Supplementary Material: Continuously-Tempered PDMP samplers

1 Proof of Theorem 1

The measure $q(\boldsymbol{x}, \beta)p(\beta)d\boldsymbol{x}d\beta$ has a density on the open set $\mathbb{R}^d \times [0, 1)$. We will sample it using the Zig-Zag process on the extended space $E_0 = (\mathbb{R}^d \times [0, 1)) \times \{-1, 1\}^{d+1}$. On the other hand, the measure $\delta_{\beta=1}q(\boldsymbol{x})d\boldsymbol{x}$ is essentially a density on $\mathbb{R}^d$. We will sample it using Zig-Zag on the extended space $F = \mathbb{R}^d \times \{-1, 1\}^d$.

Following the construction of [6, 5], we "stitch" E_0 and F together through an active boundary. Let B^{in} be the "entrance" boundary at the temperature $\beta = 1$: $B^{in} = (\mathbb{R}^d \times \{1\}) \times (\{-1, 1\}^d \times \{-1\})$, and let B^{out} be the "exit" boundary $B^{out} = (\mathbb{R}^d \times \{1\}) \times (\{-1, 1\}^d \times \{1\})$.

The process Z_t is as follows: when in E_0, should it hit the boundary B^{out} at time t^-, i.e. $Z_{t^-} \in B^{out}$, it jumps to F at time t: $Z_t \in F$. Note that the process Z_t never enters B^{out}. When in F, the process jumps back to the entrance boundary B^{in} with rate $\eta(\boldsymbol{z})$. The state space is:

$$E = E_0 \cup B^{in} \cup F.$$

More precisely, if $Z_{t^-} \in B^{out}$, then $Z_t = g(Z_{t^-})$ with g being the projection that removes the temperature coordinate and velocity:

$$\begin{aligned} g : B^{out} &\to F \\ (\boldsymbol{x}, 1, \boldsymbol{v}, 1) &\mapsto (\boldsymbol{x}, \boldsymbol{v}). \end{aligned}$$

Conversely, if the process jumps from F to B^{in} at time t then $Z_t = f(Z_{t^-})$ with

$$\begin{aligned} f : F &\to B^{in} \\ (\boldsymbol{x}, \boldsymbol{v}) &\mapsto (\boldsymbol{x}, 1, \boldsymbol{v}, -1). \end{aligned}$$

With this construction, we use Theorem 2 of [6] and follow the proof of Theorem 3 of [6] for our setting to show that ω will be invariant for the process if

$$\eta(\boldsymbol{x}) = \frac{q(\boldsymbol{x}, 1)\kappa(1^-)}{q(\boldsymbol{x})\kappa(1)} \frac{1 - \alpha}{2\alpha} = \frac{\kappa(1^-)}{\kappa(1)} \frac{1 - \alpha}{2\alpha}.$$

Intuitively, this result is obtained by choosing an η that balances the flows $E_0 \cup B^{in} \to F$ and $F \to E_0 \cup B^{in}$. Starting from the target distribution, the amount of mass that flows through a point $\boldsymbol{z} = (\boldsymbol{x}, 1, \boldsymbol{v}, 1) \in B^{out}$ to $(\boldsymbol{x}, \boldsymbol{v}) \in F$ during a time interval of length dt is $\frac{1}{2^{d+1}} q(\boldsymbol{x}, 1)\kappa(1^-)(1 - \alpha)dt$. Conversely, the amount of mass that flows through a point $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{v}) \in F$ to $(\boldsymbol{x}, 1, \boldsymbol{v}, -1) \in E_0$ is $\frac{1}{2^d} q(\boldsymbol{x})\kappa(1)\alpha\eta(\boldsymbol{x})dt$. Hence, the previous choice of η balances the flows out.

