Reversible Jump PDMP Samplers for Variable Selection

Augustin Chevallier, Paul Fearnhead and Matt Sutton*

July 11, 2022

Abstract

A new class of Markov chain Monte Carlo (MCMC) algorithms, based on simulating piecewise deterministic Markov processes (PDMPs), has recently shown great promise: they are non-reversible, can mix better than standard MCMC algorithms, and can use subsampling ideas to speed up computation in big data scenarios. However, current PDMP samplers can only sample from posterior densities that are differentiable almost everywhere, which precludes their use for model choice. Motivated by variable selection problems, we show how to develop reversible jump PDMP samplers that can jointly explore the discrete space of models and the continuous space of parameters. Our framework is general: it takes any existing PDMP sampler, and adds two types of trans-dimensional moves that allow for the addition or removal of a variable from the model. We show how the rates of these trans-dimensional moves can be calculated so that the sampler has the correct invariant distribution. We remove a variable from a model when the associated parameter is zero, and this means

*This research was supported by EPSRC grants EP/R018561 and EP/R034710.

that the rates of the trans-dimensional moves do not depend on the likelihood. It is, thus, easy to implement a reversible jump version of any PDMP sampler that can explore a fixed model.

Keywords: Bayesian Statistics; Bouncy Particle Sampler; Model Choice; Monte Carlo; Zig Zag Algorithm

1 Introduction

There is currently much interest in developing MCMC algorithms based on simulating piecewise deterministic Markov processes (PDMPs). These are continuous time Markov processes that have deterministic dynamics between a set of event times, and the randomness in these processes only comes through the random event times and potentially random transitions at the events (see Davis 1993, for an introduction to PDMPs).

The idea of simulating PDMPs to sample from a target distribution of interest originated in statistical physics (Peters & de With 2012, Michel et al. 2014), but has recently been proposed as an alternative to standard MCMC to sample from posterior distributions in Bayesian Statistics, with algorithms such as the Bouncy Particle Sampler (Bouchard-Côté et al. 2018) and the ZigZag algorithm (Bierkens & Roberts 2017, Bierkens et al. 2019) amongst others (Vanetti et al. 2017, Markovic & Sepehri 2018, Wu & Robert 2020, Michel et al. 2020, Bierkens et al. 2020). See Fearnhead et al. (2018) for an introduction to this area.

To sample from a density $\pi(\boldsymbol{\theta})$ most current PDMP samplers introduce a velocity component, $\boldsymbol{v}$, of the same dimension as $\boldsymbol{\theta}$, and have deterministic dynamics that correspond to a constant velocity model (though see Vanetti et al. 2017, for alternative PDMP algorithms). At the random events the velocity component changes. Algorithms differ in terms of the event rate and how the velocity changes at each event, but each has a simple

recipe for choosing these so that the resulting PDMP has $\pi(\boldsymbol{\theta})$ as its invariant distribution. These recipes depend on $\pi(\boldsymbol{\theta})$ through the gradient of $\log \pi(\boldsymbol{\theta})$, which importantly means that $\pi(\boldsymbol{\theta})$ only needs to be known up to proportionality, but also that $\pi(\boldsymbol{\theta})$ needs to be differentiable almost everywhere. The advantages of PDMP samplers are that they are non-reversible, and thus can mix more quickly than standard reversible MCMC algorithms (Diaconis et al. 2000), and, when sampling from posterior distributions, they can use a small sample of data points at each iteration whilst still targeting the true posterior distribution (Bierkens et al. 2019).

However, the restriction to sampling from densities that are differentiable means that current PDMP samplers cannot be used in model choice problems. The aim of this paper is to address this limitation, with a particular motivation of PDMP samplers that can be used in variable selection problems that are common in, for example, linear regression and generalized linear models. We show how to design efficient PDMP samplers which allow movement between different models.

A simple way to implement PDMP samplers for variable selection problems is to use continuous spike-and-slab priors on the parameters (Ishwaran & Rao 2005, George & McCulloch 1993), which, rather than setting some parameters exactly to 0, have priors that place substantial mass close to 0. With such a prior, the resulting posterior density is differentiable, and existing PDMP samplers can be used (Goldman et al. 2021). However such an approach has three disadvantages. First, under such a prior it can be hard to interpret the results as we do not formally get posterior probabilities on whether certain variables should be included in the model. Second, they introduce an extra tuning parameter to the prior which governs the shape of the spike of the component. Third, as we show in Section 2.2, using PDMP samplers to sample from the resulting posterior can be computationally inefficient: the samplers will need to simulate many events so that the parameters associated to variables that should not be in the model are kept close to 0.

We demonstrate how to adapt existing PDMP samplers to variable selection problems. Specifically they evolve as the PDMP sampler when exploring the posterior associated with a given model, but with two additional events: if any parameter value hits 0 the PDMP jumps to the smaller model where the corresponding variable is removed; whilst with some rate there are events that re-introduce variables into the model. We show in Section 3 how to calculate the rate and transition for these new types of event so that the sampler has the correct invariant distribution. To calculate these we need different techniques than those used for existing PDMP samplers, as we need to account for the behaviour of the process when parameters hit zero. The techniques we use are most similar to those in Bierkens et al. (2018), which considers PDMPs with restricted domains. However in that paper the dynamics at the boundary of the domain could be chosen so that the net flow of probability at the boundary is zero; whereas we need to balance the probability flow out of a model which occurs when a parameter hits zero with the flow into the model caused by the events that re-introduce variables.

Our approach is not the only way of extending PDMP samplers to the variable selection problem. One could also introduce moves that propose adding or deleting a variable from the model regardless of the current state of the PDMP. Such ideas have been proposed for other continuous-time samplers, see Grenander & Miller (1994) or Phillips & Smith (1996) for their use within jump-diffusion samplers, and Stephens (2000) for their use within continuous-time Markov jump process samplers. Exactly the same type of moves between models could be implemented for PDMPs. However we believe our approach, that only allows removing variables when the corresponding parameter value hits zero, has important practical advantages. Most importantly, because we add or remove a variable when its corresponding parameter is 0 the rates of these events do not depend on the likelihood, but only the prior. This makes simulating these events simple – in fact for many common priors the rate at which we add a variable will be constant. Thus any PDMP

sampler suitable for a fixed model can easily be extended to allow for variable selection. By comparison the moves and rates described in Grenander & Miller (1994) would involve rates that depend on the likelihood ratio between the current and new model. Simulating events with this rate will be challenging, as the rate will vary, in a complicated way, when the parameter values change. Calculating this rate will be costly as it requires evaluating the log-likelihood, which is a sum over the data points. Furthermore, most algorithms to simulate events require bounds on the rates, and calculating good bounds will be problem specific and potentially difficult. After a preprint of this work appeared (Chevallier et al. 2020), a related method, called the sticky PDMP (Bierkens et al. 2021), has been proposed. For variable selection problems, the main difference between the two approaches is that the sticky PDMP sampler remembers the velocity of a component when it is re-added back into the model, and thus will continue in the same direction as it was moving before that variable was removed.

The approach we present is generic, in that it can take any current PDMP sampler and be used to obtain a version that can be applied to the variable selection problem. We call the new class of samplers reversible jump PDMP samplers, due to the analogy with reversible jump MCMC (Green 1995), though the class of moves we allow are less general than those available for standard reversible jump MCMC. We show how to derive reversible-jump versions of both ZigZag and the Bouncy Particle Sampler in Section 4, before investigating empirically these algorithms on both logistic regression and robust linear regression models.

Proofs of all theorems are relegated to the appendix. Code for implementing the new reversible jump PDMP samplers is available from the `R` package `rjpdmp` available from the Comprehensive R Archive Network (CRAN).

2 PDMP Samplers and Model Choice

2.1 Variable Selection

We will consider model selection problems that arise from variable selection. The general framework is that we have a vector of parameters, $\boldsymbol{\theta} = (\theta_1, ..., \theta_d)$, and each model is characterised by setting some subset of the θ_js to 0. This is a common setting across linear models, generalised linear models and various extensions.

To make ideas concrete, consider a linear model

$$\boldsymbol{Y} = \sum_{j=1}^{d} \boldsymbol{X}_j \theta_j + \epsilon$$

where $\boldsymbol{Y}$ is a vector of response variables, each $\boldsymbol{X}_j$ is a vector of covariates, and ϵ is an additive noise vector. When d is large it is common to fit such a model under a sparsity assumption, namely that many of the θ_js are 0.

In a Bayesian analysis, such a sparsity assumption is encapsulated in our choice of prior on $\boldsymbol{\theta}$. To aid interpretation of the variable selection priors it is common to introduce a latent variable $\boldsymbol{\gamma} = (\gamma_1, ..., \gamma_d)'$ where $\gamma_j = 1$ if the covariate $\boldsymbol{X}_j$ is included in the model, i.e. if $\theta_j \neq 0$. We let $|\boldsymbol{\gamma}| = \sum \gamma_j$ be the number of covariates included in the corresponding model. Indexing $\boldsymbol{\theta}_{\boldsymbol{\gamma}}$ as the sub-vector of $\boldsymbol{\theta}$ with only the selected variables, any prior can be written in a hierarchical form where we have a prior on $\boldsymbol{\gamma}$, then conditional on $\boldsymbol{\gamma}$ we have a prior on $\boldsymbol{\theta}_{\boldsymbol{\gamma}}$, and set all remaining entries of $\boldsymbol{\theta}$ to 0.

A special case is where each component of $\boldsymbol{\theta}$ is independent of the others. In which case the prior can be written as

$$\theta_j \sim w_j g_j(\theta_j) + (1 - w_j)\delta_0(\theta_j), \qquad j = 1, ..., d,$$

where $w_j \in (0,1)$ is the prior probability that $\gamma_j = 1$, δ_0 is a Dirac measure at zero and $g_j(\theta_j)$ is a distribution that models our prior beliefs for θ_j conditional on that variable being included in the model. Bayesian approaches to variable selection that put a probability mass on $\theta_j = 0$ in this way will be referred to as Dirac spike and slab methods. Notable examples of these methods include Mitchell & Beauchamp (1988), Kuo & Mallick (1998), Geweke (1996), Smith & Kohn (1996) and Bottolo & Richardson (2010).

While this formulation is natural from a modelling perspective, sampling from the resulting posterior distribution can be challenging, with, for example, MCMC samplers that use gradient information such as Hamiltonian Monte Carlo (Neal 2011) not being applicable. To circumvent this issue it is common to use an approximation to this prior which replaces the point mass at 0 with a density that is peaked around 0, such as

$$\theta_j \sim w_j \mathcal{N}(0, \tau_j^2) + (1 - w_j)\mathcal{N}(0, \tau_j^2 c_j^2), \qquad j = 1, ..., d, \tag{1}$$

where c_j is taken small so that $\mathcal{N}(0, c_j^2 \tau_j^2)$ approximates the Dirac spike. We will refer to Bayesian variable selection methods that replace the Dirac in the prior with a continuous approximation as continuous spike and slab methods.

2.2 PDMP Samplers

Piecewise Deterministic Markov Processes (PDMPs) are an emerging class of non-reversible continuous-time samplers. We will consider sampling from a distribution with density $\pi(\boldsymbol{\theta})$ defined on some space $\mathcal{X}$. Current samplers augment the state to include a velocity vector and sample from a distribution on $E = \mathcal{X} \times \mathcal{V}$. In the following, for $\boldsymbol{z} \in E$ we will use the notation $\boldsymbol{z} = (\boldsymbol{\theta}, \boldsymbol{v})$ with $\boldsymbol{\theta} \in \mathcal{X}$ a position and $\boldsymbol{v} \in \mathcal{V}$ a velocity.

A PDMP can be defined by (i) deterministic dynamics between a set of random event times; (ii) the state-dependent rate at which events occur, $\lambda(\boldsymbol{z})$; and (iii) a probability

distribution for the change in state at each event, with density $q(\boldsymbol{z}'|\boldsymbol{z})$. We will consider PDMPs whose deterministic trajectories follow a differential equation of the form:

$$\frac{\mathrm{d}(\boldsymbol{\theta}_t, \boldsymbol{v}_t)}{\mathrm{d}t} = (\boldsymbol{v}_t, \Phi(\boldsymbol{\theta}_t, \boldsymbol{v}_t))$$

with $\Phi : E \to \mathcal{V}$ a smooth function. This setting contains the usual PDMP samplers such as ZigZag (Bierkens et al. 2019), the Bouncy Particle Sampler (Bouchard-Côté et al. 2018), or the Coordinate Sampler (Wu & Robert 2020).

For example, the ZigZag algorithm to simulate from $\pi(\boldsymbol{\theta})$ for $\boldsymbol{\theta} \in \mathbb{R}^d$, would introduce a velocity vector $\boldsymbol{v} \in \{-1, 1\}^d$, and deterministic dynamics

$$\frac{\mathrm{d}(\boldsymbol{\theta}_t, \boldsymbol{v}_t)}{\mathrm{d}t} = (\boldsymbol{v}_t, 0),$$

which are the dynamics of a constant velocity model. For ZigZag, events occur at a rate that depends on the gradient of $\pi(\boldsymbol{\theta})$ in each component of the velocity, and at an event one component of the velocity is switched. The rate at which the jth component of the velocity, $\boldsymbol{v}_t^j$, is switched is just

$$\max\left\{0, -\boldsymbol{v}_t^j \frac{\partial}{\partial \boldsymbol{\theta}^j} \log \pi(\boldsymbol{\theta}_t)\right\}.$$

These rates depend on the target distribution just through the gradient of the log of the target – which importantly means that we need only know the target distribution up to proportionality. Algorithm 1 gives computational pseudocode for simulating from the ZigZag process.

We can apply current PDMPs, such as ZigZag, to the Bayesian variable selection problem if we use the continuous spike-and-slab prior (1). Realisations of such a sampler are shown in the left two plots of Figure 1 as we vary how concentrated the spike distribution

Algorithm 1: ZigZag algorithm

Inputs: initial position, $\boldsymbol{\theta}$, and velocity, $\boldsymbol{v}$; t_{max}
while $t < t_{max}$*:* **do**
  for *each* $i \leq d$ **do**
    Compute t_i the event time associated to the Poisson rate $\lambda_i(t) = \max\left\{0, -\boldsymbol{v}^i \frac{\partial \log \pi}{\partial \boldsymbol{\theta}^i}(\boldsymbol{\theta} + t\boldsymbol{v})\right\}$.
  end
  Set $i_{flip} = \arg\min(t_i)$, $t_{flip} = \min(t_i)$
  Set $\boldsymbol{\theta} = \boldsymbol{\theta} + \boldsymbol{v}(t_{flip} - t)$
  Set $v^{i_{flip}} = -v^{i_{flip}}$
  Set $t = t_{flip}$
end

is. For the more concentrated case, the sampler becomes inefficient as it involves many switching events when the state variable is close to 0.

Intuitively as we make the variance of the spike component of the prior tend to 0 the prior converges to a prior with a point mass at 0; furthermore we can observe the output of our PDMP sampler "converging" to a process shown in the right-hand plot of Figure 1. Here, rather than the state having periods where it oscillates around 0, it has periods of time where $\theta_j = 0$ and thus has many fewer events. This limiting process motivates the new class of PDMP samplers that we develop.

3 Reversible Jump PDMP Samplers

Let $\mathcal{M} = (\mathcal{M}_1, \mathcal{M}_2, ...)$ be a set of models indexed by a parameter k. Each model, $\mathcal{M}_k$, has a corresponding state space $\mathcal{X}^k$ of dimension d_k. For the sake of clarity, we will limit

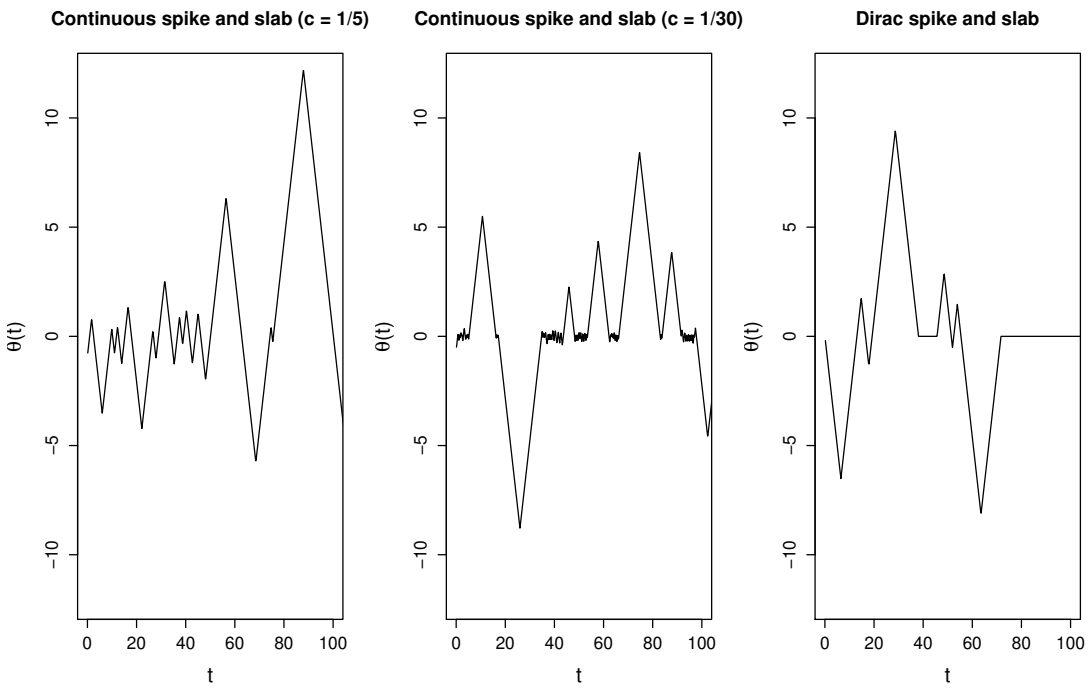

Figure 1: Sample paths of PDMPs implementing variable selection in 1 dimension. The left and centre plots show the trajectories for a continuous spike-and-slab prior $0.5\mathcal{N}(0, \tau^2) + 0.5\mathcal{N}(0, \tau^2 c^2)$ where $\tau^2 = 16$. As c decreases the spike component in the mixture approaches a Dirac mass. The figure on the right is the limiting process where we set the velocity to zero allowing the variable to stay fixed at zero.

ourselves to variable selection. In our case $\mathcal{X}^k$ has a specific form:

$$\mathcal{X}^k = \prod_i \mathbb{R}^{\gamma_i^k}$$

where we abuse notation by using $\mathbb{R}^0 = \{0\}$ and where $(\gamma_i^k)_i$ is a sequence of numbers in $\{0, 1\}$ representing whether a variable is enabled for model $\mathcal{M}_k$. Let π be the target posterior defined on $\mathcal{X} = \cup_k \mathcal{X}^k$. We further assume that the restriction of π to each $\mathcal{X}^k$ has a density, and denote this by $\pi_k(\boldsymbol{\theta})$. The first ingredient of our sampler is a collection

of PDMPs defined for each model. Each PDMP sampler adds a velocity space $\mathcal{V}^k$ to the space $\mathcal{X}^k$ and samples from the space $E_k = \mathcal{X}^k \times \mathcal{V}^k$. Finally, for each model $\mathcal{M}_k$, the associated PDMP sampler has an invariant distribution proportional to ν_k, where

$$\nu_k(\boldsymbol{\theta}, \boldsymbol{v}) = \pi_k(\boldsymbol{\theta}) p_k(\boldsymbol{v}|\boldsymbol{\theta}) \qquad (\boldsymbol{\theta}, \boldsymbol{v}) \in \mathcal{X}^k \times \mathcal{V}^k,$$

for some set of conditional densities $p_k(\boldsymbol{v}|\boldsymbol{\theta})$.

Remark 1. *For samplers such as ZigZag, $p_k(d\boldsymbol{v}|\boldsymbol{\theta})$ is a measure with support on a discrete set – it places probability mass $1/2^k$ on each of the possible 2^k velocities allowed by ZigZag. By choosing $\mathcal{V}^k$ to be the support of $p_k(d\boldsymbol{v}|\boldsymbol{\theta})$, $p_k(d\boldsymbol{v}|\boldsymbol{\theta})$ still has a density – and integrals can be interpreted as sums over the support points. This allows us to treat all samplers within the same framework.*

The second ingredient of our reversible jump PDMP sampler is a set of jumps between models. For our case of variable selection, we only allow adding or removing one variable at a time. Hence, let

$$\mathcal{T} = \{(i, j) \mid \text{there exists a } k \text{ such that } \gamma_k^i = 1, \gamma_k^j = 0; \text{ and } \gamma_l^i = \gamma_l^j, l \neq k\}$$

be a set of pairs of transitions between models, with these ordered (i, j) such that model $\mathcal{M}_j$ is obtained from $\mathcal{M}_i$ by removing one of the variables in $\mathcal{M}_i$. For transition $i \to j$, we define an active boundary $\Gamma_{i,j} = \mathcal{X}^j \times \mathcal{V}^i \subset E_i$, a subspace of E_i. Each trajectory passing through $\Gamma_{i,j}$ has some probability $p_{i,j}$ of jumping to E_j using a deterministic jump function $g_{i,j} : \Gamma_{i,j} \to E_j$. We assume that the jump function does not change the parameter $\boldsymbol{\theta}$. So, if a trajectory $\boldsymbol{z}_t$ of our process has a left limit at time t, $\boldsymbol{z}_{t-}$ in $\Gamma_{i,j}$, then with some probability $p_{i,j}$, $\boldsymbol{z}_t = g_{i,j}(\boldsymbol{z}_{t-}) \in E_j$ with $\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-}$. For transition $j \to i$, we introduce a Poisson rate, $\beta_{i,j}(\boldsymbol{z})$, and a jump kernel, $Q_{i,j}(\cdot|\boldsymbol{z})$, such that if the trajectory is in E_j, then

with rate $\beta_{i,j}(\boldsymbol{z}_{t-})$, $\boldsymbol{z}_t$ is drawn from $Q_{i,j}(\cdot|\boldsymbol{z}_{t-}) \in \Gamma_{i,j}$. We impose symmetry in the jumps between models so that for any $\boldsymbol{z}' \sim Q_{i,j}(\cdot|\boldsymbol{z})$, $g_{i,j}(\boldsymbol{z}') = \boldsymbol{z}$, i.e. the probability measure $Q_{i,j}(\cdot|\boldsymbol{z})$ is supported by $g_{i,j}^{-1}(\boldsymbol{z})$.

