

# Non-Reversible Parallel Tempering: a Scalable Highly Parallel MCMC Scheme

Saifuddin Syed\*, Alexandre Bouchard-Côté\*, George Deligiannidis†, Arnaud Doucet†

July 28, 2021

## Abstract

Parallel tempering (PT) methods are a popular class of Markov chain Monte Carlo schemes used to sample complex high-dimensional probability distributions. They rely on a collection of  $N$  interacting auxiliary chains targeting tempered versions of the target distribution to improve the exploration of the state-space. We provide here a new perspective on these highly parallel algorithms and their tuning by identifying and formalizing a sharp divide in the behaviour and performance of reversible versus non-reversible PT schemes. We show theoretically and empirically that a class of non-reversible PT methods dominates its reversible counterparts and identify distinct scaling limits for the non-reversible and reversible schemes, the former being a piecewise-deterministic Markov process and the latter a diffusion. These results are exploited to identify the optimal annealing schedule for non-reversible PT and to develop an iterative scheme approximating this schedule. We provide a wide range of numerical examples supporting our theoretical and methodological contributions. The proposed methodology is applicable to sample from a distribution  $\pi$  with a density  $L$  with respect to a reference distribution  $\pi_0$  and compute the normalizing constant  $\int L \, d\pi_0$ . A typical use case is when  $\pi_0$  is a prior distribution,  $L$  a likelihood function and  $\pi$  the corresponding posterior distribution.

## 1 Introduction

Markov Chain Monte Carlo (MCMC) methods are widely used to approximate expectations with respect to a probability distribution with density  $\pi(x)$  known up to a normalizing constant, i.e.,  $\pi(x) = \gamma(x)/\mathcal{Z}$  where  $\gamma$  can be evaluated pointwise but the normalizing constant  $\mathcal{Z}$  is unknown. When  $\pi$  has multiple well-separated modes, highly varying curvature or when one is interested in sampling over combinatorial spaces, standard MCMC algorithms can perform very poorly. This work is motivated by the need for practical methods for these difficult sampling problems. A natural direction to address them is to use multiple cores and to distribute the computation.

---

\*Department of Statistics, University of British Columbia, Canada.

†Department of Statistics, University of Oxford, UK.

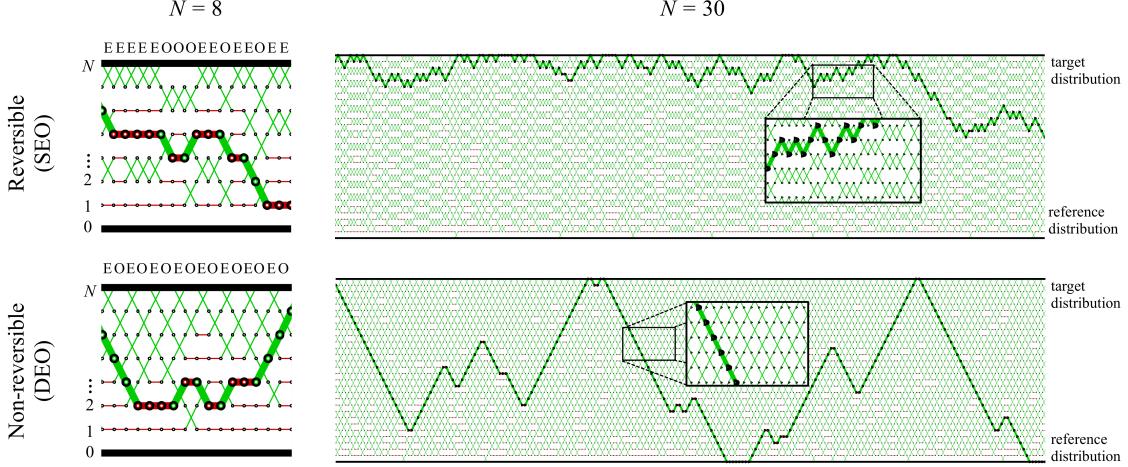


Figure 1: Reversible (top) and non-reversible (bottom) PT for  $N = 8$  (left) and  $N = 30$  auxiliary chains (right) using equally spaced annealing parameters on a Bayesian change-point detection model [Davidson-Pilon, 2015] where  $\pi_0$  is the prior,  $\pi$  the posterior. The sequence of swap moves forms  $N + 1$  index process trajectories (paths formed by the red and green edges). We show one such path in bold. The reversible and non-reversible PT clearly exhibit different scaling behaviour which we formalize in Section 6.

## 1.1 Parallel Tempering

One popular approach for multi-core and distributed exploration of complex distributions is Parallel Tempering (PT) which was introduced independently in statistics [Geyer, 1991] and physics [Hukushima and Nemoto, 1996]; see also [Swendsen and Wang, 1986] for an earlier related proposal. Since its inception, PT remains to this day a very popular MCMC method to sample from complex multimodal target distributions arising in physics, chemistry, biology, statistics, and machine learning; see, e.g., [Issaoun et al., 2021, Ballnus et al., 2017, Chandra et al., 2019, Cho et al., 2010, Desjardins et al., 2014, Diaz et al., 2020, Dorri et al., 2020, Kamberaj, 2020, Müller and Bouckaert, 2020].

To sample from the target distribution  $\pi$ , PT introduces a sequence of auxiliary *tempered* or *annealed* probability distributions with densities  $\pi^{(\beta_i)}(x) \propto L(x)^{\beta_i} \pi_0(x)$  for  $i = 0, 1, \dots, N$ , where  $\pi_0$  is an easy-to-sample reference distribution,  $L(x) = \pi(x)/\pi_0(x)$  and the sequence  $0 = \beta_0 < \beta_1 < \dots < \beta_N = 1$  defines the *annealing schedule*. This bridge of auxiliary distributions is used to progressively transform samples from the *reference distribution* ( $\beta = 0$ ) into samples from the *target distribution* ( $\beta = 1$ ), for which only poorly mixing MCMC kernels may be available. For example, in the Bayesian setting where the target distribution is the posterior, we can choose the reference distribution as the prior, from which we can often obtain independent samples.

More precisely, PT algorithms are based on Markov chains in which the states are  $(N + 1)$ -tuples,  $\mathbf{x} = (x^0, x^1, x^2, \dots, x^N) \in \mathcal{X}^{N+1}$ , and whose stationary distribution is given by  $\boldsymbol{\pi}(\mathbf{x}) = \prod_{i=0}^N \pi^{(\beta_i)}(x^i)$  [Geyer, 1991]. At each iteration, PT proceeds by applying in parallel  $N + 1$  MCMC kernels targeting  $\pi^{(\beta_i)}$  for  $i = 0, \dots, N$ . We call these model-specific kernels the *local exploration kernels*. The chains closer to the reference chain (i.e. those with annealing parameter  $\beta_i$  close to zero) can typically traverse regions of low probability mass under  $\pi$  while the chain at  $\beta = 1$  ensures that

asymptotically we obtain samples from the target distribution. Frequent communication between the chains at the two ends of the spectrum is therefore critical for good performance, and achieved by proposing to swap the states of chains at adjacent annealing parameters. *Even swap* moves (rows labelled ‘E’), respectively *Odd swap* moves (labelled ‘O’), propose to exchange states at chains with an even index  $i$ , respectively odd index  $i$ , and  $i + 1$ . These proposals are accepted or rejected according to a Metropolis mechanism.

## 1.2 Deterministic and stochastic even-odd schemes

A key notion used to analyze the behaviour of PT is the *index process*. To provide intuition on this process, it is helpful to discuss briefly how PT is distributed over several machines. An important point is that instead of having pairs of machines exchanging high-dimensional states when a swap is accepted (which could be detrimental due to network latency), the machines should just exchange the annealing parameters. Suppose now we initialize machine  $j$  with annealing parameter  $\beta_j$ . Then after  $n$  scans, the annealing parameters are permuted among the  $N + 1$  machines according to the permutation  $\mathbf{I}_n = (I_n^0, \dots, I_n^N)$  of  $\{0, \dots, N\}$ , so that machine  $j$  has annealing parameter  $\beta_{I_n^j}$  at iteration  $n$  of PT. Each *index process*  $I_n^j$ , formally introduced in Section 2.4, is initialized using  $I_0^j = j$  and tracks how the state of the corresponding chain evolves over the annealing schedule thanks to the swap moves; see Figure 1. The index process  $I_n^j$  thus monitors the information transfer between the reference and target on machine  $j$  and as such determines partly the effectiveness of PT.

There have been many proposals made to improve this information transfer by adjusting the annealing schedule; see, e.g., [Kone and Kofke, 2005, Atchadé et al., 2011, Miasojedow et al., 2013]. These proposals are useful but do not address a crucial limitation of standard PT algorithms. In a distributed context, one can select randomly at each iteration whether to apply Even or Odd swap moves in parallel. The resulting stochastic even-odd swap (SEO) scheme, henceforth referred to as *reversible PT* as it admits a reversible scaling limit (see Section 6), yields index processes exhibiting a diffusive behaviour; see top row of Figure 1. Hence we can expect that when  $N$  is large it takes roughly  $O(N^2)$  swap attempts for a state at  $\beta_0 = 0$  to reach  $\beta_N = 1$  [Diaconis et al., 2000]. The user thus faces a trade-off. If  $N$  is too large, the acceptance probabilities of the swap moves are high but it takes a time of order  $O(N^2)$  for a state at  $\beta = 0$  to reach  $\beta = 1$ . If  $N$  is too low, the acceptance probabilities of swap moves deteriorate resulting in poor mixing between the different chains. Informally, even in a multi-core or distributed setting, for  $N$  large, the  $O(N)$  gains in being able to harness more cores do not offset the  $O(N^2)$  cost of the diffusion (see Section 3.5 where we formalize this argument). As a consequence, the general consensus is that the temperatures should be chosen to allow for about a 20–40% acceptance rate to maximize the squared jump distance travelled per swap in the space of annealing parameters  $[0, 1]$  [Rathore et al., 2005, Kone and Kofke, 2005, Lingenheil et al., 2009, Atchadé et al., 2011]. Adding more chains past this threshold actually deteriorates the performance of reversible PT and there have even been attempts to adaptively reduce the number of additional chains [Lacki and Miasojedow, 2016]. This

is a lost opportunity, since PT is otherwise particularly suitable to implementation on multi-core or distributed architectures.

An alternative to the SEO scheme is the deterministic even-odd swap (DEO) scheme introduced in [Okabe et al., 2001] where one deterministically alternates Even and Odd swap moves. We refer to DEO as *non-reversible PT* as it admits a non-reversible scaling limit (see Section 6). In particular, the resulting index processes do not appear to exhibit a diffusive, i.e. random walk type, behaviour, illustrated by the bottom row of Figure 1. This non-diffusive behaviour of non-reversible PT explains its excellent empirical performance when compared to alternative reversible PT schemes observed in practice [Lingenheil et al., 2009] for any given schedule. However non-reversible PT, like reversible PT, is sensitive to the choice of the schedule. All the aforementioned tuning strategies developed for PT implicitly assume a reversible framework and do not apply to non-reversible PT (as empirically verified in [Lingenheil et al., 2009]). The main contribution of this paper is to identify some of the theoretical properties of non-reversible PT so as to establish optimal tuning guidelines for this algorithm and propose a novel schedule optimization scheme to implement them. In all our experiments, the resulting iteratively optimized non-reversible PT scheme markedly outperforms the reversible and non-reversible PT schemes currently available (see Section 7).

### 1.3 Overview of our contributions

After introducing formally the SEO and DEO schemes in Section 2, our first contribution is a non-asymptotic result showing that the non-reversible DEO scheme is guaranteed to outperform its reversible SEO counterpart. The notion of optimality we analyze is the *round trip rate*, which quantifies how often information from the reference distribution percolates to the target; see Section 3. The theoretical analysis is based on a simplifying assumption called Efficient Local Exploration (ELE), which is not expected to hold exactly in real scenarios. However we show empirically that there are practical methods to approximate ELE and that even when ELE is violated the key predictions made by the theory closely match empirical behaviour. In this sense ELE can be thought of as a useful *model* for understanding PT algorithms.

In Section 4 we introduce the local and global communication barrier  $\lambda(\beta), \Lambda$  which encode the local and global efficiency of PT. We then show that for non-reversible PT the round trip rate converges to  $(2 + 2\Lambda)^{-1}$ , in contrast to the reversible counterpart for which it decays to zero. To provide some intuition on how the small algorithmic difference between SEO and DEO can have such a profound impact, consider the scenario where PT would use the same distributions at the two end-points,  $\pi = \pi_0$ , as well as for all intermediate distributions so that  $\Lambda = 0$ . Clearly this is not a realistic scenario, but in this simple context the index processes of DEO and SEO are easy to describe and contrast. In both DEO and SEO,  $\pi = \pi_0$  implies that all proposed swaps will be accepted. For DEO, this makes the index process fully deterministic, performing direct trips from index 0 to  $N$  and back. Such a process could be compared to a “conveyor belt” with the property that no matter what is the value of  $N$ , one novel trip from chain 0 reaches chain  $N$  every two

iterations, i.e. the round trip rate is  $(2 + 2\Lambda)^{-1} = 1/2$  as  $\Lambda = 0$ . For SEO, even when  $\pi = \pi_0$  the index process is still random and can be readily seen to be a simple discrete random walk. This implies that as  $N$  increases, the round trip rate decreases to zero.

In practice, achieving high round trip rates requires careful tuning of the annealing schedule. In Section 5 we combine the analysis from Section 3 and Section 4 to develop a novel methodology to optimize the annealing parameters. The optimal tuning guidelines we provide are different from existing (reversible) PT guidelines and the novel methodology is highly parallel as its performance does not collapse when a very large number of chains is used. However using a large number of chains does have a diminishing return, therefore we propose a mechanism to determine the optimal trade-off between the number of chains and the number of independent PT algorithms one should use.

In Section 6 we identify the scaling limit of the index processes for both reversible and non-reversible PT as the number of parallel chains goes to infinity. We show that this scaling limit is a piecewise-deterministic Markov process for non-reversible PT whereas it is a diffusion for reversible PT as suggested by the dynamics of the bold paths in Figure 1.

Finally in Section 7, we present a variety of experiments validating our theoretical analysis and novel methodology. The method is implemented in an open source Bayesian modelling language available at <https://github.com/UBC-Stat-ML/blangSDK>. Our software implementation allows the user to specify the model in BUGS-like language [Lunn et al., 2000]. From this model declaration, a suitable sequence of annealed distributions is instantiated and a schedule optimized using our iterative method.

## 2 Setup and notation

### 2.1 Parallel tempering

Henceforth we will assume that the *target* and *reference* probability distributions  $\pi$  and  $\pi_0$  on  $\mathcal{X}$  admit strictly positive densities with respect to a common dominating measure  $dx$ . We will also denote these densities somewhat abusively by  $\pi$  and  $\pi_0$ . It will be useful to define  $V_0(x) = -\log \pi_0(x)$  and  $V(x) = -\log L(x)$ , where  $L(x) = \pi(x)/\pi_0(x)$  is assumed finite for all  $x \in \mathcal{X}$ . Using this notation, the *annealed distribution* at an annealing parameter  $\beta$  is given by

$$\pi^{(\beta)}(x) = \frac{L(x)^\beta \pi_0(x)}{\mathcal{Z}(\beta)} = \frac{e^{-\beta V(x) - V_0(x)}}{\mathcal{Z}(\beta)}, \quad (1)$$

where  $\mathcal{Z}(\beta) = \int_{\mathcal{X}} L(x)^\beta \pi_0(x) dx$  is the corresponding normalizing constant. We denote the *annealing schedule* by  $0 = \beta_0 < \beta_1 < \dots < \beta_N = 1$ . In our analysis we will view it as a partition  $\mathcal{P}_N = \{\beta_0, \dots, \beta_N\}$  of  $[0, 1]$  with mesh-size  $\|\mathcal{P}_N\| = \sup_i \{\beta_i - \beta_{i-1}\}$ .

We define the PT state  $\bar{\mathbf{x}} = (\mathbf{x}, \mathbf{i})$  where  $\mathbf{x} = (x^0, \dots, x^N) \in \mathcal{X}^{N+1}$  and  $\mathbf{i} = (i^0, \dots, i^N) \in \text{Perm}(N)$  the group of permutations on  $\{0, \dots, N\}$ . PT involves constructing a Markov Chain  $\bar{\mathbf{X}}_n = (\mathbf{X}_n, \mathbf{I}_n)$  over  $\mathcal{X}^{N+1} \times \text{Perm}(N)$  invariant with respect to  $\bar{\pi}(\bar{\mathbf{x}}) = (N+1)!^{-1} \prod_{i=0}^N \pi^{(\beta_i)}(x^i)$ .

The  $i$ -th component of  $\mathbf{X}_n = (X_n^0, \dots, X_n^N)$  tracks the states associated with annealing parameter  $\beta_i$  at iteration  $n$  while the permutation  $\mathbf{I}_n = (I_n^0, \dots, I_n^N)$  tracks how the annealing parameters are shuffled among machines. In particular at iteration  $n$ , machine  $j$  stores the  $I_n^j$ -th component of  $\mathbf{X}_n$  and annealing parameter  $\beta_{I_n^j}$ . The sequence of states associated with  $\beta_i$  is called the  $i$ -th *chain* and the state on machine  $j$  is the  $j$ -th *replica*.

For both SEO and DEO, the overall  $\bar{\pi}$ -invariant Markov kernel  $\mathbf{K}_n^{\text{PT}}$  describing the algorithm is obtained by the composition of a  $\bar{\pi}$ -invariant local exploration kernel  $\mathbf{K}^{\text{expl}}$  and communication kernel  $\mathbf{K}_n^{\text{comm}}$ ,

$$\mathbf{K}_n^{\text{PT}}(\bar{\mathbf{x}}, A) = \mathbf{K}_n^{\text{comm}} \mathbf{K}^{\text{expl}}(\bar{\mathbf{x}}, A) := \int \mathbf{K}_n^{\text{comm}}(\bar{\mathbf{x}}, d\bar{\mathbf{x}}') \mathbf{K}^{\text{expl}}(\bar{\mathbf{x}}', A). \quad (2)$$

The difference between SEO and DEO is in the communication phase, namely  $\mathbf{K}_n^{\text{comm}} = \mathbf{K}^{\text{SEO}}$  in the former case and  $\mathbf{K}_n^{\text{comm}} = \mathbf{K}_n^{\text{DEO}}$  in the latter. Markov kernels corresponding to the reversible (SEO) and non-reversible (DEO) PT algorithms are described informally in the introduction and illustrated in Figure 1.

## 2.2 Local exploration kernels

The local exploration kernels are defined in the same way for SEO and DEO. They are also model specific, so we assume we are given one  $\pi^{(\beta_i)}$ -invariant kernel  $K^{(\beta_i)}$  for each annealing parameter  $\beta_i \in \mathcal{P}_N$ . These can be based on  $n_{\text{expl}}$  steps of Hamiltonian Monte Carlo, Metropolis–Hastings, Gibbs Sampling, Slice Sampling, etc. We construct the overall local exploration kernel by applying the annealing parameter specific kernels to each component independently from each other:

$$\mathbf{K}^{\text{expl}}((\mathbf{x}, \mathbf{i}), A_0 \times A_1 \times \dots \times A_N \times \{\mathbf{i}'\}) = \prod_{i=0}^N K^{(\beta_i)}(x^i, A_i) \delta_{\mathbf{i}}(\mathbf{i}'), \quad (3)$$

where  $\delta_{\mathbf{i}}$  denotes the Dirac delta.

In our computational model, we implicitly assume that the local exploration kernel at  $\beta = 0$  is special in that it can provide independent exact samples from  $\pi_0$ . Mathematically,  $K^{(0)}(x, A_0) = \pi_0(A_0)$ . This assumption is satisfied in most Bayesian models equipped with proper prior distributions, but also in other situations such as Markov random fields (see Appendix I.6).

## 2.3 Communication kernels.

Before defining the communication scheme, we first construct its fundamental building block, a *swap*. A swap is a Metropolis–Hastings move with a deterministic proposal which consists of swapping  $\beta_i$  and  $\beta_{i+1}$  across machines. From a current state  $\bar{\mathbf{x}} = (\mathbf{x}, \mathbf{i})$ , a proposed swap state is denoted  $\bar{\mathbf{x}}^{(i,i+1)} = (\mathbf{x}^{(i,i+1)}, \mathbf{i}^{(i,i+1)})$  where

$$\mathbf{x}^{(i,i+1)} = (x^0, \dots, x^{i-1}, x^{i+1}, x^i, x^{i+2}, \dots, x^N), \quad (4)$$

and  $\mathbf{i}^{(i,i+1)} \in \text{Perm}(N)$  is the permutation obtained by swapping  $i$  and  $i + 1$  in  $\mathbf{i}$ . The Metropolis–Hastings kernel  $\mathbf{K}^{(i,i+1)}$  corresponding to this update is given by

$$\mathbf{K}^{(i,i+1)}(\bar{\mathbf{x}}, \cdot) = (1 - \alpha^{(i,i+1)}(\bar{\mathbf{x}}))\delta_{\bar{\mathbf{x}}}(\cdot) + \alpha^{(i,i+1)}(\bar{\mathbf{x}})\delta_{\bar{\mathbf{x}}^{(i,i+1)}}(\cdot). \quad (5)$$

The function  $\alpha^{(i,i+1)}(\bar{\mathbf{x}})$  is the corresponding acceptance probability equal to

$$\alpha^{(i,i+1)}(\bar{\mathbf{x}}) = \min \left\{ 1, \frac{\bar{\pi}(\bar{\mathbf{x}}^{(i,i+1)})}{\bar{\pi}(\bar{\mathbf{x}})} \right\} = \exp \left( \min \{0, (\beta_{i+1} - \beta_i)(V(x^{i+1}) - V(x^i))\} \right) \quad (6)$$

and the maximal collection of adjacent swap kernels that can be proposed in parallel without interference are

$$\mathbf{K}^{\text{even}} = \prod_{i \text{ even}} \mathbf{K}^{(i,i+1)}, \quad \mathbf{K}^{\text{odd}} = \prod_{i \text{ odd}} \mathbf{K}^{(i,i+1)}, \quad (7)$$

which we call the *even* and *odd kernels* respectively.

For SEO, the kernel  $\mathbf{K}_n^{\text{comm}} = \mathbf{K}^{\text{SEO}}$  is given by a mixture of the even and odd kernels in equal proportion while for DEO the kernel  $\mathbf{K}_n^{\text{comm}} = \mathbf{K}_n^{\text{DEO}}$  is given by a deterministic alternation between even and odd kernels, that is

$$\mathbf{K}^{\text{SEO}} = \frac{1}{2}\mathbf{K}^{\text{even}} + \frac{1}{2}\mathbf{K}^{\text{odd}}, \quad \mathbf{K}_n^{\text{DEO}} = \begin{cases} \mathbf{K}^{\text{even}} & \text{if } n \text{ is even,} \\ \mathbf{K}^{\text{odd}} & \text{if } n \text{ is odd.} \end{cases} \quad (8)$$

We provide pseudo-code for the DEO scheme in Algorithm 1. The pseudo-code also estimates the average rejection probabilities  $r^{(i,i+1)}$  of swap moves between chains  $i$  and  $i + 1$  which are used to optimize the annealing schedule in Section 5. When the schedule is fixed, lines 1, 12, 17 can be omitted, and one should use “for  $i \in P$ ” on line 10. For simplicity, the swap in line 15 is shown for a *parallel computing* context, where several cores have a shared memory, and hence line 15 is simply an exchange of pointers in an array, which is efficient thanks to memory sharing. For a *distributed computing* implementation, where several machines do not share memory and instead need to communicate over the network, it becomes advantageous to swap annealing parameters instead of states. Refer to Appendix H for pseudo-code for the distributed computing implementation.

## 2.4 The index process

As discussed in Section 1.2, the  $j$ -th component of permutation  $\mathbf{I}_n = (I_n^0, \dots, I_n^N)$  encodes the flow of information between the reference and target on machine  $j$ . We define the *index process* for machine  $j$  as  $(I_n^j, \varepsilon_n^j) \in \{0, \dots, N\} \times \{-1, 1\}$ , where  $\varepsilon_n^j = 1$  if the annealing parameter on machine  $j$  is proposed an increase after scan  $n$ , and  $-1$  otherwise. The concept is best understood visually: refer to the bold piecewise linear paths illustrating  $I_n^{N-2}$  for  $N = 8$  and  $I_n^{N-1}$  for  $N = 30$  in Figure 1.

Let  $P^{(i,i+1)} \in \{0, 1\}$  denote an indicator that a swap is proposed between chains  $i$  and  $i + 1$

---

**Algorithm 1** DEO(number of scans  $n_{\text{scan}}$ , annealing schedule  $\mathcal{P}_N$ )

---

```

1:  $r^{(i,i+1)} \leftarrow 0$  for all  $i \in \{0, 1, \dots, N-1\}$             $\triangleright$  Swap rejection statistics used in Section 5.4 to
   optimize the schedule
2:  $\mathbf{x} \leftarrow \mathbf{x}_0$                                           $\triangleright$  Initialize chain
3: for  $n$  in  $1, 2, \dots, n_{\text{scan}}$  do
4:   if  $n$  is even then                                      $\triangleright$  Non-reversibility inducing alternation
5:      $P \leftarrow \{i : 0 \leq i < N, i \text{ is even}\}$            $\triangleright$  Even subset of  $\{0, \dots, N-1\}$ 
6:   else
7:      $P \leftarrow \{i : 0 \leq i < N, i \text{ is odd}\}$            $\triangleright$  Odd subset of  $\{0, \dots, N-1\}$ 
8:   for  $i$  in  $0, \dots, N$  do                                 $\triangleright$  Local exploration phase (parallelizable)
9:      $x^i \sim K^{(\beta_i)}(x_{n-1}^i, \cdot)$ 
10:    for  $i$  in  $0, \dots, N-1$  do                          $\triangleright$  Communication phase (parallelizable)
11:       $\alpha \leftarrow \alpha^{(i,i+1)}$                             $\triangleright$  Equation (6).
12:       $r^{(i,i+1)} \leftarrow r^{(i,i+1)} + (1 - \alpha)$ 
13:       $A \sim \text{Bern}(\alpha)$ 
14:      if  $i \in P$  and  $A = 1$  then
15:         $(x^i, x^{i+1}) \leftarrow (x^{i+1}, x^i)$             $\triangleright$  See also Appendix H for a distributed computing
   implementation, where annealing parameters are swapped instead of states.
16:     $\mathbf{x}_n \leftarrow \mathbf{x}$ 
17:     $r^{(i,i+1)} \leftarrow r^{(i,i+1)}/n_{\text{scan}}$  for all  $i \in \{0, 1, \dots, N-1\}$             $\triangleright$  Equation (31)
18: return  $(\mathbf{x}_1, \dots, \mathbf{x}_{n_{\text{scan}}}), (r^{(0,1)}, \dots, r^{(N-1,N)})$ 

```

---

at iteration  $n$ . The realized swaps are then defined from the proposal indicators as  $S^{(i,i+1)} = P^{(i,i+1)} A^{(i,i+1)}$ , where  $A^{(i,i+1)} | \bar{\mathbf{X}} \sim \text{Bern}(\alpha^{(i,i+1)}(\bar{\mathbf{X}}))$  are acceptance indicator variables (Figure 2 in Appendix J). The index process satisfies the following recursive relation: initialize  $I_0^j = j$  and  $\varepsilon_0^j = 1$  if  $P_0^{(j,j+1)} = 1$  and  $-1$  otherwise. For  $n > 0$ , we have

$$I_{n+1}^j = \begin{cases} I_n^j + \varepsilon_n^j & \text{if } S_n^{(I_n^j, I_n^j + \varepsilon_n^j)} = 1, \\ I_n^j & \text{otherwise,} \end{cases}, \quad \varepsilon_{n+1}^j = \begin{cases} 1 & \text{if } P_n^{(I_n^j, I_{n+1}^j + 1)} = 1, \\ -1 & \text{otherwise.} \end{cases} \quad (9)$$

We will use the dynamics of the index process to explain the differences between SEO and DEO communication. The only difference between the two is in the proposal indicators. Define  $\mathbf{P}_n = (P_n^{(0,1)}, P_n^{(1,2)}, \dots, P_n^{(N-1,N)})$ ,  $\mathbf{P}_n$  is deterministic for DEO, i.e.  $\mathbf{P}_n = \mathbf{P}_{\text{even}} = (1, 0, 1, \dots)$  for even  $n$  and  $\mathbf{P}_n = \mathbf{P}_{\text{odd}} = (0, 1, 0, \dots)$  for odd  $n$ . In SEO, we have  $\mathbf{P}_n \sim \text{Unif}\{\mathbf{P}_{\text{even}}, \mathbf{P}_{\text{odd}}\}$ .

For SEO, the variables  $\varepsilon_n \sim \text{Unif}\{-1, 1\}$  are i.i.d, and consequently the index process exhibits a random walk behaviour. In contrast for DEO, we have  $\varepsilon_{n+1}^j = \varepsilon_n^j$  so long as  $I_{n+1}^j = I_n^j + \varepsilon_n^j$  and  $\varepsilon_{n+1}^j = -\varepsilon_n^j$  otherwise. Therefore the index process for DEO performs a more systematic exploration of the space as the direction  $\varepsilon_{n+1}^j$  is only reversed when a swap involving machine  $j$  is rejected or if the boundary is reached. The qualitative differences between the two regimes can be seen in Figure 1 (see also Figure 15 in the Supplementary Material). In particular the index process for DEO in these figures behaves very differently as  $N$  increases, this will be explored formally in Section 6.

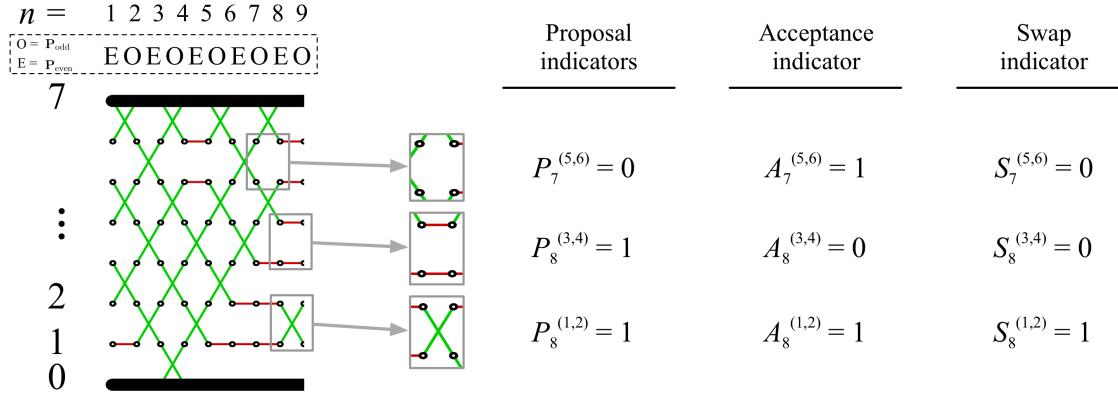


Figure 2: Illustration of the proposal, acceptance and swap indicators.

As mentioned in the introduction, we refer to the PT algorithm with SEO and DEO communication as *reversible* PT and *non-reversible* PT respectively. Our terminology is somewhat abusive but is justified by the analysis in Section 3.4 and Section 6, where it is shown that, under certain assumptions, the index process is reversible for SEO while it is non-reversible for DEO.

