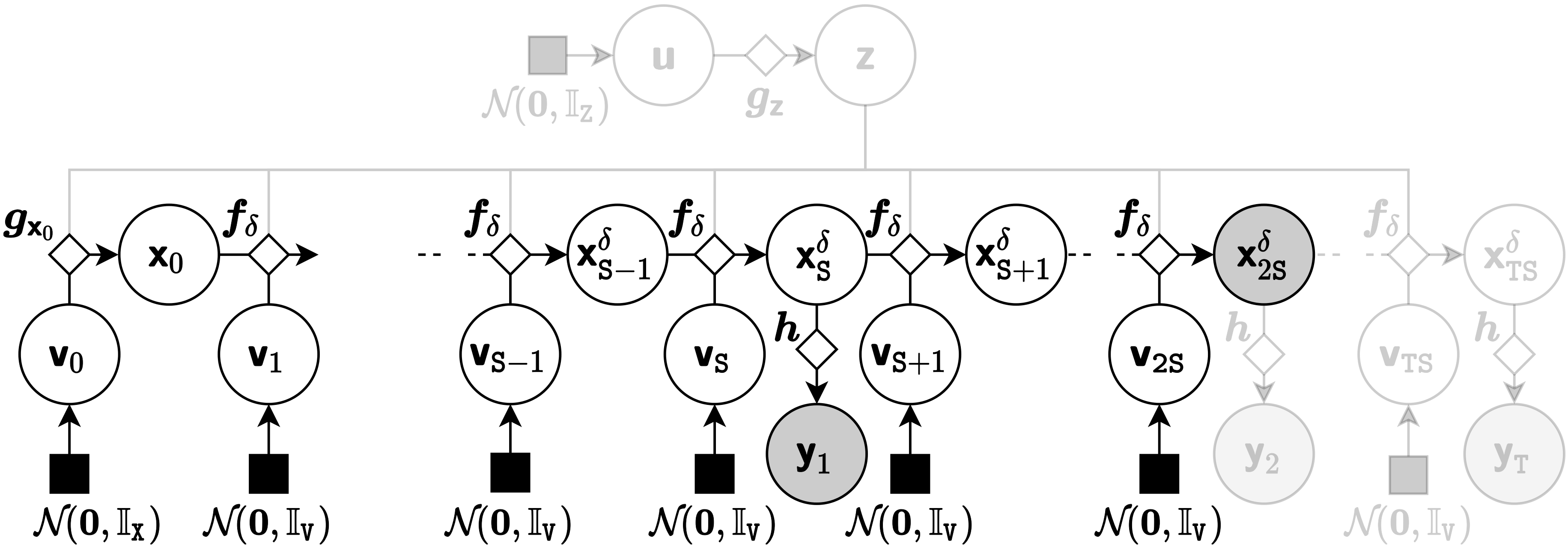
Blocking scheme

For adjacent pairs of observation times we condition on the second full latent state of the pair.



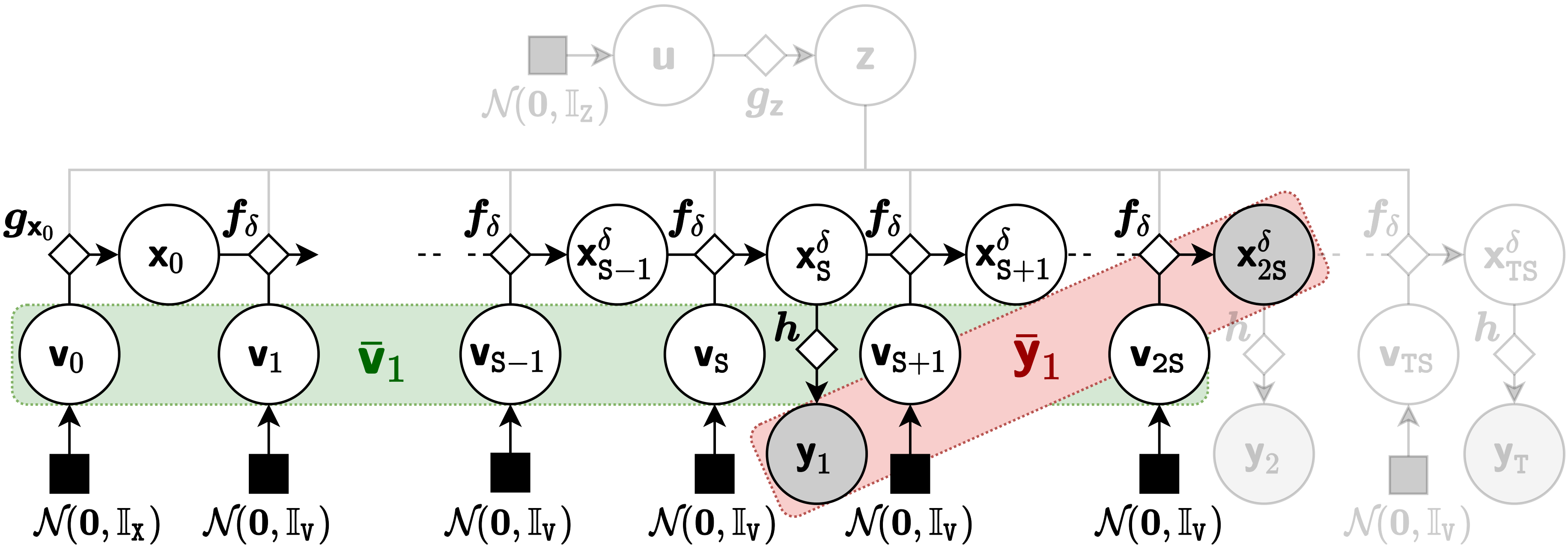
Blocking scheme

Generalise by splitting into subsequences or *blocks* of R observation times.