As an example, for the ZigZag sampler, where velocities are plus or minus 1 in each component, we could choose the function $g_{i,j}(\boldsymbol{z})$ that sets the velocity associated with the variable that is removed from the model to 0 and keeps other components unchanged. The reverse transition, defined by $Q_{i,j}$, would then need to sample the velocity component associated with the added velocity component from $\{-1, 1\}$, and leave other components unchanged. If we have a sampler where the velocity is constrained to lie in the sphere, that is $\boldsymbol{v} \cdot \boldsymbol{v} = 1$ we cannot just set a component $\boldsymbol{v}$ to 0, as the resulting velocity will no longer lie on the sphere. Instead, $g_{i,j}$ could set the appropriate component of $\boldsymbol{v}$ to 0 and then re-normalise the velocity. The reverse transition could propose a value of the velocity for the added component, from $[-1, 1]$, and then re-scale the other velocity components.

3.1 Invariant distribution

Let $\nu = \sum_k \nu_k$ be a measure on $\cup_k E_k$, where ν_k is the invariant distribution of the base PDMP restricted to E_k. By construction, the $\boldsymbol{\theta}$-marginal distribution of ν is π and this section provides sufficient conditions on $Q_{i,j}, p_{i,j}$ and $\beta_{i,j}$ for ν to be an invariant distribution of the process.

To get a directly usable conditions on $\beta_{i,j}, p_{i,j}$ and $Q_{i,j}$ that ensure the target measure is invariant, some additional notation must be introduced. When jumping from $\boldsymbol{z} = (\boldsymbol{\theta}, \boldsymbol{v}) \in E_j$ to $\boldsymbol{z}' = (\boldsymbol{\theta}, \boldsymbol{v}') \in E_i$, the dimension of the velocity vector needs to be increased by one, but the position is unchanged. (Strictly speaking, the position also increases in dimension by one, but the additional position variable is set to 0.) Hence $g_{i,j}^{-1}(\boldsymbol{z})$ is a one-dimensional

manifold, which can be related to a subset, U say, of $\mathbb{R}$ by introducing a function

$$G_{i,j} : U \times E_j \to E_i$$

such that for any $\alpha \in U$, we have $G_{i,j}(\alpha, \boldsymbol{z}) \in g_{i,j}^{-1}(\{\boldsymbol{z}\})$ and for a fixed $\boldsymbol{z} \in E_j$, $\alpha \mapsto G_{i,j}(\alpha, \boldsymbol{z})$ is a one to one mapping from U to $g_{i,j}^{-1}(\boldsymbol{z})$ (similar ideas are seen in reversible jump MCMC; see Green 1995). For a given $\boldsymbol{z} \in E_j$, it is natural to rewrite the jump kernel $Q_{i,j}$ in terms of $\alpha \in U$, and henceforth we abuse notation and write $Q_{i,j}(\cdot|\boldsymbol{z})$ as a density on U. That is we can simulate the transition from E_j to E_i, by simulating $\alpha \sim Q_{i,j}(\cdot|\boldsymbol{z})$ and then setting $\boldsymbol{z}' = G_{i,j}(\alpha, \boldsymbol{z})$.

So for the ZigZag sampler, where $g_{i,j}$ sets a component of the velocity to 0, $g_{i,j}^{-1}(\boldsymbol{z})$, will be equal to the set of two states that have the same position as $\boldsymbol{z}$ and whose velocities are identical to that of $\boldsymbol{z}$ except for the component associated with the added variable. The two velocities in this set would correspond to replacing the velocity of 0 for that component with either 1 or -1. In this case U would just be the value of the velocity associated with the added component. We can simulate from the reverse transition by simulating U and calculating the new velocity from the old state $\boldsymbol{z}$ and U – this mapping defines $G_{i,j}$.

Finally, define $\boldsymbol{n}_{i,j}$ to be a normal to the boundary $\Gamma_{i,j}$ and let

$$\nu_{i,j}(\boldsymbol{z}) = \nu(\boldsymbol{z})|\langle \boldsymbol{v}, \boldsymbol{n}_{i,j} \rangle| \tag{2}$$

be an unnormalised density on $\Gamma_{i,j}$ and let $\bar{\nu}_{i,j}$ be the pushforward measure on E_j of $\nu_{i,j}$ by $g_{i,j}$:

$$\bar{\nu}_{i,j}(B) = \int_{g_{i,j}^{-1}(B)} \nu_{i,j}(\boldsymbol{z}) \mathrm{d}\boldsymbol{z} \qquad \text{for any } B \subset E_j \text{ measurable.} \tag{3}$$

Informally this is the measure under $\nu_{i,j}$ associated with values of $\boldsymbol{\theta} \in \Gamma_{i,j}$ that would be mapped by $g_{i,j}$ to the set B.

For our example of the ZigZag sampler, $|\langle \boldsymbol{v}, \boldsymbol{n}_{i,j} \rangle| = 1$ as velocities are defined so that they are ± 1 in all directions normal to the boundaries. Thus $\nu_{i,j}(\boldsymbol{z})$ is just the unnormalised density of $\boldsymbol{z}$ under $\nu(\boldsymbol{z})$ restricted to $\boldsymbol{z} \in \Gamma_{i,j}$. The push-forward measure $\tilde{\nu}_{i,j}$ for a set B in the space of the smaller model is just the measure under $\nu_{i,j}$ of all the values of the state $\boldsymbol{z}$ on the boundary $\Gamma_{i,j}$ of the larger model that would be mapped to B when we moved model. That is, for a state in B we consider the pair of states that have the same position and whose velocity is the same except that the velocity of 0 at the removed component is replaced with 1 or -1, and calculate the measure of all such pairs of states corresponding to states in B.

We now present our conditions on $\beta_{i,j}$, $Q_{i,j}$ and $p_{i,j}$. It is helpful to consider separately the cases where the space of velocities is continuous and the case where it is discrete.

Theorem 1. *Assume the space of velocities is continuous, that for each k, the base PDMP on E_k has ν_k as its invariant distribution, and define $\lambda(\boldsymbol{z})$ for $\boldsymbol{z} \in E_k$ to be the jump rate of the base PDMP on E_k. Further assume:*

1. *the mean jump rate of the base PDMPs is finite, i.e. $\int \lambda(\boldsymbol{z}) d\nu < \infty$;*
2. *for all k, π_k is in C^1, i.e. continuously differentiable, on $\mathcal{X}_k$;*
3. *for any k, and any $\boldsymbol{v} \in \mathcal{V}^k$, $\int_{\mathcal{X}_k} |\nabla \pi_k(\boldsymbol{\theta}) \cdot \boldsymbol{v}| d\boldsymbol{\theta} < \infty$.*

Then the measure ν is invariant if for every $(i,j) \in \mathcal{T}$, the following conditions hold

$$\beta_{i,j}(\boldsymbol{z}) = p_{i,j} \frac{\bar{\nu}_{i,j}(\boldsymbol{z})}{\nu(\boldsymbol{z})} \qquad \text{for } \boldsymbol{z} \in E_j \tag{4}$$

$$Q_{i,j}(\alpha|\boldsymbol{z}) = \frac{\nu_{i,j}(G_{i,j}(\alpha, \boldsymbol{z}))|J_{G_{i,j}}(\alpha, \boldsymbol{z})|}{\bar{\nu}_{i,j}(\boldsymbol{z})} \qquad \text{for } \boldsymbol{z} \in E_j \text{ and } \alpha \in U, \tag{5}$$

where $J_{G_{i,j}}$ denotes the Jacobian associated with the transformation $G_{i,j}$.

Proof. See Supplementary Material. □

Intuitively, this result can be understood as a detailed balance condition: we balance the probability flow for each jump $\boldsymbol{z} \to \boldsymbol{z}'$ from E_i to E_j, with that of the reverse jump $\boldsymbol{z}' \to \boldsymbol{z}$.

For discrete velocity spaces the result is slightly simpler, as the Jacobian term is not required. In this case we replace (5) with the condition

$$Q_{i,j}(\alpha|\boldsymbol{z}) = \frac{\nu_{i,j}(G_{i,j}(\alpha, \boldsymbol{z}))}{\bar{\nu}_{i,j}(\boldsymbol{z})} \qquad \text{for } \boldsymbol{z} \in E_j \text{ and } \alpha \in U. \tag{6}$$

3.2 A Reversible Jump PDMP Algorithm

Pseudo-code outlining how we can simulate the resulting Reversible Jump PDMP is given in Algorithm 2. In Lines 2 to 12, we loop over possible events – which can either be events where we jump to a model with an additional active variable, a component of the PDMP position hitting zero, or events within the base PDMP – and simulate the times for each of these possible events. In Lines 13 to 17 we calculate when the first of these possible events occurs and update the process to that time. In Lines 18 to 32 we then update the PDMP state based on which type of event has occurred.

In this algorithm, the additional computational cost of the reversible jump moves, over and above the cost of simulating the base PDMP sampler, is proportional to the number of variables. For each active variable we have to calculate the next time its position hits 0, which involves solving a scalar linear equation. For each inactive variable we need to simulate the time at which we next introduce the variable. As we show in the next section, in most cases the rates for these event are simple, often constant, and importantly do not depend on the likelihood. In practice more efficient implementations than Algorithm 2 will be possible that re-use previously simulated times of events. For example, if an event does

not change the velocity associated with a given variable, then we do not need to recalculate the time that variable hits 0.

4 Example Samplers

4.1 ZigZag Sampler

We first derive the jump rates and transitions for the ZigZag sampler described in Section 2.2.

We choose $g_{i,j}$ to be the orthogonal projection, that is the projection that sets the velocity of the disabled variable to 0. Let $(i,j) \in \mathcal{T}$ be a transition. For any $\boldsymbol{v} \in \mathcal{V}^i$, we have $|\langle \boldsymbol{v}, \boldsymbol{n}_{i,j} \rangle| = 1$, thus from our definition of $\nu_{i,j}(\boldsymbol{z})$ in (2)

$$\nu_{i,j}(\boldsymbol{z}) = \nu(\boldsymbol{z}) = \pi_i(\boldsymbol{\theta}) 2^{-|\gamma_i|}.$$

For $\boldsymbol{z} \in E_j$, since a velocity component in $\{-1, 1\}$ is projected to 0, then from (3)

$$\bar{\nu}_{i,j}(\boldsymbol{z}) = 2\pi_i(\boldsymbol{\theta}) 2^{-|\gamma_i|} = \pi_i(\boldsymbol{\theta}) 2^{-|\gamma_j|}.$$

Since $g_{i,j}$ is the projection that sets to 0 the disabled variable, $g_{i,j}^{-1}(\boldsymbol{\theta}, \boldsymbol{v}) = \{(\boldsymbol{\theta}, \boldsymbol{v} + \boldsymbol{n}_{i,j}), (\boldsymbol{\theta}, \boldsymbol{v} - \boldsymbol{n}_{i,j})\}$ and denoting the new velocity of the added component by $\alpha \in \{-1, 1\}$, we chose

$$G_{i,j}(\alpha, \boldsymbol{\theta}, \boldsymbol{v}) = \alpha \boldsymbol{n}_{i,j} + \boldsymbol{v}.$$

Algorithm 2: Reversible-Jump PDMP algorithm

Inputs: the initial position, $\boldsymbol{\theta}$, velocity, $\boldsymbol{v}$, and model i; and t_{max}
while $t < t_{max}$*:* **do**
  for *each* j *such that* $(j, i) \in \mathcal{T}$ **do**
    Compute τ_j^{RJ} the event time associated to the Poisson rate $\beta_{i,j}(\boldsymbol{\theta} + t\boldsymbol{v}, \boldsymbol{v})$
  end
  Set $\tau^{RJ} = \min \tau_j^{RJ}$, $j = \text{argmin } \tau_j^{RJ}$
  for *each* k *such that* $(i, k) \in \mathcal{T}$ **do**
    Compute $\tau_k^0 = -\frac{\boldsymbol{\theta}_k}{\boldsymbol{v}_k}$ the intersection of the trajectory with the hyperplane $\boldsymbol{\theta}_k = 0$
    if $t_k^0 < 0$ **then**
      Set $\tau_k^0 = +\infty$
    end
  end
  Set $\tau^0 = \min \tau_j^0$, $k = \text{argmin } \tau_k^0$
  Compute τ^{BASE} the event time associated to the Poisson rate of the base PDMP.
  Set $\tau^{evt} = \min(\tau^{RJ}, \tau^0, \tau^{BASE})$
  Set $\boldsymbol{\theta} = \boldsymbol{\theta} + \tau^{evt}\boldsymbol{v}$
  Set $t = t + \tau^{evt}$
  if $\tau^{evt} = \tau^{BASE}$ **then**
    Sample $(\boldsymbol{\theta}, \boldsymbol{v}) \sim Q_{BASE}(\cdot|(\boldsymbol{\theta}, \boldsymbol{v}))$
  end
  if $\tau^{evt} = \tau^0$ **then**
    Simulate U, from a uniform on $[0, 1]$
    if $U < p_{i,k}$ **then**
      Set $\boldsymbol{v} = g_{i,j}(\boldsymbol{v})$
      Set $i = k$
    end
  end
  if $\tau^{evt} = \tau^{RJ}$ **then**
    Sample $\alpha \sim Q_{i,j}(\cdot|(\boldsymbol{\theta}, \boldsymbol{v}))$
    Set $\boldsymbol{v} = G_{i,j}(\alpha, \boldsymbol{\theta}, \boldsymbol{v})$
    Set $i = j$
  end
end

Furthermore, we have from the version of Theorem 1 for discrete velocity spaces that

$$\beta_{i,j}(\boldsymbol{z}) = p_{i,j}\frac{\pi_i(\boldsymbol{\theta})}{\pi_j(\boldsymbol{\theta})} \tag{7}$$

$$Q_{i,j}(\alpha|\boldsymbol{z}) = 1/2 \quad \text{for } \alpha \in \{-1, 1\}. \tag{8}$$

For our variable selection problem, the ratio of the posterior density that appears in $\beta_{i,j}$ will simplify to the ratio of the priors as the likelihood terms are common and cancel. If we have independent priors on the parameters for each variable this term will be a constant, which simplifies the simulation of the events at which we add new variables into our model. This comment applies also to the rates for the Bouncy Particle Sampler which we derive next. See the Supplementary Material for pseudocode for simulating from this process.

4.2 Bouncy Particle Sampler

We consider two versions of the Bouncy Particle Sampler. The first version has velocities on the unit sphere, so

$$\mathcal{V}^i = \{\boldsymbol{v} \in \mathbb{R}^{|\gamma^k|} \text{ such that } \|\boldsymbol{v}\| = 1\}.$$

Like the ZigZag sampler, the deterministic dynamics are given by a constant velocity model. The event rate for sampling from a density $\pi_k(\boldsymbol{\theta})$ is

$$\lambda(\boldsymbol{z}) = \max\{0, -\boldsymbol{v} \cdot \nabla_{\boldsymbol{\theta}} \log \pi_k(\boldsymbol{\theta})\},$$

with the velocity reflecting in the normal to $\log \pi_k(\boldsymbol{\theta})$ at an event: if $\boldsymbol{n}_k(x)$ is the normal to $\log \pi_k(\boldsymbol{\theta})$, then the new velocity is

$$\boldsymbol{v}' = \boldsymbol{v} - 2\left(\frac{\boldsymbol{v} \cdot \boldsymbol{n}_k(x)}{\boldsymbol{n}_k(x) \cdot \boldsymbol{n}_k(x)}\right)\boldsymbol{n}_k(x).$$

The Bouncy Particle Sampler also often has refresh events, at which a completely new velocity is sampled.

Extending the Bouncy Particle Sampler to the variable selection problem requires a more careful analysis than for the ZigZag sampler due to the geometry of the velocity space. Since velocities lie in the unit sphere, we choose $g_{i,j}$ to be the orthogonal projection followed by a rescaling. Hence, $g_{i,j}^{-1}(\boldsymbol{\theta}, \boldsymbol{v}) = \{(\boldsymbol{\theta}, \alpha \boldsymbol{n}_{i,j} + \sqrt{1-\alpha^2}\boldsymbol{v}) | \alpha \in [-1,1]\}$ and denoting the new velocity of the added component by $\alpha \in [-1,1]$, we chose

$$G_{i,j}(\alpha, \boldsymbol{\theta}, \boldsymbol{v}) = \alpha \boldsymbol{n}_{i,j} + \sqrt{1-\alpha^2}\boldsymbol{v}.$$

The following proposition states how to choose the jump rate, $\beta_{i,j}$ and the density for α.

Proposition 1. *For the Bouncy Particle Sampler with velocities on the unit sphere, (4) and (5) are satisfied if, for all $(i,j) \in \mathcal{T}$ with $|\gamma^j| > 0$:*

$$\beta_{i,j}(\boldsymbol{z}) = p_{i,j} \frac{\pi_i(\boldsymbol{\theta})}{\pi_j(\boldsymbol{\theta})} \frac{2A_{sphere}(|\gamma^j|)}{A_{sphere}(|\gamma^i|)} \frac{1}{|\gamma^j|}$$
$$Q_{i,j}(\alpha|\boldsymbol{z}) = \frac{|\alpha||\gamma^j|\sqrt{1-\alpha^2}^{|\gamma^j|-2}}{2} \text{ for } \boldsymbol{z} \in E_j \text{ and } \alpha \in (-1,1)$$

with $A_{sphere}(|\gamma^i|) = \frac{\Gamma(\frac{|\gamma^i|}{2})}{2\pi^{\frac{|\gamma^i|}{2}}}$ the area of the unit sphere of $\mathbb{R}^{|\gamma^i|}$; and if $|\gamma^j| = 0$, where the Bouncy Particle Sampler and ZigZag are equivalent, we use (7) and (8).

The second version of the Bouncy Particle Sampler has velocities in $\mathbb{R}^{|\gamma^k|}$, with their density being standard Gaussian and independent of $\boldsymbol{\theta}$. The dynamics are as previously. The geometry of this case is simpler and we choose $g_{i,j}$ to be the orthogonal projection of the velocity. Hence $g_{i,j}^{-1}(\boldsymbol{\theta}, \boldsymbol{v}) = \{(\boldsymbol{\theta}, \boldsymbol{v} + \alpha \boldsymbol{n}_{i,j}) | \alpha \in \mathbb{R}\}$ and we chose

$$G_{i,j}(\alpha, \boldsymbol{\theta}, \boldsymbol{v}) = \alpha \boldsymbol{n}_{i,j} + \boldsymbol{v}.$$

Proposition 2. *For the Bouncy Particle Sampler with Gaussian velocities, (4) and (5) are satisfied if, for all* $(i, j) \in \mathcal{T}$*:*

$$\beta_{i,j}(\boldsymbol{z}) = p_{i,j}\frac{\pi_i(\boldsymbol{\theta})}{\pi_j(\boldsymbol{\theta})}\frac{2}{\sqrt{2\pi}}$$
$$Q_{i,j}(\alpha|\boldsymbol{z}) = 2|\alpha|e^{-\frac{1}{2}\alpha^2} \text{ for } \boldsymbol{z} \in E_j \text{ and } \alpha \in \mathbb{R}.$$

5 Simulation Study

In this section we demonstrate the potential advantage of our new samplers compared to alternative approaches for Bayesian variable selection. To compare between different samplers we consider the Monte Carlo estimates of the posterior probabilities of inclusion, the posterior means for the regression coefficients, and the posterior means conditioned on the model. For a given sampler the statistical efficiency is measured by the mean squared error of the sampler, denoted by $\sigma^2_{\text{sampler}}$, and is calculated using R runs of the sampler as

$$\sigma^2_{\text{sampler}} = \frac{1}{R}\sum_{i=1}^{R}(\hat{q}_r - q)^2 \tag{9}$$

where $\hat{q}_r$ for $r = 1, ..., R$ are the estimates of a quantity of interest from the R runs, and q is either the exact posterior quantity of interest, if available, or it is the estimate from an independent long run of an MCMC method. In multiple dimensions the statistical efficiency is measured as the median $\sigma^2_{\text{sampler}}$ over all dimensions. To compare the performance of different samplers we also consider a measure of efficiency relative to a reference sampler. If we denote the reference sampler by ref, then we define Relative Statistical Efficiency

(RSE) and Relative Efficiency (RE) by

$$\text{RSE} = \frac{\sigma^2_{\text{ref}}\, n_{\text{ref}}}{\sigma^2_{\text{sampler}}\, n_{\text{sampler}}}, \qquad \text{RE} = \frac{\sigma^2_{\text{ref}}\, t_{\text{ref}}}{\sigma^2_{\text{sampler}}\, t_{\text{sampler}}},$$

where n_{sampler} and n_{ref} are the number of iterations of the algorithms and t_{sampler} and t_{ref} are the computation times of the algorithms. The RSE measures the relative efficiency of the algorithms per iteration whereas the RE measures the efficiency per second. For interpretation, an RSE or RE value of 2 implies that the sampler is 2 times *more efficient* than the reference method. The sensitivity of the methods to the choice of reversible jump parameters $p_{i,j}$ and regular PDMP tuning parameters is explored empirically in the Supplementary Material. Based on these results we fixed $p_{i,j} = 0.6$ for all i and j in all reversible jump PDMP samplers and implemented BPS with normal velocity distribution with a fixed refreshment rate of 0.1 that was constant across models. We set this rate to be 0.1 based on results from some initial tuning runs.

5.1 Logistic regression

First we compare PDMP based samplers with existing MCMC methods for a logistic regression model with spike and slab priors on the regression coefficients. The MCMC competitors are a collapsed Gibbs sampler, and two reversible jump samplers. The Gibbs sampler uses the Polya-Gamma augmentation of Polson et al. (2013) to make efficient moves through model space. We implemented one reversible jump MCMC sampler using the NIMBLE software package (de Valpine et al. 2017) using an independent normal proposal for selected variables. The other is a reversible-jump version of HMC, which, at each iteration, with probability 1/2 uses an HMC move within a model, and otherwise uses a reversible jump move between models. More details on the samplers are given in the Supplementary Material.