### 3 Non-asymptotic analysis of PT algorithms

#### 3.1 Model of compute time

We start with a definition of what we model as one unit of compute time: throughout the paper, we assume a massively parallel or distributed computational setup and sampling once from each of local exploration kernel  $K^{(\beta)}$  has cost  $O(n_{\text{expl}})$  which dominates the cost of the swap kernel  $\mathbf{K}^{(i,i+1)}$ . Consequently a scan of PT has cost  $O(n_{\text{expl}})$  and for a fixed computational budget  $n_{\text{total}}$ , the total number of scans is  $n_{\text{scan}} = O(n_{\text{total}}/n_{\text{expl}})$ .

The assumption that the per-iteration cost of PT is independent of the number of chains is reasonable in GPU and parallel computing scenarios, since the communication cost for each swap does not increase with the dimension of the problem (by swapping annealing parameters instead of states). We also assume that the number of PT scans will dominate the number of parallel cores available, i.e.  $n_{\text{scan}} \gg N$ . This is reasonable when addressing challenging sampling problems. Although there are numerous empirical studies on multi-core and distributed implementation of PT [Altekar et al., 2004, Mingas and Bouganis, 2012, Fang et al., 2014], we are not aware of previous theoretical work investigating such a computational model.

#### 3.2 Performance metrics for PT methods

The standard notion of computational efficiency of MCMC schemes is the effective sample size (ESS) per compute time. However, for PT methods, since the ESS per compute time depends on the details of the problem specific local exploration kernels  $\mathbf{K}^{\text{expl}}$ , alternatives have been developed

in the literature to assess the performance of  $\mathbf{K}_n^{\text{comm}}$  which are independent of  $\mathbf{K}^{\text{expl}}$  [Katzgraber et al., 2006, Lingenheil et al., 2009].

We are motivated by the Bayesian context where it is typically possible to obtain one independent sample from the reference distribution  $\pi_0$  (i.e. from the prior distribution) at each iteration. We say that an *annealed restart* has occurred on machine  $j$  when  $I_n^j$  goes from 0 to  $N$  (i.e.  $\beta$  goes from 0 to 1), which corresponds to a sample generated from  $\pi_0$  propagating to the target  $\pi$ . Informally an annealed restart can be thought of as a sampling equivalent to what is known in optimization as a random restart. We say a *round trip* has occurred on machine  $j$  when  $I_n^j$  goes from 0 to  $N$  and then goes back to 0 (i.e.  $\beta$  goes from 0 to 1 to 0).

Formally, we recursively define  $T_{\downarrow,0}^j = \inf\{n : (I_n^j, \varepsilon_n^j) = (0, -1)\}$  and for  $k \geq 1$ ,

$$T_{\uparrow,k}^j = \inf\{n > T_{\downarrow,k-1}^j : (I_n^j, \varepsilon_n^j) = (N, 1)\}, \quad (10)$$

$$T_{\downarrow,k}^j = \inf\{n > T_{\uparrow,k}^j : (I_n^j, \varepsilon_n^j) = (0, -1)\}. \quad (11)$$

The  $k$ -th annealed restart and round trip for machine  $j$  occurs at scan  $T_{\uparrow,k}^j$  and  $T_{\downarrow,k}^j$  respectively. Let  $\mathcal{T}_n$  and  $\mathcal{R}_n$  be the total number of annealed restarts and round trips respectively during the first  $n$  scans of the DEO algorithm.

We wish to optimize for the percentage of iterations that result in an annealed restart, i.e.  $\tau = \lim_{n \rightarrow \infty} \mathbb{E}[\mathcal{T}_n]/n$ , where we use abusively the same random variables for SEO and DEO but differentiate these schemes by using the probability measures  $\mathbb{P}_{\text{SEO}}$  and  $\mathbb{P}_{\text{DEO}}$  with associated expectation operators  $\mathbb{E}_{\text{SEO}}$  and  $\mathbb{E}_{\text{DEO}}$ . We use  $\mathbb{P}$  and  $\mathbb{E}$  for statements that hold for both algorithms. If  $\mathcal{T}_n^j$  and  $\mathcal{R}_n^j$  are the total number of annealed restarts and round trips during the first  $n$  iterations on machine  $j$  respectively, then we have  $\mathcal{R}_n^j \leq \mathcal{T}_n^j \leq \mathcal{R}_n^j + 1$ . Consequently,  $\mathcal{R}_n \leq \mathcal{T}_n \leq \mathcal{R}_n + N + 1$  and thus  $\tau = \lim_{n \rightarrow \infty} \mathbb{E}[\mathcal{R}_n]/n$ .

In the PT literature,  $\tau$  is commonly referred to as the *round trip rate* and has been used to compare the effectiveness of various PT algorithms [Katzgraber et al., 2006, Lingenheil et al., 2009]. Empirically we observed that round trips per unit cost strongly correlate with ESS per unit cost as seen in Figure 3, making the round trip rate a natural objective function to compare and tune parallel tempering algorithms.

Another performance metric commonly used in the PT literature is the *expected square jump distance* (ESJD) [Atchadé et al., 2011, Kone and Kofke, 2005]. While this criterion is useful within the context of reversible PT for selecting the optimal number of parallel chains, the ESJD is too coarse to compare reversible to non-reversible PT methods as, for any given annealing schedule, the ESJD is identical in both cases.

### 3.3 Model assumptions

The analysis of the round trip times is in general intractable because the index process is not Markovian. Indeed, simulating a transition depends on the swap indicators  $S_n^{(i,i+1)}$  (see Section 2.4), the distributions of which themselves depend on the state configuration  $\bar{\mathbf{X}}$ . To simplify the

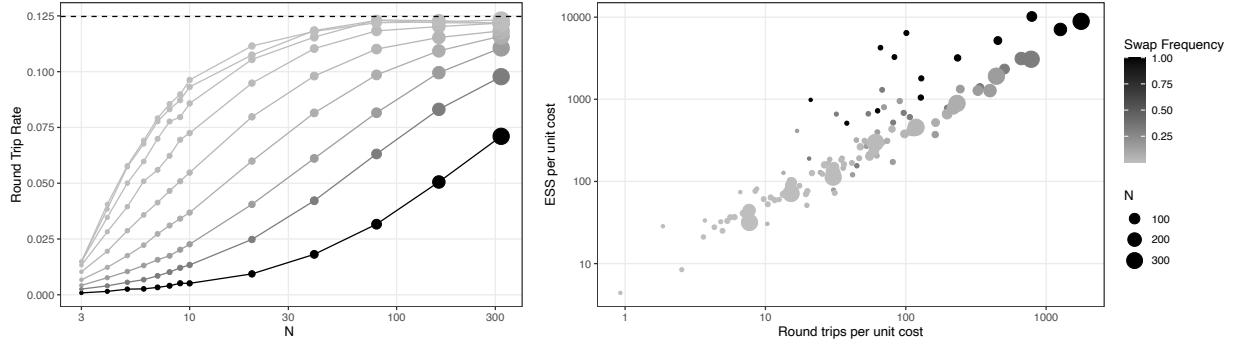


Figure 3: The round trip rate and ESS for a  $5 \times 5$  Ising Model with a magnetic moment 0.1 and  $n_{\text{expl}}$  1-bit flips between scans ranging from 1 to 400, and  $N$  ranging from 3 to 320. The schedule is tuned using Algorithm 4 with  $n_{\text{tune}} = 25000$  scans for tuning,  $n_{\text{sample}} = 25000$  scans for sampling. (Left) Round trip rate versus  $N$  for different swap attempt frequency ( $1/n_{\text{expl}}$ ). The dotted line is the optimal round trip rate  $\bar{\tau}$  predicted by Theorem 3. (Right) The ESS per unit cost versus round trip rate per unit cost for each run, with a correlation coefficient of 0.81.

analysis, we will make in the remainder of the paper the following simplifying assumptions:

- (A1) *Stationarity*:  $\bar{\mathbf{X}}_0 \sim \bar{\pi}$  and thus  $\bar{\mathbf{X}}_n \sim \bar{\pi}$  for all  $n$  as the kernel  $\mathbf{K}_n^{\text{PT}}$  is  $\bar{\pi}$ -invariant.
- (A2) *Efficient Local Exploration (ELE)*: For  $X \sim \pi^{(\beta)}$  and  $X'|X \sim K^{(\beta)}(X, \cdot)$ , the random variables  $V(X)$  and  $V(X')$  are independent.
- (A3) *Integrability*:  $V^3$  is integrable with respect to  $\pi_0$  and  $\pi$ .

It follows from Assumptions (A1)–(A2) and (6) that the behaviour of the communication scheme only depends on the distribution of the state  $\bar{\mathbf{X}}_n$  via the  $N+1$  univariate distributions of the chain-specific energies  $V^{(i)} = V(X^{(i)})$ ,  $i \in \{0, 1, 2, \dots, N\}$ . This allows us to build a theoretical analysis which makes no structural assumption on the state space  $\mathcal{X}$  or the target  $\pi$  as typically done in the literature: for example, [Atchadé et al., 2011] assume a product space  $\mathcal{X} = \mathcal{X}_0^d$  for large  $d$ , and [Predescu et al., 2004] assume  $\pi^{(\beta)}$  satisfies a constant heat capacity.

Admittedly the ELE assumption (A2) does not hold in practical applications. ELE can be approximated by increasing the number of local exploration kernels applied between consecutive swap ( $n_{\text{expl}}$ ). However one may worry that to achieve a good approximation in challenging problems,  $n_{\text{expl}}$  would have to be set to a value so large as to defy the practicality of our analysis. Surprisingly, we have observed empirically that this was not the case in the multimodal problems we considered. Figure 4 displays results in four models where a local exploration kernel alone induces good mixing of the energy chain  $V(X_n)$  (hence ELE can be approximated) yet the local exploration kernel alone is insufficient to achieve good mixing on the full state space,  $\bar{\mathbf{X}}_n$  (so that PT is justified and indeed yields efficient exploration of the configuration space). This gap is possible since  $V(X)$  is 1-dimensional and potentially unimodal even when  $X$  is not. This is the motivation for ELE since assuming the independence of  $V(X)$  and  $V(X')$  is weaker than assuming the independence of  $X$  and  $X'$  (as hypothesized e.g. in Section 5.1 of [Atchadé et al., 2011]). Obviously ELE is still

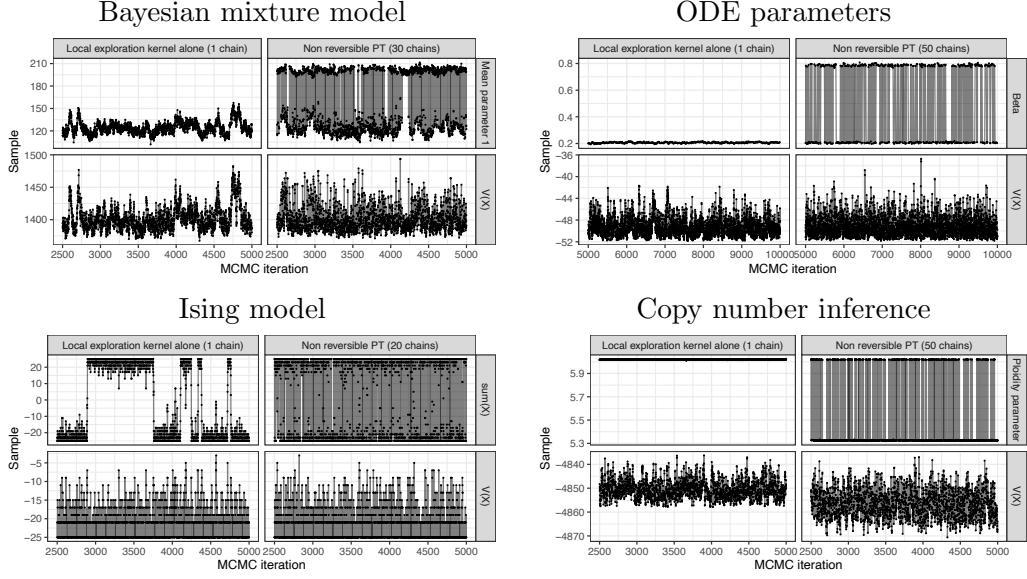


Figure 4: Four multimodal examples (described in Section 7.1 and Appendix I) where a local exploration kernel provides a reasonable approximation of the ELE assumption. For each inference problem, we show trace plots for MCMC based on single chain (i.e. the local exploration kernel alone; left facet), and for a non-reversible PT algorithm based on the same local exploration kernel (right facet). The top facets each show a component of  $X$ , and the bottom facets,  $V(X)$ .

expected to be a somewhat crude simplifying assumption in very complex problems; e.g. for the highly challenging high-dimensional copy number inference problem illustrated in Figure 4.

In Section 7.2, we describe additional empirical results supporting that ELE is a useful model for the purpose of designing and analyzing PT algorithms. In the severe ELE violation regime, we show that the key quantities used in our analysis are either well approximated, or approached as  $N$  increases.

Assumptions (A1)-(A2) allow us to express the swap indicators as *independent* Bernoulli random variables  $S_n^{(i,i+1)} \sim \text{Bern}(s^{(i,i+1)})$  where  $s^{(i,i+1)}$  is given by the expectation of Equation (6),

$$s^{(i,i+1)} = \mathbb{E} \left[ \alpha^{(i,i+1)}(\bar{\mathbf{X}}) \right] = \mathbb{E} \left[ \exp \left( \min \left\{ 0, (\beta_{i+1} - \beta_i) (V^{(i+1)} - V^{(i)}) \right\} \right) \right], \quad (12)$$

the expectation being over two independent random variables  $V^{(i)}, V^{(i+1)}$ , satisfying  $V^{(i)} \stackrel{d}{=} V(X^{(\beta_i)})$  for  $X^{(\beta_i)} \sim \pi^{(\beta_i)}$ .

### 3.4 Reversibility and non-reversibility of the index process

Under assumptions (A1)-(A2), each index process  $(I_n^j, \varepsilon_n^j)$  is Markovian for  $j = 0, \dots, N$  with transition kernel  $K^{\text{SEO}}$  and  $K^{\text{DEO}}$  for reversible and non-reversible PT respectively. See Appendix A for an explicit representation of these kernels. We drop the superscript  $j$  when the particular machine is not relevant.

The kernel  $K^{\text{SEO}}$  defines a reversible Markov chain on  $\{0, \dots, N\} \times \{-1, 1\}$  with uniform stationary distribution while  $K^{\text{DEO}}$  satisfies the skew-detailed balance condition with respect to the same distribution,

$$K^{\text{DEO}}((i, \varepsilon), (i', \varepsilon')) = K^{\text{DEO}}((i', -\varepsilon'), (i, -\varepsilon)), \quad (13)$$

and is thus non-reversible. It falls within the generalized Metropolis–Hastings framework, see, e.g., [Lelièvre et al., 2010].

Reversibility necessitates that the Markov chain must be allowed to backtrack its movements. This leads to inefficient exploration of the state space. As a consequence, non-reversibility is typically a favourable property for MCMC chains. A common recipe to design non-reversible sampling algorithms consists of expanding the state space to include a “lifting” parameter that allows for a more systematic exploration of the state space [Chen et al., 1999, Diaconis et al., 2000, Turitsyn et al., 2011, Vucelja, 2016].

The index process  $(I_n, \varepsilon_n)$  for non-reversible PT can be interpreted as a “lifted” version of the index process for reversible PT with lifting parameter  $\varepsilon_n$ . Under DEO communication,  $I_n$  travels in the direction  $\varepsilon_n$  and only reverses direction when  $I_n$  reaches a boundary or when a swap rejection occurs. This “lifting” construction helps explain the qualitatively different behaviour between reversible and non-reversible PT and will be further explored when identifying the scaling limit of  $(I_n, \varepsilon_n)$  in Section 6. The lifted PT of [Wu, 2017] exploits a similar construction but only one of the  $N + 1$  index processes is lifted instead of all of them for DEO. A lifted version of simulated tempering has also been proposed by [Sakai and Hukushima, 2016].

### 3.5 Non-asymptotic domination of non-reversible PT

Assumptions (A1)-(A2) ensure that for each  $j = 0, \dots, N$ ,  $\mathcal{R}_n^j$  is a delayed renewal process with round trip times  $T_k^j = T_{\downarrow, k}^j - T_{\downarrow, k-1}^j$  for  $k \geq 1$  and  $j = 0, \dots, N$ . In particular,  $T_k^j$  are independent and identically distributed and, for convenience, we introduce the random variable  $T \stackrel{d}{=} T_k^j$ . By the key renewal theorem, we have

$$\tau = \sum_{j=0}^N \lim_{n \rightarrow \infty} \frac{\mathbb{E}[\mathcal{R}_n^j]}{n} = \frac{N+1}{\mathbb{E}[T]}. \quad (14)$$

An analytical expression for  $\mathbb{E}[T]$  for reversible PT was first derived by [Nadler and Hansmann, 2007]. We extend this result to non-reversible PT in Theorem 1.

**Theorem 1.** *For any annealing schedule  $\mathcal{P}_N = \{\beta_0, \dots, \beta_N\}$ ,*

$$\mathbb{E}_{\text{SEO}}[T] = 2(N+1)N + 2(N+1)E(\mathcal{P}_N), \quad (15)$$

$$\mathbb{E}_{\text{DEO}}[T] = 2(N+1) + 2(N+1)E(\mathcal{P}_N), \quad (16)$$

where  $E(\mathcal{P}_N) = \sum_{i=1}^N r^{(i-1,i)} / s^{(i-1,i)}$ , and  $r^{(i-1,i)} = 1 - s^{(i-1,i)}$  is the probability of rejecting a swap between chains  $i$  and  $i+1$ .

The proof can be found in Appendix B.

Intuitively, Theorem 1 implies  $\mathbb{E}[T]$  can be decomposed as the independent influence of communication scheme  $\mathbf{K}_n^{\text{comm}}$  and schedule  $\mathcal{P}_N$  respectively. When all proposed swaps are accepted (i.e.  $\pi = \pi_0$ ), the index process for reversible PT reduces to a simple random walk on  $\{0, \dots, N\}$ , whereas for non-reversible PT, the index processes takes a direct path from 0 to  $N$  and back. Therefore, the first term in (15) and (16) represents the expected time for a round trip to occur in this idealized, rejection-free setting. The second term of (15) and (16) are identical and represent the additional time required to account for rejected swaps under schedule  $\mathcal{P}_N$ . Motivated by Theorem 1, we will refer to  $E(\mathcal{P}_N)$  as the *schedule inefficiency*.

By applying Theorem 1 to Equation (14), we get a non-asymptotic formula for the round trip rate in terms of  $E(\mathcal{P}_N)$ .

**Corollary 1.** *For any annealing schedule  $\mathcal{P}_N$  we have*

$$\tau_{\text{SEO}}(\mathcal{P}_N) = \frac{N+1}{\mathbb{E}_{\text{SEO}}[T]} = \frac{1}{2N + 2E(\mathcal{P}_N)}, \quad (17)$$

$$\tau_{\text{DEO}}(\mathcal{P}_N) = \frac{N+1}{\mathbb{E}_{\text{DEO}}[T]} = \frac{1}{2 + 2E(\mathcal{P}_N)}. \quad (18)$$

Consequently,  $\tau_{\text{DEO}}(\mathcal{P}_N) > \tau_{\text{SEO}}(\mathcal{P}_N)$  for  $N > 1$ .

Corollary 1 implies that non-reversible PT dominates reversible PT for any  $N > 1$  and any annealing schedule  $\mathcal{P}_N$ .

## 4 Asymptotic analysis of PT algorithms

### 4.1 The communication barrier

We begin by analyzing the behaviour of the PT swaps as  $\|\mathcal{P}_N\| = \max_i |\beta_i - \beta_{i-1}|$  goes to zero. We define the swap and rejection functions  $s, r : [0, 1]^2 \rightarrow [0, 1]$  respectively as,

$$s(\beta, \beta') = \mathbb{E} \left[ \exp \left( \min \{0, (\beta' - \beta)(V^{(\beta')} - V^{(\beta)})\} \right) \right], \quad (19)$$

$$r(\beta, \beta') = 1 - s(\beta, \beta'), \quad (20)$$

where  $V^{(\beta)} \stackrel{d}{=} V(X^{(\beta)})$  for  $X^{(\beta)} \sim \pi^{(\beta)}$  and  $V^{(\beta)}, V^{(\beta')}$  are independent. The quantities  $s(\beta, \beta')$  and  $r(\beta, \beta')$  are symmetric in their arguments and represent the probability of swap and rejection occurring between  $\beta$  and  $\beta'$  respectively under the ELE assumption (A2). Note that  $s^{(i-1,i)} = s(\beta_{i-1}, \beta_i)$ .

To take the limit as  $\|\mathcal{P}_N\| \rightarrow 0$ , it will be useful to understand the behaviour of  $r(\beta, \beta')$  when  $\beta \approx \beta'$ . The key quantity that drives this asymptotic regime is given by a function  $\lambda : [0, 1] \rightarrow [0, \infty)$

defined as the instantaneous rate of rejection of a proposed swap at annealing parameter  $\beta$ ,

$$\lambda(\beta) = \lim_{\delta \rightarrow 0} \frac{r(\beta, \beta + \delta)}{|\delta|}. \quad (21)$$

See Figure 7 (center) for examples of estimated  $\lambda$  from various models. We define the integral of  $\lambda$  by  $\Lambda(\beta) = \int_0^\beta \lambda(\beta') d\beta'$  and denote  $\Lambda = \Lambda(1)$ . Extending Proposition 1 in [Predescu et al., 2004] provides the following result.

**Theorem 2.** *The instantaneous rejection rate  $\lambda$  is twice continuously differentiable and is equal to*

$$\lambda(\beta) = \frac{1}{2} \mathbb{E} \left[ |V_1^{(\beta)} - V_2^{(\beta)}| \right], \quad (22)$$

where  $V_1^{(\beta)}, V_2^{(\beta)}$  are independent random variables with law identical to the law of  $V^{(\beta)}$ . Moreover, we have

$$r(\beta, \beta') = |\Lambda(\beta') - \Lambda(\beta)| + O(|\beta' - \beta|^3). \quad (23)$$

See Appendix C for a proof and general smoothness properties of  $\lambda$ . In particular, the existence and smoothness of  $\lambda$  is guaranteed by Assumption (A3).

Theorem 2 shows that  $\lambda$  encodes up to second order the behaviour of  $r$  as the annealing parameter difference between the chains goes to 0. When  $\lambda(\beta)$  is high, swaps are much more likely to be rejected, implying  $\lambda(\beta)$  measures the difficulty of local communication for a chain with annealing parameter  $\beta$ .

Notice that  $\Lambda \geq 0$  with equality if and only if  $\lambda(\beta) = 0$  for all  $\beta \in [0, 1]$ . It can be easily verified from (22) that  $\lambda = 0$  if and only if  $V^{(\beta)}$  is constant  $\pi^{(\beta)}$ -a.s. for all  $\beta \in [0, 1]$  which happens precisely when  $\pi_0 = \pi$ . So  $\Lambda$  defines a natural symmetric divergence measuring the difficulty of communication between  $\pi_0$  and  $\pi$ . In particular, for any schedule  $\mathcal{P}_N$ , the sum of the rejection rates is approximately constant and equal to  $\Lambda$  as formalized in Corollary 2.

**Corollary 2.** *For any schedule  $\mathcal{P}_N$ , we have*

$$\sum_{i=1}^N r(\beta_{i-1}, \beta_i) = \Lambda + O(N \|\mathcal{P}_N\|^3). \quad (24)$$

Motivated by Theorem 2 and Corollary 2 we will henceforth refer to  $\lambda$ ,  $\Lambda$  and  $\Lambda(\cdot)$  as the *local*, *global* and *cumulative communication barriers* respectively.

## 4.2 Asymptotic analysis of round trip rate

The communication barrier allows us to study the asymptotic performance of PT when the number of parallel chains is large. When  $N$  increases, the round trip rate for reversible PT decays to 0 while it converges to a positive constant  $\bar{\tau}$  for non-reversible PT as seen in left plot in Figure 3. We formally show this in Theorem 3.

**Theorem 3.** As  $\|\mathcal{P}_N\| \rightarrow 0$  we have:

(a) The communication inefficiency satisfies

$$\Lambda \leq E(\mathcal{P}_N) = \Lambda + O(\|\mathcal{P}_N\|). \quad (25)$$

(b) The round trip rate for reversible PT,  $\tau_{SEO}$ , goes to zero:

$$\tau_{SEO}(\mathcal{P}_N) \sim \frac{1}{2N + 2\Lambda} \longrightarrow 0. \quad (26)$$

(c) The round trip rate for non-reversible PT,  $\tau_{DEO}$  satisfies

$$\tau_{DEO}(\mathcal{P}_N) \longrightarrow \bar{\tau} = \frac{1}{2 + 2\Lambda} > 0, \quad (27)$$

where the converges occurs with rate  $O(\|\mathcal{P}_N\|)$ .

See Appendix D for a proof.

In general,  $\Lambda$  is large when  $\pi_0$  deviates significantly from  $\pi$ . Since  $\Lambda$  is problem specific, this identifies a limitation of PT present even in its non-reversible flavour, namely that adding more cores to the task will never be harmful, but does have a diminishing return. The bound  $(2 + 2\Lambda)^{-1}$  could indeed be very small for complex problems. Moreover, it is independent of the choice of annealing schedule, hence it cannot be improved by the schedule optimization procedure described in Section 5.

When  $N$  is fixed, we will see in Section 5.1 that  $\tau_{DEO}, \tau_{SEO}$  are both maximized when  $r^{(i-1,i)} = r^*$  for all  $i$ . In this case,  $E(\mathcal{P}_N) = N(1 - r^*)^{-1} r^*$  and one has

$$\frac{N}{1 + Nr^*} \leq \frac{\tau_{DEO}(\mathcal{P}_N)}{\tau_{SEO}(\mathcal{P}_N)} = \frac{N}{1 + (N - 1)r^*} \leq \frac{1}{r^*}. \quad (28)$$

This implies that for an optimally chosen schedule, non-reversible PT improves the round trip rate over reversible PT by at most a factor of  $1/r^*$ . Corollary 2 implies  $Nr^* \approx \Lambda$ , allowing us to quantify the looseness of (28) directly in terms of  $\Lambda$ .

$$\frac{1}{r^*} - \frac{N}{1 + Nr^*} = \frac{1}{r^*} \left( \frac{1}{1 + Nr^*} \right) \approx \frac{1}{r^*} \left( \frac{1}{1 + \Lambda} \right). \quad (29)$$

We can also investigate the performance of PT for multi-modal or high-dimensional targets by studying the behaviour of  $\Lambda$ . Assuming the target distribution factorizes [Woodard et al., 2009, Atchadé et al., 2011, Roberts and Rosenthal, 2014], we can show that  $\Lambda$  is robust to the number of modes and grows at rate  $O(\sqrt{d})$  as the dimension  $d$  of the target increases. See Appendices E and F for details.

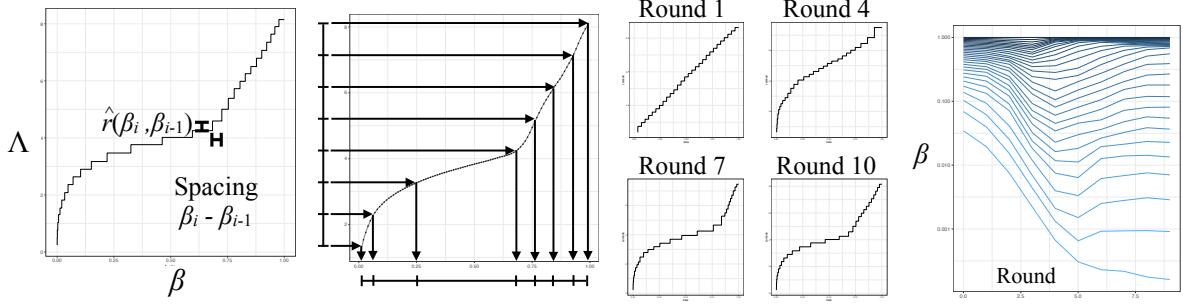


Figure 5: Proposed annealing schedule optimization method: example on the Bayesian mixture model of Section 7. (1) The cumulative barrier  $\Lambda(\cdot)$  is estimated using (32) at each point  $\beta_k$  of an initial partition. (2) The cumulative barrier is interpolated using monotonic cubic interpolation, and a new schedule (shown beside the abscissa axis) is obtained by computing the inverse under  $\Lambda(\cdot)$  of a regular grid (shown beside the ordinate). (3) The process 1–2 is repeated iteratively. (4) The sequence of annealing schedules obtained as a function of the round (colours represent different grid points in the schedule).

## 5 Tuning non-reversible PT

### 5.1 Optimal annealing schedule

We first discuss how to optimize the annealing schedule  $\mathcal{P}_N$  to maximize  $\tau_{\text{DEO}}$  when  $N$  is fixed, which is equivalent to minimizing the schedule inefficiency,  $E(\mathcal{P}_N) = \sum_{i=1}^N r_i / (1 - r_i)$  where  $r_i = r(\beta_{i-1}, \beta_i)$ . To get a tractable approximate characterization of the feasible region of  $r_1, r_2, \dots, r_N$ , we use Corollary 2, which implies  $\sum_{i=1}^N r_i \approx \Lambda$  for all schedules  $\mathcal{P}_N$ . Therefore assuming  $\|\mathcal{P}_N\|$  is small enough to ignore the error term, finding the optimal schedule  $\mathcal{P}_N^*$  is approximately equivalent to minimizing  $\sum_{i=1}^N r_i / (1 - r_i)$ , subject to the constraint  $\sum_{i=1}^N r_i = \Lambda$  and  $r_i > 0$ . Using Lagrange multipliers this leads to a solution where the rejection probabilities  $r_i^*$  are constant in  $i$ , with a shared value denoted by  $r^*$ .

Consequently the optimal schedule  $\mathcal{P}_N^* = \{\beta_0^*, \dots, \beta_N^*\}$  satisfies  $r(\beta_{i-1}^*, \beta_i^*) = r^*$  for all  $i$ . Theorem 2 and Corollary 2 imply that  $r^*$  must satisfy,  $r^* \approx \Lambda(\beta_i^*) - \Lambda(\beta_{i-1}^*)$  for all  $i$  and  $r^* \approx \Lambda/N$  with  $O(N^{-3})$  error. By equating these two estimates for  $r^*$  and summing from  $i = 0, \dots, k$  we get

$$\Lambda(\beta_k^*) \approx \frac{k}{N} \Lambda, \quad (30)$$

with an error of  $O(N^{-2})$ . If we ignore error terms, (30) implies that  $\beta_k^* \approx G(k/N)$  where  $G = F^{-1}$  and  $F(\beta) = \Lambda(\beta)/\Lambda$  (see Figure 5(2)). See Appendix I.4.1 for an example where an analytic result is available, however in general  $\Lambda(\cdot)$  is not known but instead estimated from the MCMC output.

The ‘‘equi-acceptance’’ result in (30) is consistent with other theoretical frameworks and notions of efficiency [Atchadé et al., 2011, Lingenheil et al., 2009, Kofke, 2002, Predescu et al., 2004]. However implementing this equi-acceptance recommendation in practice is non-trivial. Previous work relied on Robbins-Monro schemes [Atchadé et al., 2011, Miasojedow et al., 2013], which introduce sensitive tuning parameters. In contrast, provided  $\Lambda(\cdot)$  can be estimated reliably (which we establish with Algorithm 3 in the next section), (30) provides the basis for a straightforward and effective schedule

optimization scheme, described in Algorithm 2.

