

Generalized Parallel Tempering on Bayesian Inverse Problems

Jonas Latz¹, Juan P. Madrigal-Cianci^{*2}, Fabio Nobile², and Raúl Tempone³

¹*Department of Applied Mathematics and Theoretical Physics, University of Cambridge.*

²*SB-MATH-CSQI, École Polytechnique Fédérale de Lausanne.*

³*Computer, Electrical and Mathematical Sciences and Engineering, KAUST, and Alexander von Humboldt professor in Mathematics of Uncertainty Quantification, RWTH Aachen University.*

Abstract

In the current work we present two generalizations of the Parallel Tempering algorithm, inspired by the so-called continuous-time Infinite Swapping algorithm, which found its origins in the molecular dynamics community, and can be understood as the limit of a continuous-time Parallel Tempering algorithm with state-dependent swapping rates. In the current work, we extend this idea to the context of time-discrete Markov chains and present two Markov chain Monte Carlo algorithms that follow the same paradigm as the continuous-time infinite swapping procedure. We analyze the reversibility and ergodicity properties of our generalized PT algorithms. Numerical results on sampling from different target distributions originating from Bayesian inverse problems, show that the proposed methods significantly improve sampling efficiency over more traditional sampling algorithms such as Random Walk Metropolis and (standard) Parallel Tempering.

Keywords

Bayesian inversion. Parallel tempering. Infinite swapping. Markov chain Monte Carlo. Uncertainty quantification

1 Introduction

Modern computational facilities and recent advances in computational techniques have made the use of Markov Chain Monte Carlo (MCMC) methods

^{*}Corresponding author: juan.madrigalcianci@epfl.ch

feasible for some large-scale Bayesian inverse problems (BIP), where the goal is to characterize the posterior distribution of a set of parameters θ of a computational model which describes some physical phenomena, conditioned on some (usually indirectly) measured data y . However, some computational difficulties are prone to arise when dealing with *difficult to explore* posteriors, i.e., posterior distributions that are multi-modal, or that concentrate around a non-linear, lower-dimensional manifold, since some of the more commonly-used Markov transition kernels in MCMC algorithms, such as random walk Metropolis (RWM) or preconditioned Crank-Nicholson (pCN), are not well-suited in such situations. This in turn can make the computational time needed to properly *explore* these complicated target distributions arbitrarily long. Some recent works address these issues by employing Markov transitions kernels that use geometric information [7]; however, this requires efficient computation of the gradient of the posterior density, which might not always be feasible, particularly when the underlying computational model is a so-called “black-box”.

In recent years, there has been an active development of computational techniques and algorithms to overcome these issues using a *tempering strategy* [13, 17, 26, 33, 40]. Of particular importance for the work presented here is the Parallel Tempering (PT) algorithm [17, 24, 33] (also known as *replica exchange*), which finds its origins in the physics and molecular dynamics community. The general idea behind such methods is to simultaneously run K independent MCMC chains, where each chain is invariant with respect to a *flattened* (referred to as *tempered*) version of the posterior of interest μ , while, at the same time, proposing to swap states between any two chains every so often. Such a swap is then accepted using the standard Metropolis-Hastings (MH) acceptance-rejection rule. Intuitively, chains with a larger smoothing parameter (referred to as *temperature*) will be able to better explore the parameter space. Thus, by proposing to exchange states between chains that target posteriors at different temperatures, it is possible for the chain of interest (i.e., the one targeting μ) to mix faster, and to avoid the undesirable behavior of some MCMC samplers, to get “stuck” in a mode. Moreover, the fact that such an exchange of states is accepted with the typical MH acceptance-rejection rule, will guarantee that the chain targeting μ remains invariant with respect to such probability measure [17].

Tempering ideas have been successfully used to sample from posterior distributions arising in different fields of science, ranging from astrophysics to machine learning [12, 17, 33, 39]. [29, 41] have studied the convergence of the PT algorithm from a theoretical perspective and provided minimal conditions for its rapid mixing. Moreover, the idea of tempered distributions has not only been applied in combination with parallel chains. For example, the simulated tempering method [30] uses a single chain and varies the temperature within this chain. In addition, tempering forms the basis of efficient particle filtering methods for stationary model parameters in Sequential Monte Carlo settings [5, 6, 22, 23, 26] and Ensemble Kalman Inversion [9].

A generalization over the PT approach, originating from the molecular dynamics community, is the so-called *Infinite Swapping (IS)* algorithm [16, 35].

As opposed to PT, this IS paradigm is a continuous-time Markov process and considers the limit where states between chains are swapped infinitely often. It is shown in [16] that such an approach can in turn be understood as a swap of dynamics, i.e., kernel and temperature (as opposed to states) between chains. We remark that once such a change in dynamics is considered, it is not possible to distinguish particles belonging to different chains. However, since the stationary distribution of each chain is known, importance sampling can be employed to compute posterior expectations with respect to the target measure of interest. Infinite Swapping has been successfully applied in the context of computational molecular dynamics and rare event simulation [15, 28, 35]; however, to the best of our knowledge, such methods have not been implemented in the context of Bayesian inverse problems. In light of this, the current work aims at importing such ideas to the BIP setting, by presenting them in a discrete-time Metropolis-Hastings Markov chain Monte Carlo context. We will refer to these algorithms as *Generalized Parallel Tempering* (GPT). We emphasize, however, that these methods are *not* a time discretization of the continuous-time Infinite Swapping presented in [16], but, in fact, a discrete-time Markov process inspired by the ideas presented therein with suitably defined state-dependent probabilities of swapping states or dynamics. We now summarize the main contributions of this work.

We propose and implement two discrete-time MCMC generalized PT algorithms in the setting of Bayesian inverse problems. Furthermore, we prove that our GPT methods have the right invariant measure, by showing reversibility of the generated Markov chains, and prove their ergodicity. Finally, we implement the proposed GPT algorithms for an array of Bayesian inverse problems, comparing their efficiency to that of Random walk Metropolis (RWM) and standard PT. Experimental results show improvements in terms of computational efficiency of GPT over RWM and PT, thus making the proposed methods attractive from a computational perspective. We notice that a PT algorithm with state-dependent swapping probabilities has been proposed in [24], however, such a work only consider pairwise swapping of chains and a different construction of the swapping probabilities, resulting in a less-efficient sampler, at least for the BIPs addressed in this work. Our ergodicity result relies on an L_2 spectral gap analysis. It is known [38] that when a Markov chain is both reversible and has a positive L_2 -spectral gap, one can in turn provide non-asymptotic error bounds on the mean square error of an ergodic estimator of the chain. Our bounds on the L_2 -spectral gap, however, are far from being sharp and could possibly be improved using e.g., domain decomposition ideas as in [41]. Such analysis is left for a future work.

The rest of this paper is organized as follows. Section 2 is devoted to the introduction of the notation, Bayesian inverse problems, and Markov chain Monte Carlo methods. In Section 3, we provide a brief review of (standard) PT (Section 3.2), and introduce the two versions of the GPT algorithm in Sections 3.3 and 3.4, respectively. In fact, we present a general framework that accommodates both the standard PT algorithms and our generalized versions. In Section 4, we recall some of the standard theory of Markov chains in Section 4.1 and

state the main theoretical results of the current work (Theorems 4.5 and 4.7) in Section 4.2. The proof of these Theorems is given by a series of Propositions in Section 4.2. We present some numerical experiments in Section 5, and draw some conclusions in Section 6.

2 Problem setting

2.1 Notation

Let $(W, \|\cdot\|)$ be a separable Banach space with associated Borel σ -algebra $\mathcal{B}(W)$, and let ν_W be a σ -finite “reference” measure on W . For any measure μ on $(W, \mathcal{B}(W))$ that is absolutely continuous with respect to ν_W (in short $\mu \ll \nu_W$), we define the Radon-Nikodym derivative $\pi_\mu := \frac{d\mu}{d\nu_W}$. We denote by $\overline{\mathcal{M}}(W)$ the set of real-valued signed measures on $(W, \mathcal{B}(W))$, and by $\mathcal{M}(W) \subset \overline{\mathcal{M}}(W)$ the set of probability measures on $(W, \mathcal{B}(W))$.

Let $\mathcal{Q} : W \rightarrow \mathbb{R}$ be an integrable function with respect to a measure $\mu \ll \nu_W$, which we call *quantity of interest*. We define the *expected value* of \mathcal{Q} with respect to μ by

$$\mu(\mathcal{Q}) := \mathbb{E}_\mu[\mathcal{Q}] := \int_W \mathcal{Q} d\mu = \int_W \mathcal{Q} \pi_\mu d\nu_W.$$

Let now W_1, W_2 be two Banach spaces with reference measures ν_{W_1}, ν_{W_2} , and let $\mu_1 \ll \nu_{W_1}, \mu_2 \ll \nu_{W_2}$ be two probability measures, with corresponding densities given by π_{μ_1}, π_{μ_2} . The *product* of these two measures is defined by

$$\boldsymbol{\mu}(A) = (\mu_1 \times \mu_2)(A) = \iint_A \pi_{\mu_1}(\theta_1) \pi_{\mu_2}(\theta_2) \nu_{W_1}(d\theta_1) \nu_{W_2}(d\theta_2), \quad \forall A \in \mathcal{B}(W_1 \times W_2).$$

Joint measures on $W_1 \times W_2, \mathcal{B}(W_1 \times W_2)$ will always be written in boldface.

A *Markov kernel* on a Banach space W is a function $p : W \times \mathcal{B}(W) \rightarrow [0, 1]$ such that

1. For each A in $\mathcal{B}(W)$, the mapping $W \ni \theta \mapsto p(\theta, A)$, is a $\mathcal{B}(W)$ -measurable real-valued function.
2. For each θ in W , the mapping $\mathcal{B}(W) \ni A \mapsto p(\theta, A)$, is a probability measure on $(W, \mathcal{B}(W))$.

We denote by P the Markov transition operator associated to p , which acts to the left on measures, $\nu \mapsto \nu P \in \mathcal{M}(W)$, and to the right on functions, $f \mapsto Pf$, measurable on $(W, \mathcal{B}(W))$, such that

$$\begin{aligned} (\nu P)(A) &= \int_W p(\theta, A) \nu(d\theta), \quad \forall A \in \mathcal{B}(W), \\ (Pf)(\theta) &= \int_W f(z) p(\theta, dz), \quad \forall \theta \in W. \end{aligned}$$

Let W_1, W_2 be two separable Banach spaces and P_k , $k = 1, 2$, be Markov transition operators associated to kernels $p_k : W_k \times \mathcal{B}(W_k) \mapsto [0, 1]$. We define the *tensor product Markov operator* $\mathbf{P} := P_1 \otimes P_2$ as the Markov operator associated with the product measure $\mathbf{p}(\boldsymbol{\theta}, \cdot) = p_1(\theta_1, \cdot) \times p_2(\theta_2, \cdot)$, $\boldsymbol{\theta} = (\theta_1, \theta_2) \in W_1 \times W_2$. In particular, $\nu\mathbf{P}$ is the measure on $(W_1 \times W_2, \mathcal{B}(W_1 \times W_2))$ that satisfies

$$(\nu\mathbf{P})(A_1 \times A_2) = \iint_{W_1 \times W_2} p_1(\theta_1, A_1) p_2(\theta_2, A_2) \nu(d\theta_1, d\theta_2),$$

for all $A_1 \in \mathcal{B}(W_1)$ and $A_2 \in \mathcal{B}(W_2)$. Moreover, $(\mathbf{P}f) : W_1 \times W_2 \rightarrow \mathbb{R}$ is the function given by

$$(\mathbf{P}f)(\boldsymbol{\theta}) = \iint_{W_1 \times W_2} f(z_1, z_2) p_1(\theta_1, dz_1) p_2(\theta_2, dz_2),$$

for an appropriate $f : W_1 \times W_2 \rightarrow \mathbb{R}$.

We say that a Markov operator P (resp. \mathbf{P}) is *invariant* with respect to a measure ν (resp. ν) if $\nu P = \nu$ (resp. $\nu\mathbf{P} = \nu$). A related concept to invariance is that of reversibility; a Markov kernel $p : W \times \mathcal{B}(W) \mapsto [0, 1]$ is said to be *reversible* (or ν -reversible) with respect to a measure $\nu \in \mathcal{M}(W)$ if

$$\int_B p(\theta, A) \nu(d\theta) = \int_A p(\theta, B) \nu(d\theta), \quad \forall A, B \in \mathcal{B}(W). \quad (1)$$

Clearly, if a Markov kernel is reversible with respect to a measure ν , then the associated Markov operator P has ν as an invariant measure. The reverse is not true, in general. For two given ν -invariant Markov operators P_1, P_2 , we say that $P_1 P_2$ is a *composition* of Markov operators, not to be confused with $P_1 \otimes P_2$. Furthermore, given a composition of K ν -invariant Markov operators $P_c := P_1 P_2 \dots P_K$, we say that P_c is *palindromic* if $P_1 = P_K$, $P_2 = P_{K-1}$, \dots , $P_k = P_{K-k+1}$, $k = 1, 2, \dots, K$. It is known (see, e.g., [8, Section 1.12.17]) that a palindromic, ν -invariant Markov operator P_c has an associated Markov transition kernel p_c which is ν -reversible.

2.2 Bayesian inverse problems

Let $(\Theta, \|\cdot\|_\Theta)$ and $(Y, \|\cdot\|_Y)$ be separable Banach spaces with associated Borel σ -algebras $\mathcal{B}(\Theta)$, $\mathcal{B}(Y)$. In Bayesian Inverse Problems we aim at characterizing the probability distribution of a set of parameters $\theta \in \Theta$ conditioned on some measured data $y \in Y$, where the relation between θ and y is given by:

$$y = \mathcal{F}(\theta) + \eta, \quad \eta \sim \mu_{\text{noise}}. \quad (2)$$

Here η is some random noise with known distribution μ_{noise} (assumed to have a density π_{noise} with respect to some reference measure ν_Y on Y) and $\mathcal{F} : \Theta \mapsto Y$ is the so-called *forward mapping operator*. Such an operator can model, e.g., the numerical solution of a possibly non-linear Partial Differential Equation (PDE)

which takes θ as a set of parameters. Assume that $\theta \sim \mu_{\text{pr}}$ and define the *potential* function $\Phi(\theta; y) : \Theta \times Y \mapsto [0, \infty)$ as

$$\Phi(\theta; y) = -\log [\pi_{\text{noise}}(y - \mathcal{F}(\theta))] , \quad (3)$$

where the function $\Phi(\theta; y)$ is a measure of the misfit between the recorded data y and the predicted value $\mathcal{F}(\theta)$, and often depends only on $\|y - \mathcal{F}(\theta)\|_Y$. Furthermore, assuming that the prior measure μ_{pr} has a density π_{pr} with respect to some σ -finite measure ν_{Θ} , we have that the solution to the Bayesian inverse problem is given by (see, for example, [25, Theorem 2.5])

$$\pi^y(\theta) := \pi(\theta|y) = \frac{1}{Z} e^{-\Phi(\theta; y)} \pi_{\text{pr}}(\theta), \quad (4)$$

where μ^y (with corresponding ν_{Θ} -density π^y) is referred to as the *posterior measure* and $Z := \int_{\Theta} \exp(-\Phi(\theta; y)) \mu_{\text{pr}}(d\theta)$. The Bayesian approach to the inverse problem consists of updating our knowledge concerning the parameter θ , i.e., the prior, given the information that we observed in Equation (2). One way of doing so is to generate samples from the posterior measure μ^y . A common method for performing such a task is to use Markov chain Monte Carlo (MCMC) algorithms, as detailed in the next section. Once samples $\{\theta^{(n)}\}_{n=0}^N$ have been obtained by some MCMC algorithm, the posterior expectation $\mathbb{E}_{\mu^y}[\mathcal{Q}]$ of some μ^y -integrable quantity of interest $\mathcal{Q} : \Theta \mapsto \mathbb{R}$ can be approximated by the following ergodic estimator

$$\mathbb{E}_{\mu^y}[\mathcal{Q}] \approx \hat{\mathcal{Q}} := \frac{1}{N-b} \sum_{n=b}^N \mathcal{Q}(\theta^{(n)}),$$

where $b < N$ is the so-called *burn-in* period, used to reduce the bias typically associated to MCMC algorithms.

