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 Chevallier et al. [2021, 2020], 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 Chevallier et al. [2021] and follow the proof of Theorem 3 of Chevallier et al. [2021] 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.

Remark 1 (Extension to other PDMP samplers) *Remark 1 from the main text states that tempered version of any PDMP-based sampler can be constructed more generally by adding a tempering variable β with associated velocity ± 1 and Zig-Zag rate function (irrespective of the base PDMP). When β hits 1, the process reduces to the PDMP sampler on π and is reintroduced with rate given by that of Theorem 1.*

Assuming that the base PDMP velocity space is $\mathcal{V}$ with associated probability $p_{\mathcal{V}}(v)$, we highlight the changes needed for the proof of Theorem 1 below.

1. *Change E_0 to $E_0 = \mathbb{R}^d \times [0, 1) \times \mathcal{V} \times \{-1, 1\}$,*

Preprint. Under review.

2. *Change the boundaries to* $B^{in} = \mathbb{R}^d \times \{1\} \times \mathcal{V} \times \{-1\}$ *and* $B^{out} = \mathbb{R}^d \times \{1\} \times \mathcal{V} \times \{1\}$,

3. *Prove that the assumption of Theorem 3 of Chevallier et al. [2021] is still valid for the probability distribution* $q(\boldsymbol{x}, \beta)p(\beta)p_{\mathcal{V}}(v)d\boldsymbol{x}d\beta dv$.

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

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 $\boldsymbol{z}_{t+s} = (\boldsymbol{x}_t + s\boldsymbol{v}_t, \boldsymbol{v}_t)$, thus the rate until the next event of type i will be

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

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 $\tilde{\lambda}_i(s)/\lambda_i^+(s)$.

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 [Bierkens et al., 2020, 2021, Bouchard-Côté et al., 2018, Chevallier et al., 2020] to construct thinning bounds and is repeated here for completeness — further details may be found in the derivation from Section 3.3 of Bierkens et al. [2019].

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 Pagani et al. [2020].

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 Sutton and Fearnhead [2021].

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 Sutton and Fearnhead [2021]. Following Chevallier et al. [2020] 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 Sutton and Fearnhead [2021].

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

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).

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 $\alpha \neq 0$.

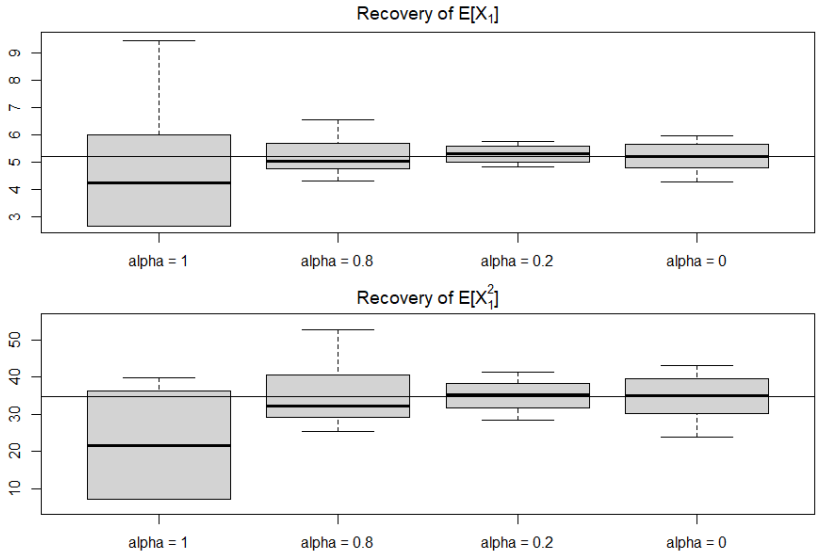

Figure 1: Recovery of the first and second moments of X_1 for the Gaussian mixture model

6.2 Root mean square error for Examples 4.1 and 4.3

The methods presented in Table 1 and 3 of the paper present the results in terms of a work normalised efficiency metric. This measure aims to capture the computational efficiency of the thinning procedure and any post processing evaluations for the Importance Sampling method. However, the metric is slightly unnatural compared to directly observing the root mean square error (RMSE). Table 3 and Table 4 below show the RMSE of the tempered approaches when they are run for the same number of event proposals. Since all methods are run for the same number of iterations, these tables also

indicate how the samplers would perform if the thinning bounds were exact (i.e. thinning efficiency = 1).