---

**Algorithm 2** UpdateSchedule(communication barrier  $\Lambda(\cdot)$ , schedule size  $N$ )

---

- 1:  $\Lambda \leftarrow \Lambda(1)$
  - 2: **for**  $k$  in  $0, 1, 2, \dots, N$  **do**
  - 3:     Find  $\beta_k^*$  such that  $\Lambda(\beta_k^*) = \frac{k}{N} \Lambda$  using e.g. bisection.
  - 4: **return**  $\mathcal{P}_N^* = (\beta_0^*, \beta_1^*, \beta_2^*, \dots, \beta_N^*)$
- 

## 5.2 Estimation of the communication barrier

Assume we have access to a collection of samples  $(\bar{\mathbf{X}}_1, \bar{\mathbf{X}}_2, \dots, \bar{\mathbf{X}}_n)$  from a non-reversible PT scheme based on an arbitrary annealing schedule  $\mathcal{P}_N$ . These samples may come from a short pilot run, or, as described in the next section, from the previous round of an iterative scheme. For a given schedule  $\mathcal{P}_N$ , when the central limit theorem for Markov chains holds, the Monte Carlo estimates for the rejection rates satisfy

$$\hat{r}^{(i-1,i)} = \frac{1}{n} \sum_{k=1}^n \alpha^{(i-1,i)}(\bar{\mathbf{X}}_k) = r^{(i-1,i)} + O_p(n^{-1/2}). \quad (31)$$

Next, using Theorem 2 we obtain  $\sum_{i=1}^k r^{(i-1,i)} = \Lambda(\beta_k) + O(N\|\mathcal{P}_N\|^3)$ . This motivates the following approximation for  $\Lambda(\beta_k)$ ,

$$\hat{\Lambda}(\beta_k) = \sum_{i=1}^k \hat{r}^{(i-1,i)}, \quad (32)$$

which has an error of order  $O_p(\sqrt{N/n} + N\|\mathcal{P}_N\|^3)$  (see Figure 5 (left)).

Given  $\hat{\Lambda}(\beta_0), \dots, \hat{\Lambda}(\beta_N)$ , we estimate the function  $\Lambda(\beta)$  via interpolation, with the constraint that the interpolated function should be monotone increasing since  $\lambda(\beta) \geq 0$ . Specifically, we use the Fritsch-Carlson monotone cubic spline method [Fritsch and Carlson, 1980] and denote the monotone interpolation by  $\hat{\Lambda}(\beta)$ . While we only use  $\hat{\Lambda}(\beta)$  in our schedule optimization procedure, it is still useful to estimate  $\lambda(\beta)$  for visualization purpose. We use the derivative of our interpolation,  $\hat{\lambda}(\beta) = \hat{\Lambda}'(\beta)$ , which is a piecewise quadratic function.

The ideas described in this section so far are summarized in Algorithm 3, which given rejection statistics collected for a fixed annealing schedule provides an estimate of the communication barrier.

---

**Algorithm 3** CommunicationBarrier(rejection rate  $\{r^{(i-1,i)}\}$ , schedule  $\mathcal{P}_N$ )

---

- 1: For each  $\beta_i \in \mathcal{P}_N$ , compute  $\hat{\Lambda}(\beta_i)$  ▷ Equation (32)
  - 2:  $S \leftarrow \{(\beta_0, \hat{\Lambda}(\beta_0)), (\beta_1, \hat{\Lambda}(\beta_1)), \dots, (\beta_N, \hat{\Lambda}(\beta_N))\}$
  - 3: Compute a monotone increasing interpolation  $\hat{\Lambda}(\cdot)$  of the points  $S$  ▷ e.g. using [Fritsch and Carlson, 1980]
  - 4: **return**  $\hat{\Lambda}(\cdot)$
- 

As a byproduct of Algorithm 3 we also obtain a consistent estimator  $\hat{\tau} = (2 + 2\hat{\Lambda})^{-1}$  for the

optimal round trip rate  $\bar{\tau}$ , where  $\hat{\Lambda} = \hat{\Lambda}(1)$ . We show in Figure 8 an example where (A2) is severely violated, yet  $\lambda$  is still reliably estimated. This allows us to compare the empirically observed round trip rate against  $\hat{\tau}$ , and hence estimate how far an implementation deviates from optimal performance.

### 5.3 Tuning $N$

Theorem 3 shows that non-reversible PT does not deteriorate in performance as we increase  $N$  unlike reversible PT, however the gains in round trip rate eventually become marginal. When  $N$  is very large we expect to accumulate more round trips by running  $k > 1$  parallel copies of PT. As we shall see, the large  $N$  asymptotic is still required however in order to determine the optimal number  $k^*$  of PT copies.

Suppose there are  $k$  copies of PT running in parallel consisting of  $N + 1$  chains with optimal annealing schedule  $\mathcal{P}_N$ . If  $\bar{N}$  is the total number of cores available then  $k$  and  $N$  satisfy the constraint  $k(N + 1) = \bar{N}$ . By Corollary 1 the total round trip rate across all  $k$  copies of PT is

$$\tau = k \tau_{\text{DEO}}(\mathcal{P}_N) = \frac{\bar{N}}{2(N + 1)(1 + E(\mathcal{P}_N))}. \quad (33)$$

From Section 5.1, the optimal schedule  $\mathcal{P}_N^*$  has a corresponding swap rejection rate  $r^* = \Lambda/N + O(N^{-3})$ . Substituting this into (33) we get  $\tau = \tau_\Lambda(N) + O(N^{-1})$ , where

$$\tau_\Lambda(N) = \frac{\bar{N}(1 - \Lambda/N)}{2(N + 1)(1 - \Lambda/N + \Lambda)}. \quad (34)$$

Ignoring error terms,  $\tau_\Lambda(\cdot)$  is optimized when we run  $k^* = \bar{N}/(1 + N^*)$  copies of PT with  $N^* + 1 = 2\Lambda + 1$  chains to achieve an optimum round trip rate  $\tau^* = k^*/(2 + 4\Lambda)$ .

Note that when  $\tau$  is optimized, we have  $r^* \approx 1/2$ , which differs from the 0.77 optimal rejection rate from the reversible PT literature [Atchadé et al., 2011, Kofke, 2002, Predescu et al., 2004]. In Section 7.2, we show empirically that when the ELE assumption (A2) is severely violated, the recommendation in the present section is turned into a bound,  $r^* < 1/2$ , as increasing  $N$  appears to alleviate ELE violations.

### 5.4 Iterative schedule optimization

Suppose we have  $\bar{N}$  cores available and a computational budget of  $n_{\text{scan}}$  scans of PT, one scan consisting in one application of  $\mathbf{K}^{\text{PT}}$ . The first  $n_{\text{tune}}$  scans are used to find an accurate estimate of the communication barrier  $\hat{\Lambda}(\cdot)$  and the remaining  $n_{\text{sample}} = n_{\text{scan}} - n_{\text{tune}}$  scans to sample from the target. Algorithm 4 approximates  $\hat{\Lambda}$  with error  $O_p(\sqrt{\bar{N}/n_{\text{tune}}} + \bar{N}^{-2})$  by iteratively using Algorithms 2 and 3 for `maxRound`= $\log_2 n_{\text{tune}}$  rounds until the tuning budget is depleted. Equipped with  $\hat{\Lambda}(\cdot)$ , we can use the remaining  $n_{\text{sample}}$  scans to run  $k^* = \bar{N}/(N^* + 1)$  copies of non-reversible PT with  $N^* = 2\hat{\Lambda}(1)$  chains with optimal schedule  $\mathcal{P}_{N^*}^*$  from Algorithm 2. In cases where it is

suspected that (A2) may be severely violated (as evidenced for example by a large gap between  $\bar{\tau}$  and the estimated  $\tau$  as described in Section 5.2), it may be advantageous to also attempt  $N^* = \bar{N}$  in line 9 of Algorithm 4. We show empirically in Figure 7 (top) that in a range of synthetic and real world problems our scheme converges using a small number of iterations, in the order of `maxRound` = 10. In our experiments we used  $n_{\text{tune}} = n_{\text{scan}}/2$  and discarded the samples from the first  $n_{\text{tune}}$  scans as burn-in.

---

**Algorithm 4** NRPT(number of cores  $\bar{N}$ , tuning budget  $n_{\text{tune}}$ , sampling budget  $n_{\text{sample}}$ )

---

```

1:  $\mathcal{P}_{\bar{N}} \leftarrow$  initial annealing schedule of size  $\bar{N} + 1$  (e.g. uniform)
2: maxRound  $\leftarrow \log_2(n_{\text{tune}})$ 
3:  $n \leftarrow 1$  ▷ Scans per round
4: for round in 1, 2, ..., maxRound do
5:    $\{r^{(i-1,i)}\} \leftarrow \text{DEO}(n, \mathcal{P}_{\bar{N}})$  ▷ Algorithm 1
6:    $\hat{\Lambda}(\cdot) \leftarrow \text{CommunicationBarrier}(\{r^{(i-1,i)}\}, \mathcal{P}_{\bar{N}})$  ▷ Algorithm 3
7:    $\mathcal{P}_{\bar{N}} \leftarrow \text{UpdateSchedule}(\hat{\Lambda}(\cdot), \bar{N})$  ▷ Algorithm 2
8:    $n \leftarrow 2n$  ▷ Rounds use an exponentially increasing number of scans
9:    $N^* \leftarrow 2\hat{\Lambda}(1)$  ▷ See Section 5.3
10:   $\mathcal{P}^* \leftarrow \text{UpdateSchedule}(\hat{\Lambda}(\cdot), N^*)$  ▷ Optimal schedule
11:   $k^* \leftarrow \bar{N}/(N^* + 1)$  ▷ Number of copies of PT
12:  for 1, ...,  $k^*$  do
13:     $(\bar{x}_1, \dots, \bar{x}_{n_{\text{sample}}}) \leftarrow \text{DEO}(n_{\text{sample}}, \mathcal{P}^*)$ 
14:  return  $(\bar{x}_1, \dots, \bar{x}_{n_{\text{sample}}})$ 

```

---

## 5.5 Normalizing constant computation

Algorithm 4 can be easily modified to obtain efficient estimators for quantities of the form  $\int_0^1 \mathbb{E}[f(V^{(\beta)}, \beta)] d\beta$  with an error of  $O_p(\sqrt{\bar{N}/n_{\text{tune}}} + \bar{N}^{-2})$  for well-behaved functions  $f$ . Examples of such quantities include log-normalizing constants [Kirkwood, 1935, Gelman and Meng, 1998, Xie et al., 2011], KL-divergence [Dabak and Johnson, 2002], and Fisher-Rao distance [Amari, 2016] between  $\pi$  and  $\pi_0$ .

For example, by taking a Riemann sum in the thermodynamic integration identity [Ogata, 1989], the log-normalizing constant  $\log \mathcal{Z}(1)$  of  $\pi$  can be approximated using

$$\log \frac{\mathcal{Z}(1)}{\mathcal{Z}(0)} = - \int_0^1 \mu(\beta) d\beta = - \sum_{i=1}^N \mu(\beta_i)(\beta_i - \beta_{i-1}) + O(N^{-2}), \quad (35)$$

where  $\mu(\beta) = \mathbb{E}[V^{(\beta)}]$  and  $\mathcal{Z}(0)$  the normalizing constant of  $\pi_0$ . By running DEO with schedule  $\mathcal{P}_N$  for  $n$  scans we substitute a Monte Carlo estimate  $\hat{\mu}(\beta_k)$  for  $\mu(\beta_k)$  into (35) to get an estimator  $\log(\hat{\mathcal{Z}}(1)/\hat{\mathcal{Z}}(0))$  with error  $O_p(\sqrt{N/n} + N^{-2})$ . See also [Gelman and Meng, 1998, Xie et al., 2011] for closely related normalization constant estimators. Figure 7 (bottom) in Section 7.1 shows how the estimate for  $\log \mathcal{Z}(1)$  evolves with the number of rounds in Algorithm 4 for 16 different models.

## 6 Scaling limits of index process

The non-reversible communication scheme introduced in [Okabe et al., 2001] was presumably devised on algorithmic grounds (it performs the maximum number of swap attempts in parallel) since no theoretical justification was provided and the non-reversibility of the scheme was not mentioned. The arguments given in [Lingenheil et al., 2009] to explain the superiority of non-reversible communication over various PT algorithms rely on a misleading assumption, namely a diffusive scaling limit for the index process. Figure 1 (see also Figure 15 in the Supplementary Material) suggests that the index process behaves qualitatively differently as  $N$  increases for reversible and non-reversible PT. The goal of this section is to investigate these differences by identifying some scaling limits. Such limits exist under assumptions (A1)-(A3) specified in Section 3.3 and Section 4.1.

Suppose  $G : [0, 1] \rightarrow [0, 1]$  is an increasing differentiable function satisfying  $G(0) = 0$  and  $G(1) = 1$ . We say that  $G$  is a *schedule generator* for  $\mathcal{P}_N = \{\beta_0, \dots, \beta_N\}$  if  $\beta_i = G(i/N)$ . We will now assume without loss of generality that the sequence of schedules  $\mathcal{P}_N$  are generated by some common  $G$ . In particular the mean value theorem implies  $\|\mathcal{P}_N\| = O(N^{-1})$  as  $N \rightarrow \infty$ . This is not a strict requirement as most annealing schedules commonly used fall within this framework: the uniform schedule  $\mathcal{P}_N^{\text{uniform}} = \{0, 1/N, \dots, 1\}$  is generated by  $G(w) = w$ , the optimal schedule  $\mathcal{P}_N^* = \{\beta_0^*, \dots, \beta_N^*\}$  derived in Section 5.1 is approximately generated by  $G(w) = F^{-1}(w)$ , where  $F(\beta) = \Lambda(\beta)/\Lambda$ . If  $\pi_0(x) \propto \pi(x)^\gamma$  for some  $\gamma \in (0, 1)$ , and  $L(x) \propto \pi(x)^{1-\gamma}$  then  $G(w) = \frac{\gamma^{1-w}-\gamma}{1-\gamma}$  corresponds to the geometric schedule commonly used by practitioners. In contrast, we show in Figure 20 examples of schedule generators  $G$  corresponding to estimated optimal  $\mathcal{P}_N^*$  from various models. Figure 20 shows that in real world inference scenarios, the optimal  $G$  often qualitatively differs from the simple geometric schedule.

Suppose  $(I_n, \varepsilon_n)$  is the index process for an annealing schedule  $\mathcal{P}_N$  generated by  $G$ . To establish a scaling limit for  $(I_n, \varepsilon_n)$ , it will be convenient to work in a continuous time setting. To do this, we suppose the times that PT iterations occur are distributed according to a Poisson process  $\{M(\cdot)\}$  with mean  $\mu_N$ . The number  $M(t)$  of PT iterations that occur by time  $t \geq 0$  thus satisfies  $M(t) \sim \text{Poisson}(\mu_N t)$ . We define the *scaled index process* by  $Z^N(t) = (W^N(t), \varepsilon^N(t))$  where  $W^N(t) = I_{M(t)}/N$  and  $\varepsilon^N(t) = \varepsilon_{M(t)}$ .

Define the piecewise-deterministic Markov process (PDMP) [Davis, 1993]  $Z(t) = (W(t), \varepsilon(t))$  on  $[0, 1] \times \{-1, 1\}$  as follows:  $W(t)$  moves in  $[0, 1]$  with velocity  $\varepsilon(t)$  and the sign of  $\varepsilon(t)$  is reversed at an inhomogeneous rate  $\lambda(G(W(t))G'(W(t)))$  or when  $W(t)$  hits a boundary; see [Bierkens et al., 2018] for a discussion of PDMP on restricted domains. The process  $Z$  corresponds to an inhomogeneous persistent random walk with reflective boundary conditions; see [Masoliver et al., 1992] for a description of the Fokker-Plank equation and first passage times. As the optimal schedule is generated by  $G = F^{-1}$  for  $F(\beta) = \Lambda(\beta)/\Lambda$ , we have  $\lambda(G(w))G'(w) = \Lambda$  for all  $w \in [0, 1]$ , so  $\varepsilon(t)$  changes sign at a constant rate  $\Lambda$ . See Figure 13 for sample trajectories of  $Z$  for different values of  $\Lambda$ . The main result of this section is the following theorem.

**Theorem 4.** Suppose  $\mathcal{P}_N$  is family of schedules generated by  $G$ , then

- (a) For reversible PT if  $\mu_N = N^2$  and if  $W^N(0)$  converges weakly to  $W(0)$  then  $W^N$  converges weakly to a diffusion  $W$  with initial condition  $W(0)$ , where  $W$  is a Brownian motion on  $[0, 1]$  with reflective boundary conditions. The process  $W$  admits  $\text{Unif}([0, 1])$  as stationary distribution.
- (b) For non-reversible PT if  $\mu_N = N$  and if  $Z^N(0)$  converges weakly to  $Z(0)$ , then  $Z^N$  converges weakly to the PDMP  $Z$  with initial condition  $Z(0)$ . The process  $Z$  admits  $\text{Unif}([0, 1] \times \{-1, 1\})$  as stationary distribution.

See Appendix G for a detailed construction of the scaled index process via their infinitesimal generators, and the proof of Theorem 4.

Theorem 4 implies for reversible PT that, if we speed time by a factor of  $N^2$ , then the index process scales to a diffusion  $W$  independent of the choice of  $\pi_0, \pi$  and of the schedule. This is in contrast to non-reversible PT where, if we speed time by a factor of  $N$ , the index process converges to a PDMP  $Z$  depending on  $\pi_0, \pi$  through  $\lambda$  and the schedule through  $G$ .

## 7 Examples

### 7.1 Empirical behaviour of the schedule optimization method

We applied the proposed NRPT algorithm to 16 models from the statistics and physics literature to demonstrate its versatility. The models include 9 Bayesian models ranging from simple standard models such as generalized linear models and Bayesian mixtures, to complex ones such as cancer copy-number calling, ODE parameter estimation, spike-and-slab classification and two types of phylogenetic models. This is complemented by three models from statistical mechanics and four artificial models. The datasets considered include eight real datasets spanning diverse data-types and size, e.g. state-of-the-art measurements such as whole-genome single-cell sequencing data (494 individual cells from two types of cancer, triple negative breast cancer and high-grade serous ovarian [Dorri et al., 2020]) and mRNA transfection time series [Leonhardt et al., 2014], as well as more conventional ones such as mtDNA data and various feature selection/classification datasets. See Figure 6 and references therein for details. See also Appendix I for more information on the models and additional experimental results. All models and algorithms are implemented in the Blang modelling language [Bouchard-Côté et al., 2019].

In each of the 16 models and for each round of Algorithm 4 (line 4), we computed the mean swap acceptance probability across all neighbour chains. We then summarized these means across chains using a box plot. The equi-acceptance objective function of Section 5.1 can be visually understood as collapsing this box plot into a single point. We show in Figure 7 (top) the progression of these swap acceptance probabilities. In all examples considered, equi-acceptance is well approximated within 10 rounds.

Model (and dataset when applicable)	$n$	$d$	$\hat{\Lambda}$	$N$
<i>Change point</i> detection (text message data, [Davidson-Pilon, 2015])	74	3	4.0	20
<i>Copy number</i> inference (whole genome ovarian cancer data, Section 7.5)	6 206	30	13.0	50
<i>Discrete multimodal</i> distribution (Appendix I.4.3)	N/A	3	0.4	30
Weakly identifiable <i>elliptic curve</i> (Appendix I.4)	N/A	2	4.4	30
General Linear Model ( <i>GLM</i> ) (Challenger O-ring dataset, [Dalal et al., 1989])	23	2	3.3	15
Bayesian <i>hierarchical</i> model (historical rocket failure data, Appendix I.4)	5 667	369	12.0	30
<i>Ising</i> model (Appendix I.4.2)	N/A	25	3.1	30
Ising model with <i>magnetic field</i> (Appendix I.4.2)	N/A	25	2.3	30
Bayesian <i>mixture</i> model (Section I.4.4)	300	305	8.2	30
Bayesian <i>mixture</i> model (subset of 150 datapoints)	150	155	5.5	20
Isotropic <i>normal</i> distribution	N/A	5	1.4	30
<i>ODE</i> parameters (mRNA data, [Leonhardt et al., 2014])	52	5	6.4	50
<i>Phylogenetic</i> inference (single cell breast cancer data, [Dorri et al., 2020])	192 763	192 765	88	300
<i>Phylogenetic species</i> tree inference (mtDNA, [Hayasaka et al., 1988])	249	10 395	7.1	30
Unidentifiable <i>product</i> parameterization (Appendix I.4)	100 000	2	3.7	15
<i>Rotor</i> (XY) model, [Hsieh et al., 2013]	N/A	25	3.3	40
<i>Spike-and-slab</i> classification (RMS Titanic passengers data [Hind, 2019])	200	19	4.7	30

Figure 6: Summary of models used in the experiments, with the number of observations  $n$  (when applicable), the number of latent random variables  $d$ , estimated  $\hat{\Lambda}$ , and the default number of chains used. An abbreviation for each is shown in italic.

Within the range of models considered, we observed a diversity of local barriers  $\lambda$  estimated by Algorithm 4 (Figure 7 (middle)). Most statistical models exhibit a high but narrow peak in the neighbourhood of the reference ( $\beta = 0$ ). However, a subset of models including statistical models (mixture, ode, phylo-cancer, spike-slab) and physics models (Ising, magnetic, rotor) exhibit additional peaks away from  $\beta = 0$ . See Figure 20 in Appendix J.1 for the corresponding schedule generators  $G$  and global barriers  $\Lambda$ .

## 7.2 Robustness to ELE violation

To investigate empirically whether the NRPT methodology is robust to the violation of the ELE assumption, we ran NRPT with a range of values for  $n_{\text{expl}}$  on the models shown in Figure 8. Let  $d_{\text{var}}$  denote the number of variables in each model. We run experiments with  $n_{\text{expl}} = 0, (1/2)d_{\text{var}}, d_{\text{var}}, 2d_{\text{var}}, 4d_{\text{var}}, \dots, 32d_{\text{var}}$  (the only exception is the reference chain ( $\beta = 0$ ), where we always use  $n_{\text{expl}} = 1$  since we can get exact samples from  $\pi_0$ ). The key quantity used by the NRPT algorithm for schedule optimization is the communication barrier  $\lambda$ . The results shown in Figure 8 demonstrate that in all models considered the function  $\lambda$  is reliably estimated even when ELE is severely violated, provided  $n_{\text{expl}} > 0$ . Similarly, since the estimates of  $\Lambda$  and  $\bar{\tau}$  are derived from  $\lambda$ , these quantities can be accurately estimated even when ELE is severely violated (see Figure 19 in Appendix I.4.2). The results shown in Figure 3 support that increasing  $n_{\text{expl}}$  reduces the difference between the theoretical and observed round trip rate. Moreover, Figure 3 also demonstrate a second strategy to alleviate ELE violation, which is to increase  $N$  while fixing  $n_{\text{expl}}$ .

In the next experiment shown in Figure 9, we compare the two mechanisms available for alleviating ELE violation, namely increasing  $N$  and increasing  $n_{\text{expl}}$ . First consider the black line in Figure 9 showing the regime where ELE is best approximated in these experiments. This is in

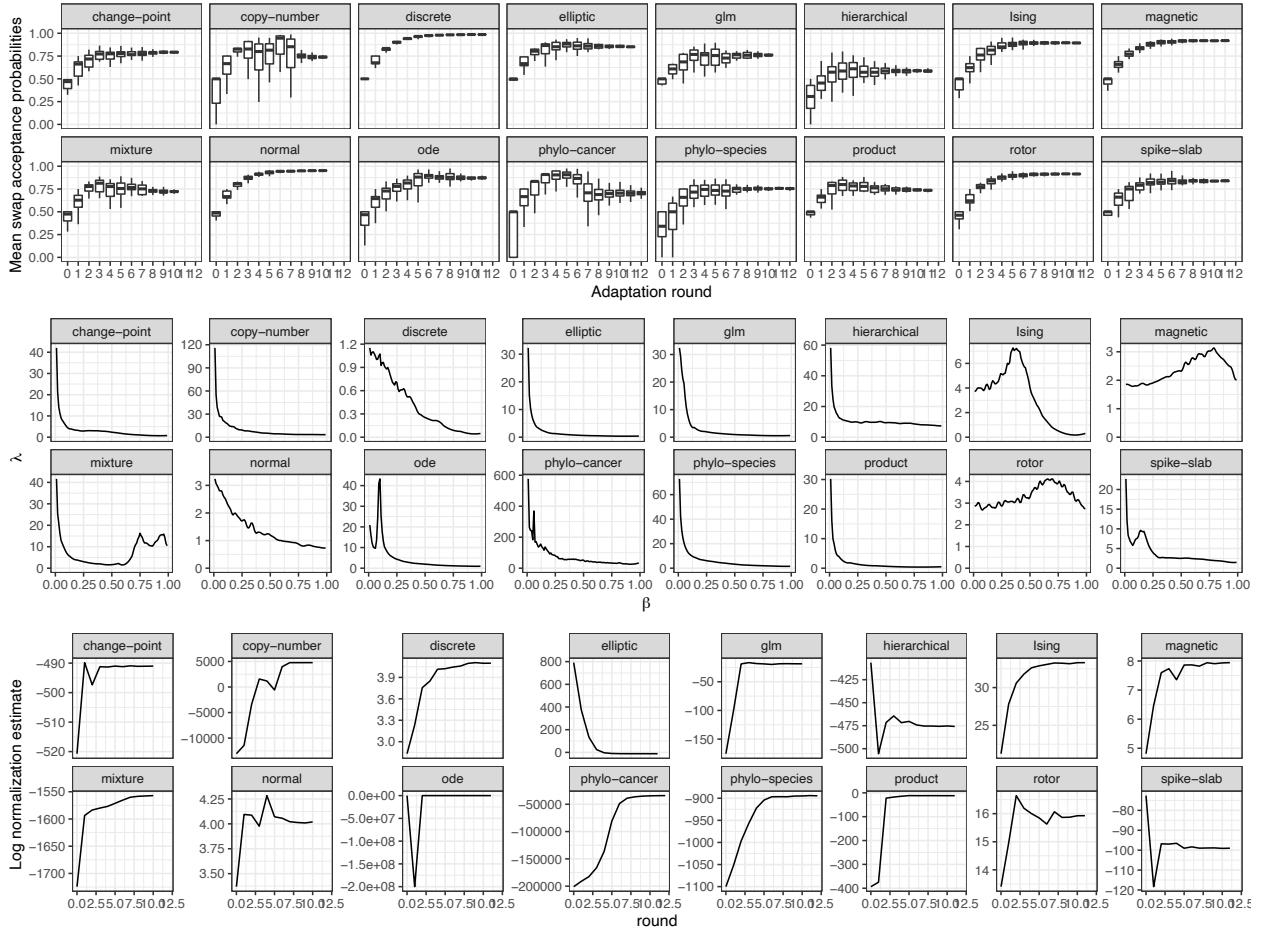


Figure 7: Empirical behaviour of NRPT on 16 models. Top: distribution of acceptance rates observed at each round of Algorithm 4. Middle: estimates of the local communication barrier  $\hat{\lambda}$ . Bottom: progression of the log normalization constant estimates (Section 5.5) produced during each NRPT adaptation round.

close agreement with the theoretical guidelines developed in Section 5.3. At the same time, the light blue lines in Figure 9 show that in situations where the local exploration kernel is not easy to parallelize, it can be advantageous to decrease  $n_{\text{expl}}$  and to compensate with a number of chains  $N$  higher than  $2\Lambda$ , leading to an average swap rejection rate  $r < 1/2$ .

### 7.3 Comparison with other parallel tempering schemes

We benchmarked the empirical running time of Algorithm 4 compared to previous adaptive PT methods [Atchadé et al., 2011, Miasojedow et al., 2013]. The methods we considered are: (1) the stochastic optimization adaptive method for reversible schemes proposed in [Atchadé et al., 2011]; (2), a second stochastic optimization scheme, which still selects the optimal number of chains using the 23% rule but uses an improved update scheme from [Miasojedow et al., 2013], refer to Appendix I.3 for details; (3) our non reversible schedule optimization scheme (NRPT); and finally, (4) our scheme, combined with a better initialization based on a preliminary execution of

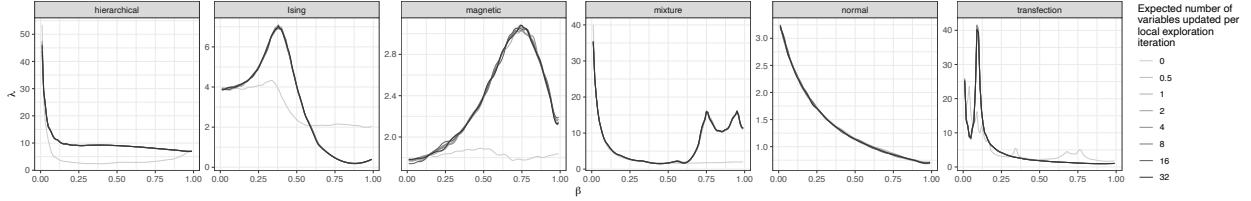


Figure 8: Estimate  $\hat{\lambda}$  of the local communication barrier  $\lambda$  for different values of  $n_{\text{expl}} \geq 0$  and different models (refer to Figure 6 for model descriptions). The estimates are consistent as long as  $n_{\text{expl}} > 0$ , even when assumption (A2) is severely violated. Refer to Section 7.2 for experiment details.

