Asymptotically exact inference in differentiable generative models

MATT GRAHAM (m.m.graham@nus.edu.sg) AMOS STORKEY (a.storkey@ed.ac.uk)

dx.doi.org/10.1214/17-EJS1340SI git.io/dgm

PROBLEM: INFERENCE IN GENERATIVE MODELS

Given: A generative model of

$\mathbf{x}$: observed variables $\in \mathcal{X}$,
$\mathbf{z}$: latent variables $\in \mathcal{Z}$,

where we can sample $(\mathbf{x}, \mathbf{z})$ pairs, but may not have access to a density $\mathrm{p}_{\mathbf{x},\mathbf{z}}$.

Task: Estimate conditional expectations

$$\mathbb{E}[f(\mathbf{z}) \mid \mathbf{x} = \check{\mathbf{x}}] = \int_{\mathcal{Z}} f(\mathbf{z})\, \mathrm{P}_{\mathbf{z}|\mathbf{x}}(\mathrm{d}\mathbf{z} \mid \check{\mathbf{x}}),$$

of the latent variables $\mathbf{z}$ given observed values $\check{\mathbf{x}}$ for the $\mathbf{x}$ variables.

DIFFERENTIABLE GENERATIVE MODELS

A generative model for $(\mathbf{x}, \mathbf{z})$ can be expressed in the form

$$\mathbf{u} \sim \mathrm{P}_{\mathbf{u}}, \quad \mathbf{z} = \mathrm{g}_{\mathbf{z}}(\mathbf{u}), \quad \mathbf{x} = \mathrm{g}_{\mathbf{x}}(\mathbf{u}),$$

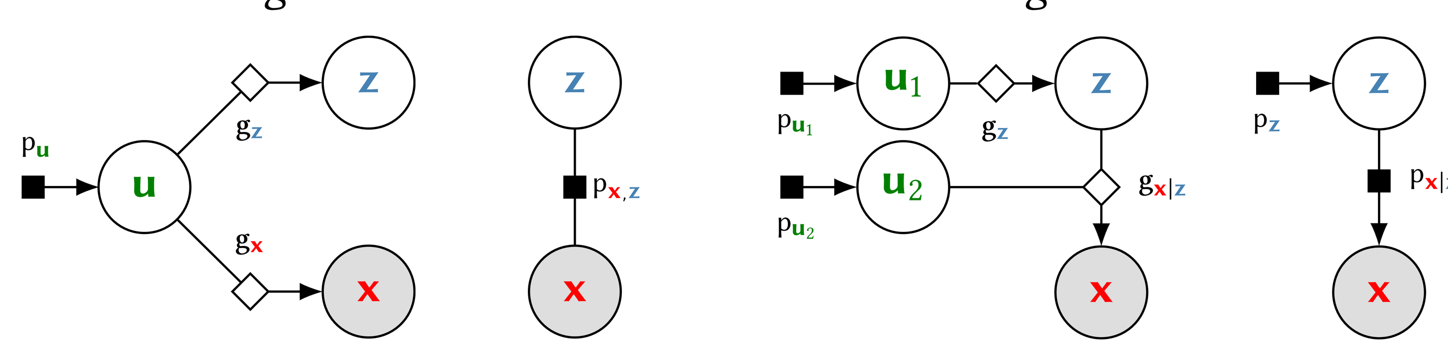
with $\mathbf{u} \in \mathcal{U}$ the *random inputs* to *generator functions* $\mathrm{g}_{\mathbf{z}}$ and $\mathrm{g}_{\mathbf{x}}$.

We define a *differentiable generative model* as further satisfying

- $\mathcal{U} = \mathbb{R}^{D_u}$, $\mathcal{Z} = \mathbb{R}^{D_z}$ and $\mathcal{X} = \mathbb{R}^{D_x}$: real-valued variables,
- $\mathrm{P}_{\mathbf{u}}$ has a density $\mathrm{p}_{\mathbf{u}}$ with respect to the Lebesgue measure,
- the gradient $\nabla \mathrm{p}_{\mathbf{u}}$ and Jacobian $\mathbf{J}_{\mathrm{g}_{\mathbf{x}}}$ exist almost everywhere.

