Continuously-Tempered PDMP samplers

Anonymous Author(s)
Affiliation
Address
email

Abstract

New sampling algorithms based on simulating continuous-time stochastic processes called piece-wise deterministic Markov processes (PDMPs) have shown considerable promise. However, these methods can struggle to sample from multimodal or heavy-tailed distributions. We show how tempering ideas can improve the mixing of PDMPs in such cases. We introduce an extended distribution defined over the state of the posterior distribution and an inverse temperature, which interpolates between a tractable distribution when the inverse temperature is 0 and the posterior when the inverse temperature is 1. The marginal distribution of the inverse temperature is a mixture of a continuous distribution on $[0, 1)$ and a point mass at 1: which means that we obtain samples when the inverse temperature is 1, and these are draws from the posterior, but sampling algorithms will also explore distributions at lower temperatures which will improve mixing. We show how PDMPs, and particularly the Zig-Zag sampler, can be implemented to sample from such an extended distribution. The resulting algorithm is easy to implement and we show empirically that it can outperform existing PDMP-based samplers on challenging multimodal posteriors.

1 Introduction

Recently there has been considerable interest in developing sampling algorithms based on simulating continuous time stochastic processes called a piecewise deterministic processes (PDMPs) Davis [1993]. These sampling algorithms are qualitatively similar to Hamiltonian Monte Carlo algorithms Neal [2011]. To simulate from some targst distribution $\pi(\boldsymbol{x})$, they work with an augmented state-space $(\boldsymbol{x}, \boldsymbol{v})$, where the $\boldsymbol{x}$ component can be viewed as position, and the $\boldsymbol{v}$ component as a velocity. The motivation for this is that these processes will encourage exploration of $\pi(\boldsymbol{x})$ by simulating continuous velocity paths in between random event times at which the velocity changes. Examples of such sampling algorithms include the Bouncy Particle Sampler Peters and de With [2012], Bouchard-Côté et al. [2018], the Zig-Zag algorithm Bierkens et al. [2019], the Coordinate Sampler Wu and Robert [2020] and the Boomerang Sampler Bierkens et al. [2020]. See Fearnhead et al. [2018] for an overview of these methods. Importantly, it is simple to write down the event rate, and how the velocity should change at each event, to ensure these PDMPs have π as their invariant distribution. These depend on π only through the gradient of $\log \pi$, and thus π only needs to be known up to a constant of proportionality.

These PDMP samplers have a number of advantages, including non-reversible dynamics (which is known to improve mixing relative to reversible processes Diaconis et al. [2000], Bierkens [2016]), and the ability to reduce computation-per-iteration by either leveraging sparsity structure in the model Bouchard-Côté et al. [2018], Sutton and Fearnhead [2021] or using only sub-samples of the data to approximate the log-likelihood at each iteration (whilst still guaranteeing sampling from the target Bierkens et al. [2019]). However, like other MCMC algorithms, particularly those that use gradient

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

information, these PDMP samplers can struggle to mix for multi-modal target distributions, or for heavy-tailed targets Vasdekis and Roberts [2021].

One of the more successful techniques for enabling an MCMC algorithm to sample from challenging, e.g. multi-modal, target distributions is to use tempering. There are various forms of tempering, but each is based on defining either a discrete set or continuum of distributions that interpolate between a distribution that is simple to sample from (viewed as at high temperature) and the target distribution (at a low temperature). The idea is that allowing mixing across this set will improve mixing, as moving between modes will be easier for the distributions at higher temperatures. Examples of such algorithms include parallel tempering Swendsen and Wang [1986], simulated tempering Marinari and Parisi [1992], and continuous tempering Graham and Storkey [2017].

In this paper we show how continuous tempering ideas can be used with PDMP samplers. We have chosen *continuous* tempering, as opposed to the alternative tempering approaches, as it is the method that can most naturally benefit from the continuous-time nature of PDMP dynamics. To the best of our knowledge, this is the first attempt at using tempering ideas to improve PDMP samplers. The general idea is to define a joint distribution on the state of the PDMP, $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{v})$, and the inverse temperature, β. The target distribution of interest is the $\boldsymbol{x}$-marginal of this joint distribution when $\beta = 1$. We define the joint distribution so that it has a point mass at $\beta = 1$ – thus simulating from it will lead to a proportion of the resulting samples being from the target. We then use a PDMP sampler to simulate from this joint distribution. Constructing the appropriate dynamics of the PDMP sampler is non-trivial in this case as we have to deal with its behaviour as it transitions from the continuous distribution on $\beta \in [0, 1)$ and the point mass at $\beta = 1$. We use recent ideas for PDMP samplers with discontinuities Chevallier et al. [2021, 2020] to solve this challenge.