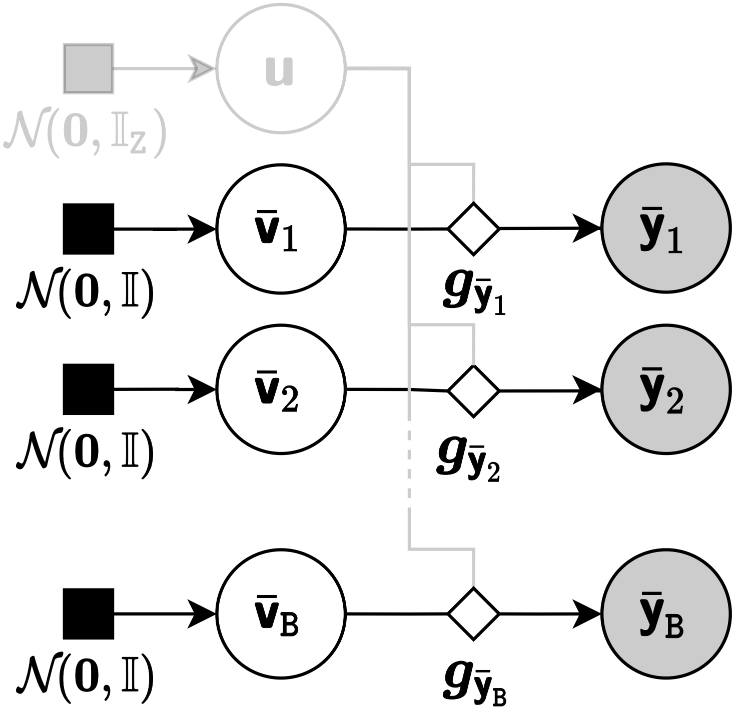
Blocking scheme

Group the noise vectors and observations / conditioned states in each block.



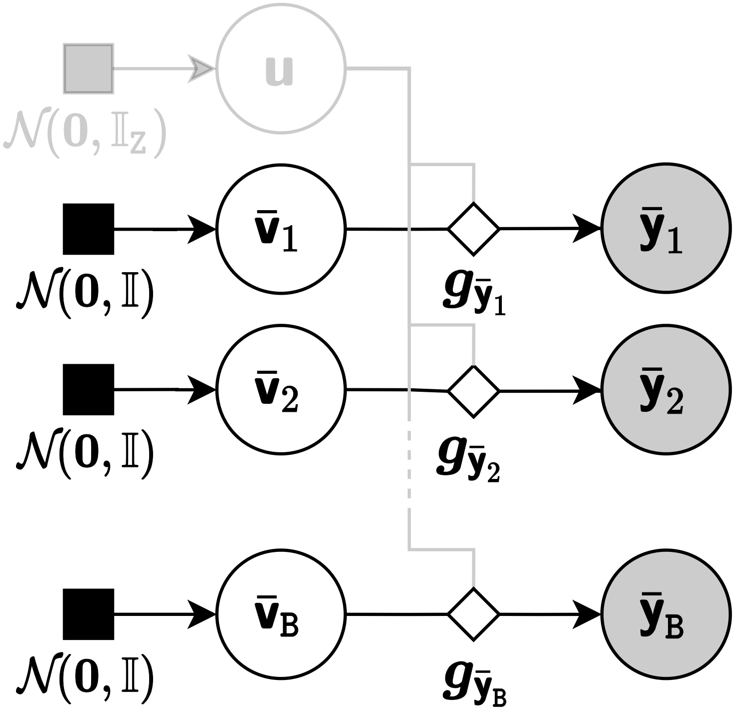
Blocking scheme

Each 'observation' block then only depends on the correspond noise vector block and parameters.