UNDIRECTED AND DIRECTED GENERATIVE MODELS

Undirected generative model *Directed generative model*

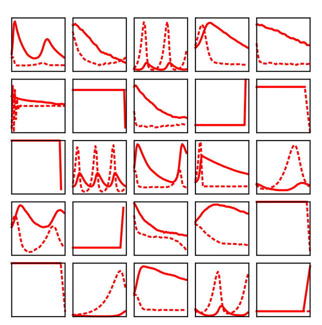


$\mathbf{z} = \mathrm{g}_{\mathbf{z}}(\mathbf{u})$ $\mathbf{x} = \mathrm{g}_{\mathbf{x}}(\mathbf{u})$ $\mathbf{z} = \mathrm{g}_{\mathbf{z}}(\mathbf{u}_1)$ $\mathbf{x} = \mathrm{g}_{\mathbf{x}|\mathbf{z}}(\mathbf{z}, \mathbf{u}_2)$

○ Variable node ■ Probabilistic factor ◇ Deterministic factor

EXAMPLES OF DIFFERENTIABLE GENERATIVE MODELS

For all the example models here $\mathrm{p}_{\mathbf{u}}(\mathbf{u}) = \mathcal{N}(\mathbf{u} \mid \mathbf{0}, \mathbf{I})$.

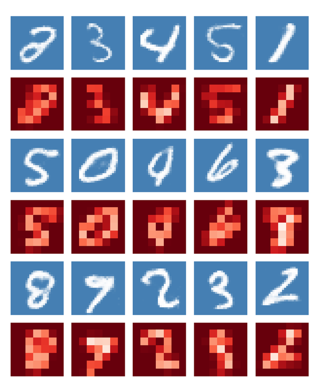


Stochastic Lotka-Volterra predator-prey population model,

prey population: $\mathrm{d}x_1 = (z_1 x_1 - z_2 x_1 x_2)\mathrm{d}t + \mathrm{d}n_1$,
predator population: $\mathrm{d}x_2 = (z_4 x_1 x_2 - z_3 x_2)\mathrm{d}t + \mathrm{d}n_2$.

Simulator for system can be expressed as a directed model

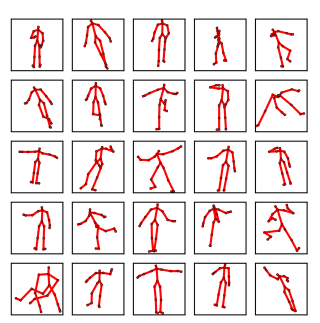
$\mathbf{z} = \mathrm{g}_{\mathbf{z}}(\mathbf{u}_1) = \exp(\sigma \odot \mathbf{u}_1 + \mu)$: sample parameters from prior,
$\mathbf{x} = \mathrm{g}_{\mathbf{x}|\mathbf{z}}(\mathbf{z}, \mathbf{u}_2)$: Euler-Maruyama integration of SDEs.



Generative model of monochrome digit images $\mathbf{z}$

$$\mathbf{u} = (\mathbf{u}_h, \mathbf{u}_n), \quad \mathbf{z} = \mathrm{f}_{\text{sigmoid}}(\mathbf{m}(\mathbf{u}_h) + \mathbf{s}(\mathbf{u}_h) \odot \mathbf{u}_n)$$

with $\mathbf{m}$ and $\mathbf{s}$ parameteric functions trained to match $\mathrm{P}_{\mathbf{z}}$ to the distribution of a dataset e.g. MNIST, with blurred and downsampled observed images $\mathbf{x} = \mathbf{D}\mathbf{z}$.