While we express ideas based on and similar to Graham and Storkey [2017], a key difference is the inclusion of a point-mass at $\beta = 1$, this means we can obtain samples from $\pi(\boldsymbol{x})$ rather than having to resort to importance sampling to correct samples drawn at different temperatures. It is easy to introduce a point mass into the dynamics of the PDMP sampler due to its continuous sample paths: we simulate paths for $\beta \in [0, 1)$ until the process hits $\beta = 1$ – we then simulate paths with β fixed to 1 for an exponential period of time before returning to $\beta \in [0, 1)$. By comparison, using a point-mass within a Hamiltonian Monte Carlo sampler is difficult as the discretised sample paths will not hit $\beta = 1$ with probability 1. Yao et al. [2020] consider an *indirect* approach to sampling with a pointmass at $\beta = 1$ using a continuous link function. Our approach is distinct and more direct, allowing one to exploit the unique advantages of PDMP samplers — such as the ability to perform subsampling without altering the ergodic distribution and application to sampling transdimensional distributions.

The benefits of introducing the point mass at $\beta = 1$ are numerically investigated, and practical considerations related to choosing tuning parameters to encourage a desired proportion of time at the target distribution are presented (Section 3.4). We find that, analogously to standard methods in discrete-time, the tempered counterparts of PDMP samplers outperform vanilla PDMP on challenging sampling problems.

2 The Zig-Zag sampler

For the purposes of brevity and ease of exposition, we focus specifically on a continuous-tempered version of the Zig-Zag sampler. However, we stress that the underlying ideas can easily be applied to *any* PDMP sampler, for example, the Bouncy Particle Sampler Bouchard-Côté et al. [2018] or the Boomerang Sampler Bierkens et al. [2020]; see the comments at the end of Section 3.2.

Consider the problem of sampling from a target density defined for $\boldsymbol{x} \in \mathcal{X} := \mathbb{R}^d$ by

$$\pi(\boldsymbol{x}) = \frac{1}{Z} \exp(-U(\boldsymbol{x}))$$

where $U : \mathcal{X} \to \mathbb{R}$ is a continuous differentiable function referred to as the potential and Z is the, potentially unknown, normalising constant $Z = \int_{\mathcal{X}} \exp(-U(\boldsymbol{x}))d\boldsymbol{x}$. We will denote the un-normalised target density by q, so $\pi(\boldsymbol{x}) = q(\boldsymbol{x})/Z$.

The *Zig-Zag* process is a continuous-time piece-wise deterministic Markov process, which can be defined so as to have $\pi(\boldsymbol{x})$ as its invariant distribution. The process is defined on an extended state-space that can be viewed as consisting of a position, $\boldsymbol{x}$ and a velocity component, $\boldsymbol{v}$. For the

Zig-Zag process, the velocity is restricted to be ± 1 in each axis direction. Thus the extended space is $E = \mathbb{R}^d \times \{-1, 1\}^d$. We write $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{v})$ for $\boldsymbol{z} \in E$ from here on. We will use subscripts to denote time, and superscripts to denote components. So $\boldsymbol{z}_t$ will be the state at time t, while $\boldsymbol{x}_t^i$ will be the ith component of the position at time t.

For an event at time t we use the notation $\boldsymbol{z}_{t-}$ to be the state immediately before the event, and $\boldsymbol{z}_t$ the state immediately after it. The dynamics of the Zig-Zag process are deterministic between a set of random event times. At each event time the direction of one component of the velocity is switched. The deterministic dynamics are specified by a constant velocity model. So, if there are no events between times t and $t + h$, for $h > 0$, the change of state is given by $\boldsymbol{z}_{t+h} = (\boldsymbol{x}_{t+h}, \boldsymbol{v}_{t+h}) = (\boldsymbol{x}_t + h\boldsymbol{v}_t, \boldsymbol{v}_t)$.

The events occur with a rate that depends on the current state. For the Zig-Zag process we have d types of event, each of which results in the flipping of one of the d components of the velocity process. To ensure that we have $\pi(\boldsymbol{x})$ as the $\boldsymbol{x}$-marginal of the process's invariant distribution, these rates are defined to be, for $i = 1, \ldots, d$,

$$\lambda_i(\boldsymbol{x}_t, \boldsymbol{v}_t) = \max(0, \boldsymbol{v}_t^i \partial_{\boldsymbol{x}^i} U(\boldsymbol{x})),$$

with the transition at the corresponding event being that $\boldsymbol{v}_t^i = -\boldsymbol{v}_{t-}^i$, and all other elements of the state are unchanged.