The logistic regression model has a d-dimensional regression parameter $\boldsymbol{\theta} \in \mathbb{R}^d$, and a binary response $y_i \in \{0, 1\}$ which is distributed as

$$P(y_i = 1|\boldsymbol{x}_i, \boldsymbol{\theta}) = \frac{\exp(\sum_{j=1}^{d} x_{i,j}\theta_j)}{1 + \exp(\sum_{j=1}^{d} x_{i,j}\theta_j)}$$

where $\boldsymbol{x}_i$ is the d-vector of covariates for observation i. In our simulation study, each vector $\boldsymbol{x}_i$ is simulated from a multivariate normal with mean zero and $d \times d$ covariance matrix $\boldsymbol{\Sigma}$. We use a prior that is independent for each θ_j and where $\theta_j \sim \frac{10}{d}\mathcal{N}(0, 10) + (1 - \frac{10}{d})\delta_0$, which corresponds to a prior that favors models with 10 selected variables. Data was generated using this model and the following choices for $\boldsymbol{\theta}$ and covariance matrix $\boldsymbol{\Sigma}$:

1. A pair of correlated covariates, one of which is in the model: $\boldsymbol{\theta} = (1, 0, ...0)^T$ with $\boldsymbol{\Sigma}_{2,1} = \boldsymbol{\Sigma}_{2,1} = 0.9$, $\boldsymbol{\Sigma}_{i,i} = 1$, and $\boldsymbol{\Sigma}_{i,j} = 0$ otherwise.

2. Structured correlation between all covariates with six active covariates:
   $\boldsymbol{\theta} = (3, 3, -2, 3, 3, -2, 0, ..., 0)^T$ with $\boldsymbol{\Sigma}_{i,j} = \exp(-|i - j|)$.

3. No correlation between covariates and six active covariates:
   $\boldsymbol{\theta} = (3, 3, -2, 3, 3, -2, 0, ..., 0)^T$, with $\boldsymbol{\Sigma}_{i,i} = 1$ and $\boldsymbol{\Sigma}_{i,j} = 0$ if $i \neq j$.

4. Multiple pairs of correlated variables with six active covariates:
   $\boldsymbol{\theta} = (3, 3, -2, 3, 3, -2, 0, ..., 0)^T$ with $\boldsymbol{\Sigma}_{i+d/2,i} = \boldsymbol{\Sigma}_{i,i+d/2} = 0.9$ for $1 \leq i \leq 6$, $\boldsymbol{\Sigma}_{i,i} = 1$ and $\boldsymbol{\Sigma}_{i,j} = 0$ otherwise.

These simulation scenarios are analogous to others previously considered in the literature for linear regression. Scenarios 1 and 3 are are similar to those considered by Wang et al. (2011) and Zanella & Roberts (2019) while Scenario 2 is similar to one considered by Yang et al. (2016). Scenario 4 is an extension of Scenario 1 to allow for more correlated

variables (and results for a further extension that allows for higher correlation is shown in the Supplementary Material). We present results for both ZigZag and the Bouncy Particle Sampler with Gaussian distributed velocities in Tables 1 to 4 (very similar results are obtained using the Bouncy Particle Sampler with velocities uniformly on the unit sphere).

Across the four scenarios, Gibbs variable selection performs the best in low sample sizes ($n < d$). PDMP methods are competitive with Gibbs sampling when the number of predictors and sample size is comparable and offer substantial efficiency gains for larger sample sizes. Smaller gains can also be seen when the dimension is increased with fixed sample size. Both BPS and ZigZag methods offer similar relative efficiencies across the experiments. The greatest efficiency gains for our PDMP methods are seen in Table 1 with Tables 2, 3 and 4 offering lower gains in performance. This may be due to smaller models being more likely in Scenario 1, as these require less computational effort for the PDMP methods since fewer gradient calculations are required to simulate within these lower dimensional models. RJ-HMC offers improvement to the simpler independent normal reversible jump sampler for larger sample sizes. While HMC offers better within model moves it comes at a higher computational cost and no real advantage to moving through model space.

Unlike the reversible jump approaches, the Gibbs sampler makes efficient moves through model space. This sampler makes use of marginalisation over the parameter values $\boldsymbol{\theta}$ to more effectively jump between models. However, when the number of observations is large there is increased computational requirement for sampling the Polya-Gamma random variables. This results in less efficient overall sampling than the PDMP samplers when n is large.

The use of subsampling methods for Bayesian variable selection problems is a recently emerging area (Song et al. 2020, Buchholz et al. 2019). One of the attractions of PDMP samplers is that they can be implemented in a way where they only access a small subset

Table 1: Scenario 1 (pair of correlated variables): Relative efficiencies for methods, against a Reversible Jump algorithm, for the marginal posterior means (Mean) and marginal posterior probabilities of inclusion (PI). Bold figures show the best performing sampler.

	ZigZag		BPS		Gibbs		RJ-HMC	
n, d	PI	Mean	PI	Mean	PI	Mean	PI	Mean
100, 100	0.86	**4.77**	0.44	4.49	**3.82**	4.21	0.04	0.62
200, 100	1.58	**33.02**	0.97	23.94	**8.04**	8.65	1.28	1.89
400, 100	2.54	**47.22**	1.56	36.33	**10.13**	9.75	1.50	1.95
800, 100	4.50	**35.82**	3.36	29.96	**11.87**	10.45	1.59	1.58
100, 200	1.29	1.35	1.45	1.79	**5.11**	**4.72**	1.34	2.09
200, 200	5.21	**46.26**	2.71	30.86	**12.14**	12.75	2.10	2.35
400, 200	10.36	**83.95**	6.21	53.86	**15.64**	15.40	2.23	2.65
800, 200	19.03	**150.42**	14.87	86.04	**19.48**	19.00	2.49	3.00
100, 400	1.10	1.49	1.71	2.36	**3.59**	**3.72**	0.50	1.60
200, 400	16.67	**82.93**	11.91	50.70	**17.57**	17.19	3.24	4.01
400, 400	**48.39**	**141.70**	30.76	97.78	22.50	23.03	3.93	5.12
800, 400	**97.84**	**203.96**	51.61	122.92	22.36	22.26	4.12	5.35

of data at each iteration, whilst still targeting the true posterior. We now investigate how these ideas work in the variable selection problem, by comparing the efficiency of three implementations of ZigZag with that of the Gibbs sampler, and see how this depends on the number of observations. These are ZigZag using the full data, ZigZag using subsampling with a global bound, and ZigZag with subsampling control variates (see Bierkens et al. 2019, for details of both subsampling approaches).

Standard application of control variates requires calculation of the gradient at a reference point using the full likelihood. Due to the trans-dimensional nature of variable selection problems, a full gradient calculation is not well defined. For this reason we choose to make use of control variates defined for a fixed model $\mathcal{M}$ where the gradient is well defined. These control variates are only used when the sampler is in this model. For certain problems, such as generalised linear models, an $O(n)$ initial calculation relating to the like-

Table 2: Scenario 2 (General correlation): Relative efficiency for methods, against a Reversible Jump algorithm, for the marginal posterior means (Mean) and marginal posterior probabilities of inclusion (PI). Bold figures show the best performing sampler.

	ZigZag		BPS		Gibbs		RJ-HMC	
n, d	PI	Mean	PI	Mean	PI	Mean	PI	Mean
100, 100	0.34	0.69	0.27	**1.22**	**1.13**	0.87	0.00	0.11
200, 100	0.85	1.12	0.57	**1.98**	**2.07**	1.34	0.71	0.29
400, 100	1.63	2.19	1.15	**2.96**	**2.66**	2.07	1.32	1.23
800, 100	2.84	5.46	1.87	**5.71**	**2.98**	2.47	1.40	1.93
100, 200	0.65	1.15	0.89	1.85	**1.65**	**2.22**	0.02	1.58
200, 200	1.46	1.57	1.56	**2.44**	**2.29**	1.27	0.74	0.42
400, 200	**4.28**	5.11	3.61	**5.70**	4.03	2.81	2.12	2.56
800, 200	**7.85**	**11.10**	5.08	10.06	4.98	4.19	2.35	3.13
100, 400	1.34	1.77	1.98	**2.83**	**2.79**	2.28	0.10	2.30
200, 400	3.86	3.84	**4.69**	**6.26**	4.64	3.07	0.20	1.18
400, 400	**14.61**	22.55	13.00	**27.90**	8.35	7.50	3.64	5.71
800, 400	**25.26**	**37.37**	19.25	33.03	9.71	8.80	4.20	5.88

lihood can be reused to define control variates for multiple models with certain parameters set to zero. See the supplementary material for more details.

Results are shown in Figure 2. In this simulation we take the independent prior $\theta_j \sim \frac{1}{15}\mathcal{N}(0, 10) + (1 - \frac{1}{15})\delta_0$, favoring models with a single selected variable. Despite our simplistic implementation, these results indicate that Zig-Zag with control variates is becoming increasingly efficient relative to Zig-Zag using the full dataset as the number of samples increases. Furthermore we see evidence of super-efficiency – whilst the computational cost per ESS of the Gibbs sampler is expected to be linear in the number of observations, the relative efficiency plots suggest that this is sub-linear for ZigZag with control variates.

Table 3: Scenario 3 (No correlation): Relative efficiency for methods, against a Reversible Jump algorithm, for the marginal posterior means (Mean) and marginal posterior probabilities of inclusion (PI). Bold figures show the best performing sampler.

	ZigZag		BPS		Gibbs		RJ-HMC	
n, d	PI	Mean	PI	Mean	PI	Mean	PI	Mean
100, 100	0.45	0.97	0.29	**1.70**	**1.21**	1.16	0.01	0.09
200, 100	0.97	1.37	0.59	**2.08**	**2.17**	1.50	0.63	0.24
400, 100	1.47	2.13	1.06	**2.60**	**2.51**	1.80	1.20	0.88
800, 100	2.85	**5.09**	1.95	4.87	**3.25**	2.53	1.45	2.10
100, 200	0.75	0.79	0.91	1.59	**1.84**	**1.65**	0.31	0.22
200, 200	1.90	1.72	1.79	**2.23**	**2.82**	1.85	1.65	0.94
400, 200	**5.62**	11.13	4.33	**14.84**	4.88	3.94	2.08	3.04
800, 200	**9.77**	17.40	7.47	**20.26**	5.84	4.75	2.44	3.57
100, 400	1.05	1.48	1.74	**2.83**	**2.11**	1.80	0.05	1.98
200, 400	2.36	1.96	3.02	**2.63**	**3.65**	2.20	1.89	2.04
400, 400	**7.13**	8.31	6.77	**10.85**	5.32	4.08	2.88	4.21
800, 400	**24.69**	**40.56**	17.92	31.00	9.40	8.78	3.72	5.59

5.2 Robust regression

As mentioned in the introduction, a common approach to Bayesian variable selection is to use continuous spike-and-slab priors for each parameter rather than try to sample from the joint posterior of model and parameters. Such an approach is attractive as it enables standard gradient-based samplers, such as Hamiltonian Monte Carlo, to be used. We now compare such an approach, implemented with the popular Stan software (Carpenter et al. 2017), to our PDMP samplers. Our aim is to both investigate the computational efficiencies of the two approaches and to show the differences in posterior that we obtain from these different types of prior. Our comparison is based on a robust linear regression model.

In particular, we model the errors in our linear regression model as a mixture of normals

Table 4: Scenario 4 (multiple correlated pairs): Relative efficiency for methods, against a Reversible Jump algorithm, for the marginal posterior means (Mean) and marginal posterior probabilities of inclusion (PI). Bold figures show the best performing sampler.

	ZigZag		BPS		Gibbs		RJ-HMC	
n, d	PI	Mean	PI	Mean	PI	Mean	PI	Mean
100, 100	0.43	1.05	0.36	**1.84**	**1.11**	1.04	0.01	0.07
200, 100	0.90	1.20	0.69	1.52	**2.33**	**2.49**	0.36	0.45
400, 100	1.25	1.40	0.86	**1.70**	**2.17**	1.20	0.40	1.34
800, 100	2.15	**2.62**	1.54	2.43	**2.70**	1.42	0.19	0.19
100, 200	0.69	1.30	1.11	**2.34**	**1.48**	1.67	0.03	0.49
200, 200	1.65	1.90	1.74	**2.72**	**2.60**	2.26	0.76	1.74
400, 200	**3.41**	3.99	3.04	**4.96**	3.19	2.17	1.82	2.19
800, 200	**8.56**	13.33	6.25	**14.17**	5.87	5.02	2.29	3.19
100, 400	1.81	2.96	2.50	**6.90**	**2.92**	2.37	0.01	1.30
200, 400	2.71	2.19	3.68	**3.55**	**3.93**	2.51	2.17	2.28
400, 400	**11.89**	15.64	10.31	**17.66**	7.98	6.79	3.33	4.63
800, 400	**19.60**	27.23	15.05	**27.50**	8.58	7.42	3.59	5.18

with different variances. Thus

$$\boldsymbol{Y} = \sum_{i=1}^{d} \boldsymbol{X}_j \theta_j + \epsilon, \qquad \epsilon \sim \frac{1}{2}\mathrm{N}(0,1) + \frac{1}{2}\mathrm{N}(0,10^2).$$

The continuous variable selection prior we consider is the regularised horseshoe (Piironen & Vehtari 2017*a*,*b*)

$$\theta_j \sim N(0, \tau^2\tilde{\lambda}_j), \qquad \tilde{\lambda}_j = \frac{c^2\lambda_j}{c^2 + \tau^2\lambda_j}, \qquad \lambda_j \sim C^+(0,1)$$

for $j = 1, ..., d$ where $C^+(0,1)$ denotes the half-Cauchy distribution for the standard deviation λ_j. The regularised horseshoe is a variation of the horseshoe prior (Carvalho et al. 2010) that offers a continuous approximation of a Dirac spike and slab where the slab is

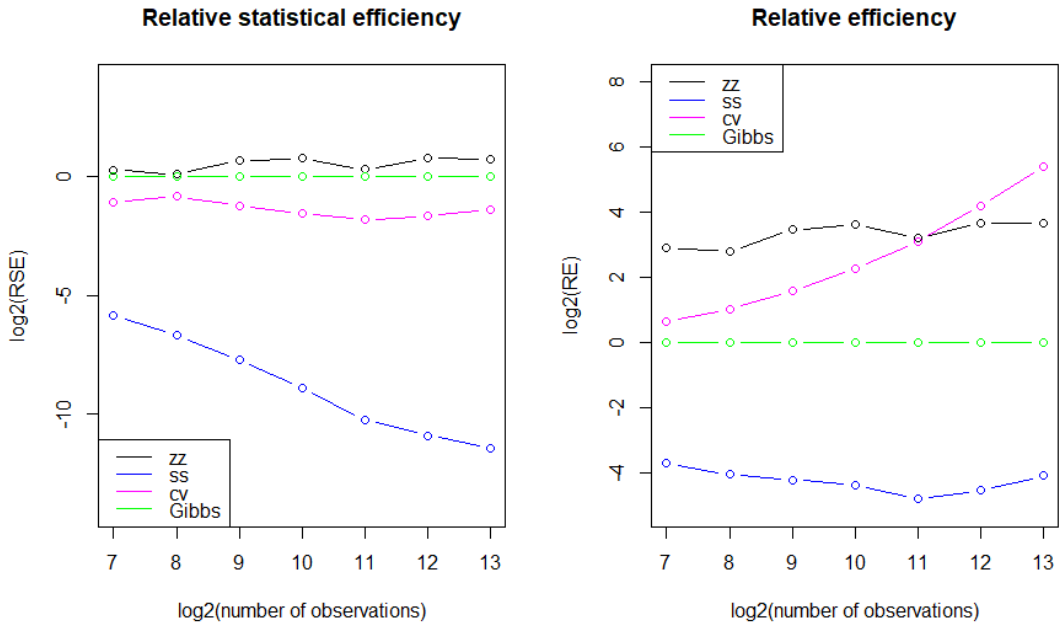

Figure 2: Log-log plots of efficiency, relative to the Gibbs sampler, of different samplers as we vary the number of observations. Plotted are the relative efficiencies for the posterior mean conditional on model $\mathcal{M}^*$ where $\mathcal{M}^*$ corresponds to the true data generated model. The dataset was generated with a 15-dimensional regression parameter $\boldsymbol{\theta} = (1, 1, 0, 0, ..., 0)$. The methods run are the Zig-Zag applied to the full dataset (zz, black), Zig-Zag with subsampling using global bounds (ss, blue), Zig-Zag with control variates (cv, magenta) and Gibbs sampling (Gibbs, green). All methods were initialised at the location of the control variate. Methods were given the same computational budget, for details see the Supplementary Material.

a normal distribution with finite variance c^2. The hyper-parameter τ controls the global shrinkage of the variables towards zero. In Carvalho et al. (2010) it was shown that for standard linear regression the optimal choice for a fixed value of τ is $\tau_0 = \sigma \frac{d_0}{(d-d_0)\sqrt{n}}$ where d_0 is a the number of nonzero variables in the sparse model and σ is the noise variance. In line with this, we compare the regularised horseshoe prior with fixed hyper-parameter

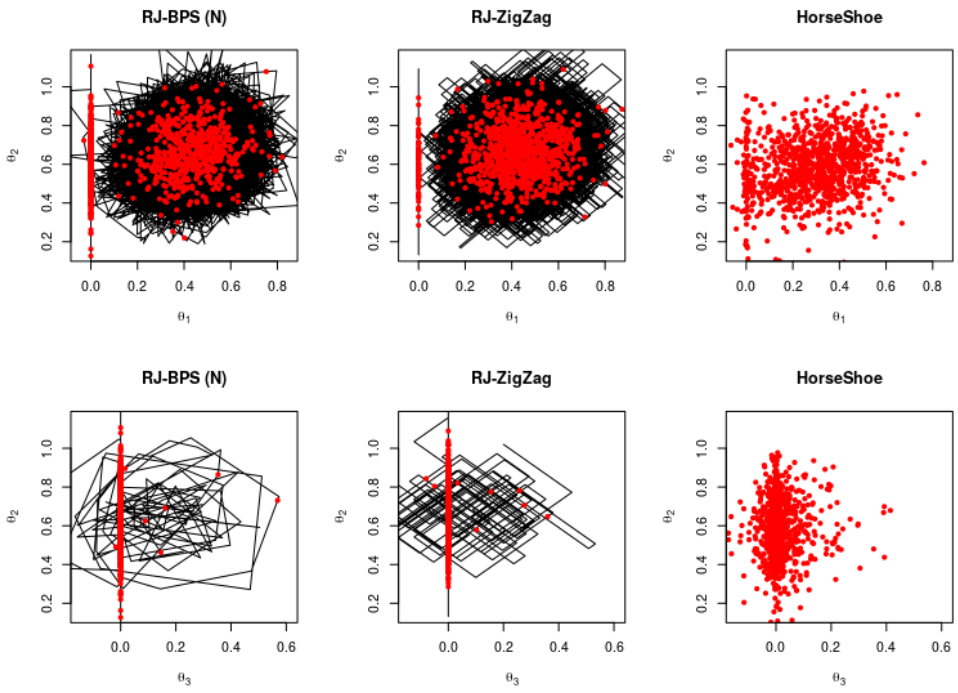

Figure 3: Dynamics of the samplers on a robust regression example with spike and slab or horseshoe prior. The top row shows the posterior for θ_1 and θ_2, bottom row shows the estimates for θ_2 and θ_3. The spike and slab distributions are sampled using the reversible jump PDMP samplers with reversible jump parameter 0.6 and refreshment for the BPS methods set to 0.5. All methods are shown with 10^3 samples (red) and the PDMP dynamics are shown in black. Sampling with the Horseshoe prior was implemented in Stan using NUTS. Both Stan and PDMP methods were run for the same computing time. To aid visualisation only the first 30% of the PDMP trajectories are shown.

$\tau_0 = \frac{d_0}{(d-d_0)\sqrt{n}}$ against the spike-and-slab prior

$$\theta_j \sim \frac{d_0}{d} N(0, c^2) + \left(1 - \frac{d_0}{d}\right) \delta_0$$

for $j = 1, ..., d$. MCMC for the model using a horseshoe prior was performed by Stan's

implementation of NUTS (Hoffman & Gelman 2014).

We first compare the variable selection dynamics for a simple model with $d = 4$ variables, $n = 120$ observations and regression parameter $\boldsymbol{\theta} = (0.5, 0.5, 0, 0)^T$. The covariate values and residuals were generated as independent draws from a standard normal and the prior expected model size is set to $d_0 = 1$. Example output for the PDMP samplers and the Stan implementation is shown in Figure 3. The posteriors show the horseshoe prior replicating the effect of the spike-and-slab through shrinking the coefficients towards zero, but it is not able to give exact zeros.

We now compare the reversible jump PDMP methods in terms of their sampling efficiency for a higher dimensional problem. The dataset is generated for $d = 200$ variables and $n = 100$ observations with regression parameter $\boldsymbol{\theta} = (2, 2, 2, 2, 0, 0, 0....0)^T$. The covariates for each observation were drawn from an AR(1) process with lag-1 correlation of 0.5. The residuals were generated from a standard Cauchy distribution.

We ran Stan with the default settings for a burnin of 1000 iterations and then for 16, 32, 64, ..., 2048 iterations. We ran the reversible jump PDMP samplers for the same wall clock time for both burnin and subsequent iterations. The experiment was repeated 50 times and the resulting boxplots of the posterior mean of θ_1 are shown in Figure 4. For the same computational budget, the reversible jump PDMP methods are able to attain better performance as can be seen by the lower Monte Carlo variability of their estimates. However, it is also apparent from this simulation that HMC is less susceptible to local modes, perhaps due to the horseshoe prior being continuous.

The predictive abilities of the methods are compared in Figure 5. Here an additional $n = 100$ observations were drawn from the same model and these were used as a hold-out dataset to validate the posterior predictive ability. The predictive ability is defined in terms of the Monte Carlo estimate of the mean square prediction performance $\frac{1}{100}\sum_{i=1}^{100}(y_i - \boldsymbol{x}_i^T\boldsymbol{\theta})^2$ where $\boldsymbol{\theta}$ is replaced by the samples generated by either Stan or samples given by a discretisation

of time for the reversible jump PDMP samplers. The reversible jump PDMP samplers all give the same predictive performance for large iteration numbers while Stan, which uses a horseshoe prior, performs slightly worse.