2 Continuously-tempered Zig-Zag rates

When sampling in E_0, the Zig-Zag process has a rate for each component of $\boldsymbol{x}$,

$$\lambda_{\boldsymbol{x}^j}(\boldsymbol{z}) = \max(0, -\boldsymbol{v}^j \partial_{\boldsymbol{x}^j} \log q(\boldsymbol{x}, \beta)) \quad j = 1, \ldots, d, \tag{1}$$

and an additional rate for the inverse temperature β,

$$\lambda_\beta(\boldsymbol{z}) = \max(0, -\boldsymbol{v}^{d+1}(\partial_\beta \log q(\boldsymbol{x}, \beta) + \partial_\beta \log \kappa(\beta)).$$

When sampling in F, there are d rates

$$\lambda_{\boldsymbol{x}^j}(\boldsymbol{z}) = \max(0, -\boldsymbol{v}^j \partial_{\boldsymbol{x}^j} \log q(\boldsymbol{x})) \quad j = 1, \ldots, d,$$

which correspond to those given in (1) since by definition $q(\boldsymbol{x}, \beta = 1) = q(\boldsymbol{x})$.

Preprint. Under review.

3 Simulation via thinning

The main practical challenge with simulating the Zig-Zag process, for example with Algorithm 1, is simulating the event times. Whilst the event times depend on the state, as the state-dynamics are deterministic until the next event, these can be re-expressed as rates that depend only on time. To see this consider the ith event, and assume that we are at time t and the current state is $\boldsymbol{z}_t$. Until there is the next event (which could be any of the d possible events), the state will evolve as $\boldsymbol{z}_{t+h} = (\boldsymbol{x}_t + h\boldsymbol{v}_t, \boldsymbol{v}_t)$, thus the rate until the next event of type i will be

$$\tilde{\lambda}_i(h) = \lambda_i(\boldsymbol{z}_{t+h}) = \max(0, \boldsymbol{v}_t^i \partial_{\boldsymbol{x}^i} U(\boldsymbol{x}_t + h\boldsymbol{v}_t)).$$

Simulating a Zig-Zag process thus requires simulating events of an inhomogeneous Poisson processes of rates $\tilde{\lambda}_i(t)$. We sample a random variable u uniformly in $[0, 1]$ and the next event time is the time t such that:

$$\int_0^t \tilde{\lambda}_i(s) ds = -\log(u). \tag{2}$$

In practice, solving this equation analytically is only possible for a restricted class of rate $\tilde{\lambda}$, such as rates that are piecewise constant or piecewise linear functions of time. Where we can not simulate event times directly, we can use an approach called *thinning*.

We find an upper rate $\lambda_i^+(s)$ such that $\tilde{\lambda}_i(s) \leq \lambda_i^+(s)$ for all s, and such that we can simulate events at rate λ_i^+ directly. We then propose events at this larger rate, and accept each proposed event with probability $\tilde{\lambda}_i(t)/\lambda_i^+(t)$.

The thinning method requires computing an upper bound $\bar{\lambda}_i$. The computational efficiency of the resulting algorithm for simulating the Zig-Zag process is directly related to the quality (tightness) of the upper bound: a loose upper bound will lead to many rejections and therefore wasted simulation effort. We note that one useful approach for constructing appropriate upper bounds in Lemma 1 below. This approach has been used many times [2–5] to construct thinning bounds and is repeated here for completeness — further details may be found in the derivation from Section 3.3 of [1].

Lemma 1. *Suppose there exists a matrix $M = (M_{ij})_{i,j=1}^d \in \mathbb{R}^{d\times d}$ such that for every $\boldsymbol{x} \in \mathbb{R}^d$ and element of the Hessian $H(\boldsymbol{x}) = (-\partial_i\partial_j \log q(\boldsymbol{x}))_{i,j=1}^d$, we have $|H(\boldsymbol{x})_{i,j}| \leq M_{i,j}$ then the following linear bound*

$$\lambda_i(s) = \max(0, -\boldsymbol{v}^i \partial_{\boldsymbol{x}^i} \log q(\boldsymbol{x} + s\boldsymbol{v})) \leq \max(0, a_i + b_i s)$$

where $a_i = -\boldsymbol{v}^i \partial_{\boldsymbol{x}^i} \log q(\boldsymbol{x})$ and $b_i = \sum_{j=1}^d M_{ij}$.

Proof. The following in-time bound holds

$$\frac{d}{ds}\left[-\boldsymbol{v}^i \partial_{\boldsymbol{x}^i} \log q(\boldsymbol{x} + s\boldsymbol{v})\right] = -\boldsymbol{v}^i \sum_{j=1}^d \boldsymbol{v}^j \partial_{\boldsymbol{x}^j}\partial_{\boldsymbol{x}^i} \log q(\boldsymbol{x} + s\boldsymbol{v}) \leq \sum_{j=1}^d M_{ij}$$

from which, $\max\left(0, -\boldsymbol{v}^i \partial_{\boldsymbol{x}^i} \log q(\boldsymbol{x} + s\boldsymbol{v})\right) \leq \max\left(0, -\boldsymbol{v}^i \partial_{\boldsymbol{x}^i} \log q(\boldsymbol{x}) + s\sum_{j=1}^d M_{ij}\right)$. □

We note also that numerical approaches to simulating the event times have been proposed [10].