Pseudo-code for simulating the Zig-Zag process is given in Algorithm 1. When we simulate such a process, the output of the algorithm is the set of event times, positions and velocities after the event times. This set is known as the PDMP skeleton and with the deterministic dynamics of the process it defines the continuous time path of the process. We can use such a simulated path to give us draws from $\pi(\boldsymbol{x})$, by discarding some suitably chosen initial path time as burn-in, and then evaluating the $\boldsymbol{x}$ component of the process at a set of evenly spaced discrete time-points (see e.g. Fearnhead et al. [2018] for alternative approaches that use the continuous-time paths).

Algorithm 1 Zig-Zag algorithm

Inputs: initial state $(\boldsymbol{x}_{t_0}, \boldsymbol{v}_{t_0})$ and number of simulated events K
$t_0 \leftarrow 0$
for k $\in 1, \ldots, K$ **do**
  Simulate an event-time τ_i for each rate $\lambda_i(\boldsymbol{x}_t, \boldsymbol{v}_t)$
  $i^* \leftarrow \text{argmin}_i\{\tau_i\}$ and $t_k \leftarrow t_{k-1} + \tau_{i^*}$
  $\boldsymbol{x}_{t_k} \leftarrow \boldsymbol{x}_{t_{k-1}} + \tau_{i^*}\boldsymbol{v}_{t_{k-1}}$ ▷ *Update position*
  $\boldsymbol{v}_{t_k}^{i^*} \leftarrow -\boldsymbol{v}_{t_{k-1}}^{i^*}$ ▷ *Update velocity*
end for
Outputs: Zig-Zag skeleton $\{t_k, \boldsymbol{x}_{t_k}, \boldsymbol{v}_{t_k}\}$.

3 Continuously-tempering with a point mass

3.1 Continuous-tempering

Continuous-tempering is an approach to improve mixing of MCMC and related sampling algorithms. It introduces a second distribution π_0, called the *base* distribution, which is assumed to have a density $\pi_0(\boldsymbol{x}) = \frac{q_0(\boldsymbol{x})}{Z_0}$. The idea is that this density will be simple to simulate from. For any $\beta \in [0, 1]$ it is now possible to define a distribution which interpolates between π and π_0 in that its density is $\pi(\boldsymbol{x})^\beta \pi_0(\boldsymbol{x})^{(1-\beta)}$. Continuous-tempering then defines a joint distribution on β and $\boldsymbol{x}$. This distribution has un-normalised density $q(\boldsymbol{x}, \beta)$, on $\mathbb{R}^d \times [0, 1]$, defined to be

$$q(\boldsymbol{x}, \beta) = q_0(\boldsymbol{x})^{1-\beta} q(\boldsymbol{x})^\beta.$$

In practice tempering methods often introduce a prior, $p(\beta)$ on $\beta \in [0, 1]$, and sample from the distribution proportional to $p(\beta)q(\boldsymbol{x}, \beta)$. The idea of introducing the prior is that it can be tuned to give a marginal distribution on β that has reasonable mass across all of the interval $[0, 1]$. Without a prior, the distribution q is likely to put almost all mass on β close to 0 or on β close to 1.

By construction, conditional on satisfying $\beta = 1$, samples drawn from $q(\boldsymbol{x}, \beta)$ are distributed according to π. However, if we use a continuous prior, samples will almost-surely *not* lie in that set,

and we have to use importance sampling to correct for $\beta \neq 1$. This leads to a weighted sample from π, with the Monte Carlo accuracy of the resulting approach depending greatly on the variability of the weights introduced by importance sampling.

To overcome this issue, we use a prior $p(\beta)$ that introduces a point mass at $\beta = 1$. Specifically, for $\alpha \in \mathbb{R}$, and an arbitrary probability density function κ on the interval $[0, 1)$, we define

$$p(\beta) = (1 - \alpha)\kappa(\beta)\mathbf{1}_{(0 \leq \beta < 1)} + \alpha\kappa(\beta)\mathbf{1}_{(\beta = 1)}.$$

With the above prior, we can define the augmented target

$$\omega(\mathrm{d}\boldsymbol{x}, \mathrm{d}\beta) \propto q(\boldsymbol{x}, \beta)(1 - \alpha)\kappa(\beta)\mathrm{d}\boldsymbol{x}\mathrm{d}\beta + q(\boldsymbol{x})\alpha\kappa(\beta)\mathrm{d}\boldsymbol{x}\delta_{\beta=1}. \tag{1}$$

We can extend the Zig-Zag process to sample from such a distribution. The next subsection describes the extension. The usual advantages of subsampling and transdimensional sampling using PDMPs apply for the tempered extension.