2.3 Markov Chain Monte Carlo and tempering

The main idea behind using Markov chain Monte Carlo methods to sample a measure of interest μ on $(\Theta, \mathcal{B}(\Theta))$, is to create a Markov chain whose initial state θ^0 has some distribution $\nu \in \mathcal{M}(\Theta)$ and whose Markov operator P is invariant with respect to μ , i.e., $\mu P = \mu$. The Markov chain $\{\theta^n\}_{n=0}^N$ is then generated by sampling $\theta^{(n)} \sim p(\theta^{n-1}, \cdot), \forall n \in \mathbb{N}$. One of the most common approaches for performing such a task is the Metropolis-Hastings algorithm [21, 31]. Let $q_{\text{prop}} : \Theta \times \mathcal{B}(\Theta) \mapsto [0, 1]$ be an auxiliary kernel. The Metropolis-Hastings algorithm works as follows. For $n = 1, 2, \dots$, a candidate state θ^* is sampled from $q_{\text{prop}}(\theta^n, \cdot)$, and proposed as the new state of the chain at step $n + 1$. Such a state is then accepted (i.e., we set $\theta^{n+1} = \theta^*$), with probability α_{MH} ,

$$\alpha_{\text{MH}}(\theta^n, \theta^*) = \min \left\{ 1, \frac{\pi_{\mu}(\theta^*) q_{\text{prop}}(\theta^*, \theta^n)}{\pi_{\mu}(\theta^n) q_{\text{prop}}(\theta^n, \theta^*)} \right\},$$

otherwise the current state is retained, i.e., $\theta^{n+1} = \theta^n$. Notice that, with a slight abuse of notation, we are denoting kernel and density by the same symbol q_{prop} . The Metropolis-Hastings algorithm induces the *Markov transition kernel* $p : \Theta \times \mathcal{B}(\Theta) \mapsto [0, 1]$

$$p(\theta, A) = \int_A \alpha_{\text{MH}}(\theta, \theta^*) q_{\text{prop}}(\theta, d\theta^*) + \delta_\theta(A) \int_\Theta (1 - \alpha_{\text{MH}}(\theta, \theta^*)) q_{\text{prop}}(\theta, d\theta^*),$$

for every $\theta \in \Theta$ and $A \in \mathcal{B}(\Theta)$. In most practical algorithms, the proposal state θ^* is sampled from a state-dependent auxiliary kernel $q_{\text{prop}}(\theta^n, \cdot)$. Such is the case for *random walk Metropolis* or *preconditioned Crank Nicolson*, where $q_{\text{prop}}(\theta^n, \cdot) = \mathcal{N}(\theta^n, \Sigma)$ or $q_{\text{prop}}(\theta^n, \cdot) = \mathcal{N}(\sqrt{1 - \rho^2} \theta^n, \rho \Sigma)$, $0 < \rho < 1$, respectively. However, these types of *localized* proposals tend to present some undesirable behaviors when sampling from certain *difficult* measures, which are, for example, concentrated over a manifold or are multi-modal [17]. In the first case, in order to avoid a large rejection rate, the “step-size” $\|\Sigma^{1/2}\|$ of the proposal kernel must be quite small, which will in turn produce highly-correlated samples. In the second case, chains generated by these *localized* kernels tend to get stuck in one of the modes. In either of these cases, very long chains are required to properly explore the parameter space.

One way of overcoming such difficulties is to introduce tempering. Let μ_k, μ_{pr} be probability measures on $(\Theta, \mathcal{B}(\Theta))$, $k = 1, \dots, K$, such that all μ_k are absolutely continuous with respect to μ_{pr} , and let $\{T_k\}_{k=1}^K$ be a set of K *temperatures* such that $1 = T_1 < T_2 < \dots < T_K \leq \infty$. In a Bayesian setting, μ_{pr} corresponds to the prior measure and $\mu_k, k = 1, \dots, K$ correspond to posterior measures associated to different temperatures. Denoting by π_k the μ_{pr} -density of μ_k , we set

$$\pi_k(\theta) := \frac{e^{-\Phi(\theta; y)/T_k}}{Z_k}, \quad \theta \in \Theta, \quad (5)$$

where $Z_k := \int_\Theta e^{-\Phi(\theta; y)/T_k} \mu_{\text{pr}}(d\theta)$, and with $\Phi(\theta; y)$ as the potential function defined in (3). In the case where $T_K = \infty$, we set $\mu_K = \mu_{\text{pr}}$. Notice that μ_1 corresponds to the target posterior measure.

We say that for $k = 2, \dots, K$, each measure μ_k is a *tempered* version of μ_1 . In general, the $1/T_k$ term in (5) serves as a “smoothing”¹ factor, which in turn makes μ_k easier to explore as $T_k \rightarrow \infty$. In PT MCMC algorithms, we sample from all posterior measures μ_k simultaneously. Here, we first use a μ_k -reversible Markov transition kernel p_k on each chain, and then, we propose to exchange states between chains at two consecutive temperatures, i.e., chains targeting μ_k, μ_{k+1} , $k \in \{1, \dots, K-1\}$. Such a proposed swap is then accepted or rejected with a standard Metropolis-Hastings acceptance-rejection step. This procedure is presented in Algorithm 1. We remark that such an algorithm can be modified to, for example, propose to swap states every N_s steps of the chain, or to swap states between two chains μ_i, μ_j , with i, j chosen randomly and uniformly from

¹Here, smoothing is to be understood in the sense that it *flattens* the density.

the index set $\{1, 2, \dots, K\}$. In the next section we present the generalized PT algorithms which swap states according to a random permutation of the indices drawn from a state dependent probability.

Algorithm 1 Standard PT.

```

function STANDARD PT( $N, \{p_k\}_{k=1}^N, \{\pi_k\}_{k=1}^N, \mu_{\text{pr}}$ )
  Sample  $\theta_k^{(1)} \sim \mu_{\text{pr}}, k = 1, \dots, K$ 
  for  $n = 1, 2, \dots, N - 1$  do                                 $\triangleright$  Do one step of MH on each chain
    for  $k = 1, \dots, K$  do
      Sample  $\theta_k^{(n+1)} \sim p_k(\theta_k^{(n)}, \cdot)$ 
    end for                                                     $\triangleright$  Swap states
    for  $k = 1, 2, \dots, K - 1$  do
      Swap states  $\theta_k^{(n+1)}$  and  $\theta_{k+1}^{(n+1)}$  with probability  $\alpha_{\text{swap}} =$ 
       $\min \left\{ 1, \frac{\pi_k(\theta_{k+1}^{(n+1)})\pi_{k+1}(\theta_k^{(n+1)})}{\pi_k(\theta_k^{(n+1)})\pi_{k+1}(\theta_{k+1}^{(n+1)})} \right\}$ 
    end for
  end for
  Output  $\{\theta_1^{(n)}\}_{n=1}^N$ 
end function

```

3 Generalizing Parallel Tempering

Infinite Swapping was initially developed in the context of continuous-time MCMC algorithms, which were used for molecular dynamics simulations. In continuous-time PT, the swapping of the states is controlled by a Poisson process on the set $\{1, \dots, K\}$. Infinite Swapping is the limiting algorithm obtained by letting the waiting times of this Poisson process go to zero. Hence, we swap the states of the chain infinitely often over a finite time interval. We refer to [16] for a thorough introduction and review of Infinite Swapping in continuous-time. In Section 5 of the same article, the idea to use Infinite Swapping in time-discrete Markov chains was briefly discussed. Inspired by this discussion, we present two Generalizations of the (discrete-time) Parallel Tempering strategies. To that end, we propose to either (i) swap states in the chains at every iteration of the algorithm in such a way that the swap is accepted with probability one, which we will refer to as the *Unweighted Generalized Parallel Tempering (UGPT)*, or (ii), swap dynamics (i.e., swap kernels and temperatures between chains) at every step of the algorithm. In this case, importance sampling must also be used when computing posterior expectations since this in turn provides a Markov chain whose invariant measure is not the desired one. We refer to this approach as *Weighted Generalized Parallel Tempering (WGPT)*. We begin by introducing a common framework to both PT and the two versions of GPT.

Let $(\Theta, \|\cdot\|_{\Theta})$ be a separable Banach space with associated Borel σ -algebra $\mathcal{B}(\Theta)$. Let us define the K -fold product space $\Theta^K := \times_{k=1}^K \Theta$, with associ-

ated product σ -algebra $\mathcal{B}^K := \bigotimes_{k=1}^K \mathcal{B}(\Theta)$, as well as the product measures on $(\Theta^K, \mathcal{B}^K)$

$$\boldsymbol{\mu} := \bigotimes_{k=1}^K \mu_k, \quad (6)$$

where μ_k $k = 1, \dots, K$ are the tempered measures with temperatures $1 \leq T_1 < T_2 < T_3 < \dots < T_K \leq \infty$ introduced in the previous section. Similarly, we define the product prior measure $\boldsymbol{\mu}_{\text{prior}} := \bigotimes_{k=1}^K \mu_{\text{prior}}$. Notice that $\boldsymbol{\mu}$ has a density $\pi(\boldsymbol{\theta})$ with respect to $\boldsymbol{\mu}_{\text{prior}}$ given by

$$\pi(\boldsymbol{\theta}) = \prod_{k=1}^K \pi_k(\theta_k), \quad \boldsymbol{\theta} := (\theta_1, \dots, \theta_K) \in \Theta^K,$$

with $\pi_i(\theta)$ added subscript given as in (5). The idea behind the tempering methods presented herein is to sample from $\boldsymbol{\mu}$ (as opposed to solely sampling from μ_1) by creating a Markov chain obtained from the successive application of two $\boldsymbol{\mu}$ -invariant Markov kernels \mathbf{p} and \mathbf{q} , to some initial distribution $\boldsymbol{\nu}$, usually chosen to be the prior $\boldsymbol{\mu}_{\text{prior}}$. Each kernel acts as follows. Given the current state $\boldsymbol{\theta}^n = (\theta_1^n, \dots, \theta_K^n)$, the kernel \mathbf{p} , which we will call the *standard MCMC kernel*, proposes a new, intermediate state $\tilde{\boldsymbol{\theta}}^{n+1} = (\tilde{\theta}_1^{n+1}, \dots, \tilde{\theta}_K^{n+1})$, possibly following the Metropolis-Hastings algorithm (or any other algorithm that generates a $\boldsymbol{\mu}$ -invariant Markov operator). The Markov kernel \mathbf{p} is a product kernel, meaning that each component $\tilde{\theta}_k^n$, $k = 1 \dots, K$, is generated independently of the others. Then, the *swapping kernel* \mathbf{q} proposes a new state $\boldsymbol{\theta}^{n+1} = (\theta_1^{n+1}, \dots, \theta_K^{n+1})$ by introducing an “interaction” between the components of $\tilde{\boldsymbol{\theta}}^{(n+1)}$. This interaction step can be achieved, e.g., in the case of PT, by proposing to swap two components at two consecutive temperatures, i.e., components k and $k+1$, and accepting this swap with a certain probability given by the usual Metropolis-Hastings acceptance-rejection rule. In general, the swapping kernel is applied every N_s steps of the chain, for some $N_s \geq 1$. We will devote the following subsection to the construction of the swapping kernel \mathbf{q} .

3.1 The swapping kernel \mathbf{q}

Define \mathcal{S}_K as the collection of all the bijective maps from $\{1, 2, \dots, K\}$ to itself, i.e., the set of all $K!$ possible permutations of $\text{id} := \{1, \dots, K\}$. Let $\sigma \in \mathcal{S}_K$ be a permutation, and define the swapped state $\boldsymbol{\theta}_\sigma := (\theta_{\sigma(1)}, \dots, \theta_{\sigma(K)})$, and the inverse permutation $\sigma^{-1} \in \mathcal{S}_K$ such that $\sigma \circ \sigma^{-1} = \sigma^{-1} \circ \sigma = \text{id}$. In addition, let $S_K \subseteq \mathcal{S}_K$ be any subset of \mathcal{S}_K closed with respect to inversion, i.e., $\sigma \in S_K \implies \sigma^{-1} \in S_K$. We denote the cardinality of S_K by $|S_K|$. To define the swapping kernel \mathbf{q} , we first need to define the swapping ratio and swapping acceptance probability.

Definition 3.1 (Swapping ratio). We say that a function $r : \Theta^K \times S_K \mapsto [0, 1]$ is a *swapping ratio* if it satisfies the following two conditions:

1. $\forall \boldsymbol{\theta} \in \Theta^K$, $r(\boldsymbol{\theta}, \cdot)$ is a probability mass function on S_K .
2. $\forall \sigma \in S_K$, $r(\cdot, \sigma)$ is measurable on $(\Theta^K, \mathcal{B}^K)$.

Definition 3.2 (Swapping acceptance probability). Let $\boldsymbol{\theta} \in \Theta^K$ and $\sigma, \sigma^{-1} \in S_K$. We call *swapping acceptance probability* the function $\alpha_{\text{swap}} : \Theta^K \times S_K \mapsto [0, 1]$ defined as

$$\alpha_{\text{swap}}(\boldsymbol{\theta}, \sigma) = \begin{cases} \min \left\{ 1, \frac{\pi(\boldsymbol{\theta}_\sigma) r(\boldsymbol{\theta}_\sigma, \sigma^{-1})}{\pi(\boldsymbol{\theta}) r(\boldsymbol{\theta}, \sigma)} \right\}, & \text{if } r(\boldsymbol{\theta}, \sigma) > 0, \\ 0 & \text{if } r(\boldsymbol{\theta}, \sigma) = 0. \end{cases}$$

We can now define the swapping kernel \mathbf{q} .

Definition 3.3 (Swapping kernel). Given a swapping ratio $r : \Theta^K \times S_K \mapsto [0, 1]$ and its associated swapping acceptance probability $\alpha_{\text{swap}} : \Theta^K \times S_K \mapsto [0, 1]$, we define the *swapping Markov kernel* $\mathbf{q} : \Theta^K \times \mathcal{B}^K \mapsto [0, 1]$ as

$$\mathbf{q}(\boldsymbol{\theta}, B) = \sum_{\sigma \in S_K} r(\boldsymbol{\theta}, \sigma) \left[(1 - \alpha_{\text{swap}}(\boldsymbol{\theta}, \sigma)) \delta_{\boldsymbol{\theta}}(B) + \alpha_{\text{swap}}(\boldsymbol{\theta}, \sigma) \delta_{\boldsymbol{\theta}_\sigma}(B) \right], \quad \boldsymbol{\theta} \in \Theta^K, B \in \mathcal{B}^K, \quad (7)$$

where $\delta_{\boldsymbol{\theta}}(B)$ denotes the Dirac measure in $\boldsymbol{\theta}$, i.e., $\delta_{\boldsymbol{\theta}}(B) = 1$ if $\boldsymbol{\theta} \in B$ and 0 otherwise.

The swapping mechanism should be understood in the following way: given a current state of the chain $\boldsymbol{\theta} \in \Theta^K$, the swapping kernel samples a permutation σ from S_K with probability $r(\boldsymbol{\theta}, \sigma)$ and generates $\boldsymbol{\theta}_\sigma$. This permuted state is then accepted as the new state of the chain with probability $\alpha_{\text{swap}}(\boldsymbol{\theta}, \sigma)$. Notice that the swapping kernel follows a Metropolis-Hastings-like procedure with “proposal” distribution $r(\boldsymbol{\theta}, \sigma)$ and acceptance probability $\alpha_{\text{swap}}(\boldsymbol{\theta}, \sigma)$. Moreover, as detailed in the next proposition, such a kernel is reversible with respect to $\boldsymbol{\mu}$, since it is a Metropolis-Hastings type kernel.

Proposition 3.4. The Markov kernel \mathbf{q} defined in (7) is reversible with respect to the product measure $\boldsymbol{\mu}$ defined in (6).

Proof. Let $A, B \in \mathcal{B}^K$. We want to show that

$$\int_A \mathbf{q}(\boldsymbol{\theta}, B) d\boldsymbol{\mu}(d\boldsymbol{\theta}) = \int_B \mathbf{q}(\boldsymbol{\theta}, A) d\boldsymbol{\mu}(d\boldsymbol{\theta}).$$

Thus,

$$\begin{aligned} \int_A \mathbf{q}(\boldsymbol{\theta}, B) d\boldsymbol{\mu}(d\boldsymbol{\theta}) &= \underbrace{\sum_{\sigma \in S_K} \int_A r(\boldsymbol{\theta}, \sigma) \alpha_{\text{swap}}(\boldsymbol{\theta}, \sigma) \delta_{\boldsymbol{\theta}_\sigma}(B) \pi(\boldsymbol{\theta}) \boldsymbol{\mu}_{\text{prior}}(d\boldsymbol{\theta})}_I \\ &\quad + \underbrace{\sum_{\sigma \in S_K} \int_A r(\boldsymbol{\theta}, \sigma) (1 - \alpha_{\text{swap}}(\boldsymbol{\theta}, \sigma)) \delta_{\boldsymbol{\theta}}(B) \pi(\boldsymbol{\theta}) \boldsymbol{\mu}_{\text{prior}}(d\boldsymbol{\theta})}_{II}. \end{aligned}$$