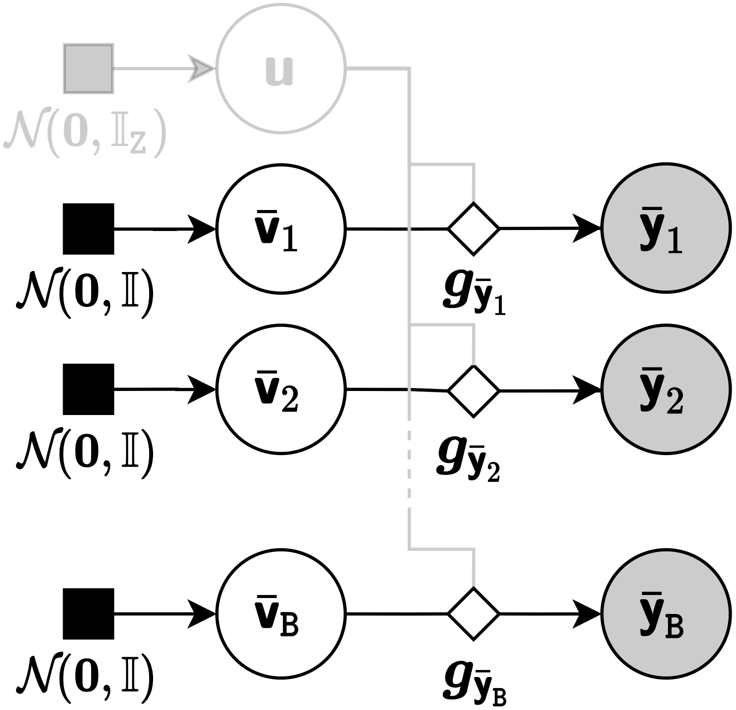
Blocking scheme

For 'blocked' generator $\boldsymbol{g}_{\bar{\mathbf{y}}_:}$, evaluation of $\partial \boldsymbol{g}_{\bar{\mathbf{y}}_:}(\boldsymbol{q})$ is $\mathcal{O}(\mathtt{RST})$ and $\partial \boldsymbol{g}_{\bar{\mathbf{y}}_:}(\boldsymbol{q})\, \partial \boldsymbol{g}_{\bar{\mathbf{y}}_:}(\boldsymbol{q})^\mathsf{T}$ is $\mathcal{O}(\mathtt{R}^2\mathtt{ST})$ cost.



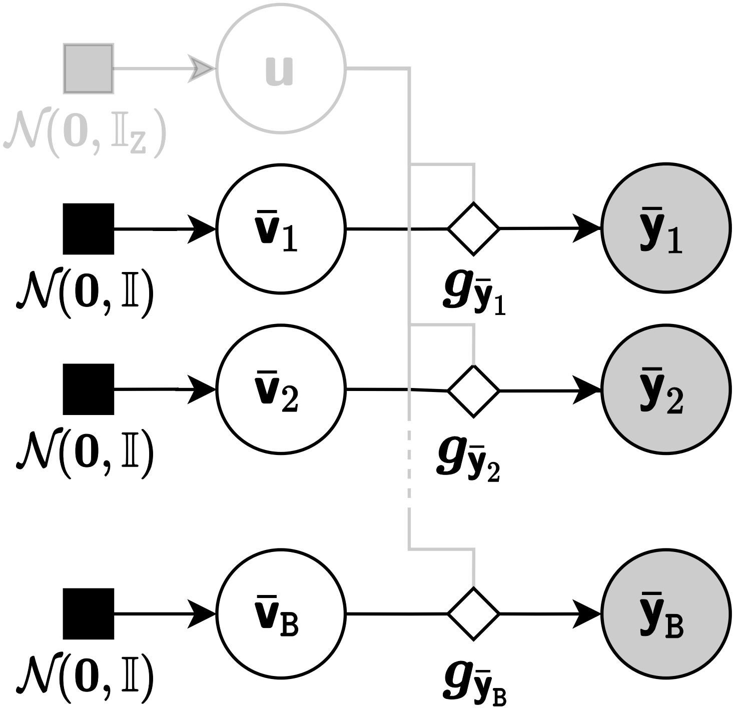
Blocking scheme

Blocking strategy similar to that used in methods using Gibbs updates, e.g. Golightly & Wilkinson (2006).



Blocking scheme

In practice need to alternate updates using two blocking partitions for ergodicity.



FitzHugh-Nagumo example

Simplified neural model defined by hypoelliptic system of stochastic differential equations

$$\begin{bmatrix} \mathrm{dx}_0 \\ \mathrm{dx}_1 \end{bmatrix} = \begin{bmatrix} \epsilon^{-1}(\mathrm{x}_1 - \mathrm{x}_2^3 - \mathrm{x}_2) \\ \gamma \mathrm{x}_1 - \mathrm{x}_2 + \beta \end{bmatrix} \mathrm{d}\tau + \begin{bmatrix} 0 \\ \sigma \end{bmatrix} \mathrm{dw}.$$

FitzHugh-Nagumo example

Simplified neural model defined by hypoelliptic system of stochastic differential equations

$$\begin{bmatrix} \mathrm{d}x_0 \\ \mathrm{d}x_1 \end{bmatrix} = \begin{bmatrix} \epsilon^{-1}(x_1 - x_2^3 - x_2) \\ \gamma x_1 - x_2 + \beta \end{bmatrix} \mathrm{d}\tau + \begin{bmatrix} 0 \\ \sigma \end{bmatrix} \mathrm{d}w.$$

Weakly informative priors on $\mathbf{z} = [\sigma;\ \epsilon;\ \gamma;\ \beta]$ & $\mathbf{x}_0$.

FitzHugh-Nagumo example

Simplified neural model defined by hypoelliptic system of stochastic differential equations

$$\begin{bmatrix} \mathrm{dx}_0 \\ \mathrm{dx}_1 \end{bmatrix} = \begin{bmatrix} \epsilon^{-1}(\mathrm{x}_1 - \mathrm{x}_2^3 - \mathrm{x}_2) \\ \gamma \mathrm{x}_1 - \mathrm{x}_2 + \beta \end{bmatrix} \mathrm{d}\tau + \begin{bmatrix} 0 \\ \sigma \end{bmatrix} \mathrm{dw}.$$

Weakly informative priors on $\mathbf{z} = [\sigma;\ \epsilon;\ \gamma;\ \beta]$ & $\mathbf{x}_0$.