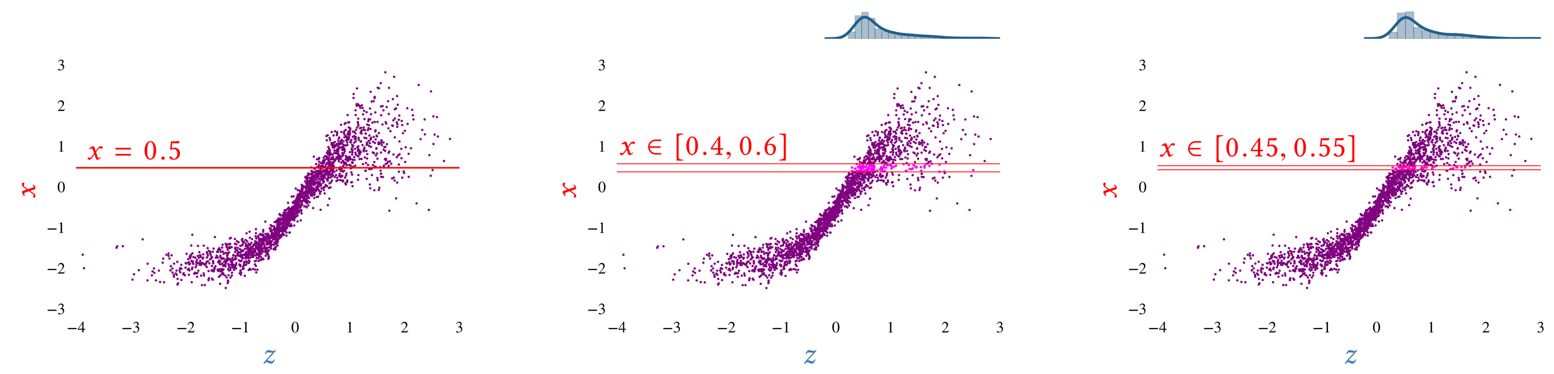
Pose generator, with learnt 3D scene variable generators

$$\mathbf{u} = (\mathbf{u}_a, \mathbf{u}_b, \mathbf{u}_c, \mathbf{u}_{1:J}), \quad \overset{\text{joint angles}}{\mathbf{z}_a} = \mathbf{f}_a(\mathbf{u}_a), \quad \overset{\text{bone lengths}}{\mathbf{z}_b} = \mathbf{f}_b(\mathbf{u}_b), \quad \overset{\text{camera parameters}}{\mathbf{z}_c} = \mathbf{f}_c(\mathbf{u}_c)$$

and a camera model to generate observed 2D joint positions

$$\overset{\text{2D proj.}}{\mathbf{x}_j} = \overset{\text{camera matrix}}{\mathbf{C}(\mathbf{z}_c)}\ \overset{\text{3D pos.}}{\mathbf{r}_j(\mathbf{z}_a, \mathbf{z}_b)} + \overset{\text{obs. noise}}{\sigma\, \mathbf{u}_j} \quad \forall j \in \{1 \dots J\}.$$

APPROXIMATE BAYESIAN COMPUTATION (ABC)



Family of approximate inference methods for generative models.

Key idea: observations $\check{\mathbf{x}}$ are decoupled from simulated observed variables $\mathbf{x}$ by a *kernel* $\mathrm{p}_{\check{\mathbf{x}}|\mathbf{x}}(\check{\mathbf{x}} \mid \mathbf{x}) = k_\epsilon(\check{\mathbf{x}} \mid \mathbf{x})$ with $\lim_{\epsilon \to 0} k_\epsilon(\check{\mathbf{x}} \mid \mathbf{x}) = \delta(\check{\mathbf{x}} - \mathbf{x})$, e.g.

$$k_\epsilon(\check{\mathbf{x}} \mid \mathbf{x}) \propto \mathbb{1}(|\check{\mathbf{x}} - \mathbf{x}| < \epsilon) \text{ (uniform ball)}, \quad k_\epsilon(\mathbf{x} \mid \mathbf{x}) = \mathcal{N}\left(\check{\mathbf{x}} \mid \mathbf{x}, \epsilon^2 \mathbf{I}\right) \text{ (Gaussian)}.$$