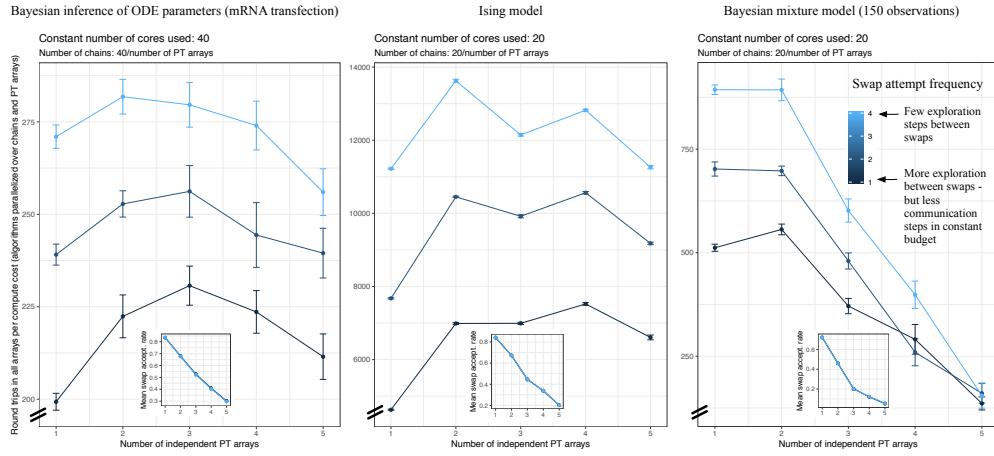


Figure 9: Trade-off between number of chains  $N$ , number of independent PT algorithms,  $k$ , and the frequency at which swaps are attempted ( $\propto 1/n_{\text{expl}}$ ). Each model uses a constant number of cores and set  $N+1 = \bar{N}/k$  chains per independent PT algorithm. Insets: swap acceptance probability for each  $k$ . The results in the black lines (where ELE is best approximated) agree with the theory of Section 5.3. For the ODE model,  $\Lambda \approx 6.4$ ,  $N^* \approx 12.8$  in agreement with  $k^* = 3$  observed. For the Ising model,  $\Lambda \approx 3.1$ ,  $N^* \approx 6.2$  in agreement with  $k^* = 4$  observed. For the Bayesian mixture model,  $\Lambda \approx 5.5$ ,  $N^* \approx 11$ , in agreement with  $k^* = 2$  observed. These results also show that in the context of local exploration moves that are not easily parallelized, it is better to use higher swap attempt frequencies (light blue lines) which achieve their optima at  $N > 2\Lambda$  and average swap rejection rate  $r < 1/2$ .

a sequential Monte Carlo algorithm (more precisely, based on a “sequential change of measure,” labelled SCM, as described in [Del Moral et al., 2006]), we use this to investigate the effect on the violation of the stationarity assumption, and for fairness, we use this sophisticated initialization method for all the methods except (3). We benchmarked the methods on four models: (a) a 369-dimensional hierarchical model applied to a dataset of rocket launch failure/success indicator variables [McDowell, 2019]; (b) a 19-dimensional Spike-and-Slab variable selection model applied to the RMS Titanic Passenger Manifest dataset [Hind, 2019]; (c) A 25-dimensional Ising model from Section I.4.2 ( $M = 5$ ); (d) a 9-dimensional model for an end-point conditioned Wright-Fisher stochastic differential equation (see, e.g., [Tataru et al., 2017]).

We provide the same number of cores and parallelized all the methods to the greatest extent possible. We refer the reader to Appendix I for implementation details, experimental setup, and

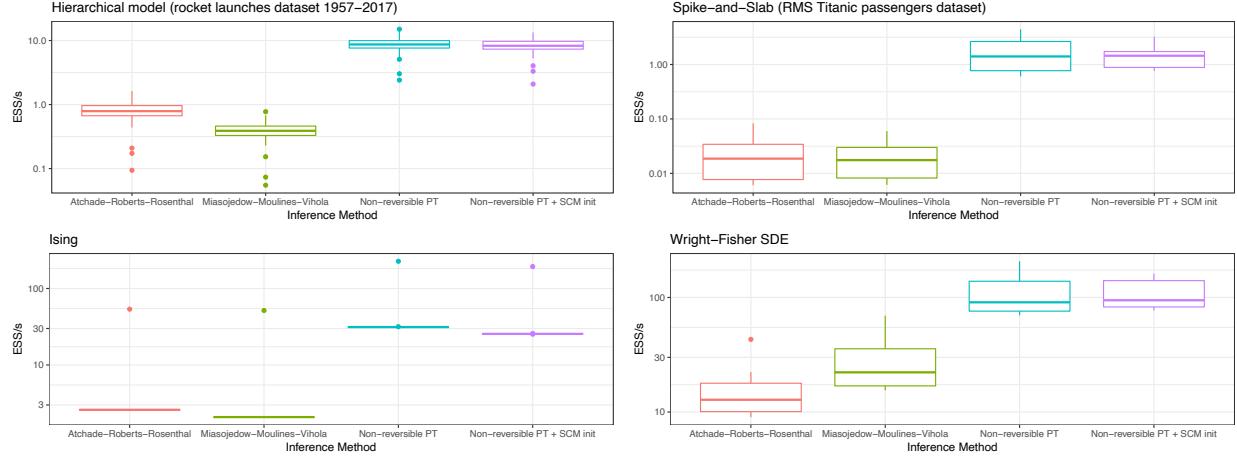


Figure 10: Effective Sample Size (ESS) per second (ordinate, in log scale) for four PT methods (abscissa). The four facets show results for the four models described in Section 7.3.

detailed description of the models and datasets used.

In Figure 10, each dot summarized in the box plots represents the ESS per total wall clock time in seconds including schedule optimization time for the marginal of one of the model variables. The results show that schedule optimization is more efficient with our proposed non-reversible scheme, with a speed-up in the 10–100 range for the four models considered. The results also suggest that sequential Monte Carlo based initialization may not have a large impact on the performance of our method. See Figure 21 in Appendix J.2 for other summary statistics from this benchmark.

## 7.4 Mixture models

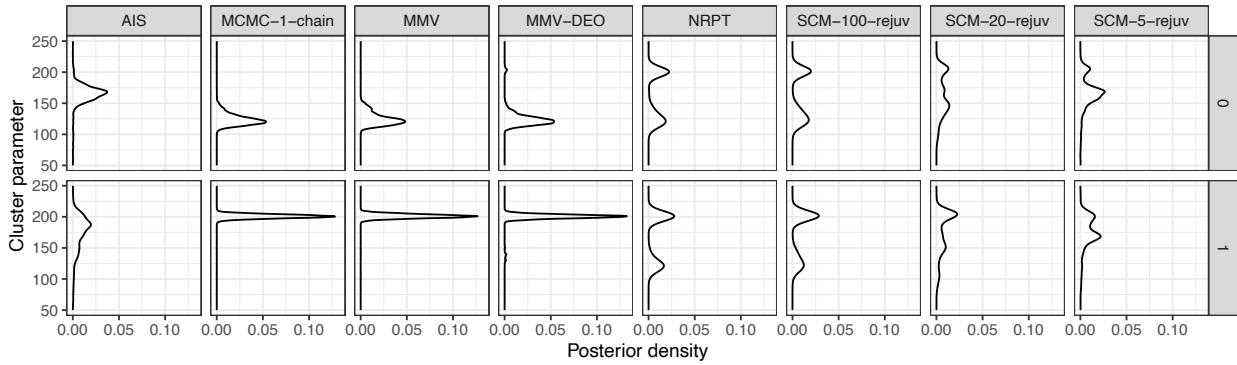


Figure 11: Mixture model example: eight approximations of the posterior distributions, showing for each approximation two of the 155 latent random variables, namely the two component mixtures’ mean parameters. Since the two latent random variables are exchangeable a posteriori (by label switching), the true marginal posterior distributions are identical for the two random variables. Amongst the approximation methods, multimodality is only captured by our algorithm (NRPT, 1.508min) and by a state-of-the-art sequential Monte Carlo combined with several rounds of rejuvenation (SCM-100, 5.064min). The benchmark is conservative in that all competing methods use a computational budget (in terms of both parallelism and wall clock time) greater or equal than NRPT.

Bayesian analysis of mixture models can give rise to a label-switching symmetry, leading to a multimodal posterior distribution (see Appendix I.4.4). In Figure 11 we compare the following inference methods on a label-switching posterior distribution: our proposed algorithm (NRPT), an MCMC run based on a single chain (i.e. the exploration kernel alone), the stochastic optimization method of Miasojedow-Moulines-Vihola (MMV), DEO but optimized using MMV (MMV-DEO), Annealed Importance Sampling (AIS) [Neal, 2001], and a sequential Monte Carlo based on a sequence of annealed distributions [Del Moral et al., 2006] (labelled SCM as before). For both SCM and AIS, we use the adaptive scheme of [Zhou et al., 2016] (see Figure 24 in Appendix J.3 for diagnostics of the SCM adaptation). The number of iterations are set so that the method with the smallest wall clock time is our proposed algorithm (NRPT: 1.508min, MMV: 2.019min, MMV-DEO: 1.665min, AIS: 1.887min, SCM: 1.778min–5.064min). For all methods, timing includes the time spent to perform schedule optimization.

Amongst MCMC methods, only NRPT correctly captures the multimodality of the target distribution. This is confirmed by the trace plots of the three MCMC methods, shown in Figure 25 in Appendix J.3. MMV automatically selected  $N = 8$  by targeting an swap acceptance probability of 23%. Post-adaptation, the swap acceptance probability was 27%. For NRPT we use  $N = 30$  but to avoid penalizing MMV for a lack of parallelism, we limited all methods to use no more than 8 threads. After schedule optimization, NRPT estimates the global communication barrier to a value of  $\hat{\Lambda} \approx 8$ , and the average swap acceptance probability was 72% (see Figure 26 in Appendix J.3 for more NRPT diagnostics).

## 7.5 Multimodality arising from single cell, whole genome copy number inference

In this section we describe an application to copy number inference in which multimodality arises from the unknown ploidy of a cancer cell. The likelihood of this model is based on a hidden Markov model over  $n = 6\,206$  observations, and after analytic marginalization of the corresponding 6 206 hidden states, the multimodal sampling problem is defined over  $d = 30$  remaining latent variables. The model is described in Appendix I.4.5, in this section we summarize the results concerning the performance of the algorithms.

We compare the quality of posterior approximations from four samplers on the High-Grade Serous Ovarian cancer dataset from [Dorri et al., 2020]. The four methods compared are: the stochastic optimization adaptive PT method MMV (adaptation selected 7 chains, average post-adaptation swap acceptance, 28.7%), DEO but optimized using MMV (MMV-DEO, adaptation selected 11 chains, average post-adaptation swap acceptance, 26.9%) our proposed algorithm algorithm (NRPT with 25 and 50 chains, average post-optimization swap acceptance of respectively 48.8% and 73.5%), and inference based on a single MCMC chain. As in Section 7.4, to avoid overly penalizing MMV for a lack of parallelism, we limited all methods to use no more than 8 threads, hence obtaining the following wall clock running times including schedule optimization when applicable: MMV, 15.91h; MMV+DEO, 14.33h; 25 chains NRPT, 8.809h; 50 chains NRPT, 15.35h;

1 chain MCMC, 2.91h. The running time of MMV was dominated by adaptation (88% of the wall clock time), and is higher than the schedule optimization time used by NRPT (67%).

Refer to Figure 12 for a comparison of the trace plots between all three methods. Since the experimental protocol provides only proportionality between read counts and copy number, not absolute expected read count for a given copy number, the likelihood does not distinguish between a given copy number profile, and another one obtained by genome duplication of the same profile (refer to Appendix I.4.5 for details). The prior favours the former, but when local copy number events occur after the genome duplication it may be possible to detect recently evolved genome duplication. Challenging multimodality arises when these subsequent events involve only small regions. In such case, the tension between the prior distribution favouring low ploidy and the noisy observation favouring higher ploidy creates a multimodal posterior distribution. We use data from one such cell in the following experiments to illustrate a realistic multimodal inference problem.

The multimodality of the posterior distribution was successfully captured by NRPT but not by a single-chain MCMC nor by the stochastic optimization method MMV (Figure 12). MMV failed to achieve any round trip because the learnt schedule is highly suboptimal: while the mean swap acceptance probability across chains is 28.7%, the minimum across chains is nearly zero due to the noise in the optimization procedure. The difficulty of exploring this particular multimodal target is compounded by the fact that the state space of each chromosome’s copy number is random, being upper-bounded by random variables  $m_c$  corresponding to each chromosome  $c$ . So in order to perform a jump doubling the cell’s ploidy the sampler has to increase a large number of discrete variables  $m_c$  simultaneously. This is reflected in the posterior distribution of the model variables denoted  $m_c$ , shown in Figure 28 in Appendix J.4. Hence the local exploration kernel, which in this example samples the variables  $m_c$  one at the time, is insufficient to jump mode, and only excursions through the prior can achieve mode jumping, via a regeneration based on sampling from the prior at  $\beta = 0$ .

NRPT used 50 chains, estimated  $\hat{\Lambda} = 13$  (see Figure 30 in Appendix J.4), and converged quickly in this large scale example, performing drastic changes in the annealing schedule in the first 8 rounds then relatively little changes in rounds 8 through the final tenth round (Figure 30).

Comparing the performance of the two NRPT variants, we estimated a round trip rate of  $9 \times 10^{-4}$  for 25-chain, and  $6 \times 10^{-3}$  for 50-chain. Notice the increase being more than a factor two supports the use of large  $N$  for exploration of complex multimodal posterior distributions. Neither MMV nor MMV-DEO achieved any round trips in the post-burn-in iterations.

## 8 Discussion

We have established that under the ELE assumption, non-reversible PT dominates reversible PT in terms of round trip rates for any choice of  $N$  and annealing schedule  $\mathcal{P}_N$  (Corollary 1). Moreover we have characterized the optimal round trip rate in terms of the communication barrier  $\Lambda$ , which we have used to identify an intrinsic limitation of both reversible and non-reversible PT (Section 4.2),

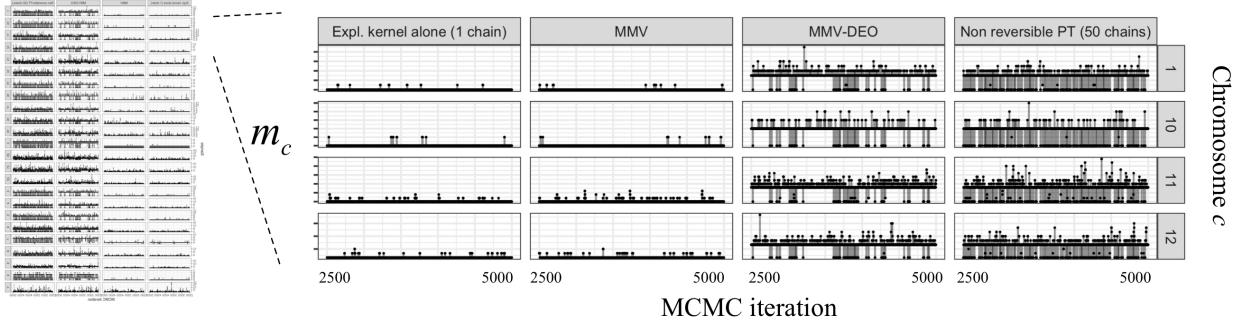


Figure 12: Trace plots for the copy number inference problem. Each facet shows a post burn-in trace plot for one of the discrete random variables  $m_c$ . Facet rows are indexed by chromosomes. Facet columns are indexed by MCMC approximation methods. The trace plots show that only NRPT frequently jumps between the two modes. MMV and the exploration kernel alone do not achieve any jump, due to the required concerted changes in a large number of discrete variables. See Figure 29 in Appendix J.4 for a larger version.

and to design a novel schedule optimization method (Section 5). We have shown that this schedule optimization method outperforms previous schedule optimizers in a range of numerical simulations, and we are already aware of other research groups that have successfully applied our methodology in astronomy [Akiyama et al., 2021, Issaoun et al., 2021] and engineering [Langmore et al., 2021].

By combining (28) with the estimate  $r^* \approx 1/2$  derived from (34), our theoretical results indicate a  $1/r^* \approx 2$ -fold increase in the round trip rate of DEO compared to SEO in an idealized setting where the ELE assumption holds and the schedule, including the number of chains, is well-tuned. The efficiency gains can be more pronounced during schedule optimization, in the course of which the number of chains can be initially larger than the optimal number of chains specified in Section 5.3 as a consequence of Theorem 3. Moreover, our experimental results show that previous methods to tune PT schedules can be fragile. Compared to these previous methods, we have observed speedups of one to two orders of magnitude thanks to our schedule optimization method.

## 9 References

- [Akiyama et al., 2021] Akiyama, K. et al. (2021). First M87 Event Horizon Telescope Results. VII. Polarization of the Ring. *The Astrophysical Journal Letters*, 910:12–59.
- [Altekar et al., 2004] Altekar, G., Dwarkadas, S., Huelsenbeck, J. P., and Ronquist, F. (2004). Parallel Metropolis coupled Markov chain Monte Carlo for Bayesian phylogenetic inference. *Bioinformatics*, 20(3):407–415.
- [Amari, 2016] Amari, S.-i. (2016). *Information Geometry and Its Applications*, volume 194. Springer.

- [Atchadé et al., 2011] Atchadé, Y. F., Roberts, G. O., and Rosenthal, J. S. (2011). Towards optimal scaling of Metropolis-coupled Markov chain Monte Carlo. *Statistics and Computing*, 21(4):555–568.
- [Ballnus et al., 2017] Ballnus, B., Hug, S., Hatz, K., Görlitz, L., Hasenauer, J., and Theis, F. J. (2017). Comprehensive benchmarking of Markov chain Monte Carlo methods for dynamical systems. *BMC Systems Biology*, 11(1):63.
- [Baxter, 2007] Baxter, R. J. (2007). *Exactly Solved Models in Statistical Mechanics*. Dover books on Physics. Dover Publications.
- [Beroukhim et al., 2010] Beroukhim, R. et al. (2010). The landscape of somatic copy-number alteration across human cancers. *Nature*, 463(7283):899–905.
- [Bierkens et al., 2018] Bierkens, J., Bouchard-Côté, A., Doucet, A., Duncan, A. B., Fearnhead, P., Lienart, T., Roberts, G. O., and Vollmer, S. J. (2018). Piecewise deterministic Markov processes for scalable Monte Carlo on restricted domains. *Statistics & Probability Letters*, 136:148–154.
- [Billingsley, 2013] Billingsley, P. (2013). *Convergence of Probability Measures*. John Wiley & Sons.
- [Böttcher et al., 2013] Böttcher, B., Schilling, R., and Wang, J. (2013). Lévy matters. III, volume 2099 of Lecture Notes in Mathematics.
- [Bouchard-Côté et al., 2019] Bouchard-Côté, A., Chern, K., Cubranic, D., Hosseini, S., Hume, J., Lepur, M., Ouyang, Z., and Sgarbi, G. (2019). Blang: Bayesian declarative modelling of arbitrary data structures. *arXiv:1912.10396*.
- [Chandra et al., 2019] Chandra, R., Jain, K., Deo, R. V., and Cripps, S. (2019). Langevin-gradient parallel tempering for Bayesian neural learning. *Neurocomputing*, 359:315–326.
- [Chen et al., 1999] Chen, F., Lovász, L., and Pak, I. (1999). Lifting Markov chains to speed up mixing. In *Proceedings of the 31st Annual ACM Symposium on Theory of Computing*, pages 275–281. ACM.
- [Cho et al., 2010] Cho, K., Raiko, T., and Ilin, A. (2010). Parallel tempering is efficient for learning restricted Boltzmann machines. In *Neural Networks (IJCNN), The 2010 International Joint Conference on*, pages 1–8. IEEE.
- [Dabak and Johnson, 2002] Dabak, A. G. and Johnson, D. H. (2002). Relations between Kullback–Leibler distance and Fisher information. Technical report, Rice University.
- [Dalal et al., 1989] Dalal, S. R., Fowlkes, E. B., and Hoadley, B. (1989). Risk Analysis of the Space Shuttle: Pre-Challenger Prediction of Failure. *Journal of the American Statistical Association*, 84(408):945–957.

- [Davidson-Pilon, 2015] Davidson-Pilon, C. (2015). *Bayesian Methods for Hackers: Probabilistic Programming and Bayesian Inference*. Addison-Wesley Professional, New York, 1st edition.
- [Davis, 1993] Davis, M. H. (1993). *Markov Models & Optimization*. Chapman and Hall.
- [Del Moral et al., 2006] Del Moral, P., Doucet, A., and Jasra, A. (2006). Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 68(3):411–436.
- [Desjardins et al., 2014] Desjardins, G., Luo, H., Courville, A., and Bengio, Y. (2014). Deep tempering. *arXiv preprint arXiv:1410.0123*.
- [Diaconis et al., 2000] Diaconis, P., Holmes, S., and Neal, R. M. (2000). Analysis of a nonreversible Markov chain sampler. *The Annals of Applied Probability*, 10(3):726–752.
- [Diaz et al., 2020] Diaz, A., Argüelles, C., Collin, G., Conrad, J., and Shaevitz, M. (2020). Where are we with light sterile neutrinos? *Physics Reports*, 884:1–59.
- [Dorri et al., 2020] Dorri, F., Salehi, S., Chern, K., Funnell, T., Williams, M., Lai, D., Andronescu, M., Campbell, K. R., McPherson, A., Aparicio, S., Roth, A., Shah, S. P., and Bouchard-Côté, A. (2020). Efficient Bayesian inference of phylogenetic trees from large scale, low-depth genome-wide single-cell data. *bioRxiv 2020.05.06.058180*.
- [Ethier and Kurtz, 2009] Ethier, S. N. and Kurtz, T. G. (2009). *Markov Processes: Characterization and Convergence*, volume 282. John Wiley & Sons.
- [Fang et al., 2014] Fang, Y., Feng, S., Tam, K.-M., Yun, Z., Moreno, J., Ramanujam, J., and Jarrell, M. (2014). Parallel tempering simulation of the three-dimensional Edwards-Anderson model with compact asynchronous multispin coding on GPU. *Computer Physics Communications*, 185(10):2467–2478.
- [Friesen, 2015] Friesen, J. (2015). *Java Threads and the Concurrency Utilities*. Apress, Berkeley, CA, USA, 1st edition.
- [Fritsch and Carlson, 1980] Fritsch, F. and Carlson, R. (1980). Monotone piecewise cubic interpolation. *SIAM Journal on Numerical Analysis*, 17(2):238–246.
- [Gelman and Meng, 1998] Gelman, A. and Meng, X.-L. (1998). Simulating normalizing constants: From importance sampling to bridge sampling to path sampling. *Statistical Science*, pages 163–185.
- [Geyer, 1991] Geyer, C. J. (1991). Markov chain Monte Carlo maximum likelihood. *Interface Proceedings*.

- [Hayasaka et al., 1988] Hayasaka, K., Gojobori, T., and Horai, S. (1988). Molecular phylogeny and evolution of primate mitochondrial DNA. *Molecular Biology and Evolution*, 5(6):626–644. Publisher: Oxford Academic.
- [Hind, 2019] Hind, P. (2019). RMS Titanic passenger dataset. data retrieved from <https://tinyurl.com/y55c8kc7>.
- [Hsieh et al., 2013] Hsieh, Y.-D., Kao, Y.-J., and Sandvik, A. W. (2013). Finite-size scaling method for the Berezinskii-Kosterlitz-Thouless transition. *Journal of Statistical Mechanics: Theory and Experiment*, 2013(09):P09001.
- [Hukushima and Nemoto, 1996] Hukushima, K. and Nemoto, K. (1996). Exchange Monte Carlo method and application to spin glass simulations. *Journal of the Physical Society of Japan*, 65(6):1604–1608.
- [Issaoun et al., 2021] Issaoun, S. et al. (2021). Persistent non-Gaussian structure in the image of Sagittarius A\* at 86 GHz. *arXiv:2104.07610*.
- [Kallenberg, 2002] Kallenberg, O. (2002). *Foundations of Modern Probability*. Springer, 2nd edition.
- [Kamberaj, 2020] Kamberaj, H. (2020). *Molecular Dynamics Simulations in Statistical Physics: Theory and Applications*. Springer.
- [Katzgraber et al., 2006] Katzgraber, H. G., Trebst, S., Huse, D. A., and Troyer, M. (2006). Feedback-optimized parallel tempering Monte Carlo. *Journal of Statistical Mechanics: Theory and Experiment*, 2006(03):P03018.
- [Kirkwood, 1935] Kirkwood, J. G. (1935). Statistical mechanics of fluid mixtures. *The Journal of Chemical Physics*, 3(5):300–313.
- [Kofke, 2002] Kofke, D. A. (2002). On the acceptance probability of replica-exchange Monte Carlo trials. *The Journal of Chemical Physics*, 117(15):6911–6914.
- [Kone and Kofke, 2005] Kone, A. and Kofke, D. A. (2005). Selection of temperature intervals for parallel-tempering simulations. *The Journal of Chemical Physics*, 122(20):206101.
- [Lacki and Miasojedow, 2016] Lacki, M. K. and Miasojedow, B. (2016). State-dependent swap strategies and automatic reduction of number of temperatures in adaptive parallel tempering algorithm. *Statistics and Computing*, 26(5):951–964.
- [Lai et al., 2020] Lai, D., Ha, G., and Shah, S. (2020). HMMcopy: Copy number prediction with correction for GC and mappability bias for HTS data.

- [Langmore et al., 2021] Langmore, I., Dikovsky, M., Geraedts, S., Norgaard, P., and von Behren, R. (2021). Hamiltonian Monte Carlo in Inverse Problems; Ill-Conditioning and Multi-Modality. *arXiv:2103.07515*.
- [Leiserson et al., 2012] Leiserson, C. E., Schardl, T. B., and Sukha, J. (2012). Deterministic parallel random-number generation for dynamic-multithreading platforms. *MIT web domain*.
- [Lelièvre et al., 2010] Lelièvre, T., Stoltz, G., and Rousset, M. (2010). *Free Energy Computations: A Mathematical Perspective*. World Scientific.
- [Leonhardt et al., 2014] Leonhardt, C., Schwake, G., Stögbauer, T. R., Rappl, S., Kuhr, J.-T., Ligon, T. S., and Rädler, J. O. (2014). Single-cell mRNA transfection studies: delivery, kinetics and statistics by numbers. *Nanomedicine: Nanotechnology, Biology, and Medicine*, 10(4):679–688.
- [Lingenheil et al., 2009] Lingenheil, M., Denschlag, R., Mathias, G., and Tavan, P. (2009). Efficiency of exchange schemes in replica exchange. *Chemical Physics Letters*, 478(1-3):80–84.
- [Lunn et al., 2000] Lunn, D. J., Thomas, A., Best, N., and Spiegelhalter, D. (2000). WinBUGS - a Bayesian modelling framework: Concepts, structure, and extensibility. *Statistics and Computing*, 10(4):325–337.
- [Masoliver et al., 1992] Masoliver, J., Porra, J. M., and Weiss, G. H. (1992). Solutions of the telegrapher’s equation in the presence of traps. *Physical Review A*, 45(4):2222.
- [McDowell, 2019] McDowell, J. (2019). Launch logs. data retrieved from <https://tinyurl.com/y5veq9yf>.
- [Miasojedow et al., 2013] Miasojedow, B., Moulines, E., and Vihola, M. (2013). An adaptive parallel tempering algorithm. *Journal of Computational and Graphical Statistics*, 22(3):649–664.
- [Mingas and Bouganis, 2012] Mingas, G. and Bouganis, C.-S. (2012). Parallel tempering MCMC acceleration using reconfigurable hardware. In Choy, O. C. S., Cheung, R. C. C., Athanas, P., and Sano, K., editors, *Reconfigurable Computing: Architectures, Tools and Applications*, Lecture Notes in Computer Science, pages 227–238. Springer Berlin Heidelberg.
- [Müller and Bouckaert, 2020] Müller, N. F. and Bouckaert, R. R. (2020). Adaptive parallel tempering for BEAST 2. *bioRxiv:10.1101/603514*.
- [Nadler and Hansmann, 2007] Nadler, W. and Hansmann, U. H. E. (2007). Dynamics and optimal number of replicas in parallel tempering simulations. *Physical Review E*, 76(6):065701.
- [Neal, 2001] Neal, R. M. (2001). Annealed importance sampling. *Statistics and Computing*, 11(2):125–139.
- [Neal, 2003] Neal, R. M. (2003). Slice sampling. *The Annals of Statistics*, 31(3):705–767.

- [Ogata, 1989] Ogata, Y. (1989). A Monte Carlo method for high dimensional integration. *Numerische Mathematik*, 55:137–157.
- [Okabe et al., 2001] Okabe, T., Kawata, M., Okamoto, Y., and Mikami, M. (2001). Replica-exchange Monte Carlo method for the isobaric–isothermal ensemble. *Chemical Physics Letters*, 335(5-6):435–439.
- [Predescu et al., 2004] Predescu, C., Predescu, M., and Ciobanu, C. V. (2004). The incomplete beta function law for parallel tempering sampling of classical canonical systems. *The Journal of Chemical Physics*, 120(9):4119–4128.
- [Rathore et al., 2005] Rathore, N., Chopra, M., and de Pablo, J. J. (2005). Optimal allocation of replicas in parallel tempering simulations. *The Journal of Chemical Physics*, 122(2):024111.
- [Roberts and Rosenthal, 2014] Roberts, G. O. and Rosenthal, J. S. (2014). Minimising MCMC variance via diffusion limits, with an application to simulated tempering. *The Annals of Applied Probability*, 24(1):131–149.
- [Sakai and Hukushima, 2016] Sakai, Y. and Hukushima, K. (2016). Irreversible simulated tempering. *Journal of the Physical Society of Japan*, 85(10):104002.
- [Saranen and Seikkala, 1988] Saranen, J. and Seikkala, S. (1988). Solution of a nonlinear two-point boundary value problem with Neumann-type boundary data. *Journal of Mathematical Analysis and Applications*, 135(2):691–701.
- [Steele and Lea, 2013] Steele, G. and Lea, D. (2013). Splittable random application programming interface. <https://docs.oracle.com/javase/8/docs/api/java/util/SplittableRandom.html>. [Online; accessed 6-May-2019].
- [Swendsen and Wang, 1986] Swendsen, R. H. and Wang, J.-S. (1986). Replica Monte Carlo simulation of spin-glasses. *Physical Review Letters*, 57(21):2607.
- [Tataru et al., 2017] Tataru, P., Simonsen, M., Bataillon, T., and Hobolth, A. (2017). Statistical inference in the Wright-Fisher model using allele frequency data. *Systematic Biology*, 66(1):e30–e46.
- [Turitsyn et al., 2011] Turitsyn, K. S., Chertkov, M., and Vucelja, M. (2011). Irreversible Monte Carlo algorithms for efficient sampling. *Physica D: Nonlinear Phenomena*, 240(4-5):410–414.
- [Vucelja, 2016] Vucelja, M. (2016). Lifting: a nonreversible Markov chain Monte Carlo algorithm. *American Journal of Physics*, 84(12):958–968.
- [Woodard et al., 2009] Woodard, D. B., Schmidler, S. C., Huber, M., et al. (2009). Conditions for rapid mixing of parallel and simulated tempering on multimodal distributions. *The Annals of Applied Probability*, 19(2):617–640.

- [Wu, 2017] Wu, F. (2017). Irreversible Parallel Tempering and an Application to a Bayesian Non-parametric Latent Feature Model. Master’s thesis, Oxford University.
- [Xie et al., 2011] Xie, W., Lewis, P. O., Fan, Y., Kuo, L., and Chen, M.-H. (2011). Improving marginal likelihood estimation for Bayesian phylogenetic model selection. *Systematic Biology*, 60(2):150–160.
- [Zahn et al., 2017] Zahn, H., Steif, A., Laks, E., Eirew, P., VanInsberghe, M., Shah, S. P., Aparicio, S., and Hansen, C. L. (2017). Scalable whole-genome single-cell library preparation without preamplification. *Nature Methods*, 14(2):167–173.
- [Zhou et al., 2016] Zhou, Y., Johansen, A. M., and Aston, J. A. (2016). Toward automatic model comparison: An adaptive sequential Monte Carlo approach. *Journal of Computational and Graphical Statistics*, 25(3):701–726.
- [Zhu et al., 2020] Zhu, P., Bouchard-Côté, A., and Campbell, T. (2020). Slice Sampling for General Completely Random Measures. In *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, volume 36.