Observations $\mathrm{y}_{\mathtt{t}} = \mathrm{x}_{0,\Delta\mathtt{t}}\ \forall \mathtt{t} \in 1{:}\mathtt{T}$ with $\Delta = 0.5$.

FitzHugh-Nagumo example

Simplified neural model defined by hypoelliptic system of stochastic differential equations

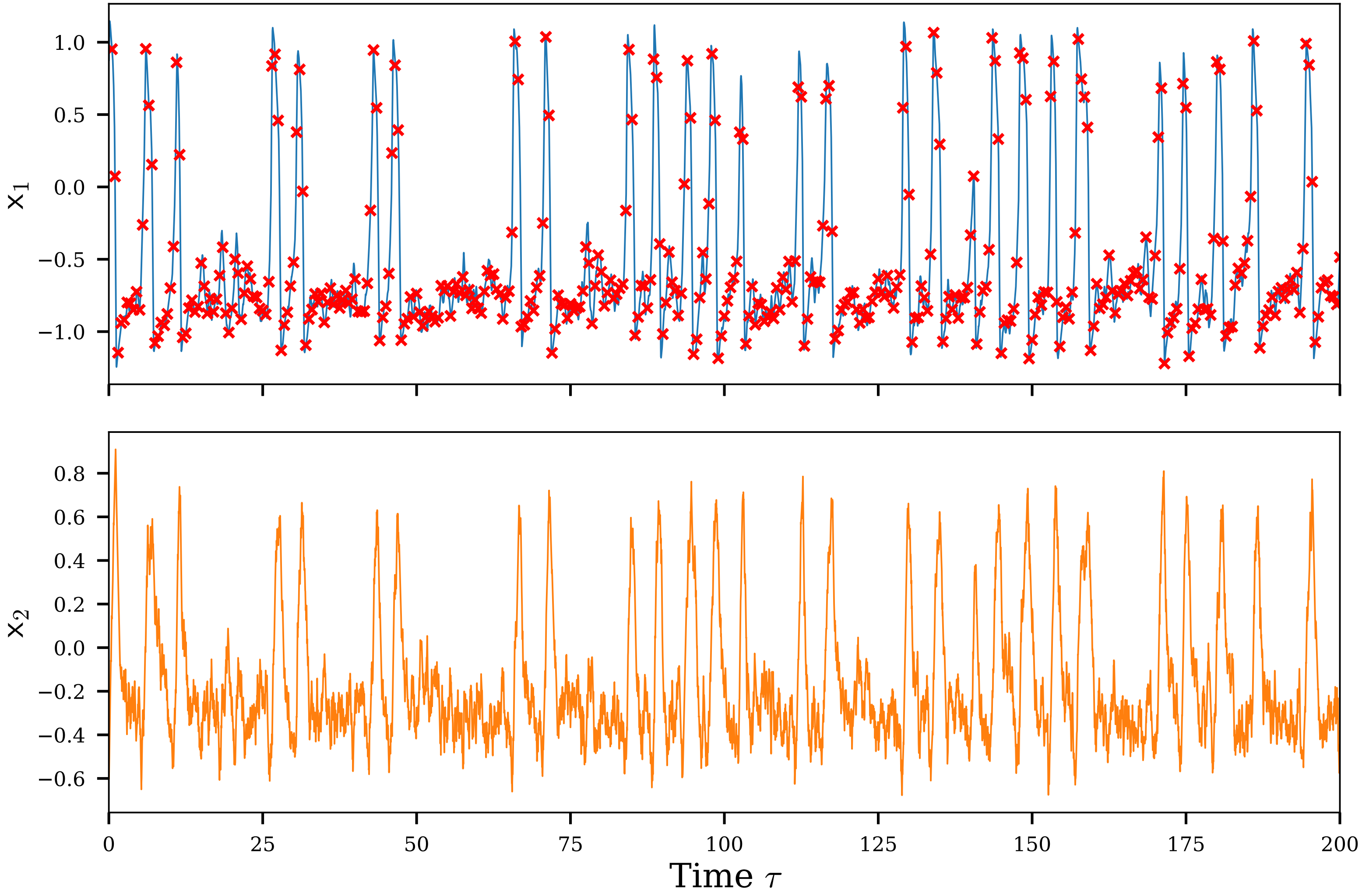
$$\begin{bmatrix} \mathrm{d}x_0 \\ \mathrm{d}x_1 \end{bmatrix} = \begin{bmatrix} \epsilon^{-1}(x_1 - x_2^3 - x_2) \\ \gamma x_1 - x_2 + \beta \end{bmatrix} \mathrm{d}\tau + \begin{bmatrix} 0 \\ \sigma \end{bmatrix} \mathrm{d}w.$$

Weakly informative priors on $\mathbf{z} = [\sigma;\ \epsilon;\ \gamma;\ \beta]$ & $\mathbf{x}_0$.

Observations $y_t = x_{0,\Delta t}$ $\forall t \in 1{:}T$ with $\Delta = 0.5$.

Use strong-order 1.5 Taylor scheme for time-discretisation $\mathbf{x}^{\delta}_{1:ST}$ with $\delta = \frac{\Delta}{S}$.