Kernel can be used to express approximate conditional expectations

$$\mathbb{E}[f(\mathbf{z}) \mid \check{\mathbf{x}} = \check{\mathbf{x}};\, \epsilon] = \frac{1}{\mathrm{p}_{\check{\mathbf{x}}}(\check{\mathbf{x}})} \int_{\mathcal{X} \times \mathcal{Z}} f(\mathbf{z})\, k_\epsilon(\check{\mathbf{x}} \mid \mathbf{x})\, \mathrm{P}_{\mathbf{x},\mathbf{z}}(\mathrm{d}\mathbf{x}, \mathrm{d}\mathbf{z}),$$

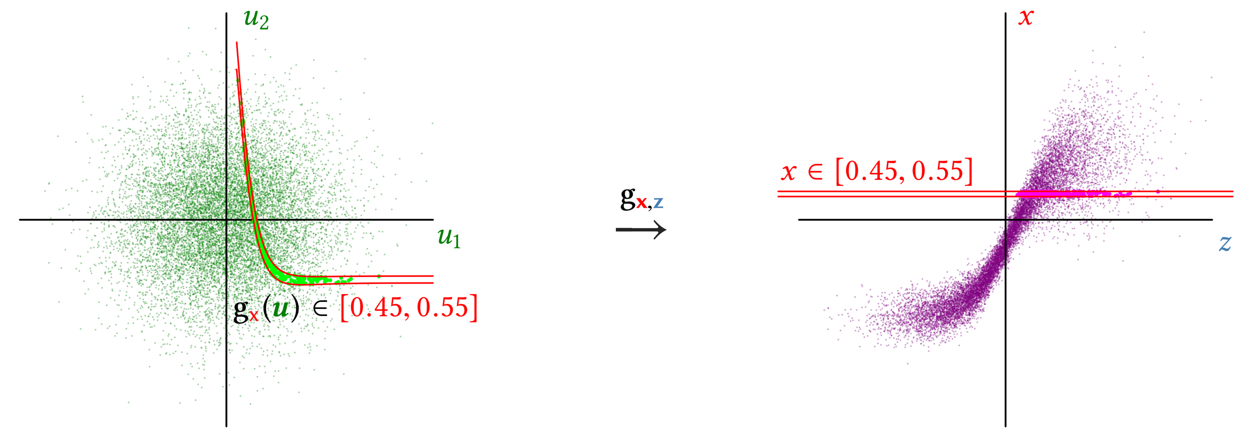
with $\lim_{\epsilon \to 0} \mathbb{E}[f(\mathbf{z}) \mid \check{\mathbf{x}} = \check{\mathbf{x}};\, \epsilon] = \mathbb{E}[f(\mathbf{z}) \mid \mathbf{x} = \check{\mathbf{x}}]$.

Typically a further approximation is made of conditioning only on reduced-dimensionality summary statistics of observed data $\mathbf{s}(\check{\mathbf{x}}) \in \mathcal{S}$

$$\mathbb{E}[f(\mathbf{z}) \mid \mathbf{s} = \mathbf{s}(\check{\mathbf{x}});\, \epsilon] = \frac{1}{\mathrm{p}_{\mathbf{s}}(\mathbf{s}(\check{\mathbf{x}}))} \int_{\mathcal{X} \times \mathcal{Z}} f(\mathbf{z})\, k_\epsilon(\mathbf{s}(\check{\mathbf{x}}) \mid \mathbf{s}(\mathbf{x}))\, \mathrm{P}_{\mathbf{x},\mathbf{z}}(\mathrm{d}\mathbf{x}, \mathrm{d}\mathbf{z}),$$

with in general $\lim_{\epsilon \to 0} \mathbb{E}[f(\mathbf{z}) \mid \mathbf{s} = \mathbf{s}(\check{\mathbf{x}});\, \epsilon] \neq \mathbb{E}[f(\mathbf{z}) \mid \mathbf{x} = \check{\mathbf{x}}]$.