Let $A_\sigma := \{z \in \Theta^K : z_{\sigma^{-1}} \in A\}$. From I , and noticing that μ_{prior} is permutation invariant, we get

$$\begin{aligned}
I &= \sum_{\sigma \in S_K} \int_A \min \left\{ 1, \frac{\pi(\theta_\sigma) r(\theta_\sigma, \sigma^{-1})}{\pi(\theta) r(\theta, \sigma)} \right\} r(\theta, \sigma) \pi(\theta) \delta_{\theta_\sigma}(B) \mu_{\text{prior}}(d\theta) \\
&= \sum_{\sigma \in S_K} \int_A \min \left\{ 1, \frac{\pi(\theta) r(\theta, \sigma)}{\pi(\theta_\sigma) r(\theta_\sigma, \sigma^{-1})} \right\} r(\theta_\sigma, \sigma^{-1}) \pi(\theta_\sigma) \delta_{\theta_\sigma}(B) \mu_{\text{prior}}(d\theta) \\
&= \sum_{\sigma \in S_K} \int_{A_\sigma} \min \left\{ 1, \frac{\pi(\theta_{\sigma^{-1}}) r(\theta_{\sigma^{-1}}, \sigma)}{\pi(\theta) r(\theta, \sigma^{-1})} \right\} r(\theta, \sigma^{-1}) \pi(\theta) \delta_\theta(B) \mu_{\text{prior}}(d\theta) \\
&= \sum_{\sigma \in S_K} \int_{A_\sigma \cap B} \min \left\{ 1, \frac{\pi(\theta_{\sigma^{-1}}) r(\theta_{\sigma^{-1}}, \sigma)}{\pi(\theta) r(\theta, \sigma^{-1})} \right\} r(\theta, \sigma^{-1}) \pi(\theta) \delta_\theta(B) \mu_{\text{prior}}(d\theta) \\
&= \sum_{\sigma \in S_K} \int_B \min \left\{ 1, \frac{\pi(\theta_{\sigma^{-1}}) r(\theta_{\sigma^{-1}}, \sigma)}{\pi(\theta) r(\theta, \sigma^{-1})} \right\} r(\theta, \sigma^{-1}) \pi(\theta) \delta_\theta(A_\sigma) \mu_{\text{prior}}(d\theta) \\
&= \sum_{\sigma \in S_K} \int_B \min \left\{ 1, \frac{\pi(\theta_{\sigma^{-1}}) r(\theta_{\sigma^{-1}}, \sigma)}{\pi(\theta) r(\theta, \sigma^{-1})} \right\} r(\theta, \sigma^{-1}) \pi(\theta) \delta_{\theta_{\sigma^{-1}}}(A) \mu_{\text{prior}}(d\theta) \\
&= \sum_{\sigma \in S_K} \int_B r(\theta, \sigma^{-1}) \pi(\theta) \alpha_{\text{swap}}(\theta, \sigma^{-1}) \delta_{\theta_{\sigma^{-1}}}(A) \mu_{\text{prior}}(d\theta) \\
&= \sum_{\sigma \in S_K} \int_B r(\theta, \sigma) \pi(\theta) \alpha_{\text{swap}}(\theta, \sigma) \delta_{\theta_\sigma}(A) \mu_{\text{prior}}(d\theta).
\end{aligned}$$

For the second term II we simply have

$$\begin{aligned}
II &= \sum_{\sigma \in S_K} \int_A r(\theta, \sigma) (1 - \alpha_{\text{swap}}(\theta, \sigma)) \delta_\theta(B) \pi(\theta) \mu_{\text{prior}}(d\theta) \\
&= \sum_{\sigma \in S_K} \int_{A \cap B} r(\theta, \sigma) (1 - \alpha_{\text{swap}}(\theta, \sigma)) \delta_\theta(B) \pi(\theta) \mu_{\text{prior}}(d\theta) \\
&= \sum_{\sigma \in S_K} \int_B r(\theta, \sigma) (1 - \alpha_{\text{swap}}(\theta, \sigma)) \delta_\theta(A) \pi(\theta) \mu_{\text{prior}}(d\theta).
\end{aligned}$$

□

This generic form of the swapping kernel provides the foundation for both PT and GPT. We describe these algorithms in the following subsections.

3.2 The Parallel Tempering case

We first show how a PT algorithm that only swaps states between the i^{th} and j^{th} components of the chain can be cast in the general framework presented above.

To that end, let $\sigma_{i,j}$ be the permutation of $(1, 2, \dots, K)$, which only permutes the i^{th} and j^{th} components, while leaving the other components invariant (i.e., such that $\sigma(i) = j$, $\sigma(j) = i$, and $\sigma(k) = k$, $k \neq i, k \neq j$). We can take $S_k = \{\sigma_{i,j}, i, j = 1, \dots, K\}$ and define the PT swapping ratio between components i and j by $r_{i,j}^{(\text{PT})} : \Theta^K \times S_K \mapsto [0, 1]$ as

$$r_{i,j}^{(\text{PT})}(\boldsymbol{\theta}, \sigma) := \begin{cases} 1 & \text{if } \sigma = \sigma_{i,j}, \\ 0 & \text{otherwise.} \end{cases}$$

Notice that this implies that $r_{i,j}^{(\text{PT})}(\boldsymbol{\theta}_\sigma, \sigma^{-1}) = r_{i,j}^{(\text{PT})}(\boldsymbol{\theta}, \sigma)$ since $\sigma_{i,j}^{-1} = \sigma_{i,j}$ and $r_{i,j}^{(\text{PT})}$ does not depend on $\boldsymbol{\theta}$, which in turn leads to the swapping acceptance probability $\alpha_{\text{swap}}^{(\text{PT})} : \Theta^K \times S_K \mapsto [0, 1]$ defined as:

$$\alpha_{\text{swap}}^{(\text{PT})}(\boldsymbol{\theta}, \sigma_{i,j}) := \min \left\{ 1, \frac{\pi(\boldsymbol{\theta}_{\sigma_{i,j}})}{\pi(\boldsymbol{\theta})} \right\}, \quad \alpha_{\text{swap}}^{(\text{PT})}(\boldsymbol{\theta}, \sigma) = 0, \quad \sigma \neq \sigma_{i,j}.$$

Thus, we can define the swapping kernel for the Parallel Tempering algorithm that swaps components i and j as follows:

Definition 3.5 (Pairwise Parallel Tempering swapping kernel). Let $\boldsymbol{\theta} \in \Theta^K$, $\sigma_{i,j} \in S_K$. We define the *Parallel Tempering swapping kernel*, which proposes to swap states between the i^{th} and j^{th} chains as $\mathbf{q}_{i,j}^{(\text{PT})} : \Theta^K \times \mathcal{B}^K \mapsto [0, 1]$ given by

$$\begin{aligned} \mathbf{q}_{i,j}^{(\text{PT})}(\boldsymbol{\theta}, B) &= \sum_{\sigma \in S_K} r_{i,j}^{(\text{PT})}(\boldsymbol{\theta}, \sigma) \left((1 - \alpha_{\text{swap}}^{(\text{PT})}(\boldsymbol{\theta}, \sigma)) \delta_{\boldsymbol{\theta}}(B) + \alpha_{\text{swap}}^{(\text{PT})}(\boldsymbol{\theta}, \sigma) \delta_{\boldsymbol{\theta}_\sigma}(B) \right) \\ &= \left(1 - \min \left\{ 1, \frac{\pi(\boldsymbol{\theta}_{\sigma_{i,j}})}{\pi(\boldsymbol{\theta})} \right\} \delta_{\boldsymbol{\theta}}(B) \right) + \min \left\{ 1, \frac{\pi(\boldsymbol{\theta}_{\sigma_{i,j}})}{\pi(\boldsymbol{\theta})} \right\} \delta_{\boldsymbol{\theta}_{\sigma_{i,j}}}(B), \quad \forall B \in \mathcal{B}^K. \end{aligned}$$

In practice, however, the PT algorithm considers various sequential swaps between chains, which can be understood by applying the composition of kernels $\mathbf{q}_{i,j}^{(\text{PT})} \mathbf{q}_{k,\ell}^{(\text{PT})} \dots$ at every swapping step. In its most common form [8, 17, 33], the PT algorithm, hereafter referred to as standard PT (which on a slight abuse of notation we will denote by PT), proposes to swap states between chains at two consecutive temperatures. Its swapping kernel $\mathbf{q}^{(\text{PT})} : \Theta^K \times \mathcal{B}^K \mapsto [0, 1]$ is given by

$$\mathbf{q}^{(\text{PT})} := \mathbf{q}_{1,2}^{(\text{PT})} \mathbf{q}_{2,3}^{(\text{PT})} \dots \mathbf{q}_{K-1,K}^{(\text{PT})}.$$

Moreover, the algorithm described in [17], proposes to swap states every $N_s \geq 1$ steps of MCMC. The complete kernel for the PT kernel is then given by [8, 17, 33]

$$\mathbf{p}^{(\text{PT})} := \mathbf{q}_{1,2}^{(\text{PT})} \mathbf{q}_{2,3}^{(\text{PT})} \dots \mathbf{q}_{K-1,K}^{(\text{PT})} \mathbf{p}^{N_s}, \quad (8)$$

where \mathbf{p} is a standard reversible Markov transition kernel used to evolve the individual chains independently. Although the kernel \mathbf{p} as well as each of the

$\mathbf{q}_{i,i+1}$ are $\boldsymbol{\mu}$ -reversible, notice that (8) does not have a palindromic structure, and as such it is not necessarily $\boldsymbol{\mu}$ -reversible. One way of making the PT algorithm reversible with respect to $\boldsymbol{\mu}$ (although not very common in practice, to the best of the authors' knowledge) is to consider the palindromic form

$$\mathbf{p}^{(\text{RPT})} := \left(\mathbf{q}_{1,2}^{(\text{PT})} \mathbf{q}_{2,3}^{(\text{PT})} \dots \mathbf{q}_{K-1,K}^{(\text{PT})} \right) \mathbf{p}^{N_s} \left(\mathbf{q}_{K,K-1}^{(\text{PT})} \dots \mathbf{q}_{3,2}^{(\text{PT})} \mathbf{q}_{2,1}^{(\text{PT})} \right).$$

3.3 Unweighted Generalized Parallel Tempering

The idea behind the Unweighted Generalized Parallel Tempering algorithm is to generalize PT so that (i) $N_s = 1$ provides a proper mixing of the chains, (ii) the algorithm is reversible with respect to $\boldsymbol{\mu}$, and (iii) the algorithm considers arbitrary sets S_K of swaps (always closed w.r.t inversion), instead of only pairwise swaps. We begin by constructing a kernel of the form (7). Let $r^{(\text{UW})} : \Theta^K \times S_K \mapsto [0, 1]$ be a function defined as

$$r^{(\text{UW})}(\boldsymbol{\theta}, \sigma) := \frac{\pi(\boldsymbol{\theta}_\sigma)}{\sum_{\sigma' \in S_K} \pi(\boldsymbol{\theta}_{\sigma'})}, \quad \boldsymbol{\theta} \in \Theta^K, \sigma \in S_K. \quad (9)$$

Clearly, (9) is a swapping ratio according to Definition 3.1. As such, given some state $\boldsymbol{\theta} \in \Theta^K$, $r^{(\text{UW})}(\boldsymbol{\theta}, \sigma)$ assigns a state-dependent probability to each of the $|S_K|$ possible permutations in S_K . A permutation $\sigma \in S_K$ is then accepted with probability $\alpha_{\text{swap}}^{(\text{UW})}(\boldsymbol{\theta}, \sigma)$, given by

$$\alpha_{\text{swap}}^{(\text{UW})}(\boldsymbol{\theta}, \sigma) := \min \left\{ 1, \frac{\pi(\boldsymbol{\theta}_\sigma) r^{(\text{UW})}(\boldsymbol{\theta}_\sigma, \sigma^{-1})}{\pi(\boldsymbol{\theta}) r^{(\text{UW})}(\boldsymbol{\theta}, \sigma)} \right\}. \quad (10)$$

Thus, we can define the swapping kernel for the UGPT algorithm, which takes the form of (7), with the particular choice of $r(\boldsymbol{\theta}, \sigma) = r^{(\text{UW})}(\boldsymbol{\theta}, \sigma)$ and $\alpha_{\text{swap}}(\boldsymbol{\theta}, \sigma) = \alpha_{\text{swap}}^{(\text{UW})}(\boldsymbol{\theta}, \sigma)$. Notice that $\alpha_{\text{swap}}^{(\text{UW})}(\boldsymbol{\theta}, \sigma) = 1, \forall \sigma \in S_K$. Indeed, if we further examine Equation (10), we see that

$$\frac{\pi(\boldsymbol{\theta}_\sigma) r^{(\text{UW})}(\boldsymbol{\theta}_\sigma, \sigma^{-1})}{\pi(\boldsymbol{\theta}) r^{(\text{UW})}(\boldsymbol{\theta}, \sigma)} = \frac{\pi(\boldsymbol{\theta}_\sigma)}{\pi(\boldsymbol{\theta})} \cdot \frac{\pi(\boldsymbol{\theta})}{\pi(\boldsymbol{\theta}_\sigma)} \cdot \frac{\sum_{\sigma'} \pi(\boldsymbol{\theta}_{\sigma'})}{\sum_{\hat{\sigma}} \pi(\boldsymbol{\theta}_{\hat{\sigma}})} = \frac{\pi(\boldsymbol{\theta}_\sigma)}{\pi(\boldsymbol{\theta})} \cdot \frac{\pi(\boldsymbol{\theta})}{\pi(\boldsymbol{\theta}_\sigma)} = 1.$$

In practice, this means that the proposed permuted state is always accepted with probability 1. The expression of the UGPT kernel then simplifies as follows.

Definition 3.6 (unweighted swapping kernel). The *unweighted swapping kernel* $\mathbf{q}^{(\text{UW})} : \Theta^K \times \mathcal{B}^K \mapsto [0, 1]$ is defined as

$$\mathbf{q}^{(\text{UW})}(\boldsymbol{\theta}, B) = \sum_{\sigma \in S_K} r^{(\text{UW})}(\boldsymbol{\theta}, \sigma) \delta_{\boldsymbol{\theta}_\sigma}(B), \quad \forall \boldsymbol{\theta} \in \Theta^K, B \in \mathcal{B}^K.$$

Applying this swapping kernel successively with the kernel $\mathbf{p} = p_1 \times p_2 \times \dots \times p_K$ in the order $\mathbf{q}^{(\text{UW})} \mathbf{p} \mathbf{q}^{(\text{UW})} =: \mathbf{p}^{(\text{UW})}$ gives what we call *Unweighted Generalized Parallel Tempering kernel* $\mathbf{p}^{(\text{UW})}$. Lastly, we write the UGPT in operator form as

$$\mathbf{P}^{(\text{UW})} := \mathbf{Q}^{(\text{UW})} \mathbf{P} \mathbf{Q}^{(\text{UW})},$$

where \mathbf{P} and $\mathbf{Q}^{(\text{UW})}$ are the Markov operators corresponding to the kernels \mathbf{p} and $\mathbf{q}^{(\text{UW})}$, respectively. We now investigate the reversibility of the UGPT kernel. We start with a rather straightforward result.

Proposition 3.7. Suppose that, for any $k = 1, 2, \dots, K$, p_k is μ_k -reversible. Then, $\mathbf{p} = p_1 \times \dots \times p_K$ is reversible with respect to $\boldsymbol{\mu}$.

Proof. We prove reversibility by confirming that equation (1) holds true. To that end, let $\boldsymbol{\theta} \in \Theta^K$, $A, B \in \mathcal{B}^K$, where A and B tensorize, i.e., $A := \prod_{k=1}^K A_k$ and $B := \prod_{k=1}^K B_k$, with $A_1, \dots, A_K, B_1, \dots, B_K \in \mathcal{B}(\Theta)$. Then,

$$\begin{aligned} \int_A \boldsymbol{\pi}(\boldsymbol{\theta}) \mathbf{p}(\boldsymbol{\theta}, B) d\boldsymbol{\theta} &= \prod_{k=1}^K \int_{A_k} \pi(\theta_k) p(\theta_k, B_k) d\theta_k \\ &= \prod_{k=1}^K \int_{B_k} \pi(\theta_k) p(\theta_k, A_k) d\theta_k = \int_B \boldsymbol{\pi}(\boldsymbol{\theta}) \mathbf{p}(\boldsymbol{\theta}, A) d\boldsymbol{\theta}. \end{aligned}$$

Showing that the previous equality holds for sets A, B that tensorize is indeed sufficient to show that the claim holds for any $A, B \in \mathcal{B}^K$. This follows from Carathéodory's Extension Theorem applied as in the proof of uniqueness of product measures; see [2, §1.3.10, 2.6.3], for details. \square

We can now prove the reversibility of the chain generated by $\mathbf{p}^{(\text{UW})}$.

Proposition 3.8 (Reversibility of the UGPT chain). Suppose that, for any $k = 1, 2, \dots, K$, p_k is μ_k -reversible. Then, the Markov chain generated by $\mathbf{p}^{(\text{UW})}$ is $\boldsymbol{\mu}$ -reversible.

Proof. It follows from proposition 3.4 and 3.7 that the kernels $\mathbf{q}^{(\text{UW})}$ and \mathbf{p} are $\boldsymbol{\mu}$ -reversible. Furthermore, since $\mathbf{p}^{(\text{UW})}$ is a *palindromic* composition of kernels, each of which is reversible with respect to $\boldsymbol{\mu}$, then, $\mathbf{p}^{(\text{UW})}$ is reversible with respect to $\boldsymbol{\mu}$ [8]. \square

The UGPT algorithm proceeds by iteratively applying the kernel $\mathbf{p}^{(\text{UW})}$ to a predefined initial state. In particular, states are updated using the procedure outlined in Algorithm 2.

Remark 3.9. In practice, one does not need to perform $|S_K|$ posterior evaluations when computing $r^{(\text{UW})}(\boldsymbol{\theta}^n, \cdot)$, rather “just” K of them. Indeed, since $\pi_j(\theta_k^n) \propto \pi(\theta_k^n)^{T_j}$, $k, j = 1, 2, \dots, K$, we just need to store the values of $\pi(\theta_k^n)$, $k = 1, 2, \dots, K$, for a fixed n , and then permute over the temperature indices.