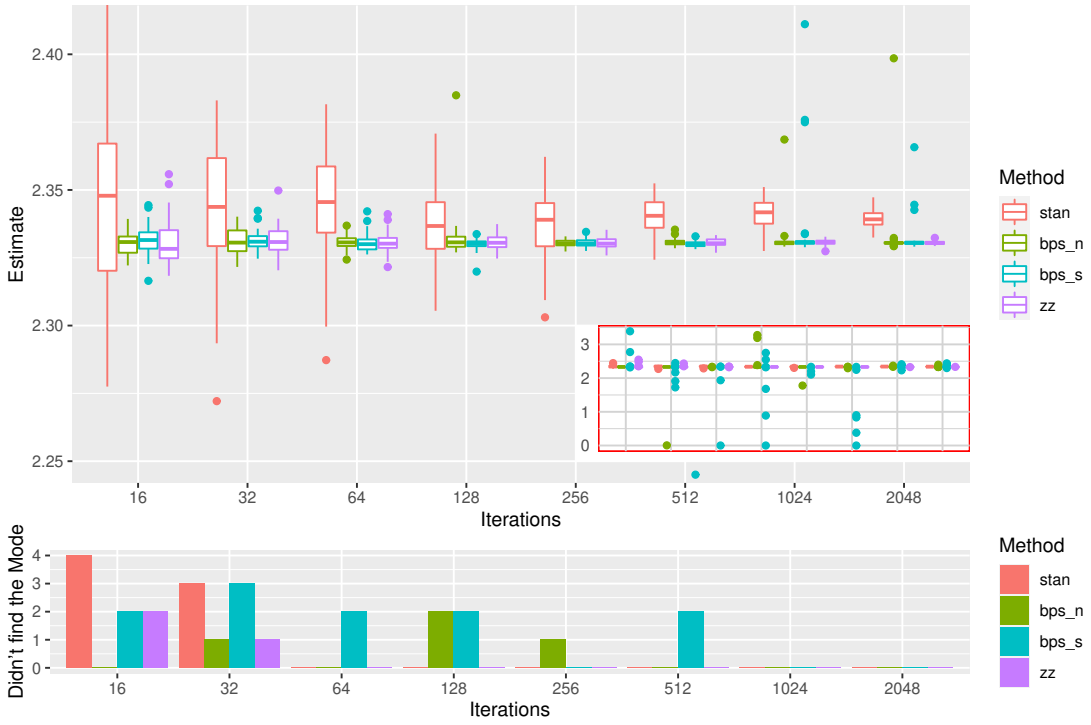

Figure 4: Sampling efficiency for reversible jump PDMP vs Stan for the robust regression example. The PDMP samplers are ZigZag (zz), Bouncy Particle Sample with normally distributed velocities (bps_n) and with velocities distributed uniformly on the sphere (bps_s). The top figure shows boxplots of the posterior mean of θ_1 for increasing computational budget, with outliers from the sampler removed for visualisation purposes. These removed outliers correspond to times that the sampler has become stuck in a local mode where $\theta_1 = 0$. The subplot shows the full results including outliers from the samplers. The Stan sampler is sampling from a different posterior to the PDMP methods, and this is seen in the estimates converging to slightly different values; but Monte Carlo efficiency can be assessed by comparing the variability of the estimates. The bottom figure shows the number of times that the samplers did not find the global mode.

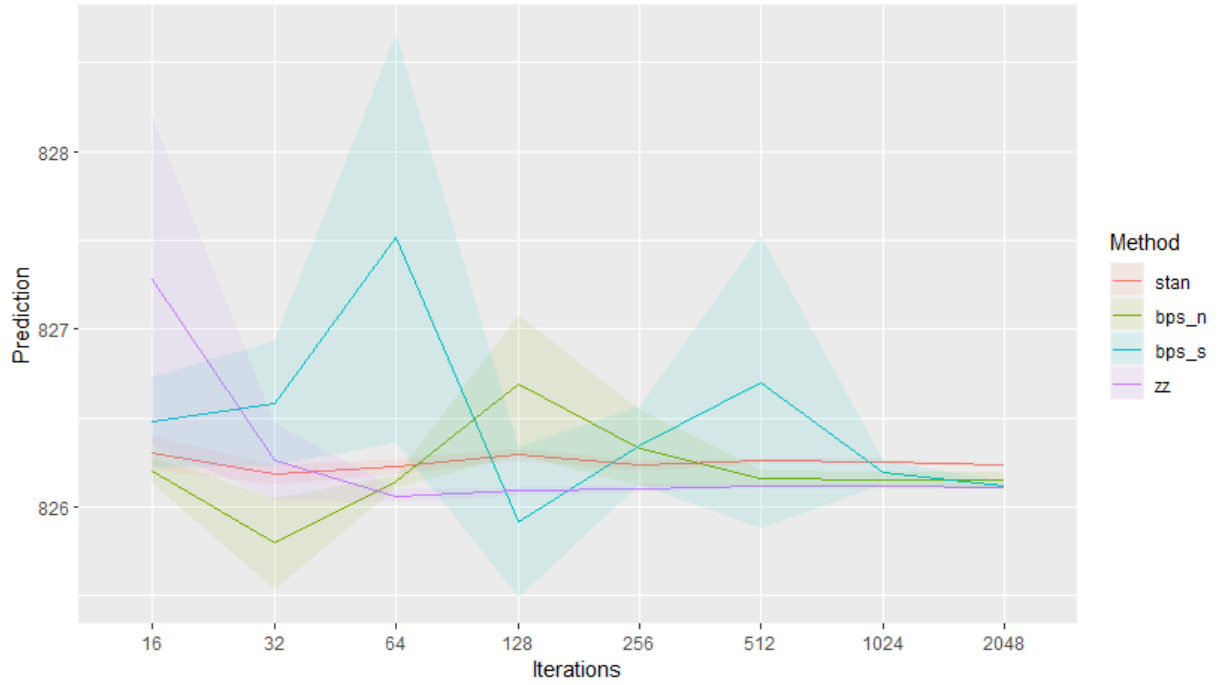

Figure 5: Predictive ability of reversible jump PDMP vs Stan for the robust regression example. The PDMP samplers are ZigZag (zz), Bouncy Particle Sample with normally distributed velocities (bps_n) and with velocities distributed uniformly on the sphere (bps_s). The predictive ability is measured by Monte Carlo estimates of the mean square predictive performance.

6 Discussion

We have shown how PDMP samplers can be extended so that they can sample from the joint posterior over model and parameters in variable selection problems. There are a number of open challenges that stem from this work. As with any MCMC algorithm, the reversible jump PDMP samplers have tuning parameters. The additional tuning parameters are the probabilities of moving between models when parameters hit zero. As a default, we recommend setting these probabilities all to the same value, and our simulation results were

based on choosing this value after empirically evaluating the performance of the samplers on one simple problem. Whilst the samplers mixed well, it is likely that better mixing could be achieved if more informed choices of tuning parameters were made, and theory for guiding such choices is needed (e.g. see Sherlock & Thiery 2020, for theory on choosing the refresh rate of the Bouncy Particle Sampler).

The form of our reversible jump PDMP samplers is based on particular features of the variable selection problem. In other model choice settings, different trans-dimensional moves may be needed. The theory we developed should be able to be adapted to give rules for choosing rates of such moves. For example, in the case of sampling from mixture models one could introduce moves that remove a component when that component's weight hits zero; and when we add a component we simulate new values for the component parameters from the prior. An advantage of such a construction would be that the rate of adding or removing components would not depend on the likelihood. Also our trans-dimensional moves are reversible, that is they balance probability flow from model i to model j by the flow of probability from model j to model i – it would be interesting to see if non-reversible trans-dimensional moves could be constructed.

It is likely that the reversible jump PDMP samplers will still struggle in situations where the posterior is multi-modal with well separated modes. For such cases it would be interesting to try and incorporate ideas such as tempering (Marinari & Parisi 1992) to allow for better mixing across modes. Also better mixing may be possible if we could introduce non-reversibility into exploring models, as in Power & Goldman (2019) and Gagnon & Maire (2020), though it is difficult to see how to incorporate lifting ideas that introduce such non-reversibility to our samplers. Finally, whilst we have considered only models with a finite number of variables, we believe the theory and PDMPs would extend to models with a countable number of variables, such as for polynomial regression. The challenge in using such PDMPs will be efficiently sampling when new variables are introduced into

the model. If the rate at which each variable is added is simple and we can analytically calculate the sum of these rates, then this should be possible by simulating when a variable is added with a rate equal to this sum. At each such event we then simulate which variable to add.

References

Bierkens, J., Bouchard-Côté, A., Doucet, A., Duncan, A. B., Fearnhead, P., Lienart, T., Roberts, G. O. & Vollmer, S. J. (2018), 'Piecewise deterministic Markov processes for scalable Monte Carlo on restricted domains', *Statistics & Probability Letters* **136**, 148–154.

Bierkens, J., Fearnhead, P. & Roberts, G. (2019), 'The zig-zag process and super-efficient sampling for Bayesian analysis of big data', *The Annals of Statistics* **47**(3), 1288–1320.

Bierkens, J., Grazzi, S., Kamatani, K. & Roberts, G. (2020), The boomerang sampler, *in* 'International Conference on Machine Learning', PMLR, pp. 908–918.

Bierkens, J., Grazzi, S., van der Meulen, F. & Schauer, M. (2021), 'Sticky PDMP samplers for sparse and local inference problems'. arXiv:2103.08478.

Bierkens, J., Kamatani, K. & Roberts, G. O. (2022), 'High-dimensional scaling limits of piecewise deterministic sampling algorithms', *Annals of Applied Probability* **To appear**.

Bierkens, J. & Roberts, G. (2017), 'A piecewise deterministic scaling limit of lifted Metropolis–Hastings in the Curie–Weiss model', *The Annals of Applied Probability* **27**(2), 846–882.

Bottolo, L. & Richardson, S. (2010), 'Evolutionary stochastic search for Bayesian model exploration', *Bayesian Analysis* **5**, 583–618.

Bouchard-Côté, A., Vollmer, S. J. & Doucet, A. (2018), 'The bouncy particle sampler: A nonreversible rejection-free Markov chain Monte Carlo method', *Journal of the American Statistical Association* **113**(522), 855–867.

Buchholz, A., Ahfock, D. & Richardson, S. (2019), 'Distributed computation for marginal likelihood based model choice', *arXiv.1910.04672* .

Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P. & Riddell, A. (2017), 'Stan: A probabilistic programming language', *Journal of Statistical Software* **76**(1).

Carvalho, C. M., Polson, N. G. & Scott, J. G. (2010), 'The horseshoe estimator for sparse signals', *Biometrika* **97**(2), 465–480.

Chevallier, A., Fearnhead, P. & Sutton, M. (2020), 'Reversible jump PDMP samplers for variable selection'. arXiv:2010.11771.

Chevallier, A., Power, S., Wang, A. & Fearnhead, P. (2021), 'PDMP Monte Carlo methods for piecewise-smooth densities'. arXiv.2111.05859.

Chipman, H. A., George, E. I. & McCulloch, R. E. (2001), 'The practical implementation of Bayesian model selection (with discussion)', *Model Selection (P. Lahiri ed.)* pp. 65–134.

Davis, M. (1993), *Markov Models and Optimization*, Chapman Hall.

de Valpine, P., Turek, D., Paciorek, C., Anderson-Bergman, C., Temple Lang, D. & Bodik, R. (2017), 'Programming with models: writing statistical algorithms for general model structures with NIMBLE', *Journal of Computational and Graphical Statistics* **26**, 403–413.

Deligiannidis, G., Paulin, D., Bouchard-Côté, A. & Doucet, A. (2018), 'Randomized Hamiltonian Monte Carlo as scaling limit of the bouncy particle sampler and dimension-free convergence rates'. arXiv.1808.04299.

Diaconis, P., Holmes, S. & Neal, R. M. (2000), 'Analysis of a nonreversible Markov chain sampler', *Annals of Applied Probability* **10**, 726–752.

Durmus, A., Guillin, A. & Monmarché, P. (2021), 'Piecewise deterministic Markov processes and their invariant measure', *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques* **57**(3), 1442–1475.

Fearnhead, P., Bierkens, J., Pollock, M. & Roberts, G. O. (2018), 'Piecewise deterministic Markov processes for continuous-time Monte Carlo', *Statistical Science* **33**(3), 386–412.

Gagnon, P. & Maire, F. (2020), 'Lifted samplers for partially ordered discrete state-spaces'. arXiv:2003.05492.

George, E. I. & McCulloch, R. E. (1993), 'Variable selection via Gibbs sampling', *Journal of the American Statistical Association* **88**(423), 881–889.

Geweke, J. (1996), Variable selection and model comparison in regression, *in* 'Bayesian Statistics 5', Oxford University Press, pp. 609–620.

Goldman, J. V., Sell, T. & Singh, S. S. (2021), 'Gradient-based Markov chain Monte Carlo for Bayesian inference with non-differentiable priors', *Journal of the American Statistical Association* pp. 1–12.

Green, P. J. (1995), 'Reversible jump Markov chain Monte Carlo computation and Bayesian model determination', *Biometrika* **82**(4), 711–732.

Grenander, U. & Miller, M. I. (1994), 'Representations of knowledge in complex systems', *Journal of the Royal Statistical Society: Series B (Methodological)* **56**(4), 549–581.

Hoffman, M. D. & Gelman, A. (2014), 'The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo.', *Journal of Machine Learning Research* **15**(1), 1593–1623.

Ishwaran, H. & Rao, J. S. (2005), 'Spike and slab variable selection: frequentist and Bayesian strategies', *The Annals of Statistics* **33**(2), 730–773.

Kuo, L. & Mallick, B. K. (1998), 'Variable selection for regression models', *Sankhya Series B* **60**, 65–81.

Marinari, E. & Parisi, G. (1992), 'Simulated tempering: a new Monte Carlo scheme', *EPL (Europhysics Letters)* **19**(6), 451.

Markovic, J. & Sepehri, A. (2018), 'Bouncy hybrid sampler as a unifying device', *arXiv:1802.04366* .

Michel, M., Durmus, A. & Sénécal, S. (2020), 'Forward event-chain Monte Carlo: Fast sampling by randomness control in irreversible Markov chains', *Journal of Computational and Graphical Statistics* **29**, 689–702.

Michel, M., Kapfer, S. C. & Krauth, W. (2014), 'Generalized event-chain Monte Carlo: Constructing rejection-free global-balance algorithms from infinitesimal steps', *The Journal of Chemical Physics* **140**(5), 054116.

Mitchell, T. J. & Beauchamp, J. J. (1988), 'Bayesian variable selection in linear regression', *Journal of the American Statistical Association* **83**(404), 1023–1032.

Neal, R. M. (2011), MCMC using Hamiltonian dynamics, *in* A. Brooks, A. Gelman, G. L. Jones & X. Meng, eds, 'Handbook of Markov chain Monte Carlo', Chapman & Hall, pp. 113–162.

Peters, E. A. & de With, G. (2012), 'Rejection-free Monte Carlo sampling for general potentials', *Physical Review E* **85**(2), 026703.

Phillips, D. B. & Smith, A. F. (1996), Bayesian model comparison via jump diffusions, *in* W. R. Gilks, S. Richardson & D. Spiegelhalter, eds, 'Markov chain Monte Carlo in practice', Chapman & Hall, CRC, pp. 215–240.

Piironen, J. & Vehtari, A. (2017*a*), On the hyperprior choice for the global shrinkage parameter in the horseshoe prior, Vol. 54 of *Proceedings of Machine Learning Research*, PMLR, pp. 905–913.

Piironen, J. & Vehtari, A. (2017*b*), 'Sparsity information and regularization in the horseshoe and other shrinkage priors', *Electron. J. Statist.* **11**(2), 5018–5051.

Polson, N. G., Scott, J. G. & Windle, J. (2013), 'Bayesian inference for logistic models using Pólya–Gamma latent variables', *Journal of the American Statistical Association* **108**(504), 1339–1349.

Power, S. & Goldman, J. V. (2019), 'Accelerated sampling on discrete spaces with non-reversible Markov processes'. arXiv:1912.04681.

Sherlock, C. & Thiery, A. H. (2020), 'A discrete bouncy particle sampler', *Biometrika, to appear* .

Smith, M. & Kohn, R. (1996), 'Nonparametric regression using Bayesian variable selection', *Journal of Econometrics* **75**(2), 317 – 343.

Song, Q., Sun, Y., Ye, M. & Liang, F. (2020), 'Extended stochastic gradient Markov chain Monte Carlo for large-scale Bayesian variable selection', *Biometrika* **107**, 997–1004.

Stephens, M. (2000), 'Bayesian analysis of mixture models with an unknown number of components-an alternative to reversible jump methods', *Annals of Statistics* pp. 40–74.

Vanetti, P., Bouchard-Côté, A., Deligiannidis, G. & Doucet, A. (2017), 'Piecewise-Deterministic Markov Chain Monte Carlo', *arXiv:1707.05296* .

Wang, S., Nan, B., Rosset, S. & Zhu, J. (2011), 'Random lasso', *Ann. Appl. Stat.* **5**(1), 468–485.

Wu, C. & Robert, C. P. (2020), 'Coordinate sampler: a non-reversible Gibbs-like MCMC sampler', *Statistics and Computing* **30**(3), 721–730.

Yang, Y., Wainwright, M. J. & Jordan, M. I. (2016), 'On the computational complexity of high-dimensional Bayesian variable selection', *Ann. Statist.* **44**(6), 2497–2532.

Zanella, G. & Roberts, G. (2019), 'Scalable importance tempering and Bayesian variable selection', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **81**(3), 489–517.

Supplementary Material: Reversible Jump PDMP Samplers for Variable Selection

A Proofs

A.1 Proof of Theorem 1

To simplify notations, we use $\boldsymbol{z} = (\boldsymbol{\theta}, \boldsymbol{v})$ interchangeably through the proof. We follow previous papers (Vanetti et al. 2017, Fearnhead et al. 2018, Bierkens et al. 2018) in our general approach for calculating the invariant distribution of the PDMP, and start by overviewing this. We will use the infinitesimal generator of the PDMP, and we let $(A, \mathcal{D}(A))$ be the infinitesimal generator and it's domain. Further let $\mathrm{E}_{\boldsymbol{z}}$ denote expectation with respect to the PDMP process $\boldsymbol{Z}_t$ started with initial condition $\boldsymbol{Z}_0 = \boldsymbol{z}$.

Then, by definition of the generator, for functions in the domain, $f \in D(A)$,

$$(Af)(\boldsymbol{z}) = \lim_{t \to 0^+} \frac{\mathrm{E}_{\boldsymbol{z}}\{f(\boldsymbol{Z}_t)\} - f(\boldsymbol{z})}{t}.$$

So Af can be viewed as the time-derivative of the expectation of $f(\boldsymbol{Z}_t)$. If $\nu(\cdot)$ is the invariant distribution of the PDMP, and we start the process from ν, $\boldsymbol{Z}_0 \sim \nu(\cdot)$, then expectations will be constant. This means that

$$\int (Af)(\boldsymbol{z})\nu(\boldsymbol{z})\mathrm{d}\boldsymbol{z} = 0. \tag{10}$$

The proof then inverts this argument. Intuitively the idea is that if (10) holds for some density ν and for sufficiently many functions f then ν must be an invariant density of the PDMP (in the same way that if two distributions have the same expectation as each other for a sufficiently large class of functions then those distributions must be identical). Results

in, for example, Durmus et al. (2021) and Chevallier et al. (2021) can be used to make this argument rigorous.

Following this intuition, the outline of the proof is as follows. By applying integration by parts to (10) we can obtain a sufficient condition on $\nu(\cdot)$ for (10) to hold for functions f in $D(A)$. Below we show that, for the choice of $\beta_{i,j}$ and $Q_{i,j}$ given in the statement of the theorem, that this sufficient condition holds for our target distribution, $\nu(\cdot)$. The fact that $\nu(\cdot)$ is thus the invariant distribution then follows immediately from the results in Chevallier et al. (2021).

First we need to define the form of the generator A. The form of the generator for a PDMP with boundaries is given in Davis (1993), however the definition of boundaries in Davis (1993) differs slightly from ours. Our process still fits the definition of Davis (1993): one simply needs to separate each $\mathcal{X}^k$ in two at the boundary and consider the two parts as two separated disjoint spaces. So for example, if we have a model with a single variable then we would need to split the state-space for that variable into two, one for positive values and one for negative values. More generally, if space E_k corresponds to a model with d_k variables, then we need to separate it into 2^{d_k} regions corresponding to the different combinations of the parameters associated to each variable being positive or negative.

This has an implication on the function spaces considered. The domain of the generator will contain only continuous functions, but these need only be continuous on each of the disjoint spaces. In other words, if we view these functions as defined on the state space E, they can be discontinuous at the boundaries at 0. In practice, for $(\boldsymbol{\theta}, \boldsymbol{v})$ on a boundary $\Gamma_{i,j}$, we consider two "sides" of the boundary and write $f(\boldsymbol{\theta}^+, \boldsymbol{v}) = \lim_{t\to 0, t>0} f(\boldsymbol{\theta} + t\boldsymbol{v}, \boldsymbol{v})$ and $f(\boldsymbol{\theta}^-, \boldsymbol{v}) = \lim_{t\to 0, t>0} f(\boldsymbol{\theta} - t\boldsymbol{v}, \boldsymbol{v})$. Furthermore, points in the entrance boundary are added to the state space in Davis construction (Davis 1993, p. 57), hence we write $f(\boldsymbol{\theta}, \boldsymbol{v}) = f(\boldsymbol{\theta}^+, \boldsymbol{v})$ for $(\boldsymbol{\theta}, \boldsymbol{v}) \in \Gamma_{i,j} \times \mathcal{V}_i$. On the other hand, for points on the exit

boundary, i.e. the "$\boldsymbol{\theta}^-$ side", the trajectory with velocity $\boldsymbol{v}$ would instantaneously hit the boundary $\Gamma_{i,j}$, and this is excluded from the state space. Hence the process can never jump to the "$\boldsymbol{\theta}^-$ side", only reach it (in the limit) through the deterministic dynamic. For a detailed description of this we refer to the construction found in (Davis 1993, p. 57).

Furthermore, we should add an index to the state space to differentiate between the different quadrants of $\mathcal{X}_k$ (which are now separated in different spaces), but since they do not overlap, this information is redundant with the position $\boldsymbol{\theta}$, and we drop this index for clarity.

From Davis (1993) we have that the generator can be written as

$$Af = \boldsymbol{v} \cdot \nabla_{\boldsymbol{\theta}} f(\boldsymbol{z}) + \lambda(\boldsymbol{z}) \int \{f(\boldsymbol{z}') - f(\boldsymbol{z})\} \mathcal{Q}(\mathrm{d}\boldsymbol{z}', \boldsymbol{z}),$$

where $\mathcal{Q}$ denotes the transition kernel at events.

Thus

$$\begin{aligned} \int_E (Af)(\boldsymbol{z}) \nu(\mathrm{d}\boldsymbol{z}) = & \sum_i \int_{E_i} \boldsymbol{v} \cdot \nabla_{\boldsymbol{\theta}} f(\boldsymbol{z}) \nu_i(\boldsymbol{z}) \mathrm{d}\boldsymbol{z} \\ + & \sum_i \int_{E_i} \lambda_i(\boldsymbol{z}) \int \{f(\boldsymbol{z}') - f(\boldsymbol{z})\} Q_i(\boldsymbol{z}'|\boldsymbol{z}) \nu_i(\boldsymbol{z}) \mathrm{d}\boldsymbol{z}' \mathrm{d}\boldsymbol{z}, \end{aligned}$$

where we have split the integration over E into a sum of the integrals over each E_i, and then used the definition of the rate and transition density for each E_i.