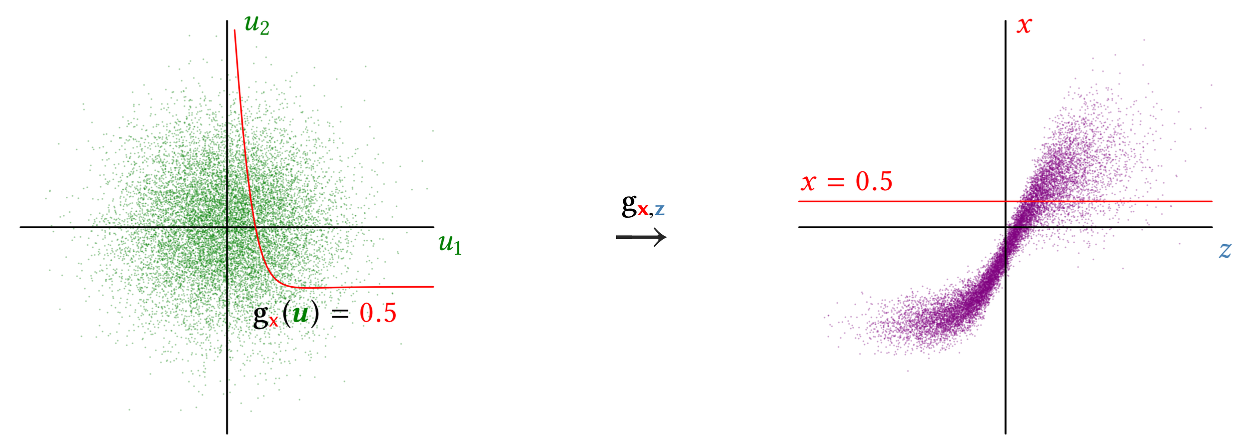
INFERENCE IN THE GENERATOR INPUT SPACE



We can rewrite the ABC conditional expectation as an integral over the input space

$$\mathbb{E}[f(\mathbf{z}) \mid \check{\mathbf{x}} = \check{\mathbf{x}};\, \epsilon] = \frac{1}{\mathrm{p}_{\check{\mathbf{x}}}(\check{\mathbf{x}})} \int_{\mathcal{U}} f \circ \mathrm{g}_{\mathbf{z}}(\mathbf{u})\, k_\epsilon(\check{\mathbf{x}} \mid \mathrm{g}_{\mathbf{x}}(\mathbf{u}))\, \mathrm{p}_{\mathbf{u}}(\mathbf{u})\, \mathrm{d}\mathbf{u}.$$

We can therefore perform ABC inference by constructing a MCMC transition operator in input space with target density $\pi_\epsilon(\mathbf{u}) \propto k_\epsilon(\check{\mathbf{x}} \mid \mathrm{g}_{\mathbf{x}}(\mathbf{u}))\, \mathrm{p}_{\mathbf{u}}(\mathbf{u})$.