4 Thinning bounds for geometric tempering

If one can use Lemma 1 to simulate the Zig-Zag at inverse temperature $\beta = 0$ and $\beta = 1$, then implementing thinning for the geometrically tempered target is trivial. This approach assumes standard geometric tempering $q(\boldsymbol{x}, \beta) = q_0(\boldsymbol{x})^{1-\beta} q(\boldsymbol{x})^\beta$ and $\kappa(\beta) = \exp(-\sum_{k=0}^m \psi_k \beta^k)$ where $\psi_k \in \mathbb{R}$ are constants chosen according to Section 3.4.

Suppose we have matrices M and M^0 which bound the Hessians of the target $H(\boldsymbol{x})$ and base $H_0(\boldsymbol{x})$ distributions.

The rate function for the movement in the $\boldsymbol{x}^j$ coordinate will depend on

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

The following bound $\lambda(s) \leq \bar{\lambda}(s)$ applies

$$\bar{\lambda}(s) = \max(0, (\beta + \boldsymbol{v}^{d+1}s)(a_q + b_q s) + (1 - (\beta + \boldsymbol{v}^{d+1}s))(a_{q_0} + b_{q_0}s))$$

where

$$a_q = -\boldsymbol{v}^j \partial_{\boldsymbol{x}^j} \log q(\boldsymbol{x}), \qquad a_{q_0} = -\boldsymbol{v}^j \partial_{\boldsymbol{x}^j} \log q_0(\boldsymbol{x})$$

$$b_q = \sum_{j=1}^{d} M_{ji} \qquad b_{q_0} = \sum_{j=1}^{d} M^0_{ji}.$$

Further simplification gives,

$$\bar{\lambda}(s) = \max(0, a + bs + cs^2) \tag{3}$$

where $a = \beta a_q + (1-\beta)a_{q_0}$, $b = \boldsymbol{v}^{d+1}(a_q - a_{q_0}) + \beta b_q + (1-\beta)b_{q_0}$, and $c = \boldsymbol{v}^{d+1}(b_q - b_{q_0})$.

The rate function for the movement in the β coordinate will depend on

$$\partial_\beta \left[\log \kappa(\beta) + \log q(\boldsymbol{x}, \beta)\right] = -\sum_{k=1}^{m-1} k\psi_k \beta^{k-1} + \log q(\boldsymbol{x}) - \log q_0(\boldsymbol{x}).$$

The following in-time bound holds for q and an analogous bound holds for q_0

$$\begin{aligned} \frac{d}{ds^2}\left[-\boldsymbol{v}^{d+1} \log q(\boldsymbol{x} + s\boldsymbol{v})\right] &= \frac{d}{ds}\left[-\boldsymbol{v}^{d+1} \sum_{j=1}^{d} \boldsymbol{v}^j \partial_{\boldsymbol{x}^j} \log q(\boldsymbol{x} + s\boldsymbol{v})\right] \\ &= -\boldsymbol{v}^{d+1} \sum_{i=1}^{d} \boldsymbol{v}^i \sum_{j=1}^{d} \boldsymbol{v}^j \partial_{\boldsymbol{x}^i} \partial_{\boldsymbol{x}^j} \log q(\boldsymbol{x} + s\boldsymbol{v}) \\ &\leq \sum_{i=1}^{d} \sum_{j=1}^{d} M_{ij}, \end{aligned}$$

from which,

$$-\boldsymbol{v}^{d+1} \log q(\boldsymbol{x} + s\boldsymbol{v}) \leq a_q + b_q s + c_q s^2$$

where

$$a_q = -\boldsymbol{v}^{d+1} \log q(\boldsymbol{x}), \qquad b_q = -\boldsymbol{v}^{d+1} \sum_{j=1}^{d} \boldsymbol{v}^j \partial_{\boldsymbol{x}^j} \log q(\boldsymbol{x}), \qquad c_q = \frac{1}{2} \sum_{i=1}^{d} \sum_{j=1}^{d} M_{ij}.$$

The rate function for the inverse temperature β may be bounded by

$$\bar{\lambda}(s) = \max(0, -\sum_{k=1}^{m-1} k\psi_k(\beta + s\boldsymbol{v}^{d+1})^{k-1} + a_q + b_q s + c_q s^2 + a_{q+0} + b_{q_0}s + c_{q_0}s^2). \tag{4}$$

The rates from (3) and (4) are polynomial in s, so thinning can be implemented using the approach of [11].