Let now $\mathcal{Q} : \Theta \mapsto \mathbb{R}$ be a quantity of interest. The posterior mean of \mathcal{Q} , $\mu(\mathcal{Q}) := \mu_1(\mathcal{Q})$ is approximated using $N \in \mathbb{N}$ samples by the following ergodic estimator $\hat{\mathcal{Q}}_{(\text{UW})}$:

$$\mu(\mathcal{Q}) \approx \hat{\mathcal{Q}}_{(\text{UW})} = \frac{1}{N-b} \sum_{n=b}^N \mathcal{Q}(\theta_1^{(n)}).$$

Algorithm 2 Unweighted Generalized Parallel Tempering.

```

function GENERALIZED PARALLEL TEMPERING( $\mathbf{p}, N, \nu$ )
  Sample  $\theta^{(1)} \sim \nu$ 
  for  $n = 1, 2, \dots, N - 1$  do
    Sample  $\theta_\sigma^{(n)} \sim \mathbf{q}^{(\text{UW})}(\theta^{(n)}, \cdot)$  ▷ first swapping kernel
    Sample  $\mathbf{z}^{(n+1)} \sim \mathbf{p}(\theta_\sigma^{(n)}, \cdot)$  ▷ Markov transition kernel  $\mathbf{p}$  kernel
    Sample  $\theta^{(n+1)} \sim \mathbf{q}^{(\text{UW})}(\mathbf{z}^{(n+1)}, \cdot)$  ▷ second swapping kernel
  end for
  Output  $\{\theta_1^{(n)}\}_{n=1}^N$ 
end function

```

3.3.1 A comment on the pairwise state-dependent PT method of [24]

The work [24] presents a similar state-dependent swapping. We will refer to the method presented therein as Pairwise State Dependent Parallel Tempering (PSDPT). Such a method, however, differs from UGPT from the fact that (i) only pairwise swaps are considered and (ii) it is not rejection free. We summarize such a method for the sake of completeness. Let $S_{K,\text{pairwise}}$ denote the group of pairwise permutations of $(1, 2, \dots, K)$. Given a current state $\theta \in \Theta^K$, the PSDPT algorithm samples a pairwise permutation $\theta_{\sigma_{i,j}} \in S_{K,\text{pairwise}}$ with probability $r_{i,j}^{(\text{PSDPT})}(\theta, \sigma_{i,j})$ given by

$$r_{i,j}^{(\text{PSDPT})}(\theta, \sigma_{i,j}) := \frac{\exp(-|\Phi(\theta_i, y) - \Phi(\theta_j; y)|)}{\sum_{k,l} \exp(-|\Phi(\theta_k, y) - \Phi(\theta_l; y)|)},$$

and then accepts this swap with probability

$$\alpha_{\text{swap}}^{(\text{PSDPT})}(\theta, \sigma_{i,j}) := \min \left\{ 1, \left(\frac{\pi_1(\theta_i)}{\pi_1(\theta_j)} \right)^{\frac{1}{T_j} - \frac{1}{T_i}} \right\}.$$

This method is attractive from an implementation point of view in the sense that it promotes pairwise swaps that have a similar *energy*, and as such, are *likely* (yet not guaranteed) to get accepted. In contrast, UGPT *always* accepts the new proposed state, which in turn leads to a larger amount of *global* moves, thus providing a more efficient algorithm. This is verified on the numerical experiments.

3.4 Weighted Generalized Parallel Tempering

Following the intuition of the continuous-time Infinite Swapping approach of [16, 35], we propose a second discrete-time algorithm, which we will refer to as *Weighted Generalized Parallel Tempering* (WGPT). The idea behind this method is to swap the dynamics of the process, that is, the Markov kernels

and temperatures, instead of swapping the states such that any given swap is accepted with probability 1. We will see that the Markov kernel obtained when swapping the dynamics is not invariant with respect to the product measure of interest $\boldsymbol{\mu}$; therefore, an importance sampling step is needed when computing posterior expectations.

For a given permutation $\sigma \in S_K$, we define the *swapped Markov kernel* $\mathbf{p}_\sigma : \Theta^K \times \mathcal{B}^K \mapsto [0, 1]$ and the *swapped product posterior measure* $\boldsymbol{\mu}_\sigma$ (on the measurable space $(\Theta^K, \mathcal{B}^K)$) as:

$$\begin{aligned}\mathbf{p}_\sigma(\boldsymbol{\theta}, \cdot) &= p_{\sigma(1)}(\theta_1, \cdot) \times \cdots \times p_{\sigma(K)}(\theta_K, \cdot), \\ \boldsymbol{\mu}_\sigma &:= \mu_{\sigma(1)} \times \cdots \times \mu_{\sigma(K)},\end{aligned}$$

where the swapped posterior measure has a density with respect to $\boldsymbol{\mu}_{\text{prior}}$ given by

$$\pi_\sigma(\boldsymbol{\theta}) := \pi_{\sigma(1)}(\theta_1) \times \cdots \times \pi_{\sigma(K)}(\theta_K), \quad \boldsymbol{\theta} \in \Theta^K, \sigma \in S_K \quad (11)$$

Moreover, we define the swapping weights

$$w_\sigma(\boldsymbol{\theta}) := \frac{\pi_\sigma(\boldsymbol{\theta})}{\sum_{\sigma' \in S_K} \pi_{\sigma'}(\boldsymbol{\theta})}, \quad \boldsymbol{\theta} \in \Theta^K, \sigma \in S_K. \quad (12)$$

Note that, in general, $\pi_\sigma(\boldsymbol{\theta}) \neq \pi(\boldsymbol{\theta}_\sigma)$, and as such, $w_\sigma(\boldsymbol{\theta}) \neq r^{(\text{UW})}(\boldsymbol{\theta}, \sigma)$, with w_σ defined as in (12).

Definition 3.10. We define the *Weighted Generalized Parallel Tempering* kernel $\mathbf{p}^{(\text{W})} : \Theta^K \times \mathcal{B}^K \mapsto [0, 1]$ as the following state-dependent, convex combination of kernels:

$$\mathbf{p}^{(\text{W})}(\boldsymbol{\theta}, \cdot) := \sum_{\sigma \in S_K} w_\sigma(\boldsymbol{\theta}) \mathbf{p}_\sigma(\boldsymbol{\theta}, \cdot), \quad \boldsymbol{\theta} \in \Theta^K, \sigma \in S_K.$$

Thus, the WGPT chain is obtained by iteratively applying $\mathbf{p}^{(\text{W})}$. We show in Proposition 3.11 that the resulting Markov chain has invariant measure

$$\boldsymbol{\mu}_\text{W} = \frac{1}{|S_K|} \sum_{\sigma \in S_K} \boldsymbol{\mu}_\sigma = \tilde{\boldsymbol{\mu}} \times \cdots \times \tilde{\boldsymbol{\mu}},$$

with $\tilde{\boldsymbol{\mu}} = \frac{1}{|S_K|} \sum_{\sigma} \mu_\sigma$, i.e., the average with tensorization. Furthermore, $\boldsymbol{\mu}_\text{W}$ has a density (w.r.t the prior $\boldsymbol{\mu}^0$) given by

$$\pi_\text{W}(\boldsymbol{\theta}) = \frac{1}{|S_K|} \sum_{\sigma \in S_K} \pi_\sigma(\boldsymbol{\theta}), \quad \boldsymbol{\theta} \in \Theta^K,$$

and a similar average and then tensorization representation applies to π_W . We now proceed to show that $\mathbf{p}^{(\text{W})}(\boldsymbol{\theta}, \cdot)$ is $\boldsymbol{\mu}_\text{W}$ -reversible (hence $\boldsymbol{\mu}_\text{W}$ -invariant).

Proposition 3.11 (Reversibility of the WGPT chain). Suppose that, for any $k = 1, 2, \dots, K$ p_k is μ_k -reversible. Then, the Markov chain generated by $\mathbf{p}^{(\text{W})}$ is $\boldsymbol{\mu}_\text{W}$ -reversible.

Proof. We show reversibility by showing that (1) holds true. Thus, for $\theta \in \Theta^K$, $A, B \in \mathcal{B}^K$, with $A := A_1 \times \cdots \times A_K$, $A_k \in \mathcal{B}(\Theta)$, and with B_k defined in a similar way, we have that:

$$\begin{aligned}
\int_A \mathbf{p}^{(W)}(\theta, B) \pi_W(\theta) d\theta &= \int_A \left[\sum_{\sigma \in S_K} w_\sigma(\theta) \mathbf{p}_\sigma(\theta, B) \right] \frac{\sum_{\rho \in S_K} \pi_\rho(\theta)}{|S_K|} \mu_{\text{prior}}(d\theta) \\
&= \int_A \left[\sum_{\sigma \in S_K} \frac{\pi_\sigma(\theta)}{\sum_{\sigma' \in S_K} \pi_{\sigma'}(\theta)} \mathbf{p}_\sigma(\theta, B) \right] \frac{\sum_{\rho \in S_K} \pi_\rho(\theta)}{|S_K|} \mu_{\text{prior}}(d\theta) \\
&= \frac{1}{|S_K|} \sum_{\sigma \in S_K} \int_A \pi_\sigma(\theta) \mathbf{p}_\sigma(\theta, B) \mu_{\text{prior}}(d\theta) \\
&= \frac{1}{|S_K|} \sum_{\sigma \in S_K} \int_B \pi_\sigma(\theta) \mathbf{p}_\sigma(\theta, A) \mu_{\text{prior}}(d\theta) \quad (\text{by Proposition 3.7}) \\
&= \frac{1}{|S_K|} \sum_{\sigma \in S_K} \int_B \frac{\pi_\sigma(\theta)}{\sum_{\sigma' \in S_K} \pi_{\sigma'}(\theta)} \mathbf{p}_\sigma(\theta, A) \sum_{\rho \in S_K} \pi_\rho(\theta) \mu_{\text{prior}}(d\theta) \\
&= \sum_{\sigma \in S_K} \int_B w_\sigma(\theta) \mathbf{p}_\sigma(\theta, A) \pi_W(\theta) \mu_{\text{prior}}(d\theta) \\
&= \int_B \mathbf{p}^{(W)}(\theta, A) \pi_W(\theta) \mu_{\text{prior}}(d\theta).
\end{aligned}$$

where once again, in light of Carathéodory's Extension Theorem, it is sufficient to show that reversibility holds for sets that tensorize. \square

We remark that the measure μ_W is not of interest per se. However, we can use importance sampling to compute posterior expectations. Let $\mathcal{Q}(\theta) := \mathcal{Q}(\theta_1)$ be a μ -integrable quantity of interest. We can write

$$\mathbb{E}_{\mu_1}[\mathcal{Q}] = \mathbb{E}_\mu[\mathcal{Q}(\theta_1)] = \mathbb{E}_{\mu_W} \left[\mathcal{Q}(\theta_1) \frac{\pi(\theta)}{\pi_W(\theta)} \right] = \frac{1}{|S_K|} \sum_{\sigma \in S_K} \mathbb{E}_{\mu_W} \left[\mathcal{Q}(\theta_{\sigma(1)}) \frac{\pi(\theta_\sigma)}{\pi_W(\theta_\sigma)} \right].$$

The last equality can be justified since μ_W is invariant by permutation of coordinates. Thus, we can define the following (weighted) ergodic estimator $\hat{\mathcal{Q}}_{(W)}$ of the posterior mean of a quantity of interest \mathcal{Q} by

$$\begin{aligned}
\mu(\mathcal{Q}) \approx \hat{\mathcal{Q}}_{(W)} &= \frac{1}{|S_K|} \frac{1}{N} \sum_{\sigma \in S_K} \sum_{n=1}^N \frac{\pi(\theta_\sigma^{(n)})}{\pi_W(\theta_\sigma^{(n)})} \mathcal{Q}(\theta_{\sigma(1)}^{(n)}) \\
&= \frac{1}{|S_K|} \frac{1}{N} \sum_{\sigma \in S_K} \sum_{n=1}^N \hat{w}(\theta^{(n)}, \sigma) \mathcal{Q}(\theta_{\sigma(1)}^{(n)}), \tag{13}
\end{aligned}$$

where we have denoted the importance sampling weights by $\hat{w}(\theta, \sigma) := \frac{\pi(\theta_\sigma)}{\pi_W(\theta_\sigma)}$ and where N is the number of samples in the chain. Notice that $w(\theta, \sigma) =$

$\hat{w}(\theta, \sigma^{-1})$. As a result, the WGPT algorithm produces an estimator based on NK weighted samples, rather than “just” N , at the same computational cost of UGPT. Thus, the previous estimator evaluates the quantity of interest \mathcal{Q} not only in the points $\mathcal{Q}(\theta_1^{(n)})$, but also in all states of the parallel chains, $\mathcal{Q}(\theta_{\sigma(1)}^{(n)})$ for all $\sigma \in S_K$, namely $\mathcal{Q}(\theta_k^{(n)})$, $k = 1, 2, \dots, K$. The Weighted Generalized Parallel Tempering procedure is shown in Algorithm 3. To reiterate, we remark that sampling from $\mathbf{p}_\sigma(\theta^{(n)}, \cdot)$ involves a swap of dynamics, i.e., kernels and temperatures.

Algorithm 3 Weighted Generalized Parallel Tempering.

function WEIGHTED GENERALIZED PARALLEL TEMPERING(\mathbf{p}, N, ν)
 Sample $\theta^{(1)} \sim \nu$
 for $n = 1, 2, \dots, N - 1$ **do**
 Sample $\sigma \sim \{w_{\sigma'}(\theta^n)\}_{\sigma' \in S_K}$ ▷ sample the permutation σ with
 probability $w_\sigma(\theta^n)$
 Sample $\theta^{(n+1)} \sim \mathbf{p}_\sigma(\theta^{(n)}, \cdot)$ ▷ Sample state with the swapped Markov
 kernel
 end for
 Output $\{\theta^{(n)}\}_{n=1}^N, \{\{w_{\sigma'}(\theta^n)\}_{\sigma' \in S_K}\}_{n=1}^N$.
end function

Just as in Remark 3.9, one only needs to evaluate the posterior K times (instead of $|S_K|$) to compute $w_{(\cdot)}(\theta^n)$.

4 Ergodicity of Generalized Parallel Tempering

4.1 Preliminaries

We assume that the chains generated by the MCMC kernels p_k , for $k = 1, \dots, K$, are aperiodic, μ_k -irreducible [3], and have invariant measure μ_k on the measurable space $(\Theta, \mathcal{B}(\Theta))$. Let $r \in [1, \infty)$ and $\mu \in \mathcal{M}(\Theta)$ be a “reference” probability measure. On a BIP setting, this reference measure is considered to be the posterior. We define the following spaces

$$L_r = L_r(\Theta, \mu) = \left\{ f : \Theta \mapsto \mathbb{R}, \mu\text{-measurable, s.t. } \|f\|_r^r := \int |f(\theta)|^r \mu(d\theta) < \infty \right\},$$

$$L_r^0 = L_r(\Theta, \mu) = \left\{ f \in L_r(\Theta, \mu), \text{ s.t. } \mu(f) := \int_\Theta f(\theta) \mu(d\theta) = 0 \right\}.$$

Moreover, when $r = \infty$, we define

$$L_\infty(\Theta, \mu) := \left\{ f : \Theta \mapsto \mathbb{R}, \text{ s.t. } \inf_{\substack{\mu(B)=0 \\ B \in \mathcal{B}(\Theta)}} \sup_{y \in \Theta \setminus B} |f(y)| < \infty \right\}.$$

Notice that, clearly, $L_r^0(\Theta, \mu) \subset L_r(\Theta, \mu)$. In addition we define the spaces of measures

$$\mathcal{M}_r(\Theta, \mu) := \{\nu \in \mathcal{M}(\Theta) \text{ s.t. } \nu \ll \mu, \|\nu\|_{L_r(\Theta, \mu)} < \infty\},$$

where $\|\nu\|_{L_r(\Theta, \mu)} := \left\| \frac{d\nu}{d\mu} \right\|_{L_r(\Theta, \mu)}.$

Notice that the definition of L_r -norm depends on the reference measure μ , and on Θ .

A Markov operator $P : L_r(\Theta, \mu) \mapsto L_r(\Theta, \mu)$ with invariant measure μ is a bounded linear operator whose norm is given by

$$\|P\|_{L_r(\Theta, \mu) \mapsto L_r(\Theta, \mu)} := \sup_{\|f\|_{L_r(\Theta, \mu)}=1} \|Pf\|_{L_r(\Theta, \mu)}, \quad f \in L_r(\Theta, \mu).$$

It is well-known (see, e.g., [38]) that any Markov operator P on $L_r(\Theta, \mu)$ with invariant measure μ can be understood as a weak contraction in $L_r(\Theta, \mu)$, i.e., $\|P\|_{L_r(\Theta, \mu) \mapsto L_r(\Theta, \mu)} \leq 1$. To quantify the convergence of a Markov chains generated by a Markov operator P , we define the concept of geometric ergodicity. Let $r \in [1, \infty]$. A Markov operator P with invariant measure $\mu \in \mathcal{M}(\Theta)$ is said to be $L_r(\Theta, \mu)$ -geometrically ergodic if for all probability measures $\nu \in \mathcal{M}_r(\Theta, \mu)$ there exists an $\alpha \in (0, 1)$ and $C_\nu < \infty$ such that

$$\|\nu P^n - \mu\|_{L_r(\Theta, \mu)} \leq C_\nu \alpha^n, \quad n \in \mathbb{N}. \quad (14)$$

A related concept to L_2 -geometric ergodicity is that of L_2 -spectral gap. A Markov operator $P : L_2(\Theta, \mu) \mapsto L_2(\Theta, \mu)$ with invariant measure $\mu \in \mathcal{M}(\Theta)$ has an $L_2(\Theta, \mu)$ -spectral gap $1 - \beta > 0$, with $\beta < 1$, if the following holds

$$\|P\|_{L_2^0(\Theta, \mu) \mapsto L_2^0(\Theta, \mu)} \leq \beta. \quad (15)$$

The next Proposition, whose proof can be found e.g., in [38], relates the existence of an L_2 -spectral gap to the geometric ergodicity of the chain (with $\beta \leq \alpha$, in general).