Table 3: 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**

Table 4: 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 $\mathbb{E}[X_k]$	$\mathbb{E}[X_k^2]$	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**

6.3 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 Gelman and Meng [1998]. For our experiments, we use numerical integration as described in Section 2.3 of Gelman and Meng [1998] further details may be found therein. We use the trapezoidal rule to estimate

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

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 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 being 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 Graham and Storkey [2017] and Nemeth et al. [2019].

6.4 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 — e.g. calculation of the average root mean square error.

7 Full simulation results for parallel tempering

Tables 5 and 6 provide further simulations. We compare our tempering approach with both reversible [Woodard et al., 2009] and non-reversible [Syed et al., 2022] parallel tempering (denoted R-PT and NR-PT respectively). As in the main paper, we report the work normalised efficiency relative to the standard (untempered) Zig-Zag. A Zig-Zag kernel is used at each temperature level and run for $S = 0.1, 1, 2$ units of stochastic time. We use a geometric temperature sequence $[1, a^1, a^2, ..., a^n]$ as commonly recommended in the literature [Tawn et al., 2020] and consider results for $a = 0.1, 0.3, 0.5, 0.7$, with $n = 3, 5, 7$.

Table 5: Parallel tempering full results Gaussian mixture model Example 4.1 (averaged over 20 replications).

				Relative work normalised efficiency			
Method	a	n	S	$\mathbb{E}[X_1]$	$\mathbb{E}[X_2]$	$\mathbb{E}[X_1^2]$	$\mathbb{E}[X_2^2]$
NR-PT	0.1	3	0.100	4.553	5.277	4.916	6.019
NR-PT	0.1	5	0.100	7.274	6.312	7.375	7.425
NR-PT	0.1	7	0.100	7.205	5.815	8.138	6.710
NR-PT	0.1	3	1	8.318	8.478	8.692	10.022
NR-PT	0.1	5	1	6.641	5.694	6.900	6.784
NR-PT	0.1	7	1	9.160	11.029	9.441	12.883
NR-PT	0.1	3	2	7.076	8.207	7.683	9.261
NR-PT	0.1	5	2	9.253	7.327	9.313	8.806
NR-PT	0.1	7	2	7.174	6.247	7.510	6.829
NR-PT	0.3	3	0.100	6.537	7.980	7.187	8.828
NR-PT	0.3	5	0.100	7.793	6.436	7.737	7.728
NR-PT	0.3	7	0.100	7.891	7.950	8.252	9.379
NR-PT	0.3	3	1	11.019	8.653	12.538	9.723
NR-PT	0.3	5	1	11.626	14.757	11.679	15.972
NR-PT	0.3	7	1	10.527	11.776	10.306	12.310
NR-PT	0.3	3	2	10.766	9.288	11.287	11.112
NR-PT	0.3	5	2	9.198	11.156	9.595	13.853
NR-PT	0.3	7	2	11.071	10.914	11.453	12.639
NR-PT	0.5	3	0.100	7.825	4.428	7.994	5.304
NR-PT	0.5	5	0.100	6.225	7.583	6.470	8.010
NR-PT	0.5	7	0.100	7.196	6.453	8.189	7.469
NR-PT	0.5	3	1	7.651	8.540	7.589	10.570
NR-PT	0.5	5	1	9.456	11.155	10.838	12.371
NR-PT	0.5	7	1	13.003	9.564	13.675	11.428
NR-PT	0.5	3	2	10.801	7.776	11.185	9.179
NR-PT	0.5	5	2	11.072	12.320	11.352	13.951
NR-PT	0.5	7	2	8.758	8.534	9.726	9.279
NR-PT	0.7	3	0.100	2.345	1.245	2.468	1.368
NR-PT	0.7	5	0.100	3.263	4.011	3.201	4.682
NR-PT	0.7	7	0.100	8.700	4.785	9.295	5.654
NR-PT	0.7	3	1	2.590	1.639	2.632	1.865
NR-PT	0.7	5	1	6.307	5.965	6.501	6.751
NR-PT	0.7	7	1	12.185	8.942	12.956	10.512
NR-PT	0.7	3	2	3.198	1.506	2.903	1.804
NR-PT	0.7	5	2	7.265	7.018	7.648	8.009
NR-PT	0.7	7	2	10.794	8.230	11.100	9.004