## Supplementary material

### Appendix A Markov kernel for the index process

For SEO, initialize  $(I_0, \varepsilon_0) = (i, \varepsilon)$ . We then define the Markov transition kernel,  $(I_{n+1}, \varepsilon_{n+1})|(I_n, \varepsilon_n) \sim P^{\text{SEO}}((I_n, \varepsilon_n), \cdot)$  in two steps. In the first step, simulate

$$I_{n+1}|(I_n, \varepsilon_n) = (i, \varepsilon) \sim \begin{cases} (i + \varepsilon) \wedge N \vee 0 & \text{with probability } s^{(i, i+\varepsilon)}, \\ i & \text{otherwise,} \end{cases} \quad (36)$$

where the expression “ $\wedge N \vee 0$ ” enforces the annealing parameter boundaries. In the second step, independently sample  $\varepsilon_{n+1} \sim \text{Unif}\{-1, +1\}$ .

Similarly for DEO, initialize  $(I_0, \varepsilon_0) = (i_0, \varepsilon_0)$ . Analogous to the SEO construction, we define  $(I_{n+1}, \varepsilon_{n+1})|(I_n, \varepsilon_n) \sim P^{\text{DEO}}((I_n, \varepsilon_n), \cdot)$  in two steps. We first update  $I_{n+1}|(I_n, \varepsilon_n) = (i, \varepsilon)$  as in (36), but apply the deterministic update in the second step,

$$\varepsilon_{n+1} = \begin{cases} \varepsilon & \text{if } I_{n+1} = i + \varepsilon, \\ -\varepsilon & \text{otherwise.} \end{cases} \quad (37)$$

### Appendix B Proof of Theorem 1

To simplify notation for the rest of the proof, let  $T_\uparrow$  and  $T_\downarrow$  be the hitting times to the target and reference defined by,

$$T_\uparrow = \min\{n : (I_n, \varepsilon_n) = (N, 1)\}, \quad T_\downarrow = \min\{n : (I_n, \varepsilon_n) = (0, -1)\}. \quad (38)$$

We will also denote

$$s_i = s^{(i-1, i)}, \quad r_i = r^{(i-1, i)}. \quad (39)$$

(a) If we define  $a_\bullet^i = \mathbb{E}_{\text{SEO}}(T_\bullet|I_0 = i)$  for  $i = 0, \dots, N$  and  $\bullet \in \{\uparrow, \downarrow\}$ , then we have

$$\mathbb{E}_{\text{SEO}}(T) = a_\uparrow^0 + a_\downarrow^N. \quad (40)$$

By the Markov property, for  $i = 1, \dots, N-1$ ,  $a_\bullet^i$  satisfies the recursion

$$a_\bullet^i = \frac{1}{2}s_{i+1}(a_\bullet^{i+1} + 1) + \frac{1}{2}s_i(a_\bullet^{i-1} + 1) + \frac{1}{2}(r_{i+1} + r_i)(a_\bullet^i + 1). \quad (41)$$

For  $i = 1, \dots, N$ , we substitute in  $b_\bullet^i = a_\bullet^i - a_\bullet^{i-1}$  into (41). After simplification,  $b_\bullet^i$  satisfies

the following recursive relation

$$-2 = s_{i+1}b_\bullet^{i+1} - s_i b_\bullet^i. \quad (42)$$

The solutions to (42) are

$$s_i b_\bullet^i = s_1 b_\bullet^1 - 2(i-1), \quad (43)$$

or equivalently

$$s_i b_\bullet^i = s_N b_\bullet^N + 2(N-i). \quad (44)$$

We now deal with the case of  $\uparrow$  and  $\downarrow$  separately.

- (a) To determine  $a_\uparrow^0$ , we note that if  $I_0 = 0$  then  $I_1 = 1$  with probability  $\frac{1}{2}s_1$  and  $I_1 = 0$  otherwise. So  $a_\uparrow^0$  satisfies

$$a_\uparrow^0 = \frac{1}{2}s_1(a_\uparrow^1 + 1) + \left(1 - \frac{1}{2}s_1\right)(a_\uparrow^0 + 1), \quad (45)$$

or equivalently

$$s_1 b_\uparrow^1 = -2. \quad (46)$$

Substituting this into (43) implies  $s_i b_\uparrow^i = -2i$ . By summing  $b_\uparrow^i = a_\uparrow^i - a_\uparrow^{i-1}$  from  $i = 1, \dots, N$  and, noting  $a_\uparrow^N = 0$ , we get

$$a_\uparrow^0 = \sum_{i=1}^N \frac{2i}{s_i}. \quad (47)$$

- (b) Similarly to determine  $a_\downarrow^N$  we note that if  $I_0 = N$  then  $I_1 = N-1$  with probability  $\frac{1}{2}s_N$  and  $I_1 = N$  otherwise. So  $a_\downarrow^N$  satisfies

$$a_\downarrow^N = \frac{1}{2}s_N(a_\downarrow^{N-1} + 1) + \left(1 - \frac{1}{2}s_N\right)(a_\downarrow^N + 1), \quad (48)$$

or equivalently

$$s_N b_\downarrow^N = 2. \quad (49)$$

Substituting this into (44) implies  $s_i b_\downarrow^i = 2 + 2(N-i)$ . By summing  $b_\downarrow^i = a_\downarrow^i - a_\downarrow^{i-1}$  from

$i = 1, \dots, N$  and, noting  $a_{\downarrow}^0 = 0$ , we get

$$a_{\downarrow}^N = \sum_{i=1}^N \frac{2(N-i)+2}{s_i}. \quad (50)$$

Substituting in (47) and (50) into (40), it follows that

$$\mathbb{E}_{\text{SEO}}(T) = \sum_{i=1}^N \frac{2i}{s_i} + \sum_{i=1}^N \frac{2(N-i)+2}{s_i} \quad (51)$$

$$= 2(N+1) \sum_{i=1}^N \frac{1}{s_i} \quad (52)$$

$$= 2N(N+1) + 2(N+1) \sum_{i=1}^N \frac{r_i}{s_i}. \quad (53)$$

- (b) If we define  $a_{\bullet}^{i,\varepsilon} = \mathbb{E}_{\text{DEO}}(T_{\bullet}|I_0 = i, \varepsilon_0 = \varepsilon)$  for  $i = 0, \dots, N$ ,  $\varepsilon \in \{+, -\}$  and  $\bullet \in \{\uparrow, \downarrow\}$ , then we have

$$\mathbb{E}_{\text{DEO}}(T) = a_{\uparrow}^{0,-} + a_{\downarrow}^{N,+}. \quad (54)$$

Note that for  $i = 1, \dots, N-1$   $a_{\bullet}^{i,\varepsilon}$  satisfies the recursion relations

$$a_{\bullet}^{i,+} = s_{i+1}(a_{\bullet}^{i+1,+} + 1) + r_{i+1}(a_{\bullet}^{i,-} + 1), \quad (55)$$

$$a_{\bullet}^{i,-} = s_i(a_{\bullet}^{i-1,-} + 1) + r_i(a_{\bullet}^{i,-} + 1). \quad (56)$$

If we substitute  $c_{\bullet}^i = a_{\bullet}^{i,+} + a_{\bullet}^{i-1,-}$ , and  $d_{\bullet}^i = a_{\bullet}^{i,+} - a_{\bullet}^{i-1,-}$  into (55) and (56) and simplify, we obtain

$$a_{\bullet}^{i+1,+} - a_{\bullet}^{i,+} = r_{i+1}d_{\bullet}^{i+1} - 1, \quad (57)$$

$$a_{\bullet}^{i,-} - a_{\bullet}^{i-1,-} = r_i d_{\bullet}^i + 1. \quad (58)$$

By subtracting and adding (57) and (58), we obtain a joint recursion relation for  $c_{\bullet}^i$  and  $d_{\bullet}^i$  of the form

$$c_{\bullet}^{i+1} - c_{\bullet}^i = r_{i+1}d_{\bullet}^{i+1} + r_i d_{\bullet}^i, \quad (59)$$

$$d_{\bullet}^{i+1} - d_{\bullet}^i = r_{i+1}d_{\bullet}^{i+1} + r_i d_{\bullet}^i - 2. \quad (60)$$

Note that (60) can be rewritten as

$$s_{i+1}d_{\bullet}^{i+1} - s_i d_{\bullet}^i = -2. \quad (61)$$

Once one has expressions for  $c_\bullet^i$  and  $d_\bullet^i$ , then we can recover  $a_\bullet^{i,\varepsilon}$  by using

$$a_\bullet^{i,+} = \frac{c_\bullet^i + d_\bullet^i}{2}, \quad (62)$$

$$a_\bullet^{i,-} = \frac{c_\bullet^i - d_\bullet^i}{2}. \quad (63)$$

We now deal with the  $\uparrow$  and  $\downarrow$  cases separately.

- (a) Note that  $a_\uparrow^{0,-} = a_\uparrow^{0,+} + 1$ . We can substitute this into (57) to get  $s_1 d_\uparrow^1 = -2$ , which combined with (61) implies

$$s_i d_\uparrow^i = -2i. \quad (64)$$

Since  $a_\uparrow^{N,+} = 0$  we have  $c_\uparrow^N = -d_\uparrow^N$ , so by summing (59) we get

$$2a_\uparrow^{0,-} = c_\uparrow^1 - d_\uparrow^1 \quad (65)$$

$$= c_\uparrow^N - d_\uparrow^1 - \sum_{i=1}^{N-1} (c_\uparrow^{i+1} - c_\uparrow^i) \quad (66)$$

$$= -d_\uparrow^N - d_\uparrow^1 - \sum_{i=1}^{N-1} (r_{i+1} d_\uparrow^{i+1} + r_i d_\uparrow^i) \quad (67)$$

$$= -s_N d_\uparrow^N - s_1 d_\uparrow^1 - 2 \sum_{i=1}^N r_i d_\uparrow^i. \quad (68)$$

After substituting in (64) into (68), we obtain

$$a_\uparrow^{0,-} = N + 1 + \sum_{i=1}^N \frac{2ir_i}{s_i}. \quad (69)$$

- (b) Note that  $a_\downarrow^{N,+} = a_\downarrow^{N,-} + 1$ . We can substitute this expression into (58) to get  $s_N d_\downarrow^N = 2$ , which combined with (61) implies

$$s_i d_\downarrow^i = 2(N - i + 1). \quad (70)$$

Since  $a_{\downarrow}^{0,-} = 0$  we have  $c_{\downarrow}^1 = d_{\downarrow}^1$ , so by summing (59) we get

$$2a_{\downarrow}^{N,+} = c_{\downarrow}^N + d_{\downarrow}^N \quad (71)$$

$$= c_{\downarrow}^1 + d_{\downarrow}^N + \sum_{i=1}^{N-1} (c_{\downarrow}^{i+1} - c_{\downarrow}^i) \quad (72)$$

$$= d_{\downarrow}^1 + d_{\downarrow}^N + \sum_{i=1}^{N-1} (r_{i+1}d_{\downarrow}^{i+1} + r_i d_{\downarrow}^i) \quad (73)$$

$$= s_1 d_{\downarrow}^1 + s_N d_{\downarrow}^N + 2 \sum_{i=1}^N r_i d_{\downarrow}^i. \quad (74)$$

After substituting in (70) into (74), we obtain

$$a_{\downarrow}^{N,+} = N + 1 + \sum_{i=1}^N \frac{2(N-i+1)r_i}{s_i}. \quad (75)$$

Finally, by substituting (69) and (75) into (54), it follows that

$$\mathbb{E}_{\text{DEO}}(T) = 2(N+1) + 2(N+1) \sum_{i=1}^N \frac{r_i}{s_i}. \quad (76)$$

## Appendix C Proof of Theorem 2

We begin by recalling the following estimate for  $r(\beta, \beta')$  from Proposition 1 in [Predescu et al., 2004]:

**Proposition 1.** [Predescu et al., 2004] Suppose  $V^3$  is integrable with respect to  $\pi_0$  and  $\pi$ . For  $\beta \leq \beta'$ , let  $\bar{\beta} = \frac{\beta+\beta'}{2}$ , then we have

$$r(\beta, \beta') = (\beta' - \beta)\lambda(\bar{\beta}) + O(|\beta' - \beta|^3), \quad (77)$$

where  $\lambda$  satisfies (22).

When  $\beta < \beta'$ ,  $(\beta' - \beta)\lambda(\bar{\beta})$  is the Riemann sum for  $\int_{\beta}^{\beta'} \lambda(b)db$  with a single rectangle. Let  $C^k([0, 1])$  be the set of  $k$  times continuously differentiable function on  $[0, 1]$ . If  $\lambda \in C^2([0, 1])$ , then standard midpoint rule error estimates yield

$$\left| \int_{\beta}^{\beta'} \lambda(b)db - (\beta' - \beta)\lambda(\bar{\beta}) \right| \leq \frac{1}{12} \left\| \frac{d^2 \lambda}{d\beta^2} \right\|_{\infty} |\beta' - \beta|^3. \quad (78)$$

Therefore, if  $\lambda \in C^2([0, 1])$  then we can substitute (78) in (77) in Proposition 1, to obtain Theorem 2. This follows from Proposition 2.

**Proposition 2.** If  $V^k$  is integrable with respect to  $\pi_0$  and  $\pi$ , then  $\lambda \in C^{k-1}([0, 1])$ .

*Proof.* Suppose  $V^k$  is integrable with respect to  $\pi_0$  and  $\pi$ , we want to show here that  $\lambda : [0, 1] \rightarrow \mathbb{R}_+$  given by

$$\lambda(\beta) = \frac{1}{2} \int_{\mathcal{X}^2} |V(x) - V(y)| \pi^{(\beta)}(x) \pi^{(\beta)}(y) dx dy \quad (79)$$

is in  $C^{k-1}([0, 1])$ . If we define  $L(x, y) = L(x)L(y)$  and  $\pi_0(x, y) = \pi_0(x)\pi_0(y)$ , we can rewrite (79) as

$$\lambda(\beta) = \frac{1}{2\mathcal{Z}(\beta)^2} \int_{\mathcal{X}^2} |V(x) - V(y)| L(x, y)^\beta \pi_0(x, y) dx dy \quad (80)$$

$$= \frac{g(\beta)}{2\mathcal{Z}(\beta)^2}, \quad (81)$$

where  $\mathcal{Z}, g : [0, 1] \rightarrow \mathbb{R}_+$  are defined by

$$\mathcal{Z}(\beta) = \int_{\mathcal{X}} L(x)^\beta \pi_0(x) dx, \quad (82)$$

$$g(\beta) = \int_{\mathcal{X}^2} |V(x) - V(y)| L(x, y)^\beta \pi_0(x, y) dx dy. \quad (83)$$

Since  $\mathcal{Z}(\beta) > 0$  on  $[0, 1]$ , if we can show that  $\mathcal{Z}, g \in C^{k-1}([0, 1])$  then it implies that  $\lambda \in C^{k-1}([0, 1])$ . This is established in Lemma 2.  $\square$

**Lemma 1.** *If  $V^k$  is integrable with respect to  $\pi_0$  and  $\pi$  for  $k \in \mathbb{N}$ . Then for all  $\beta \in [0, 1]$ ,  $j \leq k$ ,  $V^j$  is  $\pi^{(\beta)}$ -integrable.*

*Proof.* We begin by noting that for  $L > 0$  and  $\beta \in [0, 1]$ , we have  $L^\beta \leq 1 + L$ . This implies

$$\int_{\mathcal{X}} |V(x)|^k \pi^{(\beta)}(x) dx \quad (84)$$

$$= \frac{1}{\mathcal{Z}(\beta)} \int_{\mathcal{X}} |V(x)|^k L(x)^\beta \pi_0(x) dx \quad (85)$$

$$\leq \frac{1}{\mathcal{Z}(\beta)} \int_{\mathcal{X}} |V(x)|^k \pi_0(x) dx + \frac{1}{\mathcal{Z}(\beta)} \int_{\mathcal{X}} |V(x)|^k L(x) \pi_0(x) dx \quad (86)$$

$$= \frac{\mathcal{Z}(0)}{\mathcal{Z}(\beta)} \int_{\mathcal{X}} |V(x)|^k \pi_0(x) dx + \frac{\mathcal{Z}(1)}{\mathcal{Z}(\beta)} \int_{\mathcal{X}} |V(x)|^k \pi(x) dx \quad (87)$$

$$< \infty. \quad (88)$$

Therefore since  $V^k$  is  $\pi_0$  and  $\pi$ -integrable,  $V^k$  is  $\pi^{(\beta)}$ -integrable. Finally by Jensen's inequality we have for  $j \geq k$ ,

$$\int_{\mathcal{X}} |V(x)|^j \pi^{(\beta)}(x) dx \leq \left( \int_{\mathcal{X}} |V(x)|^k \pi^{(\beta)}(x) dx \right)^{\frac{j}{k}} < \infty. \quad (89)$$

$\square$

**Lemma 2.** Suppose  $V^k$  is integrable with respect to  $\pi_0$  and  $\pi$  for some  $k \in \mathbb{N}$  then:

(a)  $\mathcal{Z} \in C^k([0, 1])$  with derivatives satisfying,

$$\frac{d^j \mathcal{Z}}{d\beta^j} = \int_{\mathcal{X}} (-1)^j V(x)^j L(x)^\beta \pi_0(x) dx, \quad (90)$$

for  $j \leq k$ .

(b)  $g \in C^{k-1}([0, 1])$  with derivatives satisfying,

$$\frac{d^j g}{d\beta^j} = \int_{\mathcal{X}^2} (-1)^j |V(x) - V(y)| (V(x) + V(y))^j L(x, y)^\beta \pi_0(x, y) dx dy, \quad (91)$$

for  $j < k$ .

*Proof.* (a) Let  $h(x, \beta) = L(x)^\beta \pi_0(x) = \exp(-\beta V(x)) \pi_0(x)$  which satisfies

$$\frac{\partial^j}{\partial \beta^j} h(x, \beta) = (-1)^j V(x)^j L(x)^\beta \pi_0(x). \quad (92)$$

Note for all  $\beta \in [0, 1]$  and  $j \leq k$ ,

$$\sup_{\beta \in [0, 1]} \left| \frac{\partial^j}{\partial \beta^j} h(x, \beta) \right| \leq |V(x)|^j \pi_0(x) + |V(x)|^j L(x) \pi_0(x). \quad (93)$$

The left hand side of (93) dominates  $\frac{\partial^j h}{\partial \beta^j}$  uniformly in  $\beta$  and is integrable by Lemma 1. The result follows using the Leibniz integration rule.

(b) Let  $\tilde{h}(x, y, \beta) = |V(x) - V(y)| L(x, y)^\beta \pi_0(x, y)$ . By using  $\log L(x, y) = -V(x) - V(y)$ , we get

$$\frac{\partial^j}{\partial \beta^j} \tilde{h}(x, y, \beta) = (-1)^j |V(x) - V(y)| (V(x) + V(y))^j L(x, y)^\beta \pi_0(x, y). \quad (94)$$

Similar to (a), we have for all  $\beta \in [0, 1]$ ,  $j \leq k-1$ ,

$$\begin{aligned} \sup_{\beta \in [0, 1]} \left| \frac{\partial^j}{\partial \beta^j} \tilde{h}(x, y, \beta) \right| &\leq |V(x) - V(y)| |V(x) + V(y)|^j \pi_0(x, y) \\ &\quad + |V(x) - V(y)| |V(x) + V(y)|^j L(x, y) \pi_0(x, y), \end{aligned} \quad (95)$$

The left hand side of (95) dominates  $\frac{\partial^j \tilde{h}}{\partial \beta^j}$  uniformly in  $\beta$ . It is integrable by Lemma 1 and using the fact that  $V^k$  is integrable with respect to  $\pi_0$  and  $\pi$ . The result follows using the Leibniz integration rule.

□

## Appendix D Proof of Theorem 3

We first note that (b) and (c) follow immediately from (a) and Corollary 1. So to prove Theorem 3 it is sufficient to show (a).

We use the fact that for all  $\mathcal{P}_N$

$$\sum_{i=1}^N r^{(i-1,i)} \leq E(\mathcal{P}_N) \leq \frac{1}{\min_j s^{(j-1,j)}} \sum_{i=1}^N r^{(i-1,i)}. \quad (96)$$

By Theorem 2, we have  $\min_j s_j = 1 + O(\|\mathcal{P}_N\|)$  and, by Corollary 2, we have  $\sum_{i=1}^N r^{(i-1,i)} = \Lambda + O(N\|\mathcal{P}_N\|^3)$  which combined with (96) implies

$$E(\mathcal{P}_N) = \Lambda + O(\|\mathcal{P}_N\|). \quad (97)$$

Therefore  $E(\mathcal{P}_N)$  converges to  $\Lambda$  at a  $O(\|\mathcal{P}_N\|)$  rate as  $\|\mathcal{P}_N\| \rightarrow 0$ .

## Appendix E Multimodal decomposition of communication barrier

Since PT is often used to sample from multimodal targets, it is natural to ask how the communication barrier behaves under the presence of modes. Similar to [Woodard et al., 2009], we partition  $\mathcal{X}$  into the disjoint union  $\mathcal{X} = \bigcup_{k=1}^K \mathcal{X}_k$  where we have  $\mathcal{X}_k$  represents region of the  $k$ -th mode of  $\pi$  and the target decomposes as a mixture of its modes its modes  $\pi(x) = \sum_{k=1}^K p_k \pi(x|\mathcal{X}_k)$  where  $p_k = \pi(\mathcal{X}_k)$  and  $\pi(x|\mathcal{X}_k) = p_k^{-1} \pi(x) \mathbb{I}_{\mathcal{X}_k}(x)$  are the probability mass and distribution of the  $k$ -th mode. If we assume that the reference distribution puts the same relative mass on each mode as the target,  $\pi_0(\mathcal{X}_k) = \pi(\mathcal{X}_k)$ , then  $\pi^{(\beta)}$  decomposes as

$$\pi^{(\beta)}(x) \propto \sum_{i=1}^K p_k \pi^{(\beta)}(x|\mathcal{X}_k). \quad (98)$$

Similarly  $V(x) = \sum_{k=1}^K V_k(x) \mathbb{I}_{\mathcal{X}_k}(x)$  where  $V_k(x) = -\log(\pi(x|\mathcal{X}_k)/\pi_0(x|\mathcal{X}_k))$ . Define  $\lambda_{k,k'}(\beta)$  and  $\Lambda_{k,k'}$  as the local and global communication barrier between mode  $k$  and  $k'$  by

$$\lambda_{k,k'}(\beta) = \frac{1}{2} \mathbb{E} \left[ |V_k^{(\beta)} - V_{k'}^{(\beta)}| \right], \quad \Lambda_{k,k'} = \int_0^1 \lambda_{k,k'}(\beta) d\beta, \quad (99)$$

where  $V_k^{(\beta)} \stackrel{d}{=} V_k(X_k^{(\beta)})$  for  $X_k^{(\beta)} \sim \pi^{(\beta)}(x|\mathcal{X}_k)$  and  $V_k^{(\beta)}, V_{k'}^{(\beta)}$  are independent. An immediate consequence of (22) in Theorem 2 implies the following decomposition of the communication barrier.

**Proposition 3** (Multimodal decomposition). *If  $\pi_0(\mathcal{X}_k) = \pi(\mathcal{X}_k)$  then,*

$$\lambda(\beta) = \sum_{k=1}^K \sum_{k'=1}^K p_k p_{k'} \lambda_{k,k'}(\beta), \quad \Lambda = \sum_{k=1}^K \sum_{k'=1}^K p_k p_{k'} \Lambda_{k,k'}, \quad (100)$$

In particular (100) implies  $\Lambda$  is a weighted average over the communication barriers between modes. In the particular case where the modes are exchangeable, we have  $\Lambda_{k,k'} = \Lambda$  and thus  $\Lambda$  is exactly invariant to  $K$ . We remark that the meta-model used in this section is not realistic for practical problems as we do not know a priori the location of the modes. This meta-model is only used to obtain intuition on the multimodal scalability of the method. Section 7.1 illustrates the empirical behaviour of the method in several genuinely challenging multimodal problems.

## Appendix F High-dimensional scaling of communication barrier.

We determine here the asymptotic behaviour of  $\lambda$  and  $\Lambda$  when the dimension of  $\mathcal{X}$  is large. To make the analysis tractable, we assume that  $\pi_d(x) = \prod_{i=1}^d \pi(x_i)$  as in [Atchadé et al., 2011, Roberts and Rosenthal, 2014]. This provides a model for weakly dependent high-dimensional distributions. We only make this structural assumption on the state space and distribution to establish Proposition 4 below.

The corresponding annealed distributions are thus given by

$$\pi_d^{(\beta)}(x) = \prod_{i=1}^d \pi^{(\beta)}(x_i). \quad (101)$$

Let  $\lambda_d$  and  $\Lambda_d$  be the local and global communication barriers for  $\pi_d$  respectively.

**Proposition 4** (High Dimensional Scaling). *Define  $\sigma^2(\beta) = \text{Var}(V^{(\beta)})$ , for all  $\beta \in [0, 1]$  we have as  $d \rightarrow \infty$ ,*

$$\lambda_d(\beta) \sim \sqrt{\frac{d}{\pi}} \sigma(\beta), \quad \Lambda_d \sim \sqrt{\frac{d}{\pi}} \int_0^1 \sigma(\beta) d\beta. \quad (102)$$

It follows from Proposition 4 that  $\lambda_d$  and  $\Lambda_d$  increase at a  $O(d^{1/2})$  rate as  $d \rightarrow \infty$ .

*Proof.* For  $k = 1, 2$ , let us define  $\mathbf{V}_k^{(\beta)} \stackrel{d}{=} \mathbf{V}(X_k^{(\beta)})$  where  $X_k^{(\beta)} \sim \pi_d^{(\beta)}$  and  $\mathbf{V}(x) = \sum_{i=1}^d V(x_i)$ . The independence structure from Equation (101) tells us that  $\mathbf{V}_k^{(\beta)}$  can be decomposed as  $\mathbf{V}_k^{(\beta)} = \sum_{i=1}^d V_{ki}^{(\beta)}$  where  $V_{ki}^{(\beta)}$  are iid with a distribution identical to  $V^{(\beta)}$ , and therefore

$$\mathbf{V}_1^{(\beta)} - \mathbf{V}_2^{(\beta)} = \sum_{i=1}^d V_{1i}^{(\beta)} - V_{2i}^{(\beta)}. \quad (103)$$

The random variables  $\{V_{1i}^{(\beta)} - V_{2i}^{(\beta)}\}_{i=1}^d$  are iid with mean zero and variance  $2\sigma^2(\beta)$ . By the central limit theorem,

$$\frac{\mathbf{V}_1^{(\beta)} - \mathbf{V}_2^{(\beta)}}{\sqrt{2\sigma^2(\beta)d}} = \frac{1}{\sqrt{d}} \sum_{i=1}^d \frac{V_{1i}^{(\beta)} - V_{2i}^{(\beta)}}{\sqrt{2\sigma^2(\beta)}} \xrightarrow[d \rightarrow \infty]{} \tilde{Z} \sim N(0, 1). \quad (104)$$

Thus we have

$$\lambda_d(\beta) = \frac{1}{2} \mathbb{E} \left[ |\mathbf{V}_1^{(\beta)} - \mathbf{V}_2^{(\beta)}| \right] \quad (105)$$

$$= \frac{1}{2} \sqrt{2\sigma^2(\beta)d} \mathbb{E} \left[ \left| \frac{\mathbf{V}_1^{(\beta)} - \mathbf{V}_2^{(\beta)}}{\sqrt{2\sigma^2(\beta)d}} \right| \right]. \quad (106)$$

The sequence of variables indexed by  $d$  in the expectation in (106) is also uniformly integrable. This follows by noting that the second moment of the integrand in (106) is uniformly bounded in  $d$ :

$$\sup_d \mathbb{E} \left[ \left| \frac{\mathbf{V}_1^{(\beta)} - \mathbf{V}_2^{(\beta)}}{\sqrt{2\sigma^2(\beta)d}} \right|^2 \right] = \sup_d \frac{1}{2\sigma^2(\beta)d} \sum_{i=1}^d \text{Var} \left[ V_{1i}^{(\beta)} - V_{2i}^{(\beta)} \right] = 1. \quad (107)$$

By  $d \rightarrow \infty$  and using (104) we have

$$\lim_{d \rightarrow \infty} \sqrt{\frac{2}{\sigma^2(\beta)d}} \lambda_d(\beta) = \mathbb{E}|\tilde{Z}| = \sqrt{\frac{2}{\pi}}, \quad (108)$$

which proves (102).

To obtain the dimensional scaling limit for  $\Lambda_d$ , we use Jensen's inequality,

$$\frac{\lambda_d(\beta)}{\sqrt{d}} = \frac{1}{2\sqrt{d}} \mathbb{E} \left[ |\mathbf{V}_1^{(\beta)} - \mathbf{V}_2^{(\beta)}| \right] \quad (109)$$

$$\leq \frac{1}{2\sqrt{d}} \sqrt{\text{Var} \left[ |\mathbf{V}_1^{(\beta)} - \mathbf{V}_2^{(\beta)}| \right]} \quad (110)$$

$$= \frac{\sigma(\beta)}{\sqrt{2}}. \quad (111)$$

Finally, (102), (111) along with dominated convergence yield

$$\lim_{d \rightarrow \infty} \frac{\Lambda_d}{\sqrt{d}} = \int_0^1 \lim_{d \rightarrow \infty} \frac{\lambda_d(\beta)}{\sqrt{d}} d\beta = \int_0^1 \frac{\sigma(\beta)}{\sqrt{\pi}} d\beta. \quad (112)$$

□

## Appendix G Scaling limit of index process

### G.1 Scaled index process

To establish scaling limits for  $(I_n, \varepsilon_n)$ , it will be convenient to work in a continuous time setting. To do this, we suppose the times that PT iterations occur are distributed according to a Poisson process  $\{M(\cdot)\}$  with mean  $\mu_N$ . The number of PT iterations that occur by time  $t \geq 0$  satisfies  $M(t) \sim \text{Poisson}(\mu_N t)$ . We define the *scaled index process* by  $Z^N(t) = (W^N(t), \varepsilon^N(t))$  where

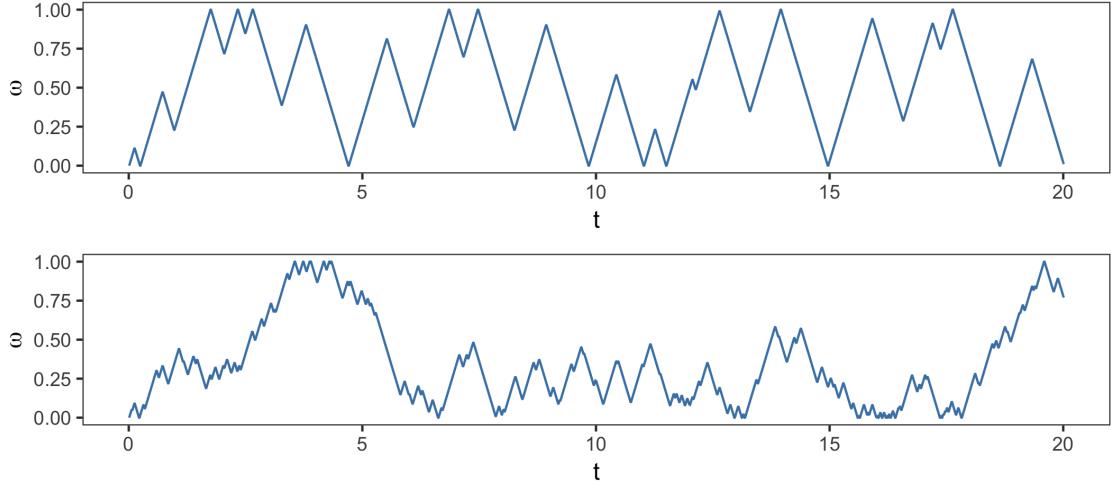


Figure 13: Sample trajectories of  $W(t)$  where  $Z(t) = (W(t), \varepsilon(t))$  under an optimal schedule generated by  $G = F^{-1}$  for  $F(\beta) = \Lambda(\beta)/\Lambda$  for  $\Lambda = 1$  (top) and  $\Lambda = 10$  (bottom) respectively.