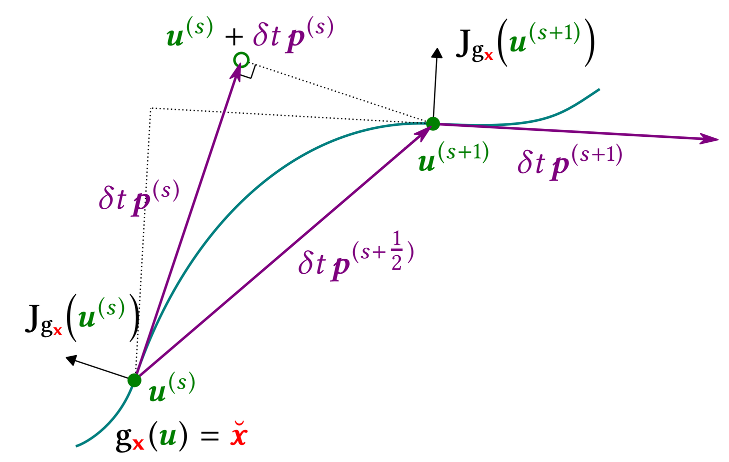
By taking the $\epsilon \to 0$ limit of the above and applying *Federer's Co-Area Formula* the exact conditional expectation can be expressed as an integral over an implicitly defined manifold $\mathrm{g}_{\mathbf{x}}^{-1}(\check{\mathbf{x}}) = \{\mathbf{u} \in \mathcal{U} : \mathrm{g}_{\mathbf{x}}(\mathbf{u}) = \check{\mathbf{x}}\}$ embedded in $\mathcal{U}$ [1]

$$\mathbb{E}[f(\mathbf{z}) \mid \mathbf{x} = \check{\mathbf{x}}] = \frac{1}{\mathrm{p}_{\mathbf{x}}(\check{\mathbf{x}})} \int_{\mathrm{g}_{\mathbf{x}}^{-1}(\check{\mathbf{x}})} f \circ \mathrm{g}_{\mathbf{z}}(\mathbf{u}) \left|\mathbf{J}_{\mathrm{g}_{\mathbf{x}}}(\mathbf{u})\mathbf{J}_{\mathrm{g}_{\mathbf{x}}}(\mathbf{u})^{\mathsf{T}}\right|^{-\frac{1}{2}} \mathrm{p}_{\mathbf{u}}(\mathbf{u})\, \mathcal{H}^{D_u - D_x}(\mathrm{d}\mathbf{u}).$$

Therefore we can perform inference by constructing an MCMC operator with stationary distribution with density $\pi(\mathbf{u}) \propto \left|\mathbf{J}_{\mathrm{g}_{\mathbf{x}}}(\mathbf{u})\mathbf{J}_{\mathrm{g}_{\mathbf{x}}}(\mathbf{u})^{\mathsf{T}}\right|^{-\frac{1}{2}} \mathrm{p}_{\mathbf{u}}(\mathbf{u})$ on $\mathrm{g}_{\mathbf{x}}^{-1}(\check{\mathbf{x}})$.

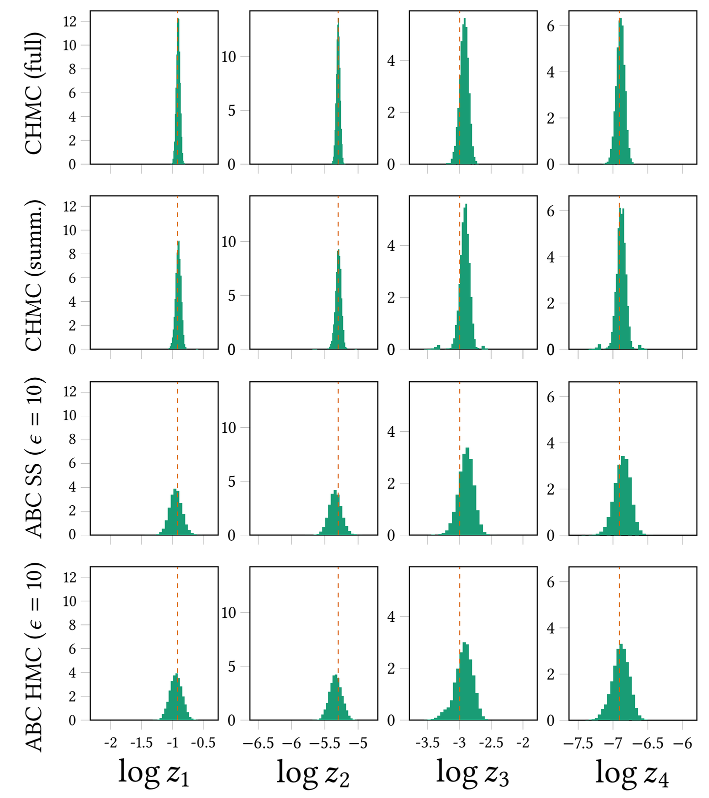
CONSTRAINED HAMILTONIAN MONTE CARLO (CHMC)

