Discontinuous Hamiltonian Monte Carlo for discrete parameters and discontinuous likelihoods

Akihiko Nishimura
Department of Biomathematics, University of California - Los Angeles
David Dunson
Department of Statistical Science, Duke University
Jianfeng Lu
Department of Mathematics, Duke University

September 10, 2018

Abstract

Hamiltonian Monte Carlo (HMC) is a powerful sampling algorithm employed by several probabilistic programming languages. Its fully automatic implementations have made HMC a standard tool for applied Bayesian modeling. While its performance is often superior to alternatives under a wide range of models, one prominent weakness of HMC is its inability to handle discrete parameters. In this article, we present *discontinuous HMC*, an extension that can efficiently explore discrete spaces involving ordinal parameters as well as target distributions with discontinuous densities. The proposed algorithm is based on two key ideas: embedding of discrete parameters into a continuous space and simulation of Hamiltonian dynamics on a piecewise smooth density function. When properly-tuned, discontinuous HMC is guaranteed to outperform a Metropolis-within-Gibbs algorithm as the two algorithms coincide under a specific (and sub-optimal) implementation of discontinuous HMC. It is additionally shown that the dynamics underlying discontinuous HMC have a remarkable similarity to a zig-zag process, a continuous-time Markov process behind a state-of-the-art non-reversible rejection-free sampler. We apply our algorithm to challenging posterior inference problems to demonstrate its wide applicability and competitive performance.

Keywords: Bayesian inference, event-driven Monte Carlo, integrator, Markov chain Monte Carlo, numerical methods, rejection-free

1 Introduction

Markov chain Monte Carlo (MCMC) is routinely used for Bayesian inference to generate samples from posterior distributions. While specialized MCMC algorithms exist for restricted model classes, most generic MCMC algorithms — adaptable to a broad range of posterior distributions — are often highly inefficient and scale poorly in the number of parameters. Originally proposed by Duane et al. (1987) and more recently popularized in the statistics community through the works of Neal (1996, 2010), Hamiltonian Monte Carlo (HMC) promises a better scalability (Neal, 2010; Beskos et al., 2013) and has enjoyed wide-ranging successes as one of the most reliable approaches in general settings (Gelman et al., 2013; Kruschke, 2014; Monnahan et al., 2016). Probabilistic programming languages Stan and PyMC rely on HMC and its variants to carry out posterior inferences automatically and have made HMC a standard tool for applied Bayesian modeling (Stan Development Team, 2016; Salvatier et al., 2016).

Often quoted as the main limitation of HMC is its lack of support for discrete parameters (Gelman et al., 2015; Monnahan et al., 2016). Inference problems involving discrete parameters come up in a wide range of fields, including ecology, social science, system reliability, and epidemiology (Berger et al., 2012; Schwarz and Seber, 1999; Warren and Warren, 2013; Basu and Ebrahimi, 2001; Parkin and Bray, 2009). The difficulty in extending HMC to a discrete parameter space stems from the fact that the construction of HMC proposals relies on a numerical solution of a differential equation. The use of a surrogate continuous target distribution may be possible in some special cases (Zhang et al., 2012), but approximating a discrete parameter of a likelihood by a continuous one is difficult in general (Berger et al., 2012).

This article presents *discontinuous HMC* (DHMC), an extension that can efficiently explore discrete spaces involving ordinal discrete parameters as well as continuous ones. More generally, the algorithm is well-suited whenever the discrete space has a natural ordering in a sense that two neighboring parameters imply similar log-likelihoods (see Section 2.1 for further explanation). DHMC can also handle discontinuous posterior densities, which for example arise from models with structural change points (Chib, 1998; Wagner et al., 2002), latent thresholds (Neelon and Dunson, 2004; Nakajima and West, 2013), and pseudo-likelihoods (Bissiri et al., 2016). DHMC retains the generality that makes HMC suitable for automatic posterior inference as in Stan and PyMC. For any given target distribution, each iteration of DHMC only requires evaluations of the density and of the following quantities:

1. full conditional densities of discrete parameters (up to normalizing constants)

2. either the gradient of the log density with respect to continuous parameters or their individual full conditional densities (up to normalizing constants)

Evaluations of full conditionals can be done algorithmically and efficiently through directed acyclic graph frameworks, taking advantage of conditional independence structures (Lunn et al., 2009). Algorithmic evaluation of the gradient is also efficient (Griewank and Walther, 2008) and its implementations are widely available as open-source modules (Carpenter et al., 2015).

In our framework, the discrete parameters of a model are first embedded into a continuous space, inducing parameters with piecewise constant densities. The key theoretical insight is that Hamiltonian dynamics with a discontinuous potential energy can be integrated analytically near its discontinuity in a way that exactly preserves the total energy. This fact was realized by Pakman and Paninski (2013) and used to sample from binary distributions through embedding them into a continuous space. Afshar and Domke (2015) explored, to a limited extent, applying the same idea in extending HMC more generally. These algorithms are instances of *event-driven Monte Carlo* in the computational physics literature, dating back to Alder and Wainwright (1959).

Another recent development in extending HMC to a more complex parameter space is Dinh et al. (2017). Though related, their work and ours have little overlap in terms of the main contributions. Their work is more of a proof of concept to demonstrate how HMC could be extended to a parameter space involving trees. We instead focus on the issue of discrete parameters and develop techniques not only of theoretical interest but also of immediate utility, demonstrating concrete and significant improvements over existing approaches.

Several novel techniques are introduced to turn the basic idea of event-driven HMC into a general and practical sampling algorithm for discrete parameters and, more generally, target distributions with discontinuous densities. We propose a product Laplace distribution for the momentum variable as a more effective alternative to the usual Gaussian distribution in dealing with discontinuous target distributions. We develop an efficient integrator of Hamiltonian dynamics based on a Laplace momentum by splitting the differential operator into its coordinate-wise components. This integrator exactly preserves the Hamiltonian and leads to a type of *rejection-free* Markovian transitions (Peters and de With, 2012). When applying only one step of the proposed integrator, DHMC coincides with a Metropolis-within-Gibbs sampler. This fact guarantees the theoretical superiority of properly tuned DHMC. DHMC indeed outperforms a Metropolis-within-Gibbs in practice because DHMC can take advantage of the momentum information and induce large transition moves by taking multiple integration steps.

The rest of the paper is organized as follows. We start Section 2 by describing our strategy for embedding discrete parameters into a continuous space in such a way that the resulting distribution can be explored efficiently by DHMC. We then develop a general technique for handling discontinuous target densities within the HMC framework based on the idea of event-driven Monte Carlo. In Section 3, we discuss the shortcomings of a Gaussian momentum in extending HMC to discontinuous targets and propose a Laplace momentum as a more effective alternative. Section 4 examines theoretical properties of DHMC with a Laplace momentum and establishes its connections to Metropolis-within-Gibbs and a zig-zag sampler. Section 5 presents simulation results with real data examples to demonstrate that DHMC outperforms alternative approaches. Well-documented codes to reproduce the simulation results are available at `https://github.com/aki-nishimura/discontinuous-hmc`.

2 Event-driven HMC for discrete parameters

Given a parameter $\boldsymbol{\theta} \sim \pi_\Theta(\cdot)$ of interest, HMC introduces an auxiliary *momentum* variable $\boldsymbol{p}$ and samples from a joint distribution $\pi(\boldsymbol{\theta}, \boldsymbol{p}) = \pi_\Theta(\boldsymbol{\theta})\pi_P(\boldsymbol{p})$ for some symmetric distribution

$\pi_P(\boldsymbol{p}) \propto \exp(-K(\boldsymbol{p}))$. More generally, the distribution of the momentum can depend on $\boldsymbol{\theta}$ (see Girolami and Calderhead, 2011 for example), but here we restrict our discussion to the momentum distributions independent of $\boldsymbol{\theta}$. The function $K(\boldsymbol{p})$ is referred to as the *kinetic energy* and $U(\boldsymbol{\theta}) = -\log \pi_\Theta(\boldsymbol{\theta})$ as the *potential energy* due to the physical laws that motivate HMC. The total energy $H(\boldsymbol{\theta}, \boldsymbol{p}) = U(\boldsymbol{\theta}) + K(\boldsymbol{p})$ is often called the *Hamiltonian*. HMC generates a proposal by simulating trajectories of *Hamiltonian dynamics* where the evolution of the state $(\boldsymbol{\theta}, \boldsymbol{p})$ is governed by *Hamilton's equations*:

$$\frac{\mathrm{d}\boldsymbol{\theta}}{\mathrm{d}t} = \nabla_{\boldsymbol{p}} K(\boldsymbol{p}), \quad \frac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}t} = -\nabla_{\boldsymbol{\theta}} U(\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} \log \pi_\Theta(\boldsymbol{\theta}). \tag{1}$$

Though the differential equation (1) makes no sense when $\boldsymbol{\theta}$ is discrete, this section shows how to extend the HMC framework to accommodate discrete parameters.

2.1 Embedding discrete parameters into a continuous space

Let N denote a discrete parameter with prior distribution $\pi_N(\cdot)$ and assume for now that N takes positive integer values $\{1, 2, 3, \ldots\}$. For example, the inference goal may be estimation of the population size N given the observation $y \,|\, q, N \sim \text{Binom}(q, N)$ and the objective prior $\pi_N(N) \propto N^{-1}$ (Berger et al., 2012). We embed N into a continuous space by introducing a latent parameter $\widetilde{N}$ whose relationship with N is defined to be

$$N = n \quad \text{if and only if} \quad \widetilde{N} \in (a_n, a_{n+1}] \tag{2}$$

for an increasing sequence of real numbers $0 = a_1 \leq a_2 \leq a_3 \leq \ldots$. To match the prior distribution of N, the corresponding (piecewise constant) prior density of $\widetilde{N}$ is given by

$$\pi_{\widetilde{N}}(\tilde{n}) = \sum_{n \geq 1} \frac{\pi_N(n)}{a_{n+1} - a_n} \mathbb{1}\{a_n < \tilde{n} \leq a_{n+1}\} \tag{3}$$

where the Jacobian-like factor $(a_{n+1} - a_n)^{-1}$ adjusts for embedding into non-uniform intervals.

Although the choice $a_n = n$ for all n is an obvious one, a non-uniform embedding is useful in effectively carrying out a parameter transformation of N. For example, a log-transform embedding $a_n = \log n$ substantially improves the mixing of DHMC when the target distribution is a heavy-tailed function of N or when $\log N$ has weaker correlations with the rest of the parameters. Similar parameter transformations for continuous parameters are common techniques when using the standard HMC and are also applicable to DHMC.

2.2 How HMC fails on discontinuous target densities

We first introduce terminology from numerical analysis, useful in our discussion of HMC and DHMC. An *integrator* is an algorithm that numerically approximates an evolution of the exact solution to a differential equation. An HMC variant require *reversible* and *volume-preserving* integrators to guarantee the symmetry of its proposal distribution (see Section 4.1 and Neal, 2010).

HMC is a class of Metropolis algorithms (Metropolis et al., 1953) whose proposal distributions $(\boldsymbol{\theta}, \boldsymbol{p}) \to (\boldsymbol{\theta}^*, \boldsymbol{p}^*)$ are generated as follows:

1. Sample the momentum from its marginal distribution $\boldsymbol{p} \sim \pi_P(\cdot)$.

2. Using a reversible and volume-preserving integrator, approximate $(\boldsymbol{\theta}(t), \boldsymbol{p}(t))_{t \geq 0}$ — the solution of the differential equation (1) with the initial condition $(\boldsymbol{\theta}(0), \boldsymbol{p}(0)) = (\boldsymbol{\theta}, \boldsymbol{p})$. Use the approximate solution $(\boldsymbol{\theta}^*, \boldsymbol{p}^*) \approx (\boldsymbol{\theta}(\tau), \boldsymbol{p}(\tau))$ for some $\tau > 0$ as a proposal.

The proposal $(\boldsymbol{\theta}^*, \boldsymbol{p}^*)$ then is accepted with the usual acceptance probability

$$\min \left\{ 1, \exp \left(- H(\boldsymbol{\theta}^*, \boldsymbol{p}^*) + H(\boldsymbol{\theta}, \boldsymbol{p}) \right) \right\} \tag{4}$$

where $H(\boldsymbol{\theta}, \boldsymbol{p}) = -\log \pi(\boldsymbol{\theta}, \boldsymbol{p})$ denotes the Hamiltonian as before. With an accurate integrator, the acceptance probability of $(\boldsymbol{\theta}^*, \boldsymbol{p}^*)$ can be close to 1 because of the *conservation of energy property* of Hamiltonian dynamics: $H(\boldsymbol{\theta}(t), \boldsymbol{p}(t)) = H(\boldsymbol{\theta}(0), \boldsymbol{p}(0))$ for all $t \geq 0$ for the exact solution. The standard integrator for HMC is the *leapfrog* scheme in which the evolution $(\boldsymbol{\theta}(t), \boldsymbol{p}(t)) \to (\boldsymbol{\theta}(t+\epsilon), \boldsymbol{p}(t+\epsilon))$ is approximated by the following successive update equations:

$$\boldsymbol{p} \leftarrow \boldsymbol{p} - \frac{\epsilon}{2} \nabla_{\boldsymbol{\theta}} U(\boldsymbol{\theta}), \quad \boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \epsilon \nabla_{\boldsymbol{p}} K(\boldsymbol{p}), \quad \boldsymbol{p} \leftarrow \boldsymbol{p} - \frac{\epsilon}{2} \nabla_{\boldsymbol{\theta}} U(\boldsymbol{\theta}) \tag{5}$$

When $\pi_\Theta(\cdot)$ is smooth, approximating the evolution $(\boldsymbol{\theta}(0), \boldsymbol{p}(0)) \to (\boldsymbol{\theta}(\tau), \boldsymbol{p}(\tau))$ with $L = \lfloor \tau / \epsilon \rfloor$ leapfrog steps results in an error of order $O(\epsilon^2)$ so that $H(\boldsymbol{\theta}^*, \boldsymbol{p}^*) = H(\boldsymbol{\theta}, \boldsymbol{p}) + O(\epsilon^2)$. When $\pi_\Theta(\cdot)$ has a discontinuity, however, the leapfrog updates (5) completely fail to account for the instantaneous change in $\pi_\Theta(\cdot)$ and in general incur an error of order $O(1)$. The standard HMC implementation therefore comes with no guarantee of a decent acceptance rate when the parameter of interest $\boldsymbol{\theta}$ has a discontinuous density.

2.3 Event-driven approach at discontinuity

Typical integrators, including the leapfrog scheme, are designed for differential equations with smooth time derivatives. If the discontinuity boundaries of its potential energy can be detected, however, Hamiltonian dynamics can be integrated so as to account for the instantaneous change in $\pi_\Theta(\cdot)$. We now describe how to make sense of Hamiltonian dynamics corresponding to a discontinuous potential energy $U(\boldsymbol{\theta}) = -\log \pi_\Theta(\boldsymbol{\theta})$ whose discontinuity set forms a piecewise smooth manifold.

While a discontinuous function does not have a derivative in a classical sense, the gradient $\nabla U(\boldsymbol{\theta})$ can be defined through a notion of *distributional derivatives* and the corresponding Hamilton's equations (1) can be interpreted as a *measure-valued differential inclusion* (Stewart, 2000). A solution of a (measure-valued) differential inclusion problem is in general not unique unlike that of a smooth ordinary differential equation. To find the solution that preserves the critical properties of Hamiltonian dynamics, therefore we rely on a so-called *selection principle* (Ambrosio, 2008) and construct a solution with desirable properties as follows.

Define a sequence of smooth approximations $U_\delta(\boldsymbol{\theta})$ of $U(\boldsymbol{\theta})$ for $\delta > 0$ through the convolution $U_\delta = U * \phi_\delta$ with $\phi_\delta(\boldsymbol{\theta}) = \delta^{-d} \phi(\delta^{-1} \boldsymbol{\theta})$ for a compactly supported smooth function $\phi \geq 0$ such that $\int \phi = 1$. Here the integer d denotes the dimension of $\boldsymbol{\theta}$. Now let $(\boldsymbol{\theta}_\delta(t), \boldsymbol{p}_\delta(t))$ be the solution of Hamilton's equations with the potential energy U_δ. It can then be shown that the pointwise limit $(\boldsymbol{\theta}(t), \boldsymbol{p}(t)) = \lim_{\delta \to 0} (\boldsymbol{\theta}_\delta(t), \boldsymbol{p}_\delta(t))$ exists for almost every initial condition and we define the dynamics corresponding to $U(\boldsymbol{\theta})$ as this limit. This construction

in particular provides a rigorous mathematical foundation for the special cases of discontinuous Hamiltonian dynamics derived by Pakman and Paninski (2013) and Afshar and Domke (2015) through physical intuitions.

The behavior of the limiting dynamics near the discontinuity is deduced as follows. Suppose that the trajectory $(\boldsymbol{\theta}(t), \boldsymbol{p}(t))$ hits the discontinuity at an event time t_e and let t_e^- and t_e^+ denote infinitesimal moments before and after that. Since the discontinuity set of $U(\boldsymbol{\theta})$ was assumed to be piecewise smooth, at a discontinuity point $\boldsymbol{\theta}$ we have

$$\lim_{\delta \to 0} \nabla_{\boldsymbol{\theta}} U_\delta(\boldsymbol{\theta}) / \|\nabla_{\boldsymbol{\theta}} U_\delta(\boldsymbol{\theta})\| = \boldsymbol{\nu}(\boldsymbol{\theta}) \tag{6}$$

where $\boldsymbol{\nu}(\boldsymbol{\theta})$ denotes a unit vector orthonormal to the discontinuity boundary pointing in the direction of higher potential energy. More precisely, the preceding statement holds away from a set of measure zero, in which $\boldsymbol{\theta}$ belongs to an intersection of multiple discontinuity boundaries and does not have a well-defined orthonormal vector. The relations (6) and $\mathrm{d}\boldsymbol{p}_\delta/\mathrm{d}t = -\nabla_{\boldsymbol{\theta}} U_\delta$ imply that the only change in $\boldsymbol{p}(t)$ upon encountering the discontinuity occurs in the direction of $\boldsymbol{\nu}_e = \boldsymbol{\nu}(\boldsymbol{\theta}(t_e))$ i.e.

$$\boldsymbol{p}(t_e^+) = \boldsymbol{p}(t_e^-) - \gamma\, \boldsymbol{\nu}_e \tag{7}$$

for some $\gamma > 0$. There are two possible types of change in $\boldsymbol{p}$ depending on the potential energy difference ΔU_e at the discontinuity, which we formally define as

$$\Delta U_e = \lim_{\epsilon \to 0^+} U\left(\boldsymbol{\theta}(t_e) + \epsilon \boldsymbol{p}(t_e^-)\right) - U\left(\boldsymbol{\theta}(t_e^-)\right) \tag{8}$$

When the momentum does not provide enough kinetic energy to overcome the potential energy increase ΔU_e, the trajectory bounces back against the plane orthogonal to $\boldsymbol{\nu}_e$. Otherwise, the trajectory moves through the discontinuity by transferring kinetic energy to potential energy. Either way, the magnitude of an instantaneous change γ can be determined via the energy conservation law:

$$K(\boldsymbol{p}(t_e^+)) - K(\boldsymbol{p}(t_e^-)) = U(\boldsymbol{\theta}(t_e^-)) - U(\boldsymbol{\theta}(t_e^+)) \tag{9}$$

Figure 1 — which is explained in more details in Section 3 — provides a visual illustration of the trajectory behavior at a discontinuity.

3 Discontinuous HMC with Laplace momentum

3.1 Issue with Gaussian momentum

Most of the existing variations of HMC assume a (conditionally) Gaussian distribution on the momentum variable. It is a natural choice in the original molecular dynamics applications (Duane et al., 1987) and is arguably the most intuitive in terms of the underlying physics (Neal, 2010). Correspondingly, the existing event-driven HMC algorithms use a Gaussian momentum (Pakman and Paninski, 2013; Afshar and Domke, 2015).

For sampling discrete parameters, however, discontinuous Hamiltonian dynamics based on a Gaussian momentum have a serious shortcoming. In order to approximate the dynamics

accurately, an integrator must detect every single discontinuity encountered by a trajectory and then compute the potential energy difference each time (see supplement Section S1). To see why this is a serious problem, consider a discrete parameter $N \in \mathbb{Z}^+$ with a substantial posterior uncertainty, say $\mathrm{Var}(N \,|\, \mathrm{data}) \approx 1000^2$. We can then expect that a Metropolis move such as $N \to N + 1000$ would be accepted with a moderate probability, which costs us a single likelihood evaluation. On the other hand, if we were to sample a continuously embedded counter part $\widetilde{N}$ of N using DHMC with the Gaussian momentum based integrator of Algorithm 3, a transition of the corresponding magnitude would require *1000 likelihood evaluations*. Such a high computational cost for otherwise simple parameter updates makes the algorithm practically useless.