5 Thinning in the Examples

In Examples 1 and 3, we use geometric tempering from an approximating multivariate-Gaussian distribution. Thinning is implemented using the arguments from section 4 and bounds on the Hessians for the base and target distributions.

5.1 Hessian bound for a multivariate Gaussian

One common choice for a base distribution is a d-dimensional multivariate Gaussian. Here

$$q(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^d \Sigma}} \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu})^\top \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\mu})\right)$$

where the Hessian is $H(\boldsymbol{x}) = (-\partial_i \partial_j \log q(\boldsymbol{x}))_{i,j=1}^d$ so we have the upper-bound $|H(\boldsymbol{x})| \leq M$ where $M_{ij} = |\Sigma^{-1}_{ij}|$.

5.2 Hessian bound for mixture of Gaussians

The target has un-normalised density,

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

Following the argument in Section 4, it suffices to find a matrix that bounds the Hessian of $\log q(x)$. Since q is the mixture of independent Gaussians it also follows that $H(\boldsymbol{x})_{i,j} = 0$ for $i \neq j$.

Let $\phi_k(x) = \exp\left(-\frac{1}{2\sigma^2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^\top(\boldsymbol{x}-\boldsymbol{\mu}_k)\right)$ then,

$$\partial_{\boldsymbol{x}^i} \log q(x) = \frac{\sum_{k=1}^{K} \phi_k(\boldsymbol{x})\frac{1}{\sigma^2}(\boldsymbol{x}^i - \boldsymbol{\mu}_k^i)}{\sum_{k=1}^{K}\phi_k(\boldsymbol{x})} = \frac{1}{\sigma^2}\left(\boldsymbol{x}^i - \frac{1}{q(\boldsymbol{x})}\sum_{k=1}^{K}\boldsymbol{\mu}_k^i\phi_k(\boldsymbol{x})\right)$$

with second derivative,

$$\begin{aligned}\partial_{\boldsymbol{x}^i}\partial_{\boldsymbol{x}^i}\log q(\boldsymbol{x}) &= \frac{1}{\sigma^2}\left(1 + \frac{1}{\sigma^2}\left[\sum_{k=1}^{K}(\boldsymbol{\mu}_k^i)^2\frac{\phi_k(\boldsymbol{x})}{q(\boldsymbol{x})} - \left(\sum_{k=1}^{K}\boldsymbol{\mu}_k^i\frac{\phi_k(\boldsymbol{x})}{q(\boldsymbol{x})}\right)^2\right]\right)\\ &\leq \frac{1}{\sigma^2}\left(1 + \frac{1}{\sigma^2}\frac{1}{4}(M^i - m^i)^2\right)\end{aligned}$$

where $M^i = \max_k\{\boldsymbol{\mu}_k^i\}$ and $m^i = \min_k\{\boldsymbol{\mu}_k^i\}$ and the bound follows from Popovicius inequality. We have the bound $|H(\boldsymbol{x})| \leq M$ where $M_{ij} = 0$ for $i \neq j$ and $M_{i,i} = \frac{1}{\sigma^2}\left(1 + \frac{1}{\sigma^2}\frac{1}{4}(M^i - m^i)^2\right)$ otherwise.