				Relative work normalised efficiency			
Method	a	n	S	$\mathbb{E}[X_1]$	$\mathbb{E}[X_2]$	$\mathbb{E}[X_1^2]$	$\mathbb{E}[X_2^2]$
R-PT	0.1	3	0.100	5.191	4.728	5.602	5.364
R-PT	0.1	5	0.100	6.680	5.138	6.815	6.096
R-PT	0.1	7	0.100	5.360	4.838	5.526	5.798
R-PT	0.1	3	1	5.944	6.078	5.889	7.531
R-PT	0.1	5	1	7.142	6.750	7.382	7.448
R-PT	0.1	7	1	6.042	5.808	6.499	6.549
R-PT	0.1	3	2	6.734	5.720	6.707	6.661
R-PT	0.1	5	2	5.045	5.899	5.278	6.534
R-PT	0.1	7	2	5.451	5.622	5.661	6.299
R-PT	0.3	3	0.100	5.036	6.373	5.388	7.655
R-PT	0.3	5	0.100	5.873	7.242	6.314	8.069
R-PT	0.3	7	0.100	6.917	6.389	7.249	7.889
R-PT	0.3	3	1	8.894	7.108	9.703	8.115
R-PT	0.3	5	1	9.549	7.645	9.912	8.949
R-PT	0.3	7	1	8.337	7.743	8.489	9.521
R-PT	0.3	3	2	8.405	6.953	9.049	7.906
R-PT	0.3	5	2	5.995	5.915	6.072	6.906
R-PT	0.3	7	2	7.861	5.312	7.615	6.860
R-PT	0.5	3	0.100	6.409	4.737	6.401	5.312
R-PT	0.5	5	0.100	7.721	7.053	8.220	8.165
R-PT	0.5	7	0.100	5.898	9.587	5.922	10.642
R-PT	0.5	3	1	10.136	7.672	10.830	8.842
R-PT	0.5	5	1	8.991	9.408	9.051	9.825
R-PT	0.5	7	1	8.295	8.156	7.915	9.893
R-PT	0.5	3	2	8.114	6.149	8.368	6.875
R-PT	0.5	5	2	9.742	8.316	10.369	9.118
R-PT	0.5	7	2	8.658	7.226	8.794	9.183
R-PT	0.7	3	0.100	3.059	1.170	2.823	1.390
R-PT	0.7	5	0.100	4.436	2.945	4.425	3.594
R-PT	0.7	7	0.100	6.603	4.919	6.343	6.122
R-PT	0.7	3	1	2.611	1.721	2.464	2.036
R-PT	0.7	5	1	8.036	6.063	8.068	7.087
R-PT	0.7	7	1	12.275	11.273	12.591	12.648
R-PT	0.7	3	2	3.131	1.758	3.098	2.021
R-PT	0.7	5	2	6.550	6.102	6.468	7.096
R-PT	0.7	7	2	8.885	7.927	9.025	9.582

Table 6: Parallel tempering full results Boltzman Machine Example 4.3 (averaged over 20 replications).