3.2 Hamiltonian dynamics based on Laplace momentum

The above example illustrates that, for discontinuous Hamiltonian dynamics to be of practical value in sampling discrete parameters, we need to be able to devise a reversible integrator that can jump through multiple discontinuities in a small number of target density evaluations while approximately preserving the total energy. It is not at all obvious if such an integrator exists for the dynamics based on a Gaussian momentum.

As we will show below, there is a simple solution to this problem if we instead employ an independent Laplace distribution $\pi_P(\boldsymbol{p}) \propto \prod_i \exp(-m_i^{-1}|p_i|)$. Similar momentum distributions have been considered within the traditional HMC framework based on the leapfrog integrator for improving numerical stability and mixing rate in certain settings (Zhang et al., 2016; Lu et al., 2016; Livingstone et al., 2016).

Hamilton's equation under the independent Laplace momentum is given by

$$\frac{\mathrm{d}\boldsymbol{\theta}}{\mathrm{d}t} = \boldsymbol{m}^{-1} \odot \mathrm{sign}(\boldsymbol{p}), \quad \frac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}t} = -\nabla_{\boldsymbol{\theta}} U(\boldsymbol{\theta}) \tag{10}$$

where $\odot$ denotes an element-wise multiplication. A unique characteristic of the dynamics (10) is that the time derivative of $\boldsymbol{\theta}(t)$ depends only on the signs of p_i's and not on their magnitudes. In particular, if we know that $p_i(t)$'s do not change their signs on the time interval $t \in [\tau, \tau + \epsilon]$, then we also know that

$$\boldsymbol{\theta}(\tau + \epsilon) = \boldsymbol{\theta}(\tau) + \epsilon\, \boldsymbol{m}^{-1} \odot \mathrm{sign}\left(\boldsymbol{p}(\tau)\right) \tag{11}$$

irrespective of the intermediate values $U(\boldsymbol{\theta}(t))$ along the trajectory $(\boldsymbol{\theta}(t), \boldsymbol{p}(t))$ for $t \in [\tau, \tau + \epsilon]$. Our integrator's ability to jump through multiple discontinuities of $U(\boldsymbol{\theta})$ in single target density evaluation depends critically on this property of the dynamics. While the value of $\boldsymbol{p}(\tau + \epsilon)$ is dependent on the intermediate values $U(\boldsymbol{\theta}(t))$, solving for this dependence is greatly simplified by splitting the differential operator of (10) into its coordinate-wise components as will be shown in the next section.

3.3 Integrator for Laplace momentum via operator splitting

Operator splitting is a technique to approximate the solution of a differential equation by decomposing it into components each of which can be solved more easily (McLachlan and

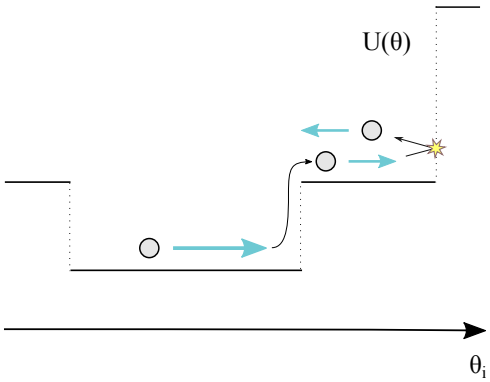

Figure 1: An example trajectory $\boldsymbol{\theta}(t)$ of discontinuous Hamiltonian dynamics. The trajectory has enough kinetic energy to move across the first discontinuity by transferring some of kinetic energy to potential energy. Across the second discontinuity, however, the trajectory has insufficient kinetic energy to compensate for the potential energy increase and bounces back as a result.

Quispel, 2002). Hamiltonian splitting methods more commonly found in the HMC literature are special cases (Neal, 2010). A convenient splitting scheme for (10) can be devised by considering the equation for each coordinate (θ_i, p_i) while the other parameters $(\boldsymbol{\theta}_{\text{-}i}, \boldsymbol{p}_{\text{-}i})$ are fixed:

$$\frac{\mathrm{d}\theta_i}{\mathrm{d}t} = m_i^{-1}\,\mathrm{sign}(p_i), \quad \frac{\mathrm{d}p_i}{\mathrm{d}t} = -\partial_{\theta_i} U(\boldsymbol{\theta}), \quad \frac{\mathrm{d}\boldsymbol{\theta}_{\text{-}i}}{\mathrm{d}t} = \frac{\mathrm{d}\boldsymbol{p}_{\text{-}i}}{\mathrm{d}t} = \mathbf{0} \tag{12}$$

There are two possible behaviors for the solution $(\boldsymbol{\theta}(t), \boldsymbol{p}(t))$ of (12) for $t \in [\tau, \tau+\epsilon]$, depending on the amount of the initial momentum $p_i(\tau)$. Let $\boldsymbol{\theta}^*(t)$ denote a potential path of $\boldsymbol{\theta}(t)$:

$$\theta_i^*(t) = \theta_i(\tau) + (t-\tau) m_i^{-1}\mathrm{sign}(p_i(\tau)), \quad \boldsymbol{\theta}_{\text{-}i}^*(t) = \boldsymbol{\theta}_{\text{-}i}(\tau) \tag{13}$$

In case the initial momentum is large enough that $m_i^{-1}|p_i(\tau)| > U\big(\boldsymbol{\theta}^*(t)\big) - U(\boldsymbol{\theta}(\tau))$ for all $t \in [\tau, \tau+\epsilon]$, we have

$$\boldsymbol{\theta}(\tau+\epsilon) = \boldsymbol{\theta}^*(\tau+\epsilon) = \boldsymbol{\theta}(\tau) + \epsilon\, m_i^{-1}\mathrm{sign}(p_i(\tau))\boldsymbol{e}_i \tag{14}$$

Otherwise, the momentum p_i flips (i.e. $p_i \leftarrow -p_i$) and the trajectory $\boldsymbol{\theta}(t)$ reverses its course at the event time t_e given by

$$t_e = \inf\left\{t \in [\tau, \tau+\epsilon] : U\big(\boldsymbol{\theta}^*(t)\big) - U(\boldsymbol{\theta}(\tau)) > K(\boldsymbol{p}(\tau))\right\} \tag{15}$$

See Figure 1 for a visual illustration of the trajectory $\boldsymbol{\theta}(t)$. By emulating the qualitative behavior of the solution $(\boldsymbol{\theta}(t), \boldsymbol{p}(t))$, we obtain an efficient integrator of the coordinate-wise equation (12) as given in Algorithm 1. While the parameter $\boldsymbol{\theta}$ does not get updated when $m_i^{-1}|p_i| < \Delta U$ (line 5), the momentum flip $p_i \leftarrow -p_i$ (line 9) ensures that the next numerical

integration step leads the trajectory toward a higher density of $\pi_\Theta(\boldsymbol{\theta})$.[1]

The solution of the original (unsplit) differential equation (10) is approximated by sequentially updating each coordinate of $(\boldsymbol{\theta}, \boldsymbol{p})$ with Algorithm 1 as illustrated in Figure 2. The reversibility of the resulting proposal is guaranteed by randomly permuting the order of the coordinate updates. (Alternatively, one can split the operator symmetrically to obtain a reversible integrator; for example, first update $\boldsymbol{\theta}$ in the order $\theta_1, \theta_2, \ldots, \theta_d$ and then in the order $\theta_d, \theta_{d-1}, \ldots, \theta_1$. See Section 4.1 as well as McLachlan and Quispel (2002) for more details.) The integrator does not reproduce the exact solution but nonetheless preserves the Hamiltonian exactly, yielding a rejection-free proposal. While this remains true with any stepsize ϵ, for good mixing the stepsize needs to be chosen small enough that the condition on Line 5 is satisfied with high probability; see Section 4.3 for further discussion on tuning ϵ.

Algorithm 1: Coordinate-wise integrator for dynamics with Laplace momentum

```
1  Function CoordIntegrator (θ, p, i, ε):
2      θ* ← θ
3      θ*_i ← θ*_i + ε m_i^{-1} sign(p_i)
4      ΔU ← U(θ*) − U(θ)
5      if m_i^{-1} |p_i| > ΔU  then
6          θ_i ← θ*_i
7          p_i ← p_i − sign(p_i) m_i ΔU
8      else
9          p_i ← −p_i
10     return θ, p
```

3.4 Mixing momentum distributions for continuous and discrete parameters

The potential energy $U(\boldsymbol{\theta})$ would be a smooth function of θ_i if both the prior and likelihood depend smoothly on θ_i as is often the case for a continuous parameter. On the other hand, $U(\boldsymbol{\theta})$ will be a discontinuous function of θ_i if θ_i is a continuous embedding of a discrete parameter. The coordinate-wise update of Algorithm 1 leads to a valid proposal mechanism whether or not $U(\boldsymbol{\theta})$ has discontinuities along the coordinate θ_i. If $U(\boldsymbol{\theta})$ varies smoothly along some coordinates of $\boldsymbol{\theta}$, however, we can devise an integrator that takes advantage of such smooth dependence.

To describe the integrator, we write $\boldsymbol{\theta} = (\boldsymbol{\theta}_I, \boldsymbol{\theta}_J)$ where the collections of indices I and J are defined as

$$I = \{i \in \{1, \ldots, d\} : U(\boldsymbol{\theta}) \text{ is a smooth function of } \theta_i\}, \quad J = \{1, \ldots, d\} \setminus I \tag{16}$$

[1] The following feature of Algorithm 1 is worth mentioning, though it does not affect the validity of DHMC as established in Section 4. While the numerical solution converges to the exact solution of (12) as $\epsilon \to 0$, the behavior of the two can diverge substantially for a fixed $\epsilon > 0$. For instance, suppose $U(\boldsymbol{\theta}^*(0)) = U(\boldsymbol{\theta}^*(\epsilon))$ but $\sup_{0<t<\epsilon} U(\boldsymbol{\theta}^*(t)) = \infty$. In this case, the exact solution cannot travel through the infinite energy barrier between $\boldsymbol{\theta}^*(0)$ and $\boldsymbol{\theta}^*(\epsilon)$ while the numerical solution can tunnel through the energy barrier.

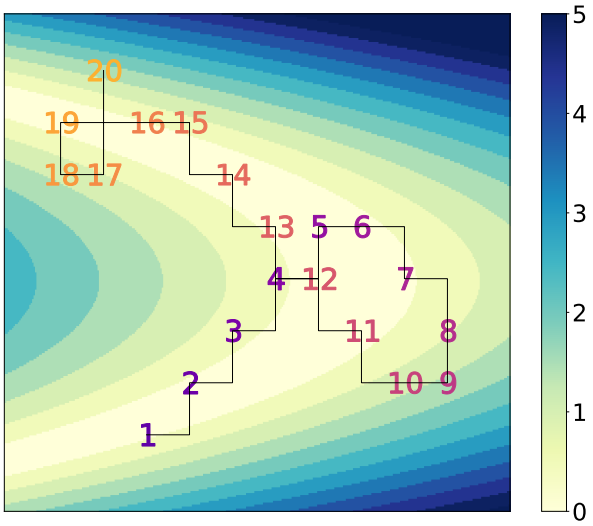

Figure 2: A trajectory of Laplace momentum based Hamiltonian dynamics $(\theta_1(t), \theta_2(t))$ approximated by the coordinate-wise integrator of Algorithm 1. The log density of the target distribution changes in the increment of 0.5 and has "banana-shaped" contours. Each numerical integration step consist of the coordinate-wise update along the horizontal axis followed by that along the vertical axis. (The order of the coordinate updates is randomized chosen at the beginning of numerical integration to ensure reversibility.) The trajectory initially travels in the direction of the initial momentum, a process marked by the numbers $1 - 4$. At the 5th numerical integration step, however, the trajectory does not have sufficient kinetic energy to take a step upward and hence flips the momentum downward. Such momentum flips also occur at the 8th and 9th numerical integration steps, again changing the direction of the trajectory. The rest of the trajectory follows the same pattern.

(More precisely, we assume that the parameter space has a partition that $\mathbb{R}^{|I|} \times \mathbb{R}^{|J|} = \cup_k \mathbb{R}^{|I|} \times \Omega_k$ such that $U(\boldsymbol{\theta})$ is smooth on $\mathbb{R}^{|I|} \times \Omega_k$ for each k.) We write $\boldsymbol{p} = (\boldsymbol{p}_I, \boldsymbol{p}_J)$ correspondingly and define the distribution of $\boldsymbol{p}$ as a product of Gaussian and independent Laplace so that the kinetic energy is given by

$$K(\boldsymbol{p}) = \frac{1}{2} \boldsymbol{p}_I^\intercal \boldsymbol{M}_I^{-1} \boldsymbol{p}_I + \sum_{j \in J} m_j^{-1} |p_j| \tag{17}$$

where $\boldsymbol{M}_I$ and $\boldsymbol{M}_J = \mathrm{diag}(\boldsymbol{m}_J)$ are *mass matrices* (Neal, 2010). With a slight abuse of terminology, we will refer to the parameters with discontinuous conditional densities $\boldsymbol{\theta}_J$ as discontinuous parameters.

The integrator is again based on operator splitting; we update the smooth parameter $(\boldsymbol{\theta}_I, \boldsymbol{p}_I)$ first, then the discontinuous parameter $(\boldsymbol{\theta}_J, \boldsymbol{p}_J)$, followed by another update of $(\boldsymbol{\theta}_I, \boldsymbol{p}_I)$. The discontinuous parameters are updated according to the coordinate-wise operators (12) as described in Section 3.3. The update of $(\boldsymbol{\theta}_I, \boldsymbol{p}_I)$ is based on a decomposition familiar from the leapfrog scheme:

$$\frac{\mathrm{d}\boldsymbol{p}_I}{\mathrm{d}t} = \nabla_{\boldsymbol{\theta}_I} \log \pi(\boldsymbol{\theta}), \quad \frac{\mathrm{d}\boldsymbol{\theta}_I}{\mathrm{d}t} = \mathbf{0}, \quad \frac{\mathrm{d}\boldsymbol{\theta}_J}{\mathrm{d}t} = \frac{\mathrm{d}\boldsymbol{p}_J}{\mathrm{d}t} = \mathbf{0} \tag{18}$$

$$\frac{\mathrm{d}\boldsymbol{\theta}_I}{\mathrm{d}t} = \boldsymbol{M}_I^{-1}\boldsymbol{p}_I, \qquad \frac{\mathrm{d}\boldsymbol{p}_I}{\mathrm{d}t} = \mathbf{0}, \quad \frac{\mathrm{d}\boldsymbol{\theta}_J}{\mathrm{d}t} = \frac{\mathrm{d}\boldsymbol{p}_J}{\mathrm{d}t} = \mathbf{0} \tag{19}$$

The pseudo code of Algorithm 2 describes the integrator with all the ingredients put together. The symbol φ denotes a bijective map (i.e. permutation) on J.

Compared to the coordinate-wise updates, the joint update of continuous parameters in this integrator has a couple of advantages. First, the joint update is much more efficient when there is little conditional independence structure that the coordinate-wise updates can take advantage of. Secondly, even with conditional independence structure, the joint update likely has a lower computational cost as an interpreter or compiler (of a programing language) can more easily optimize the required computation. On the other hand, the coordinate-wise updates have an advantage of being rejection-free by virtue of exact energy preservation and may be preferable for posteriors with substantial conditional independence structure such as in latent Markov random field models. Our numerical results in Section 5 use the integrator developed in this section.

Algorithm 2: Integrator for discontinuous HMC

Function `DiscIntegrator` $(\boldsymbol{\theta}, \boldsymbol{p}, \epsilon, \varphi = \texttt{Permute}(J))$**:**

- $\boldsymbol{p}_I \leftarrow \boldsymbol{p}_I + \frac{\epsilon}{2}\nabla_{\boldsymbol{\theta}_I} \log \pi(\boldsymbol{\theta})$
- $\boldsymbol{\theta}_I \leftarrow \boldsymbol{\theta}_I + \frac{\epsilon}{2}\boldsymbol{M}_I^{-1}\boldsymbol{p}_I$
- **for** j in J **do**
  - $\boldsymbol{\theta}, \boldsymbol{p} \leftarrow$ `CoordIntegrator`$(\boldsymbol{\theta}, \boldsymbol{p}, \varphi(j), \epsilon)$ `// Update discontinuous params`
- $\boldsymbol{\theta}_I \leftarrow \boldsymbol{\theta}_I + \frac{\epsilon}{2}\boldsymbol{M}_I^{-1}\boldsymbol{p}_I$
- $\boldsymbol{p}_I \leftarrow \boldsymbol{p}_I + \frac{\epsilon}{2}\nabla_{\boldsymbol{\theta}_I} \log \pi(\boldsymbol{\theta})$
- **return** $\boldsymbol{\theta}, \boldsymbol{p}$

4 Theoretical properties of discontinuous HMC

In this section, we study various theoretical properties of DHMC. Section 4.1 and 4.2 establish that the DHMC proposal kernel is reversible and irreducible, guaranteeing that the ergodic average of DHMC samples converges to the target distribution (see Geyer, 2011, and the references therein). The ergodicity and unbiasedness of DHMC is further verified in the supplement Section S2 through simulation. After discussing practical tuning issues in Section 4.3, we analyze the efficiency of DHMC through its interpretation as a generalization of Metropolis-within-Gibbs (Section 4.4) and through error analysis of the DHMC integrator (Section 4.5).

4.1 Reversibility

As in the existing HMC variants, the reversibility of DHMC is a direct consequence of the reversibility and volume-preserving property of our integrator in Algorithm 2. We will establish these properties of the integrator momentarily, but first we comment that our integrator has an unconventional feature of being reversible only "in distribution." To explain this, let $\boldsymbol{\Psi}$ denote a bijective map on the space $(\boldsymbol{\theta}, \boldsymbol{p})$ corresponding to the approximation of (discontinuous) Hamiltonian dynamics through multiple numerical integration steps. *Reversibility* of an integrator means that $\boldsymbol{\Psi}$ satisfies

$$(\boldsymbol{R} \circ \boldsymbol{\Psi})^{-1} = \boldsymbol{R} \circ \boldsymbol{\Psi} \quad \text{or equivalently} \quad \boldsymbol{\Psi}^{-1} = \boldsymbol{R} \circ \boldsymbol{\Psi} \circ \boldsymbol{R} \tag{20}$$

where $\boldsymbol{R} : (\boldsymbol{\theta}, \boldsymbol{p}) \to (\boldsymbol{\theta}, -\boldsymbol{p})$ is a momentum flip operator. Due to the updates of discrete parameters in a random order, the map $\boldsymbol{\Psi}$ induced by our integrator is non-deterministic and satisfies the reversibility condition (20) in distribution: $(\boldsymbol{R} \circ \boldsymbol{\Psi})^{-1} \overset{d}{=} \boldsymbol{R} \circ \boldsymbol{\Psi}$.

Once we establish the reversibility (in distribution) and volume-preserving property of the integrator, the reversibility of DHMC proposals follows from these properties of our integrator by the standard arguments with only a minor modification. For these arguments, we refer the readers to Neal (2010) for a heuristic presentation and to Fang et al. (2014) for a more general and mathematically precise treatment.

Analysis of our integrator's reversibility and volume-preserving property is similar to that of typical HMC integrators except for the coordinate-wise integration part of Algorithm 1. These properties for the coordinate-wise integrator are first established in Lemma 1, based on which the same properties for Algorithm 2 are established in Theorem 2.

Lemma 1. *For a piecewise smooth potential energy $U(\boldsymbol{\theta})$, the coordinate-wise integrator of Algorithm 1 is volume-preserving and reversible for any coordinate index i except on a set of measure zero. In other words, there exists a set of measure zero S such that, the bijective maps $\boldsymbol{\Psi}_i$ corresponding to the i-th coordinate updates are volume-preserving and reversible when restricted to $\mathbb{R} \setminus S$. Moreover, updating multiple coordinates with the integrator in a random index order $\varphi(1), \ldots, \varphi(d)$ is again reversible (in distribution) provided that the random permutation φ satisfies $(\varphi(1), \varphi(2), \ldots, \varphi(d)) \overset{d}{=} (\varphi(d), \varphi(d-1), \ldots, \varphi(1))$.*

Theorem 2. *For a piecewise smooth potential energy $U(\boldsymbol{\theta})$, the integrator of Algorithm 2 is volume-preserving and reversible except on a set of measure zero.*

The proofs are in Appendix A.