It is helpful to first consider the behaviour of the PDMP at exploring a single model. To do this consider a function f of sufficient regularity with bounded support on E_i, that we can apply integration-by-parts to get:

$$\int_{E_i} \boldsymbol{v} \cdot \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}, \boldsymbol{v}) \nu_i(\boldsymbol{\theta}, \boldsymbol{v}) \mathrm{d}\boldsymbol{\theta} \mathrm{d}\boldsymbol{v} = - \int_{E_i} f(\boldsymbol{\theta}, \boldsymbol{v}) \nabla_{\boldsymbol{\theta}} \nu_i(\boldsymbol{\theta}, \boldsymbol{v}) \cdot \boldsymbol{v} \, \mathrm{d}\boldsymbol{\theta} \mathrm{d}\boldsymbol{v}.$$

Furthermore, using Fubini's theorem with the fact that $\int |\lambda(\boldsymbol{z})| d\nu < \infty$:

$$\int_{E_i} \lambda_i(\boldsymbol{z}) \int_{\mathcal{V}^i} (f(\boldsymbol{\theta}, \boldsymbol{v}') - f(\boldsymbol{z})) Q_i(\mathrm{d}\boldsymbol{v}'|\boldsymbol{z}) d\nu(\boldsymbol{z}) = \int_{E_i} f(\boldsymbol{z}) \left[\int_{\mathcal{V}^i} Q_i(\boldsymbol{v}'|\boldsymbol{z}) \lambda_i(\boldsymbol{\theta}, \boldsymbol{v}') \nu(\boldsymbol{\theta}, \boldsymbol{v}') \mathrm{d}\boldsymbol{v}' \right] \mathrm{d}\boldsymbol{z} - \int_{E_i} f(\boldsymbol{z}) \lambda_i(\boldsymbol{z}) \nu(\boldsymbol{z}) \mathrm{d}\boldsymbol{z}.$$

Thus for such a function

$$\begin{aligned} \int_E (Af)(\boldsymbol{z}) \nu(\boldsymbol{z}) \mathrm{d}\boldsymbol{z} &= \int_{E_i} (Af)(\boldsymbol{z}) \nu_i(\boldsymbol{z}) \mathrm{d}\boldsymbol{z} = - \int_{E_i} f(\boldsymbol{\theta}, \boldsymbol{v}) \nabla_{\boldsymbol{\theta}} \nu_i(\boldsymbol{\theta}, \boldsymbol{v}) \cdot \boldsymbol{v} \, \mathrm{d}\boldsymbol{\theta} \mathrm{d}\boldsymbol{v} \\ &+ \int_{E_i} f(\boldsymbol{z}) \left[\int_{\mathcal{V}^i} Q_i(\boldsymbol{v}', \boldsymbol{z}) \lambda_i(\boldsymbol{\theta}, \boldsymbol{v}') \nu(\boldsymbol{\theta}, \boldsymbol{v}') \mathrm{d}\boldsymbol{v}' \right] \mathrm{d}\boldsymbol{z} - \int_{E_i} f(\boldsymbol{z}) \lambda_i(\boldsymbol{z}) \nu(\boldsymbol{z}) \mathrm{d}\boldsymbol{z} \\ &= \int_{E_i} f(\boldsymbol{z}) \left[\int_{\mathcal{V}^i} Q_i(\boldsymbol{v}'|\boldsymbol{z}) \lambda_i(\boldsymbol{\theta}, \boldsymbol{v}') \nu(\boldsymbol{\theta}, \boldsymbol{v}') \mathrm{d}\boldsymbol{v}' - \lambda_i(\boldsymbol{z}) \nu(\boldsymbol{z}) - \nabla_{\boldsymbol{\theta}} \nu(\boldsymbol{\theta}, \boldsymbol{v}) \cdot \boldsymbol{v} \right] \mathrm{d}\boldsymbol{z}. \quad (11) \end{aligned}$$

Essentially because of the above argument applied to the generator of the base PDMP sampler for E_i, our assumption that the base PDMP sampler on E_i leaves ν_i invariant means that Q_i and λ_i are such that

$$\int_{\mathcal{V}^i} Q_i(\boldsymbol{v}', \boldsymbol{z}) \lambda_i(\boldsymbol{\theta}, \boldsymbol{v}') \nu(\boldsymbol{\theta}, \boldsymbol{v}') \mathrm{d}\boldsymbol{v}' - \lambda_i(\boldsymbol{z}) \nu(\boldsymbol{z}) - \nabla_{\boldsymbol{\theta}} \nu(\boldsymbol{\theta}, \boldsymbol{v}) \cdot \boldsymbol{v} = 0,$$

and thus (11) is 0.

To derive the invariant distribution for the reversible jump PDMP process we need to consider more general functions f, that are non-zero for multiple model spaces. We can apply the same approach to re-arrange $\int (Af)(\boldsymbol{z}) \nu(\boldsymbol{z}) \mathrm{d}\boldsymbol{z}$ as above, but now extra boundary terms will appear when we perform the integration by parts, see e.g. Bierkens et al. (2018)

and Theorem 2 of Chevallier et al. (2021). This gives that for all $f \in \mathcal{D}(A)$:

$$\begin{aligned}\int_{E_i} Af(\boldsymbol{\theta},\boldsymbol{v})\mathrm{d}\nu_i(\boldsymbol{\theta},\boldsymbol{v}) &= -\int_{E_i} f(\boldsymbol{\theta},\boldsymbol{v})\nabla_{\boldsymbol{\theta}}\nu_i(\boldsymbol{\theta},\boldsymbol{v})\cdot\boldsymbol{v}\,\mathrm{d}\boldsymbol{\theta}\mathrm{d}\boldsymbol{v} \\ &+ \int_{E_i} f(\boldsymbol{z})\left[\int_{\mathcal{V}^i} Q_i(\boldsymbol{v}'|\boldsymbol{z})\lambda_i(\boldsymbol{\theta},\boldsymbol{v}')\nu(\boldsymbol{\theta},\boldsymbol{v}')\mathrm{d}\boldsymbol{v}'\right]\mathrm{d}\boldsymbol{z} - \int_{E_i} f(\boldsymbol{z})\lambda_i(\boldsymbol{z})\nu(\boldsymbol{z})\mathrm{d}\boldsymbol{z} \\ &+ \sum_{j,(i,j)\in\mathcal{T}}\int_{\Gamma_{i,j}} (f(\boldsymbol{\theta}^-,\boldsymbol{v}) - f(\boldsymbol{\theta},\boldsymbol{v}))\nu_i(\boldsymbol{\theta},\boldsymbol{v})|\langle\boldsymbol{v},n(\boldsymbol{\theta})\rangle|d\sigma(\boldsymbol{\theta})\mathrm{d}\boldsymbol{v} \\ &+ \sum_{j,(j,i)\in\mathcal{T}}\int_{E_i}\beta_{j,i}(\boldsymbol{\theta},\boldsymbol{v})\int_{\mathcal{V}^j}(f(\boldsymbol{\theta},\boldsymbol{v}') - f(\boldsymbol{\theta},\boldsymbol{v}))Q_{j,i}(\mathrm{d}\boldsymbol{v}'|\boldsymbol{v})\nu_i(\boldsymbol{\theta},\boldsymbol{v})\mathrm{d}\boldsymbol{\theta}\mathrm{d}\boldsymbol{v},\end{aligned}$$

where, as defined above, $f(\boldsymbol{\theta},\boldsymbol{v}) = \lim_{t\to 0,t>0} f(\boldsymbol{\theta}+t\boldsymbol{v},\boldsymbol{v})$ and $f(\boldsymbol{\theta}^-,\boldsymbol{v}) = \lim_{t\to 0,t>0} f(\boldsymbol{\theta}-t\boldsymbol{v},\boldsymbol{v})$ for $(\boldsymbol{\theta},\boldsymbol{v}) \in \Gamma_{i,j}$. The terms of the first two lines are as for the simpler case above, and the terms in the latter two lines correspond to the boundary terms.

Using (11), we have the terms in the first two lines are zero. Thus, summing over all i, we get:

$$\begin{aligned}\int_E Af(\boldsymbol{\theta},\boldsymbol{v})\mathrm{d}\nu(\boldsymbol{\theta},\boldsymbol{v}) &= \sum_{(i,j)\in\mathcal{T}}\int_{\Gamma_{i,j}} (f(\boldsymbol{\theta}^-,\boldsymbol{v}) - f(\boldsymbol{\theta},\boldsymbol{v}))\nu_i(\boldsymbol{\theta},\boldsymbol{v})|\langle\boldsymbol{v},n(\boldsymbol{\theta})\rangle|\mathrm{d}\sigma(\boldsymbol{\theta})\mathrm{d}\boldsymbol{v} \\ &+ \sum_{(i,j)\in\mathcal{T}}\int_{E_j}\beta_{i,j}(\boldsymbol{\theta},\boldsymbol{v})\int_{\mathcal{V}^i}(f(\boldsymbol{\theta},\boldsymbol{v}') - f(\boldsymbol{\theta},\boldsymbol{v}))Q_{i,j}(\mathrm{d}\boldsymbol{v}'|\boldsymbol{v})\nu_i(\boldsymbol{\theta},\boldsymbol{v})\mathrm{d}\boldsymbol{\theta}\mathrm{d}\boldsymbol{v}.\end{aligned}$$

To proceed we need to use properties of the domain of the generator, and thus of functions f that we are considering. These can be obtained by considering the extended infinitesimal generator of the PDMP, and using the fact that the domain of A is a subset of the domain of the extended infinitesimal generator. We summarise the key results we need, though see Davis (1993) for more details.

Denote the extended infinitesimal generator by $\mathfrak{A}f$ and its domain $\mathcal{D}(\mathfrak{A})$; these are given for general PDMPs in Theorem 26.14 of Davis (1993). The behaviour of the PDMP

at the boundary is not encoded in the expression $\mathfrak{A}f$ but in the domain $\mathcal{D}(\mathfrak{A})$: the domain includes only the set of functions such that for every $\boldsymbol{z}$ on the boundary,

$$f(\boldsymbol{z}) = \int_E f(\boldsymbol{z}')Q(\mathrm{d}\boldsymbol{z}', \boldsymbol{z}),$$

where Q is the jump kernel at the boundary. In our case, the jump at the boundary is the deterministic projection $g_{i,j}$ with probability $p_{i,j}$, or crossing the boundary. Hence the condition becomes

$$f(\boldsymbol{\theta}^-, \boldsymbol{v}) = p_{i,j} f(g_{i,j}(\boldsymbol{\theta}, \boldsymbol{v})) + (1 - p_{i,j}) f(\boldsymbol{\theta}, \boldsymbol{v})$$

for $(\boldsymbol{\theta}, \boldsymbol{v}) \in \Gamma_{i,j}$.

Hence, since $\mathcal{D}(A) \subset \mathcal{D}(\mathfrak{A})$, for $f \in \mathcal{D}(A)$:

$$\begin{aligned}\int_{\Gamma_{i,j}} (f(\boldsymbol{\theta}^-, \boldsymbol{v}) - f(\boldsymbol{\theta}, \boldsymbol{v}))\nu_{i,j}(\boldsymbol{\theta}, \boldsymbol{v})\mathrm{d}\sigma(\boldsymbol{\theta})\mathrm{d}\boldsymbol{v} &= p_{i,j} \int_{\Gamma_{i,j}} (f(g_{i,j}(\boldsymbol{\theta}, \boldsymbol{v})) - f(\boldsymbol{\theta}, \boldsymbol{v}))\nu_{i,j}(\boldsymbol{\theta}, \boldsymbol{v})\mathrm{d}\sigma(\boldsymbol{\theta})\mathrm{d}\boldsymbol{v} \\ &= p_{i,j} \int_{E_j} f(\boldsymbol{z})\mathrm{d}\bar{\nu}_{i,j}(\boldsymbol{z}) - p_{i,j} \int_{\Gamma_{i,j}} f(\boldsymbol{\theta}, \boldsymbol{v})\nu_{i,j}(\boldsymbol{\theta}, \boldsymbol{v})\mathrm{d}\sigma(\boldsymbol{\theta})d\boldsymbol{v},\end{aligned}$$

where the last equation is obtained by definition of the pushforward measure $\bar{\nu}_{i,j}$. Finally, using a change the change of variable from $\boldsymbol{z}' = (\boldsymbol{\theta}', \boldsymbol{v}')$ to $\alpha, \boldsymbol{z}$ defined as $\boldsymbol{z}' = G_{i,j}(\alpha, \boldsymbol{z})$:

$$\int_{\Gamma_{i,j}} f(\boldsymbol{z}')\nu_{i,j}(\boldsymbol{z}')\,\mathrm{d}\boldsymbol{z}' = \int_{E_j} \int_U f(G_{i,j}(\alpha, \boldsymbol{z}))\nu_{i,j}(G(\alpha, \boldsymbol{z}))|J_{G_{i,j}}(\alpha, \boldsymbol{z})|\,\mathrm{d}\alpha\,\mathrm{d}\boldsymbol{z}.$$

Thus:

$$\int_{\Gamma_{i,j}} (f(\boldsymbol{\theta}^-, \boldsymbol{v}) - f(\boldsymbol{\theta}, \boldsymbol{v}))\nu_{i,j}(\boldsymbol{\theta}, \boldsymbol{v})\mathrm{d}\sigma(\boldsymbol{\theta})\mathrm{d}\boldsymbol{v} =$$
$$p_{i,j} \int_{E_j} f(\boldsymbol{z})\mathrm{d}\bar{\nu}_{i,j}(\boldsymbol{z}) - p_{i,j} \int_{E_j} \int_U f(G_{i,j}(\alpha, \boldsymbol{z}))\nu_{i,j}(G(\alpha, \boldsymbol{z}))|J_{G_{i,j}}(\alpha, \boldsymbol{z})|\,\mathrm{d}\alpha\,\mathrm{d}\boldsymbol{z}.$$

Since $Q_{i,j}$ integrates to 1, we get:

$$\int_{E_j} \beta_{i,j}(\boldsymbol{\theta}, \boldsymbol{v}) \int_{\mathcal{V}^i} (f(\boldsymbol{\theta}, \boldsymbol{v}') - f(\boldsymbol{\theta}, \boldsymbol{v}))Q_{i,j}(\mathrm{d}\boldsymbol{v}'|\boldsymbol{v})\nu_i(\boldsymbol{\theta}, \boldsymbol{v})\mathrm{d}\boldsymbol{\theta}\mathrm{d}\boldsymbol{v}$$
$$= \int_{E_j} \beta_{i,j}(\boldsymbol{\theta}, \boldsymbol{v}) \left(\int_{\mathcal{V}^i} f(\boldsymbol{\theta}, \boldsymbol{v}')Q_{i,j}(\mathrm{d}\boldsymbol{v}'|\boldsymbol{v}) - f(\boldsymbol{\theta}, \boldsymbol{v}) \right) \nu_i(\boldsymbol{\theta}, \boldsymbol{v})\mathrm{d}\boldsymbol{\theta}\mathrm{d}\boldsymbol{v}.$$

Using the definitions of $G_{i,j}$, and the fact that $Q_{i,j}(\cdot|\boldsymbol{v})$ has support on $g_{i,j}^{-1}(\{\boldsymbol{z}\})$:

$$\int_{\mathcal{V}^i} f(\boldsymbol{\theta}, \boldsymbol{v}')Q_{i,j}(\mathrm{d}\boldsymbol{v}'|\boldsymbol{v}) = \int_U f(G_{i,j}(\alpha, \boldsymbol{z}))Q_{i,j}(\alpha|\boldsymbol{z})\,\mathrm{d}\alpha.$$

We deduce:

$$\int_E Af d\nu = \int_{E_j} \beta_{i,j}(\boldsymbol{\theta}, \boldsymbol{v}) \left(\int_U f(G_{i,j}(\alpha, \boldsymbol{z}))Q_{i,j}(\alpha|\boldsymbol{z})\,\mathrm{d}\alpha - f(\boldsymbol{\theta}, \boldsymbol{v}) \right) \nu_i(\boldsymbol{\theta}, \boldsymbol{v})\mathrm{d}\boldsymbol{\theta}\mathrm{d}\boldsymbol{v}$$
$$+ p_{i,j} \int_{E_j} f(\boldsymbol{z})\mathrm{d}\bar{\nu}_{i,j}(\boldsymbol{z}) - p_{i,j} \int_{E_j} \int_U f(G_{i,j}(\alpha, \boldsymbol{z}))\nu_{i,j}(G(\alpha, \boldsymbol{z}))|J_{G_{i,j}}(\alpha, \boldsymbol{z})|\,\mathrm{d}\alpha\,\mathrm{d}\boldsymbol{z}.$$

Since from the assumption of the Theorem

$$\beta_{i,j}(\boldsymbol{z}) = p_{i,j}\frac{\bar{\nu}_{i,j}(\boldsymbol{z})}{\nu(\boldsymbol{z})} \qquad \text{for } \boldsymbol{z} \in E_j$$
$$Q_{i,j}(\alpha|\boldsymbol{z}) = \frac{\nu_{i,j}(G_{i,j}(\alpha, \boldsymbol{z}))|J_{G_{i,j}}(\alpha, \boldsymbol{z})|}{\bar{\nu}_{i,j}(\boldsymbol{z})} \qquad \text{for } \boldsymbol{z} \in E_j \text{ and } \alpha \in U$$

we have $\int_E Af \mathrm{d}\nu = 0$ for all $f \in \mathcal{D}(A)$. That this implies ν is invariant, follows directly from Theorem 1 of Chevallier et al. (2021).

A.2 Discrete Velocity Spaces

The following is the equivalent to Theorem 1 but for discrete velocity spaces. The only difference is that in this case we do not need a Jacobian term in the definition of the transition kernel $Q_{i,j}$.

Theorem 2. *Assume the velocity space is discrete, that the base PDMP on E_k has ν_k as its invariant distribution, and that the following conditions hold:*

1. *$\int |\lambda(\boldsymbol{z})| d\nu < \infty$ where λ is the jump rate of the underlying PDMP without jumps between models.*
2. *For all k, π_k is C^1 on $\mathcal{X}_k$.*
3. *For any k, and any $\boldsymbol{v} \in \mathcal{V}^k$, $\nabla \pi_k \cdot \boldsymbol{v}$ is in $L_1(\mathrm{Leb})$.*

Then the measure ν is invariant if for every $(i,j) \in \mathcal{T}$, the following conditions hold

$$\beta_{i,j}(\boldsymbol{z}) = p_{i,j} \frac{\bar{\nu}_{i,j}(\boldsymbol{z})}{\nu(\boldsymbol{z})} \qquad \text{for } \boldsymbol{z} \in E_j$$
$$Q_{i,j}(\alpha|\boldsymbol{z}) = \frac{\nu_{i,j}(G_{i,j}(\alpha, \boldsymbol{z}))}{\bar{\nu}_{i,j}(\boldsymbol{z})} \qquad \text{for } \boldsymbol{z} \in E_j \text{ and } \alpha \in U.$$

Proof. The proof is largely the same as the proof of Theorem 1. The only change is how to treat the change of variable $\boldsymbol{z}' = G_{i,j}(\alpha, \boldsymbol{z})$. First, notice that $G_{i,j}$ leaves the position invariant since $g_{i,j}$ leaves the position invariant. Then, since the velocity space is discrete

and $\alpha \mapsto G_{i,j}(\alpha, \boldsymbol{z})$ is a one to one mapping for a fixed $\boldsymbol{z} \in E_j$:

$$\begin{aligned}\int_{\Gamma_{i,j}} f(\boldsymbol{z}')\nu_{i,j}(\boldsymbol{z}')\,\mathrm{d}\boldsymbol{z}' &= \int_{\mathcal{X}^j} \sum_{\boldsymbol{v}'\in\mathcal{V}^i} f(\boldsymbol{\theta}', \boldsymbol{v}')\nu_{i,j}(\boldsymbol{\theta}', \boldsymbol{v}')\,\mathrm{d}\boldsymbol{\theta}' \\ &= \int_{\mathcal{X}^j} \sum_{\boldsymbol{v}\in E_j} \sum_{\alpha\in U} f(G_{i,j}(\alpha, (\boldsymbol{\theta}, \boldsymbol{v}))\nu_{i,j}(G_{i,j}(\alpha, (\boldsymbol{\theta}, \boldsymbol{v})))\,\mathrm{d}\boldsymbol{\theta}.\end{aligned}$$

which we can write in integral form as

$$\int_{\Gamma_{i,j}} f(\boldsymbol{z}')\nu_{i,j}(\boldsymbol{z}')|\langle \boldsymbol{v}', \boldsymbol{n}_{i,j}\rangle|\,\mathrm{d}\boldsymbol{z}' = \int_{E_j}\int_U f(G_{i,j}(\alpha, (\boldsymbol{\theta}, \boldsymbol{v}))\nu_{i,j}(G_{i,j}(\alpha, (\boldsymbol{\theta}, \boldsymbol{v})))\,\mathrm{d}\alpha\mathrm{d}\boldsymbol{z}.$$

□

A.3 Proof of Proposition 1

Proof. Here:

$$\mathcal{V}^i = \{\boldsymbol{v} \in \mathbb{R}^{|\gamma^i|} \text{ such that } \|\boldsymbol{v}\| = 1\}$$

For $\boldsymbol{z} \in E_i$:

$$\nu(\boldsymbol{z}) = \pi_i(\boldsymbol{\theta})\frac{1}{A_{sphere}(|\gamma^i|)}$$

with $A_{sphere}(|\gamma^i|) = \frac{\Gamma(\frac{|\gamma^i|}{2})}{2\pi^{\frac{|\gamma^i|}{2}}}$ the area of the unit sphere of $\mathbb{R}^{|\gamma^i|}$.