Method	a	n	S	Relative work normalised efficiency $\mathbb{E}[X_k]$	$\mathbb{E}[X_k^2]$
NR-PT	0.1	3	0.100	0.262	0.089
NR-PT	0.1	5	0.100	0.288	0.086
NR-PT	0.1	7	0.100	0.273	0.085
NR-PT	0.1	3	1	1.726	0.587
NR-PT	0.1	5	1	1.966	0.577
NR-PT	0.1	7	1	2.180	0.717
NR-PT	0.1	3	2	2.856	0.843
NR-PT	0.1	5	2	3.220	0.761
NR-PT	0.1	7	2	3.380	1.051
NR-PT	0.3	3	0.100	0.310	0.105
NR-PT	0.3	5	0.100	0.453	0.147
NR-PT	0.3	7	0.100	0.456	0.142
NR-PT	0.3	3	1	2.079	0.684
NR-PT	0.3	5	1	2.583	0.838
NR-PT	0.3	7	1	2.786	0.781
NR-PT	0.3	3	2	3.589	1.051
NR-PT	0.3	5	2	4.171	1.338
NR-PT	0.3	7	2	4.454	1.259
NR-PT	0.5	3	0.100	0.239	0.154
NR-PT	0.5	5	0.100	0.443	0.137
NR-PT	0.5	7	0.100	0.598	0.198
NR-PT	0.5	3	1	0.957	0.776
NR-PT	0.5	5	1	3.261	1.029
NR-PT	0.5	7	1	3.215	0.890
NR-PT	0.5	3	2	1.723	1.297
NR-PT	0.5	5	2	5.634	1.448
NR-PT	0.5	7	2	6.101	1.434
NR-PT	0.7	3	0.100	0.250	0.146
NR-PT	0.7	5	0.100	0.302	0.265
NR-PT	0.7	7	0.100	0.379	0.278
NR-PT	0.7	3	1	0.721	0.664
NR-PT	0.7	5	1	0.956	1.080
NR-PT	0.7	7	1	2.489	1.127
NR-PT	0.7	3	2	1.132	1.270
NR-PT	0.7	5	2	1.367	1.606
NR-PT	0.7	7	2	4.391	1.924

Method	a	n	S	Relative work normalised efficiency $\mathbb{E}[X_k]$	$\mathbb{E}[X_k^2]$
R-PT	0.1	3	0.100	0.262	0.090
R-PT	0.1	5	0.100	0.290	0.098
R-PT	0.1	7	0.100	0.283	0.092
R-PT	0.1	3	1	1.462	0.485
R-PT	0.1	5	1	1.536	0.452
R-PT	0.1	7	1	1.729	0.461
R-PT	0.1	3	2	2.619	0.814
R-PT	0.1	5	2	2.796	0.795
R-PT	0.1	7	2	2.883	1.083
R-PT	0.3	3	0.100	0.307	0.109
R-PT	0.3	5	0.100	0.422	0.124
R-PT	0.3	7	0.100	0.474	0.152
R-PT	0.3	3	1	1.956	0.581
R-PT	0.3	5	1	1.994	0.564
R-PT	0.3	7	1	3.311	0.817
R-PT	0.3	3	2	4.127	1.141
R-PT	0.3	5	2	3.877	1.152
R-PT	0.3	7	2	3.673	1.055
R-PT	0.5	3	0.100	0.222	0.121
R-PT	0.5	5	0.100	0.368	0.134
R-PT	0.5	7	0.100	0.499	0.145
R-PT	0.5	3	1	1.106	0.865
R-PT	0.5	5	1	2.683	0.789
R-PT	0.5	7	1	3.890	0.976
R-PT	0.5	3	2	1.610	1.492
R-PT	0.5	5	2	5.201	1.513
R-PT	0.5	7	2	5.321	1.560
R-PT	0.7	3	0.100	0.251	0.142
R-PT	0.7	5	0.100	0.271	0.176
R-PT	0.7	7	0.100	0.331	0.203
R-PT	0.7	3	1	0.739	0.816
R-PT	0.7	5	1	0.928	0.671
R-PT	0.7	7	1	2.410	1.209
R-PT	0.7	3	2	1.117	1.331
R-PT	0.7	5	2	1.560	1.473
R-PT	0.7	7	2	4.402	1.984

References

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.

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.

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

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.

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

S. Syed, A. Bouchard-Côté, G. Deligiannidis, and A. Doucet. Non-reversible parallel tempering: A scalable highly parallel mcmc scheme. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 84(2):321–350, 2022. doi: https://doi.org/10.1111/rssb.12464. URL https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/rssb.12464.

N. G. Tawn, G. O. Roberts, and J. S. Rosenthal. Weight-preserving simulated tempering. *Statistics and Computing*, 30(1):27–41, Feb 2020. ISSN 1573-1375. doi: 10.1007/s11222-019-09863-3. URL https://doi.org/10.1007/s11222-019-09863-3.

D. Woodard, S. Schmidler, and M. Huber. Sufficient conditions for torpid mixing of parallel and simulated tempering. *Electron. J. Probab.*, 14:780–804, 2009. ISSN 1083-6489. doi: 10.1214/EJP.v14-638. URL http://ejp.ejpecp.org/article/view/638.