We also establish in Theorem 3 a more general result on the reversibility and volume-preserving property of discontinuous Hamiltonian dynamics. The result in particular justifies the use of the alternative event-driven integrator Algorithm 3 in the supplemental appendix and may be useful in constructing other event-driven integrators with various momentum distributions. A *solution operator* $\boldsymbol{\Psi}_t$ of a differential equation (or more generally of a differential inclusion) is a map such that $(\boldsymbol{\theta}(t), \boldsymbol{p}(t)) = \boldsymbol{\Psi}_t(\boldsymbol{\theta}_0, \boldsymbol{p}_0)$ is a solution of the equation with the initial condition $(\boldsymbol{\theta}(0), \boldsymbol{p}(0)) = (\boldsymbol{\theta}_0, \boldsymbol{p}_0)$. Also, *symplecticity* is a property of Hamiltonian dynamics which in particular implies volume-preservation; see Appendix B for the definition.

Theorem 3. *Let $U(\boldsymbol{\theta})$ be a piecewise constant potential energy function whose discontinuity set is piecewise linear. Suppose that a kinetic energy $K(\boldsymbol{p})$ is symmetric, convex, piecewise smooth, and satisfies the growth condition $K(\boldsymbol{p}) \to \infty$ as $\|\boldsymbol{p}\| \to \infty$. Then the solution operator $\boldsymbol{\Psi}_t$ of discontinuous Hamiltonian dynamics as defined in Section 2.3 is symplectic and reversible except on a set of measure zero.*

The proof is in Appendix B. Theorem 3 generalizes the result of Afshar and Domke (2015) in three ways: it holds for any kinetic energy satisfying the assumption, implies a stronger conclusion of symplecticity, and provides a rigorous quantification of non-differentiable sets. We believe that the conclusion of Theorem 3 holds under a much more general potential and kinetic energy since the reversibility and symplecticity are essential properties of smooth Hamiltonian dynamics. It is well beyond the scope of this paper to analyze the properties of more general non-smooth Hamiltonian dynamics, however, as the study of such dynamics requires more sophisticated mathematical tools; see Fetecau (2003) and Brogliato (2016) for more on this topic.

4.2 Ergodicity

Care needs to be taken when applying the coordinate-wise integrator as its use with a fixed stepsize ϵ results in a reducible Markov chain which is not ergodic; we discuss this issue here and present a simple remedy. Providing a general condition for the geometric convergence rate of DHMC, as has been done for the standard HMC by Livingstone et al. (2016), is left for a future work. However, in Section 4.4 we will discuss in ways through which DHMC can inherit the convergence rate of a related sampler.

Consider the transition probability of (multiple iterations of) DHMC based on the integrator of Algorithm 2. Given the initial state $\boldsymbol{\theta}_0$, the integrator of Algorithm 1 moves the i-th coordinate of $\boldsymbol{\theta}$ only by the distance $\pm\epsilon m_i^{-1}$ regardless of the values of the momentum variable. The transition probability in the $\boldsymbol{\theta}$-space with $\boldsymbol{p}$ marginalized out, therefore, is supported on a grid

$$\Omega = \{(\boldsymbol{\theta}_I, \boldsymbol{\theta}_J) : \boldsymbol{\theta}_J = \boldsymbol{\theta}_{0,J} + \epsilon \boldsymbol{m} \odot \boldsymbol{k} \text{ for a vector of integers } \boldsymbol{k}\} \tag{21}$$

where $\boldsymbol{\theta}_J$ as in (16) denotes the coordinates of $\boldsymbol{\theta}$ with discontinuous conditionals.

The pathological behavior above can be avoided simply by randomizing the stepsize at each iteration, say $\epsilon \sim \text{Uniform}(\epsilon_{\min}, \epsilon_{\max})$. In fact, randomizing the stepsize over a small interval is considered a good practice in any case to account for the possibility that some regions of the parameter space require smaller stepsizes for efficient exploration (Neal, 2010). While the coordinate-wise integrator does not suffer from the stability issue of the leapfrog scheme, the quantity ϵm_i^{-1} nonetheless needs to be in the same order of magnitude as the length scale of θ_i; see Section 4.3.

4.3 Role of mass matrix and stepsize

As in the case of standard HMC, using a non-identity mass matrix has the effect of pre-conditioning a target distribution through reparametrization (Neal, 2010). More precisely, for a matrix $\boldsymbol{A}_I$ and a diagonal matrix $\boldsymbol{A}_J$, the performance of DHMC in a sampling space

$(\boldsymbol{A}_I\boldsymbol{\theta}_I, \boldsymbol{A}_J\boldsymbol{\theta}_J, \boldsymbol{p}_I, \boldsymbol{p}_J)$ is identical to that in $(\boldsymbol{\theta}_I, \boldsymbol{\theta}_J, \boldsymbol{A}_I^\intercal\boldsymbol{p}_I, \boldsymbol{A}_J^\intercal\boldsymbol{p}_J)$. Since the choice of a mass matrix for a Gaussian momentum is a well-studied topic (Neal, 2010; Girolami and Calderhead, 2011), we focus on the choice of mass m_j for a Laplace momentum $p_j \sim \text{Laplace}(\text{scale} = m_j)$.

We generally expect that sampling is facilitated by a reparametrization $\boldsymbol{\theta}_J \to \boldsymbol{\Lambda}_J^{-1/2}\boldsymbol{\theta}_J$ for $\boldsymbol{\Lambda}_J = \text{diag}(\text{var}(\boldsymbol{\theta}_J))$. This suggests that, together with the property of DHMC under parameter transformation discussed above, the mass should be chosen as $m_j \approx \text{var}(\theta_j)^{-1/2}$. This choice is further motivated by the fact that a parameter θ_j is updated by an increment of $\pm\epsilon m_j^{-1}$ by the coordinate-wise integrator of Algorithm 1, and is related to the issue of choosing stepsize ϵ we discuss now.

The stepsize ϵ should be adjusted so that ϵm_j^{-1} has the same order of magnitude as a typical scale of the conditional distribution of θ_j. Unlike a leapfrog integrator that becomes unstable as ϵ increases, the coordinate-wise integrator remains exactly energy-preserving but at some point a large stepsize will cause DHMC to "get stuck" at the current state. The numerical integration scheme of DHMC will keep flipping the momentum $p_j \leftarrow -p_j$ (Line 9 of Algorithm 1) without updating θ_j until the following condition is met:

$$U\big(\boldsymbol{\theta} + \epsilon m_j^{-1}\text{sign}(p_j)\boldsymbol{e}_j\big) - U(\boldsymbol{\theta}) < m_j^{-1}|p_j| \overset{d}{=} \text{Exp}(1) \tag{22}$$

where $\boldsymbol{e}_j$ denotes the j-th standard basis vector. When ϵm_j^{-1} becomes larger than a typical scale of θ_j, it becomes unlikely for the condition (22) to be satisfied and leads to infrequent updates of θ_j.

Based on these observations, useful statistics for tuning the stepsize would be

$$\begin{aligned}&\mathbb{P}_{\pi_\Theta \times \pi_P}\left\{U\big(\boldsymbol{\theta} + \epsilon m_j^{-1}\text{sign}(p_j)\boldsymbol{e}_j\big) - U(\boldsymbol{\theta}) > m_j^{-1}|p_j|\right\} \\ &\qquad = \mathbb{E}_{\pi_\Theta \times \pi_P}\left[\min\left\{1, \exp\Big(U(\boldsymbol{\theta}) - U\big(\boldsymbol{\theta} + \epsilon m_j^{-1}\text{sign}(p_j)\boldsymbol{e}_j\big)\Big)\right\}\right]\end{aligned} \tag{23}$$

which plays a role analogous to the acceptance rate of Metropolis proposals. The statistics (23) can be estimated, for example, by counting the frequency of momentum flips during each DHMC iteration, and can then be used to tune the stepsize through stochastic optimization (Andrieu and Thoms, 2008; Hoffman and Gelman, 2014). One would want the statistics to be well above zero but not too close to 1, balancing the mixing rate and computational cost of each DHMC iteration. Theoretical analysis of the optimal statistics value is beyond the scope of this paper, but the value $0.7 \sim 0.9$ is perhaps reasonable in analogy with the optimal acceptance rate of HMC (Betancourt et al., 2014).

4.4 Metropolis-within-Gibbs with momentum

Consider a version of DHMC in which all the parameters are updated with the coordinate-wise integrator of Algorithm 1; in other words, the integrator of Algorithm 2 is applied with $J = \{1, \ldots, d\}$ and an empty indexing set I. This version of DHMC turns out to be a generalization of random-scan Metropolis-within-Gibbs (also known as one-variable-at-a-time Metropolis) algorithm. We therefore refer to this version of DHMC alternatively as *Metropolis-within-Gibbs with momentum*.

To make the connection to Metropolis-within-Gibbs clear, we first write out this version of DHMC explicitly. We use $\pi_\mathcal{E}(\cdot)$ and $\pi_\Phi(\cdot)$ to denote the distribution of a stepsize ϵ and a permutation φ of $\{1, \ldots, d\}$. As before, we require that $\varphi \sim \pi_\Phi(\cdot)$ satisfies

$(\varphi(1), \ldots, \varphi(d)) \stackrel{d}{=} (\varphi(d), \ldots, \varphi(1))$. With these notations, one iteration of Metropolis-within-Gibbs with momentum can be expressed as follows:

1. Draw $\epsilon \sim \pi_{\mathcal{E}}(\cdot)$, $\varphi \sim \pi_{\Phi}(\cdot)$, and $p_j \sim \text{Laplace}(\text{scale} = m_j)$ for $j = 1, \ldots, d$.
2. Repeat for L times a sequential update of the coordinate (θ_j, p_j) for $j = \varphi(1), \ldots, \varphi(d)$ via Algorithm 1 with stepsize ϵ.

In this version of DHMC, the integrator exactly preserves the Hamiltonian and the acceptance-rejection step can be omitted.

When $L = 1$, the above algorithm recovers Metropolis-within-Gibbs with a random scan order $\varphi \sim \pi_{\Phi}(\cdot)$. This can be seen by realizing that Lines 5 – 9 of Algorithm 1 coincide with the standard Metropolis acceptance-rejection procedure for θ_j. More precisely, the coordinate-wise integrator updates θ_j to $\theta_j + \epsilon m_j^{-1}\text{sign}(p_j)$ only if

$$\exp\left(-U(\boldsymbol{\theta}^*) + U(\boldsymbol{\theta})\right) > \exp\left(-m_j^{-1}|p_j|\right) \stackrel{d}{=} \text{Uniform}(0, 1) \tag{24}$$

where the last distributional equality follows from the fact $m_j^{-1}|p_j| \stackrel{d}{=} \text{Exp}(1)$. Theorem 4 below summarizes the above discussion.

Theorem 4. *Consider a version of DHMC updating all the parameters with the coordinate-wise integrator of Algorithm 1. When taking only one numerical integration step, this version of DHMC coincides with Metropolis-within-Gibbs with a random scan order $\varphi \sim \pi_{\Phi}(\cdot)$ and a symmetric proposal $\theta_j \pm \epsilon m_j^{-1}$ for each parameter with $\epsilon \sim \pi_{\mathcal{E}}(\cdot)$.*

As formulated in Theorem 4, the version of DHMC corresponds to a Metropolis-within-Gibbs with the univariate proposal distributions coupled via the shared parameter $\epsilon \sim \pi_{\mathcal{E}}(\cdot)$. We could also consider a version of DHMC with a fixed stepsize $\epsilon = 1$ but with a mass matrix randomized $(m_1^{-1}, \ldots, m_d^{-1}) \sim \pi_{M^{-1}}(\cdot)$ before each numerical integration step. This version would correspond to a more standard Metropolis-within-Gibbs with independent univariate proposals.

Being a generalization of Metropolis-within-Gibbs, DHMC is guaranteed a superior performance:

Corollary 5. *Under any efficiency metric (which may account for computational costs per iteration), an optimally tuned DHMC is guaranteed to outperform a class of random-scan Metropolis-within-Gibbs samplers as described in Theorem 4.*

In particular, an optimally tuned DHMC will inherit the geometric ergodicity of a corresponding Metropolis-within-Gibbs sampler, sufficient conditions for which are investigated in Johnson et al. (2013). In practice, the addition of momentum to Metropolis-within-Gibbs allows for a more efficient update of correlated parameters as empirically shown in the supplement Section S5.1.

4.5 Scalability in the number of parameters

Beskos et al. (2013) analyzes the scaling of the computational cost of HMC as the number of parameters d grows. For a target distribution of the form $\pi(\boldsymbol{\theta}) = \prod_{i=1}^{d} \pi_0(\theta_i)$, they show that the computational cost of HMC needs to scale as $O(d^{5/4})$ in order to maintain a $O(1)$ acceptance probability. This scaling property is superior to those of Metropolis-adjusted Langevin and random walk Metropolis algorithms and, in essence, a consequence of the fact that the global error in Hamiltonian incurred by the leapfrog algorithm is $O(\epsilon^2)$. As we show in the supplement Section S6, the global error in Hamiltonian by the DHMC integrator is also $O(\epsilon^2)$. It can therefore be expected that the scaling property of DHMC is comparable to that of HMC.

Since the coordinate-wise integrator of Algorithm 1 incurs no error in Hamiltonian, the version of DHMC as described in Section 4.4 potentially has a superior scaling property to HMC depending on the structure of a target distribution. For example, if the sequential evaluation of the individual conditional distributions can be carried out with the $O(d)$ computation, then this version of DHMC could have a $O(d)$ scaling of computational costs.

4.6 Relation to zig-zag sampler

Zig-zag sampler is a state-of-the-art rejection-free non-reversible Monte Carlo algorithm based on a piece-wise deterministic Markov process called a *zig-zag process* (Bierkens et al., 2016; Fearnhead et al., 2016; Bierkens et al., 2017). Both a zig-zag process and Laplace momentum based Hamiltonian dynamics (10) generate the same form of a piecewise linear trajectory $\boldsymbol{\theta}(t)$ segmented by event times. We delve deeper into the remarkable similarity between the two processes in the supplement Section S3.

5 Numerical results

We use two challenging posterior inference problems to demonstrate that DHMC is an efficient and general-purpose sampler that extends the scope of applicable models well beyond traditional HMC. Additional numerical results in the supplemental appendix (Section S5) further illustrate the breadth of DHMC's capability.

Currently, few general and efficient approaches exist for sampling from a discrete parameter or a discontinuous target density when the posterior is not conjugate. While adaptive Metropolis algorithms (Haario et al., 2001, 2006) based on Gaussian proposal distributions can be effective on a low-dimensional target distribution with limited non-linearity, such algorithms scale poorly in the number of parameters (Roberts et al., 1997). Metropolis-within-Gibbs with component-wise adaptation can scale better provided the conditional densities can be efficiently evaluated, but suffers from strong dependence among the parameters (Haario et al., 2005). We will use these algorithms as benchmarks when they are viable options. As another benchmark against discontinuous HMC, we use a best current practice implemented in a probabilistic programming language PyMC (Salvatier et al., 2016). Other HMC-based probabilistic programming languages do not support inference on discrete parameters to our knowledge.

PyMC samples continuous and discrete parameters alternately, sampling continuous ones with NUTS, a variant of HMC with automatic path length adjustments (Hoffman and Gelman, 2014), and discrete ones with univariate Metropolis updates. We will refer to this approach as a NUTS-Gibbs sampler. We tilt the comparison in favor of NUTS-Gibbs by updating each discrete parameter with a full conditional update through multinomial sampling from all possible values, truncated to a reasonable range if the sampling space is infinite. With our high-level language (Python) implementation, calculations required for such multinomial sampling require a small amount of computer time relative to continuous parameter updates, with these calculations more easily optimized through vectorization.

For each example, the stepsize and path length (i.e. the number of numerical integration steps) for DHMC were manually adjusted over short preliminary runs, visually examining the trace plots. Since this stepsize is well-calibrated with respect to the joint distribution of the continuous and discrete parameters, it is also well-calibrated as a stepsize for the conditional updates of the continuous parameters in the NUTS-Gibbs sampler.[2] Thus we used the stepsize from DHMC to sample the continuous parameters in the NUTS-Gibbs sampler.

In both the samplers, continuous parameters with range constraints are transformed into unconstrained ones to facilitate sampling (Stan Development Team, 2016; Salvatier et al., 2016). More precisely, the constraint $\theta > 0$ is handled by a log transform $\theta \to \log \theta$ and $\theta \in [0, 1]$ by a logit transform $\theta \to \log(\theta/(1-\theta))$ as done in Stan and PyMC. DHMC handles constraints on discrete parameters automatically through the coordinate-wise integrator of Algorithm 1.

Efficiencies of the algorithms are compared through effective sample sizes (ESS) (Geyer, 2011). As is commonly done in the MCMC literature, we compute the effective sample sizes of the first and second moment estimators for each parameter and report the minimum ESS across all the parameters. ESS's are estimated using the method of batch means with 25 batches (Geyer, 2011), averaged over the estimates from 8 independent chains.[3] We report a confidence interval for the ESS estimator of the form $\pm 1.96\,\hat{\sigma}$ with the empirical variance $\hat{\sigma}^2$.

[2] A stepsize for the NUTS-Gibbs sampler must be small enough to adapt to the geometries of all the possible distributions of the continuous parameters conditional on the discrete ones (PyMC Development Team, 2017). Being calibrated with respect to the joint distribution to be as large as possible while still generating stable trajectories, the stepsize used for DHMC should be close to an optimal choice for the NUTS-Gibbs sampler.

[3] While the consistency of a batch means estimator requires the number of batches to grow with the length of a chain, some of our benchmark samplers mix so slowly that it seemed practically impossible to run them long enough for such an estimator to be unbiased (Flegal and Jones, 2010). An estimator with a fixed number of batches is at least unbiased for a long enough chain, so we chose to use a fixed number of batches and assess its accuracy by running multiple independent chains. For reversible chains, our batch means estimates were similar to those of the monotone positive sequence estimator of Geyer (1992). The R package CODA (Plummer et al., 2006) provides an alternative estimator but it appeared to overestimate ESS's for very slowly mixing chains while the batch means and monotone sequence estimators agreed.

5.1 Jolly-Seber model: estimation of unknown open population size and survival rate from multiple capture-recapture data

The Jolly-Seber model and its numerous extensions are widely used in ecology to estimate unknown animal population sizes as well as related parameters of interest (Schwarz and Seber, 1999). The method is based on the following experimental design. Animals of a particular species having an unknown population size are captured, marked (for example by tagging), and released back to the environment. This procedure is repeated over multiple capture occasions. At each occasion, the number of marked and unmarked animals among the captured ones are recorded. Individuals survive from one capture occasion to another with an unknown survival rate. Also, the population is assumed to be "open" so that individuals may enter (either through birth or immigration) or leave the area under the study.

In order to be consistent with the literature on capture-recapture models, the notations within this section will deviate from the rest of the paper. Assuming that data are collected over $i = 1, \ldots, T$ capture occasions, the unknown parameters of the model are $\{U_i, p_i\}_{i=1}^{T}$ and $\{\phi_i\}_{i=1}^{T-1}$, each of which represents

- $U_i =$ number of unmarked animals right before the ith capture occasion.
- $p_i =$ capture probability of each animal at the ith capture occasion.
- $\phi_i =$ survival probability of each animal from the ith to $(i+1)$th capture occasion.

We assign standard objective priors $p_i, \phi_i \sim \text{Unif}(0, 1)$ and $\pi(U_1) \propto U_1^{-1}$. The prior conditional distributions $U_{i+1} \,|\, U_i, \phi_i$ are described in the supplement Section S4, along with the expression for the likelihood function and other details on the Jolly-Seber model.

We take the black-kneed capsid population data from Jolly (1965) as summarized in Seber (1982). The data records the capture-recapture information over $T = 13$ successive capture occasions. The posterior distribution of U_i may be heavy-tailed with conditional variance that depends highly on the capture probability p_i, so we alleviate these potential issues through log-transformed embedding of U_i's (Section 2.1). NUTS-Gibbs sampler updates U_i's through multinomial samplings from the integers between 0 and 5,000. Also, as this example happens to be low-dimensional enough, we try a random walk Metropolis with optimal Gaussian proposals by pre-computing the true posterior covariance with a long adaptive Metropolis chain (Roberts et al., 1997; Haario et al., 2001). DHMC can also take advantage of the posterior covariance information through the mass matrix, so we also try DHMC with a diagonal mass matrix whose entries are set according to the estimated posterior variance of each parameter. Starting from stationarity, we run 10^4 iterations of DHMC and NUTS-Gibbs and 5×10^5 iterations of Metropolis.