3.2 Continuously-tempered Zig-Zag

Unlike most MCMC samplers, Zig-Zag is a continuous-time process. This makes it simpler to deal with the point-mass at $\beta = 1$, using ideas from Chevallier et al. [2020]: we simulate paths for $\beta \in [0, 1)$ until β hits the boundary at $\beta = 1$; when this happens, we set the velocity component for β, denoted as $\boldsymbol{v}_t^{d+1}$, to zero so that the Zig-Zag process explores the distribution ω restricted to $\beta = 1$; the Zig-Zag process stays with $\beta = 1$ until the first event in a new Poisson process, at which point we set the velocity component for β to -1 to allow exploration of the distribution for $\beta \in [0, 1)$. We define the rate of the events when we leave $\beta = 1$ to be η. Algorithm 2 provides a sketch of the relevant modifications of the standard Zig-Zag process for events concerning the inverse temperature variable β.

Algorithm 2 Tempered Zig-Zag algorithm

1: Run Zig-Zag for $\beta_t \in [0, 1)$ until first time t at which trajectory hits $\beta = 1$
2: $\boldsymbol{v}_t^{d+1} \leftarrow 0$ ▷ *Kill the velocity component for* β
3: Simulate an event-time τ with rate η
4: Run the Zig-Zag process with $\beta = 1$, for time τ ▷ *Sample* π
5: $\boldsymbol{v}_{t+\tau}^{d+1} \leftarrow -1$ ▷ *Reintroduce* β
6: Goto Step 1.

An appropriate choice of rate to ensure that the resulting algorithm generates a process with ω as its limiting distribution is given in the following result (the proof of is located in the supplement).

Theorem 1 *Assuming $\kappa(\beta)$ is continuous and the rate function in Algorithm 2 is chosen as*

$$\eta = \frac{1 - \alpha}{2\alpha},$$

Then, the resulting Zig-Zag process is ω-ergodic.

The theorem statement above considers the case where κ is continuous but we note in the proof that more general choice is possible with a slight adjustment to the rate. Importantly, the rate $\eta(\boldsymbol{x})$ is constant, and thus simulating the time at which we transition from $\beta = 1$ to $\beta < 1$ is independent of the path of the process and simulating the event time is simple. While we have described the use of continuous tempering as an extension of the Zig-Zag process, the same construction readily extends to other PDMP samplers — one simply replaces the Zig-Zag dynamics for $\beta \in [0, 1)$ and $\beta = 1$ in Algorithm 2 with the dynamics of the corresponding PDMP sampler.

3.3 Importance sampling estimator

By construction, continuously-tempered Zig-Zag will spend a substantial amount of time in states with $\beta = 1$, and thus the states at those times can give draws from $\pi(\boldsymbol{x})$. We can use importance sampling ideas from Graham and Storkey [2017] to re-weight samples when $0 \leq \beta < 1$ to give weighted samples from $\pi(\boldsymbol{x})$; but this only applies if $\kappa(\beta) \propto \xi^{1-\beta}$ for some constant ξ.

The idea of importance sampling is that, for $\beta \in [0,1)$ we can marginalise out β from ω, and the marginal distribution for $\boldsymbol{x}$ is

$$\omega_{[0,1)}(\boldsymbol{x}) = \int_0^1 \kappa(\beta) q(\boldsymbol{x}) \left(\frac{q_0(\boldsymbol{x})}{q(\boldsymbol{x})}\right)^{1-\beta} \mathrm{d}\beta = q(\boldsymbol{x}) \int_0^1 \left(\frac{\xi q_0(\boldsymbol{x})}{q(\boldsymbol{x})}\right)^{1-\beta} \mathrm{d}\beta.$$

Defining $\Delta(\boldsymbol{x}) = \log q_0(\boldsymbol{x}) + \log \xi - \log q(\boldsymbol{x})$, the above is equal to

$$q(\boldsymbol{x}) \int_0^1 \exp\{(1-\beta)\Delta(\boldsymbol{x})\} \mathrm{d}\beta = \exp\{\Delta(\boldsymbol{x})\}\Delta(\boldsymbol{x})^{-1}(1 - \exp\{-\Delta(\boldsymbol{x})\}) = \frac{\exp\{\Delta(\boldsymbol{x})\} - 1}{\Delta(\boldsymbol{x})}.$$

Thus if we have this as our proposal distribution, then the corresponding importance sampling weights will be $w(\boldsymbol{x}) = \Delta(\boldsymbol{x})/[\exp\{\Delta(\boldsymbol{x})\} - 1]$. This will involve additional post-sampling computation of the importance weights at the samples taken from the Zig-Zag trajectory.

3.4 Calibration of $\kappa(\beta)$

The efficiency of continuous-tempering depends on an appropriate choice of κ and α. Ideally, they should be chosen so to balance a non-negligible amount of time at $\beta = 1$, while simultaneously allowing occasional excursions to lower inverse temperatures to help mix, e.g. between modes.