In this case we can define $G_{i,j} : (-1, 1) \times E_j \to E_i$ such that $\boldsymbol{z}' = (\boldsymbol{\theta}', \boldsymbol{v}') = G_{i,j}(\alpha, \boldsymbol{z})$, where $\boldsymbol{z} = (\boldsymbol{\theta}, \boldsymbol{v})$, as $\boldsymbol{\theta}' = \boldsymbol{\theta}$ and

$$\boldsymbol{v}' = \sqrt{1-\alpha^2}\boldsymbol{v} + \alpha\boldsymbol{n}_{i,j}.$$

Let $B \subset \mathcal{V}^j$

$$\begin{aligned}\int_B \bar{\nu}_{i,j}(\boldsymbol{\theta}, \boldsymbol{v})\mathrm{d}\boldsymbol{v} &= \pi_i(\boldsymbol{\theta}) \int_{g_{i,j}^{-1}(B)} |<\boldsymbol{v}', \boldsymbol{n}_{i,j}>| \frac{1}{A_{sphere}(|\gamma^i|)}\mathrm{d}\boldsymbol{v}' \\ &= \pi_i(\boldsymbol{\theta}) \frac{1}{A_{sphere}(|\gamma^i|)} \int_{[-1,1]\times B} |<\sqrt{1-\alpha^2}\boldsymbol{v} + \alpha \boldsymbol{n}_{i,j}, \boldsymbol{n}_{i,j}>| |J_{G_{i,j}}| \mathrm{d}\alpha \mathrm{d}\boldsymbol{v} \\ &= 2\pi_i(\boldsymbol{\theta}) \frac{1}{A_{sphere}(|\gamma^i|)} \int_{(0,1]\times B} \alpha |J_{G_{i,j}}| \mathrm{d}\alpha \mathrm{d}\boldsymbol{v},\end{aligned}$$

where the last inequality uses the symmetry of the integral with respect to α, and that $\langle \boldsymbol{v}, \boldsymbol{n}_{i,j} \rangle = 0$ by definition of velocities in E_j.

The determinant of $G_{i,j}$ must be carefully computed since the velocities lives in the sphere and not on the full vector space. We get:

$$|J_{G_{i,j}}(\alpha, \boldsymbol{v})| = \sqrt{1-\alpha^2}^{|\gamma^j|-2}.$$

Therefore for $|\gamma^j| > 0$:

$$\bar{\nu}_{i,j}(\boldsymbol{z}) = \pi_i(\boldsymbol{\theta}) \frac{2}{A_{sphere}(|\gamma^i|)} . \frac{1}{|\gamma^j|}$$

Hence for $|\gamma^j| > 0$:

$$\beta_{i,j} = p_{i,j} \frac{\pi_i(\boldsymbol{\theta})}{\pi_j(\boldsymbol{\theta})} \frac{2A_{sphere}(|\gamma^j|)}{A_{sphere}(|\gamma^i|)} \frac{1}{|\gamma^j|},$$

and

$$Q_{i,j}(\alpha|\boldsymbol{z}) = \frac{|\alpha||\gamma^j|\sqrt{1-\alpha^2}^{|\gamma^j|-2}}{2} \text{ for } \boldsymbol{z} \in E_j \text{ and } \alpha \in (-1, 1)$$

For $|\gamma^j| = 0$, BPS and ZigZag are equivalent, thus we use ZigZag rates. □

A.4 Proof of Proposition 2

Proof. Here:

$$\mathcal{V}^i = \mathbb{R}^{|\gamma^i|}$$

Therefore, for $\boldsymbol{z} \in E_i$:

$$\nu(\boldsymbol{z}) = \pi_i(\boldsymbol{\theta}) \frac{1}{(2\pi)^{\frac{|\gamma^i|}{2}}} e^{-\frac{1}{2}\|\boldsymbol{v}\|^2}$$

For $(i,j) \in \mathcal{T}$ with $|\gamma^j| > 0$, we can define $G_{i,j} : (-1,1) \times E_j \to E_i$ such that $\boldsymbol{z}' = (\boldsymbol{\theta}', \boldsymbol{v}') = G_{i,j}(\alpha, \boldsymbol{z})$, where $\boldsymbol{z} = (\boldsymbol{\theta}, \boldsymbol{v})$, as $\boldsymbol{\theta}' = \boldsymbol{\theta}$ and

$$\boldsymbol{v}' = \boldsymbol{v} + \alpha \boldsymbol{n}_{i,j}.$$

Clearly, we have $|J_{G_{i,j}}(\alpha, \boldsymbol{z})| = 1$. Let $B \subset \mathcal{V}^j$

$$\begin{aligned}
\int_B \bar{\nu}_{i,j}(\boldsymbol{\theta}, \boldsymbol{v}) \mathrm{d}\boldsymbol{v} &= \pi_i(\boldsymbol{\theta}) \int_{g_{i,j}^{-1}(B)} | < \boldsymbol{v}', \boldsymbol{n}_{i,j} > | \frac{1}{(2\pi)^{\frac{|\gamma^i|}{2}}} e^{-\frac{1}{2}\|\boldsymbol{v}'\|^2} \mathrm{d}\boldsymbol{v}' \\
&= \pi_i(\boldsymbol{\theta}) \frac{1}{(2\pi)^{\frac{|\gamma^i|}{2}}} \int_{\mathbb{R}\times B} | < \boldsymbol{v} + \alpha \boldsymbol{n}_{i,j}, \boldsymbol{n}_{i,j} > | e^{-\frac{1}{2}\|\boldsymbol{v}+\alpha\boldsymbol{n}_{i,j}\|^2} |J_{G_{i,j}}| \mathrm{d}\alpha \mathrm{d}\boldsymbol{v} \\
&= \pi_i(\boldsymbol{\theta}) \frac{1}{(2\pi)^{\frac{|\gamma^i|}{2}}} \int_{\mathbb{R}\times B} |\alpha| e^{-\frac{1}{2}(\alpha^2 + \|\boldsymbol{v}\|^2)} \mathrm{d}\alpha \mathrm{d}\boldsymbol{v} \\
&= \pi_i(\boldsymbol{\theta}) 2 \frac{1}{(2\pi)^{\frac{|\gamma^i|}{2}}} \int_B e^{-\frac{1}{2}(\|\boldsymbol{v}\|^2)} \mathrm{d}\boldsymbol{v}.
\end{aligned}$$

Therefore for $|\gamma^j| > 0$:

$$\bar{\nu}_{i,j}(\boldsymbol{z}) = \pi_i(\boldsymbol{\theta}) 2 \frac{1}{(2\pi)^{\frac{|\gamma^i|}{2}}} e^{-\frac{1}{2}(\|\boldsymbol{v}\|^2)},$$

and

$$\beta_{i,j} = p_{i,j} \frac{\pi_i(\boldsymbol{\theta})}{\pi_j(\boldsymbol{\theta})} \frac{2}{\sqrt{2\pi}}$$

Furthermore, $|J_{G_{i,j}}| = 1$ thus:

$$Q_{i,j}(\alpha|\boldsymbol{z}) = 2|\alpha|e^{-\frac{1}{2}\alpha^2} \text{ for } \boldsymbol{z} \in E_j \text{ and } \alpha \in \mathbb{R}.$$

□

B Sensitivity to tuning parameters

B.1 The reversible jump parameter $p_{i,j}$

Our reversible jump PDMP sampler introduces one additional tuning parameter to any for the base PDMP sampler. This is the reversible jump parameter $p_{i,j}$, that specifies the probability of removing a variable if the variable's parameter hits 0. Higher values thus encourage more mixing between models and smaller values encourage more mixing within models. A similar parameter exists for all reversible jump MCMC algorithms, as we need to specify the relative proportion of within and between model proposals.

It is natural to fix $p_{i,j}$ to the same value for all pairs i, j. We empirically look at the sensitivity of our methods to this parameter. To explore this we consider the following simple spike and slab model:

$$\pi(\theta) = \prod_{i=1}^{d} [s\phi(\theta_i) + (1-s)\delta_0(\theta_i)] \tag{12}$$

where $\phi(\cdot)$ denotes the density of a normal distribution and $s \in (0, 1)$ is the probability

a variable is included in the model. We consider this simple model as it allows for exact calculations of posterior marginal means and marginal variable inclusion probabilities.

We will focus on the reversible jump Zig-Zag algorithm, as there are no other tuning parameters for this algorithm. We consider the spike and slab model with $d = 50$ variables and $s = 0.2, 0.4, 0.6$ and 0.8. For $s \approx 0$ or $s \approx 1$, only a single model will have high posterior probability whereas for $s \approx 0.5$ many models are likely and searching the model space will be more challenging.

The reversible jump Zig-Zag was run 100 times using 10^5 events for each setting of s with $p_{i,j} \in \{0.2, 0.4, 0.6, 0.8\}$. Figure 6 shows the average statistical efficiency in terms of estimating the marginal posterior means and marginal posterior probabilities over these 100 runs. The figure also shows histograms of number of models visited for each simulation and choice of $p_{i,j}$.

As expected, we see that the Zig-Zag has a harder time estimating marginal posterior inclusion probabilities for $s \approx 0.5$ and a harder time estimating marginal means when $s \approx 1$. By increasing the parameter $p_{i,j}$ the number of models seen is increased regardless of the value of s. When $p_{i,j}$ is large more models are visited and the marginal posterior probabilities are more accurately estimated. However if it is visiting many models the marginal means are less accurately estimated. For this reason a practitioner may like to set the value $p_{i,j} \approx 0.8$ if efficient model space exploration is desired or $p_{i,j} \approx 0.2$ if efficient parameter space exploration is desired. These simulations all started the process at stationarity, an additional concern may arise if a good initial starting point is not known. In this case it makes sense to favor higher $p_{i,j}$ to not get stuck in models with low posterior probability. In our experiments we fixed $p_{i,j} = 0.6$ as a rough trade-off between parameter and model space exploration.

To further check the robustness of this tuning parameter we re-ran the simulations from the logistic regression example in Section 5.1 using Zig-Zag with $p_{i,j} = 0.1, 0.3, 0.6, 0.7$ and

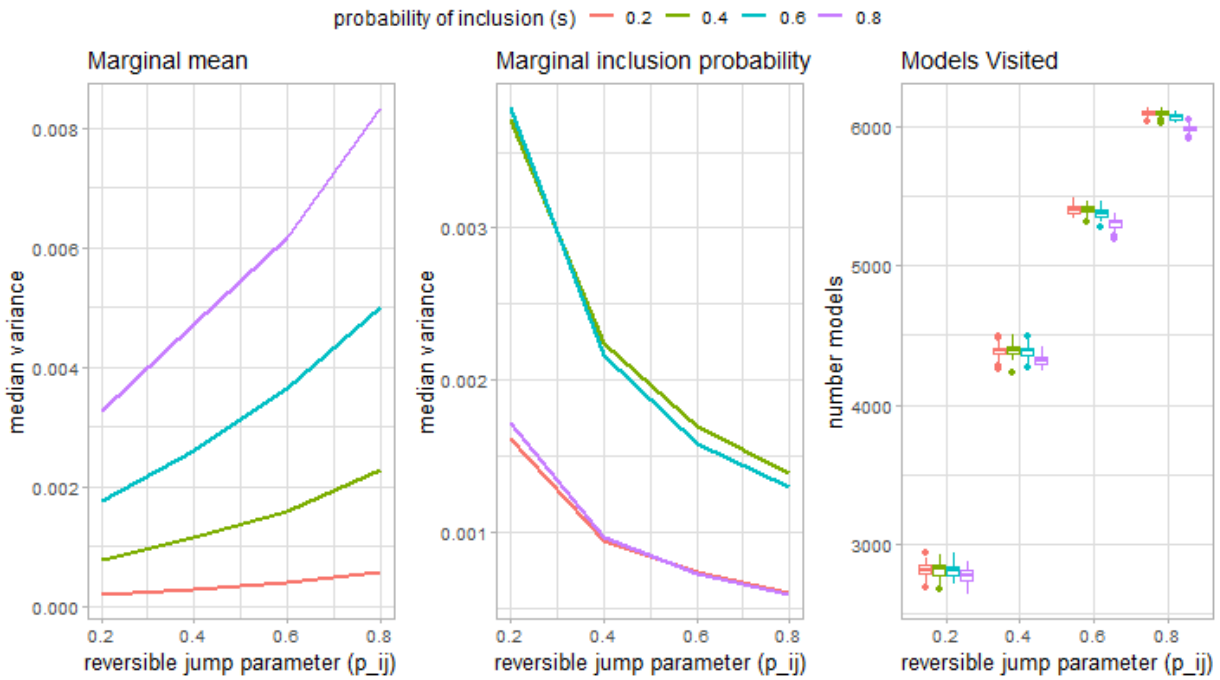

Figure 6: Effect of $p_{i,j}$ parameter on model and parameter exploration using Reversible Jump ZigZag. The target has $d = 50$ variables and $s \in \{0.2, 0.4, 0.6, 0.8\}$ is the probability of inclusion for each variable. Plots show the Monte Carlo variance against $p_{i,j}$.

0.9. The tables 5-8 compare the efficiency of Zig-Zag with these parameters relative to Zig-Zag with parameter $p_{i,j} = 0.6$. From these tables we see that $p_{i,j} = 0.6$ gives reasonable performance across all scenarios. We initialised at the null model for all simulations, causing the sampler to have an initial model exploration phase. Consequently lower values of $p_{i,j}$ perform worse as it takes longer to reach model space with high probability.

B.2 The refreshment parameter for RJ-BPS methods

Unlike Zig-Zag the BPS algorithm requires an additional refreshment event to ensure that it targets the invariant distribution. This refreshment event occurs with a constant rate

Table 5: Scenario 1 (pair of correlated variables): Relative efficiency (RE) for methods, against Zig-Zag with $p_{i,j} = 0.6$.

	$p_{i,j} = 0.1$		$p_{i,j} = 0.3$		$p_{i,j} = 0.7$		$p_{i,j} = 0.9$	
n, d	PI	Mean	PI	Mean	PI	Mean	PI	Mean
100, 100	0.68	0.28	1.36	0.65	0.89	1.11	0.70	1.09
200, 100	1.63	0.25	2.17	0.58	0.83	0.94	0.60	0.99
400, 100	1.57	0.32	2.09	0.72	0.81	1.03	0.58	1.05
800, 100	0.99	0.28	1.87	0.68	0.81	1.05	0.57	1.01
100, 200	0.38	0.43	0.63	0.61	0.73	0.77	0.80	0.94
200, 200	0.79	0.27	1.49	0.63	0.86	1.02	0.63	1.10
400, 200	0.84	0.30	1.60	0.71	0.83	1.14	0.62	1.22
800, 200	0.78	0.28	1.58	0.64	0.81	1.01	0.57	1.05
100, 400	0.27	0.33	0.62	0.68	1.16	1.13	1.20	1.10
200, 400	0.44	0.26	1.11	0.66	0.87	1.04	0.71	1.07
400, 400	0.30	0.26	0.93	0.72	0.87	1.06	0.71	1.11
800, 400	0.23	0.26	0.64	0.66	0.99	1.14	0.90	1.15

known as the refreshment rate λ_{ref}. For reversible jump algorithms there is an additional challenge of needing to specify how the refreshment rate should depend on the model we are currently exploring. Following from the results on how the optimal refreshment rate should scale with the dimension of the posterior (Bierkens et al. 2022, Deligiannidis et al. 2018) we will investigate whether and how the refreshment rate should scale with the size of the model we are exploring.

In answering this we need to be aware of how the speed of BPS scales with dimension, as this is different for the two versions of BPS that we consider. For BPS with velocities on the unit sphere, the speed is always 1 regardless of the dimension of the model. For BPS with a standard normal velocity distribution, the speed scales like $\sqrt{d}$, where d is the dimension of the model. While this does not change the process's dynamics this implies a different asymptotic scaling of the refreshment rate. A constant rate for velocities on the sphere would be equivalent to scaling the refreshment rate by $\sqrt{d}$ for velocities which are normally distribution.

Table 6: Scenario 2 (General correlation): Relative efficiency (RE) for methods, against Zig-Zag with $p_{i,j} = 0.6$.

	$p_{i,j} = 0.1$		$p_{i,j} = 0.3$		$p_{i,j} = 0.7$		$p_{i,j} = 0.9$	
n, d	PI	Mean	PI	Mean	PI	Mean	PI	Mean
100, 100	0.25	0.32	0.66	0.79	1.08	1.15	1.00	1.42
200, 100	0.26	0.29	0.77	0.74	0.92	0.98	0.92	1.09
400, 100	0.20	0.28	0.62	0.71	1.00	1.17	0.89	1.14
800, 100	0.15	0.21	0.54	0.60	1.00	0.89	1.00	1.00
100, 200	0.19	0.24	0.61	0.70	0.88	0.92	1.05	1.06
200, 200	0.17	0.22	0.58	0.61	1.04	0.85	1.08	0.99
400, 200	0.16	0.23	0.55	0.69	1.06	0.93	1.24	1.09
800, 200	0.16	0.21	0.46	0.57	1.07	0.98	1.38	0.99
100, 400	0.14	0.13	0.47	0.47	1.10	0.96	1.30	1.06
200, 400	0.16	0.19	0.49	0.58	1.11	1.01	1.29	1.02
400, 400	0.14	0.16	0.47	0.55	1.10	0.98	1.32	1.01
800, 400	0.16	0.16	0.46	0.59	1.20	1.06	1.43	1.01

The choice of optimal refreshment rate is a nuanced problem and may depend on the posterior functional of interest. Deligiannidis et al. (2018) find that, when using a normal velocity distribution, scaling the refreshment rate as $O(1)$ is better for low dimensional summaries; but scaling like $o(\sqrt{d})$ is better for posterior functionals such as the negative log posterior. They argue that the former, which include posterior means and variances, are often of primary interest.

We investigated how the refreshment rate should scale with d empirically. Again we consider the spike and slab prior distribution (12). We will empirically observe the process for $p_{i,j} = 0.6$ with $s = 0.5$ as we increase the dimensionality of the problem.

We consider two choices for refreshment parameter when using the normal velocity distribution: a constant $\lambda_{ref} = 1.424$ (in line with recommendations from Deligiannidis et al. 2018, for low-dimensional summaries) and $\lambda_{ref} = 1.424\sqrt{|M_\gamma|}$ (in line with the recommendations from Bierkens et al. 2022). The second choice is equivalent to a fixed refreshment rate for BPS with velocities uniformly distributed on the sphere. We show

Table 7: Scenario 3 (uncorrelated): Relative efficiency (RE) for methods, against Zig-Zag with $p_{i,j} = 0.6$.

	$p_{i,j} = 0.1$		$p_{i,j} = 0.3$		$p_{i,j} = 0.7$		$p_{i,j} = 0.9$	
n, d	PI	Mean	PI	Mean	PI	Mean	PI	Mean
100, 100	0.24	0.29	0.65	0.68	0.99	1.00	0.88	1.05
200, 100	0.23	0.28	0.68	0.72	0.93	1.10	0.90	1.11
400, 100	0.20	0.27	0.64	0.66	1.04	1.13	0.95	1.06
800, 100	0.18	0.24	0.61	0.62	1.03	1.03	0.99	1.01
100, 200	0.19	0.23	0.61	0.79	1.05	1.15	1.14	1.08
200, 200	0.16	0.25	0.49	0.58	1.09	1.10	1.06	1.12
400, 200	0.17	0.25	0.56	0.69	1.08	1.04	1.09	0.95
800, 200	0.15	0.23	0.49	0.66	1.06	0.99	1.22	0.98
100, 400	0.12	0.09	0.53	0.64	1.10	1.01	1.36	1.14
200, 400	0.19	0.23	0.52	0.56	1.08	0.94	1.19	0.90
400, 400	0.16	0.22	0.49	0.67	1.20	1.20	1.44	1.18
800, 400	0.16	0.18	0.46	0.54	1.17	1.00	1.42	1.04

plots of the trajectories for the first two components θ_1 and θ_2 for these different scalings in Figure 7.

We see empirically that asymptotically constant refreshment with velocities uniform on the sphere behaves the same as refreshing the Normal velocities with rate scaling as $\mathrm{O}(\sqrt{|M_\gamma|})$. For higher dimensions scaling the refreshment with the size of the active model induces random walk like behavior. For this reason we favor BPS with a fixed refreshment rate.

C Further correlation in logistic regression

Pairs plots and marginal KDEs from a typical run of the logistic regression example Scenario 4 with $n = 100$ and $d = 100$ are shown in Figure 8 for the coordinates $\theta_1, \theta_{51}, \theta_2$ and θ_{52}.

Simulation 4 was re-run with a higher correlation. This new scenario is specified with:

5. Multiple pairs of correlated variables with six active covariates:

Table 8: Scenario 4 (multiple correlated): Relative efficiency (RE) for methods, against Zig-Zag with $p_{i,j} = 0.6$.

	$p_{i,j} = 0.1$		$p_{i,j} = 0.3$		$p_{i,j} = 0.7$		$p_{i,j} = 0.9$	
n, d	PI	Mean	PI	Mean	PI	Mean	PI	Mean
100, 100	0.23	0.32	0.70	0.82	1.02	1.07	1.05	1.01
200, 100	0.24	0.27	0.59	0.55	1.01	1.06	0.99	1.10
400, 100	0.21	0.30	0.66	0.64	1.06	1.01	1.10	1.27
800, 100	0.14	0.26	0.49	0.60	1.01	1.00	1.09	1.00
100, 200	0.20	0.25	0.61	0.61	1.11	0.92	1.22	1.06
200, 200	0.20	0.31	0.58	0.66	1.09	0.90	1.13	1.05
400, 200	0.15	0.23	0.49	0.69	1.03	1.03	1.24	1.07
800, 200	0.16	0.22	0.52	0.67	1.22	1.16	1.24	1.16
100, 400	0.15	0.23	0.53	0.80	1.14	1.06	1.31	1.10
200, 400	0.13	0.14	0.49	0.57	1.11	1.00	1.32	1.23
400, 400	0.16	0.19	0.52	0.67	1.15	1.01	1.42	1.11
800, 400	0.14	0.13	0.47	0.56	1.19	1.07	1.45	1.12

$\boldsymbol{\theta} = (3, 3, -2, 3, 3, -2, 0, ..., 0)^T$ with $\boldsymbol{\Sigma}_{i+d/2,i} = \boldsymbol{\Sigma}_{i,i+d/2} = 0.99$ for $1 \leq i \leq 6$, $\boldsymbol{\Sigma}_{i,i} = 1$ and $\boldsymbol{\Sigma}_{i,j} = 0$ otherwise.

Pairs plots and marginal KDEs from a typical run are shown in Figure 9 and performance is described in Table 9.

D General implementation details

Inference in Bayesian model selection relies on expectations with respect to a posterior target distribution, $\pi(\boldsymbol{\theta}, \boldsymbol{\gamma})$. The parameters are $\boldsymbol{\theta}$ while $\boldsymbol{\gamma}$ is a vector which indexes the model with elements $\gamma_j = 1$ if the jth variable is included and $\gamma_j = 0$ otherwise. The posterior has the form

$$\pi(\boldsymbol{\theta}, \boldsymbol{\gamma}) \propto L(y^{1:n}|\boldsymbol{\theta}, \boldsymbol{\gamma})\pi_0(\boldsymbol{\theta}|\boldsymbol{\gamma})\pi_0(\boldsymbol{\gamma}),$$

Table 9: Scenario 5 (multiple highly correlated pairs): Relative efficiency (RE) for methods, against a Reversible Jump algorithm, for the marginal posterior means (Mean) and marginal posterior probabilities of inclusion (PI).