The performance of each algorithm is summarized in Table 1 where "DHMC (diagonal)" and "DHMC (identity)" indicate DHMC with a diagonal and identity mass matrix respectively. The table clearly indicates a superior performance of DHMC over NUTS-Gibbs and Metropolis with approximately 60 and 7-fold efficiency increase respectively when using a diagonal mass matrix. The posterior distribution exhibits high negative correlations between U_i and p_i, and all the algorithms recorded their worst ESS in the first capture probability p_1; see Figure 3. Some correlations are observed among the other pairs of parameters, but not to the same degree.

Table 1: Performance summary of each algorithm on the Jolly-Serber model example. The term ($\pm \ldots$) is the error estimate of our ESS estimators. Path length is averaged over each iteration. "Iter time" shows the computational time for one iteration of each algorithm relative to the fastest one.

	ESS per 100 samples	ESS per minute	Path length	Iter time
DHMC (diagonal)	45.5 ($\pm$ 5.2)	424	45	87.7
DHMC (identity)	24.1 ($\pm$ 2.6)	126	77.5	157
NUTS-Gibbs	1.04 ($\pm$ 0.087)	6.38	150	133
Metropolis	0.0714 ($\pm$ 0.016)	58.5	1	1

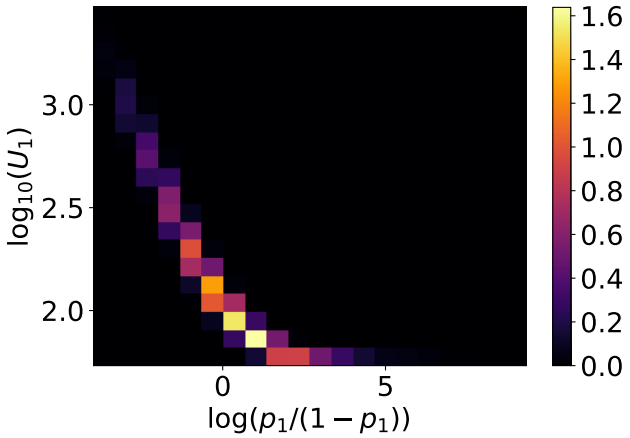

Figure 3: Two-dimensional empirical density function showing the posterior marginal of (p_1, U_1) with parameter transformations.

Our results demonstrate that, although the continuous and discontinuous parameters are not updated simultaneously in the DHMC integrator of Algorithm 2, the trajectory can nonetheless explore the joint distributions of highly dependent variables through the momentum information. It is also worth noting that the advantage of DHMC over Metropolis will likely increase as the parameter space dimension grows because of their scaling properties (Roberts et al., 1997; Beskos et al., 2013); see also Section 5.2 for the comparison of the two algorithms in a higher dimensional problem.

5.2 Generalized Bayesian belief update based on loss functions

Motivated by model misspecification and difficulty in modeling all aspects of a data generating process, Bissiri et al. (2016) propose a generalized Bayesian framework, which replaces the log-likelihood with a surrogate based on a utility function. Given an additive loss $\ell(y, \boldsymbol{\theta})$ for the data y and parameter of interest $\boldsymbol{\theta}$, the prior $\pi(\boldsymbol{\theta})$ is updated to obtain the generalized posterior:

$$\pi_{\text{post}}(\boldsymbol{\theta}) \propto \exp\left(-\ell(\boldsymbol{y}, \boldsymbol{\theta})\right) \pi(\boldsymbol{\theta}) \tag{25}$$

While (25) coincides with a pseudo-likelihood type approach, Bissiri et al. (2016) derives the formula as a coherent and optimal update from a decision theoretic perspective.

Here we consider a binary classification problem with an error-rate loss:

$$\ell(\boldsymbol{y}, \boldsymbol{\beta}) = \sum_{i=1} \mathbb{1}\left\{y_i \boldsymbol{x}_i^\intercal \boldsymbol{\beta} < 0\right\} \tag{26}$$

where $y_i \in \{-1, 1\}$, $\boldsymbol{x}_i$ is a vector of predictors, and $\boldsymbol{\beta}$ is a regression coefficient. The target distribution of the form (25) based on the loss function (26) is suggested as a challenging test case for an MCMC algorithm by Chopin and Ridgway (2017). We use the SECOM data from UCI machine learning repository, which records various sensor data that can be used to predict the production quality (pass or fail) of a semi-conductor. We first removed the predictors with more than 20 missing cases and then removed the observations that still had missing predictors, leaving us 1,477 cases with 376 predictors. All the predictors were then normalized and the regression coefficients β_i's were given $\mathcal{N}(0, 1)$ priors.

Chopin and Ridgway (2017) presents a wide range of algorithms to approximate or sample from a posterior distribution of binary classification problems, but none of them except random-walk Metropolis seems to apply to the target distribution of interest here — Figure 4 shows the conditional density of the intercept parameter as an illustration. Their numerical results also indicated that, after accounting for computational cost, none of the more complex algorithms consistently outperform Metropolis even for a parameter space of dimension as large as 180. We therefore compare DHMC to Metropolis with a proposal covariance matrix proportional to the empirical posterior covariance estimated by 10^5 iterations of DHMC. The scaling of the proposal covariance as suggested by Haario et al. (2001) resulted in an acceptance probability of less than 4%, so we scaled the proposal covariance to achieve the acceptance probability of 0.234 with stochastic optimization (Andrieu and Thoms, 2008). We also run Metropolis-within-Gibbs that updates one parameter at a time with the acceptance probability calibrated to be around 0.44 as recommended in Gelman et al. (1996). We run DHMC for 10^4 iterations, Metropolis for 10^7 iterations, and Metropolis-within-Gibbs for 5×10^4 iterations from stationarity. The Metropolis samples — the discarded samples contribute little to the ESS's due to extremely high auto-correlations.

Table 2 summarizes the performance of each algorithm. DHMC outperforms Metropolis and Metropolis-within-Gibbs approximately by a factor of 330 and 2 respectively. The mixing of Metropolis suffers substantially from the dimensionality of the target (377 parameters). We actually could not store all the 10^7 Metropolis samples in memory, so we kept only every 100 iterations for computing the ESS's; the discarded samples contribute little to the ESS's due to extremely high auto-correlations. Conditional updates of Metropolis-within-Gibbs mix relatively well as the posterior correlation happens to be quite modest — the ratio of the largest and smallest eigenvalues of the estimated posterior covariance matrix was approximately $46 \approx 6.8^2$.

6 Discussion

We have presented discontinuous HMC, an extension of HMC that can sample from discontinuous target densities while inheriting generality and efficiency of HMC. DHMC easily

Table 2: Performance summary of each algorithm on the generalized Bayesian posterior example. The term ($\pm \ldots$) is the error estimate of our ESS estimators. Path length is averaged over each iteration. "Iter time" shows the computational time for one iteration of each algorithm relative to the fastest one.

	ESS per 100 samples	ESS per minute	Path length	Iter time
DHMC	26.3 ($\pm$ 3.2)	76	25	972
Metropolis	0.00809 ($\pm$ 0.0018)	0.227	1	1
Metropolis-Gibbs	0.514 ($\pm$ 0.039)	39.8	1	36.2

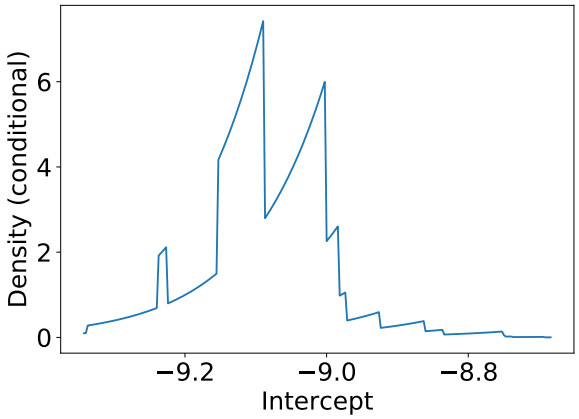

Figure 4: The posterior conditional density of the intercept parameter in the generalized Bayesian posterior example. The other parameters are fixed at the posterior draw with the highest posterior density among the DHMC samples. The density is not continuous since the loss function is not.

accommodates discrete parameters through the embedding strategy presented in Section 2.1. When using a Laplace momentum for each parameter, DHMC turns out to be a generalization of Metropolis-within-Gibbs that takes advantage of momentum information (Section 4.4). A preliminary result in the supplement Section S5.1 indicates that this version of DHMC may be useful on its own — regardless of continuity in a target density.

We are currently developing methods for automatic tuning of DHMC based on its properties described in Section 4 as well as various techniques that have been developed for tuning HMC (Hoffman and Gelman, 2014; Wang et al., 2013; Stan Development Team, 2016). We hope to integrate DHMC into an existing probabilistic programming language so that it can be deployed more widely and have its capability and limitations fully tested. Since DHMC requires calculations of log-differences in full conditional distributions, it is critical for its optimized implementation that conditional independence structure is exploited and redundant calculations avoided. Such operations are already automated, for example, by PyMC using the Theano library as a back-end (Theano Development Team, 2016).

In addition to being a highly practical algorithm, DHMC also motivates new directions in methodological and theoretical research on Monte Carlo methods. In the HMC literature to this date, there have been limited interests in non-Gaussian momenta as well as limited

research effort to develop a new class of integrators aside from incremental improvements in numerical accuracy (Livingstone et al., 2017; Blanes et al., 2014). Our work demonstrates a successful use of a non-Gaussian momentum along with a custom-made integrator in practical applications. The ideas introduced here could conceivably be extended to devise HMC-like algorithms in more complex parameter spaces as considered in Dinh et al. (2017). Also, the remarkable similarity between a zig-zag process and Hamiltonian dynamics underlying DHMC (Section 4.6) may indicate a deeper connection between piecewise deterministic Markov processes and Hamiltonian dynamics based samplers. A unifying framework for the two approaches, if it exists, could provide recipes for further innovations in Monte Carlo algorithms.

Acknowledgements

David Dunson is supported by National Science Foundation grant 1546130 and Office of Naval Research grant N00014-14-1-0245. Jianfeng Lu is supported by National Science Foundation grant DMS-1454939.

Appendix A: Proof of Lemma 1 and Theorem 2

Lemma 1. One step of Algorithm 1 corresponds to a map $\boldsymbol{\Psi}_{i,\epsilon} : (\boldsymbol{\theta}, \boldsymbol{p}) \to (\boldsymbol{\theta}^*, \boldsymbol{p}^*)$ as follows assuming $p_i \neq 0$. Let $\boldsymbol{e}_i$ denote the ith standard basis vector. We can express $(\boldsymbol{\theta}^*, \boldsymbol{p}^*)$ in terms of $(\boldsymbol{\theta}, \boldsymbol{p})$ as

$$\boldsymbol{\theta}^* = \boldsymbol{\theta} + \epsilon m_i^{-1} \text{sign}(p_i) \boldsymbol{e}_i, \quad \boldsymbol{p}^* = \boldsymbol{p} - m_i \left\{ U(\boldsymbol{\theta}^*) - U(\boldsymbol{\theta}) \right\} \boldsymbol{e}_i \tag{27}$$

if $U(\boldsymbol{\theta} + \epsilon m_i^{-1} \text{sign}(p_i) \boldsymbol{e}_i) - U(\boldsymbol{\theta}) > m_i^{-1} p_i$, and otherwise

$$\boldsymbol{\theta}^* = \boldsymbol{\theta}, \quad \boldsymbol{p}^* = -\boldsymbol{p} \tag{28}$$

The update equations (27) and (28) are well-defined and differentiable except on the measure-zero set S, which we define momentarily. Under both (27) and (28), we have $\partial \boldsymbol{\theta}^* / \partial \boldsymbol{p} = 0$ and it can be easily shown that

$$\det \left(\frac{\partial (\boldsymbol{\theta}^*, \boldsymbol{p}^*)}{\partial (\boldsymbol{\theta}, \boldsymbol{p})} \right) = \det \left(\frac{\partial \boldsymbol{\theta}^*}{\partial \boldsymbol{\theta}} \right) \det \left(\frac{\partial \boldsymbol{p}^*}{\partial \boldsymbol{p}} \right) = 1 \tag{29}$$

establishing the volume-preservation. The reversibility as defined in (20) can be directly verified by solving the update equations (27) and (28) for $(\boldsymbol{\theta}, -\boldsymbol{p})$ as a function of $(\boldsymbol{\theta}^*, -\boldsymbol{p}^*)$.

We now quantify the set S on which the above argument may break down and show that it has measure zero. Let $\mathcal{D}$ denote the discontinuity set of $U(\boldsymbol{\theta})$ and $\mathcal{D} + \boldsymbol{v}$ denote a set of points in $\mathcal{D}$ shifted by a vector $\boldsymbol{v}$. It is easy to see that the update equations (27) and (28) are well-defined and differentiable except when $(\boldsymbol{\theta}, \boldsymbol{p})$ belongs to one of the sets below:

$$\mathcal{D} \times \mathbb{R}^d, \ \left(\mathcal{D} \pm \epsilon m_i^{-1} \boldsymbol{e}_i \right) \times \mathbb{R}^d, \ \left\{ p_i = 0 \right\}, \ \left\{ U(\boldsymbol{\theta} + \epsilon m_i^{-1} \text{sign}(p_i) \boldsymbol{e}_i) - U(\boldsymbol{\theta}) = m_i^{-1} p_i \right\} \tag{30}$$

Each of the sets above consists of lower-dimensional manifolds of the parameter space and hence has measure zero. We now define the set S as the union of all the sets (30) over $i = 1, \ldots, d$. Being a finite union of measure-zero sets, the set S also has measure zero.

Lastly, we prove the reversibility of multiple coordinate updates corresponding to a map $\boldsymbol{\Psi}_{\varphi(d),\epsilon} \circ \ldots \circ \boldsymbol{\Psi}_{\varphi(1),\epsilon}$ with a random permutation φ. From the reversibility of each $\boldsymbol{\Psi}_{\epsilon,i}$, we deduce that

$$\boldsymbol{R} \circ \left(\boldsymbol{\Psi}_{\varphi(d),\epsilon} \circ \ldots \circ \boldsymbol{\Psi}_{\varphi(1),\epsilon}\right) \circ \boldsymbol{R} = \boldsymbol{\Psi}^{-1}_{\varphi(d),\epsilon} \circ \ldots \circ \boldsymbol{\Psi}^{-1}_{\varphi(1),\epsilon} = \left(\boldsymbol{\Psi}_{\varphi(1),\epsilon} \circ \ldots \circ \boldsymbol{\Psi}_{\varphi(d),\epsilon}\right)^{-1} \tag{31}$$

By our assumption on the distribution of φ, we have

$$\left(\boldsymbol{\Psi}_{\varphi(1),\epsilon} \circ \ldots \circ \boldsymbol{\Psi}_{\varphi(d),\epsilon}\right)^{-1} \overset{d}{=} \left(\boldsymbol{\Psi}_{\varphi(d),\epsilon} \circ \ldots \circ \boldsymbol{\Psi}_{\varphi(1),\epsilon}\right)^{-1} \tag{32}$$

establishing the reversibility of $\boldsymbol{\Psi}_{\varphi(d),\epsilon} \circ \ldots \circ \boldsymbol{\Psi}_{\varphi(1),\epsilon}$ in distribution. □

Theorem 2. Let $\boldsymbol{\Psi}_{J,\varphi,\epsilon} = \boldsymbol{\Psi}_{\varphi(d'),\epsilon} \circ \ldots \circ \boldsymbol{\Psi}_{\varphi(1),\epsilon}$ where $\boldsymbol{\Psi}_{j,\epsilon} : (\boldsymbol{\theta}, \boldsymbol{p}) \to (\boldsymbol{\theta}^*, \boldsymbol{p}^*)$ is defined as in (27) and (28) and $\varphi(1), \ldots, \varphi(d')$ is a permutation of the indexing set J. Also define $\boldsymbol{\Psi}_{\Theta,I,\epsilon/2}$ and $\boldsymbol{\Psi}_{P,I,\epsilon/2}$ as a function of $(\boldsymbol{\theta}, \boldsymbol{p})$ such that

$$\boldsymbol{\Psi}_{\Theta,I,\epsilon/2} : \boldsymbol{\theta}_I \to \boldsymbol{\theta}_I + \frac{\epsilon}{2}\boldsymbol{M}_I^{-1}\boldsymbol{p}_I, \quad \boldsymbol{\Psi}_{P,I,\epsilon/2} : \boldsymbol{p}_I \to \boldsymbol{p}_I - \frac{\epsilon}{2}\nabla_{\boldsymbol{\theta}_I} U(\boldsymbol{\theta}) \tag{33}$$

while leaving all the other coordinate unchanged. The integrator of Algorithm 2 can then be formally expressed as a map

$$\boldsymbol{\Psi}_{\Theta,I,\epsilon/2} \circ \boldsymbol{\Psi}_{P,I,\epsilon/2} \circ \boldsymbol{\Psi}_{J,\varphi,\epsilon} \circ \boldsymbol{\Psi}_{P,I,\epsilon/2} \circ \boldsymbol{\Psi}_{\Theta,I,\epsilon/2} \tag{34}$$

Being a symmetric composition of reversible maps, the map (34) is again reversible. The maps $\boldsymbol{\Psi}_{\Theta,I,\epsilon/2} \circ \boldsymbol{\Psi}_{P,I,\epsilon/2}$ and $\boldsymbol{\Psi}_{P,I,\epsilon/2} \circ \boldsymbol{\Psi}_{\Theta,I,\epsilon/2}$ coincide with symplectic Euler schemes in the coordinate $(\boldsymbol{\theta}_I, \boldsymbol{p}_I)$ and hence are volume preserving (Hairer et al., 2006). Since $\boldsymbol{\Psi}_{J,\varphi,\epsilon}$ is also volume-preserving by the results of Lemma 1, the composition (34) is volume-preserving. □

Appendix B: Proof of Theorem 3

First we define a notion of *symplecticity*, a property of Hamiltonian dynamics that implies a volume preservation and further has important consequences in the stability of numerical approximation schemes (Hairer et al., 2006).

Definition 1. A differentiable map $(\boldsymbol{\theta}, \boldsymbol{p}) \to (\boldsymbol{\theta}^*, \boldsymbol{p}^*)$ is called *symplectic* if

$$\frac{\partial(\boldsymbol{\theta}^*, \boldsymbol{p}^*)}{\partial(\boldsymbol{\theta}, \boldsymbol{p})}^T \boldsymbol{J} \frac{\partial(\boldsymbol{\theta}^*, \boldsymbol{p}^*)}{\partial(\boldsymbol{\theta}, \boldsymbol{p})} = \boldsymbol{J} \quad \text{for } \boldsymbol{J} = \begin{bmatrix} 0 & \boldsymbol{I}_d \\ -\boldsymbol{I}_d & 0 \end{bmatrix} \tag{35}$$

where $\boldsymbol{I}_d$ denotes a d-dimensional identity matrix. A dynamics is called symplectic if its solution operator is.

Proof of Theorem 3. Reversibility is a standard property of smooth Hamiltonian dynamics with a symmetric kinetic energy (Hairer et al., 2006). Defined as a (point-wise) limit of smooth dynamics, discontinuous dynamics therefore is also reversible.