For $\beta \in [0,1]$ define, $Z(\beta) = \int_{\mathbb{R}^d} q(\boldsymbol{x}, \beta) d\boldsymbol{x}$. The induced β-marginal of ω for $\beta_0 \in [0,1)$ is

$$\omega(\beta_0) = \frac{(1-\alpha)\kappa(\beta_0)Z(\beta_0)}{(1-\alpha)\int_0^1 \kappa(\beta)Z(\beta)d\beta + \alpha\kappa(1)Z(1)}.$$

The above suggests that an appropriate choice is $\kappa(\beta) \propto Z(\beta)^{-1}$, as this would both induce β to be marginally *uniform* under ω for $\beta \in [0,1)$, and cause the parameter α to directly represent the probability that $\beta = 1$ under ω. When not using the important sampling strategy, the proposed approach is thus to choose $\kappa(\beta) \propto \widehat{Z}(\beta)^{-1}$, where $\widehat{Z}(\beta) = \exp\left\{\sum_{k=0}^{m-1} a_k \beta^k\right\}$ for some coefficients $\{a_k\}_{k=0}^{m-1}$ that are estimated by regressing point estimates of $\log Z(\beta)$ obtained via the path sampling identity. In practice, an estimate $\widehat{Z}(\beta)$ can be obtained from a pilot run of the algorithm with $\alpha \ll 1$ or using methods described in Gelman and Meng [1998].

4 Numerical experiments

4.1 Mixture of Gaussians

In our first example, the target corresponds to a mixture of Gaussian distributions with equal weights and variances so our target has unnormalised density

$$q(\boldsymbol{x}) = \sum_{i=1}^{K} \exp\left(-\frac{1}{2\sigma^2}(\boldsymbol{x} - \boldsymbol{\mu}_i)^\top(\boldsymbol{x} - \boldsymbol{\mu}_i)\right),$$

where $K = 5$, $\sigma^2 = 0.2$ and $\{\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_5\}$ were generated uniformly on the region $[0,10] \times [0,10]$ and are given in the supplementary material. Figure 1 plots the PDMP trajectories for the tempered and untempered Zig-Zag sampler, in addition to the trajectories for inverse temperature fixed at $\beta = 1$. For this example we choose $q_0(x)$ as a Gaussian $\mathcal{N}(\nu, \Sigma)$ centred at $\nu = (5,5)^T$ with covariance matrix $\Sigma = 2\mathbf{I}_2$.

Table 1 displays the root-mean-square-error (RMSE) of the Monte Carlo estimates from the Zig-Zag and tempered Zig-Zag averaged over 20 runs. All methods were simulated for 50,000 event-times with the first 40% used as burnin in the standard Zig-Zag and used for both burn-in and estimating the polynomial $\kappa(\beta)$ in the tempered samplers. The standard Zig-Zag is not able to explore the multiple modes yielding worse estimates of the first and second moments. We also compare to a direct Zig-Zag analogue of the continuously tempered HMC algorithm Graham and Storkey [2017] where no mass is given to $\beta = 1$ and $\kappa(\beta) \propto 1$, the estimator is based on importance sampling as defined in Section 3.3. Further details and boxplots showing variability of the estimated moments are given in the supplementary material.

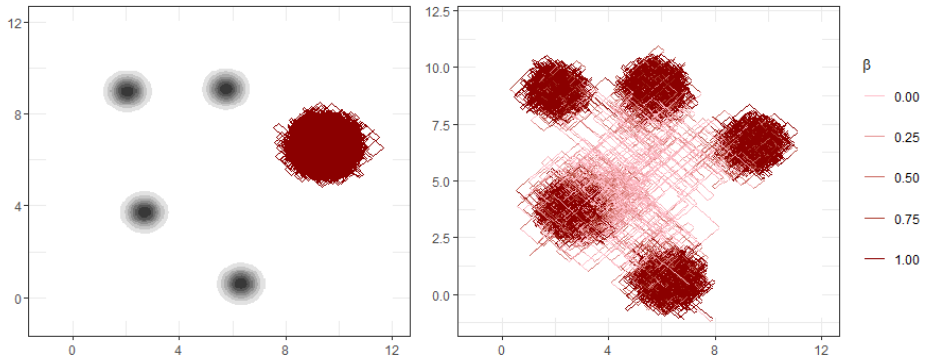

Figure 1: Trajectories of the Zig-Zag process simulated for 30,000 events for a multi-modal Gaussian mixture model, Zig-Zag (left) and continuously tempered Zig-Zag (right) with $\alpha = 0.7$.

Table 1: Recovery of first two moments of a Gaussian mixture (averaged over 20 replications).