Simulated data $T = 400$ and $S = 25$



Experiments

Measure average wall-clock time per integrator step $\hat{\tau}_{\text{step}}$ and per effective sample $\hat{\tau}_{\text{eff}}$ for

1. $\mathtt{S} \in \{25, 50, 100, 200, 400\}$ and fixed $\mathtt{T} = 100$.
2. $\mathtt{T} \in \{25, 50, 100, 200, 400\}$ and fixed $\mathtt{S} = 25$.

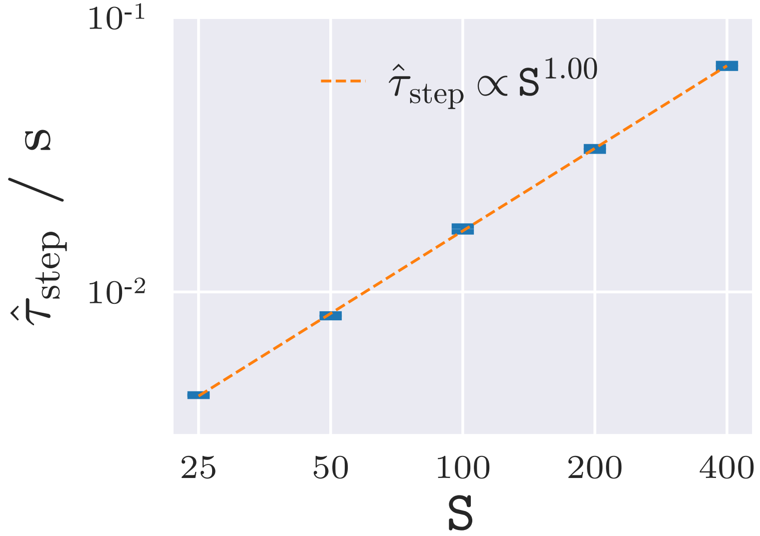
Experiments

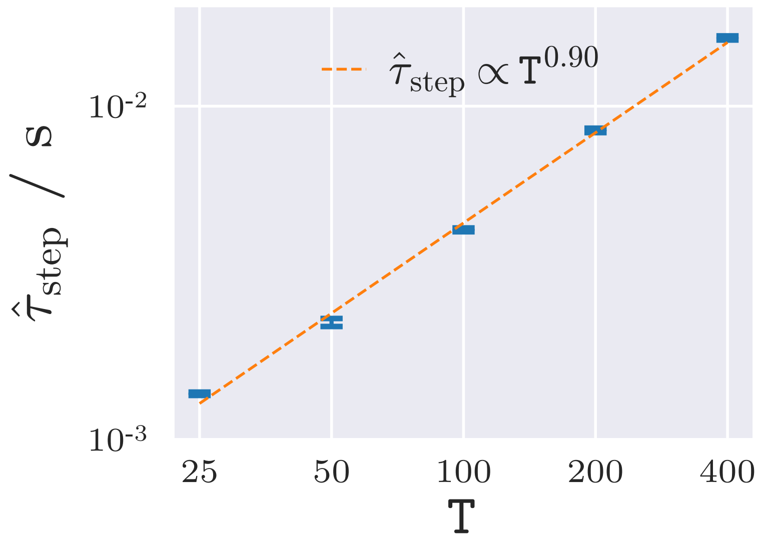
Measure average wall-clock time per integrator step $\hat{\tau}_{\text{step}}$ and per effective sample $\hat{\tau}_{\text{eff}}$ for

1. $\mathtt{S} \in \{25, 50, 100, 200, 400\}$ and fixed $\mathtt{T} = 100$.
2. $\mathtt{T} \in \{25, 50, 100, 200, 400\}$ and fixed $\mathtt{S} = 25$.

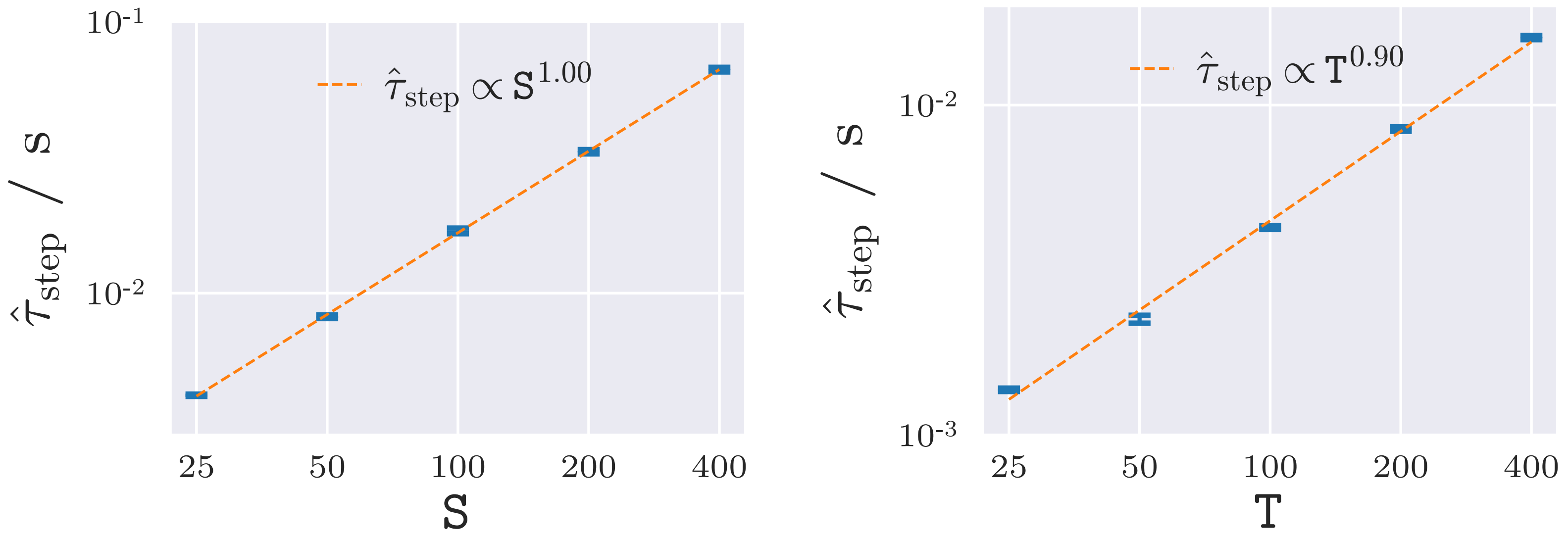
In both cases use a fixed block size of $\mathtt{R} = 5$.