$W^N(t) = I_{M(t)}/N$  and  $\varepsilon^N(t) = \varepsilon_{M(t)}$  (example realizations in Figure 13).

For convenience, we will denote  $\beta_w = G(w)$  and use  $z = (w, \varepsilon) \in [0, 1] \times \{-1, 1\}$  to be a *scaled index*. Define  $C(\mathbb{R}_+, \mathcal{S})$  and  $D(\mathbb{R}_+, \mathcal{S})$  to be set of functions  $f : \mathbb{R}_+ \rightarrow \mathcal{S}$  that are continuous and càdlàg respectively.

The process  $Z^N \in D(\mathbb{R}_+, [0, 1] \times \{-1, 1\})$  takes values on the discrete set  $\mathcal{P}_N^{\text{uniform}} \times \{-1, 1\}$  and is only well-defined when  $Z^N(0) = z_0 \in \mathcal{P}_N^{\text{uniform}} \times \{-1, 1\}$ . To establish convergence, it is useful to extend it to a process  $Z^N$  which can be initialized at any  $z_0 \in [0, 1] \times \{-1, 1\}$ . Suppose  $Z^N(0) = z_0 \in [0, 1] \times \{-1, 1\}$  and let  $T_1, T_2, \dots$  be the iteration times generated by the Poisson process  $M$ . We construct  $Z^N(t)$  as follows: define  $Z^N(t) = z_n$  for  $t \in [T_n, T_{n+1})$  and update  $z_{n+1}|z_n$  via a transition kernel dependent on the communication scheme. We determine this transition kernel mirroring the construction from Section 3.5.

Before doing this, it will be useful to define the backward and forward shift operators  $\Phi_-^N, \Phi_+^N : [0, 1] \rightarrow [0, 1]$  by,

$$\Phi_-^N(w) = \begin{cases} w - \frac{1}{N} & w \in [\frac{1}{N}, 1], \\ \frac{1}{N} - w & w \in [0, \frac{1}{N}), \end{cases} \quad (113)$$

and similarly,

$$\Phi_+^N(w) = \begin{cases} w + \frac{1}{N} & w \in [0, 1 - \frac{1}{N}], \\ 1 - (\frac{1}{N} - (1 - w)) & w \in (1 - \frac{1}{N}, 1]. \end{cases} \quad (114)$$

Intuitively  $\Phi_\varepsilon^N(w)$  represents the location in  $[0, 1]$  after  $w$  moves a distance  $\frac{1}{N}$  in the direction of  $\varepsilon$  with a reflection at 0 and 1.

## G.2 Scaled index process for reversible PT

Under the reversible communication scheme, if  $z_n = (w_n, \varepsilon_n) \in \{0, 1/N, \dots, 1\} \times \{-1, 1\}$ , then we have  $w_{n+1} = \Phi_{\varepsilon_n}^N(w_n)$  if a swap successfully occurred and  $w_{n+1} = w_n$  otherwise. In both cases,  $\varepsilon_{n+1} \sim \text{Unif}\{-1, +1\}$ . Since  $\Phi_\varepsilon^N(w)$  is not only well-defined for  $w \in \mathcal{P}_N^{\text{uniform}}$  but for  $w \in [0, 1]$ , we naturally extend this construction to any  $w \in [0, 1]$ .

Formally, we generate  $(w_{n+1}, \varepsilon_{n+1})$  in two steps. In the first step we simulate,

$$w_{n+1}|w_n, \varepsilon_n \sim \begin{cases} \Phi_{\varepsilon_n}^N(w_n) & \text{with probability } s(\beta_{w_n}, \beta_{\Phi_{\varepsilon_n}^N(w_n)}), \\ w_n & \text{otherwise.} \end{cases} \quad (115)$$

In the second step we simulate  $\varepsilon_{n+1} \sim \text{Unif}\{-1, +1\}$ . This defines a continuous time Markov pure jump process  $W^N \in D(\mathbb{R}_+, [0, 1])$  with jumps occurring according to an exponential of rate  $\mu_N$  and is well defined when initialized at any state  $w_0 \in [0, 1]$ .

From Theorem 19.2 in [Kallenberg, 2002], the infinitesimal generator for  $W^N$  with reversible communication is

$$\mathcal{L}_{W^N} f(w) = \frac{\mu_N}{2} \sum_{\varepsilon \in \{\pm 1\}} (f(\Phi_\varepsilon^N(w)) - f(w)) s(\beta_w, \beta_{\Phi_\varepsilon^N(w)}), \quad (116)$$

where the domain  $\mathcal{D}(\mathcal{L}_{W^N})$  is given by the set of functions such that  $\mathcal{L}_{W^N} f$  is continuous. Since  $\Phi_+^N, \Phi_-^N$  are continuous, we have  $\mathcal{D}(\mathcal{L}_{W^N}) = C([0, 1])$ .

## G.3 Scaled index process for non-reversible PT

Before defining the transition kernel for the scaled index process under non-reversible communication, it will be convenient to define the propagation function  $\Phi^N : [0, 1] \times \{-1, 1\} \rightarrow [0, 1] \times \{-1, 1\}$  for  $z = (w, \varepsilon)$ ,

$$\Phi^N(z) = \begin{cases} (\Phi_\varepsilon^N(w), \varepsilon) & \text{if } \Phi_\varepsilon^N(w) = w + \frac{\varepsilon}{N}, \\ (\Phi_\varepsilon^N(w), -\varepsilon) & \text{otherwise,} \end{cases} \quad (117)$$

and similarly the rejection function  $R : [0, 1] \times \{-1, 1\} \rightarrow [0, 1] \times \{-1, 1\}$ ,

$$R(z) = (w, -\varepsilon). \quad (118)$$

Under the non-reversible scheme, if  $z_n = (w_n, \varepsilon_n) \in \mathcal{P}_N^{\text{uniform}} \times \{-1, 1\}$ , then we have  $z_{n+1} = \Phi^N(z_n)$  when a swap is accepted and  $z_{n+1} = R(z_n)$  otherwise. Since  $\Phi^N(z)$  and  $R(z)$  are well-defined for all of  $z \in [0, 1] \times \{-1, 1\}$ , we naturally extend this construction to any  $z \in [0, 1] \times \{-1, 1\}$ .

Formally, we generate  $z_{n+1}$  according to the transition kernel,

$$z_{n+1}|z_n \sim \begin{cases} \Phi^N(z_n) & \text{with probability } s(\beta_{w_n}, \beta_{\Phi_{\varepsilon_n}^N(w_n)}), \\ R(z_n) & \text{otherwise.} \end{cases} \quad (119)$$

This defines a continuous time Markov pure jump process  $Z^N \in D(\mathbb{R}_+, [0, 1] \times \{-1, 1\})$  with jumps occurring at an exponential of rate  $\mu_N$ . This process is well defined when initialized at any  $z_0 \in [0, 1] \times \{-1, 1\}$ .

Analogously to the reversible case, under non-reversible communication, the infinitesimal generator for  $Z^N$  is

$$\mathcal{L}_{Z^N} f(z) = \mu_N (f(\Phi^N(z)) - f(z)) s(\beta_w, \beta_{\Phi_{\varepsilon}^N(w)}) + \mu_N (f(R(z)) - f(z)) r(\beta_w, \beta_{\Phi_{\varepsilon}^N(w)}), \quad (120)$$

where  $z = (w, \varepsilon)$  and  $\mathcal{D}(\mathcal{L}_{Z^N})$  is given by the set of functions  $f$  such that  $\mathcal{L}_{Z^N} f$  is continuous. Since  $\Phi^N$  has discontinuities at  $(\frac{1}{N}, -1)$  and  $(1 - \frac{1}{N}, 1)$ , we can verify that  $f \in \mathcal{D}(\mathcal{L}_{Z^N})$  if and only if  $f(w_0, -1) = f(w_0, 1)$  for  $w_0 \in \{0, 1\}$ .

#### G.4 Weak limits for scaled index processes

Define  $W \in C(\mathbb{R}_+, [0, 1])$  to be the diffusion on  $[0, 1]$  with generator

$$\mathcal{L}_W f(w) = \frac{1}{2} \frac{d^2 f}{dw^2}, \quad (121)$$

where the domain  $\mathcal{D}(\mathcal{L}_W)$  is the set of functions  $f \in C^2([0, 1])$  such that  $f'(0) = f'(1) = 0$ .  $W$  is a Brownian motion on  $[0, 1]$  with reflective boundary conditions admitting the uniform distribution  $\text{Unif}([0, 1])$  as stationary distribution.

Define  $Z \in C(\mathbb{R}_+, [0, 1] \times \{-1, 1\})$  to be the PDMP on  $[0, 1] \times \{-1, 1\}$  given by  $Z(t) = (W(t), \varepsilon(t))$  where  $W(t)$  moves in  $[0, 1]$  with velocity  $\varepsilon(t)$  and the sign of  $\varepsilon(t)$  is reversed at the arrivals times of a non-homogeneous Poisson process of rate  $\lambda(G(W(t)))G'(W(t))$  or when  $W(t)$  reaches the boundary  $\{(0, -1), (1, +1)\}$ ; see [Bierkens et al., 2018] for a discussion of PDMP on restricted domains. The infinitesimal generator of  $Z$  is given by

$$\mathcal{L}_Z f(z) = \varepsilon \frac{\partial f}{\partial w}(z) + \lambda(\beta_w) G'(w) (f(R(z)) - f(z)), \quad (122)$$

for any  $f \in \mathcal{D}(\mathcal{L}_Z)$ , the set of functions  $f \in C^1([0, 1] \times \{-1, 1\})$  such that  $f(w_0, -1) = f(w_0, 1)$  and  $\frac{\partial f}{\partial w}(w_0, -1) = -\frac{\partial f}{\partial w}(w_0, 1)$  for  $w_0 \in \{0, 1\}$ .

#### G.5 Proof of scaling limit for reversible PT

We will prove Theorem 4(a) by using Theorem 17.25 from [Kallenberg, 2002].

**Theorem 5** (Trotter, Sova, Kurtz, Mackevičius). *Let  $X, X^1, X^2, \dots$  be Feller processes defined on a state space  $S$  with generators  $\mathcal{L}, \mathcal{L}_1, \mathcal{L}_2, \dots$  respectively. If  $D$  is a core for  $\mathcal{L}$ , then the following statements are equivalent:*

1. *If  $f \in D$ , there exist  $f_N \in \mathcal{D}(\mathcal{L}_N)$  such that  $\|f_N - f\|_\infty \rightarrow 0$  and  $\|\mathcal{L}_N f_N - \mathcal{L} f\|_\infty \rightarrow 0$  as  $N \rightarrow \infty$ .*
2. *If  $X^N(0)$  converges weakly to  $X(0)$  in  $S$ , then  $X^N$  converges weakly to  $X$  in  $D(\mathbb{R}_+, S)$ .*

We will be applying Theorem 5 with  $\mathcal{L} = \mathcal{L}_W$  defined as  $\mathcal{L}_W f = \frac{1}{2} f''$  for  $f \in \mathcal{D}(\mathcal{L}_W)$  where

$$\mathcal{D}(\mathcal{L}_W) := \{f \in C^2([0, 1]) : f'(0) = f'(1) = 0\}, \quad (123)$$

and  $\mathcal{L}_N = \mathcal{L}_{W^N}$  defined in (116), which we recall here for the reader's sake

$$\mathcal{L}_{W^N} f(w) = \frac{N^2}{2} \sum_{\varepsilon \in \{\pm 1\}} (f(\Phi_\varepsilon^N(w)) - f(w)) s(\beta_w, \beta_{\Phi_\varepsilon^N(w)}), \quad w \in [0, 1], \quad (124)$$

with  $\Phi_\pm^N(w)$  defined in (113), (114) and  $\beta_w = G(w)$ . Recall from the discussion just before (116) that  $\mathcal{L}_{W^N}$  defines a Feller semigroup.

First notice that in [Kallenberg, 2002], the transition semi-group and generator of a Feller process taking values in a metric space  $S$  are defined on  $C_0(S)$ , the space of functions vanishing at infinity. Equivalently  $f \in C_0(S)$  if and only for any  $\delta > 0$  there exists a compact set  $K \subset S$  such that for  $x \notin K$ ,  $|f(x)| < \delta$ . In our case since  $S = [0, 1]$  is compact  $C_0(S) = C(S)$ , which justifies the definition of the generator  $\mathcal{L}_W$  given above.

### G.5.1 The Feller property of $\mathcal{L}_W$ .

Similarly  $\mathcal{L}_W$  can be seen to define a Feller semigroup on  $C([0, 1])$  by the Hille-Yosida theorem; see [Kallenberg, 2002, Theorem 19.11].

Indeed the first condition is satisfied since any function  $f \in C([0, 1])$  can be uniformly approximated within  $\epsilon > 0$  by a polynomial  $p_\epsilon$ , that is a smooth function, by the Stone-Weirstrass theorem. We can further uniformly approximate  $p_\epsilon$  within  $\epsilon$  by a  $C^2$  function  $\hat{p}_\epsilon$  with vanishing derivatives at the endpoints. For example one can let, for a  $\delta$  to be chosen later,  $\hat{p}_\epsilon(x) = p_\epsilon(x)$  for  $x \in (\delta, 1 - \delta)$  and for  $x \leq \delta$  set  $\hat{p}_\epsilon(x) = \int_0^x \rho_\delta(y) p'_\epsilon(y) dy + c$ , where  $\rho_\delta$  is a smooth, increasing transition function such that  $\rho_\delta(x) = 0$  for  $x < 0$ ,  $\rho_\delta(x) = 1$  for  $x > \delta$ , for example let  $\rho_\delta = \rho(x/\delta)$ ,  $\rho(x) = g(x)/(g(x) + g(1-x))$  and  $g(x) = \exp(-1/x) \mathbf{1}_{\{x>0\}}$ . We choose  $c$  so that  $\hat{p}_\epsilon(x)$  is continuous at  $\delta$ . A similar construction can be used for the right-endpoint. One can then check that indeed  $\hat{p}_\epsilon \in C^2([0, 1])$ ,  $\hat{p}'_\epsilon(0) = \hat{p}'_\epsilon(1) = 0$  and that for  $\delta$  small enough  $\|\hat{p}_\epsilon - p_\epsilon\|_\infty < \epsilon$ .

The second condition of [Kallenberg, 2002, Theorem 19.11] requires that for some  $\mu > 0$ , the set  $(\mu - \mathcal{L}_W)(\mathcal{D}(\mathcal{L}_W))$  is dense in  $C([0, 1])$ . Let  $g \in C([0, 1])$  be given. We apply [Saranen and Seikkala, 1988, Corollary 2.2], with  $f(t, y) = 2\mu y - 2g$ , which is clearly square integrable in  $t$  and

$2\mu$ -Lipschitz in  $y$ . Then [Saranen and Seikkala, 1988, Corollary 2.2] implies that for small enough  $\mu > 0$  the two-point Neumann-boundary value problem

$$\begin{aligned} \mu u - \frac{1}{2}u'' &= g \\ u'(0) = u'(1) &= 0 \end{aligned}$$

admits a solution in the Sobolev space  $H^2([0, 1])$  of functions with square integrable first and second derivatives. This already implies that  $u \in C^1([0, 1])$ , whereas the continuity of  $g$  and of  $u$  a priori implies the continuity of  $u''$  since  $u'' = 2\mu u - g$ . Overall, for any  $g \in C([0, 1])$  we can find  $u \in \mathcal{D}(\mathcal{L}_W)$  such that  $g = (\mu - \mathcal{L}_W)u$  establishing the second condition of [Kallenberg, 2002, Theorem 19.11].

The third condition of [Kallenberg, 2002, Theorem 19.11] is that  $(\mathcal{L}_W, \mathcal{D}(\mathcal{L}_W))$  satisfies the *positive maximum principle*, that is if for some  $f \in \mathcal{D}(\mathcal{L}_W)$  and  $x_0 \in [0, 1]$  we have  $f(x_0) \geq f(x) \vee 0$  for all  $x \in [0, 1]$ , then  $f''(x_0) \leq 0$ . Suppose first that the maximum is attained at an interior point  $x_0 \in (0, 1)$ ; since  $f \in C^2([0, 1])$ , by definition of  $\mathcal{D}(\mathcal{L}_W)$ ,  $f''(x_0) \geq 0$ . If on the other hand the positive maximum is attained at  $x_0 = 0$ , suppose that  $f''(0) > \epsilon$  for all  $x \leq \epsilon$ . Thus for  $0 < y < \epsilon$  small enough, since  $f'(0) = 0$  we have

$$f(y) = f(0) + \int_0^y f'(s)ds = f(0) + \int_0^y \int_0^s f''(r)dr dy \geq f(0) + \frac{\epsilon}{2}y^2 > f(0),$$

thus arriving at a contradiction.

We have thus established that  $(\mathcal{L}_W, \mathcal{D}(\mathcal{L}_W))$  satisfies all conditions of [Kallenberg, 2002, Theorem 19.11] and therefore generates a Feller process. Now we can apply Theorem 5 to prove Theorem 4(a). We only need to check the first condition of Theorem 5. In this direction, first note that by definition  $\Phi_{\pm}^N(w) = w \pm 1/N$  for  $w \in [1/N, 1 - 1/N]$ . Thus in this case using a Taylor expansion we have for  $w_-^* \in [w - 1/N, w]$  and  $w_+^* \in [w, w + 1/N]$  that for any  $f \in \mathcal{D}(\mathcal{L}_W)$ ,

$$\begin{aligned} f(\Phi_+^N(w)) - 2f(w) + f(\Phi_-^N(w)) &= f(w) + \frac{1}{N}f'(w) + \frac{1}{2N^2}f''(w_+^*) \\ &\quad + f(w) - \frac{1}{N}f'(w) + \frac{1}{2N^2}f''(w_-^*) - 2f(w) \end{aligned} \tag{125}$$

$$= \frac{1}{2N^2}(f''(w_+^*) + f''(w_-^*)). \tag{126}$$

Since  $f''$  is uniformly continuous it follows that as  $N \rightarrow \infty$ ,

$$\sup_{w \in [0, 1]} |f''(w_{\pm}^*) - f''(w)| = o(1), \tag{127}$$

and therefore for  $w \in [1/N, 1 - 1/N]$  we have

$$\sup_{w \in [0, 1]} \left| f(\Phi_+^N(w)) - 2f(w) + f(\Phi_-^N(w)) - \frac{f''(w)}{N^2} \right| = o\left(\frac{1}{N^2}\right). \tag{128}$$

When  $w \in [0, 1/N]$  or  $w \in (1 - 1/N, 1]$  we instead perform a Taylor expansion around 0 or 1 respectively. We only do the calculation in the first case, the other case being similar. Let  $w \in [0, 1/N)$  in which case, since  $f'(0) = 0$ , for  $w^*, w_-^*, w_+^* \in [0, 2/N]$

$$\begin{aligned} f(\Phi_+^N(w)) - 2f(w) + f(\Phi_-^N(w)) &= f(0) + \Phi_+^N(w)f'(0) + \frac{1}{2} [\Phi_+^N(w)]^2 f''(w_+^*) \\ &\quad + f(0) + \Phi_-^N(w)f'(0) + \frac{1}{2} [\Phi_-^N(w)]^2 f''(w_-^*) \\ &\quad - 2f(0) - 2f'(0)w - 2\frac{f''(w^*)}{2}w^2 \end{aligned} \quad (129)$$

$$= \frac{f''(0)}{2} \left\{ [\Phi_+^N(w)]^2 + [\Phi_-^N(w)]^2 - 2w^2 \right\} + o(N^{-2}), \quad (130)$$

where the error term is uniform in  $w$  and was obtained by combining the facts that  $f''$  is uniformly continuous and that  $|\Phi_\pm^N|, |w| \leq 2/N$ . Finally notice that since  $w \in [0, 1/N]$ , then

$$[\Phi_+^N(w)]^2 + [\Phi_-^N(w)]^2 - 2w^2 = \left[ w + \frac{1}{N} \right]^2 + \left[ \frac{1}{N} - w \right]^2 - 2w^2 = \frac{2}{N^2}. \quad (131)$$

Finally we will need the following weaker version of Theorem 1, whose proof is postponed to the end of this section.

**Lemma 3.** *Suppose that  $\pi_0(|V|), \pi(|V|) < \infty$ . Then there exists a constant  $C > 0$  such that*

$$\sup_{\beta} |s(\beta, \beta + \delta) - 1| \leq C\delta. \quad (132)$$

Using Lemma 3, we see that for some constant  $C > 0$

$$\sup_{w \in [0, 1]} \left| s\left(\beta_w, \beta_{\Phi_\pm^N(w)}\right) - 1 \right| \leq C \sup_w |G(w) - G(\Phi_\pm^N(w))| \leq \frac{C\|G'\|_\infty}{N}, \quad (133)$$

and therefore

$$\mathcal{L}_N f(w) = \frac{N^2}{2} \sum_{\varepsilon \in \{\pm 1\}} (f(\Phi_\varepsilon^N(w)) - f(w)) s(\beta_w, \beta_{\Phi_\varepsilon^N(w)}) \quad (134)$$

$$= \frac{N^2}{2} \sum_{\varepsilon \in \{\pm 1\}} (f(\Phi_\varepsilon^N(w)) - f(w)) [1 + o(N^{-1})] \quad (135)$$

$$= \frac{N^2}{2} (f(\Phi_+^N(w)) - 2f(w) + f(\Phi_-^N(w))) [1 + o(N^{-1})] = \frac{N^2}{2} \frac{f''(w)}{N^2} [1 + o(1)], \quad (136)$$

where the error term is uniform in  $w$ . Thus  $\mathcal{L}_N f \rightarrow \mathcal{L} f$  uniformly.

of Lemma 3. Using the bound  $0 \leq 1 - \exp(-x) \leq x$  for  $x \geq 0$  we have

$$\begin{aligned} & |s(\beta, \beta') - 1| \\ & \leq \iint \pi^{(\beta)}(dx) \pi^{(\beta')}(dy) \left[ 1 - \exp(-\max\{0, (\beta' - \beta)[V(x) - V(y)]\}) \right] \end{aligned} \quad (137)$$

$$\leq |\beta' - \beta| \iint \pi^{(\beta)}(dx) \pi^{(\beta')}(dy) \max\{0, [V(x) - V(y)]\} \quad (138)$$

$$\leq |\beta' - \beta| \iint \pi^{(\beta)}(dx) \pi^{(\beta')}(dy) (|V(x)| + |V(y)|) \quad (139)$$

$$\leq 2|\beta' - \beta| \sup_{\beta} \pi^{(\beta)}(|V|). \quad (140)$$

□

## G.6 Proof of scaling limit for non-reversible PT

We will prove Theorem 4(b) in a slightly round about way. We will define the auxiliary processes  $\{U^N(\cdot)\}$ ,  $\{U(\cdot)\}$  living on the unit circle  $\mathbb{S}^1 := \{z \in \mathbb{C} : |z| = 1\}$  along with a mapping  $\phi : \mathbb{S}^1 \mapsto [0, 1] \times \{\pm 1\}$  such that  $Z^N = \phi(U^N)$  and  $Z = \phi(U)$ . We will first show that the law of  $U^N$  converges weakly to  $U$ .

Before defining the processes we point out that we will identify  $\mathbb{S}^1$  with  $[0, 2\pi]$  in the usual way by working in mod  $2\pi$  arithmetic. Notice that in this way

$$C(\mathbb{S}^1) = \{f \in C([0, 2\pi]) : f(0) = f(2\pi)\}. \quad (141)$$

The reason for working with these auxiliary processes is that we can now avoid working with PDMPs with boundaries, helping us to remove a layer of technicalities.

For any  $N$  we define  $\Sigma^N : \mathbb{S}^1 \mapsto \mathbb{S}^1$  through  $\Sigma^N(\theta) = \theta + 2\pi/N$ . Consider then a continuous-time process  $U^N$  that jumps at the arrival times of a homogeneous Poisson process with rate  $N$  according to the kernel

$$Q^N(\theta, d\theta') = s \left( \tilde{\beta}_\theta, \tilde{\beta}_{\Sigma^N(\theta)} \right) \delta_{\Sigma^N(\theta)}(d\theta') + \left[ 1 - s \left( \tilde{\beta}_\theta, \tilde{\beta}_{\Sigma^N(\theta)} \right) \right] \delta_{2\pi-\theta}(d\theta'), \quad (142)$$

where

$$\tilde{\beta}_\theta = \begin{cases} G\left(\frac{\theta}{\pi}\right), & \theta \in [0, \pi), \\ G\left(\frac{2\pi-\theta}{\pi}\right), & \theta \in [\pi, 2\pi]. \end{cases} \quad (143)$$

Define the map

$$\phi(\theta) = \begin{cases} \left(\frac{\theta}{\pi}, +1\right), & \theta \in [0, \pi), \\ \left(\frac{2\pi-\theta}{\pi}, -1\right), & \theta \in [\pi, 2\pi]. \end{cases} \quad (144)$$

Essentially we think of the circle as comprising of two copies of  $[0, 1]$  glued together at the end points. The top one is traversed in an increasing direction and the bottom one in a decreasing

direction. When glued together and viewed as a circle these dynamics translate in a counter-clockwise rotation with occasional reflections with respect to the  $x$ -axis at the time of events. With this picture in mind it should be clear that  $\phi(U^N) = Z^N$ .

We also define the limiting process  $U$  as follows. First let

$$\tilde{\lambda}(\theta) = (\lambda \circ G)(\phi^1(\theta))G'(\phi^1(\theta)), \quad (145)$$

where  $\phi^1(\theta)$  is the first coordinate of  $\phi(\theta)$ . Notice at this point that  $\phi^1 : \mathbb{S}^1 \mapsto [0, 1]$  is continuous and satisfies  $\phi^1(\theta) = \phi^1(-\theta)$  for any  $\theta \in [0, 2\pi)$ , whence we obtain that  $\tilde{\lambda}(-\theta) = \tilde{\lambda}(\theta)$ . Given  $U(0) = \theta$ , let  $T_1$  be a random variable such that

$$\mathbb{P}[T_1 \geq t] = \exp \left\{ - \int_0^t \tilde{\lambda}(\theta + s) ds \right\}, \quad (146)$$

and define the process as  $U(s) = \theta + s \bmod 2\pi$  for all  $s < T_1$  and set  $U(T_1) = -U(T_1-) \bmod 2\pi$ . Iterating this procedure will define the  $\mathbb{S}^1$ -valued PDMP  $\{U(\cdot)\}$ . We first need the next lemma.

**Lemma 4.** *Suppose  $V$  is integrable with respect to  $\pi_0$  and  $\pi$ . The process  $U$  defined above is a Feller process, its infinitesimal generator is given by*

$$\mathcal{L}_U f(\theta) = f'(\theta) + \tilde{\lambda}(\theta) [f(2\pi - \theta) - f(\theta)], \quad (147)$$

with domain

$$\mathcal{D}(\mathcal{L}_U) = \{f \in C^1([0, \pi]) : f(0) = f(2\pi)\}, \quad (148)$$

and invariant measure  $d\theta/2\pi$ .

*Proof.* First, note that since  $\mathbb{S}^1$  is compact  $C_0(\mathbb{S}^1) = C(\mathbb{S}^1)$  and thus to study the Feller process we consider the semi-group  $\{P_U^t\}_t$  defined by the process  $U$  as acting on  $C(\mathbb{S}^1)$ . To prove the Feller property we can thus use [Davis, 1993, Theorem 27.6]. Since there is no boundary in the definition of  $U$  the first assumption is automatically verified,  $Qf(\theta) = f(-\theta) \in C(\mathbb{S}^1)$  for any continuous  $f$ . We also know that the rate  $\tilde{\lambda}$  is bounded whereas by Proposition 2 and the fact that  $G \in C^1[0, 1]$  we know that  $\tilde{\lambda}$  is also continuous. Therefore the third condition of [Davis, 1993, Theorem 27.6] holds and thus  $U$  is Feller.

The infinitesimal generator will be defined on  $\mathcal{D}(\mathcal{L}_U) \subseteq C(\mathbb{S}^1)$ . The domain is defined as the class of functions  $f \in C(\mathbb{S}^1)$  such that

$$g(\theta) = \lim_{h \rightarrow 0} \frac{1}{h} [P_U^t f(\theta) - f(\theta)] \in C(\mathbb{S}^1), \quad (149)$$

where the limit is uniform in  $\theta$ . However by [Böttcher et al., 2013, Theorem 1.33], we can also consider pointwise limits without enlarging the domain. Using the definition of  $U$  we then have for

$\theta \in [0, 2\pi)$  that

$$\frac{1}{h} \mathbb{E}^\theta [f(U_h) - f(\theta)] = \frac{1}{h} \left[ f(\theta + h) \mathbb{P}^\theta [T_1 \geq h] - f(\theta) \right] + \frac{1}{h} \mathbb{E}^\theta [f(U_h) \mathbf{1}\{T_1 \leq h\}]. \quad (150)$$

Since for  $x \geq 0$  we have  $|\exp(-x) - 1 + x| \leq Cx^2$  for some constant  $C > 0$ , and using the uniform continuity of  $\tilde{\lambda}$  we can see that

$$\left| \exp \left\{ - \int_0^h \tilde{\lambda}(\theta + s) ds \right\} - 1 + \tilde{\lambda}(\theta)h \right| = o(h), \quad (151)$$

uniformly in  $\theta$ , and thus

$$\frac{1}{h} f(\theta + h) \mathbb{P}^\theta [T_1 \geq h] - f(\theta) = \frac{1}{h} \left[ f(\theta + h) \left[ 1 - \tilde{\lambda}(\theta)h + o(h) \right] - f(\theta) \right] \quad (152)$$

$$= \frac{1}{h} [f(\theta + h) - f(\theta)] - \tilde{\lambda}(\theta)f(\theta) + o(1). \quad (153)$$

In addition

$$\begin{aligned} & \frac{1}{h} \mathbb{E}^\theta [f(U_h) \mathbf{1}\{T_1 \leq h\}] \\ &= \frac{1}{h} \int_0^h \tilde{\lambda}(\theta + s) \exp \left\{ - \int_0^s \tilde{\lambda}(\theta + r) dr \right\} ds P_U^{h-s} Q f(\theta) \end{aligned} \quad (154)$$

$$\rightarrow \tilde{\lambda}(\theta) Q f(\theta), \quad (155)$$

for any  $f \in C(\mathbb{S}^1)$  by strong continuity of  $\{P_U^t\}$  (Feller property) and continuity of  $\tilde{\lambda}$ . Overall we thus have that  $f \in \mathcal{D}(\mathcal{L}_U)$  if and only if

$$\frac{1}{h} \mathbb{E}^\theta [f(U_h) - f(\theta)] = \frac{f(\theta + h) - f(\theta)}{h} + \tilde{\lambda}(\theta) [Q f(\theta) - f(\theta)] + o(1) \quad (156)$$

$$\rightarrow g(\theta) \in C(\mathbb{S}^1), \quad (157)$$

which is clearly equivalent to  $f \in C^1(\mathbb{S}^1)$ .