We turn to the proof of symplecticity. Under the assumption of Theorem 3, the evolution of discontinuous Hamiltonian dynamics from a state $(\boldsymbol{\theta}, \boldsymbol{p})$ at $t = 0$ to $(\boldsymbol{\theta}^*, \boldsymbol{p}^*)$ at $t = \tau$ is given as follows. Dividing up the time intervals into a smaller pieces if necessary, we can without loss of generality assume that a trajectory $(\boldsymbol{\theta}(t), \boldsymbol{p}(t))$ encounters only one discontinuity at $\boldsymbol{\theta}(t_e)$ during the interval $[0, \tau]$. Since $U(\boldsymbol{\theta})$ is piecewise constant, the momentum remains constant and $\boldsymbol{\theta}(t)$ travels in a straight line except when hitting the discontinuity. The relationship between $(\boldsymbol{\theta}, \boldsymbol{p})$ and $(\boldsymbol{\theta}^*, \boldsymbol{p}^*)$ is therefore given by

$$\begin{aligned} \boldsymbol{\theta}^* &= \boldsymbol{\theta} + t_e \nabla_{\boldsymbol{p}} K(\boldsymbol{p}) + (\tau - t_e) \nabla_{\boldsymbol{p}} K(\boldsymbol{p}^*) \\ \boldsymbol{p}^* &= \boldsymbol{p} + \gamma(\boldsymbol{p}) \boldsymbol{\nu}_e \end{aligned} \tag{36}$$

where $\gamma(\boldsymbol{p})$ is defined implicitly as a solution of the following relations. If $\Delta U_e < K(\boldsymbol{p}) - \min_c K(\boldsymbol{p} - c\boldsymbol{\nu}_e)$ for ΔU_e defined as in (8), we define $\gamma(\boldsymbol{p})$ as a value that satisfies:

$$K(\boldsymbol{p} - \gamma \boldsymbol{\nu}_e) = K(\boldsymbol{p}) + \Delta U_e \quad \text{with } \gamma > 0 \tag{37}$$

Otherwise, $\gamma(\boldsymbol{p})$ is defined through the relation:

$$K(\boldsymbol{p} - \gamma \boldsymbol{\nu}_e) = K(\boldsymbol{p}) \quad \text{with } \gamma > 0 \tag{38}$$

The uniqueness of solutions to the above relations is guaranteed by the convexity and growth condition on $K(\boldsymbol{p})$, and hence $\gamma(\boldsymbol{p})$ is well-defined. The event time t_e is also a function of $(\boldsymbol{\theta}, \boldsymbol{p})$ and can easily shown to be

$$t_e(\boldsymbol{\theta}, \boldsymbol{p}) = \frac{\alpha - \langle \boldsymbol{\theta}, \boldsymbol{\nu}_e \rangle}{\langle \nabla_{\boldsymbol{p}} K(\boldsymbol{p}), \boldsymbol{\nu}_e \rangle} \tag{39}$$

where α is the distance from the origin of the discontinuity plane of U at $\boldsymbol{\theta}(t_e)$. Assuming that $\boldsymbol{\theta}(t_e)$ is not at the intersection of the linear discontinuity planes and that $\Delta U_e \neq K(\boldsymbol{p}) - \min_c K(\boldsymbol{p} - c\boldsymbol{\nu}_e)$, the relation (36) correctly describes the evolution of the dynamics on a small enough neighborhood of $(\boldsymbol{\theta}, \boldsymbol{p})$ with $\gamma(\boldsymbol{p})$ defined either through (37) or (38). The map $(\boldsymbol{\theta}, \boldsymbol{p}) \to (\boldsymbol{\theta}^*, \boldsymbol{p}^*)$ therefore is differentiable and Lemma 6 establishes the symplecticity through direct computation.

Lastly, we turn to the almost everywhere differentiability of discontinuous Hamiltonian dynamics. To characterize a state at which the solution operator fails to be differentiable, we first define the following sets:

- $\mathcal{D} = \{\, \boldsymbol{\theta} : \text{ multiple discontinuity boundaries of } U \text{ intersects at } \boldsymbol{\theta} \}$
- $\mathcal{U} = \{\, \Delta > 0 : \Delta = U(\boldsymbol{\theta}) - U(\boldsymbol{\theta}') \text{ for some } \boldsymbol{\theta}, \boldsymbol{\theta}' \}$
- $\mathcal{V} = \{\, \boldsymbol{\nu} : \boldsymbol{\nu} \text{ is orthonormal to a discontinuity boundary of } U \}$

The above sets are all countable by our assumption on $U(\boldsymbol{\theta})$. Based on the behavior of a trajectory as described in the previous paragraph, a trajectory from the initial state $(\boldsymbol{\theta}_0, \boldsymbol{p}_0)$

potentially experiences a non-differentiable behavior at time t only if the initial state belongs to one of the sets below:

$$\bigcup_{\boldsymbol{\theta}\in\mathcal{D}} \{(\boldsymbol{\theta} + s\nabla_{\boldsymbol{p}} K(\boldsymbol{p}), \boldsymbol{p}) : s \in \mathbb{R}\}, \quad \bigcup_{\Delta\in\mathcal{U},\boldsymbol{\nu}\in\mathcal{V}} \left\{(\boldsymbol{\theta},\boldsymbol{p}) : K(\boldsymbol{p}) - \min_c K(\boldsymbol{p} - c\boldsymbol{\nu}) = \Delta\right\}$$
$$\left\{(\boldsymbol{\theta},\boldsymbol{p}) : t = \frac{\alpha - \langle\boldsymbol{\theta},\boldsymbol{\nu}_e\rangle}{\langle\nabla_{\boldsymbol{p}} K(\boldsymbol{p}),\boldsymbol{\nu}_e\rangle}\right\} \tag{40}$$

Being a countable union of lower dimensional manifolds, all of the sets above have measure zero. □

Lemma 6. *The map* (36) *is symplectic for* $\gamma(\boldsymbol{p})$ *and* $t_e(\boldsymbol{\theta},\boldsymbol{p})$ *as defined through* (37), (38), *and* (39).

Proof. To simplify expressions, we denote $\boldsymbol{w} = \nabla_{\boldsymbol{p}} K(\boldsymbol{p})$, $\boldsymbol{w}^* = \nabla_{\boldsymbol{p}} K(\boldsymbol{p}^*)$, and let $\boldsymbol{\mathcal{H}}$ and $\boldsymbol{\mathcal{H}}^*$ denote the Hessians of K at $\boldsymbol{p}$ and $\boldsymbol{p}^*$. First, an implicit differentiation of either (37) or (38) with some algebra yields

$$\frac{\partial\gamma}{\partial\boldsymbol{p}} = \frac{\boldsymbol{w}^\intercal - \boldsymbol{w}^{*\intercal}}{\langle\boldsymbol{w}^*,\boldsymbol{\nu}\rangle} \tag{41}$$

Differentiating (36) with respect to $(\boldsymbol{\theta},\boldsymbol{p})$, we obtain

$$\begin{aligned} \frac{\partial\boldsymbol{\theta}^*}{\partial\boldsymbol{\theta}} &= \boldsymbol{I} - \frac{(\boldsymbol{w}-\boldsymbol{w}^*)\boldsymbol{\nu}_e^\intercal}{\langle\boldsymbol{w},\boldsymbol{\nu}_e\rangle}, & \frac{\partial\boldsymbol{\theta}^*}{\partial\boldsymbol{p}} &= t_e\boldsymbol{\mathcal{H}} - \frac{t_e}{\langle w,\boldsymbol{\nu}_e\rangle}(\boldsymbol{w}-\boldsymbol{w}^*)\boldsymbol{\nu}_e^\intercal\boldsymbol{\mathcal{H}} + (\tau - t_e)\boldsymbol{\mathcal{H}}^*\frac{\partial\boldsymbol{p}^*}{\partial\boldsymbol{p}} \\ \frac{\partial\boldsymbol{p}^*}{\partial\boldsymbol{\theta}} &= 0, & \frac{\partial\boldsymbol{p}^*}{\partial\boldsymbol{p}} &= \boldsymbol{I} + \frac{\boldsymbol{\nu}_e(\boldsymbol{w}-\boldsymbol{w}^*)^\intercal}{\langle\boldsymbol{w}^*,\boldsymbol{\nu}_e\rangle} \end{aligned} \tag{42}$$

When $\partial\boldsymbol{p}^*/\partial\boldsymbol{\theta} = \boldsymbol{0}$, the symplecticity condition (35) simplifies to:

$$\frac{\partial\boldsymbol{\theta}^{*\intercal}}{\partial\boldsymbol{\theta}}\frac{\partial\boldsymbol{p}^*}{\partial\boldsymbol{p}} = \boldsymbol{I}, \quad \frac{\partial\boldsymbol{p}^{*\intercal}}{\partial\boldsymbol{p}}\frac{\partial\boldsymbol{\theta}^*}{\partial\boldsymbol{p}} = \left(\frac{\partial\boldsymbol{p}^{*\intercal}}{\partial\boldsymbol{p}}\frac{\partial\boldsymbol{\theta}^*}{\partial\boldsymbol{p}}\right)^\intercal \tag{43}$$

The first equality in (43) is easily verified from (42). To establish the second equality of (43), we need to verify the symmetry of the matrix

$$\frac{\partial\boldsymbol{p}^{*\intercal}}{\partial\boldsymbol{p}}\frac{\partial\boldsymbol{\theta}^*}{\partial\boldsymbol{p}} = t_e\frac{\partial\boldsymbol{p}^{*\intercal}}{\partial\boldsymbol{p}}\left(\boldsymbol{I} - \frac{(\boldsymbol{w}-\boldsymbol{w}^*)\boldsymbol{\nu}_e^\intercal}{\langle\boldsymbol{w},\boldsymbol{\nu}_e\rangle}\right)\boldsymbol{\mathcal{H}} + (\tau - t_e)\frac{\partial\boldsymbol{p}^{*\intercal}}{\partial\boldsymbol{p}}\boldsymbol{\mathcal{H}}^*\frac{\partial\boldsymbol{p}^*}{\partial\boldsymbol{p}} \tag{44}$$

The first term of (44) simplifies to $t_e\boldsymbol{\mathcal{H}}$, which is symmetric, and the second term is obviously symmetric. □

References

Afshar, H. M. and Domke, J. (2015) Reflection, refraction, and Hamiltonian Monte Carlo. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, 3007–3015.

Alder, B. J. and Wainwright, T. E. (1959) Studies in molecular dynamics. I. general method. *The Journal of Chemical Physics*, **31**, 459–466.

Ambrosio, L. (2008) Transport equation and cauchy problem for non-smooth vector fields. In *Calculus of variations and nonlinear partial differential equations*, 1–41. Springer.

Andrieu, C. and Thoms, J. (2008) A tutorial on adaptive MCMC. *Statistics and Computing*, **18**, 343–373.

Basu, S. and Ebrahimi, N. (2001) Bayesian capture-recapture methods for error detection and estimation of population size: Heterogeneity and dependence. *Biometrika*, **88**, 269–279.

Berger, J. O., Bernardo, J. M. and Sun, D. (2012) Objective priors for discrete parameter spaces. *Journal of the American Statistical Association*, **107**, 636–648.

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

Betancourt, M., Byrne, S. and Girolami, M. (2014) Optimizing the integrator step size for Hamiltonian Monte Carlo. *arXiv:1411.6669*.

Bierkens, J., Bouchard-Côté, A., Doucet, A., Duncan, A. B., Fearnhead, P., Roberts, G. and Vollmer, S. J. (2017) Piecewise deterministic Markov processes for scalable Monte Carlo on restricted domains. *arXiv:1701.04244*.

Bierkens, J., Fearnhead, P. and Roberts, G. (2016) The zig-zag process and super-efficient sampling for Bayesian analysis of big data. *arXiv:1607.03188*.

Bissiri, P. G., Holmes, C. C. and Walker, S. G. (2016) A general framework for updating belief distributions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **78**, 1103–1130.

Blanes, S., Casas, F. and Sanz-Serna, J. M. (2014) Numerical integrators for the hybrid Monte Carlo method. *SIAM Journal on Scientific Computing*, **36**, A1556–A1580.

Brogliato, B. (2016) *Nonsmooth Mechanics. Models, Dynamics and Control*. Springer.

Carpenter, B., Hoffman, M. D., Brubaker, M., Lee, D., Li, P. and Betancourt, M. (2015) The Stan math library: Reverse-mode automatic differentiation in C++. *arXiv:1509.07164*.

Chib, S. (1998) Estimation and comparison of multiple change-point models. *Journal of Econometrics*, **86**, 221–241.

Chopin, N. and Ridgway, J. (2017) Leave Pima indians alone: binary regression as a benchmark for Bayesian computation. *Statistical Science*, **32**, 64–87.

Dinh, V., Bilge, A., Zhang, C. and Matsen, F. A. (2017) Probabilistic path Hamiltonian Monte Carlo. *arXiv:1702.07814*.

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

Fang, Y., Sanz-Serna, J. M. and Skeel, R. D. (2014) Compressible generalized hybrid Monte Carlo. *The Journal of Chemical Physics*, **140**, 174108.

Fearnhead, P., Bierkens, J., Pollock, M. and Roberts, G. O. (2016) Piecewise deterministic Markov processes for continuous-time Monte Carlo. *arXiv:1611.07873*.

Fetecau, R. C. (2003) *Variational methods for nonsmooth mechanics*. Ph.D. thesis, California Institute of Technology.

Flegal, J. M. and Jones, G. L. (2010) Batch means and spectral variance estimators in Markov chain Monte Carlo. *The Annals of Statistics*, **38**, 1034–1070.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A. and Rubin, D. B. (2013) *Bayesian Data Analysis*. CRC Press.

Gelman, A., Lee, D. and Guo, J. (2015) Stan: a probabilistic programming language for Bayesian inference and optimization. *Journal of Educational and Behavior Science*, **40**, 530–543.

Gelman, A., Roberts, G. O. and Gilks, W. R. (1996) Efficient Metropolis jumping rules. *Bayesian Statistics*, **5**, 599–607.

Geyer, C. (2011) Introduction to Markov chain Monte Carlo. *Handbook of Markov Chain Monte Carlo*, 3–48.

Geyer, C. J. (1992) Practical Markov chain Monte Carlo. *Statistical Science*, **7**, 473–483.

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

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

Haario, H., Laine, M., Mira, A. and Saksman, E. (2006) Dram: efficient adaptive MCMC. *Statistics and Computing*, **16**, 339–354.

Haario, H., Saksman, E. and Tamminen, J. (2001) An adaptive metropolis algorithm. *Bernoulli*, 223–242.

— (2005) Componentwise adaptation for high dimensional MCMC. *Computational Statistics*, **20**, 265–273.

Hairer, E., Lubich, C. and Wanner, G. (2006) *Geometric Numerical Integration. Structure-Preserving Algorithms for Ordinary Differential Equations*. Springer-Verlag.

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

Johnson, A. A., Jones, G. L. and Neath, R. C. (2013) Component-wise Markov chain Monte Carlo: Uniform and geometric ergodicity under mixing and composition. *Statistical Science*, 360–375.

Jolly, G. M. (1965) Explicit estimates from capture-recapture data with both death and immigration-stochastic model. *Biometrika*, **52**, 225–247.

Kruschke, J. (2014) *Doing Bayesian Data Analysis, Second Edition: A Tutorial with R, JAGS, and Stan*. Academic Press.

Livingstone, S., Betancourt, M., Byrne, S. and Girolami, M. (2016) On the geometric ergodicity of Hamiltonian Monte Carlo. *arXiv:1601.08057*.

Livingstone, S., Faulkner, M. F. and Roberts, G. O. (2017) Kinetic energy choice in Hamiltonian/hybrid Monte Carlo. *arXiv:1706.02649*.

Lu, X., Perrone, V., Hasenclever, L., Teh, Y. W. and Vollmer, S. J. (2016) Relativistic Monte Carlo. *arXiv:1609.04388*.

Lunn, D., Spiegelhalter, D., Thomas, A. and Best, N. (2009) The BUGS project: Evolution, critique and future directions. *Statistics in medicine*, **28**, 3049–3067.

McLachlan, R. I. and Quispel, G. R. W. (2002) Splitting methods. *Acta Numerica*, **11**, 341–434.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. and Teller, E. (1953) Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, **21**, 1087–1092.

Monnahan, C. C., Thorson, J. T. and Branch, T. A. (2016) Faster estimation of Bayesian models in ecology using Hamiltonian Monte Carlo. *Methods in Ecology and Evolution*, 339–348.

Nakajima, J. and West, M. (2013) Bayesian analysis of latent threshold dynamic models. *Journal of Business & Economic Statistics*, **31**, 151–164.

Neal, R. M. (1996) *Bayesian Learning for Neural Networks*. Springer-Verlag.

— (2010) MCMC using Hamiltonian Dynamics. In *Handbook of Markov chain Monte Carlo*. CRC Press.

Neelon, B. and Dunson, D. B. (2004) Bayesian isotonic regression and trend analysis. *Biometrics*, **60**, 398–406.

Pakman, A. and Paninski, L. (2013) Auxiliary-variable exact Hamiltonian Monte Carlo samplers for binary distributions. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, 2490–2498.

Parkin, D. M. and Bray, F. (2009) Evaluation of data quality in the cancer registry: Principles and methods part II. completeness. *European Journal of Cancer*, **45**, 756–764.

Peters, E. A. J. F. and de With, G. (2012) Rejection-free Monte Carlo sampling for general potentials. *Physical Review E*, **85**, 026703.

Plummer, M., Best, N., Cowles, K. and Vines, K. (2006) CODA: Convergence diagnosis and output analysis for MCMC. *R News*, **6**, 7–11.

PyMC Development Team (2017) "Compound Steps in Sampling". *PyMC3 online documentation*. URL: `https://docs.pymc.io/notebooks/sampling_compound_step.html`.

Roberts, G. O., Gelman, A. and Gilks, W. R. (1997) Weak convergence and optimal scaling of random walk Metropolis algorithms. *The Annals of Applied Probability*, **7**, 110–120.

Salvatier, J., Wiecki, T. V. and Fonnesbeck, C. (2016) Probabilistic programming in Python using PyMC3. *PeerJ Computer Science*.

Schwarz, C. J. and Seber, G. A. F. (1999) Estimating animal abundance: Review III. *Statistical Science*, **14**, 427–456.

Seber, G. A. F. (1982) *The estimation of animal abundance*. Griffin London.

Stan Development Team (2016) *Stan Modeling Language Users Guide and Reference Manual, Version 2.14.0*. URL: `http://mc-stan.org/`.

Stewart, D. E. (2000) Rigid-body dynamics with friction and impact. *SIAM Review*, **42**, 3–39.

Theano Development Team (2016) Theano: A Python framework for fast computation of mathematical expressions. *arXiv:1605.02688*.

Wagner, A. K., Soumerai, Stephen B .and Zhang, F. and Ross-Degnan, D. (2002) Segmented regression analysis of interrupted time series studies in medication use research. *Journal of Clinical Pharmacy and Therapeutics*, **27**, 299–309.

Wang, Z., Mohamed, S. and de Freitas, N. (2013) Adaptive Hamiltonian and Riemann manifold Monte Carlo samplers. In *Proceedings of the 30th International Conference on Machine Learning*, vol. 28, 1462–1470.

Warren, R. and Warren, J. R. (2013) Unauthorized immigration to the United States: Annual estimates and components of change, by state, 1990 to 2010. *International Migration Review*, **47**, 296–329.

Zhang, Y., Sutton, C., Storkey, A. and Ghahramani, Z. (2012) Continuous relaxations for discrete Hamiltonian Monte Carlo. In *Proceedings of the 25th International Conference on Neural Information Processing Systems*, 3194–3202.

Zhang, Y., Wang, X., Chen, C., Henao, R., Fan, K. and Carin, L. (2016) Towards unifying Hamiltonian Monte Carlo and slice sampling. *arXiv:1602.07800*.

Supplement to "Discontinuous Hamiltonian Monte Carlo for discrete parameters and discontinuous likelihoods"

S1 Event-driven integrator for a Gaussian momentum

Here we describe an implementation of the event-driven integrator proposed by Pakman and Paninski (2013) and Afshar and Domke (2015). The integrator is designed to approximate a discontinuous Hamiltonian dynamics with a Gaussian momentum $K(\boldsymbol{p}) = \|\boldsymbol{p}\|^2/2$. For simplicity, we assume that a parameter space $\boldsymbol{\theta}$ consists only of the embedded discrete parameters as described in Section 2.1, so that the target $\pi_\Theta(\cdot)$ is piecewise constant with the discontinuity set consisting of the boundaries of hyper-cubes. The pseudo code of an event-driven integrator in this setting is given in Algorithm 3. The integrator is energy-preserving and hence yields a rejection-free proposal.