Proposition 4.1. Let $P : L_2(\Theta, \mu) \mapsto L_2(\Theta, \mu)$ be a μ -reversible Markov transition operator. The existence of an $L_2(\Theta, \mu)$ -spectral gap implies $L_r(\Theta, \mu)$ -geometric ergodicity for any $r \in [1, \infty]$.

Proof. The previous claim is shown in [38, Proposition 3.17 and Appendix A.4]. It is also shown in [38] that, in general, $\beta \leq \alpha$, with α, β given as in Equations (14) and (15). \square

Our path to prove ergodicity of the GPT algorithms will be to show the existence of an L_2 -spectral gap.

4.2 Geometric ergodicity and L_2 -spectral gap for GPT

The main results of this section are presented in Theorem 4.5 and Theorem 4.7, which show the existence of an L_2 -spectral gap for both the UGPT and WGPT algorithms, respectively.

We begin with the definition of overlap between two probability measures. Such a concept will later be used to bound the spectral gap of the GPT algorithms.

Definition 4.2 (Density overlap). Let μ_k, μ_j be two probability measures on the measurable space $(\Theta, \mathcal{B}(\Theta))$, each having respective densities $\pi_k(\theta), \pi_j(\theta)$, $\theta \in \Theta$, with respect to some common reference measure ν_Θ also on $(\Theta, \mathcal{B}(\Theta))$. We define the *overlap* between $\pi_k(\theta)$ and $\pi_j(\theta)$ as

$$\eta_{\nu_\Theta}(\pi_k, \pi_j) := \int_{\Theta} \min\{\pi_k(\theta), \pi_j(\theta)\} \nu_\Theta(d\theta) = 1 - \frac{1}{2} \|\mu_k - \mu_j\|_{L_1(\Theta, \nu_\Theta)}.$$

An analogous definition holds for π_σ, π_ρ , with $\rho, \sigma \in S_K$.

Assumption 4.3. For $k = 1, \dots, K$, let $\mu_k \in \mathcal{M}_1(\Theta, \mu_{\text{prior}})$ be given as in (5), $p_k : \Theta \times \mathcal{B}(\Theta) \mapsto [0, 1]$ be the Markov kernel associated to the k^{th} dynamics and let $P_k : L_r(\Theta, \mu_k) \mapsto L_r(\Theta, \mu_k)$ be its corresponding μ_k -invariant Markov operator. In addition, for $\sigma, \rho \in S_K$, define the measures $\mu_\sigma, \mu_\rho \in \mathcal{M}(\Theta^K)$ as in Equation (6). Throughout this work it is assumed that:

- C1. The Markov kernel p_k is μ_k -reversible.
- C2. The Markov operator P_k has an $L_2(\Theta, \mu_k)$ -spectral gap.
- C3. For any $\sigma, \rho \in S_K$, $\Lambda_{\sigma, \rho} := \eta_{\mu_{\text{prior}}}(\pi_\sigma, \pi_\rho) > 0$, with π_σ, π_ρ defined as in (11).

These assumptions are relatively mild. In particular, C1 and C2 are known to hold for many commonly-used Markov transition kernels, such as RWM, Metropolis-adjusted Langevin Algorithm, Hamiltonian Monte Carlo, (generalized) preconditioned Crank-Nicolson, among others, under mild regularity conditions on π [3, 20]. Assumption C3 holds true given the construction of the product measures in Section 3.

We now present an auxiliary result that we will use to bound the spectral gap of both the Weighted and Unweighted GPT algorithms.

Proposition 4.4. Suppose that Assumption 4.3 holds and let $\mathbf{P} := \bigotimes_{k=1}^K P_k : L_2(\Theta^K, \mu) \mapsto L_2(\Theta^K, \mu)$, with invariant measure $\mu = \mu_1 \times \dots \times \mu_K$. Then, \mathbf{P} has an $L_2(\Theta^K, \mu)$ -spectral gap, i.e., $\|\mathbf{P}\|_{L_2^0(\Theta^K, \mu) \mapsto L_2^0(\Theta^K, \mu)} < 1$. Moreover, the Markov chain obtained from \mathbf{P} is $L_r(\Theta^K, \mu)$ -geometrically ergodic, for any $r \in [1, \infty]$.

Proof. We limit ourselves to the case $K = 2$, since the case for $K > 2$ follows by induction. Denote by $I : L_2(\Theta, \mu_k) \mapsto L_2(\Theta, \mu_k)$, $k = 1, 2$ the identity Markov transition operator, and let $f \in L_2(\Theta^2, \mu)$. Notice that f admits a spectral

representation in $L_2(\Theta^2, \boldsymbol{\mu})$ given by $f(\boldsymbol{\theta}) = \sum_{k,j} \phi_k(\theta_1) \psi_j(\theta_2) c_{k,j}$, with $c_{k,j} \in \mathbb{R}$, and where $\{\phi_k\}_{k \in \mathbb{N}}$ is a complete orthonormal basis (CONB) of $L_2(\Theta, \mu_1)$ and $\{\psi_j\}_{j \in \mathbb{N}}$ is a CONB of $L_2(\Theta, \mu_2)$, so that $\{\phi_k \otimes \psi_j\}_{k,j \in \mathbb{N}}$ is a CONB of $L_2(\Theta^2, \boldsymbol{\mu})$. Moreover, we assume that $\phi_0 = \psi_0 = 1$, and write, for notational simplicity $\|P_1\| = \|P_1\|_{L_2(\Theta, \mu_1) \mapsto L_2(\Theta, \mu_1)}$, and $\|P_2\| = \|P_2\|_{L_2(\Theta, \mu_2) \mapsto L_2(\Theta, \mu_2)}$. Lastly, denote $f_0 = f - c_{0,0}$, so that $f_0 \in L_2^0(\Theta^2, \boldsymbol{\mu})$. Notice that

$$\begin{aligned}
\|(P_1 \otimes I)f_0\|_{L_2(\Theta^2, \boldsymbol{\mu})}^2 &= \left\| \sum_{(k,j) \neq (0,0)} (P_1 \phi_k) \psi_j c_{k,j} \right\|_{L_2(\Theta^2, \boldsymbol{\mu})}^2 \\
&= \left\| \sum_{j=0}^{\infty} \left(\sum_{k=1}^{\infty} P_1 \phi_k c_{k,j} \right) \psi_j + \sum_{j=1}^{\infty} c_{0,j} P_1 \phi_0 \psi_j \right\|_{L_2(\Theta^2, \boldsymbol{\mu})}^2 \\
&= \sum_{j=1}^{\infty} \left\| \sum_{k=1}^{\infty} P_1 \phi_k c_{k,j} + c_{0,j} P_1 \phi_0 \right\|_{L_2(\Theta, \mu_1)}^2 + \left\| \sum_{k=1}^{\infty} P_1 \phi_k c_{k,0} \right\|_{L_2(\Theta, \mu_1)}^2 \\
&= \sum_{j=1}^{\infty} \left\| P_1 \left(\sum_{k=1}^{\infty} \phi_k c_{k,j} \right) \right\|_{L_2(\Theta, \mu_1)}^2 + \sum_{j=1}^{\infty} \|c_{0,j} \phi_0\|_{L_2(\Theta, \mu_1)}^2 + \left\| P_1 \left(\sum_{k=1}^{\infty} \phi_k c_{k,0} \right) \right\|_{L_2(\Theta, \mu_1)}^2 \\
&\leq \sum_{j=1}^{\infty} \left(\|P_1\|^2 \sum_{k=1}^{\infty} c_{k,j}^2 + c_{0,j}^2 \right) + \|P_1\|^2 \sum_{k=1}^{\infty} c_{k,0}^2 \\
&= \|P_1\|^2 \|f_0\|_{L_2(\Theta^2, \boldsymbol{\mu})}^2 + (1 - \|P_1\|^2) \sum_{j=1}^{\infty} (c_{0,j})^2.
\end{aligned}$$

Proceeding similarly, we can obtain an equivalent bound for $\|(I \otimes P_2)f_0\|_{L_2(\Theta^2, \boldsymbol{\mu})}^2$.

We are now ready to bound $\|\mathbf{P}\|_{L_2(\Theta^2, \boldsymbol{\mu}) \mapsto L_2(\Theta^2, \boldsymbol{\mu})}^2$:

$$\begin{aligned}
\|\mathbf{P}f_0\|_{L_2(\Theta^2, \boldsymbol{\mu})}^2 &= \|(P_1 \otimes P_2)f_0\|_{L_2(\Theta^2, \boldsymbol{\mu})}^2 = \|(P_1 \otimes I)(I \otimes P_2)f_0\|_{L_2(\Theta^2, \boldsymbol{\mu})}^2 \\
&\leq \|P_1\|^2 \|(I \otimes P_2)f_0\|_{L_2(\Theta^2, \boldsymbol{\mu})}^2 + (1 - \|P_1\|^2) \left(\sum_{j=1}^{\infty} \left((I \otimes P_2) \sum_{(\ell,k) \neq (0,0)} c_{\ell,k} \phi_{\ell} \psi_k, \phi_0 \psi_j \right) \right)^2
\end{aligned}$$

$$\begin{aligned}
&= \|P_1\|^2 \|(I \otimes P_2)f_0\|_{L_2(\Theta^2, \mu)}^2 + (1 - \|P_1\|^2) \left(\sum_{j=1}^{\infty} \left(\sum_{k=1}^{\infty} c_{0,k}(P_2\psi_k), \psi_j \right) \right)^2 \\
&\leq \|P_1\|^2 \|(I \otimes P_2)f_0\|_{L_2(\Theta^2, \mu)}^2 + (1 - \|P_1\|^2) \left\| P_2 \left(\sum_{k=1}^{\infty} c_{0,k}\psi_k \right) \right\|_{L_2(\Theta, \mu_2)}^2 \\
&\leq \|P_1\|^2 \|P_2\|^2 \|f_0\|_{L_2(\Theta^2, \mu)}^2 + \|P_1\|^2 (1 - \|P_2\|^2) \left(\sum_{j=1}^{\infty} c_{j,0}^2 \right) + (1 - \|P_1\|^2) \|P_2\|^2 \left(\sum_{k=1}^{\infty} c_{0,k}^2 \right)
\end{aligned}$$

Assuming without loss of generality that $\|P_1\| \geq \|P_2\|$, we can use the inequality above to bound

$$\begin{aligned}
\|P f_0\|_{L_2(\Theta^2, \mu)}^2 &\leq \|P_1\|^2 \|P_2\|^2 \|f_0\|_{L_2(\Theta^2, \mu)}^2 + \|P_1\|^2 (1 - \|P_2\|^2) \underbrace{\left(\sum_{j=1}^{\infty} c_{j,0}^2 + \sum_{k=1}^{\infty} c_{0,k}^2 \right)}_{\leq \|f_0\|_{L_2(\Theta^2, \mu)}^2} \\
&\leq \|P_1\|^2 \|f_0\|_{L_2(\Theta^2, \mu)}^2.
\end{aligned}$$

Thus, we have that $\|P\|_{L_2^0(\Theta^2, \mu) \rightarrow L_2^0(\Theta^2, \mu)} \leq \max_{k=1,2} \{\|P_k\|_{L_2^0(\Theta, \mu_k) \rightarrow L_2^0(\Theta, \mu_k)}\} < 1$. The previous result can easily be extended to $K > 2$. Lastly, $L_r(\Theta^K, \mu)$ -geometric ergodicity $\forall r \in [1, \infty]$ follows from proposition 4.1. \square

We can use the previous result to prove the geometric ergodicity of the UGPT algorithm:

Theorem 4.5 (Ergodicity of UGPT). *Suppose Assumption 4.3 holds and denote by μ the invariant measure of the UGPT Markov operator $\mathbf{P}^{(\text{UW})}$. Then, $\mathbf{P}^{(\text{UW})}$ has an $L_2(\Theta^K, \mu)$ -spectral gap. Moreover, the chain generated by $\mathbf{P}^{(\text{UW})}$ is $L_r(\Theta^K, \mu)$ -geometrically ergodic for any $r \in [1, \infty]$.*

Proof. Recall that $\mathbf{P}^{(\text{UW})} := \mathbf{Q}^{(\text{UW})} \mathbf{P} \mathbf{Q}^{(\text{UW})}$. From the definition of operator norm, we have that

$$\begin{aligned}
\|\mathbf{P}^{(\text{UW})}\|_{L_2^0(\Theta^K, \mu) \rightarrow L_2^0(\Theta^K, \mu)} &\leq \|\mathbf{Q}^{(\text{UW})}\|_{L_2^0(\Theta^K, \mu) \rightarrow L_2^0(\Theta^K, \mu)}^2 \|\mathbf{P}\|_{L_2^0(\Theta^K, \mu) \rightarrow L_2^0(\Theta^K, \mu)} \\
&\leq \|\mathbf{P}\|_{L_2^0(\Theta^K, \mu) \rightarrow L_2^0(\Theta^K, \mu)} < 1,
\end{aligned}$$

where the previous line follows from Proposition 4.4 and the fact that $\mathbf{Q}^{(\text{UW})}$ is a weak contraction in $L_2(\Theta^K, \mu)$ (see, e.g., [4, Proposition 1]). Lastly, $L_r(\Theta^K, \mu)$ -geometric ergodicity $\forall r \in [1, \infty]$ follows from proposition 4.1 and the fact that $\mathbf{P}^{(\text{UW})}$ is μ -reversible by Proposition 3.8. \square

We now turn to proving geometric ergodicity for the WGPT algorithm. We begin with an auxiliary result, lower-bounding the variance of a μ_W -integrable functional $f \in L_2(\Theta^K, \mu_W)$.

Proposition 4.6. Let $f \in L_2^0(\Theta^K, \mu_W)$ be a μ_W -integrable function such that $\|f\|_{L_2(\Theta^K, \mu_W)} = 1$, and denote by $\mathbb{V}_{\mu_W}[f]$, $\mathbb{V}_{\mu_\sigma}[f]$ the variance of f with respect to μ_W, μ_σ , respectively with $\sigma \in S_K$. In addition, suppose Assumption 4.3 holds. Then, it can be shown that

$$0 < \frac{\Lambda_m}{2 - \Lambda_m} \leq \frac{1}{|S_K|} \sum_{\sigma \in S_K} \mathbb{V}_{\mu_\sigma}[f] \leq \mathbb{V}_{\mu_W}[f] = 1,$$

with $\Lambda_m = \min_{\sigma, \rho \in S_K} \{\Lambda_{\sigma, \rho}\}$ and $\Lambda_{\sigma, \rho}$ as in Assumption C3.

Proof. The proof is technical and tedious and is proposed in the Appendix 1. \square

We are finally able to prove the ergodicity of the WGPT algorithm.