Method	α	$\omega(\beta = 1)$	Root-mean-square error (RMSE) $\mathbb{E}[X_1]$	$\mathbb{E}[X_2]$	$\mathbb{E}[X_1^2]$	$\mathbb{E}[X_2^2]$	Thinning efficiency
Zig-Zag	1	1	2.557	2.740	30.577	31.761	0.057
Zig-Zag CT	0.8	0.789	0.650	0.741	7.898	7.182	0.080
	0.7	0.703	0.399	0.683	4.563	6.418	0.090
	0.5	0.499	0.329	0.539	4.199	4.930	0.114
	0.3	0.302	0.304	0.453	**3.216**	**4.155**	0.139
	0.2	0.197	**0.294**	0.472	3.756	4.617	0.153
	0.1	0.097	0.349	**0.389**	3.987	4.198	0.167
Zig-Zag CT (IS)	0	0	0.483	0.472	5.567	5.030	**0.301**

We find that the tuning procedure yielded precise control over the time spent at $\beta = 1$, and that for α between 0.1 and 0.3 the results are similar. Tempering with a point mass was found to give more efficient estimates of the first and second moments. This may be because the importance sampling estimate spends more time at $\beta \approx 0$. In addition to exhibiting lower RMSEs, the tempered versions of the Zig-Zag sampler have better computational properties. The thinning efficiency, measured as the proportion of proposals that result in an event-time simulation is $\approx .06$ at $\beta = 1$ but improves significantly when the sampler can transition to lower temperatures. While the importance sampling estimator for $\beta = 0$ has the best thinning efficiency it requires additional post processing to evaluate the importance sampling weights.

4.2 Transdimensional example

We apply the tempered Zig-Zag to the challenge of sampling a transdimensional distribution. Such target distributions arise naturally in variable selection problems. The resulting target (posterior) distribution is a discrete mixture of 2^p models, where p is the number of variables in the dataset.

Such a setting produces a challenge, as typical samplers such as HMC do not extend naturally to such spaces. On the other hand, PDMPs have recently been extended to sample transdimensional distributions Chevallier et al. [2020, 2021], Bierkens et al. [2021]. This extension employs within-model gradient information to efficiently explore the space and jumps between models when a parameter hits zero. The approach is beneficial as an informative likelihood function's gradient will direct less informative variables towards zero and more informative ones away from zero, aiding exploration of the sampler.

This example explores how tempering a target with a point mass can improve performance over the standard transdimensional Zig-Zag process. Specifically, we define a family of example problems of increasing difficulty in the sense that for higher values of an underlying parameter m, separation

between the mode of the slab component and the spike at zero is increased. Allow

$$q(\boldsymbol{x}) = \prod_{i=1}^{2} (w\phi(x_i; m, \sigma^2) + (1-w)\delta_0(x_i)), \tag{2}$$

where $\phi(x_i; m, \sigma^2)$ is the normal density and $w = 0.5$ is the probability of a variable being included. We fix $\sigma^2 = 0.5$ and increase m from $m = 0$ to $m = 4$. To enable tempering, define

$$q(\boldsymbol{x}, \beta) = \prod_{i=1}^{2} (w\phi(x_i; m\beta, \sigma^2) + (1-w)\delta_0(x_i)),$$

where at $\beta = 0$ the spike and slab is centred at zero, encouraging variables to enter and exit the model. While this is a somewhat artificial example, it is important to note that it encompasses precisely the situations which transdimensional PDMPs will find challenging, and allows us to explore robustness for increasing levels of pathology in the purest setting possible.

Figure 2 (rightmost panel) displays the dynamics of the tempered Zig-Zag for this problem for the choice of $m = 4$. Note that the standard transdimensional Zig-Zag process (bottom left) becomes stuck in a single model whilst its tempered counterpart is able to transition easily between models and thus visit the required areas for which each coordinate is equal to zero.

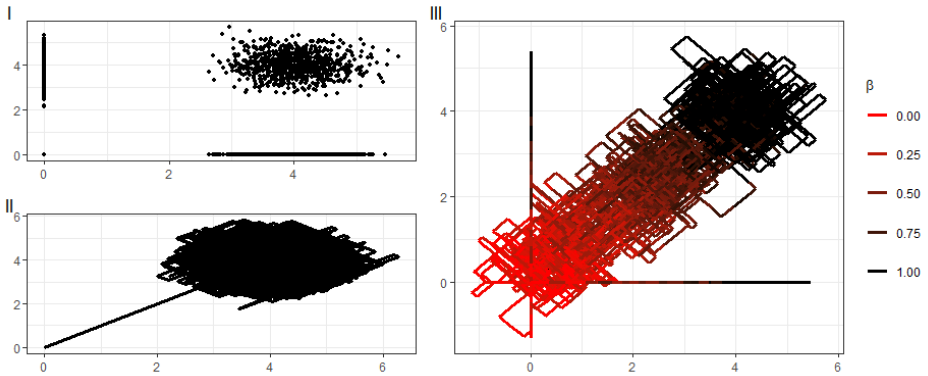

Figure 2: Sampling from (2) with $m = 4$. I. Exact samples from the spike-slab distribution. II. Standard Zig-Zag for 10^4 event-times. III. Tempered Zig-Zag trajectories with $\alpha = 0.5$ for 10^4 event-times.