	ZigZag		BPS		Gibbs		RJ-HMC	
n, d	PI	Mean	PI	Mean	PI	Mean	PI	Mean
100, 100	0.37	0.93	0.38	**2.10**	0.84	0.82	0.01	0.05
200, 100	0.89	1.41	0.78	**2.02**	**2.18**	1.70	0.04	0.91
400, 100	1.61	2.69	1.40	**3.66**	**2.82**	3.07	0.79	1.50
800, 100	2.98	5.82	2.32	**6.99**	**4.25**	4.51	0.83	1.04
100, 200	0.60	1.13	1.00	**2.42**	**1.51**	1.94	0.02	0.67
200, 200	1.22	1.47	1.55	**2.45**	**2.19**	1.60	0.15	1.78
400, 200	4.42	6.79	3.87	**9.75**	**4.53**	4.67	1.19	2.26
800, 200	**10.05**	19.15	6.89	**19.81**	6.99	8.10	2.26	3.16
100, 400	1.20	1.69	2.01	**4.50**	**2.15**	1.64	0.01	2.22
200, 400	4.98	6.55	**6.18**	**11.51**	5.26	4.83	0.64	2.24
400, 400	9.90	11.50	**9.97**	**17.03**	7.56	7.30	2.74	4.94
800, 400	**24.08**	36.70	18.25	**38.97**	9.68	9.83	3.46	5.11

where $L(y^{1:n}|\boldsymbol{\theta}, \boldsymbol{\gamma})$ defines a likelihood function for observations $y^{1:n}$, $\pi_0(\boldsymbol{\theta}|\boldsymbol{\gamma})$ and $\pi_0(\boldsymbol{\gamma})$ denote prior distribution for $\boldsymbol{\theta}$ and $\boldsymbol{\gamma}$. We abuse notation writing $\boldsymbol{\theta}_{\boldsymbol{\gamma}}$ to denote the subvector of $\boldsymbol{\theta}$ with only the elements where $\gamma_j = 1$. Moreover, we write $\pi(\boldsymbol{\theta}_{\boldsymbol{\gamma}})$ for $\pi(\boldsymbol{\theta} \mid \boldsymbol{\gamma})$ where $\pi(\boldsymbol{\theta} \mid \boldsymbol{\gamma}) = 0$ whenever $|\theta_j| > 0$ with corresponding $\gamma_j = 0$.

When simulating from the reversible jump PDMP sampler there are two types of events: normal events for the PDMP sampler within a model $\boldsymbol{\gamma}$ and model jump events. The standard PDMP events are taken with respect to $\pi(\boldsymbol{\theta}|\boldsymbol{\gamma})$ so rates to sample are given using the usual Bouncy Particle Sampler or Zig-Zag rates on the conditioned model

$$\lambda^{BPS}(s) = \left(-\boldsymbol{v}_{\boldsymbol{\gamma}} \cdot \nabla_{\boldsymbol{\theta}_{\boldsymbol{\gamma}}} \log \pi(\boldsymbol{\theta}_{\boldsymbol{\gamma}} + s v_{\boldsymbol{\gamma}})\right)^+,$$
$$\lambda_i^{ZZ}(s) = \left(-v_i \nabla_{\theta_i} \log \pi(\boldsymbol{\theta}_{\boldsymbol{\gamma}} + s v_{\boldsymbol{\gamma}})\right)^+, \qquad \text{for } i \in \{i : \gamma_i = 1\}.$$

In practice to simulate these events we first bound the rates by a simple function, which for the examples we consider will be linear in time. We then simulate events from a Poisson process with this linear-in-time rate, which can be done exactly, and use thinning to generate the actual events in the PDMP. Derivations of the linear-in-time bounds on the rates that we use are now given, before we give the rates for jumps between models.

D.1 Rates for logistic and robust regression

Here we give details on simulating rates for the logistic regression example. We will slightly change notation, and write $\beta_{\boldsymbol{\gamma},\boldsymbol{\gamma}'}$ and $p_{\boldsymbol{\gamma},\boldsymbol{\gamma}'}$ for the rates and probabilities associated with the moves between models $\boldsymbol{\gamma}$ and $\boldsymbol{\gamma}'$. Taking a simple independent Gaussian prior the reintroduction rate simplifies to

$$\beta_{\boldsymbol{\gamma},\boldsymbol{\gamma}'} = p_{\boldsymbol{\gamma},\boldsymbol{\gamma}'} \frac{1}{\sqrt{2\pi\sigma^2}} \frac{w}{(1-w)},$$

where $\boldsymbol{\gamma}$ and $\boldsymbol{\gamma}'$ are defined as above, and we retain the convention that $\boldsymbol{\gamma}'$ is obtained from $\boldsymbol{\gamma}$ by removing one variable from the model. The standard PDMP rates are used for $\pi(\boldsymbol{\theta}|\boldsymbol{\gamma}) = \pi(\boldsymbol{\theta}_{\boldsymbol{\gamma}})$. So to simplify notation we will assume a fixed dimension and write $\boldsymbol{\theta}$ dropping the indexing with $\boldsymbol{\gamma}$. The log posterior for both logistic and robust regression can be written in the form

$$-\log \pi(\boldsymbol{\theta}) = \sum_{i=1}^{n} g(e_i) + \frac{\boldsymbol{\theta}^T\boldsymbol{\theta}}{2\sigma^2}. \tag{13}$$

For logistic regression $e_i = -\boldsymbol{x}_i^T\boldsymbol{\theta}$ and $g(e_i) = -\log\left(\frac{\exp(y_i)}{1+\exp(e_i)}\right)$, and for robust regression $e_i = y_i - \boldsymbol{x}_i^T\boldsymbol{\theta}$ and $g(e_i) = -\log\left(\exp(-\frac{1}{2}e_i^2) + \frac{1}{10}\exp(-\frac{1}{200}e_i^2)\right)$. We consider bounding the event rates for the Bouncy Particle Sampler and ZigZag below.

Bouncy Particle Sampler: Let $f(\boldsymbol{\theta} + t\boldsymbol{v}) = -\nabla_{\boldsymbol{\theta}} \log \pi(\boldsymbol{\theta} + t\boldsymbol{v})$ the event rate depends on the quantity

$$\langle \boldsymbol{v}, f(\boldsymbol{\theta} + t\boldsymbol{v}) \rangle = \langle v, \sum_{i=1}^{n} \boldsymbol{x}_i^T g'(e_i(t)) + \frac{1}{\sigma^2}(\boldsymbol{\theta} + t\boldsymbol{v}) \rangle.$$

where $g'(e_i(t))$ is the derivative of g evaluated evaluated at $e_i(t) = -\boldsymbol{x}_i^T(\boldsymbol{\theta} + tv)$ for logistic regression and $e_i(t) = y_i - \boldsymbol{x}_i^T(\boldsymbol{\theta} + tv)$ for robust regression. The in-time derivative of this quantity is

$$\frac{d}{dt}\langle \boldsymbol{v}, f(\boldsymbol{\theta} + t\boldsymbol{v}) \rangle = \langle \boldsymbol{v}, \sum_{i=1}^{n} \boldsymbol{x}_i^T g''(e_i(t)) \boldsymbol{x}_i^T \boldsymbol{v} + \frac{1}{\sigma^2}\boldsymbol{v} \rangle = \sum_{i=1}^{n} g''(e_i(t))(\boldsymbol{x}_i^T \boldsymbol{v})^2 + \frac{1}{\sigma^2}\boldsymbol{v}^T\boldsymbol{v}.$$

If the in-time derivative can be bounded by a constant we can simulate using linear rates. For logistic regression $g''(e_i) \leq \frac{1}{4}$ and for robust regression $g''(e_i) < 1$. The Bouncy Particle Sampler rate is bounded by the linear rate

$$\max(0, \langle \boldsymbol{v}, f(\boldsymbol{\theta} + t\boldsymbol{v}) \rangle) \leq \max\left(0, \langle \boldsymbol{v}, f(\boldsymbol{\theta}) \rangle\right) + t\left(\frac{1}{\sigma^2}\boldsymbol{v}^T\boldsymbol{v} + c\sum_{i=1}^{n}(\boldsymbol{x}_i^T\boldsymbol{v})^2\right),$$

where c is chosen according to the application. Inversion methods for thinning a Poisson process can be used to simulate the events (Bierkens et al. 2019).

ZigZag: Let $f(\boldsymbol{\theta} + t\boldsymbol{v}) = -\frac{\mathrm{d}}{\mathrm{d}\theta_{\mathrm{i}}} \log \pi(\boldsymbol{\theta} + t\boldsymbol{v})$ the event rate depends on the quantity

$$v_i f(\boldsymbol{\theta} + t\boldsymbol{v}) = v_i \sum_{i=1}^{n} x_{ij} g'(e_i(t)) + v_i \frac{1}{\sigma^2}(\theta_i + tv_i),$$

where $g'(e_i(t))$ is defined as in the BPS rate. The in-time derivative of this quantity is

$$\frac{d}{dt} v_i f(\boldsymbol{\theta} + t\boldsymbol{v}) = v_i \sum_{i=1}^{n} x_{ij} g''(e_i(t)) \boldsymbol{x}_i^T \boldsymbol{v} + v_i^2 \frac{1}{\sigma^2} = \sum_{i=1}^{n} g''(e_i(t)) x_{ij} v_i \boldsymbol{x}_i^T \boldsymbol{v} + v_i^2 \frac{1}{\sigma^2}.$$

Using the same method as before the ZigZag rate is bounded by the linear rate

$$\max(0, v_i f(\boldsymbol{\theta} + t\boldsymbol{v})) \leq \max(0, v_i f(\boldsymbol{\theta})) + t \left(v_i^2 \frac{1}{\sigma^2} + \sum_{i=1}^{n} |c x_{ij} v_i \boldsymbol{x}_i^T \boldsymbol{v}| \right),$$

where c is chosen according to the application.

D.2 Control variates for multiple models

In this section we note a choice of control variates that can be used across a wide span of model space. We will describe the idea in context of the logistic and robust regression examples from the previous section for the Zig-Zag sampler. Similar arguments follow for constructing control variates for BPS.

We first describe the general form for constructing control variates within a single model. Suppose the log posterior follows the form of (13). The Zig-Zag sampler has within model rate $\lambda_i(\boldsymbol{\theta}, \boldsymbol{v}) = \max(0, v_i f(\boldsymbol{\theta}))$ where,

$$v_i f(\boldsymbol{\theta}) = v_i \sum_{j=1}^{n} x_{ij} g'(e_j) + v_i \frac{1}{\sigma^2} \theta_i,$$

and $e_j = \boldsymbol{x}_j^T \boldsymbol{\theta}$. Let $\boldsymbol{\theta}^*$ be a chosen control variate and $e_j^* = \boldsymbol{x}_i^T \boldsymbol{\theta}^*$ for $j = 1, ..., n$. Define

E^j for a random index $j \sim \text{Uniform}(1, 2, ..., n)$ as the variable

$$E^j = nv_i(x_{ij}g'(e_j) - x_{ij}g'(e_j^*)) + v_i \sum_{j=1}^{n} x_{ij}g'(e_j^*) + v_i \frac{1}{\sigma^2}\theta_i, \tag{14}$$

where expected value of E^j is $\mathbb{E}[E^j] = v_i f(\boldsymbol{\theta})$. Zig-Zag with control variates works by sampling with stochastic rate $\tilde{\lambda}_i(\boldsymbol{\theta}, \boldsymbol{v}) = \max(0, E^j)$. To facilitate thinning suppose that the derivatives of g' are globally and uniformly Lipschitz with constant C, so,

$$\begin{aligned}
|(x_{ij}g'(e_j) - x_{ij}g'(e_j^*))| &\leq \|e_j - e_j^*\|_2 |x_{ij}| C \\
&\leq \|\boldsymbol{x}_j^T\boldsymbol{\theta} - \boldsymbol{x}_j^T\boldsymbol{\theta}^*\|_2 |x_{ij}| C \\
&\leq \|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|_2 \|\boldsymbol{x}_j^T\|_2 |x_{ij}| C
\end{aligned}$$

for $j = 1, ..., n$. Let $C_i = \max_{j=1,...,n} \|\boldsymbol{x}_j^T\|_2 |x_{ij}| C$, thinning may be implemented using the upperbound,

$$\begin{aligned}
\tilde{\lambda}_i(\boldsymbol{\theta}, \boldsymbol{v}) &= \max(0, nv_i(x_{ij}g'(e_j) - x_{ij}g'(e_j^*)) + v_i \sum_{j=1}^{n} x_{ij}g'(e_j^*) + v_i \frac{1}{\sigma^2}\theta_i) \\
&\leq \max\left(0, \|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|_2 C_i n + v_i \sum_{j=1}^{n} x_{ij}g'(e_j^*) + v_i \frac{1}{\sigma^2}\theta_i\right).
\end{aligned}$$

Over the trajectory $\boldsymbol{\theta}(t) = \boldsymbol{\theta} + t\boldsymbol{v}$ the rate is bounded by

$$\tilde{\lambda}_i(\boldsymbol{\theta}(t), \boldsymbol{v}) \leq \max\left(0, \left(\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|_2 + t\|\boldsymbol{v}\|_2\right) C_i n + v_i \sum_{j=1}^{n} x_{ij}g'(e_j^*) + v_i \frac{1}{\sigma^2}\theta_i\right).$$

This requires an $O(n)$ evaluation of $\sum_{j=1}^{n} x_{ij}g'(e_j^*)$ and calculation of $C_i = \max_{j=1,...,n} \|\boldsymbol{x}_j^T\|_2 |x_{ij}| C$ which may be computed once prior to running the sampler.

To construct control variates that may be used when the sampler is not in model $\boldsymbol{\gamma}$ we define nested control variates. A nested control variate may be constructed for any model $\boldsymbol{\gamma}'$ where $|\boldsymbol{\gamma}| \leq |\boldsymbol{\gamma}'|$. For the model $\boldsymbol{\gamma}'$ a nested control variate may be constructed from $\boldsymbol{\theta}_\gamma$ by setting θ_j equal to the corresponding value of $\boldsymbol{\theta}_\gamma$ and padding the remainder of the vector with zeros. That is $\boldsymbol{\theta}_{\gamma'}$ has elements θ_j^* when $\gamma_j = \gamma'_j = 1$ and 0 when $\gamma_j = 0$ and $\gamma'_j = 1$. For such a control variate we have $e_j^* = \sum_{i:\gamma_i=1} x_{ij}\theta_i^* = \sum_{i:\gamma'_i=1} x_{ij}\theta_i^*$ and the computation of $v_i \sum_{j=1}^n x_{ij} g'(e_j^*)$ may be reused. The rates for the model $\boldsymbol{\gamma}'$ take the same form as (14) and may be implemented using the same thinning procedure.

D.3 Rates of jumps between models

Model jump events occur when a parameter, θ_i, hits a hyper-plane $\theta_i = 0$ and with probability $p_{\boldsymbol{\gamma},\boldsymbol{\gamma}'}$ we jump to model $\boldsymbol{\gamma}'$ where $\gamma'_i = 0$. The other type of model jump event occurs when a variable is reintroduced. For each of the deactivated variables ($\gamma_i = 0$), we simulate a time to reintroduce the variable. The rate to reintroduce the variable in the Bayesian inference problem is

$$\beta_{\boldsymbol{\gamma},\boldsymbol{\gamma}'} = p_{\boldsymbol{\gamma},\boldsymbol{\gamma}'} \frac{L(y^{1:n}|\boldsymbol{\theta}_{\boldsymbol{\gamma}})\pi_0(\boldsymbol{\theta}_{\boldsymbol{\gamma}})\pi_0(\boldsymbol{\gamma})}{L(y^{1:n}|\boldsymbol{\theta}_{\boldsymbol{\gamma}'})\pi_0(\boldsymbol{\theta}_{\boldsymbol{\gamma}'})\pi_0(\boldsymbol{\gamma}')},$$

where often computational savings are possible since the reintroduced variable will be zero $\theta_i = 0$ and it is often the case that $L(y^{1:n}|\boldsymbol{\theta}_{\boldsymbol{\gamma}'}) = L(y^{1:n}|\boldsymbol{\theta}_{\boldsymbol{\gamma}})$. In these cases the rate to reintroduce a variable will only depend on the choice of prior.

Both examples we consider had a Gaussian spike and slab prior, of the form

$$\begin{aligned}\boldsymbol{\theta}_{\boldsymbol{\gamma}} &\sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\gamma}}, \boldsymbol{\Sigma}_{\boldsymbol{\gamma}}) \\ \boldsymbol{\gamma} &\sim w^{\sum_{j=1}^d \gamma_j}(1-w)^{d-\sum_{j=1}^d \gamma_j},\end{aligned}$$

for a fixed w. The rate to reintroduce the ith variable, jumping from model $\boldsymbol{\gamma}$ to $\boldsymbol{\gamma}'$ where $\boldsymbol{\gamma}'_{-i} = \boldsymbol{\gamma}_{-i}$ with $\gamma'_i = 1$ and $\gamma_i = 0$ is given by

$$\beta_{\boldsymbol{\gamma},\boldsymbol{\gamma}'} = p_{\boldsymbol{\gamma},\boldsymbol{\gamma}'} \frac{\pi(\boldsymbol{\theta}_{\boldsymbol{\gamma}})}{\pi(\boldsymbol{\theta}_{\boldsymbol{\gamma}'})} \frac{\pi(\boldsymbol{\gamma})}{\pi(\boldsymbol{\gamma}')} = \frac{\pi(\boldsymbol{\theta}_{\boldsymbol{\gamma}})}{\pi(\boldsymbol{\theta}_{\boldsymbol{\gamma}'})} \frac{w}{(1-w)}.$$

Denoting $V_{\boldsymbol{\gamma}} = \boldsymbol{\Sigma}_{\boldsymbol{\gamma}}^{-1}$, the ratio simplifies as,

$$\frac{\pi(\boldsymbol{\theta}_{\boldsymbol{\gamma}})}{\pi(\boldsymbol{\theta}_{\boldsymbol{\gamma}'})} = \frac{|V_{\boldsymbol{\gamma}}|^{1/2} \exp\left(-\frac{1}{2}(\boldsymbol{\theta}_{\boldsymbol{\gamma}} - \mu_{\boldsymbol{\gamma}})^T V_{\boldsymbol{\gamma}} (\boldsymbol{\theta}_{\boldsymbol{\gamma}} - \mu_{\boldsymbol{\gamma}})\right)}{\sqrt{2\pi}|V_{\boldsymbol{\gamma}'}|^{1/2} \exp\left(-\frac{1}{2}(\boldsymbol{\theta}_{\boldsymbol{\gamma}'} - \mu_{\boldsymbol{\gamma}'})^T V_{\boldsymbol{\gamma}'} (\boldsymbol{\theta}_{\boldsymbol{\gamma}'} - \mu_{\boldsymbol{\gamma}'})\right)}.$$

In our examples the prior is independent across components and this ratio simplifies to a constant. As the prior mean is 0 and, if we denote the prior variance for θ_i for any active covariate i as σ^2, we have

$$\beta_{\boldsymbol{\gamma},\boldsymbol{\gamma}'} = p_{\boldsymbol{\gamma},\boldsymbol{\gamma}'} \frac{1}{\sqrt{2\pi\sigma^2}} \frac{w}{(1-w)}.$$

D.4 Pólya-Gamma Gibbs sampling for logistic regression

The Polya-Gamma Gibbs sampling approach is an auxiliary variable approach for Bayesian Logistic regression. A Polya-Gamma random variable $\omega \sim \mathcal{PG}(b, 0)$, $b > 0$, with probability density $p(\omega)$ has the property (Polson et al. 2013) that for any $\psi \in \mathbb{R}$ and $a \in \mathbb{R}$

$$\frac{\exp(\psi)^a}{(1+\exp(\psi))^b} = 2^{-b} \exp\left(\left(a - \frac{b}{2}\right)\psi\right) \int_0^\infty \exp(-\omega\frac{\psi^2}{2}) p(\omega) \mathrm{d}\omega.$$

Thus the implied conditional distribution for ψ, given auxiliary variable ω, is Gaussian. The advantage of this approach is that when updating the model $\boldsymbol{\gamma}$ we can integrate over the parameters $\boldsymbol{\theta}$ yielding much more efficient moves. The updates for the collapsed Gibbs sampling procedure follow the form:

(1) sample $\boldsymbol{\gamma} \sim \boldsymbol{\gamma} \mid \boldsymbol{\omega}$;

(2) sample $\boldsymbol{\theta} \sim \boldsymbol{\theta} \mid \boldsymbol{\omega}, \boldsymbol{\gamma}$;

(3) sample $\boldsymbol{\omega} \sim \boldsymbol{\omega} \mid \boldsymbol{\theta}, \boldsymbol{\gamma}$.

Simulation step 1.