Numerically simulating a constrained Hamiltonian dynamic with a symplectic integrator [2] defines a reversible and measure-preserving flow map on an implicitly defined constraint manifold. This can be used to construct a MCMC transition leaving a target distribution on the manifold invariant [3].



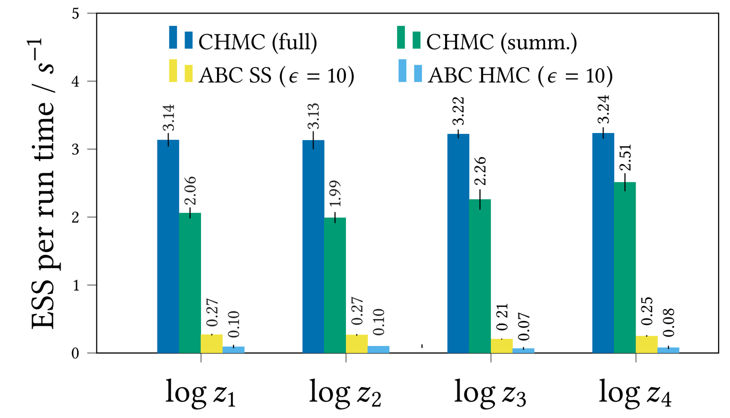
NUMERICAL EXPERIMENTS

Lotka-Volterra parameter inference

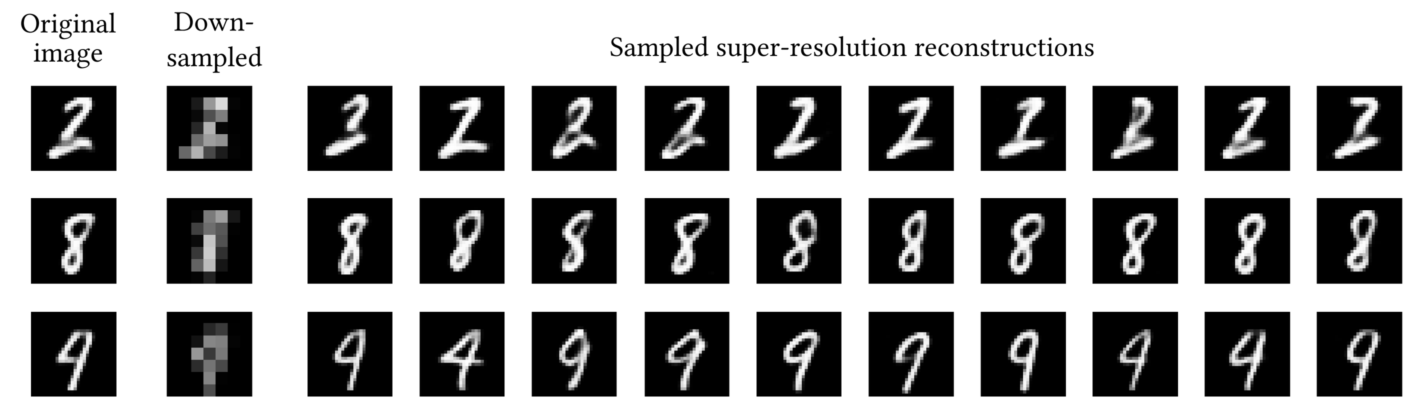


Prey and predator populations observed at 50 timepoints each ($D_x = 100$) with task to infer four system parameters $z_{1:4}$. Four MCMC methods compared