Algorithm 3: Event-driven integrator for $K(\boldsymbol{p}) = \|\boldsymbol{p}\|^2/2$

Input : initial state $(\boldsymbol{\theta}, \boldsymbol{p})$, stepsize ϵ

```
t ← 0
while t < ε do
    t_e ← the time until reaching the next discontinuity
    if t + t_e > ε then
        θ ← θ + (ε − t)p
        t ← ε
    else
        θ ← θ + t_e p
        i ← the index of the axis orthogonal to the discontinuity plane at θ
        ΔU_e ← the potential energy difference
        if p_i^2/2 > ΔU_e then
            p_i ← sqrt(p_i^2 − 2ΔU_e)
        else
            p_i ← −p_i
        t ← t + t_e
```

S2 Illustrating exactness of DHMC through simulation

To empirically back up the theoretical results of Section 4, here we use DHMC to sample from a simple posterior distribution with a closed-form marginal distribution and demonstrate that the empirical distribution of DHMC samples indeed converges to the target. It is worth

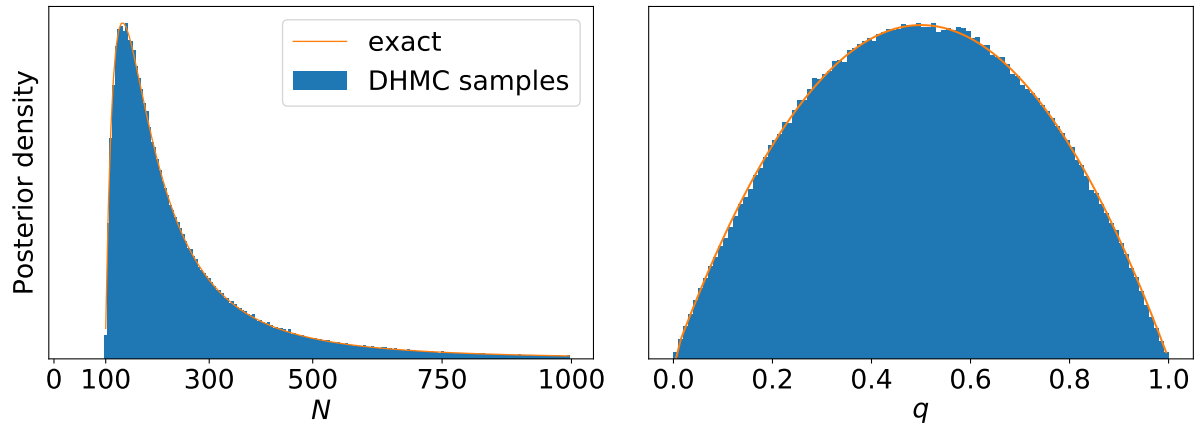

Figure S1: Empirical distributions (i.e. normalized histograms) of the DHMC samples generated for the target distribution as described in Section S2.1. The orange lines show the exact posterior mass and density functions computed from the closed-form expressions. The unknown sample size parameter N has no posterior probability below the observed number of successes $y = 100$.

mentioning that the correctness of DHMC has also been independently verified by Gram-Hansen et al. (2018), in which the DHMC samples across a range of models are compared to the outputs of existing probabilistic programming softwares.

We consider an observation model $y \mid q, N \sim \text{Binomial}(q, N)$ where both the success rate q and sample size N are unknown. We assign an objective prior $\pi(N) \propto N^{-1}$ (Berger et al., 2012) and a beta prior $q \sim \text{Beta}(\alpha, \beta)$. As a particular choice made is immaterial for the purpose of our simulation, we just pick $\alpha = \beta = 2$ and set $y = 100$. Closed-form expressions for the posterior marginals of N and q are given in Section S2.1 below.

To sample from the posterior, we use the log-transformed embedding of N (Section 2.1). The parameter q is mapped to a real line through a logit transform $q \to \log(q/(1-q))$. We use the integrator of Section 3.4 (Algorithm 2) with the Laplace momentum for N and Gaussian momentum for q. The stepsize ϵ is jittered in the range $[0.08, 0.1]$ and the number of numerical integration steps in the range $[15, 20]$.

Figure S1 shows the empirical distributions of $N \mid y$ and $q \mid y$ from 10^6 iterations of DHMC. The empirical distributions are indistinguishable from the exact distributions indicated by the orange lines. Additionally, the trace plot in Figure S2 shows that DHMC can induce a large transition in the parameter N with only a small number of numerical integration steps. This means that the DHMC integrator often jumps through a large number of discontinuities along the parameter N at each numerical integration step. This does not introduce any bias in the DHMC samples as the integrator remains reversible and volume-preserving regardless of its stepsize as discussed in Section 4.

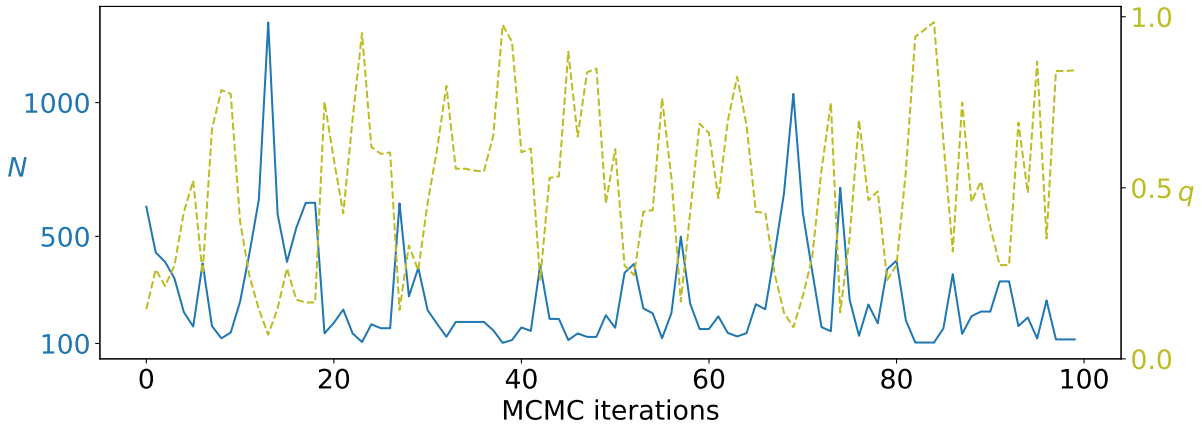

Figure S2: Trace plots for the first 100 DHMC samples generated for the target distribution as described in Section S2.1. The blue line and left y-axis indicates the parameter values of N, while the olive line and right y-axis indicates the parameter values of q.

S2.1 Derivation of the posterior marginals

For the model and priors described above, we have

$$\pi(N, q \,|\, y) \propto \frac{N!}{(N-y)!} q^y (1-q)^{N-y} \pi(q)\pi(N) \propto \frac{(N-1)!}{(N-y)!} q^{y+\alpha-1}(1-q)^{N-y+\beta-1} \tag{S1}$$

Integrating over q, we obtain

$$\pi(N \,|\, y) \propto \frac{(N-1)!}{(N-y)!} \frac{\Gamma(N-y+\beta)}{\Gamma(N+\alpha+\beta)} = \frac{(N-1)!}{(N-y)!} \frac{(N-y+\beta-1)!}{(N+\alpha+\beta-1)!} \tag{S2}$$

where the equality holds when α and β take positive integer values. As a particular choice made is immaterial for the purpose of our simulation, we just pick $\alpha = \beta = 2$ which yields

$$\pi(N \,|\, y) \propto \frac{N-y+1}{(N+3)(N+2)(N+1)N} \tag{S3}$$

We can compute the normalized mass function of $N \,|\, y$ to high accuracy by truncating it a suitably large number. Having computed $\pi(N \,|\, y)$, we can compute the posterior marginal of q via the law of total expectation $\pi(q \,|\, y) = \sum_N \pi(q \,|\, N, y)\pi(N \,|\, y)$ where $q \,|\, N, y \sim \text{Beta}(y+\alpha, N-y+\beta)$.

S3 Connections between zig-zag process and Laplace momentum Hamiltonian dynamics

Here we describe a remarkable similarity of the Laplace momentum based Hamiltonian dynamics with unit masses (i.e. $m_j = 1$ for all j) to a zig-zag process. As described in

Section 3.2 of the main manuscript, this Hamiltonian dynamics is governed by the following differential equation:

$$\frac{\mathrm{d}\boldsymbol{\theta}}{\mathrm{d}t} = \mathrm{sign}(\boldsymbol{p}), \quad \frac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}t} = -\nabla_{\boldsymbol{\theta}} U(\boldsymbol{\theta}) \tag{S4}$$

Consider a zig-zag process and Hamiltonian dynamics both starting from the state $\boldsymbol{\theta}_0$. Let $\boldsymbol{v}_0$ drawn uniformly drawn from $\{-1, +1\}^d$ be the initial velocity of the zig-zag process and $\boldsymbol{p}_0 = (p_{0,1}, \ldots, p_{0,d})$ drawn from the independent Laplace distribution be the initial momentum of the Hamiltonian dynamics. Under both the zig-zag process and Hamiltonian dynamics, the velocities remain constant while the parameter $\boldsymbol{\theta}$ moves along a straight line $\boldsymbol{\theta}^Z(t) = \boldsymbol{\theta}_0 + t\boldsymbol{v}_0$ and $\boldsymbol{\theta}^H(t) = \boldsymbol{\theta}_0 + t\,\mathrm{sign}(\boldsymbol{p}_0)$ for $t > 0$ until their respective first event times. The first event time for the zig-zag process is given as $t_e^Z = \min\{t_1^Z, \ldots, t_d^Z\}$ where

$$t_i^Z = \inf_{t'>0} \left\{ \tau_i = \int_0^{t'} \left[v_{0,i} \partial_{\theta_i} U(\boldsymbol{\theta}_0 + t\boldsymbol{v}_0) \right]^+ \mathrm{d}t' \right\} \tag{S5}$$

with $[x]^+ = \max\{0, x\}$ and τ_i's drawn from Exp(1). For the Hamiltonian dynamics, the first event time is given as $t_e^H = \min\{t_1^H, \ldots, t_d^H\}$ where

$$t_i^H = \inf_{t'>0} \left\{ |p_{0,i}| = \int_0^{t'} \mathrm{sign}(p_{0,i})\, \partial_{\theta_i} U\left(\boldsymbol{\theta}_0 + t\,\mathrm{sign}(\boldsymbol{p}_0)\right) \mathrm{d}t' \right\} \tag{S6}$$

For both processes, the events result in the velocity change $v_k \leftarrow -v_k$ and $\mathrm{sign}(p_\ell) \leftarrow -\mathrm{sign}(p_\ell)$ for $k = \mathrm{argmin}_i\{t_i^Z\}$ and $\ell = \mathrm{argmin}_i\{t_i^H\}$.

Given that $(\boldsymbol{v}_0, \boldsymbol{\tau}) \overset{d}{=} (\mathrm{sign}(\boldsymbol{p}_0), |\boldsymbol{p}_0|)$, the similarity between (S5) and (S6) is striking. In fact, if $U(\boldsymbol{\theta})$ were convex and $\boldsymbol{\theta}_0$ was the minimum of $U(\boldsymbol{\theta})$, then the two processes $\{\boldsymbol{\theta}^Z(t), 0 \le t \le t_e^Z\}$ and $\{\boldsymbol{\theta}^H(t), 0 \le t \le t_e^H\}$ coincide in distribution. After the first event time or in more general settings, however, the two processes diverge because a zig-zag process $(\boldsymbol{\theta}^Z, \mathrm{d}\boldsymbol{\theta}^Z/\mathrm{d}t) = (\boldsymbol{\theta}^Z, \boldsymbol{v})$ is Markovian while its Hamiltonian dynamics counter-part $(\boldsymbol{\theta}^H, \mathrm{d}\boldsymbol{\theta}^H/\mathrm{d}t) = (\boldsymbol{\theta}^H, \mathrm{sign}(\boldsymbol{p}))$ is not. More precisely, Hamiltonian dynamics after each event retains the magnitudes of its momentum $|p_i|$'s from the previous moment, so that the future evolution of $(\boldsymbol{\theta}^H, \mathrm{sign}(\boldsymbol{p}))$ cannot be determined only from its current value without the magnitude information. Also, Hamiltonian dynamics accumulates kinetic energy while going potential energy downhill such that $\mathrm{sign}\left(p_i(t)\right) \partial_{\theta_i} U(\boldsymbol{\theta}^H(t)) < 0$. This creates a tendency for each coordinate of a Hamiltonian dynamics trajectory $\boldsymbol{\theta}^H(t)$ to travel longer in the same direction before switching its direction compared to that of a zig-zag process.

Its close connection to a state-of-the-art sampler partially explains the empirical success of DHMC in Section S5.1, though the application of DHMC to smooth target distributions is outside the main focus of this paper. Some advantages of zig-zag sampler over others have been considered to be its non-reversibility and the fact that its entire trajectory can be used as valid samples from the target. In fact, DHMC can also be made non-reversible through partial momentum refreshments (Neal, 2010) and can utilize the entire trajectories as valid samples (Nishimura and Dunson, 2015). These strategies will likely further boost the performance of DHMC. We will leave further theoretical and empirical comparisons of DHMC and zig-zag sampler to a future work.

S4 Additional details on Jolly-Seber model

S4.1 Observed quantities / statistics

Under appropriate assumptions, details of which we refer the reader to Seber (1982), the likelihood of the Jolly-Seber model depends only on the following statistics from a capture-recapture experiment carried over $i = 1, \ldots, T$ capture occasions:

- $R_i =$ number of marked animals released after the ith capture occasion.
- $r_i =$ number of animals from the released R_i animals that are subsequently captured.
- $z_i =$ number of animals that are caught before ith capture occasion, not caught in the ith capture occasion, but caught subsequently.
- $m_i =$ number of marked animals caught at the ith capture occasion.
- $u_i =$ number of unmarked animals caught at the ith capture occasion.

S4.2 Likelihood function

The likelihood decomposes into two parts: one for the first captures of previously unmarked animals and another for their re-captures. More precisely,

$$\begin{aligned} L(\text{data} \,|\, \boldsymbol{U}, \boldsymbol{p}, \boldsymbol{\phi}) &= L(\text{first captures}) \times L(\text{re-captures}) \\ L(\text{first captures}) &\propto \prod_{i=1}^{T} \frac{U_i!}{U_i - u_i!} p_i^{u_i} (1 - p_i)^{U_i - u_i} \\ L(\text{re-captures}) &\propto \prod_{i=1}^{T-1} \chi_i^{R_i - r_i} \{\phi_i (1 - p_{i+1})\}^{z_{i+1}} (\phi_i p_{i+1})^{m_{i+1}} \end{aligned} \tag{S7}$$

where χ_i represents the conditional probability that a marked animal released after the ith capture occasion is not caught again. Mathematically, χ_i is defined recursively as

$$\chi_{T-1} = 1 - \phi_{T-1} p_T, \quad \chi_i = 1 - \phi_i \{p_{i+1} + (1 - p_{i+1})(1 - \chi_{i+1})\} \tag{S8}$$

S4.3 Prior distribution for $U_{i+1} \,|\, U_i, \phi_i$

Let B_i denote the number of "births," representing animals that are born, enters (immigration), or leaves (emigration) the population after the ith occasion and remains so until the $(i+1)$th occasion. Also let S_i denote the number of animals that are unmarked right after the ith capture occasion and survives until the next capture occasion. Then we have $U_{i+1} = B_i + S_i$ where $S_i \,|\, U_i, u_i, \phi_i \sim \text{Binom}(\phi_i, U_i - u_i)$.

The prior distribution of $\{U_i\}_{i=1}^T$ can thus be induced by assigning a prior on B_i's. In our example, we assign a convenient prior on U_i's based on the assumptions that 1) $\text{Binom}(\phi_i, U_i - u_i)$ can be approximated by $\mathcal{N}(U_i - u_i, \phi_i(1-\phi_i))$ and 2) B_i's are approximately i.i.d. $\mathcal{N}(0, \sigma_B^2)$. These assumptions motivates a prior

$$U_{i+1} \,|\, U_i, u_i, \phi_i, \sigma_B \sim \left\lfloor \mathcal{N}(U_i - u_i, \sigma_B^2 + \phi_i(1 - \phi_i)) \right\rfloor \tag{S9}$$

where $\lfloor \cdot \rfloor$ is a floor function. We used $\sigma_B = 500$ in our example of Section 5.1 in the main manuscript. An alternative prior on $\{U_i\}_{i=1}^T$ can be assigned to reflect different model and prior assumptions on the number of births. For instance, it is more natural to constrain $B_i \geq 0$ in some cases (Schwarz and Arnason, 1996) and a binomial distribution on B_i will for example induce a Poisson-binomial distribution on the conditional $U_{i+1} \,|\, U_i, u_i, \phi_i$ after marginalizing over B_i and S_i.

S4.4 Inference on unknown population sizes

In case the total population sizes $\{N_i\}_{i=1}^T$ at each capture occasion are of interest, we can generate their posterior samples using the relation $N_i = M_i + U_i$ where M_i denotes the number of marked animals right before the $(i+1)$th capture occasion. The distribution of $\{M_i\}_{i=1}^T$ follows $M_0 = 0$ and $M_{i+1} \,|\, M_i, \phi_i \sim \text{Binom}(M_i, \phi_i)$.

S5 Additional numerical results

S5.1 Comparison of DHMC and Gibbs in a synthetic example

We use a synthetic target distribution to demonstrate the difference between Metropolis-within-Gibbs with and without momentum as discussed in the main manuscript Section 4.4. While DHMC requires neither conjugacy or smoothness of the conditional densities, we choose a multivariate Gaussian target distribution so that we can compare DHMC to an optimal Metropolis-within-Gibbs implementation with the univariate proposal variances chosen according to the theory of Gelman et al. (1996). In particular, we assume that the target distribution of $\boldsymbol{\theta}$ follows that of a stationary unit variance auto-regressive process of the form

$$\theta_t = \alpha\theta_{t-1} + \sqrt{1-\alpha^2}\eta_t, \quad \theta_1, \eta_t \sim \mathcal{N}(0,1) \tag{S10}$$

for $t = 2, \ldots, 1000$ with $\alpha = 0.9$.

We compare the performances of four algorithms: DHMC (coordinate-wise), Gibbs (full conditional updates), Metropolis-within-Gibbs (univariate updates with optimal proposal variances), and NUTS (No-U-Turn-Sampler of Hoffman and Gelman (2014)). The performance of each algorithm is summarized in Table S1, which shows that DHMC outperforms not only Metropolis-within-Gibbs but also Gibbs (recall that DHMC requires no closed form conditionals at all). After accounting for the computational costs, DHMC improves Gibbs by over 50% and Metropolis-within-Gibbs by over 600%. In general, the advantage of DHMC over Gibbs is expected to increase as the correlations among the parameters increase because the use of momentum can suppress the "random walk behavior" (Neal, 2010). The covariance matrix of the target distribution here has a condition number $\approx 19^2$, which corresponds to a substantial but not particularly severe correlations.

In computing ESS per unit time, we estimated theoretical and platform-independent relative computational time of the algorithms as follows. In reasonable low-level language implementations, the computation of conditional densities should account for the majority of computational times for a typical target distribution. Therefore, computational efforts should be roughly equivalent between one numerical integration step of DHMC and one iteration

Table S1: Performance summary of each algorithm on the auto-regressive process example. The term $(\pm \ldots)$ is the error estimate of our ESS estimators. ESS per unit time normalizes the ESS's with computational efforts. Path length is averaged over each iteration. "Iter time" shows the computational time for one iteration of each algorithm relative to the fastest one.

	ESS per 100 samples	ESS per unit time	Path length	Iter time
DHMC	77.4 ($\pm$ 5.2)	1.56	49.5	49.5
NUTS	52.4 ($\pm$ 3.2)	N/A	142	N/A
Gibbs	0.949 ($\pm$ 0.076)	0.949	1	1
Metropolis-within	0.219 ($\pm$ 0.015)	0.219	1	1

of (Metropolis-within) Gibbs sampler. The computational cost of NUTS relative to these algorithms is more specific to individual target distributions, depending strongly on specific structures such as conditional independence among the parameters. For this reason, we do not attempt to compare NUTS to the other algorithms in terms of ESS per unit time.

S5.2 Multiple change-point detection for auto-regressive conditional heteroscedastic processes

Auto-regressive conditional heteroscedastic processes (ARCH) are a popular model for log-returns of speculative prices such as stock market indices. A non-stationary first-order ARCH process $\{y_t\}_{t=1}^T$ with parameters $\{a(t), b(t)\}_{t=1}^T$ assumes the distribution

$$y_t \,|\, y_{t-1}, a, b \sim \mathcal{N}(0, \sigma_t^2) \quad \text{where} \;\; \sigma_t^2 = a(t) + b(t)\, y_{t-1}^2 \tag{S11}$$

Motivated by its interpretability and advantage in forecasting, Fryzlewicz and Subba Rao (2014) propose a piecewise constant parametrization of $a(t)$ and $b(t)$ as follows:

$$\big(a(t), b(t)\big) = (a_k, b_k) \;\; \text{if} \;\; \tau_{k-1} < t \leq \tau_k \tag{S12}$$

for $k = 1, \ldots, K$ where the number of change points K and their locations $1 = \tau_0 < \tau_1 < \ldots < \tau_K$ are to be estimated along with (a_k, b_k)'s.