Compute time per integrator step



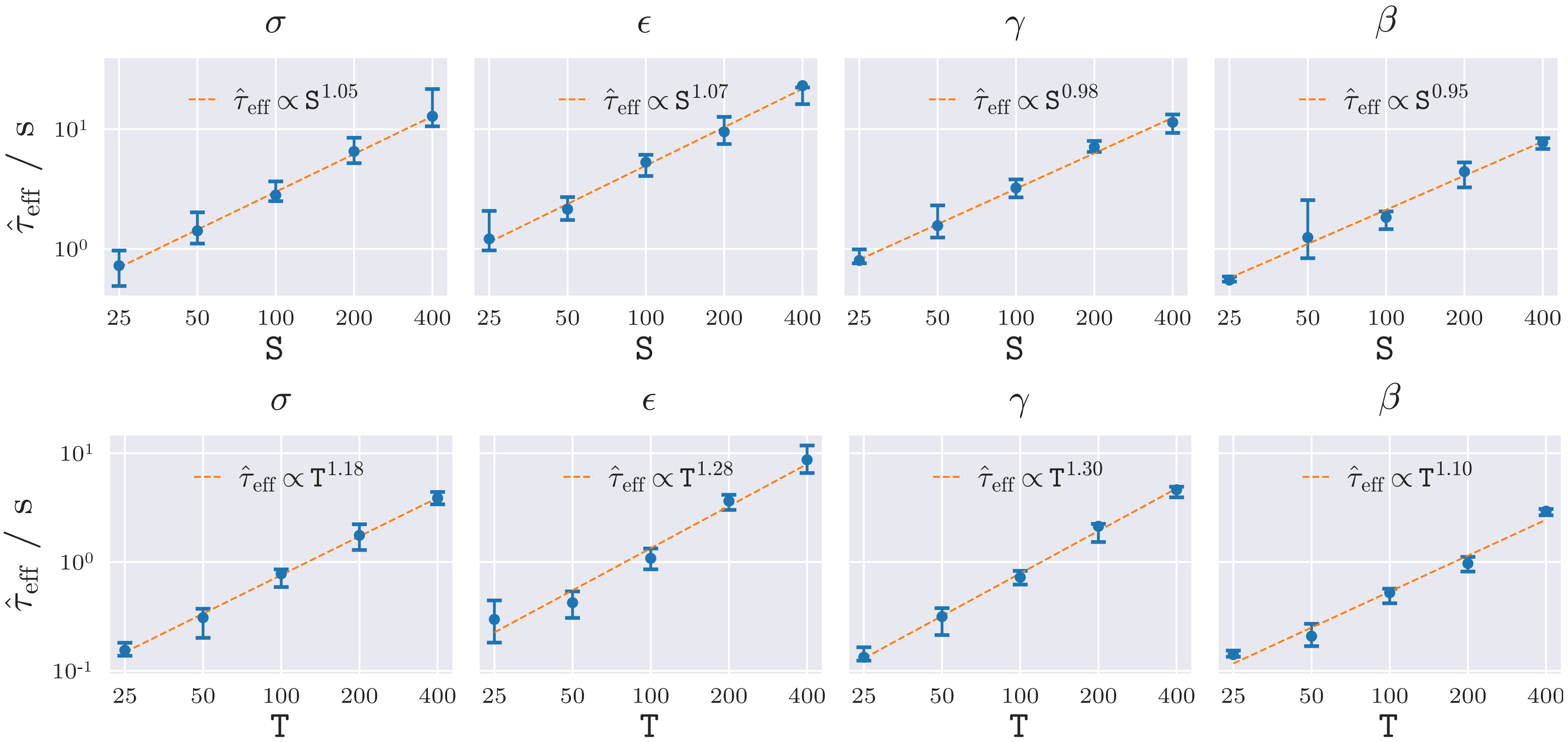


Compute time per integrator step