Table 2: Absolute error of marginal mean and probability of inclusion for X_1 (averaged over 10 simulations for increasing m in (2)).

		Mean Absolute Error				
		$m = 0$	$m = 1$	$m = 2$	$m = 3$	$m = 4$
$\mathbb{E}[X_1]$	Zig-Zag	**0.005**	**0.018**	0.633	1.498	1.998
	Zig-Zag CT	0.007	0.025	**0.022**	**0.047**	**0.214**
$\mathbb{P}(\lvert X_1 \rvert > 0)$	Zig-Zag	**0.009**	**0.012**	0.322	0.500	0.500
	Zig-Zag CT	0.010	0.023	**0.008**	**0.018**	**0.055**

Table 2 gives the absolute error for the Monte Carlo estimates of the marginal inclusion probabilities and marginal means. For lower m, the standard and tempered Zig-Zag perform similarly as the mode of the slab is close enough to ensure regular crossing of the zero-axis for the PDMP trajectories. However, as m increases, the marginal probabilities of inclusion tend to 1 and the standard Zig-Zag process becomes stuck in a single model.

4.3 Boltzmann machine relaxation

Following Nemeth et al. [2019] and Graham and Storkey [2017], the final example considers sampling a continuous relaxation of a Boltzmann machine distribution. Full details surrounding the derivation

of the distribution can be found in [Nemeth et al., 2019, Supplementary Material, Section D], though for the considered example the target density on $\mathbb{R}^{d_r}$ is of the form

$$q(\boldsymbol{x}) = \frac{2^{d_b}}{(2\pi)^{d_r/2} Z_b \exp(\frac{1}{2}\mathrm{Tr}(\boldsymbol{D}))} \exp\left(-\frac{1}{2}\boldsymbol{x}^\top \boldsymbol{x}\right) \prod_{k=1}^{d_b} \cosh(\boldsymbol{q}_k^\top \boldsymbol{x} + b_k),$$

where $\boldsymbol{q}_k$ denotes the k-th row of $\boldsymbol{Q}$, which is a $d_b \times d_r$ matrix such that $\boldsymbol{Q}\boldsymbol{Q}^\top = \boldsymbol{W} + \boldsymbol{D}$, and $\boldsymbol{D}$ is an arbitrary diagonal matrix that ensures $\boldsymbol{W} + \boldsymbol{D}$ is positive semi-definite. The above is a continuous relaxation of the Boltzmann Machine distribution on $\{-1, 1\}^{d_b}$ with probability mass function

$$q(\mathrm{s}) = Z_b^{-1} \exp\left(\frac{1}{2}\boldsymbol{s}^\top \boldsymbol{W}\boldsymbol{s} + \boldsymbol{s}^\top \boldsymbol{b}\right).$$

The moments up to second order of the relaxation and the original (discrete) Boltzmann Machine distribution are related via $\mathbb{E}[\boldsymbol{X}] = \boldsymbol{Q}^\top \mathbb{E}[\boldsymbol{S}]$ and $\mathbb{E}[\boldsymbol{X}\boldsymbol{X}^\top] = \boldsymbol{Q}^\top \mathbb{E}[\boldsymbol{S}\boldsymbol{S}^\top] + \boldsymbol{I}$. We employ a similar experimental setup as in Nemeth et al. [2019], namely a 28-dimensional example which allows for exact computation via enumeration of the moments of $\boldsymbol{S}$, and hence in turn of $\boldsymbol{X}$ (the latter being useful for evaluating sampler performance). Table 3 displays the results. We find that the best performance is given by the Zig-Zag sampler with $\alpha = 0.2$ which far outperforms the standard Zig-Zag $\alpha = 1$ and importance sampling version of the sampler. For these experiments the tuning of κ is notably sub-optimal as the proportion of time spent at zero is not fully controlled by α. Despite this, the samplers were able to outperform the importance sampling approach where $\alpha = 0$ and the standard Zig-Zag. Further details on the experiment and specification of κ may be found in the supplementary material.

Table 3: Average root-mean-square error of the first and second moments of the Boltzmann machine relaxation averaged over 20 simulations reported to 3 decimal places.