To fit the above model within a Bayesian paradigm, we infer the change points through a variable selection type approach as follows, using the horseshoe shrinkage priors of Carvalho et al. (2010). We first choose an upper bound $K_{\max}$ on the number of change points and assume a uniform prior on τ_k's on the constrained space $1 < \tau_1 < \ldots < \tau_{K_{\max}} < T$. We then model the changes in the values of $a(t)$ and $b(t)$ through a prior

$$\begin{array}{c} \log(a_k/a_{k-1}) \sim \mathcal{N}\left(0, \sigma_a \eta_{a,k}\right) \\ \log(b_k/b_{k-1}) \sim \mathcal{N}\left(0, \sigma_b \eta_{b,k}\right) \end{array} \quad \text{with} \;\; \eta_{a,k}, \eta_{b,k} \sim \text{Cauchy}^+(0,1) \tag{S13}$$

where $\text{Cauchy}^+(0,1)$ denotes the standard half-Cauchy prior and σ_a and σ_b are the global shrinkage parameters (Carvalho et al., 2010). The above approach can "select" a subset of $\tau_1, \ldots, \tau_{K_{\max}}$ as real change points by removing the others through shrinkage $a_k \approx a_{k-1}$ and $b_k \approx b_{k-1}$. We place a default prior $\sigma_a, \sigma_b \sim \text{Cauchy}^+(0,1)$ for the global shrinkage parameters (Gelman, 2006), and a weak prior $a_0, b_0 \sim \mathcal{N}(0,1)$ for the initial volatility parameters.

Table S2: Performance summary of each algorithm on the change points detection example. The term ($\pm \ldots$) is the error estimate of our ESS estimators. Path length is averaged over each iteration. "Iter time" shows the computational time for one iteration of each algorithm relative to the fastest one.

	ESS per 100 samples	ESS per minute	Path length	Iter time
DHMC	13.7 ($\pm$ 1.1)	38.7	87.3	1.03
NUTS-Gibbs	11.6 ($\pm$ 3.2)	33.5	218	1
NUTS-Metropolis	6.04 ($\pm$ 1.2)	17.5	217	1

Following Fryzlewicz and Subba Rao (2014), we fit our model to the log return values of a stock market index over a period that includes the subprime mortgage crisis. In particular, we use the daily closing values of S&P 500 on the market opening days during the period from Jan 1st, 2005 to Dec 31st, 2009.[4] The model parameters in this example are largely nonidentifiable even with the order constraint $\tau_1, \ldots, \tau_{K_{\max}}$. In such cases, it is not clear if the minimum ESS across the individual parameters is a good measure of efficiency. For this example, therefore, we calculate the minimum ESS over the first and second moments of the following quantities: the hyper-parameters σ_a and σ_b, log posterior density, and four summary statistics of the estimated functions $a(t)$ and $b(t)$ as defined in the footnote.[5] Both algorithms are run for 2.5×10^4 iterations from stationarity.

The simulation results are summarized in Table S2. While the performance of NUTS-Gibbs is comparable to DHMC, as discussed earlier DHMC has an advantage that all the necessary computations could be completely automated within the framework of existing probabilistic programming languages. For a more useful comparison, therefore, we also implemented the default sampling scheme by PyMC; each of the discrete parameters was updated a Metropolis step whose proposal distribution is a symmetric uniform integer-valued distribution with the variance calibrated to achieve the acceptance rate around 40%.

This example is challenging for DHMC as the posterior of τ_k's are in general multi-modal conditionally on the continuous parameters. The complex dependency between the local shrinkage and the other parameters creates potential paths among the modes, however. It seems that DHMC can exploit this complex posterior geometry efficiently and be competitive with NUTS-Gibbs. Figure S3 plots 100 DHMC posterior samples of the piecewise constant volatility functions $a(t)$ and $b(t)$ to illustrate the posterior structure of the model.

[4]The log return value cannot be computed when a daily closing value exactly coincides with the previous one. There were four such days during the period and these data points were removed.

[5] We define the four summary statistics $\log(\|a\|_2), \log(\|b\|_2), C_a$, and C_b as follows. The quantity $\|a\|_2$ summarizes the deviation of $a(t)$ from its posterior (pointwise empirical) mean $\hat{a}(t)$ and is defined as $\|a\|_2 = \sum_{t=1}^{T} |a(t) - \hat{a}(t)|^2$. The statistics C_a is a surrogate for the number of "change points" in the function $a(t)$:

$$C_a = \left| \{k \in \{1, \ldots, K_{\max}\} : |\log(a_k / a_{k-1})| > .1\} \right| \tag{S14}$$

The statistics $\|b\|_2$ and C_b are defined analogously.

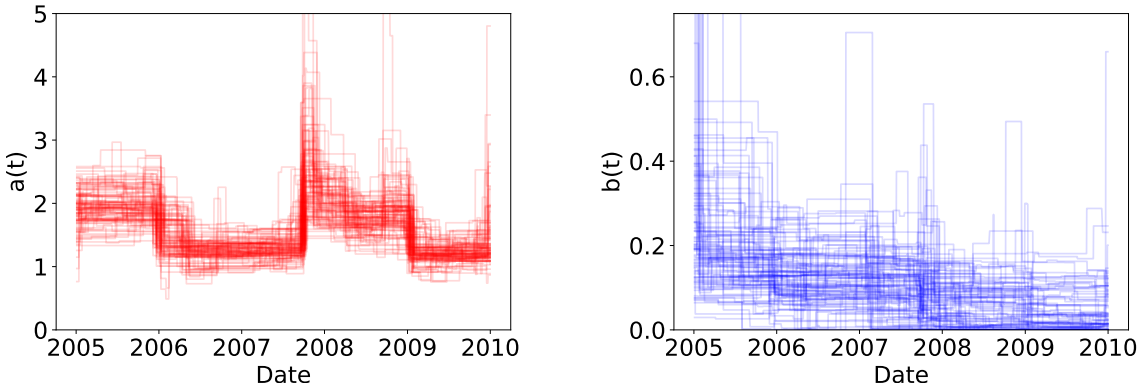

Figure S3: Posterior samples of the piecewise constant volatility functions $a(t)$ and $b(t)$ from 100 iterations of DHMC.

S6 Error analysis of DHMC integrator

Here we analyze the approximation error incurred by the integrator of Algorithm 2. We focus on the error in Hamiltonian — the amount by which the Hamiltonian fluctuates along a numerical solution — as it determines the acceptance probability of a proposal. An error incurred by one numerical integration step $(\boldsymbol{\theta}^0, \boldsymbol{p}^0) \to (\boldsymbol{\theta}^1, \boldsymbol{p}^1)$ of stepsize ϵ is known as a *local error*. Approximating the evolution $(\boldsymbol{\theta}(0), \boldsymbol{p}(0)) \to (\boldsymbol{\theta}(\tau), \boldsymbol{p}(\tau))$ requires $L(\epsilon) = \lfloor \tau/\epsilon \rfloor$ numerical integration steps and the error incurred by the map $(\boldsymbol{\theta}^0, \boldsymbol{p}^0) \to (\boldsymbol{\theta}^L, \boldsymbol{p}^L)$ is known as a *global error*. We quantify the local error of Algorithm 2 in Section S6.1 and relate it to the global error in Section S6.2.

S6.1 Local error in Hamiltonian

In analyzing Algorithm 2, it is useful to break up the algorithm into three steps; the first (partial) update of continuous parameters, the update of discrete / discontinuous parameters, and the second update of continuous parameters. The notation $(\boldsymbol{\theta}_I^{1/2}, \boldsymbol{p}_I^{1/2})$ will refer to the intermediate state after the first update of continuous parameters i.e. $\boldsymbol{p}_I^{1/2} = \boldsymbol{p}_I^0 - \frac{\epsilon}{2}\nabla_{\boldsymbol{\theta}_I} U(\boldsymbol{\theta}_I^0, \boldsymbol{\theta}_J^0)$ and $\boldsymbol{\theta}_I^{1/2} = \boldsymbol{\theta}_I^0 + \frac{\epsilon}{2}\nabla_{\boldsymbol{p}_I} K(\boldsymbol{p}_I^{1/2}, \boldsymbol{p}_J^0)$ where $K(\boldsymbol{p}) = \frac{1}{2}\boldsymbol{p}_I^\intercal \boldsymbol{M}_I^{-1}\boldsymbol{p}_I + \sum_{j\in J} m_j^{-1}|p_j|$ as before. The update $(\boldsymbol{\theta}_I^0, \boldsymbol{p}_I^0) \to (\boldsymbol{\theta}_I^{1/2}, \boldsymbol{p}_I^{1/2})$ is followed by the update $(\boldsymbol{\theta}_J^0, \boldsymbol{p}_J^0) \to (\boldsymbol{\theta}_J^1, \boldsymbol{p}_J^1)$ of discontinuous parameters, which then is followed by another continuous parameter update $(\boldsymbol{\theta}_I^{1/2}, \boldsymbol{p}_I^{1/2}) \to (\boldsymbol{\theta}_I^1, \boldsymbol{p}_I^1)$. The exact solution is denoted by $(\boldsymbol{\theta}(t), \boldsymbol{p}(t))$ with the initial condition $(\boldsymbol{\theta}(0), \boldsymbol{p}(0)) = (\boldsymbol{\theta}^0, \boldsymbol{p}^0)$.

The key result in this section is Corollary 8 below, which follows immediately from the following theorem:

Theorem 7. *The local error in Hamiltonian incurred by Algorithm 2 is given by*

$$H(\boldsymbol{\theta}^1, \boldsymbol{p}^1) - H(\boldsymbol{\theta}^0, \boldsymbol{p}^0) = \frac{\epsilon^2}{8}\left\{\xi\left(\boldsymbol{\theta}_I^{1/2}, \boldsymbol{\theta}_J^1, \boldsymbol{p}_I^{1/2}\right) - \xi\left(\boldsymbol{\theta}_I^{1/2}, \boldsymbol{\theta}_J^0, \boldsymbol{p}_I^{1/2}\right)\right\} + O(\epsilon^3) \tag{S15}$$

where ξ is defined in terms of the Hessians $\boldsymbol{\mathcal{I}}_U = \partial^2 U/\partial \boldsymbol{\theta}_I^2$ and $\boldsymbol{\mathcal{I}}_K = \partial^2 K/\partial \boldsymbol{p}_I^2$ (with respect to continuous parameters) as

$$\begin{aligned}\xi(\boldsymbol{\theta}_I, \boldsymbol{\theta}_J, \boldsymbol{p}_I) = \nabla_{\boldsymbol{\theta}_I}^\intercal U(\boldsymbol{\theta}_I, \boldsymbol{\theta}_J) \boldsymbol{\mathcal{I}}_K(\boldsymbol{p}_I) \nabla_{\boldsymbol{\theta}_I} U(\boldsymbol{\theta}_I, \boldsymbol{\theta}_J) \\ - \nabla_{\boldsymbol{p}_I}^\intercal K(\boldsymbol{p}_I) \boldsymbol{\mathcal{I}}_U(\boldsymbol{\theta}_I, \boldsymbol{\theta}_J) \nabla_{\boldsymbol{p}_I} K(\boldsymbol{p}_I)\end{aligned} \tag{S16}$$

(The derivatives of K with respect to $\boldsymbol{p}_I$ are independent of $\boldsymbol{p}_J$, so they are written simply as a function of $\boldsymbol{p}_I$ in the expression above.)

Corollary 8. *The local error in Hamiltonian incurred by Algorithm 2 is $O(\epsilon^3)$ when there is no discontinuity of U along the line connecting $\boldsymbol{\theta}_J^0$ and $\boldsymbol{\theta}_J^1$. Otherwise, the local error is $O(\epsilon^2)$.*

Proof of Corollary 8. When there is no discontinuity of U along the line connecting $\boldsymbol{\theta}_J^0$ and $\boldsymbol{\theta}_J^1$, the Taylor expansion of ξ as defined in (S16) with respect to $\boldsymbol{\theta}_J$ implies that

$$\xi\left(\boldsymbol{\theta}_I^{1/2}, \boldsymbol{\theta}_J^1, \boldsymbol{p}_I^{1/2}\right) - \xi\left(\boldsymbol{\theta}_I^{1/2}, \boldsymbol{\theta}_J^0, \boldsymbol{p}_I^{1/2}\right) = O(\|\boldsymbol{\theta}_J^1 - \boldsymbol{\theta}_J^0\|) = O(\epsilon) \tag{S17}$$

So the leading order term of (S15) becomes $O(\epsilon^3)$. □

Proof of Theorem 7. The update $(\boldsymbol{\theta}_J^0, \boldsymbol{p}_J^0) \to (\boldsymbol{\theta}_J^1, \boldsymbol{p}_J^1)$ is energy preserving by the property of the coordinate-wise integrator, so we have

$$\begin{aligned}&H(\boldsymbol{\theta}^1, \boldsymbol{p}^1) - H(\boldsymbol{\theta}^0, \boldsymbol{p}^0) \\ &\quad = H(\boldsymbol{\theta}^1, \boldsymbol{p}^1) - H(\boldsymbol{\theta}_I^{1/2}, \boldsymbol{\theta}_J^1, \boldsymbol{p}_I^{1/2}, \boldsymbol{p}_J^1) + H(\boldsymbol{\theta}_I^{1/2}, \boldsymbol{\theta}_J^0, \boldsymbol{p}_I^{1/2}, \boldsymbol{p}_J^0) - H(\boldsymbol{\theta}^0, \boldsymbol{p}^0)\end{aligned} \tag{S18}$$

Now let $(\boldsymbol{\theta}_I^0(t), \boldsymbol{p}_I^0(t))$ denote the solution of the differential equation

$$\frac{\mathrm{d}\boldsymbol{\theta}_I}{\mathrm{d}t} = \nabla_{\boldsymbol{p}_I} K(\boldsymbol{p}_I, \boldsymbol{p}_J^0), \ \frac{\mathrm{d}\boldsymbol{p}_I}{\mathrm{d}t} = -\nabla_{\boldsymbol{\theta}_I} U(\boldsymbol{\theta}_I, \boldsymbol{\theta}_J^0) \tag{S19}$$

with the initial condition $(\boldsymbol{\theta}_I^0(0), \boldsymbol{p}_I^0(0)) = (\boldsymbol{\theta}_I^0, \boldsymbol{p}_I^0)$. Similarly, let $(\boldsymbol{\theta}_I^{1/2}(t), \boldsymbol{p}_I^{1/2}(t))$ denote the solution of the differential equation

$$\frac{\mathrm{d}\boldsymbol{\theta}_I}{\mathrm{d}t} = \nabla_{\boldsymbol{p}_I} K(\boldsymbol{p}_I, \boldsymbol{p}_J^1), \ \frac{\mathrm{d}\boldsymbol{p}_I}{\mathrm{d}t} = -\nabla_{\boldsymbol{\theta}_I} U(\boldsymbol{\theta}_I, \boldsymbol{\theta}_J^1) \tag{S20}$$

with the initial condition $(\boldsymbol{\theta}_I^{1/2}(0), \boldsymbol{p}_I^{1/2}(0)) = (\boldsymbol{\theta}_I^{1/2}, \boldsymbol{p}_I^{1/2})$. By the energy preserving property of (exact) Hamiltonian dynamics, (S18) becomes

$$\begin{aligned}&H(\boldsymbol{\theta}^1, \boldsymbol{p}^1) - H(\boldsymbol{\theta}^0, \boldsymbol{p}^0) \\ &\quad = H(\boldsymbol{\theta}^1, \boldsymbol{p}^1) - H(\boldsymbol{\theta}_I^{1/2}(\epsilon/2), \boldsymbol{\theta}_J^1, \boldsymbol{p}_I^{1/2}(\epsilon/2), \boldsymbol{p}_J^1) \\ &\qquad + H(\boldsymbol{\theta}_I^{1/2}, \boldsymbol{\theta}_J^0, \boldsymbol{p}_I^{1/2}, \boldsymbol{p}_J^0) - H(\boldsymbol{\theta}_I^0(\epsilon/2), \boldsymbol{\theta}_J^0, \boldsymbol{p}^0(\epsilon/2), \boldsymbol{p}_J^0)\end{aligned} \tag{S21}$$

In essence, (S21) says that the error in Hamiltonian comes only from the numerical approximation errors in solving the differential equations (S19) and (S20). Lemma 9 below quantifies

such errors and its results can be related to the error in Hamiltonian by observing that

$$\begin{aligned}
&H(\boldsymbol{\theta}_I^{1/2}, \boldsymbol{\theta}_J^0, \boldsymbol{p}_I^{1/2}, \boldsymbol{p}_J^0) - H(\boldsymbol{\theta}_I^0(\epsilon/2), \boldsymbol{\theta}_J^0, \boldsymbol{p}_I^0(\epsilon/2), \boldsymbol{p}_J^0) \\
&\quad = U(\boldsymbol{\theta}_I^{1/2}, \boldsymbol{\theta}_J^0) - U(\boldsymbol{\theta}_I^0(\epsilon/2), \boldsymbol{\theta}_J^0) + K(\boldsymbol{p}_I^{1/2}, \boldsymbol{p}_J^0) - K(\boldsymbol{p}_I^0(\epsilon/2), \boldsymbol{p}_J^0) \\
&\quad = \nabla_{\boldsymbol{\theta}_I}^\intercal U(\boldsymbol{\theta}_I^{1/2}, \boldsymbol{\theta}_J^0)\big(\boldsymbol{\theta}_I^{1/2} - \boldsymbol{\theta}_I^0(\epsilon/2)\big) + \nabla_{\boldsymbol{p}_I}^\intercal K(\boldsymbol{p}_I^{1/2}, \boldsymbol{p}_J^0)\big(\boldsymbol{p}_I^{1/2} - \boldsymbol{p}_I^0(\epsilon/2)\big) \\
&\qquad + O\big(\|\boldsymbol{\theta}_I^{1/2} - \boldsymbol{\theta}_I(\epsilon/2)\|^2\big) + O\big(\|\boldsymbol{p}_I^{1/2} - \boldsymbol{p}_I(\epsilon/2)\|^2\big)
\end{aligned} \tag{S22}$$

Now applying (S27) of Lemma 9 with $\tilde{\epsilon} = \epsilon/2$, $(\boldsymbol{\theta}_I, \boldsymbol{p}_I) = (\boldsymbol{\theta}_I^0, \boldsymbol{p}_I^0)$, and $(\boldsymbol{\theta}_I^*, \boldsymbol{p}_I^*) = (\boldsymbol{\theta}_I^{1/2}, \boldsymbol{p}_I^{1/2})$, we obtain

$$\begin{aligned}
&H(\boldsymbol{\theta}_I^{1/2}, \boldsymbol{\theta}_J^0, \boldsymbol{p}_I^{1/2}, \boldsymbol{p}_J^0) - H(\boldsymbol{\theta}_I^0(\epsilon/2), \boldsymbol{\theta}_J^0, \boldsymbol{p}_I^0(\epsilon/2), \boldsymbol{p}_J^0) \\
&\quad = -\frac{\epsilon^2}{8}\nabla_{\boldsymbol{\theta}_I}^\intercal U(\boldsymbol{\theta}_I^{1/2}, \boldsymbol{\theta}_J^0)\,\boldsymbol{\mathcal{I}}_K(\boldsymbol{p}_I^{1/2}, \boldsymbol{p}_J^0)\nabla_{\boldsymbol{\theta}_I} U(\boldsymbol{\theta}_I^{1/2}, \boldsymbol{\theta}_J^0) \\
&\qquad + \frac{\epsilon^2}{8}\nabla_{\boldsymbol{p}_I}^\intercal K(\boldsymbol{p}_I^{1/2}, \boldsymbol{p}_J^0)\boldsymbol{\mathcal{I}}_U(\boldsymbol{\theta}_I^{1/2}, \boldsymbol{\theta}_J^0)\nabla_{\boldsymbol{p}_I} K(\boldsymbol{p}_I^{1/2}, \boldsymbol{p}_J^0) + O(\epsilon^3)
\end{aligned} \tag{S23}$$