5.3 Boltzmann machine relaxation

The target has un-normalised density,

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

Following the argument from Section 2, it suffices to find a bound on the Hessian matrix. The first order derivatives are

$$-\nabla_{\boldsymbol{x}}\log q(\boldsymbol{x}) = \boldsymbol{x} - \sum_{k=1}^{d_b}\boldsymbol{q}_k\tanh(\boldsymbol{q}_k^\top\boldsymbol{x} + b_k)$$

and the Hessian matrix is

$$H(\boldsymbol{x}) = \boldsymbol{I} - \sum_{k=1}^{d_b}\boldsymbol{q}_k\boldsymbol{q}_k^\top\mathrm{sech}^2(\boldsymbol{q}_k^\top\boldsymbol{x} + b_k).$$

As $0 \leq \mathrm{sech}^2(t) \leq 1$ we have

$$H(\boldsymbol{x}) \leq \boldsymbol{I} - \sum_{k=1}^{d_b}\min[0, \boldsymbol{q}_k\boldsymbol{q}_k^\top] = M^+, \qquad H(\boldsymbol{x}) \geq -\sum_{k=1}^{d_b}\max[0, \boldsymbol{q}_k\boldsymbol{q}_k^\top] = M^-.$$

We have the bound $|H(\boldsymbol{x})| \leq M$ where $M = \max(|M^+|, |M^-|)$. Thinning may be implemented using the argument from Section 4.

5.4 Transdimensional example

The transdimensional example does not use geometric tempering, but the bounds are simple to construct. The tempering is

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

where ϕ is the normal density function. For this example, we took $\kappa(\beta) = 1$.

Variables that are not stuck at the pointmass are simulated according to the slab $\mathcal{N}(\mu\beta, \sigma^2)$ distribution. With gradient $\partial_{\boldsymbol{x}^i} \log q(\boldsymbol{x}, \beta) = \frac{1}{\sigma^2}(\boldsymbol{x}^i - m\beta)$, their event rate is

$$\lambda_{\boldsymbol{x}^i}(s) = \max\left(0, \frac{\boldsymbol{v}^i}{\sigma^2}((\boldsymbol{x}^i + s\boldsymbol{v}^i) - \mu(\beta + s\boldsymbol{v}^{d+1})\right)$$

which is a linear function in s that may be simulated exactly using the methods of [11]. Following [5] the rate to reintroduce a variable $\boldsymbol{x}^i = 0$ is given by

$$\frac{w}{(1-w)}\phi(0; m\beta), \sigma^2) = \frac{w}{(1-w)}\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{m^2}{2\sigma^2}\beta^2\right)$$

which admits thinning using $\bar{\lambda}(s) = \frac{w}{(1-w)}\frac{1}{\sqrt{2\pi\sigma^2}}$. The rate for the temperature (when not stuck at $\beta = 1$) is

$$\lambda(s) = \max\left(0, \frac{m\boldsymbol{v}^{d+1}}{\sigma^2}\sum_{i:|\boldsymbol{x}^i|>0}\left(\boldsymbol{x}^i + s\boldsymbol{v}^i - m(\beta + s\boldsymbol{v}^{d+1})\right)\right),$$

which is a linear function of s that may be simulated exactly using the methods of [11].

6 Additional simulation details

6.1 Mixture of Gaussians

For the Mixture of Gaussians the location of the means are given below (stated to 2 decimal places).

Table 1: Location of the means for the Gaussian mixture

$\boldsymbol{\mu}^1$	2.66	5.73	2.02	9.45	6.29
$\boldsymbol{\mu}^2$	3.72	9.08	8.98	6.61	0.62

For $\sigma^2 = 0.5$ the distance between most local modes is greater than 15 standard deviations with the minimum distance being 14.85 standard deviations from it's closest neighbour. This presents a challenging posterior for sampling as seen in Figure 1 of the main paper. For this example the exact first and second moment of the Gaussian mixture can be calculated exactly and is stated in Table 2 below.

Table 2: Table of exact first and second moments of the Gaussian mixture model

	X_1	X_2
$\mathbb{E}[X_k]$	5.228	5.803
$\mathbb{E}[X_k^2]$	34.751	44.418

Boxplots showing the variability of estimated first and second moments of X_1 show the performance improvement given using tempering $\alpha \neq 1$ and using a point-mass $\alpha \neq 0$.