Method	α	$\omega(\beta = 1)$	Average RMSE $\mathbb{E}[X_k]$	Average RMSE $\mathbb{E}[X_k^2]$	Thinning efficiency
Zig-Zag	1	1	1.304	2.456	0.146
Zig-Zag CT	0.700	0.632	0.515	1.718	0.187
	0.500	0.417	0.493	1.563	0.219
	0.300	0.231	0.417	1.890	0.251
	0.200	0.145	**0.296**	**1.089**	0.267
	0.100	0.076	0.594	2.643	0.284
Zig-Zag CT (IS)	0	0	0.566	3.580	**0.505**

5 Discussion

We present a general strategy to improve the performance of PDMP samplers on challenging targets. The approach uses an extended distribution that uses an inverse temperature which interpolates between a challenging distribution of interest and a tractable base distribution. At lower values of β the mixing will improve and we allow a pointmass at inverse at $\beta = 1$ to attain exact samples from the distribution. As a proof of concept, a numerical study of these ideas surrounding the Zig-Zag sampler was employed. This revealed that there are considerable benefits to the proposed extensions, as well as to introducing the point mass at $\beta = 1$. We compared our tempering with a point-mass extension of the Zig-Zag to a version based on continuously tempered estimates that are given using importance sampling ideas from Graham and Storkey [2017]. The importance sampling approach requires restrictive choice of κ and gave inferior performance. Our approach may be readily applied to other PDMP samplers and can incorporate assets of sampling with a PDMP such as sampling transdimensional spaces and using subsampling methods. Another avenue of research is in incorporating direct simulation from q_0 when the sampler hits $\beta = 0$ — such moves would be particularly useful when q_0 is a tractable multimodal distribution.

Broader impact

Effective sampling methods are instrumental in many probabilistic modelling approaches that account for uncertainty quantification (e.g., Bayesian Machine Learning approaches). Thus, it is important to employ reliable methods in obtaining such estimates, especially if they are used for downstream tasks and decisions. The methodological innovations discussed herein aim to produce more reliable results in the context of PDMP samplers, which are a growing area of research. The proposed approaches are more suited to sampling challenging multi-modal distributions with PDMP-based samplers. As the methods are presented as a proof of concept, and do not feature as an implemented component within any deployed artificial intelligence system, there are no immediate broader consequences related to their potential failures.

References

J. Bierkens. Non-reversible Metropolis-Hastings. *Statistics and Computing*, 26(6):1213–1228, 2016.

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

J. Bierkens, S. Grazzi, K. Kamatani, and G. Roberts. The boomerang sampler. In *International Conference on Machine Learning*, pages 908–918. PMLR, 2020.

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

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

A. Chevallier, P. Fearnhead, and M. Sutton. Reversible jump PDMP samplers for variable selection. *arXiv:2010.11771*, 2020.

A. Chevallier, S. Power, A. Q. Wang, and P. Fearnhead. PDMP Monte Carlo methods for piecewise-smooth densities. *arXiv:2111.05859*, 2021.

M. H. A. Davis. *Markov Models and Optimization*. Monographs on Statistics and Applied Probability. Springer Science and Business Media, 1993.

P. Diaconis, S. Holmes, and R. M. Neal. Analysis of a nonreversible Markov chain sampler. *Annals of Applied Probability*, pages 726–752, 2000.

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

A. Gelman and X.-L. Meng. Simulating normalizing constants: From importance sampling to bridge sampling to path sampling. *Statistical science: a review journal of the Institute of Mathematical Statistics*, 13(2):163–185, 1998.

M. M. Graham and A. J. Storkey. Continuously tempered Hamiltonian Monte Carlo. In *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence (UAI)*, 2017.

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

R. M. Neal. MCMC using Hamiltonian dynamics. *Handbook of Markov chain Monte Carlo*, 2(11):2, 2011.

C. Nemeth, F. Lindsten, M. Filippone, and J. Hensman. Pseudo-extended Markov chain Monte Carlo. *Advances in Neural Information Processing Systems*, 32, 2019.

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

M. Sutton and P. Fearnhead. Concave-Convex PDMP-based sampling. *arXiv:2112.12897*, 2021.

R. H. Swendsen and J.-S. Wang. Replica Monte Carlo simulation of spin-glasses. *Physical review letters*, 57(21):2607, 1986.

G. Vasdekis and G. O. Roberts. Speed up zig-zag. *arXiv:2103.16620*, 2021.

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

Y. Yao, C. Cademartori, A. Vehtari, and A. Gelman. Adaptive path sampling in metastable posterior distributions. *arXiv:2009.00471*, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Conclusion.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Broader Impact statement.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Theorem 1.
   (b) Did you include complete proofs of all theoretical results? [Yes] See Supplementary Material.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See Supplementary Material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Numerical Experiments section.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See plots included in Supplementary Material for visualization of variability for multiple runs.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Supplementary Material.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Section 4.3.
   (b) Did you mention the license of the assets? [Yes] See Supplementary Material.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] Yes, see Supplementary Material.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] See Supplementary Material.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]