Theorem 4.7 (Ergodicity of WGPT). *Suppose Assumption 4.3 holds for some $r \in [1, \infty]$ and denote by μ_W the invariant measure of the WGPT Markov operator $\mathbf{P}^{(W)}$. Then, $\mathbf{P}^{(W)}$ has an $L_2(\Theta^K, \mu_W)$ -spectral gap. Moreover, the chain generated by $\mathbf{P}^{(W)}$ is $L_r(\Theta^K, \mu_W)$ -geometrically ergodic for any $r \in [1, \infty]$.*

Proof. Let $\mathcal{L} := \{f \in L_2^0(\Theta^K, \mu_W) : \|f\|_{L_2^0(\Theta^K, \mu_W)} = 1\}$. Then, from the definition of operator norm,

$$\begin{aligned} \left\| \mathbf{P}^{(W)} \right\|_{L_2^0(\Theta^K, \mu_W) \rightarrow L_2^0(\Theta^K, \mu_W)}^2 &= \sup_{f \in \mathcal{L}} \left\| \mathbf{P}^{(W)} f \right\|_{L_2(\Theta^K, \mu_W)}^2 \\ &= \sup_{f \in \mathcal{L}} \int_{\Theta^K} \left| \sum_{\sigma \in S_K} w_\sigma(\theta) \int_{\Theta^K} f(y) \mathbf{p}_\sigma(\theta, dy) \right|^2 \mu_W(d\theta) \\ &\leq \sup_{f \in \mathcal{L}} \int_{\Theta^K} \sum_{\sigma \in S_K} w_\sigma(\theta) \left| \int_{\Theta^K} f(y) \mathbf{p}_\sigma(\theta, dy) \right|^2 \mu_W(d\theta) \quad (\text{from convexity of } (\cdot)^2) \\ &= \sup_{f \in \mathcal{L}} \frac{1}{|S_K|} \sum_{\sigma \in S_K} \int_{\Theta^K} \left| \int_{\Theta^K} f(y) \mathbf{p}_\sigma(\theta, dy) \right|^2 \mu_\sigma(d\theta) \quad (\text{from the definition of } w_\sigma \text{ and } \mu_W). \end{aligned} \tag{16}$$

Now, let $\bar{f}_\sigma := \mu_\sigma(f)$. Notice that we have

$$\begin{aligned}
& \int_{\Theta^K} \left| \int_{\Theta^K} f(\mathbf{y}) \mathbf{p}_\sigma(\boldsymbol{\theta}, d\mathbf{y}) \right|^2 \mu_\sigma(d\boldsymbol{\theta}) \\
&= \int_{\Theta^K} \left| \int_{\Theta^K} (f(\mathbf{y}) - \bar{f}_\sigma + \bar{f}_\sigma) \mathbf{p}_\sigma(\boldsymbol{\theta}, d\mathbf{y}) \right|^2 \mu_\sigma(d\boldsymbol{\theta}) \\
&= \int_{\Theta^K} \left(\left| \int_{\Theta^K} (f(\mathbf{y}) - \bar{f}_\sigma) \mathbf{p}_\sigma(\boldsymbol{\theta}, d\mathbf{y}) \right|^2 + \left| \int_{\Theta^K} \bar{f}_\sigma \mathbf{p}_\sigma(\boldsymbol{\theta}, d\mathbf{y}) \right|^2 + 2\bar{f}_\sigma \int_{\Theta^K} (f(\mathbf{y}) - \bar{f}_\sigma) \mathbf{p}_\sigma(\boldsymbol{\theta}, d\mathbf{y}) \right) \mu_\sigma(d\boldsymbol{\theta}) \\
&= \int_{\Theta^K} \left(\int_{\Theta^K} (f(\mathbf{y}) - \bar{f}_\sigma) \mathbf{p}_\sigma(\boldsymbol{\theta}, d\mathbf{y}) \right)^2 \mu_\sigma(d\boldsymbol{\theta}) + (\bar{f}_\sigma)^2 + 2\bar{f}_\sigma \underbrace{\int_{\Theta^K} \int_{\Theta^K} (f(\mathbf{y}) - \bar{f}_\sigma) \mathbf{p}_\sigma(\boldsymbol{\theta}, d\mathbf{y}) \mu_\sigma(d\boldsymbol{\theta})}_{= 0 \text{ by stationarity}} \\
&= \left(\frac{\int_{\Theta^K} \left(\int_{\Theta^K} (f(\mathbf{y}) - \bar{f}_\sigma) \mathbf{p}_\sigma(\boldsymbol{\theta}, d\mathbf{y}) \right)^2 \mu_\sigma(d\boldsymbol{\theta})}{\int_{\Theta^K} (f(\boldsymbol{\theta}) - \bar{f}_\sigma)^2 \mu_\sigma(d\boldsymbol{\theta})} \right) \left(\int_{\Theta^K} (f(\boldsymbol{\theta}) - \bar{f}_\sigma)^2 \mu_\sigma(d\boldsymbol{\theta}) \right) + (\bar{f}_\sigma)^2 \\
&\leq \|\mathbf{P}_\sigma\|_{L_2^0(\Theta^K, \mu_\sigma) \mapsto L_2^0(\Theta^K, \mu_\sigma)}^2 \left(\int_{\Theta^K} (f(\boldsymbol{\theta}) - \bar{f}_\sigma)^2 \mu_\sigma(d\boldsymbol{\theta}) \right) + (\bar{f}_\sigma)^2 \\
&= \|\mathbf{P}_\sigma\|_{L_2^0(\Theta^K, \mu_\sigma) \mapsto L_2^0(\Theta^K, \mu_\sigma)}^2 \left(\int_{\Theta^K} f(\boldsymbol{\theta})^2 \mu_\sigma(d\boldsymbol{\theta}) \right) + \left(1 - \|\mathbf{P}_\sigma\|_{L_2^0(\Theta^K, \mu_\sigma) \mapsto L_2^0(\Theta^K, \mu_\sigma)}^2 \right) (\bar{f}_\sigma)^2 \\
&= \left(\int_{\Theta^K} f(\boldsymbol{\theta})^2 \mu_\sigma(d\boldsymbol{\theta}) \right) - \underbrace{\left(1 - \|\mathbf{P}_\sigma\|_{L_2^0(\Theta^K, \mu_\sigma) \mapsto L_2^0(\Theta^K, \mu_\sigma)}^2 \right)}_{:= \gamma, \text{ with } \gamma \in (0, 1)} \left(\int_{\Theta^K} (f(\boldsymbol{\theta}) - \bar{f}_\sigma)^2 \mu_\sigma(d\boldsymbol{\theta}) \right). \tag{17}
\end{aligned}$$

Replacing Equation (17) into Equation (16), we get

$$\begin{aligned}
\left\| \mathbf{P}^{(W)} \right\|_{L_2^0(\Theta^K, \mu_W) \mapsto L_2^0(\Theta^K, \mu_W)}^2 &\leq \sup_{f \in \mathcal{L}} \left(\int_{\Theta^K} f(\boldsymbol{\theta})^2 \mu_W(d\boldsymbol{\theta}) \right) - \frac{\gamma}{|S_K|} \sum_{\sigma \in S_K} \mathbb{V}_{\mu_\sigma}[f] \\
&\leq 1 - \gamma \left(\frac{\Lambda_m}{2 - \Lambda_m} \right) < 1 \quad (\text{by Proposition 4.6}).
\end{aligned}$$

Thus, $\mathbf{P}^{(W)}$ has an $L_2(\Theta^K, \mu_W)$ spectral gap. Once again, $L_r(\Theta^K, \mu_W)$ -geometric ergodicity (with $r \in [1, \infty]$) follows from Proposition 4.1 and the fact that $\mathbf{P}^{(W)}$ is μ_W -reversible by Proposition 3.11. \square

4.2.1 Discussion and comparison to similar theoretical result

Theorems 4.5 and 4.7 state the existence of an L_2 -spectral gap, hence L_r -geometric ergodicity for both the UGPT and the WGPT algorithm. Their proof provides also a quantification of the L_2 -spectral gap in terms of the L_2 -spectral gap of each individual Markov operator P_k . Such a bound is, however,

not satisfactory as it does not use any information on the temperature and it just states that the L_2 -spectral gap of the UWPT and WGPT chain is not worse than the smallest L_2 -spectral gap among the individual chains (without swapping). This result is not sharp, as it can be evidenced in the numerical section, where a substantial improvement in convergence is achieved by our methods.

Convergence results for the *standard* parallel tempering algorithm have been obtained in the works [33] and [41]. In particular, the work [33] has proved geometric ergodicity for the pairwise parallel tempering algorithm using the standard drift condition construction of [32]. It is unclear from that work which convergence rate is obtained for the whole algorithm. In comparison, our results are given in terms of spectral gaps. On the other hand, the work [41] presents conditions for rapid mixing of a particular type of parallel tempering algorithm, where the transition kernel is to be understood as a convex combination of such kernels, as opposed to our case, where it is to be understood as a tensorization. Their obtained results provide, for their setting, a better convergence rate than the one we obtained for the UGPT. We believe that their result can be extended to the UGPT algorithm, and this will be the focus of future work. On the other hand, the use of the ideas in [41] for the WGPT algorithm seems more problematic.

5 Numerical experiments

We now present four *toy* examples to illustrate the efficiency of both GPT algorithms discussed herein and compare them to the more traditional random walk Metropolis and standard PT algorithms. Notice that we compare the examples with respect to the “simplest” version of these methods, since more efficient variations, such as Adaptive Metropolis [19, 18], for example, can also be extended into the GPT framework. Experiments 5.1, 5.2 and 5.3 were run in a Dell (R) Precision (TM) T3620 workstation with Intel(R) Core(TM) i7-7700 CPU with 32 GB of RAM. Numerical simulations in Section 5.1 and 5.3 were run on a single thread, while the numerical simulations in Section 5.2 were run on an *embarrassingly parallel* fashion over 8 threads using the Message Passing Interface (MPI) and the Python package MPI4py [11]. Lastly, experiment 5.4 was run on the Fidis cluster of the EPFL. The scripts used to generate the results presented in this section were written in Python 3.6, and can be found in DOI: [10.5281/zenodo.3700048](https://doi.org/10.5281/zenodo.3700048)

Remark 5.1. In most Bayesian inverse problems, particularly those dealing with large-scale computational models, the computational cost is dominated by the evaluation of the forward operator, which can be, for example, the numerical approximation of a possibly non-linear partial differential equation. In the case where all possible permutations are considered (i.e., $S_K = \mathcal{S}_K$), there are $K!$ possible permutations of the states, the computation of the swapping ratio in the GPT algorithms can become prohibitively expensive if one is to evaluate $K!$ forward models, even for moderate values of K . This problem can be circumvented by storing the values $\pi(\theta_k^{(n)})$, $k = 1, \dots, K$, $n = 1, \dots, N$, since the

swapping ratio for GPT consists of permutations of these values, divided by the temperature parameters. Thus, “only” K forward model evaluations need to be computed at each step and the swapping ratio can be computed at negligible cost for moderate values of K . For higher values of K , it is advisable to only consider the union of properly chosen semi-groups A, B of \mathcal{S}_K , with $A \cap B \neq \emptyset$, such that A, B generates \mathcal{S}_K (i.e., if the smallest semi-groups that contains A and B is \mathcal{S}_K itself), and $|A \cup B| < |\mathcal{S}_K| = K!$, which is referred to as partial Infinite Swapping in the continuous case [16]. One particular way of choosing A and B is to consider, for example, A to be the set of permutations that only permute the indices associated with relatively low temperatures while leaving the other indices unchanged, and B as the set of permutations for the indices of relatively high temperatures, while leaving the other indices unchanged. Intuitively, swaps between temperatures that are, in a sense, “close” to each other tend to be chosen with a higher probability. We refer the reader to [16, Section 6.2] for a further discussion on this approach in the continuous-time setting. One additional idea would be to consider swapping schemes that, for example, only permute states between μ_i and $\mu_{i+1}, \mu_{i+2}, \dots, \mu_{i+\ell}$ for some user-defined $\ell \geq 1$ and any given $i = 1, 2, \dots, K - 1$. The intuition behind this choice also being that swaps between posteriors that are at close temperatures are more likely to occur than swaps between posteriors with a high temperature difference. We intend to explore this further in depth in future work.

5.1 Density concentrated over a quarter circle-shaped manifold

Let μ be a probability measure that has density π with respect to the uniform Lebesgue measure on the unit square $\mu_{\text{prior}} = \mathcal{U}([0, 1]^2)$ given by

$$\pi(\theta) = \frac{1}{Z} \exp\left(-10000(\theta_1^2 + \theta_2^2 - 0.8^2)^2\right) \mathbf{1}_{[0,1]^2}, \quad \theta = (\theta_1, \theta_2),$$

where Z is the normalization constant and $\mathbf{1}_{[0,1]^2}$ is the indicator function over the unit square. We remark that this example is not of particular interest *per se*; however, it can be used to illustrate some of the advantages of the algorithms discussed herein. The difficulty of sampling from such a distribution comes from the fact that its density is concentrated over a quarter circle-shaped manifold, as can be seen on the left-most plot in Figure 1. This in turn will imply that a single level RWM chain would need to take very small steps in order to properly explore such density.

We aim at estimating $\mathcal{Q}_k = \mathbb{E}_\mu[\theta_k] \approx \hat{\theta}_k$, for $k = 1, 2$. To do so, we implement five MCMC algorithms to sample from μ , namely Random Walk Metropolis (RWM), the standard PT (PT) with $N_s = 1$, as described in Section 3.2, the PSDPT algorithm of [24] and both versions of the GPT algorithm. We compare the quality of our algorithms by examining the variance of the estimators $\hat{\theta}_k$, $k = 1, 2$ computed over 100 independent MCMC runs of each algorithm, which we describe as follows. For the tempered algorithms (PT, PSDPT, UGPT, and WGPT), we consider $K = 4$ temperatures. A *rule of thumb* [17] for the choice

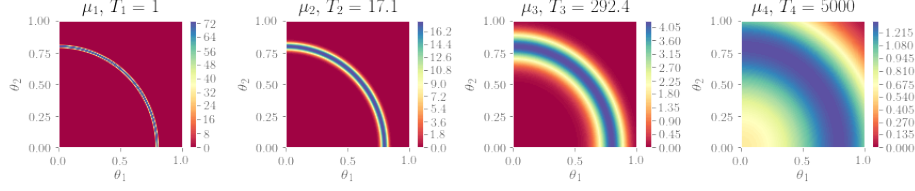


Figure 1: Tempered densities (with $T_1 = 1$, $T_2 = 17.1$, $T_3 = 292.4$, $T_4 = 5000$) for the density concentrated around a quarter circle-shaped manifold example. As we can see, the density becomes less concentrated as the temperature increases, which allows us to use RWM proposals with larger step sizes.

of temperatures is to set $T_i = a^{i-1}$, $k = 1, \dots, K$, for some positive constant $a > 1$. In particular, we choose $T_4 = 5000$, so that the tempered density π_4 becomes sufficiently simple to explore the target distribution. This gives $T_1 = 1, T_2 = 17.1, T_3 = 292.4, T_4 = 5000$. Moreover, for both GPT algorithms, we set $S_K = \mathcal{S}_K$, i.e., we consider all possible $K!$ permutations of $\{1, 2, \dots, K\}$. Notice that since this is a relatively small value of K , the computational time is dominated by the transition operator \mathbf{P} , rather than by the computation of the swapping ratio. In the current setting, the computational cost of PT and PSDPT is comparable to that of both GPT algorithms discussed herein. Each estimator is obtained by running the inversion experiment for $N = 25,000$ samples, discarding the first 20% of the samples (5000) as a burn-in. Notice that the tempering algorithms (i.e., PT, PSDPT, UGPT and WGPT) have a K -times larger computational cost than RWM, since such algorithms need to run a total of K chains. To account for this computational cost, we run the single-chain random walk Metropolis algorithm for $N_{\text{RWM}} = KN = 100,000$ iterations, and discard the first 20% of the samples obtained with the RWM algorithm (20,000) as a burn-in.

The RWM algorithm uses Gaussian proposals with covariance matrix $\Sigma_{\text{RWM}} = (0.025)^2 I_{2 \times 2}$, where $I_{2 \times 2}$ is the identity matrix in $\mathbb{R}^{2 \times 2}$. For the tempered algorithms (i.e., PT, PSDPT, and both versions of GPT), we use $K = 4$ RWM kernels p_k , $k = 1, 2, 3, 4$, with proposal density $q_{\text{prop},i}(\theta_k^{(n)}, \cdot) = \mathcal{N}(\theta_k^{(n)}, \sigma_k^2 I_{2 \times 2})$, where σ_k is shown in Table 2. This choice of σ_k gives an acceptance rate for each chain of around 0.23. Notice that σ_1 corresponds to the “step-size” of the single-temperature RWM algorithm.

	$k = 1$	$k = 2$	$k = 3$	$k = 4$
σ_k	0.022	0.090	0.310	0.650

Table 1: Step size of the RWM proposal distribution for the manifold experiment. This choice of step size provides an average acceptance rate for each chain, at each temperature, of around 0.23 for all the algorithms tested. Such values are relatively close to the “optimal” value of 0.234 in [37].

Experimental results for the ergodic run are shown in Table 5.1. We can see how both GPT algorithms provide a gain over both RWM, PT and PSDPT algorithms, with the WGPT algorithm providing the largest gain. Scatter plots of the samples obtained with each method are presented in Figure 2. Here, the subplot titled “WGPT” (second from right to left) corresponds to weighted samples from μ_W , with weight \hat{w} as in (13), while the one titled “WGPT (inv)” (rightmost) corresponds to samples from μ_W without any post-processing. Notice how the samples from the latter concentrates over a *wider* manifold, which in turn makes the target density easier to explore when using state-dependent Markov transition kernels.

	Mean		MSE		MSE _{RWM} /MSE	
	θ_1	θ_2	θ_1	θ_2	θ_1	θ_2
RWM	0.50996	0.50657	0.00253	0.00261	1.00	1.00
PT	0.50978	0.51241	0.00024	0.00021	10.7	11.0
PSDPT	0.50900	0.50956	0.00027	0.00026	9.53	10.2
UGPT	0.50986	0.50987	0.00016	0.00016	16.1	16.4
WGPT	0.51062	0.50838	0.00015	0.00014	16.9	18.4

Table 2: Results for the density concentrated around a circle-shaped manifold experiment. As we can see, both GPT algorithms provide an improvement over PT, PSDPT and RWM. The computational cost is comparable across all algorithms.

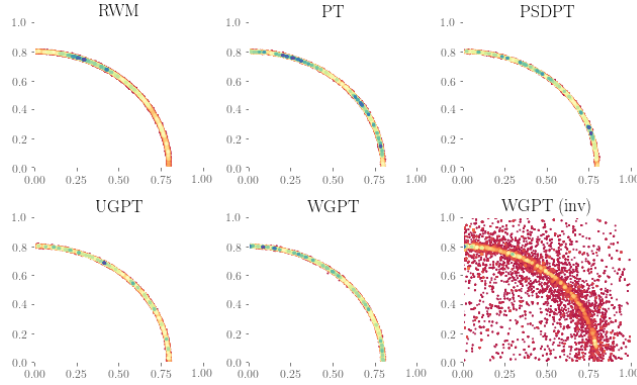


Figure 2: Scatter-plots of the samples from μ obtained with each algorithm on a single run. Top, from left to right: random walk Metropolis, PT and PSDPT. Bottom, from left to right: UGPT, WGPT (after re-weighting the samples), and WGPT, before re-weighting the samples.