In a similar manner, it follows from (S29) of Lemma 9 that

$$\begin{aligned}
&H(\boldsymbol{\theta}_I^1, \boldsymbol{\theta}_J^1, \boldsymbol{p}_I^1, \boldsymbol{p}_J^1) - H(\boldsymbol{\theta}_I^{1/2}(\epsilon/2), \boldsymbol{\theta}_J^1, \boldsymbol{p}_I^{1/2}(\epsilon/2), \boldsymbol{p}_J^1) \\
&\quad = \frac{\epsilon^2}{8}\nabla_{\boldsymbol{\theta}_I}^\intercal U(\boldsymbol{\theta}_I^1, \boldsymbol{\theta}_J^1)\,\boldsymbol{\mathcal{I}}_K(\boldsymbol{p}_I^{1/2}, \boldsymbol{p}_J^1)\nabla_{\boldsymbol{\theta}_I} U(\boldsymbol{\theta}_I^{1/2}, \boldsymbol{\theta}_J^1) \\
&\qquad - \frac{\epsilon^2}{8}\nabla_{\boldsymbol{p}_I}^\intercal K(\boldsymbol{p}_I^1, \boldsymbol{p}_J^1)\boldsymbol{\mathcal{I}}_U(\boldsymbol{\theta}_I^{1/2}, \boldsymbol{\theta}_J^1)\nabla_{\boldsymbol{p}_I} K(\boldsymbol{p}_I^{1/2}, \boldsymbol{p}_J^1) + O(\epsilon^3) \\
&\quad = \frac{\epsilon^2}{8}\nabla_{\boldsymbol{\theta}_I}^\intercal U(\boldsymbol{\theta}_I^{1/2}, \boldsymbol{\theta}_J^1)\,\boldsymbol{\mathcal{I}}_K(\boldsymbol{p}_I^{1/2}, \boldsymbol{p}_J^1)\nabla_{\boldsymbol{\theta}_I} U(\boldsymbol{\theta}_I^{1/2}, \boldsymbol{\theta}_J^1) \\
&\qquad - \frac{\epsilon^2}{8}\nabla_{\boldsymbol{p}_I}^\intercal K(\boldsymbol{p}_I^{1/2}, \boldsymbol{p}_J^1)\boldsymbol{\mathcal{I}}_U(\boldsymbol{\theta}_I^{1/2}, \boldsymbol{\theta}_J^1)\nabla_{\boldsymbol{p}_I} K(\boldsymbol{p}_I^{1/2}, \boldsymbol{p}_J^1) + O(\epsilon^3)
\end{aligned} \tag{S24}$$

The result (S15) now follows by simply noting that the derivatives of K with respect to $\boldsymbol{p}_I$ are independent of $\boldsymbol{p}_J$. □

Lemma 9. *For $(\boldsymbol{\theta}_J, \boldsymbol{p}_J)$ fixed, let $(\boldsymbol{\theta}_I(t), \boldsymbol{p}_I(t))$ denote the solution of the differential equation*

$$\frac{d\boldsymbol{\theta}_I}{dt} = \nabla_{\boldsymbol{p}_I} K(\boldsymbol{p}_I, \boldsymbol{p}_J),\ \ \frac{d\boldsymbol{p}_I}{dt} = -\nabla_{\boldsymbol{\theta}_I} U(\boldsymbol{\theta}_I, \boldsymbol{\theta}_J) \tag{S25}$$

with the initial condition $(\boldsymbol{\theta}_I(0), \boldsymbol{p}_I(0)) = (\boldsymbol{\theta}_I, \boldsymbol{p}_I)$. The approximation error of the numerical scheme

$$\boldsymbol{\theta}_I^* = \boldsymbol{\theta}_I + \tilde{\epsilon}\nabla_{\boldsymbol{p}_I} K(\boldsymbol{p}_I^*, \boldsymbol{p}_J),\ \ \boldsymbol{p}_I^* = \boldsymbol{p}_I - \tilde{\epsilon}\nabla_{\boldsymbol{\theta}_I} U(\boldsymbol{\theta}_I, \boldsymbol{\theta}_J) \tag{S26}$$

satisfies

$$\begin{aligned}
\boldsymbol{\theta}_I^* - \boldsymbol{\theta}_I(\tilde{\epsilon}) &= -\frac{\tilde{\epsilon}^2}{2}\boldsymbol{\mathcal{I}}_K(\boldsymbol{p}_I^*, \boldsymbol{p}_J)\nabla_{\boldsymbol{\theta}_I} U(\boldsymbol{\theta}_I^*, \boldsymbol{\theta}_J) + O(\tilde{\epsilon}^3) \\
\boldsymbol{p}_I^* - \boldsymbol{p}_I(\tilde{\epsilon}) &= \frac{\tilde{\epsilon}^2}{2}\boldsymbol{\mathcal{I}}_U(\boldsymbol{\theta}_I^*, \boldsymbol{\theta}_J)\nabla_{\boldsymbol{p}_I} K(\boldsymbol{p}_I^*, \boldsymbol{p}_J) + O(\tilde{\epsilon}^3)
\end{aligned} \tag{S27}$$

where $\boldsymbol{\mathcal{I}}_U = \partial^2 U/\partial \boldsymbol{\theta}_I^2$ *and* $\boldsymbol{\mathcal{I}}_K = \partial^2 K/\partial \boldsymbol{p}_I^2$ *are the hessians of* U *and* K *with respect to* $\boldsymbol{\theta}_I$ *and* $\boldsymbol{p}_I$*. Similarly, the approximation error of the numerical scheme*

$$\boldsymbol{\theta}_I^* = \boldsymbol{\theta}_I + \tilde{\epsilon}\nabla_{\boldsymbol{p}_I} K(\boldsymbol{p}_I, \boldsymbol{p}_J),\ \boldsymbol{p}_I^* = \boldsymbol{p}_I - \tilde{\epsilon}\nabla_{\boldsymbol{\theta}_I} U(\boldsymbol{\theta}_I^*, \boldsymbol{\theta}_J) \tag{S28}$$

satisfies

$$\begin{aligned} \boldsymbol{\theta}_I^* - \boldsymbol{\theta}_I(\tilde{\epsilon}) &= \frac{\tilde{\epsilon}^2}{2}\boldsymbol{\mathcal{I}}_K(\boldsymbol{p}_I, \boldsymbol{p}_J)\nabla_{\boldsymbol{\theta}_I} U(\boldsymbol{\theta}_I, \boldsymbol{\theta}_J) + O(\tilde{\epsilon}^3) \\ \boldsymbol{p}_I^* - \boldsymbol{p}_I(\tilde{\epsilon}) &= -\frac{\tilde{\epsilon}^2}{2}\boldsymbol{\mathcal{I}}_U(\boldsymbol{\theta}_I, \boldsymbol{\theta}_J)\nabla_{\boldsymbol{p}_I} K(\boldsymbol{p}_I, \boldsymbol{p}_J) + O(\tilde{\epsilon}^3) \end{aligned} \tag{S29}$$

Proof. The proofs of (S27) and (S29) are very similar, so we focus on the derivations of (S27). The Taylor expansion of $\boldsymbol{\theta}_I(t)$ yields

$$\begin{aligned} \boldsymbol{\theta}_I(\tilde{\epsilon}) - \boldsymbol{\theta}_I &= \tilde{\epsilon}\frac{\mathrm{d}\boldsymbol{\theta}}{\mathrm{d}t} + \frac{\tilde{\epsilon}^2}{2}\frac{\mathrm{d}^2\boldsymbol{\theta}}{\mathrm{d}t^2} + O(\tilde{\epsilon}^3) \\ &= \tilde{\epsilon}\nabla_{\boldsymbol{p}_I} K(\boldsymbol{p}_I, \boldsymbol{p}_J) - \frac{\tilde{\epsilon}^2}{2}\boldsymbol{\mathcal{I}}_K(\boldsymbol{p}_I, \boldsymbol{p}_J)\nabla_{\boldsymbol{\theta}_I} U(\boldsymbol{\theta}_I, \boldsymbol{\theta}_J) + O(\tilde{\epsilon}^3) \end{aligned} \tag{S30}$$

On the other hand, the Taylor expansion of $\nabla_{\boldsymbol{p}_I} K(\boldsymbol{p}_I^*, \boldsymbol{p}_J)$ in the first variable yields

$$\begin{aligned} \boldsymbol{\theta}_I^* - \boldsymbol{\theta}_I &= \tilde{\epsilon}\nabla_{\boldsymbol{p}_I} K(\boldsymbol{p}_I, \boldsymbol{p}_J) + \tilde{\epsilon}\boldsymbol{\mathcal{I}}_K(\boldsymbol{p}_I, \boldsymbol{p}_J)(\boldsymbol{p}_I^* - \boldsymbol{p}_I) + \tilde{\epsilon}\, O(\|\boldsymbol{p}_I^* - \boldsymbol{p}_I\|^2) \\ &= \tilde{\epsilon}\nabla_{\boldsymbol{p}_I} K(\boldsymbol{p}_I, \boldsymbol{p}_J) - \tilde{\epsilon}^2\boldsymbol{\mathcal{I}}_K(\boldsymbol{p}_I, \boldsymbol{p}_J)\nabla_{\boldsymbol{\theta}_I} U(\boldsymbol{\theta}_I, \boldsymbol{\theta}_J) + O(\tilde{\epsilon}^3) \end{aligned} \tag{S31}$$

Subtracting (S30) from (S31), we obtain

$$\begin{aligned} \boldsymbol{\theta}_I^* - \boldsymbol{\theta}_I(\tilde{\epsilon}) &= -\frac{\tilde{\epsilon}^2}{2}\boldsymbol{\mathcal{I}}_K(\boldsymbol{p}_I, \boldsymbol{p}_J)\nabla_{\boldsymbol{\theta}_I} U(\boldsymbol{\theta}_I, \boldsymbol{\theta}_J) + O(\tilde{\epsilon}^3) \\ &= -\frac{\tilde{\epsilon}^2}{2}\boldsymbol{\mathcal{I}}_K(\boldsymbol{p}_I^*, \boldsymbol{p}_J)\nabla_{\boldsymbol{\theta}_I} U(\boldsymbol{\theta}_I^*, \boldsymbol{\theta}_J) + O(\tilde{\epsilon}^3) \end{aligned} \tag{S32}$$

where the second equality again follows from a Taylor expansion applied to the leading order term. The error estimate for the momentum variable is similar; the Taylor expansion of $\boldsymbol{p}_I(t)$ gives

$$\boldsymbol{p}_I(\tilde{\epsilon}) - \boldsymbol{p}_I = -\tilde{\epsilon}\nabla_{\boldsymbol{\theta}_I} U(\boldsymbol{\theta}_I, \boldsymbol{\theta}_J) - \frac{\tilde{\epsilon}^2}{2}\boldsymbol{\mathcal{I}}_U(\boldsymbol{\theta}_I, \boldsymbol{\theta}_J)\nabla_{\boldsymbol{p}_I} K(\boldsymbol{p}_I, \boldsymbol{p}_J) + O(\tilde{\epsilon}^3) \tag{S33}$$

Subtracting (S33) from (S26), we obtain

$$\begin{aligned} \boldsymbol{p}_I^* - \boldsymbol{p}_I(\tilde{\epsilon}) &= \frac{\tilde{\epsilon}^2}{2}\boldsymbol{\mathcal{I}}_U(\boldsymbol{\theta}_I, \boldsymbol{\theta}_J)\nabla_{\boldsymbol{p}_I} K(\boldsymbol{p}_I, \boldsymbol{p}_J) + O(\tilde{\epsilon}^3) \\ &= \frac{\tilde{\epsilon}^2}{2}\boldsymbol{\mathcal{I}}_U(\boldsymbol{\theta}_I^*, \boldsymbol{\theta}_J)\nabla_{\boldsymbol{p}_I} K(\boldsymbol{p}_I^*, \boldsymbol{p}_J) + O(\tilde{\epsilon}^3) \end{aligned} \tag{S34}$$

□

S6.2 Global error in Hamiltonian

Theorem 10 below establishes the global error in Hamiltonian to be $O(\epsilon^2)$. For its proof, we recall that Algorithm 2 is designed under the assumption that the parameter space has a partition $\mathbb{R}^{|I|} \times \mathbb{R}^{|J|} = \cup_k \mathbb{R}^{|I|} \times \Omega_k$ such that $U(\boldsymbol{\theta})$ is smooth on $\mathbb{R}^{|I|} \times \Omega_k$ for each k. Below, in relating the local error to the global one, we make the dependence of a numerical solution on a stepsize ϵ explicit and denote the value of a numerical solution after ℓ steps by $(\boldsymbol{\theta}_\epsilon^\ell, \boldsymbol{p}_\epsilon^\ell)$.

Theorem 10. *Suppose that each Ω_k is rectangular i.e. its boundary consists of planes perpendicular to one of the coordinates of $\boldsymbol{\theta}_J$. Then the global error $H(\boldsymbol{\theta}_\epsilon^L, \boldsymbol{p}_\epsilon^L) - H(\boldsymbol{\theta}^0, \boldsymbol{p}^0)$ with $L = L(\epsilon) = \lfloor \tau/\epsilon \rfloor$ incurred by Algorithm 2 is of order $O(\epsilon^2 D)$ where D is the number of discontinuities in U encountered along the trajectory $\{\boldsymbol{\theta}(t), 0 \le t \le \tau\}$.*

The assumption stated in Theorem 10 is required for our proof of Theorem 11 and is satisfied whenever π is a discontinuous target distribution obtained by the embedding of discrete parameters described in Section 2.1. We believe the order of the global error to remain unchanged under more general conditions, but the assumption is made for convenience.

Proof. The global error is given as a sum of the local errors:

$$H(\boldsymbol{\theta}_\epsilon^L, \boldsymbol{p}_\epsilon^L) - H(\boldsymbol{\theta}^0, \boldsymbol{p}^0) = \sum_{\ell=1}^{L} \left(H(\boldsymbol{\theta}_\epsilon^\ell, \boldsymbol{p}_\epsilon^\ell) - H(\boldsymbol{\theta}_\epsilon^{\ell-1}, \boldsymbol{p}_\epsilon^{\ell-1}) \right) \tag{S35}$$

Let $D(\epsilon)$ denote the size of the set $\mathcal{D}_\epsilon$ as defined below:

$$\begin{aligned} \mathcal{D}_\epsilon = \Big\{ & \ell \in \{1, \dots, L\} : \\ & \boldsymbol{\theta}_{\epsilon,J}^\ell \text{ and } \boldsymbol{\theta}_{\epsilon,J}^{\ell-1} \text{ belong to two separate regions of the partition } \Omega_k\text{'s} \Big\} \end{aligned} \tag{S36}$$

By the result of Corollary 8, we know that the local error is $O(\epsilon^2)$ if $\ell \in \mathcal{D}_\epsilon$ and of $O(\epsilon^3)$ otherwise. Therefore, (S35) is a sum of $D(\epsilon)$ terms of $O(\epsilon^2)$ errors and $L(\epsilon) - D(\epsilon)$ terms of $O(\epsilon^3)$ errors, yielding the global error of $O(D(\epsilon)\epsilon^2)$. To complete the proof, it follows from Theorem 11 that $D(\epsilon)$ as $\epsilon \to 0$ converges to the number of discontinuities in U encountered along the trajectory $\{\boldsymbol{\theta}(t), 0 \le t \le \tau\}$. □

Theorem 11. *Under the assumption of Theorem 10, we have*

$$\sup_{\ell=1,\dots,L} \left\| (\boldsymbol{\theta}(\ell\epsilon), \boldsymbol{p}(\ell\epsilon)) - (\boldsymbol{\theta}_\epsilon^\ell, \boldsymbol{p}_\epsilon^\ell) \right\| = O(\epsilon) \tag{S37}$$

Proof. First note that the trajectory of Hamiltonian dynamics corresponding to the kinetic energy (17) can be partitioned into $\widetilde{D}$ segments $\{\boldsymbol{\theta}(t); t_m < t < t_{m+1}\}_m$ for $0 = t_0 < t_1 < \ldots < t_{\widetilde{D}} = \tau$ so that on each segment $\mathrm{d}\boldsymbol{\theta}_J/\mathrm{d}t = \boldsymbol{m}_J^{-1} \odot \mathrm{sign}(\boldsymbol{p}_J)$ is constant.

The numerical solution approximate the exact solution $\boldsymbol{\theta}(t) \to \boldsymbol{\theta}(t + \ell\epsilon)$ up to an error of $O(\epsilon^2)$ for any ℓ provided that $\boldsymbol{\theta}(t)$ and $\boldsymbol{\theta}(t + \ell\epsilon)$ belongs to the same segment $\{\boldsymbol{\theta}(t); t_m < t < t_{m+1}\}$. This is for the following reason. For all sufficiently small ϵ, the coordinate-wise updates of discontinuous parameters yields the exact solution to

$$\frac{\mathrm{d}\boldsymbol{\theta}_J}{\mathrm{d}t} = \boldsymbol{m}_J^{-1} \odot \mathrm{sign}(\boldsymbol{p}_J), \quad \frac{\mathrm{d}\boldsymbol{p}_J}{\mathrm{d}t} = -\nabla_{\boldsymbol{\theta}_J} U(\boldsymbol{\theta}), \quad \frac{\mathrm{d}\boldsymbol{\theta}_{-J}}{\mathrm{d}t} = \frac{\mathrm{d}\boldsymbol{p}_{-J}}{\mathrm{d}t} = \mathbf{0} \tag{S38}$$

provided no sign change in $\boldsymbol{p}_J$ is encountered. In this case, Algorithm 2 coincides with a symmetric splitting of Hamilton's equation in which the individual components are solved exactly and hence the numerical approximation of $\boldsymbol{\theta}(t) \to \boldsymbol{\theta}(t+\epsilon)$ locally agrees with the exact solution up to an error of $O(\epsilon^3)$ (Leimkuhler and Reich, 2005).

On the other hand, when $\boldsymbol{\theta}(t)$ and $\boldsymbol{\theta}(t+\epsilon)$ does not belong to the same segment, the coordinate-wise integrator approximates the change in $\mathrm{d}\boldsymbol{\theta}_J/\mathrm{d}t$ through the momentum flip $p_j \to -p_j$ for an appropriate j while θ_j held fixed. This approximation is accurate up to an error of $O(\epsilon)$. (When a discontinuity of U exists along the path $\{\boldsymbol{\theta}(s); t < s < t+\epsilon\}$, this error estimate holds only under the assumption on the boundaries of Ω_k's.)

To summarize, we have shown that the total accumulated error is $O(\epsilon^2)$ while the solution stays within the same segment $\{\boldsymbol{\theta}(t); t_m < t < t_{m+1}\}_m$ and then an additional error of $O(\epsilon)$ is incurred when crossing from one segment to another. Since the solution trajectory consists of $\widetilde{D}$ such segments, the total accumulated error is $O(\widetilde{D}(\epsilon + \epsilon^2)) = O(\widetilde{D}\epsilon)$. □

References for Supplement

Afshar, H. M. and Domke, J. (2015) Reflection, refraction, and Hamiltonian Monte Carlo. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, 3007–3015.

Berger, J. O., Bernardo, J. M. and Sun, D. (2012) Objective priors for discrete parameter spaces. *Journal of the American Statistical Association*, **107**, 636–648.

Carvalho, C. M., Polson, N. G. and Scott, J. G. (2010) The horseshoe estimator for sparse signals. *Biometrika*, **97**, 465–480.

Fryzlewicz, P. and Subba Rao, S. (2014) Multiple-change-point detection for auto-regressive conditional heteroscedastic processes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **76**, 903–924.

Gelman, A. (2006) Prior distributions for variance parameters in hierarchical models. *Bayesian Analysis*, **1**, 515–534.

Gelman, A., Roberts, G. O. and Gilks, W. R. (1996) Efficient Metropolis jumping rules. *Bayesian Statistics*, **5**, 599–607.

Gram-Hansen, B., Zhou, Y., Kohn, T., Yang, H. and Wood, F. (2018) Discontinuous Hamiltonian Monte Carlo for probabilistic programs. *arXiv:1804.03523*.

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

Leimkuhler, B. and Reich, S. (2005) *Simulating Hamiltonian Dynamics*. Cambridge University Press.

Neal, R. M. (2010) MCMC using Hamiltonian Dynamics. In *Handbook of Markov chain Monte Carlo*. CRC Press.

Nishimura, A. and Dunson, D. (2015) Recycling intermediate steps to improve Hamiltonian Monte Carlo. *arXiv:1511.06925*.

Pakman, A. and Paninski, L. (2013) Auxiliary-variable exact Hamiltonian Monte Carlo samplers for binary distributions. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, 2490–2498.

Schwarz, C. J. and Arnason, A. N. (1996) A general methodology for the analysis of capture-recapture experiments in open populations. *Biometrics*, 860–873.

Seber, G. A. F. (1982) *The estimation of animal abundance*. Griffin London.