Finally to see that  $d\theta/2\pi$  is invariant, having identified the domain we can easily check that for any  $f \in C(\mathbb{S}^1)$  we have

$$\int d\theta P_U^t f(\theta) = \int_{s=0}^t \int d\theta \mathcal{L}_U P_U^s f(\theta) d\theta ds. \quad (158)$$

Since  $f \in \mathcal{D}(\mathcal{L}_U)$  we have that  $P_U^s g \in \mathcal{D}(\mathcal{L}_U)$ . Since for any  $g \in \mathcal{D}(\mathcal{L}_U)$  we have

$$\int d\theta \mathcal{L}_U f(\theta) = \int_{\theta=0}^{2\pi} f'(\theta) d\theta + \int_{\theta=0}^{2\pi} \tilde{\lambda}(\theta) f(Q(\theta)) d\theta - \int_{\theta=0}^{2\pi} \tilde{\lambda}(\theta) f(\theta) d\theta \quad (159)$$

$$= f(2\pi) - f(0) + \int_{\theta=0}^{2\pi} \tilde{\lambda}(\theta) f(Q(\theta)) d\theta. \quad (160)$$

□

**Proposition 5.** Suppose  $U^N(0)$  converges weakly to  $U(0)$ , then  $U^N$  converges weakly to  $U$  in  $D(\mathbb{R}_+, [0, 1])$ .

*Proof.* We will once again use Theorem 5. The generator of  $U_N$  is given by

$$\mathcal{L}_U^N f(\theta) = N [f(\theta + 1/N) - f(\theta)] s(\tilde{\beta}_\theta, \tilde{\beta}_{\Sigma^N(\theta)}) + N [f(-\theta) - f(\theta)] r(\tilde{\beta}_\theta, \tilde{\beta}_{\Sigma^N(\theta)}). \quad (161)$$

We will consider the two terms separately. To this end notice that by (23), the boundedness of  $\lambda$  and the fact that  $G \in C^1[0, 1]$

$$\left| 1 - s(\tilde{\beta}_\theta, \tilde{\beta}_{\Sigma^N(\theta)}) \right| \leq \frac{C}{N}, \quad (162)$$

for some  $C > 0$ . Thus, using the mean value theorem, for each  $\theta \in [0, 2\pi]$ , there exists  $g_N(\theta) \in [\theta, \theta + 1/N]$  such that

$$N [f(\theta + 1/N) - f(\theta)] s(\tilde{\beta}_\theta, \tilde{\beta}_{\Sigma^N(\theta)}) = f'(g_N(\theta)) (1 + O(1/N)) = f'(\theta) ((1 + o(1)), \quad (163)$$

where the errors are uniformly bounded and to obtain the second equality above we have used the fact that  $|g_N(\theta) - \theta| \leq 1/N$  and that  $f'$  is uniformly continuous, being continuous on a compact set.

Overall we can see that as  $N \rightarrow \infty$

$$\sup_\theta \left| N [f(\theta + 1/N) - f(\theta)] s(\tilde{\beta}_\theta, \tilde{\beta}_{\Sigma^N(\theta)}) - f'(\theta) \right| \rightarrow 0. \quad (164)$$

Next, using (23) we have that

$$r(\tilde{\beta}_\theta, \tilde{\beta}_{\Sigma^N(\theta)}) = \tilde{\lambda}(\theta) \frac{1}{N} + o(N^{-1}), \quad (165)$$

where the error is uniform in  $\theta$ , whence we easily conclude that

$$N [f(-\theta) - f(\theta)] r(\tilde{\beta}_\theta, \tilde{\beta}_{\Sigma^N(\theta)}) \rightarrow \tilde{\lambda}(\theta) [Qf(\theta) - f(\theta)], \quad (166)$$

uniformly in  $\theta$ . □

### G.6.1 Proof of Theorem 4(b)

Now we are ready to prove the main result of this section. Notice that  $Z^N(\cdot) = \phi(U^N(\cdot))$  and  $Z(\cdot) = \phi(U(\cdot))$ .

From Proposition 5 we know that the finite dimensional distributions of  $U_N$  converge to those of  $U$ . If  $\phi$  were continuous we could conclude using the continuous mapping theorem. Since it is not continuous at the points  $\{0, \pi\}$ , we will be using [Billingsley, 2013, Theorem 2.7]. We have to check that the law of the limiting process, that is the law of  $\{U(\cdot)\}$  places zero mass on finite dimensional distributions that hit  $\{0, \pi\}$ , that is for  $n \in \mathbb{N}$  and  $0 < t_1 < \dots < t_n$  we want

$$\mathbb{P}[U(t_i) \in \{0, \pi\} \text{ for some } i \in \{1, \dots, n\}] = 0, \quad (167)$$

when  $U(0)$  is initialized according to  $d\theta/2\pi$ . But the above follows from the fact that  $\mathbb{P}[U(t_i) \in \{0, \pi\}] = 0$ , by stationarity when  $U(0)$  is initialised uniformly on  $\mathbb{S}^1$ .

Relative compactness of  $\{Z_N(\cdot)\}_N$  can be easily seen to follow from the compact containment condition [Ethier and Kurtz, 2009, Remark 3.7.3]. This combined with convergence of the finite dimensional distributions of  $Z^N$  to those of  $Z$  concludes the proof.

## Appendix H Distributed implementation of PT

For simplicity, in Section 2 we have shown pseudo-code for PT with a parallel computing context in mind rather than a distributed computing context. The key difference is that in the latter case, it is advantageous to swap annealing parameters instead of states to minimize network communication. For completeness, we show the pseudo-code for the distributed implementation in this section.

## Appendix I Experiment supplements

### I.1 Reproducibility.

To make our NRPT method easy to use we implemented it as an inference engine in the open source probabilistic programming language (PPL) Blang <https://github.com/UBC-Stat-ML/blangSDK>. A full description of the models used in the paper are available at <https://github.com/UBC-Stat-ML/blangDemos>, see in particular <https://github.com/UBC-Stat-ML/blangDemos/blob/master/src/main/resources/demos/models.csv> for a list of command line options and data paths used for each model. All methods use the same local exploration kernels, namely slice sampling with exponential doubling followed by shrinking [Neal, 2003]. Scripts documenting replication of our experiments are available at <https://github.com/UBC-Stat-ML/ptbenchmark>.

### I.2 Multi-core implementation.

We use lightweight threads [Friesen, 2015] to parallelize both the local exploration and communication phases, as shown in Algorithm 1. We use the algorithm of [Leiserson et al., 2012] as

---

**Algorithm 5** Distributed DEO

---

```

1:  $\mathbf{x} \leftarrow \mathbf{x}_0$                                       $\triangleright$  Initialize chain
2:  $\mathbf{i} \leftarrow (0, 1, 2, \dots, N)$        $\triangleright$  For a machine  $j$ ,  $\mathbf{i}^j$  gives the annealing parameter index for that
   machine. Initialized with the identity map.
3:  $\mathbf{j} \leftarrow (0, 1, 2, \dots, N)$   $\triangleright$  For an annealing parameter index  $i$ ,  $\mathbf{j}^i$  gives the machine processing the
   corresponding chain. This is the inverse of the above mapping, hence it is also initialized with
   the identity map.
4: for  $n$  in  $1, 2, \dots, n_{\text{scan}}$  do
5:   if  $n$  is even then                                 $\triangleright$  Non-reversibility inducing alternation
6:      $P \leftarrow \{i : 0 \leq i < N, i \text{ is even}\}$            $\triangleright$  Even subset of  $\{0, \dots, N - 1\}$ 
7:   else
8:      $P \leftarrow \{i : 0 \leq i < N, i \text{ is odd}\}$             $\triangleright$  Odd subset of  $\{0, \dots, N - 1\}$ 
9:   for  $j$  in  $0, \dots, N$  do                          $\triangleright$  Distributed local exploration phase
10:     $\beta \leftarrow \beta_{\mathbf{i}^j}$                             $\triangleright$  Fetch current annealing parameter for machine  $j$ 
11:     $x^j \sim K^{(\beta)}(x_{n-1}^j, \cdot)$ 
12:   for  $i$  in  $0, \dots, N - 1$  do                   $\triangleright$  Distributed communication phase
13:      $\alpha \leftarrow \alpha^{(i, i+1)}$                        $\triangleright$  Equation (6).
14:      $A \sim \text{Bern}(\alpha)$ 
15:     if  $i \in P$  and  $A = 1$  then
16:        $(\mathbf{j}^i, \mathbf{j}^{i+1}) \leftarrow (\mathbf{j}^{i+1}, \mathbf{j}^i)$ 
17:     for  $i$  in  $0, \dots, N$  do                 $\triangleright$  Recompute  $\mathbf{i}$  from  $\mathbf{j}$  such that  $\mathbf{j}^i = j \iff \mathbf{i}^j = i$ 
18:        $j \leftarrow \mathbf{j}^i$ 
19:        $\mathbf{i}^j \leftarrow i$ 
20:    $\mathbf{x}_n \leftarrow \mathbf{x}$ 

```

---

implemented in [Steele and Lea, 2013] to allow each PT chain to have its own random stream. This technique avoids any blocking across threads and hence makes the inner loop of our algorithm embarrassingly parallel in  $N$ . Moreover, the method of [Leiserson et al., 2012] combined with the fact that we fix random seeds means that the numerical value output by the algorithm is not affected by the number of threads used. Increasing the number of threads simply makes the algorithm run faster. In all experiments unless noted otherwise we use the maximum number of threads available in the host machine, by default an Intel i5 2.7 GHz (which supports 8 threads via hyper-threading) except for Section 7.3 where we use an Amazon EC2 instance of type c4.8xlarge, which is backed by a 2.9 GHz Intel Xeon E5-2666 v3 Processor (20 threads).

### I.3 Stochastic optimization methods.

All baseline methods are implemented in Blang (<https://github.com/UBC-Stat-ML/blangSDK>), the same probabilistic programming language used to implement our method. The code for the baseline adaption methods are available at <https://github.com/UBC-Stat-ML/blangDemos>. All methods therefore run on the Java Virtual Machine, so their wall clock running times are all comparable.

Both [Atchadé et al., 2011] and [Miasojedow et al., 2013] are based on reversible PT together

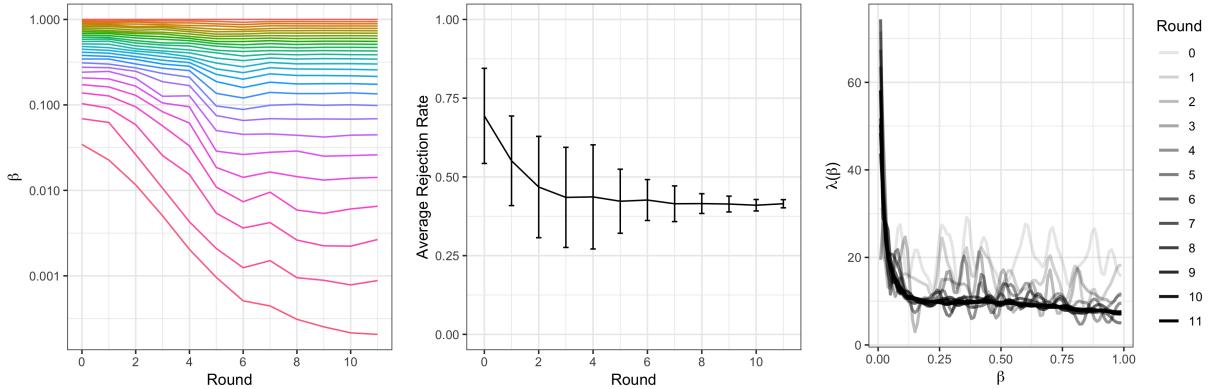


Figure 14: A demonstration of the tuning phase in Algorithm 4 ran on a hierarchical Bayesian model applied to the historical failure rates of 5 667 launches for 367 types of rockets (a 369 dimensional problem, see Appendix I.4 for details). We use  $\bar{N} = 30$  cores and 11 schedule optimization rounds, the last one consisting of  $n_{\text{tune}} = 5\,000$  and  $\hat{\Lambda} = 12.03$ . (Left) Progression of the annealing schedule (colours index parallel chains, y-axis, the values  $\beta_k$  for each schedule optimization round, in log scale). (Centre) Progression of the sample mean and standard deviation of empirical rejection probabilities  $\{\hat{r}^{(i-1,i)}\}_{i=1}^N$ . The mean stabilizes quickly, and as the schedule optimization rounds increase, the rejection probabilities converge to the mean as desired. (Right) Progression of the estimated  $\hat{\lambda}(\beta)$  as a function of the schedule optimization round.

with two different flavours of stochastic optimization to adaptively select the annealing schedule. In [Atchadé et al., 2011], the chains are added one by one, each chain targeting a swap acceptance rate of 23% from the previous one. In [Miasojedow et al., 2013], this scheme is modified in two ways: first, all annealing parameters are optimized simultaneously, and second, a different update for performing the stochastic optimization is proposed. To optimize all chains simultaneously, the authors assume that both the number of chains and the equi-acceptance probability are specified. Since this information is not provided to the other methods, in order to perform a fair comparison, for the method we label as “Miasojedow, Moulines, Vihola” we implemented a method which adds the chain one at the time while targeting the swap acceptance rate of 23% but based on the improved stochastic optimization update of [Miasojedow et al., 2013]. Specifically, both [Atchadé et al., 2011] and [Miasojedow et al., 2013] rely on updates of the form  $\rho_{n+1} = \rho_n + \gamma_n(\alpha_{n+1} - 0.23)$  where  $\gamma_n$  is an update schedule and  $\rho_n$  is a re-parameterization of difference in annealing parameter from the previous chain  $\beta$  to the one being added  $\beta'$ . The work of [Atchadé et al., 2011] uses the update  $\beta'_n = \beta(1 + \exp(\rho_n))^{-1}$ , whereas the work of [Miasojedow et al., 2013] specifies the explicit parameterization used for  $\rho$ , namely  $\rho = \log(\beta'^{-1} - \beta^{-1})$ , from which the update becomes  $\beta'_n = \beta(1 + \beta \exp(\rho_n))^{-1}$ . Moreover, while [Atchadé et al., 2011] use  $\gamma_n = (n + 1)^{-1}$ , [Miasojedow et al., 2013] suggest to use  $\gamma_n = (n + 1)^{-0.6}$ . The experiments confirm that the latter algorithm is more stable.

#### I.4 Description of models and datasets

In Section 7.3 we benchmark the methods on the following four models. First, a hierarchical model applied to a dataset of rocket launch failure/success indicator variables [McDowell, 2019].

We organized the data by types of launcher, obtaining 5,667 launches for 367 types of rockets (processed data available at [https://github.com/UBC-Stat-ML/blangDemos/blob/master/data/failure\\_counts.csv](https://github.com/UBC-Stat-ML/blangDemos/blob/master/data/failure_counts.csv)). Each type is associated with a Beta-distributed parameter with parameters tied across rocket types, with the likelihood given by a Binomial distribution (full model specification available at <https://github.com/UBC-Stat-ML/blangDemos/blob/master/src/main/java/hier/HierarchicalRockets.bl>). The second model is a Spike-and-Slab variable selection model applied to the RMS Titanic Passenger Manifest dataset [Hind, 2019]. The preprocessed data is available at <https://github.com/UBC-Stat-ML/blangDemos/tree/master/data/titanic>. The data consist in binary classification indicators for the survival of each individual passenger as well as covariates such as age, fare paid, etc. We used a Spike-and-Slab prior with a point mass at zero and a Student-t continuous component (full model specification available at <https://github.com/UBC-Stat-ML/blangDemos/blob/master/src/main/java/glms/SpikeSlabClassification.bl>). Third, we used the Ising model from Section I.4.2. Finally, we also used an end-point conditioned Wright-Fisher stochastic differential equation (see, e.g., [Tataru et al., 2017]). For this last model we used synthetic data generated by the model. The specification of this last model is available at <https://github.com/UBC-Stat-ML/blangSDK/blob/master/src/main/java/blang/validation/internals/fixtures/Diffusion.bl>.

The model-specific command line options used for all four experiments is available at <https://github.com/UBC-Stat-ML/blangDemos/blob/master/src/main/resources/demos/models.csv>. The same file also documents the location of the probabilistic programming source code and precise command line arguments for the 16 models summarized in Figure 7.

#### I.4.1 Gaussian model

Suppose  $\pi \sim N(0, \sigma^2 \mathbb{I}_d)$ , and  $\pi_0 \sim N(0, \sigma_0^2 \mathbb{I}_d)$  with  $\sigma_0 > \sigma$ . It can be shown that  $\pi^{(\beta)} \sim N(0, \sigma_\beta^2 \mathbb{I}_d)$  where  $\sigma_\beta^{-2} = (1 - \beta)\sigma_0^{-2} + \beta\sigma^{-2}$ . Theorem 1 in [Predescu et al., 2004] implies the following closed form expressions for  $\lambda(\beta)$  and  $\Lambda(\beta)$

$$\lambda(\beta) = \frac{2^{1-d}(\sigma^{-2} - \sigma_0^{-2})}{B\left(\frac{d}{2}, \frac{d}{2}\right)} \sigma_\beta^2, \quad \Lambda(\beta) = \frac{2^{2-d}}{B\left(\frac{d}{2}, \frac{d}{2}\right)} \log\left(\frac{\sigma_0}{\sigma_\beta}\right), \quad (168)$$

where  $B(a, b)$  is the Beta function. As  $d \rightarrow \infty$ , we have  $\Lambda \sim \sqrt{\frac{2d}{\pi}} \log\left(\frac{\sigma_0}{\sigma}\right)$ , which is consistent with Proposition 4. We see from Figure 16 that the empirical approximation of  $\hat{\lambda}, \hat{\Lambda}$  from Algorithm 4 are consistent with the theoretical values from in (168).

Substituting (168) into (30) we have  $\beta_k^*$  satisfies

$$\sigma_{\beta_k^*} = \sigma^{\frac{k}{N}} \sigma_0^{1 - \frac{k}{N}}. \quad (169)$$

This is the same spacing obtained (based on a different theoretical approach) in [Atchadé et al., 2011] and [Predescu et al., 2004] for the Gaussian model. This combined with Proposition 3 also justifies why the geometric schedule works for Gaussian mixture models with well-separated

modes. See Figure 15 for realizations of the index process for the reversible and non-reversible PT algorithms.

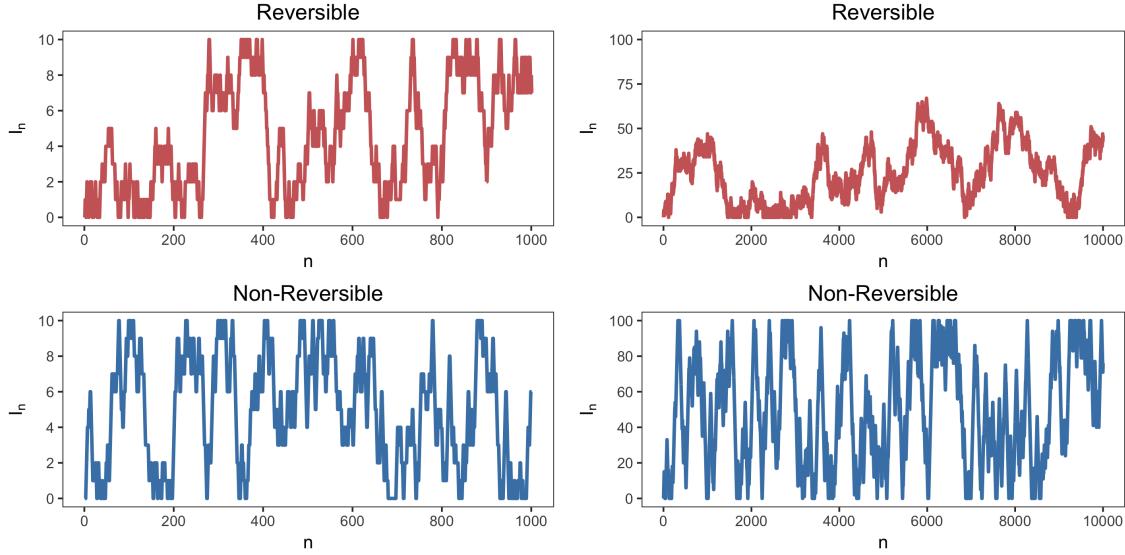


Figure 15: Sample trajectories of the component  $I_n^0$  of the index process for a Gaussian model with  $\Lambda = 5$  with schedule  $\mathcal{P}_N$  satisfying (169) for  $N = 10$  (left) and  $N = 100$  (right). The trajectories are run over  $100N$  iterations for reversible (top) and non-reversible (bottom) PT.

#### I.4.2 Ising model

We now compute numerically  $\lambda$  for the Ising model on a 2-dimensional lattice of size  $M \times M$  with magnetic moment  $\mu$ . Note that the terminology ‘‘Ising’’ is technically incorrect since we are using a finite  $M$  instead of an infinite lattice. The terminology ‘‘autologistic’’ would be more appropriate but we abuse the terminology ‘‘Ising’’ since it is widespread in the computational statistics literature. Similarly, when we talk about ‘‘phase transition’’ this should be interpreted as an approximate phase transition since  $M$  is finite.

Using the notation  $x_i \sim x_j$  to indicate sites are nearest neighbours on the lattice, the target distribution is annealed by the inverse temperature  $\beta$  and the tempered distributions are given by

$$\pi^{(\beta)}(x) = \frac{1}{Z(\beta)} \exp \left( \beta \sum_{x_i \sim x_j} x_i x_j + \mu \sum_i x_i \right). \quad (170)$$

This is an  $M^2$  dimensional model which undergoes a phase transition as  $M \rightarrow \infty$  at some critical inverse-temperature  $\beta_c$ . When  $\mu = 0$  it is known that  $\beta_c = \log(1 + \sqrt{2})/2$  [Baxter, 2007]. We consider experiments with  $\mu = 0$  and with  $\mu = 0.1$ , the latter denoted ‘‘Ising with magnetic field’’ and abbreviated ‘‘magnetic’’ in composite figures.

We observe that  $\lambda$  exhibits very different characteristics in this scenario compared to the Gaussian model: it is not monotonic and is maximized at the critical temperature. Consequently, the

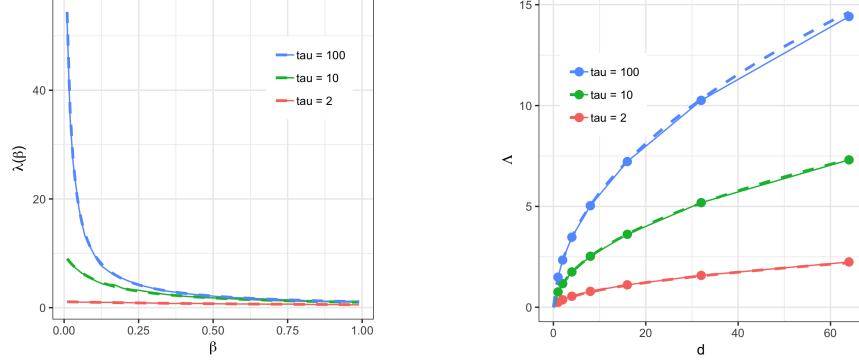


Figure 16: (Left) The local communication barrier for  $d = 8, \sigma = 1/2, 1/10, 1/100$  and  $\sigma_0 = 1$ . (Right) The global communication barrier as a function of  $d$  for  $\tau = 2, 10, 100$  and  $\sigma_0 = 1$ . The solid line is the approximation  $\hat{\lambda}(\beta)$  (respectively  $\hat{\Lambda}$ ), resulting from Algorithm 4 ( $N = 60$  and  $n = 10\,000$  scans) and the dotted line is the analytic expression from (168).

optimal annealing schedule is denser near the phase transition. We also note from Figure 17 that both  $\lambda$  and  $\Lambda$  increases roughly linearly with respect to  $M$ , similarly to the conclusion of Proposition 4, however here the conditions of Proposition 4 do not apply. As  $N$  increases, the round trip rate of reversible PT decays to 0 and non-reversible PT increase towards  $(2 + 2\Lambda)^{-1}$  as seen in Figure 18. This is consistent with Theorem 3.

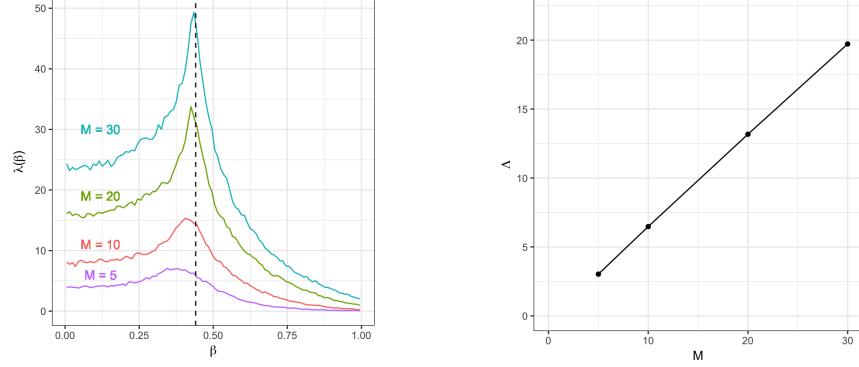


Figure 17: Estimate of the local communication barrier (left) and global communication barrier (right) for the Ising model with  $\mu = 0$  and  $M = 5, 10, 20, 30$ . The vertical line is at the phase transition.

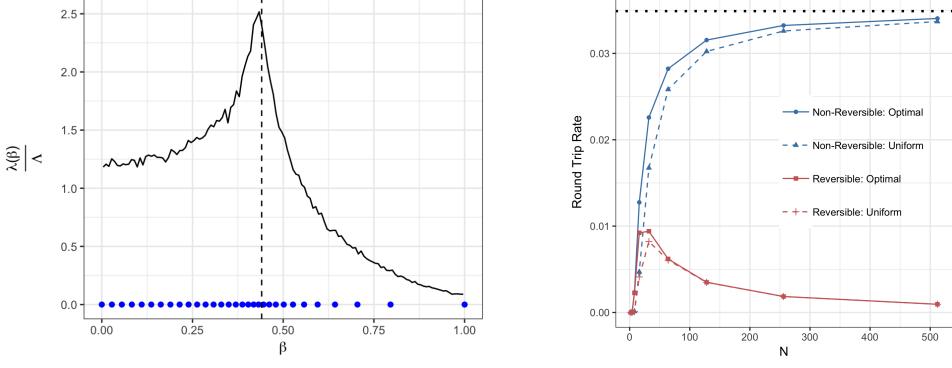


Figure 18: (Left) Optimal annealing schedule for the Ising model with  $M = 20$ ,  $\Lambda = 13.33$  with  $N = 30$ . The vertical line is at the phase transition. (Right) The round trip rates when  $M = 20$  with a uniform schedule (dashed) to the optimal schedule (solid) for both non-reversible (blue) and reversible (red) PT. The dotted horizontal line represents the approximation of the optimal round trip rate  $(2 + 2\hat{\Lambda})^{-1}$ .

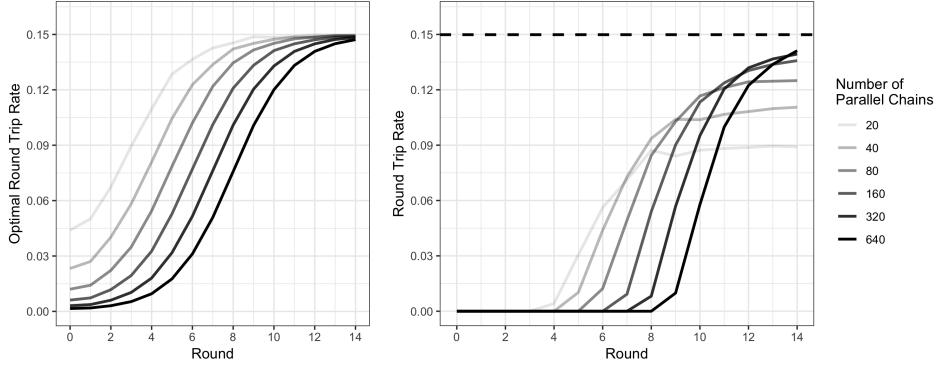


Figure 19: Further results on robustness to ELE violation on the Ising model with a magnetic moment,  $\mu = 0.1$ . Impact of increasing the number  $N$  of parallel chains for an example where ELE is severely violated (only half the variables are updated at each exploration step). (Left) Estimated upper bound  $\bar{\tau}$ . This quantity is well estimated for any  $N$ . (Right) Round trip rate  $\tau$  measured from the empirical index process trajectories. Dotted line shows  $\bar{\tau}$ . The result shows  $\tau$  approaches  $\bar{\tau}$  by increasing  $N$ . This experiment suggests that the result in Theorem 3(c) may hold under more general conditions. Moreover since  $\bar{\tau}$  is well estimated even when the number of chains is modest, the gap between  $\bar{\tau}$  and observed  $\tau$  can be readily estimated.

#### I.4.3 Discrete-multimodal problem

Consider a discrete state space  $\mathcal{X} = \{0, \dots, 2k\}$ , and let  $1_{\text{Even}} : \Omega \rightarrow \{0, 1\}$  denote the indicator function for even numbers. Define  $\pi(x) \propto a^{1_{\text{Even}}(x)}$  for  $a > 1$  and  $\pi_0(x) \propto 1$  with  $V(x) = -1_{\text{Even}}(x) \log a$ . The distribution  $\pi$  has  $k+1$  modes located where  $x$  is even with low probability ‘‘barriers’’ located at  $x$  odd. The parameter  $a$  controls the relative mass put on the modes. Therefore we have

$$\pi^{(\beta)}(x) = \frac{a^{\beta 1_{\text{Even}}(x)}}{Z(\beta)}, \quad (171)$$

where  $Z(\beta) = k + (k+1)a^\beta$ . A simple computation using (22) shows that the local communication barrier is,

$$\lambda(\beta) = \frac{k(k+1)a^\beta \log a}{(k+(k+1)a^\beta)^2} \xrightarrow{k \rightarrow \infty} \frac{a^\beta \log a}{(1+a^\beta)^2}. \quad (172)$$

By integrating we obtain the global communication barrier between  $\pi$  and  $\pi_0$ ,

$$\Lambda = \frac{k(k+1)(a-1)}{(2k+1)(k+(k+1)a)} \xrightarrow{k \rightarrow \infty} \frac{a-1}{2(a+1)}. \quad (173)$$

#### I.4.4 Mixture model

We consider a Bayesian mixture model with two mixture components. The likelihood for each component is a normal distribution with a non-conjugate Uniform(0, 100) prior on the standard deviation and a normal prior on the means (standard deviation of 100). We placed a uniform prior on the mixture proportion. We used simulated data generated from the model. While the mixture membership indicator latent random variables can be marginalized in this model, we sample them to make the posterior inference problem more challenging. Sampling these mixture membership random variables is representative of more complex models from the Bayesian non-parametric literature where marginalization of the latent variable is intractable; for example, this is the case for the stick-breaking representation of general completely random measures [Zhu et al., 2020].