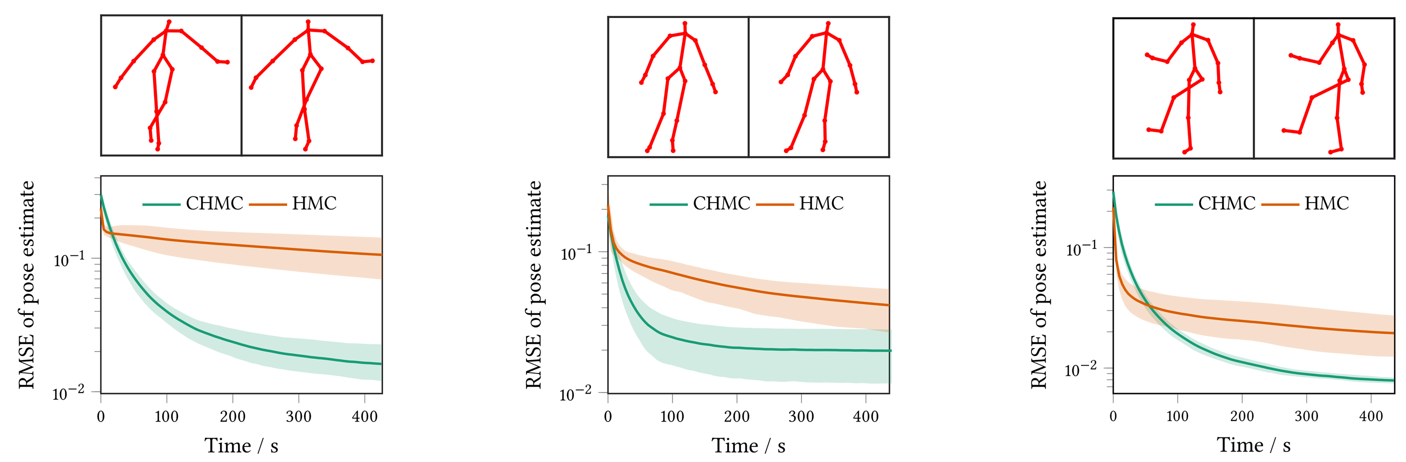
- CHMC (full) - constrained HMC conditioning on full data,
- CHMC (summ.) - constrained HMC conditioning on nine-dimensional summary statistics of data,
- ABC SS - elliptical slice sampling in generator input space with Gaussian kernel $\epsilon = 10$ and conditioning on full data,
- ABC HMC - unconstrained HMC in generator input space with Gaussian kernel $\epsilon = 10$ and conditioning on full data.



Digit image super-resolution



Binocular 3D pose estimation



CONCLUSIONS

We propose a method for inference in *differentiable generative models*.

The approach is an asymptotically exact alternative to ABC where applicable: computationally tractable inference with $\epsilon \to 0$ and full observed data.

Key idea is to view conditioning as constraining the inputs to a generator function.

We use constrained HMC to efficiently explore a target distribution on the constraint manifold corresponding to inputs consistent with observations.

REFERENCES

1. H. C. Andersen. RATTLE: A 'velocity' version of the SHAKE algorithm for molecular dynamics calculations. *J. Comput. Physics*, 1983.
2. P. Diaconis, S. Holmes and M. Shahshahani. Sampling from a Manifold. *Advances in Modern Statistical Theory and Applications*, 2013.
3. M. A. Brubaker, M. Salzmann, and R. Urtasun. A family of MCMC methods on implicitly defined manifolds. *AISTATS*, 2012.

This work was supported in part by grants EP/F500385/1 and BB/F529254/1 for the University of Edinburgh School of Informatics Doctoral Training Centre in Neuroinformatics and Computational Neuroscience (www.anc.ac.uk/dtc) from the UK Engineering and Physical Sciences Research Council (EPSRC), UK Biotechnology and Biological Sciences Research Council (BBSRC), and the UK Medical Research Council (MRC).