5.2 Multiple source elliptic BIP

We now consider a slightly more challenging problem, for which we try to recover the probability distribution of the location of a source term in a Poisson equation (Eq. (18)), based on some noisy measured data. Let $(\Theta, \mathcal{B}(\Theta), \mu_{\text{prior}})$ be the measure space, set $\Theta = \bar{D} := [0, 1]^2$, with Lebesgue (uniform) measure μ_{prior} , and consider the following Poisson's equation with homogeneous boundary conditions:

$$\begin{cases} \Delta u(x, \theta) = f(x, \theta), & x \in D, \theta \in \Theta, \\ u(x, \theta) = 0, & x \in \partial D. \end{cases} \quad (18)$$

Such equation can model, for example, the electrostatic potential $u := u(x, \theta)$ generated by a charge density $f(x, \theta)$ depending on an *uncertain* location parameter $\theta \in \Theta$. Data y is recorded on an array of 64×64 equally-spaced points in D by solving (18) with a forcing term given by

$$f(x) = \sum_{i=1}^4 e^{-1000[(x_1 - s_1^{(i)})^2 + (x_2 - s_2^{(i)})^2]}, \quad (19)$$

where the true source locations $s^{(i)}$, $i = 1, 2, 3, 4$, are given by $s^{(1)} = (0.2, 0.2)$, $s^{(2)} = (0.2, 0.8)$, $s^{(3)} = (0.8, 0.2)$, and $s^{(4)} = (0.8, 0.8)$. Such data is assumed to be polluted by an additive Gaussian noise $\eta \sim \mathcal{N}(0, \sigma^2 I_{64 \times 64})$, with $\sigma = 3.2 \times 10^{-6}$, (which corresponds to a 1% noise) and where $I_{64 \times 64}$ is the 64-dimensional identity matrix. Thus, we set $(Y, \|\cdot\|_Y) = (\mathbb{R}^{64 \times 64}, \|\cdot\|_\Sigma)$, with $\|A\|_\Sigma = (64\sigma)^{-2} \|A^T A\|_F$, for some arbitrary matrix $A \in \mathbb{R}^{64 \times 64}$, where $\|\cdot\|_F$ is the Frobenius norm. We assume a misspecified model where we only consider a single source in Eq. (19). That, is, we construct our forward operator $\mathcal{F} : \Theta \mapsto Y$ by solving (18) with a source term given by

$$f(x, \theta) = e^{-1000[(x_1 - \theta_1)^2 + (x_2 - \theta_2)^2]}. \quad (20)$$

In this particular setting, this leads to a posterior distribution with four modes since the prior density is uniform in the domain and the likelihood has a local maximum whenever $(\theta_1, \theta_2) = (s_1^{(i)}, s_2^{(i)})$, $i = 1, 2, 3, 4$. The Bayesian inverse problem at hand can be understood as sampling from the posterior measure μ , which has a density with respect to the prior $\mu_{\text{prior}} = \mathcal{U}(\bar{D})$ given by

$$\pi(\theta) = \frac{1}{Z} \exp \left(-\frac{1}{2} \|y - \mathcal{F}(\theta)\|_\Sigma^2 \right),$$

for some (intractable) normalization constant Z as in (4). We remark that the solution to (18) with a forcing term of the form of (20) is approximated using a second-order accurate finite difference approximation with grid-size $h = 1/64$ on each spatial component.

The difficulty in sampling from the current BIP arises from the fact that the resulting posterior μ is multi-modal and the number of modes is not known apriori (see Figure 3).

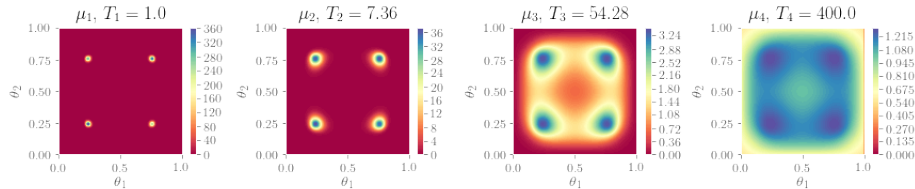


Figure 3: True tempered densities for the elliptic BIP example. Notice that the density is not symmetric, due to the additional random noise.

We follow a similar experimental setup as in the previous example, by implementing RWM, PT (with $N_s = 1$), the pairwise state dependent parallel tempering algorithm PSDPT of [24], and both versions of the GPT algorithms. For the PT, PSDPT and GPT algorithms, four different temperatures are used, with $T_1 = 1$, $T_2 = 7.36$, $T_3 = 54.28$, and $T_4 = 400$. Once again, we set $S_K = \mathcal{S}_K$ for both GPT algorithms (i.e., we consider all possible permutations). Given that $4!$ is a moderately small number, the computational cost of evaluating the forward model is much higher than the cost associated with computing the swapping ratio.

Since we have $K = 4$ temperatures, we run the RWM algorithm for K -times longer, so that the computational cost of all algorithms tested is comparable. For each run, we obtain $N = 25,000$ samples with the PT, PSDPT, and both GPT algorithms, and $N = 100,000$ samples with RWM, discarding the first 20% of the samples in both cases (5000, 20000, respectively) as a burn-in. On each of the tempered chains, we use RWM proposals, with step-sizes shown in table 3. This choice of step size provides an acceptance rate of about 0.24 across all tempered chains and all tempered algorithms. For the single-temperature RWM run, we choose a larger step size ($\sigma_{\text{RWM}} = 0.16$) so that the RWM algorithm is able to explore the whole distribution. Such a choice, however, provides a smaller acceptance rate of about 0.01 for the single-chain RWM.

Experimental results are shown in Table 4. Once again, we can see how both GPT algorithms provide a gain over both RWM and both variations of the PT algorithm, with the WGPT algorithm providing a larger gain. Scatter-plots of the obtained samples are shown in Figure 3.

	$k = 1$	$k = 2$	$k = 3$	$k = 4$
$\sigma_{i,\text{PT,PSDPT,GPT}}$	0.030	0.100	0.400	0.600
$\sigma_{i,\text{RWM}}$	0.160	-	-	-

Table 3: Step size of the RWM proposal distribution for the elliptic BIP experiment. This choice of step size provides an acceptance rate of about 0.24 for all the tempered algorithms tested. The step size for the the single-temperature RWM is chosen to be 0.16, so that the sampler can explore the whole distribution. This in turn results in an acceptance rate of about 0.01.

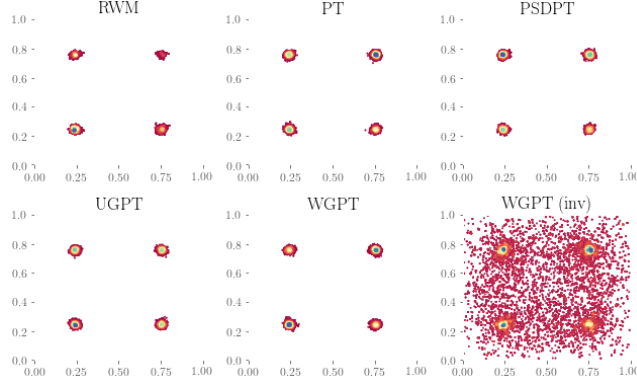


Figure 4: Scatterplots of the samples from μ obtained with different algorithms on a single run. Top, from left to right: random walk Metropolis, PT and PSDPT. Bottom, from left to right: UGPT, WGPT (after re-weighting the samples), and WGPT, before re-weighting the samples. As we can see, WGPT (before re-weighting) is able to "connect" the parameter space.

	Mean		MSE		MSE _{RWM} /MSE	
	θ_1	θ_2	θ_1	θ_2	θ_1	θ_2
RWM	0.48509	0.51867	0.00986	0.01270	1.00	1.00
PT	0.48731	0.50758	0.00042	0.00036	23.0	29.2
PSDPT	0.48401	0.50542	0.00079	0.00099	12.4	10.7
UGPT	0.48624	0.50620	0.00038	0.00027	25.9	38.2
WGPT	0.48617	0.50554	0.00025	0.00023	38.6	44.9

Table 4: Results for the elliptic BIP problem. Once again, we can see that both GPT algorithms provide an improvement over RWM, PT and PSDPT. The computational cost is comparable across all algorithms, given that the cost of each iteration is dominated by the cost of solving the underlying PDE.

5.3 1D wave source inversion

We consider a small variation of example 5.1 in [34]. Let $(\Theta, \mathcal{B}(\Theta), \mu_{\text{prior}})$ be a measure space, with $\Theta = [-5, 5]$ and uniform (Lebesgue) measure μ_{prior} . Consider the following Cauchy problem for the 1D wave equation:

$$\begin{cases} u_{tt}(x, t, \theta) - u_{xx}(x, t, \theta) = 0, & t \in (0, T], x \in \mathbb{R}, \theta \in \Theta, \\ u(x, 0, \theta) = h(x, \theta), u_t(x, 0, \theta) = 0, & x \in \mathbb{R}, \theta \in \Theta. \end{cases} \quad (21)$$

Here, $h(x, \theta)$ acts as a source term generating a initial wave pulse. Notice that Equation (21) can be easily solved using d'Alembert's formula, namely

$$u(x, t, \theta) = \frac{1}{2} (h(x - t, \theta) + h(x + t, \theta)).$$

Synthetic data y is generated by solving Equation (21) with initial data

$$h(x, \theta_1, \theta_2) = \frac{1}{2} \left(e^{-100(x-\theta_1-0.5)^2} + e^{-100(x-\theta_1)^2} + e^{-100(x-\theta_1+0.5)^2} \right. \\ \left. + e^{-100(x-\theta_2-0.5)^2} + e^{-100(x-\theta_2)^2} + e^{-100(x-\theta_2+0.5)^2} \right),$$

with $\theta_1 = -3, \theta_2 = 3$ and observed at $N_R = 11$ equally-spaced receiver locations between $R_1 = -5$ and $R_2 = 5$ on $N_T = 1000$ time instances between $t = 0$ and $T = 5$. The signal recorded by each receiver is assumed to be polluted by additive Gaussian noise $\eta \sim \mathcal{N}(0, \sigma^2 I_{1000 \times 1000})$, with $\sigma = 0.01$, which corresponds to roughly 1% noise. Thus, we set $(Y, \|\cdot\|_Y) = (\mathbb{R}^{11 \times 1000}, \|\cdot\|_\Sigma)$, with $\|A\|_\Sigma = (\sqrt{N_R} \sigma)^{-2} \sum_{i=1}^{N_R} \sum_{j=1}^{N_T} A_{i,j}^2$, $A \in \mathbb{R}^{11 \times 1000}$. Once again, we assume a misspecified model where we construct our forward operator $\mathcal{F} : \Theta \mapsto Y$ by solving (21) with a source term given by

$$h(x, \theta) = \left(e^{-100(x-\theta-0.5)^2} + e^{-100(x-\theta)^2} + e^{-100(x-\theta+0.5)^2} \right),$$

this is done to further increase the multi-modality of the resulting posterior. The Bayesian inverse problem at hand can be understood as sampling from the posterior measure μ , which has a density with respect to the prior $\mu_{\text{prior}} = \mathcal{U}([-5, 5])$ given by

$$\pi(\theta) = \frac{1}{Z} \exp \left(-\frac{1}{2} \|y - \mathcal{F}(\theta)\|_\Sigma^2 \right) = \frac{1}{Z} \exp(-\Phi(\theta; y)),$$

for some (intractable) normalization constant Z as in (4). The difficulty in solving this BIP comes from the high multi-modality of the potential $\Phi(\theta; y)$, as it can be seen in Figure 5. This, shape of $\Phi(\theta; y)$ makes the posterior difficult to explore using local proposals.

We follow a similar experimental setup as in the previous example, by implementing RWM, PT (with $N_s = 1$), PSDPT and both versions of the GPT algorithms. For the PT, PSDPT and GPT algorithms, five different temperatures are used, with $T_1 = 1$, $T_2 = 5$, $T_3 = 25$, $T_4 = 125$ and $T_5 = 625$. Once again, we set $S_K = \mathcal{S}_K$ for both GPT algorithms (i.e., we consider all possible permutations). Given that $5!$ is still a moderately small number, the computational cost across all algorithms is comparable. Similarly as before, we run the RWM algorithm K -times longer. For each run, we obtain $N = 25,000$ samples with the PT, PSDPT, and both GPT algorithms, and $N = 125,000$ samples with RWM, discarding the first 20% of the samples in both cases (5000, 25000, respectively) as a burn-in. On each of the tempered chains, we use RWM proposals, with step-sizes shown in table 5. This choice of step size provides

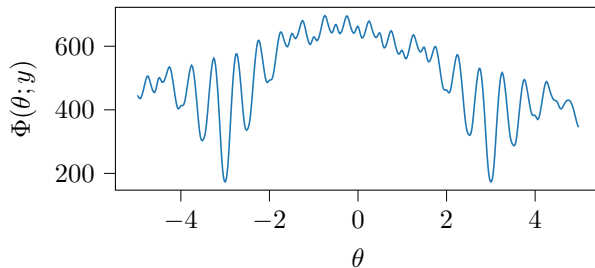


Figure 5: Multi-modal potential for the Cauchy problem. Notice the minima around $\theta = -3$ and $\theta = 3$.

an acceptance rate of about 0.4 across all tempered chains and all tempered algorithms.

Experimental results are shown in Table 6. Once again, we can see how both GPT algorithms provide a gain over both RWM and both variations of the PT algorithm, with the WGPT algorithm providing the largest gain. Notice that, given the high multi-modality of the posterior at hand, the simple RWM algorithm is not well-suited for this type of distribution, as it can be seen from its large variance; this suggests that the RWM usually gets "stuck" at one mode of the posterior. Notice that, intuitively, due to the symmetric nature of the potential, one would expect the true mean of θ to be close to 0. This value was computed by means of numerical integration and is given by $\mathbb{E}_\mu[\theta] = 0.08211$.

	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$
$\sigma_{i,\text{PT,PSDPT,GPT}}$	0.02	0.05	0.10	0.50	2.0
$\sigma_{i,\text{RWM}}$	0.5	-	-	-	-

Table 5: Step size of the RWM proposal distribution for the Cauchy BIP experiment. This choice of step size provides an acceptance rate of about 0.4 for all the tempered algorithms tested. The choice of step-size for the RWM algorithm is done in such a way that it can "jump" modes, which are at distance of roughly $1/2$.

	Mean	MSE	$\text{MSE}_{\text{RWM}}/\text{MSE}$
RWM	-0.10120	9.36709	1.000
PT	0.05118	0.03681	254.5
PSDPT	0.15840	0.21701	43.20
UGPT	0.08976	0.03032	308.9
WGPT	0.06149	0.02518	372.0

Table 6: Results for the 1D Cauchy BIP problem. Once again, we can see that both GPT algorithms provide an improvement over RWM, PT and PSDPT. The computational cost is comparable across all algorithms.

5.4 Acoustic wave inversion

Lastly, we consider a more challenging problem, for which we try to recover the probability distribution of the spatial location of a (point-like) source term, together with the material properties of the medium, on an acoustic wave equation (Eq. (22)), based on some noisy measured data. We begin by describing the mathematical model of such wave phenomena. Let $(\Theta, \mathcal{B}(\Theta), \mu_{\text{prior}})$ be the measure space, with Lebesgue (uniform) measure μ_{prior} , set $\bar{D} := [0, 3] \times [0, 2]$, $\partial D = \bar{\Gamma}_N \cup \bar{\Gamma}_{\text{Abs}}$, $\bar{\Gamma}_N \cap \bar{\Gamma}_{\text{Abs}} = \emptyset$, $|\Gamma_N|, |\Gamma_{\text{Abs}}| > 0$, and define $\Theta = D \times \Theta_\alpha \times \Theta_\beta$, where $\Theta_\alpha = [6, 14]$, $\Theta_\beta = [4500, 5500]$. Here, we are considering a rectangular spatial domain D , with the top boundary denoted by Γ_N and the side and bottom boundaries denoted by Γ_{Abs} . Lastly, let $\theta := (\theta_1, \theta_2, \alpha, \beta) \in \Theta$. Consider the following acoustic wave equation with absorbing boundary conditions:

$$\begin{cases} \alpha^2 u_{tt}(x, t, \theta) - \beta^2 \Delta u(x, t, \theta) = f(x, t, \theta), & (x, t, \theta) \in D \times (0, T) \times \Theta, \\ u(x, 0, \theta) = u_t(x, 0, \theta) = 0, & (x, t, \theta) \in D \times \{0\} \times \Theta, \\ \beta^2 \nabla u(x, t, \theta) \cdot \hat{n} = 0, & (x, t, \theta) \in \Gamma_N \times (0, T) \times \Theta, \\ \beta^2 \nabla u(x, t, \theta) \cdot \hat{n} = -\alpha \beta u_t(x, t, \theta), & (x, t, \theta) \in \Gamma_{\text{Abs}} \times (0, T) \times \Theta. \end{cases} \quad (22)$$