In addition to the three MCMC methods described in the main text, we also ran baselines based on SMC and AIS, which are popular methods to explore complex posterior distributions. These methods also depend on the construction of a sequence of annealed distributions from prior to posterior. For the SMC and AIS baselines, to select the sequence of distributions we used an adaptive scheme based on relative ESS as described in [Zhou et al., 2016]. Diagnostics of the adaptation are shown in Figure 26. These methods were parallelized at the particle level. We set the number of particles to achieve a similar running time compared to NRPT, namely 2 000 particles for SMC and 2 500 particles for AIS. We found that the quality of the posterior approximation was highly dependent on performing several rounds of rejuvenations on the final particle population. The wall-clock time with 5 and 20 rounds of rejuvenation (SMC-5, SMC-20) was comparable to NRPT (1.778min, 2.302min) however the posterior approximation is markedly poor compared to NRPT. With 100 rounds of rejuvenation, the posterior approximation matches closely that of NRPT, however this brings the computation cost to 5.064min. AIS did not perform well since the weights were highly unbalanced in the last iteration, effectively resulting in an approximation putting all mass to a single particle.

#### I.4.5 Copy number inference

A copy number alteration is a widespread type of mutation in cancers. Whereas in healthy cells each non-sexual chromosomes comes in two copies, in certain cancer cells, chunks of chromosomes of small and large size are frequently deleted or duplicated [Beroukhim et al., 2010]—one extreme type of such event is a whole-genome duplication, in which the copy number of every genomic position

is doubled. We consider the task of inferring the copy number for each location in the genome based on single-cell, whole genome data. For simplicity, we consider here a model for performing inference over the copy number profile of one individual cell at a time.

The data consists in the number of reads  $y_{i,c}$  measured at different loci. Here  $i$  denotes a sub-region or bin of a chromosome  $c$ . We use the same binning method as [Lai et al., 2020]. The goal is to infer for each bin  $i$  and chromosome  $c$  an integer copy number  $x_{i,c}$ . As previously reported in [Zahn et al., 2017], and confirmed in Figure 27, to a first order approximation the location parameter of the variable  $y_{i,c}$  is determined by (1) the known GC contents of each bin, denoted  $g_{i,c}$  and (2) the unknown copy number  $x_{i,c}$ . Here instead of performing GC-normalization as a pre-processing step, as done in [Zahn et al., 2017], we perform GC-normalization jointly with copy number inference. We describe in this section how doing so allows us to capture the uncertainty over plausible whole-genome duplication events, and induces multimodal posterior distributions.

The single cell protocol used for the dataset considered here [Zahn et al., 2017] minimizes the PCR amplification biases so that after correcting the GC-bias, the count in a bin is approximately proportional to the copy number of the bin:  $\log y_{i,c} = f(\theta, \log g_{i,c}) + \log(x_{i,c}) + \epsilon_{i,c}$ , for  $x_{i,c} > 0$ . Based on Figure 27 in Appendix J.4, we approximate  $f$  using a latent quadratic function, and hence  $\theta$  is a global parameter containing the 3 coefficients specifying a quadratic polynomial. We assume  $\epsilon_{i,c}$  are independent zero-mean normal random variables with standard deviation  $\sigma$ . To model the prior information that copy number events affect contiguous regions of the genome, for each chromosome  $c$ , we model the collection of random variables  $x_{1,c}, x_{2,c}, \dots$  using a hidden Markov model (HMM). The maximum copy number in each chromosome is modelled using a geometric random variable  $m_c$  with a shared latent parameter  $p$ .

We placed an exponential prior on  $\sigma$  with rate 1/10. We posit normal priors on the coefficients  $\theta$  (mean zero, variance 100). We put a uniform prior on  $p$ . The transition probabilities of the Markov chain are determined by the marginals of a Juke Cantor model with a global rate parameter  $\tau$ . We put a unit exponential prior on  $\tau$ . Full model specification is available at <https://github.com/UBC-Stat-ML/nowellpack/blob/multimodality-example/src/main/java/chromobreak/SingleCell.bl>.

## I.5 Multimodality of the examples considered

Figures 4 (bottom left), 22 and 23 support the multimodality of two of the examples considered in Section 7.3, namely the Ising and Spike-and-Slab examples. Multimodality of other examples considered in Section 7.1, 7.2, 7.4 and 7.5 is demonstrated in Figures 4, 27, 25, 28 and 29. Moreover, Figures 4, 25, 28 and 29 demonstrate that using standard MCMC is insufficient to explore these multimodal distributions.

## I.6 Selection of $\pi_0$

In our experiments all Bayesian models considered have proper priors, and hence we set  $\pi_0$  to the prior in all these situations. The local exploration kernel in this case consists in an independent

draw from the prior (performed by sorting the latent random variables according to a linearization of the partial order induced by the directed graphical model, and sampling their values according to these sorted laws).

For the Ising model, we let  $\pi_0$  denote a product of independent and identically distributed Bernoulli(1/2) random variables, one for each node in the Ising grid. It is then straightforward to interpolate between this product distribution and the Ising model of interest using a geometric average. Independent sampling from  $\pi_0$  is used for the local exploration kernel at  $\beta = 0$ .

Similarly, for the rotor (XY) model, we take  $\pi_0$  to be a product of independent and identically distributed Uniform( $-\pi, \pi$ ) random variables, and we use a geometric interpolation between  $\pi_0$  and the target distribution.



## Appendix J Supplementary figures for Section 7

### J.1 Section 7.1

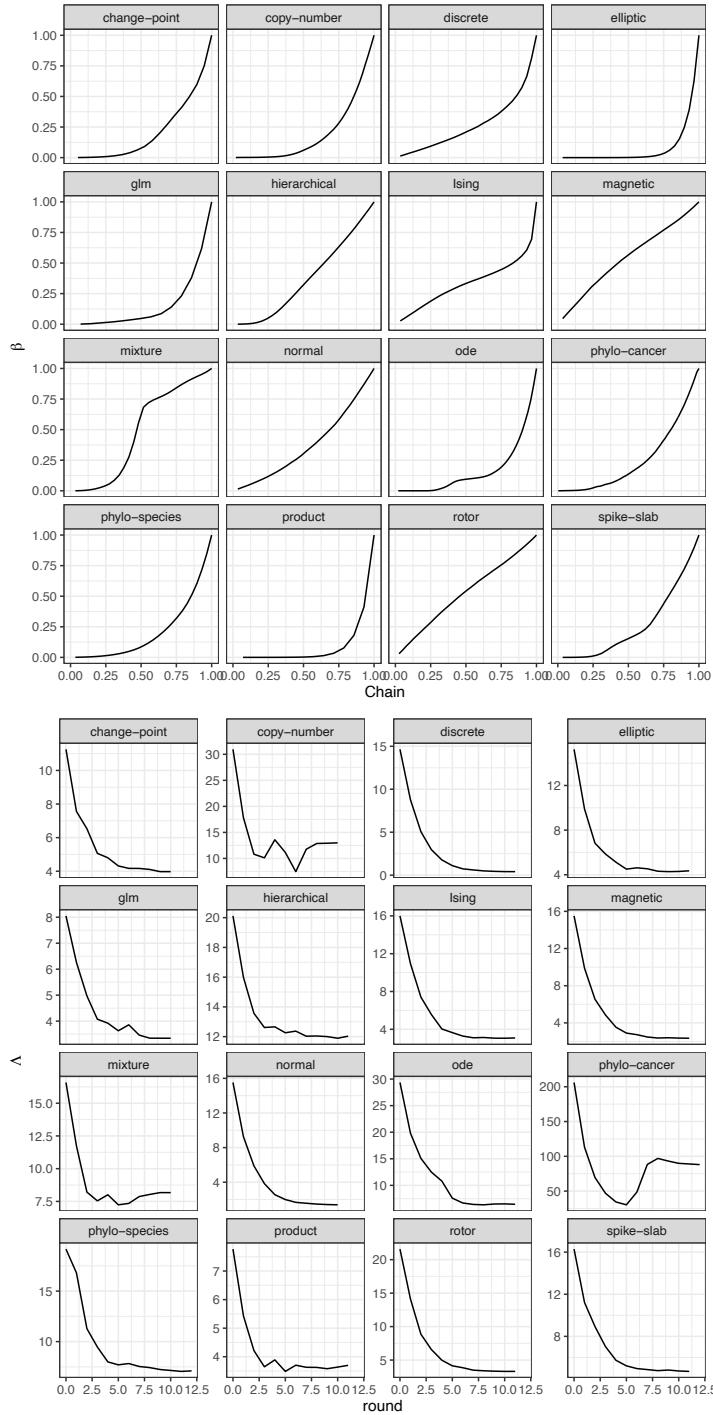


Figure 20: Top: Estimates of the schedule generator  $G$  for 16 models. The abscissa denotes the normalized chain indices  $i/N$ , the ordinates, parameters  $\beta$ . The function  $G$  is such that the partition  $G(0), G(1/N), G(2/N), \dots, G(1)$  approximates equi-acceptance. Bottom: Estimates of  $\Lambda$  for 16 models. The abscissa denotes the schedule optimization round.

## J.2 Section 7.3

Model (and dataset when applicable)	Method	$N$	$\hat{s}$ (%)	$\hat{\Lambda}$
Spike-and-slab classification (RMS Titanic passengers data, [Hind, 2019])	NRPT+SCM	15	67.6	4.54
	ARR	6	36.3	3.82
	MMV	6	39.9	3.61
	NRPT	15	67.2	4.59
Bayesian hierarchical model (historical rocket failure data, [McDowell, 2019])	NRPT+SCM	34	63.7	11.96
	ARR*	2*	0*	1*
	MMV	15	33.6	9.96
	NRPT	34	63.4	12.08
Wright-Fisher diffusion	NRPT+SCM	15	75.2	3.47
	ARR	5	43.8	2.81
	MMV	5	43.7	2.82
	NRPT	15	75.5	3.42
Ising model	NRPT+SCM	15	78.2	3.05
	ARR	4	37.9	2.48
	MMV	4	36.9	2.52
	NRPT	15	78.5	3.01

Figure 21: Summary statistics for the experiments in Section 7.3. The row marked with a star indicates failed optimization of one of the stochastic optimization schemes on the Bayesian hierarchical model.

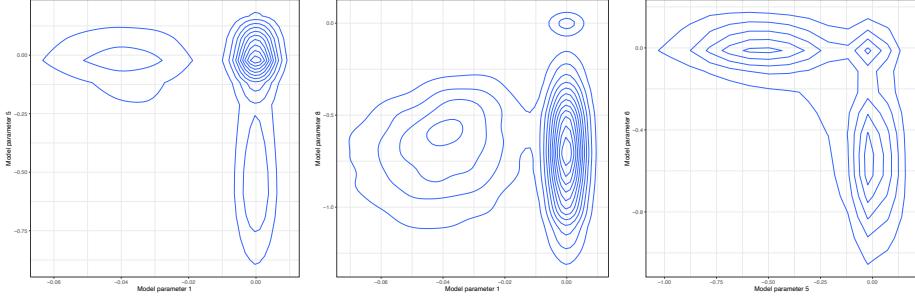


Figure 22: Examples of multimodality encountered in the Spike-and-Slab model applied to the RMS Titanic passenger dataset. Joint posterior distribution of the following regression parameters (1) `passengerAge` and `passengerClass`; (2) `passengerAge` and `SiblingsSpousesAboard`; (3) `passengerClass` and `passengerClass2`, the latter corresponding to an artificially duplicated regressor used to investigate the effect of co-linearity on the posterior approximation.

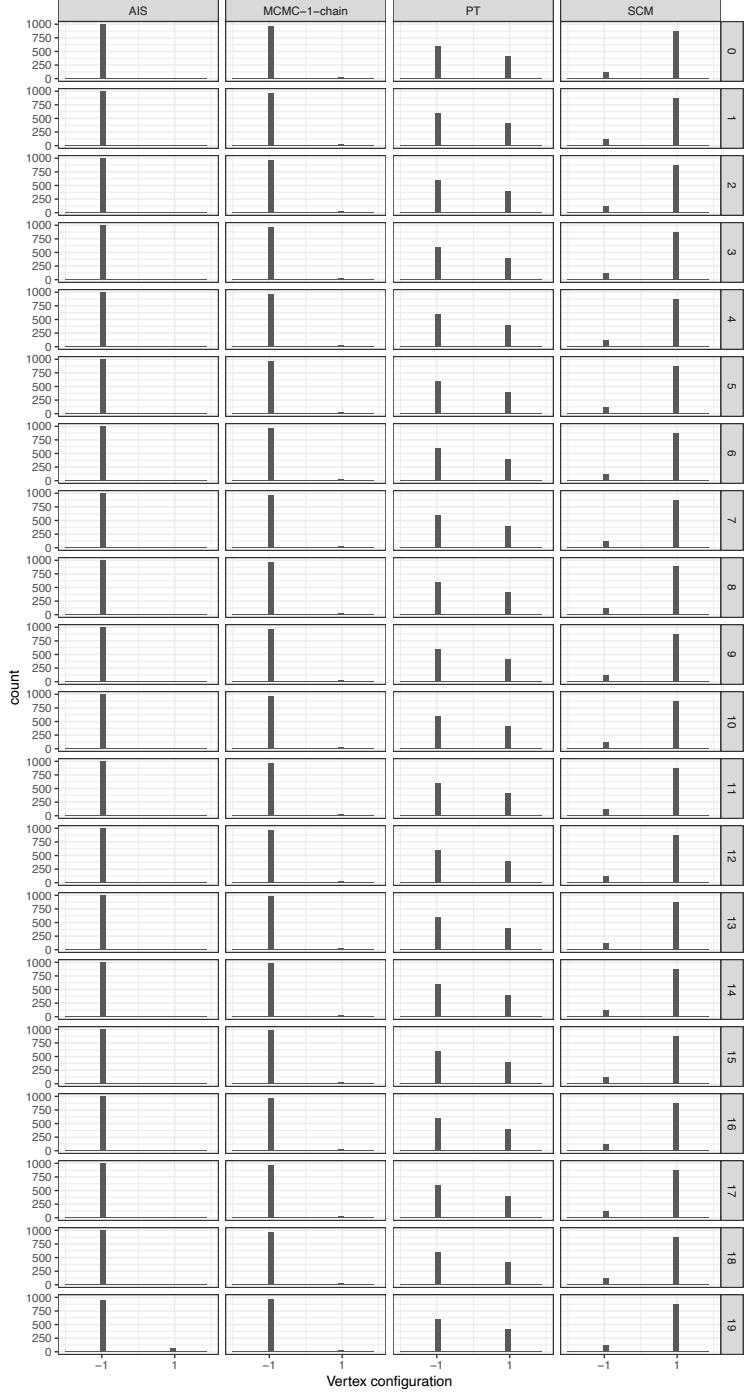


Figure 23: Example of multimodality in an Ising model. Each facet is a posterior probability mass function, facet rows index the first 20 vertices of the Ising graph, facet columns, four posterior approximation methods. By symmetry, each marginal should place equal mass at spins  $-1$  and  $1$ . The multimodality is only correctly captured by our proposed algorithm (NRPT, third column, wall clock time of 7.250s,  $N = 10$ ). For the other methods, the posterior approximation either misses the other mode completely, namely when using AIS (left column, wall clock time of 5.723s, 1000 particles, relative ESS adaptive annealing schedule) or single-chain MCMC (second column, wall clock time of 1.439s), or, for annealed SMC (fourth column, wall clock time of 8.269s, 1000 particles, relative ESS adaptive annealing schedule), detects the multimodality but not their respective proportions. All wall clock time are reported for a 2.8 GHz Intel Core i7.

### J.3 Section 7.4

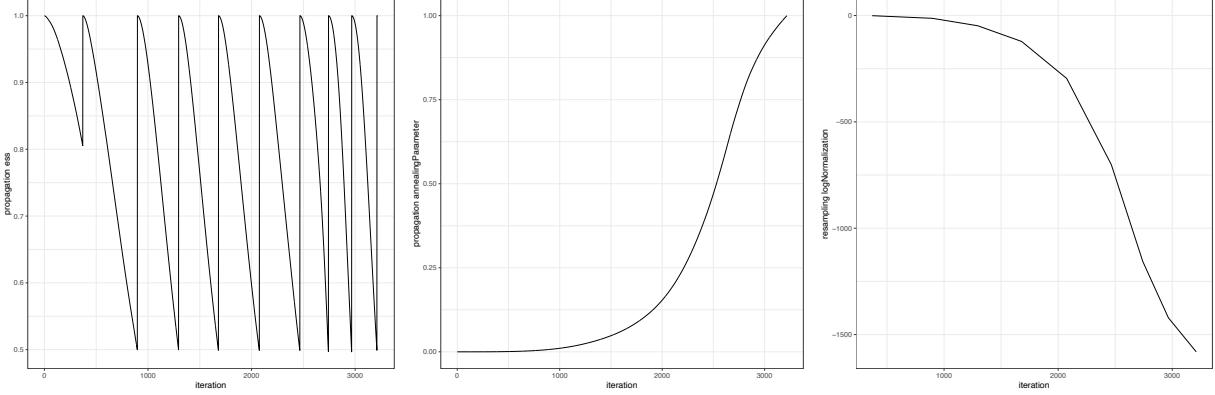


Figure 24: Diagnostics for the adaptive annealed SMC method (SCM), used as a benchmark on the Bayesian mixture problem. From left to right: (1) ESS as a function of the SMC iteration. Resampling is performed when the ESS drops below  $1/2$ ; (2) annealing parameter as a function of the SMC iteration; (3) log normalization estimates at each resampling step.

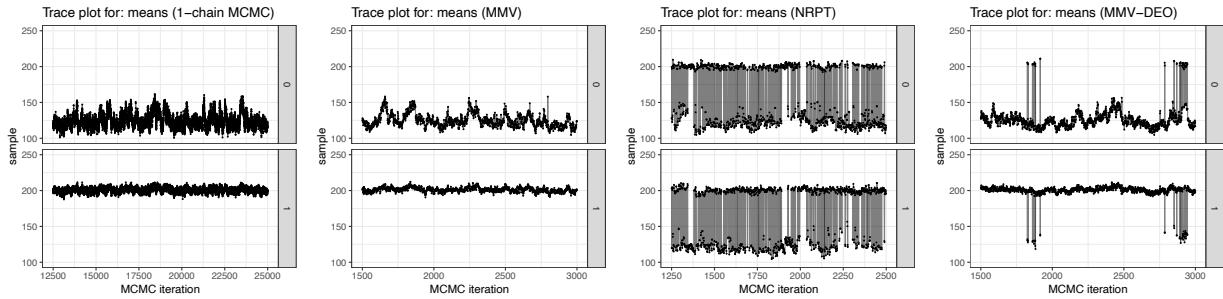


Figure 25: Mixture modelling example: post burn-in trace plots of two model parameters (facet rows) for four MCMC methods considered (facet columns). The two parameters correspond to the location parameters of two exchangeable clusters. Correct MCMC exploration of this unidentifiable model requires label switching, providing a test bed for MCMC over multimodal targets. All methods use a computational budget lower or equal to NRPT's.

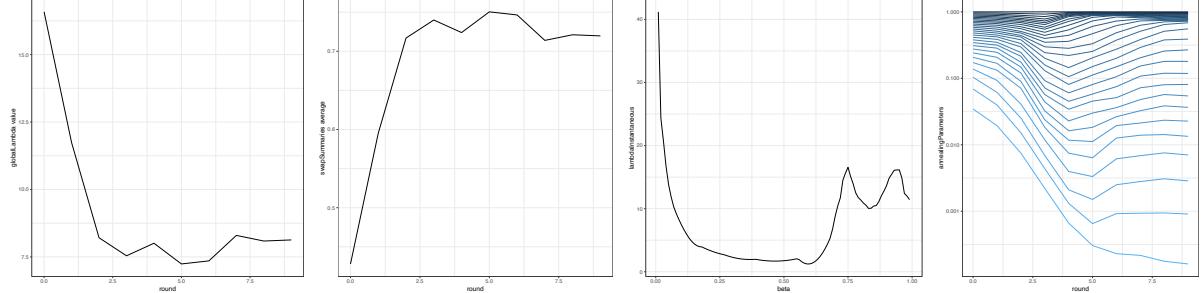


Figure 26: Diagnostics for NRPT on the mixture modelling example. From left to right: (1) Estimate of the global communication barrier  $\hat{\Lambda}$  as a function of the schedule optimization round. (2) Average swap acceptance probability across the 30 chains as a function of the schedule optimization round. The final value is 72%. (3) Estimated communication barrier  $\lambda$  output by NRPT. The two peaks can be interpreted as transition points where the cluster membership indicator variables go from disorganized (all cluster membership variables are i.i.d. at  $\beta = 0$ ), to all taking the same value within a cluster. The two clusters having different number of data points and parameters induce two distinct transition points. The empirical behaviour observed is analogous to the phase transition found in statistical mechanics models such as the Ising model. Investigation of possible phase transitions in clustering models would be an interesting future direction of investigation, although somewhat orthogonal to this work. (4) Learning curves for the annealing parameters (ordinate axis, log scale) for the 30 NRPT chains (colours) as a function of the schedule optimization round (abscissa).

#### J.4 Section 7.5

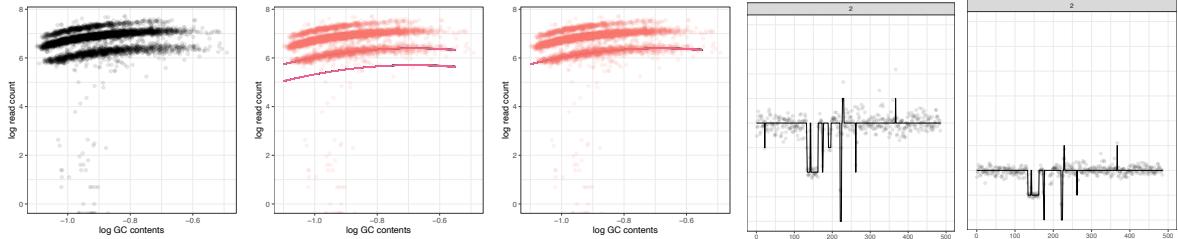


Figure 27: Copy number inference: inputs and outputs. From left to right: (1) raw data, where each dot represent a genomic bin  $i, c$ , with its abscissa showing  $\log g_{i,c}$  and its ordinate,  $\log y_{i,c}$ . (2) posterior distribution obtained from NRPT for the random function  $f(\theta, \cdot)$  obtained from NRPT. One translucent line  $f(\theta_n, \cdot)$  is drawn for each sampled  $\theta_n$ . Notice the bi-modality of the posterior distribution. (3) The approximation of the same posterior distribution based on a single chain misses one of the modes. (4) Two regions in chromosome 2 supporting a plausible genome duplication (and hence multimodality). We show here a sample index  $n$  corresponding to the tetraploid mode in the second sub-panel. The dots show the data after GC-correction based on sample  $n$ , i.e.  $\exp(\log y_{i,c} - f(\theta_n, \log g_{i,c}))$ . The line shows the imputed copy numbers at iteration  $n$ . Notice the two small regions with odd copy numbers, around position 25 and just before position 200. (5) Same visualization as panel (4) but for a sample index  $n$  corresponding to the diploid mode. Regions 25 and 200 cannot be explained without assuming genome duplication, however since the regions supporting duplication are small, the model remains uncertain about ploidy and hence multimodal.

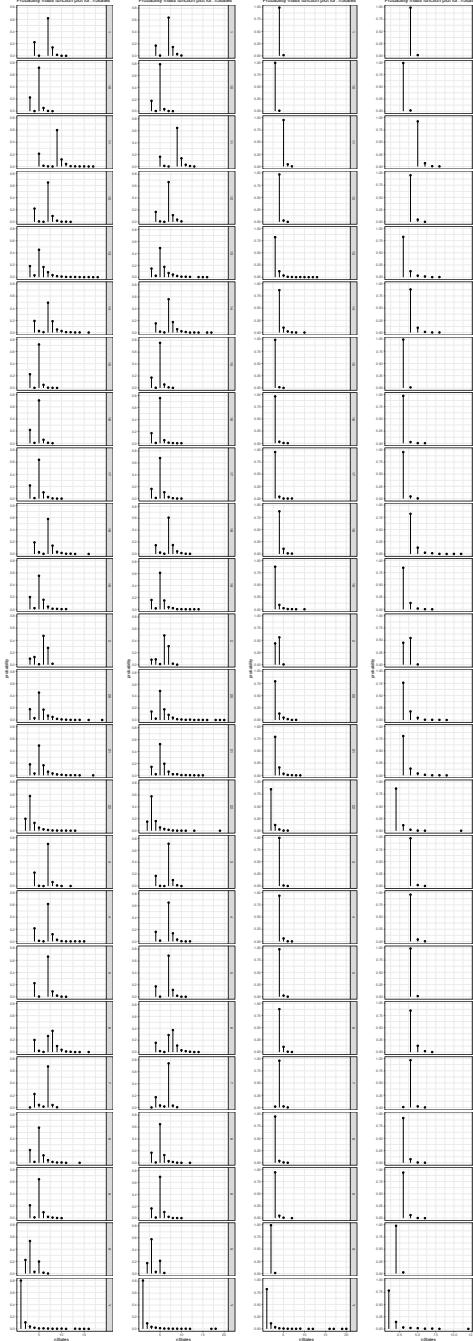


Figure 28: Multimodality of the random variables  $m_c$  controlling the size of the copy number space for each chromosome  $c$ . Each facet displays an approximate posterior probability mass function. Facet rows are indexed by chromosomes. Facet columns are indexed by posterior inference methods, which are all MCMC schemes ran for 5000 iterations. (1, 2) The multimodality arising from a weakly supported genome duplication is captured by NRPT, based on  $N = 50$  and  $N = 25$  respectively. (3) Results for an MCMC run based on the same exploration kernel misses the multimodality. (4) The stochastic adaptation method MMV fails to learn a schedule capable of performing round trips within the 5000 iterations and therefore misses the multimodality as well. See Section 7.5 for details.

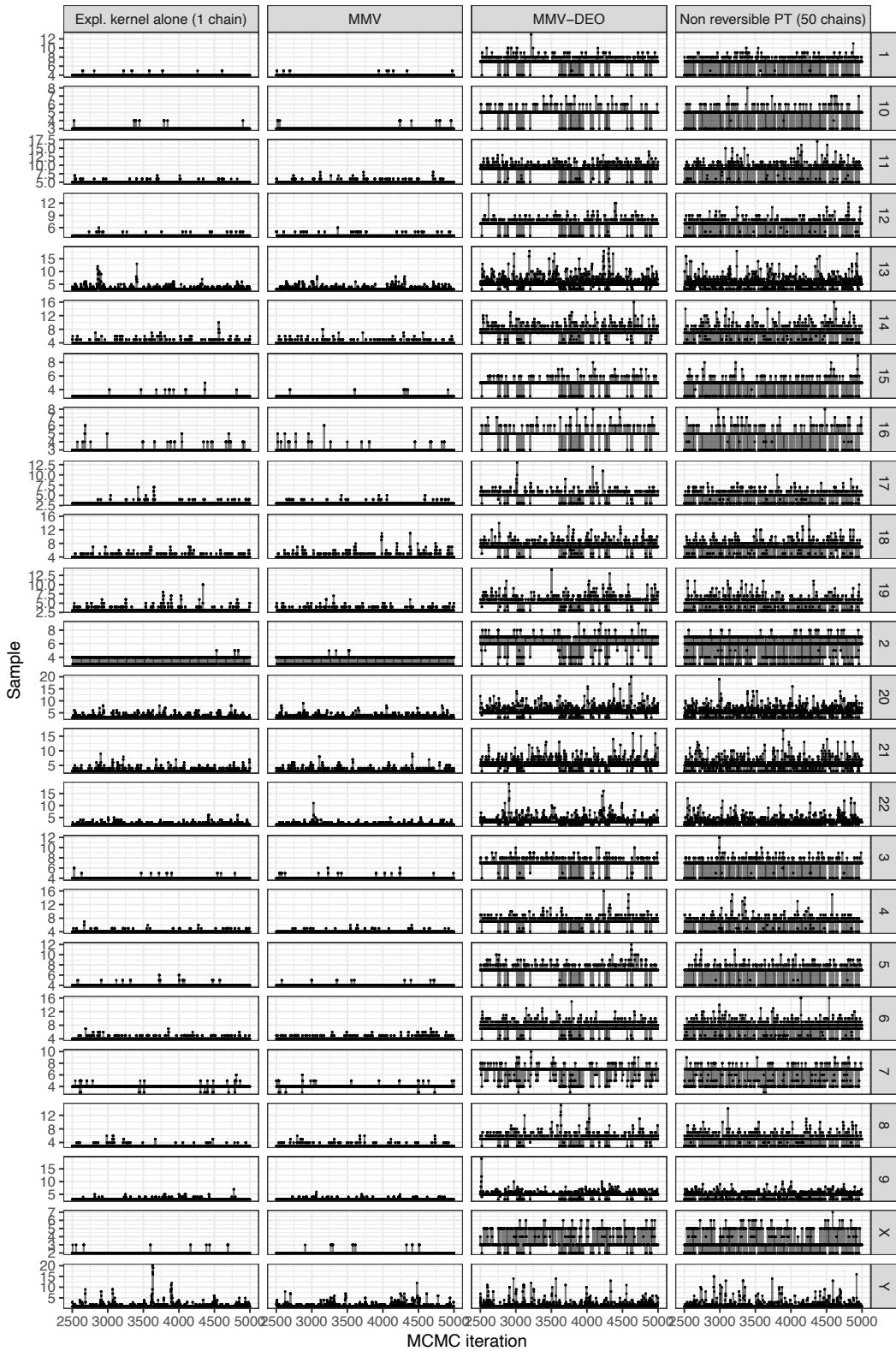


Figure 29: Enlarged version of Figure 12.

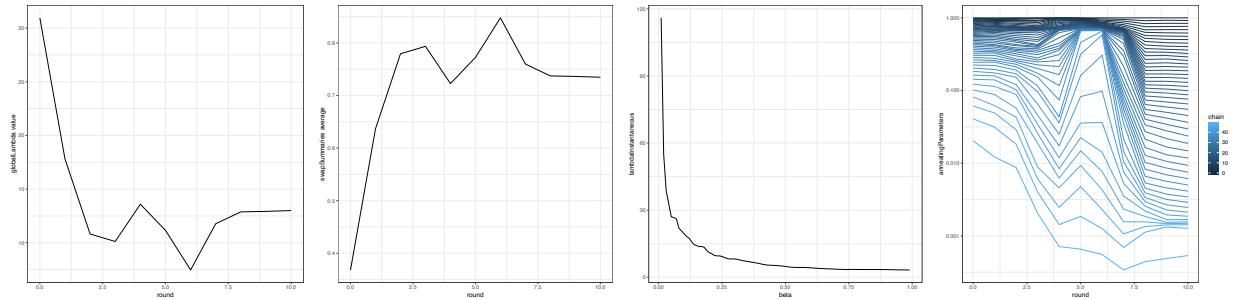


Figure 30: NRPT diagnostics on the copy number inference example. From left to right: (1) Estimate of the global communication barrier  $\hat{\Lambda}$  as a function of the schedule optimization round. (2) Average swap acceptance probability across the 50 chains as a function of the schedule optimization round. (3) Estimated communication barrier  $\lambda$  output by NRPT. (4) Learning curve for the annealing parameters (ordinate axis, log scale) for the 50 NRPT chains (colours) as a function of the schedule optimization round (abscissa).