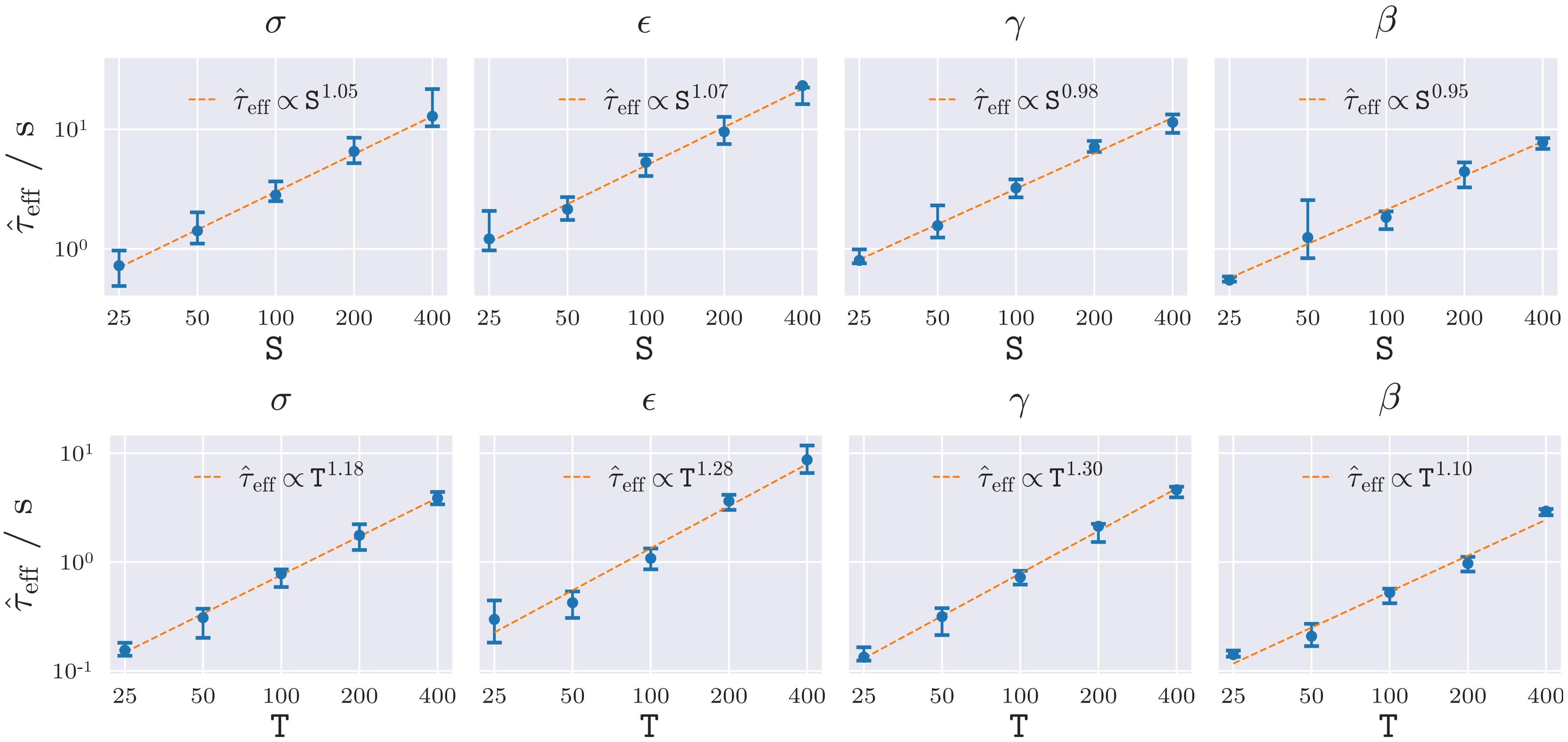


Matches with expected $\mathcal{O}(\mathtt{ST})$ scaling.