6.2 Tuning of $\kappa(\beta)$ in experiments

For the tempered Zig-Zag, $\kappa(\beta)$ is chosen to approximate $Z(\beta)$. This quantity may be estimated using many approaches outlined in [7]. For our experiments, we use numerical integration as described in Section 2.3 of [7] further details may be found therein. We use the trapezoidal rule to estimate

$$\log\widehat{Z(\beta_{(k)})} = \frac{1}{2}\sum_{j=1}^{k-1}(\beta_{(j+1)} - \beta_{(j)})(\bar{U}_{(j)} + \bar{U}_{(j+1)}),$$

where $\bar{U}_{(j)} = \mathbb{E}\left[\partial_\beta \log q(\boldsymbol{x}, \beta) \mid \beta = \beta_{(j)}\right]$ is estimated using Monte Carlo and the values of β are ordered so that $\beta_{(1)} < \beta_{(2)} < \cdots < \beta_{(k)} \leq \cdots < \beta_{(n)}$. In practice, either a finite grid of fixed values

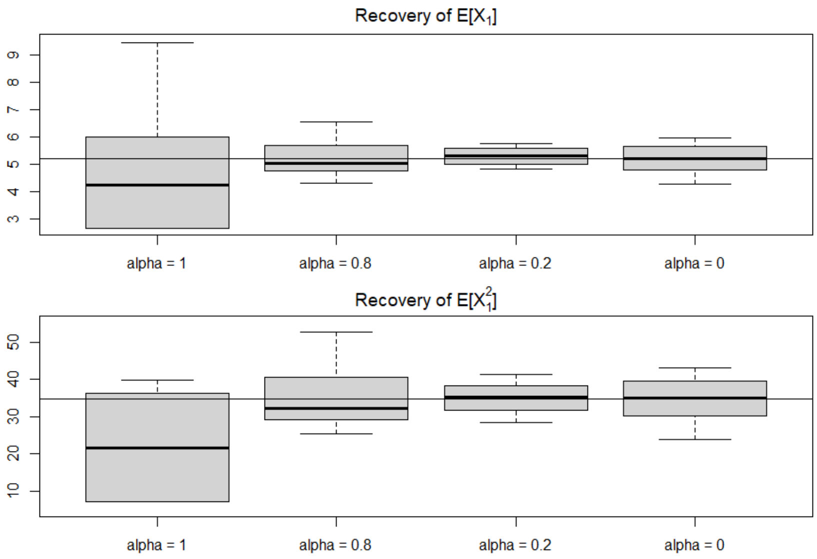

Figure 1: Recovery of the first and second moments of X_1 for the Gaussian mixture model

β from 0 to 1 can be used to construct this estimate or a prior run of the Zig-Zag with an uniformed choice $\kappa(\beta) \propto 1$ may be used to obtain these samples. We may then fit the regression model

$$\widehat{\log Z}(\beta) \approx \log \kappa(\beta),$$

to specify the polynomial terms in $\kappa(\beta)$.

In Example 1 (Gaussian mixture model), all methods were run for 50,000 events and the first 40% was used as burnin and tuning of κ. In the initial, burnin sampling we specified $\kappa(\beta) \propto 1$ with $\alpha = 0$. The tempered Zig-Zag samplers then used the events from this burnin process to construct an estimate of $\widehat{\log Z}(\beta)$. The samplers were then run for the remaining 30,000 event times and the estimated first and second moments were recorded.

In Example 2 (the transdimensional example), we fix $\kappa(\beta) = 1$ because the marginal distribution for the inverse temperature β was sufficiently close to uniformly distributed.

In Example 3 (Boltzmann machine relaxation), a finite grid of 15 β values equally spaced from 0.01 to 0.99 were used to form the construction of $\widehat{\log Z}(\beta)$. The associated choice of κ was used for all continuously tempered methods. For $\alpha = 0$ tempering with importance sampling, we used $\kappa(\beta) = \log \xi^{1-\beta}$ where ξ was specified using a variational Gaussian approximation to the target as in [8] and [9].

6.3 Computational resources

All experiments were implemented using the code accompanying the supplementary material. The multiple runs required for the simulation study were implemented in parallel using high performance computational resources. This amounted to submitting job requests for each individual replicate of the simulation studies. In each replicate the methods were given the same amount of computational resources i.e. simulated event-times. The results of the parallel runs were collected and processed to evaluate the performance of the methods — i.e. calculation of the average root mean square error.

References

[1] 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.

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

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

[4] 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.

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

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

[7] 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.

[8] 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.

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

[10] F. Pagani, A. Chevallier, S. Power, T. House, and S. Cotter. NuZZ: numerical Zig-Zag sampling for general models, 2020. `https://arxiv.org/abs/2003.03636`.

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