Let $\tilde{\pi}(\boldsymbol{\gamma} \mid \boldsymbol{\omega})$ be a density proportional to $\pi(\boldsymbol{\gamma} \mid \boldsymbol{\omega})$ such that

$$\begin{aligned}\tilde{\pi}(\boldsymbol{\gamma} \mid \boldsymbol{\omega}) &= \pi_0(\boldsymbol{\gamma}) \int_{\boldsymbol{\theta}_{\boldsymbol{\gamma}}} L(y^{1:n}|\boldsymbol{\theta}_{\boldsymbol{\gamma}}, \boldsymbol{\omega}) \pi_0(\boldsymbol{\theta}_{\boldsymbol{\gamma}}) \mathrm{d}\boldsymbol{\theta}_{\boldsymbol{\gamma}} \\ &= \frac{\pi_0(\boldsymbol{\gamma})}{\sqrt{\det(2\pi\sigma^2 \boldsymbol{I}_{\boldsymbol{\gamma}})}} \int_{\boldsymbol{\theta}_{\boldsymbol{\gamma}}} \prod_{i=1}^{n} \exp\left((y_i - 0.5)(\boldsymbol{X}_{\boldsymbol{\gamma}}\boldsymbol{\theta}_{\boldsymbol{\gamma}})_i - \frac{\omega_i}{2}((\boldsymbol{X}_{\boldsymbol{\gamma}}\boldsymbol{\theta}_{\boldsymbol{\gamma}})_i)^2 - \frac{1}{2\sigma^2}\boldsymbol{\theta}_{\boldsymbol{\gamma}}^T\boldsymbol{\theta}_{\boldsymbol{\gamma}} \right) \mathrm{d}\boldsymbol{\theta}_{\boldsymbol{\gamma}} \\ &= \pi_0(\boldsymbol{\gamma}) \sqrt{\frac{\det(2\pi \boldsymbol{V}_{\boldsymbol{\gamma}})}{\det(2\pi\sigma^2 \boldsymbol{I}_{\boldsymbol{\gamma}})}} \exp(\frac{1}{2}\kappa^T \boldsymbol{X}_{\boldsymbol{\gamma}} \boldsymbol{V}_{\boldsymbol{\gamma}} \boldsymbol{X}_{\boldsymbol{\gamma}}^T \kappa),\end{aligned}$$

where $(\boldsymbol{X}_{\boldsymbol{\gamma}}\boldsymbol{\theta}_{\boldsymbol{\gamma}})_i$ denotes the ith element of $\boldsymbol{X}_{\boldsymbol{\gamma}}\boldsymbol{\theta}_{\boldsymbol{\gamma}}$, the matrix $\boldsymbol{V}_{\boldsymbol{\gamma}} = \left(\boldsymbol{X}_{\boldsymbol{\gamma}}^T \Omega \boldsymbol{X}_{\boldsymbol{\gamma}} + \frac{1}{\sigma^2}\boldsymbol{I}_{\boldsymbol{\gamma}}\right)^{-1}$, the column vector $\kappa = y^{1:n} - 0.5$ and $\Omega = \mathrm{diag}(\omega_1, ..., \omega_n)$.

The update for $\boldsymbol{\gamma}$ is taken by updating component-wise from the conditionals $\gamma_j \mid \boldsymbol{\gamma}_{(-j)}, \boldsymbol{\omega}$ where $\boldsymbol{\gamma}_{(-j)} = (\gamma_1, ..., \gamma_{j-1}, \gamma_{j+1}, ..., \gamma_d)$. Such a proposal can be implemented using the relationship (Chipman et al. 2001)

$$Pr(\gamma_j = 1 \mid \boldsymbol{\gamma}_{(-j)}, \boldsymbol{\omega}) = \frac{\tilde{\pi}(\gamma_j = 1 \mid \boldsymbol{\gamma}_{(-j)}, \boldsymbol{\omega})}{\tilde{\pi}(\gamma_j = 0 \mid \boldsymbol{\gamma}_{(-j)}, \boldsymbol{\omega})} \left(1 + \frac{\tilde{\pi}(\gamma_j = 1 \mid \boldsymbol{\gamma}_{(-j)}, \boldsymbol{\omega})}{\tilde{\pi}(\gamma_j = 0 \mid \boldsymbol{\gamma}_{(-j)}, \boldsymbol{\omega})}\right)^{-1}.$$

Simulation step 2.

The conditional for $\boldsymbol{\theta}_{\boldsymbol{\gamma}}$ is

$$\boldsymbol{\theta}_{\boldsymbol{\gamma}} | \boldsymbol{\omega}, \boldsymbol{\gamma} \sim \mathcal{N}(\boldsymbol{m}_{\boldsymbol{\gamma}}, \boldsymbol{\Sigma}_{\gamma}),$$

where $\boldsymbol{\Sigma}_{\boldsymbol{\gamma}} = (\boldsymbol{X}_{\boldsymbol{\gamma}}^T \Omega \boldsymbol{X}_{\boldsymbol{\gamma}} + \frac{1}{\sigma^2}\boldsymbol{I}_{\boldsymbol{\gamma}})$ and $\boldsymbol{m}_{\boldsymbol{\gamma}} = \boldsymbol{\Sigma}_{\boldsymbol{\gamma}} \boldsymbol{X}_{\boldsymbol{\gamma}} \kappa$.

Simulation step 3.

The conditional for $\boldsymbol{\omega}$ is $\omega_i|\boldsymbol{\theta}, \boldsymbol{\gamma} \sim \mathcal{PG}\left(0, (\boldsymbol{X}_{\boldsymbol{\gamma}}\boldsymbol{\theta}_{\boldsymbol{\gamma}})_i\right)$ for $i = 1, ..., n$.

D.5 Reversible Jump HMC

Our reversible jump HMC competitor consists of two moves: model-jump moves and within-model moves. With probability p_m a within-model move takes place otherwise a model-jump takes place. A within-model move proposes a new value of $\boldsymbol{\theta}$ conditional on the current model using a standard HMC proposal. A model-jump proposal updates the model space by adding new variables in "birth" moves or deleting variables in "death" moves. Let S denote the set of selected variables and N be the set of non-selected variables. We considered two approaches for model space exploration. In the first approach birth or death moves are performed by randomly selecting a variable from either N or S, respectively, and switching them to the other set. Specifically a birth move will randomly select from N and reintroduce the variable placing it into S. The alternative approach consists of iterating through all variables in a deterministic order and proposing removing any include variable or adding in any removed variable. When a birth move occurs in either approach the value of the reintroduced variable is proposed using a univariate random Normal proposal centred at zero with variance one. After experimentation we found that the second approach had superior performance and this is the method used in the simulations.

E Computation of relative efficiencies in Section 5.1

In order to compute the relative efficiency we need an estimate of the statistical efficiency (9). We estimate this quantity using a reference estimate q from an independent 6-hour run of the Gibbs sampling method for each combination of n, d and Scenario in Tables 1-3 and the results of Figure 2. For the results in Tables 1-3 the quantities of interest

are the estimation of the posterior marginal inclusion probabilities $\pi(|\theta_j| > 0)$ (PPI) and the marginal posterior means $\mathbb{E}[\theta_j]$ (Mean). These two quantities allowed us to see how efficient the sampler was in terms of exploring both the parameter and model space. For the simulations in the subsampling comparison (Figure 2) the quantity of interest was the posterior mean conditioned on being in model $\boldsymbol{\gamma} = (1, 1, 0, 0, ..., 0)$). We estimate the mean square error of these terms by running 100 independent runs of each algorithm and comparing to the corresponding long Gibbs run. Methods used in Tables 1-4 were initialised at zero with no variables included in the model. For the subsampling comparison, methods were initialised at the location of the control variate (the maximum a posterior estimate using the true nonzero variables $\boldsymbol{\gamma}$). For each algorithm in the simulations of Tables 1-4 we use a computational budget of 10^6 iterations with a maximum run time of 2 minutes. For the simulations in the subsampling comparison we used a computational budget of 10^6 iterations with a maximum run time of 15 seconds. Algorithms were then compared on the basis of relative computational efficiency using RE or relative efficiency per iteration using RSE. An iteration for the Gibbs sampler is considered to be a full update of all parameters (i.e. one run of all steps in Section D.4) whereas an iteration of the PDMP methods is considered to be one simulated event time.

F Reversible Jump Algorithm for ZigZag

Pseudo-code for the reversible jump version of ZigZag is given in Algorithm 3. Lines 2 to 15 calculate the time of each possible event for each variable. If variable i is disabled, the only event that is possible is to add variable i to the model. If variable i is enabled, then we can potentially remove variable i at the next time that $\boldsymbol{\theta}^i = 0$, and we can flip the velocity $\boldsymbol{v}^i$ with the usual ZigZag rate. Lines 16 to 18 calculates the time at which the the first flip, remove or addition event occurs, and the variable associated with each of these.

Lines 19-26 consider which of the three types of event occurs first, and updates the state according to the event type and variable affected. Finally in Line 37 we update the time.

Algorithm 3: ZigZag algorithm for variable selection

Inputs: the initial position, $\boldsymbol{\theta}$, velocity, $\boldsymbol{v}$, and model γ; and t_{max}; **while** $t < t_{max}$*:* **do**
  for *each* $i \leq d$ **do**
    if $\gamma_i = 0$ **then**
      `/* variable` i `is disabled */`
      Set $t_i^{flip} = +\infty$, $t_i^{remove} = +\infty$;
      `/* Poisson process to enable variable` i `*/`
      Let $\gamma' = \gamma$ and set $\gamma'_i = 1$;
      Let t_i^{add} the event time associated to the Poisson rate $\beta_{\gamma',\gamma} = p_{\gamma',\gamma}\frac{\pi_\gamma(\boldsymbol{\theta}+t\boldsymbol{v})}{\pi_{\gamma'}(\boldsymbol{\theta}+t\boldsymbol{v})}$
    else
      `/* variable` i `is enabled */`
      Set $t_i^{add} = +\infty$;
      Set t_i^{flip} the event time associated to the Poisson rate
        $\lambda_i(t) = \max\left\{0, -\boldsymbol{v}^i \frac{\partial \log \pi}{\partial \boldsymbol{\theta}^i}(\boldsymbol{\theta} + t\boldsymbol{v})\right\}$.;
      Let $t_i^{remove} = -\boldsymbol{\theta}^i/\boldsymbol{v}^i$ the time until intersection with hyperplane $\boldsymbol{\theta}^i = 0$;
      if $t_i^{remove} < 0$ **then**
        Set $t_i^{remove} = +\infty$;
      end
    end
  end
  Set $i_{flip} = \arg\min(t_i^{flip})$, $t_{flip} = \min(t_i^{flip})$;
  Set $i_{remove}, = \arg\min(t_i^{remove})$, $t_{remove} = \min(t_i^{remove})$;
  Set $i_{add} = \arg\min(t_i^{add})$, $t_{add} = \min(t_i^{add})$;
  `/* Act on whichever happens first: a flip, adding a variable or removing a variable */`
  if $t_{flip} < t_{remove}$ *and* $t_{flip} < t_{add}$ **then**
    Set $\boldsymbol{\theta} = \boldsymbol{\theta} + \boldsymbol{v}t_{flip}$;
    Set $v^{i_{flip}} = -v^{i_{flip}}$;
  else
    if $t_{remove} < t_{add}$ **then**
      Let $\gamma' = \gamma$ and set $\gamma'_{i_{remove}} = 0$;
      Let $u \sim unif(0,1)$;
      if $u < p_{\gamma,\gamma'}$ **then**
        Set $\boldsymbol{\theta} = \boldsymbol{\theta} + \boldsymbol{v}t_{remove}$;
        Set $v^{i_{remove}} = 0$;
        Set $\gamma_{i_{remove}} = 0$;
      end
    else
      Set $\boldsymbol{\theta} = \boldsymbol{\theta} + \boldsymbol{v}t_{add}$;
      Set $v^{i_{add}}$ to $+1$ or -1 with probability $1/2$;
      Set $\gamma_{i_{add}} = 1$;
    end
  end
  Set $t = t + \min(t_{flip}, t_{remove}, t_{add})$;
end

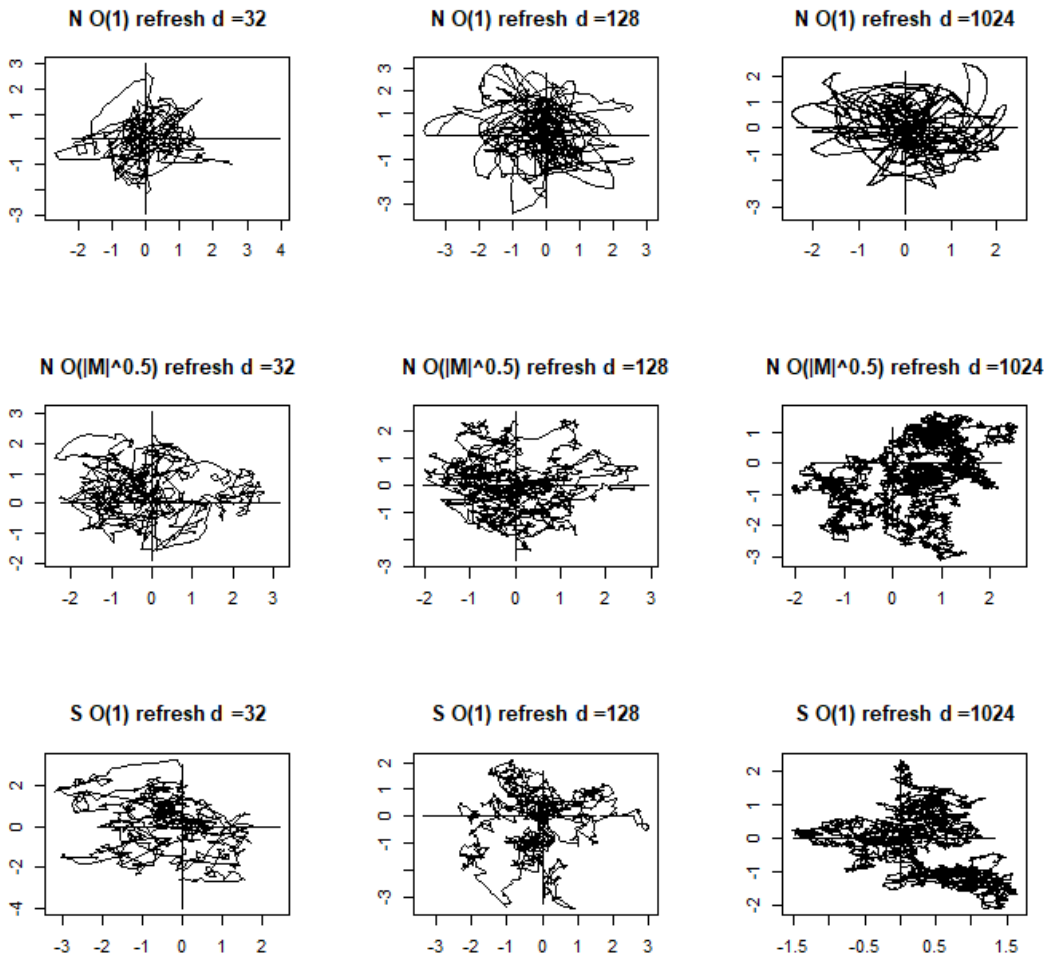

Figure 7: Trace plots for BPS for increasing dimension and two different scalings of the refresh rate for velocities. Top two rows: BPS with normal distribution of velocities (N). Bottom row: BPS with uniform distribution on the sphere (S).

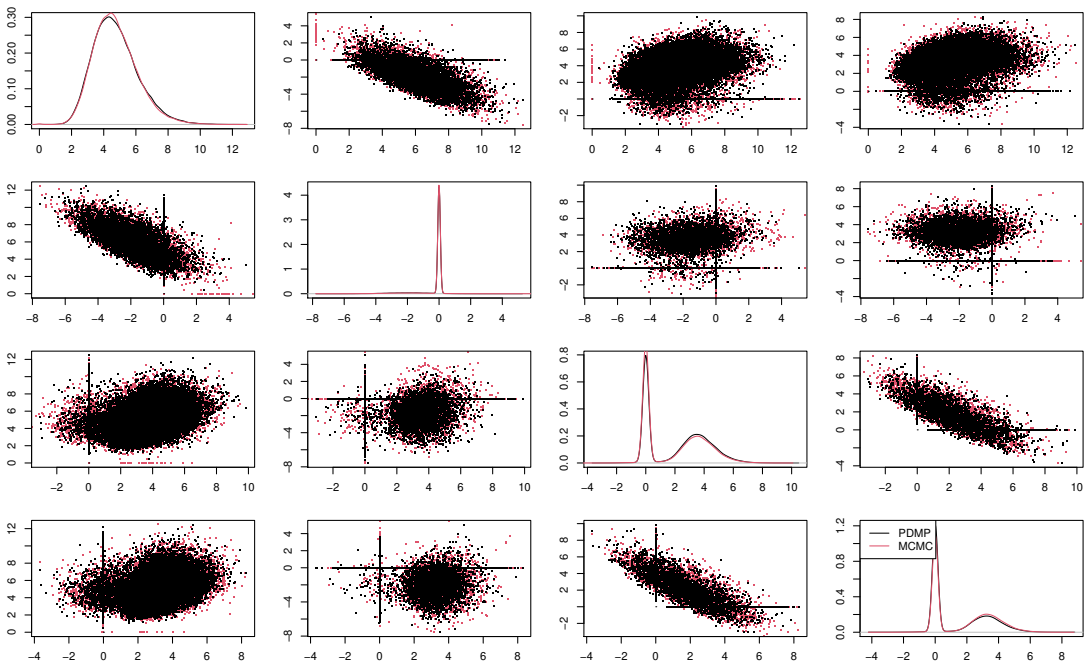

Figure 8: Pairs plots of $\theta_1, \theta_{51}, \theta_2$ and θ_{52} for Simulation 4 using Zig-Zag for the PDMP sampling (black lines and dots) and a Gibbs sampler for the MCMC samples (red lines and dots). Both samplers were run for the same computational budget and the Zig-Zag dynamics were discretised to the same sample size as the Gibbs sampler.

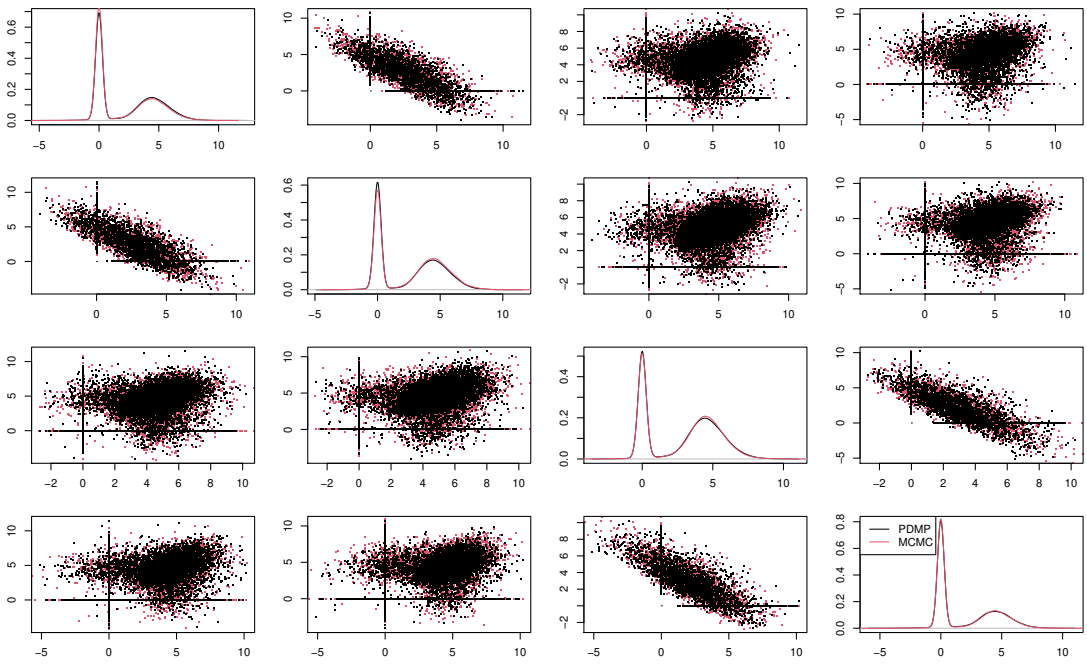

Figure 9: Pairs plots of $\theta_1, \theta_{51}, \theta_2$ and θ_{52} for Simulation 5 using Zig-Zag for the PDMP sampling (black lines and dots) and a Gibbs sampler for the MCMC samples (red lines and dots). Both samplers were run for the same computational budget and the Zig-Zag dynamics were discretised to the same sample size as the Gibbs sampler.

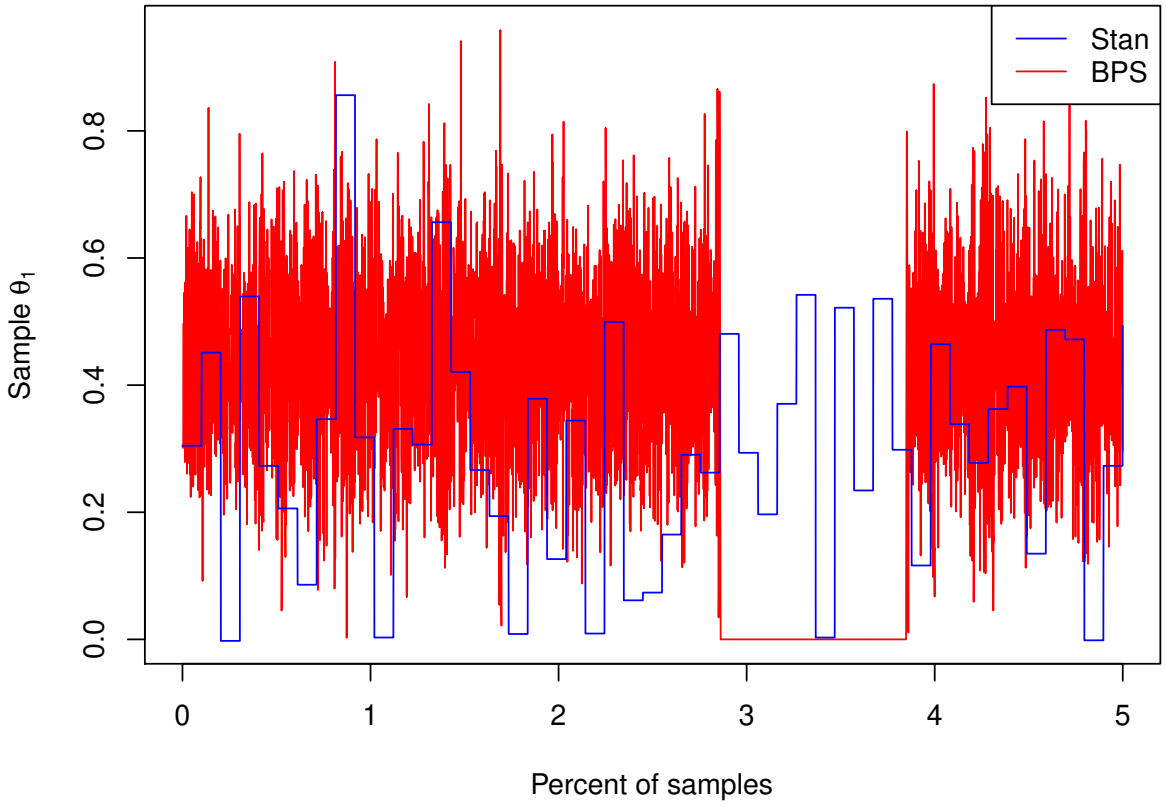

Figure 10: Simulated paths (as in Bouchard-Côté et al. (2018)) comparing the dynamics of the PDMP sampler with those of Stan. These dynamics correspond to the robust regression simulation from Figure 3. Both algorithms were run for the same computation time and the simulated trajectories correspond to the first five percent of samples from each method.