Compute time per effective sample

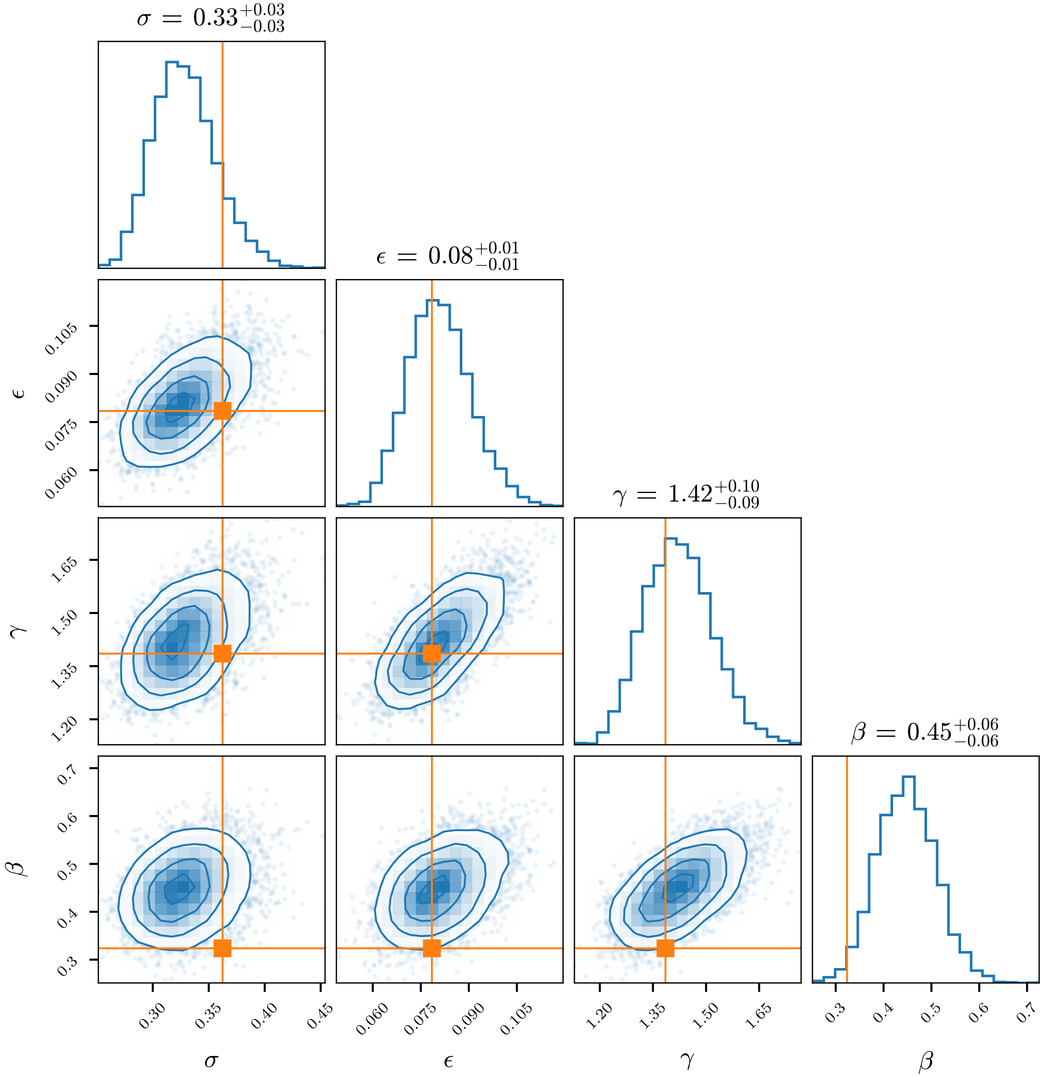


Compute time per effective sample



Cf. optimal scaling of $\mathcal{O}(\mathtt{D}^{1.25})$ for HMC in dimension $\mathtt{D}$ i.i.d. targets as $\mathtt{D} \to \infty$ (Beskos+, 2013).

Example posterior marginals $\mathtt{T} = 100$



Conclusions

- Framework for performing inference in partially observed diffusions with minimal assumptions required on model and discretisation scheme.

Conclusions

- Framework for performing inference in partially observed diffusions with minimal assumptions required on model and discretisation scheme.
- Jointly updating both parameters and latent process using a gradient-based constrained HMC method leads to rapidly mixing chains.

Conclusions

- Framework for performing inference in partially observed diffusions with minimal assumptions required on model and discretisation scheme.
- Jointly updating both parameters and latent process using a gradient-based constrained HMC method leads to rapidly mixing chains.
- By exploiting Markovian nature of model remains efficient for large numbers of observation times and dense time discretisations.

Thanks for listening!

Preprint arxiv.org/abs/1912.02982

Code git.io/m-mcmc

References

- H. C. Andersen. RATTLE: A 'velocity' version of the SHAKE algorithm for molecular dynamics calculations. *Journal of Comp. Physics*, 1983.
- G. O. Roberts and O. Stramer. On inference for partially observed non-linear diffusion models using the Metropolis-Hastings algorithm. *Biometrika*, 2001.
- O. Elerian, S. Chib and N. Shephard. Likelihood inference for discretely observed nonlinear diffusions. *Econometrica*, 2001.
- O. Papaspiliopoulos, G. O. Roberts and M. Sköld. Non-centered parameterizations for hierarchical models and data augmentation. *Bayesian Statistics*, 2003.
- A. Beskos, N. Pillai, G. Roberts, J.M. Sanz-Serna and A. Stuart. Optimal tuning of the hybrid Monte Carlo algorithm. *Bernoulli*, 2013.
- S. Chib, M. K. Pitt and N. Shephard. Likelihood based inference for diffusion driven models. *Economic Papers - Nuffield College*, 2004.
- C. Hartmann and C. Schutte. A constrained hybrid Monte Carlo algorithm and the problem of calculating the free energy in several variables. *ZAMM-Zeitschrift für Angewandte Mathematik*, 2005.
- A. Golightly and D. J. Wilkinson. Bayesian sequential inference for non-linear multivariate diffusions. *Statistics and Computing*, 2006.
- P. Diaconis, S. Holmes and M. Shahshahani. Sampling from a Manifold. *Advances in Modern Statistical Theory and Applications*, 2013.
- M. A. Brubaker, M. Saelzmann and R. Urtasun. A family of MCMC methods on implicitly defined manifolds. *AISTATS*, 2012.
- M. M. Graham and A. J. Storkey. Asymptotically exact inference in differentiable generative models. *Electronic Journal of Statistics*, 2017.
- T. Lelièvre, M. Rousset and G. Stoltz. Hybrid Monte Carlo methods for sampling probability measures on submanifolds. *Numerische Mathematik*, 2019.
- K. X. Au, M. M. Graham and A. H. Thiery. Manifold lifting: scaling MCMC methods to the vanishing noise regime. *arXiv:2003.03950*, 2020.