Here the boundary condition on the top boundary Γ_N corresponds to a Neumann boundary condition, while the boundary condition on Γ_{Abs} corresponds to the so-called absorbing boundary condition, a type of artificial boundary condition used to minimize reflection of wave hitting the boundary. Data $y \in Y$ is obtained by solving Equation (22) with a force term given by

$$f(x, t, \theta) = 10^{11} e^{-\frac{1}{2 \cdot 0.1^2} [(x_1 - \theta_1)^2 + (x_2 - \theta_2)^2]} (1 - 2 \cdot 1000 \pi^2 t^2) e^{-2 \cdot 1000^2 \pi^2 t^2}, \quad (23)$$

with a true set of parameters $\Theta \ni \theta^* := (\theta_1, \theta_2, \alpha, \beta)$ given by $\theta_1 = 1.5, \theta_2 = 1.0, \alpha = 10, \beta = 5000$, and observed on $N_R = 3$ different receiver locations $R_1 = (1.0, 2.0), R_2 = (1.5, 2.0), R_3 = (2.0, 2.0)$ at $N_T = 117$ equally-spaced time instants between $t = 0$ and $t = 0.004$. In physical terms, the parameters θ_1, θ_2 represent the source location, while the parameters α, β are related to the material properties of the medium. Notice that, on a slight abuse of notation, we have used the symbol π to represent the number $3.14159 \dots$ in Equation (23) and it should not be confused with the symbol for density. The data measured by each receiver is polluted by an additive Gaussian noise $\eta \sim \mathcal{N}(0, \sigma^2 I_{117 \times 117})$, with $\sigma = 0.013$, which corresponds to roughly a 2% noise. Thus, we have that $(Y, \|\cdot\|_Y) = (\mathbb{R}^{3 \times 117}, \|\cdot\|_\Sigma)$, where $\|A\|_\Sigma^2 := (\sqrt{N_R} \sigma)^{-2} \sum_{i=1}^{N_R} \sum_{j=0}^{N_T} A_{i,j}^2$. Thus, the forward mapping operator $\mathcal{F} : \Theta \mapsto Y$ can be understood as the numerical solution of Equation (22) evaluated at 117 discrete time instants at each of the 3 receiver locations. Such a numerical approximation is obtained by the finite element method using piece-wise linear elements and the time stepping is done using a Forward Euler scheme with sufficiently small time-steps to respect the so-called Courant-Fredrichs-Lewy condition [36]. This numerical solution

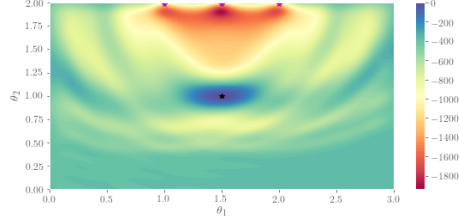


Figure 6: Plot of the log-likelihood for different values of θ_1 , θ_2 and fixed values of $\alpha = 10$ and $\beta = 5000$. The magenta points represent the receiver locations R_1, R_2, R_3 . The black point represents the true location of the source $(\theta_1, \theta_2) = (1.5, 1.0)$.

is implemented using the Python library FEniCS [27], using 40×40 triangular elements. The Bayesian inverse problem at hand can thus be understood as sampling from the posterior measure μ , which has a density with respect to the prior $\mu_{\text{prior}} = \mathcal{U}(\Theta)$ given by

$$\pi(\theta) = \frac{1}{Z} \exp \left(-\frac{1}{2} \|y - \mathcal{F}(\theta)\|_{\Sigma}^2 \right).$$

The previous BIP presents two difficulties; on the one hand, Equation (22) is, typically, expensive to solve, which in turn translates into expensive evaluations of the posterior density. On the other, the log-likelihood has an extremely complicated structure, which in turn makes its exploration difficult. This can be seen in Figure 6, where we plot of the log-likelihood for different source locations (θ_1, θ_2) and for fixed values of the material properties $\alpha = 10, \beta = 5000$. More precisely, we plot $\tilde{\Phi}((\theta_1, \theta_2); y) := -\frac{1}{2} \|y - \mathcal{F}(\theta_1, \theta_2, 10, 5000)\|_{\Sigma}^2$ on a grid of 100×100 equally spaced points (θ_1, θ_2) in D . It can be seen that, even though the log-likelihood has a clear peak around the true value of (θ_1, θ_2) , there are also regions of relatively high concentration of log-probability, surrounded by regions with a significantly smaller log-probability, making it a suitable problem for our setting.

Following the same set-up of previous experiments, we aim at estimating $\mathcal{Q}_k = \mathbb{E}_{\mu}[\theta_k] \approx \hat{\theta}_k$, for $k = 1, 2$. To do so, we implement five MCMC algorithms to sample from μ , namely Random Walk Metropolis (RWM), the standard PT (with $N_s = 1$) as described in Section 3.2, the PSDPT algorithm of [24] and both versions of the GPT algorithm. We compare the quality of our algorithms by examining the variance of the estimators $\hat{\theta}_k$, $k = 1, 2, 3, 4$ computed over 50 independent MCMC runs of each algorithm, which we describe as follows. Once again, we consider $K = 4$ temperatures for the tempered algorithms (PT, PSDPT, UGPT, and WGPT), setting temperatures to $T_i = a^{i-1}$, $k = 1, \dots, K$, for some positive constant $a > 1$. In particular, we choose $T_4 = 400$, which gives $T_1 = 1, T_2 = 7.36, T_3 = 54.28, T_4 = 400$. We follow a similar experimental setup as in the previous three experiments and consider $S_K = \mathcal{S}_K$. For the tempered

algorithms, each estimator is obtained by running the inversion experiment for $N = 7,000$ samples, discarding the first 20% of the samples (1400) as a burn-in. For the RWM algorithm, we run the inversion experiment for $N_{\text{RWM}} = KN = 28,000$ iterations, and discard the first 20% of the samples obtained (5600) as a burn-in.

Each individual chain is constructed using Gaussian RWM proposals $q_{\text{prop},k}(\theta_k^n, \cdot) = \mathcal{N}(\theta_k^n, \mathcal{C}_k)$, $k = 1, 2, 3, 4$, with covariance \mathcal{C}_k described in Table 7. The covariance is tuned in such a way that the acceptance rate of each chain is around 0.23. The variance of the estimators obtained with each method is presented in Table 8. Once again, our GPT algorithms outperform all other tested methods for this particular setting. In particular, our methods provide huge computational gains when compared to RWM and the PSDPT algorithm of [24], as well as some moderate computational gains when compared to the standard PT.

	$k = 1$	$k = 2$	$k = 3$	$k = 4$
$\mathcal{C}_{i,\text{PT,PSDPT,GPT}}^{1/2}$	Diag(0.01, 0.01, 0.2, 5)	Diag(0.06, 0.06, 0.4, 14)	Diag(0.3, 0.3, 0.6, 20)	Diag(1, 1, 1, 50)
$\mathcal{C}_{i,\text{RWM}}^{1/2}$	Diag(0.02, 0.02, 0.2, 5)	-	-	-

Table 7: Step size of the RWM proposal distribution for the acoustic BIP experiment. Here $\text{Diag}(d_1, d_2, \dots, d_N)$ is to be understood as the $N \times N$ diagonal matrix with entries d_1, d_2, \dots, d_N . This choice of covariance provides an acceptance rate of about 0.25 across all algorithms.

	Mean		Var		$\text{Var}_{\text{RWM}}/\text{Var}$	
	θ_1	θ_2	θ_1	θ_2	θ_1	θ_2
RWM	1.33801	1.54293	9.86×10^{-1}	8.21×10^{-2}	1.000000	1.000
PT	1.50121	1.00829	6.61×10^{-6}	2.77×10^{-4}	149136.1	296.2
PSDPT	1.39775	1.23119	2.48×10^{-1}	6.54×10^{-2}	3.900000	1.200
UGPT	1.50177	1.00711	2.72×10^{-6}	2.38×10^{-4}	361744.5	345.0
WGPT	1.50174	1.00601	2.08×10^{-6}	1.46×10^{-4}	472133.2	558.6

Table 8: Results for the acoustic BIP problem. Once again, we can see that both GPT algorithm provide an improvement over RWM, PT and PSDPT. The computational cost is comparable across all algorithms, given that the cost of each iteration is dominated by the cost of solving the underlying PDE.

6 Conclusions and future work

In the current work, we have proposed, implemented, and analyzed two versions of the GPT, and applied these methods to a BIP context. We demonstrate that such algorithms produce reversible and geometrically-ergodic chains under relatively mild conditions. As shown in Section 5, such sampling algorithms provide

an attractive alternative to the more standard Parallel Tempering when sampling from *difficult* (i.e., multi-modal or concentrated around a manifold) posteriors. We remark that the framework considered here-in can be combined with other, more advanced MCMC algorithms, such as, e.g., the Metropolis-adjusted Langevin algorithm (MALA), or the Delayed Rejection Adaptive Metropolis (DRAM), for example [18].

We intend to carry out a number of future extensions of the work presented herein. One of our short-term goals is to extend the methodology developed in the current work to a Multi-level Markov Chain Monte Carlo context, as in [14]. In addition, from a theoretical point of view, we would like to investigate the role that the number of chains and the choice of temperatures play on the convergence of the GPT algorithm, as it has been done previously for Parallel Tempering in [41]. Improving on the estimates presented here would likely be the focus of future work. We also believe that by excluding the identity permutation (i.e., $\text{id} \notin S_K$) on the UGPT, one could obtain a swapping kernel which is *better* in the so-called Peskun sense, see [1] for more details. We intend to carry further numerical experiments to better understand and compare swapping strategies. Furthermore, from a computational perspective, given that the framework presented in this work is, in principle, dimension independent, the methods explored in this work can also be combined with dimension-independent samplers such as the ones presented in [7, 10], thus providing a sampling algorithm robust to both multi-modality and large dimensionality of the parameter space. Given the additional computational cost of these methods, a non-trivial coupling of GPT and these methods needs to be devised. Lastly, we aim at applying the methods developed in the current work to more computationally challenging BIP, in particular those arising in seismology and seismic source inversion, where it is not uncommon to find multi-modal posterior distributions when inverting for a point source.

Auxiliary results

.1 Proof of Proposition 4.6

Proof. This proof is partially based on the proof of Theorem 1.2 in [29]. Let $\boldsymbol{\theta}, \mathbf{y} \in \Theta^K$ and define $\bar{f}_\sigma := \boldsymbol{\mu}_\sigma(f)$. The right-most inequality follows from the fact that

$$\begin{aligned} 1 = \mathbb{V}_{\boldsymbol{\mu}_W}[f] &= \int_{\Theta^K} f(\boldsymbol{\theta})^2 \boldsymbol{\mu}_W(d\boldsymbol{\theta}) = \frac{1}{|S_K|} \sum_{\sigma \in S_K} \int_{\Theta^K} f^2(\boldsymbol{\theta}) \boldsymbol{\mu}_\sigma(d\boldsymbol{\theta}) = \frac{1}{|S_K|} \sum_{\sigma \in S_K} \left(\mathbb{V}_{\boldsymbol{\mu}_\sigma}[f] + \bar{f}_\sigma^2 \right) \\ &\geq \frac{1}{|S_K|} \sum_{\sigma \in S_K} \mathbb{V}_{\boldsymbol{\mu}_\sigma}[f] \end{aligned}$$

We follow a procedure similar to the proof of [29, Theorem 1.2] for the lower bound on the variance. We introduce an ordering on $S_K = \sigma_1, \sigma_2, \dots, \sigma_{|S_K|}$,

define the matrix $C \in \mathbb{R}^{|S_K| \times |S_K|}$ as the matrix with entries

$$C_{ij} = \int_{\Theta^K} \int_{\Theta^K} (f(\boldsymbol{\theta}) - f(\mathbf{y}))^2 \boldsymbol{\mu}_{\sigma_i}(\mathrm{d}\boldsymbol{\theta}) \boldsymbol{\mu}_{\sigma_j}(\mathrm{d}\mathbf{y}),$$

where $C_{jj} = 2\mathbb{V}_{\boldsymbol{\mu}_{\sigma_j}}[f]$ and

$$\begin{aligned} 2 = 2\mathbb{V}_{\boldsymbol{\mu}_w}[f] &= \int_{\Theta^K} \int_{\Theta^K} (f(\boldsymbol{\theta}) - f(\mathbf{y}))^2 \left(\frac{1}{|S_K|} \sum_{i=1}^{|S_K|} \boldsymbol{\mu}_{\sigma_i}(\mathrm{d}\boldsymbol{\theta}) \right) \left(\frac{1}{|S_K|} \sum_{j=1}^{|S_K|} \boldsymbol{\mu}_{\sigma_j}(\mathrm{d}\mathbf{y}) \right) \\ &= \sum_{i,j} \frac{1}{|S_K|^2} C_{ij}. \end{aligned} \quad (24)$$

We thus aim at finding an upper bound of Equation (24) in terms of $(|S_K|)^{-1} \sum_{\sigma \in S_K} \mathbb{V}_{\sigma}[f]$.

By assumption C3, for any $\sigma_i, \sigma_j \in S_K$ the densities $\boldsymbol{\pi}_{\sigma_i}, \boldsymbol{\pi}_{\sigma_j}$ of $\boldsymbol{\mu}_{\sigma_i}, \boldsymbol{\mu}_{\sigma_j}$ (with respect to $\boldsymbol{\mu}_{\text{prior}}$) have an overlap $\Lambda_{\sigma_i, \sigma_j} > 0$. For brevity, in the following we use the shorthand notation $\Lambda_{i,j}$ for $\Lambda_{\sigma_i, \sigma_j}$. Thus, we can find densities $\boldsymbol{\eta}_{ij} := \Lambda_{ij}^{-1} \min_{\boldsymbol{\theta} \in \Theta^K} \{\boldsymbol{\pi}_{\sigma_i}(\boldsymbol{\theta}), \boldsymbol{\pi}_{\sigma_j}(\boldsymbol{\theta})\}, \boldsymbol{\varphi}_i, \boldsymbol{\psi}_j$ such that $\boldsymbol{\pi}_{\sigma_i} = \Lambda_{ij} \boldsymbol{\eta}_{ij} + (1 - \Lambda_{ij}) \boldsymbol{\varphi}_i$, and $\boldsymbol{\pi}_{\sigma_j} = \Lambda_{ij} \boldsymbol{\eta}_{ij} + (1 - \Lambda_{ij}) \boldsymbol{\psi}_j$. Thus, we get for the diagonal entries of the C matrix:

$$\begin{aligned} C_{ii} &= 2\mathbb{V}_{\boldsymbol{\mu}_{\sigma_i}}[f] \\ &= \int_{\Theta^K} \int_{\Theta^K} (f(\boldsymbol{\theta}) - f(\mathbf{y}))^2 (\Lambda_{ij} \boldsymbol{\eta}_{ij}(\boldsymbol{\theta}) + (1 - \Lambda_{ij}) \boldsymbol{\varphi}_i(\boldsymbol{\theta})) (\Lambda_{ij} \boldsymbol{\eta}_{ij}(\mathbf{y}) + (1 - \Lambda_{ij}) \boldsymbol{\varphi}_i(\mathbf{y})) \boldsymbol{\mu}_{\text{prior}}(\mathrm{d}\boldsymbol{\theta}) \boldsymbol{\mu}_{\text{prior}}(\mathrm{d}\mathbf{y}) \\ &= \int_{\Theta^K} \int_{\Theta^K} (f(\boldsymbol{\theta}) - f(\mathbf{y}))^2 \Lambda_{ij}^2 \boldsymbol{\eta}_{ij}(\boldsymbol{\theta}) \boldsymbol{\eta}_{ij}(\mathbf{y}) \boldsymbol{\mu}_{\text{prior}}(\mathrm{d}\boldsymbol{\theta}) \boldsymbol{\mu}_{\text{prior}}(\mathrm{d}\mathbf{y}) \\ &\quad + \int_{\Theta^K} \int_{\Theta^K} (f(\boldsymbol{\theta}) - f(\mathbf{y}))^2 \Lambda_{ij} (1 - \Lambda_{ij}) \boldsymbol{\varphi}_i(\mathbf{y}) \boldsymbol{\eta}_{ij}(\boldsymbol{\theta}) \boldsymbol{\mu}_{\text{prior}}(\mathrm{d}\boldsymbol{\theta}) \boldsymbol{\mu}_{\text{prior}}(\mathrm{d}\mathbf{y}) \\ &\quad + \int_{\Theta^K} \int_{\Theta^K} (f(\boldsymbol{\theta}) - f(\mathbf{y}))^2 \Lambda_{ij} (1 - \Lambda_{ij}) \boldsymbol{\varphi}_i(\boldsymbol{\theta}) \boldsymbol{\eta}_{ij}(\mathbf{y}) \boldsymbol{\mu}_{\text{prior}}(\mathrm{d}\boldsymbol{\theta}) \boldsymbol{\mu}_{\text{prior}}(\mathrm{d}\mathbf{y}) \\ &\quad + \int_{\Theta^K} \int_{\Theta^K} (f(\boldsymbol{\theta}) - f(\mathbf{y}))^2 (1 - \Lambda_{ij})^2 \boldsymbol{\varphi}_i(\mathbf{y}) \boldsymbol{\varphi}_i(\boldsymbol{\theta}) \boldsymbol{\mu}_{\text{prior}}(\mathrm{d}\boldsymbol{\theta}) \boldsymbol{\mu}_{\text{prior}}(\mathrm{d}\mathbf{y}) \\ &= 2\Lambda_{ij}^2 \mathbb{V}_{\boldsymbol{\eta}_{ij}}[f] + 2(1 - \Lambda_{ij})^2 \mathbb{V}_{\boldsymbol{\varphi}_i}[f] + 2\Lambda_{ij}(1 - \Lambda_{ij}) \int_{\Theta^K} \int_{\Theta^K} (f(\boldsymbol{\theta}) - f(\mathbf{y}))^2 \boldsymbol{\eta}_{ij}(\boldsymbol{\theta}) \boldsymbol{\varphi}_i(\boldsymbol{\theta}) \boldsymbol{\mu}_{\text{prior}}(\mathrm{d}\boldsymbol{\theta}) \boldsymbol{\mu}_{\text{prior}}(\mathrm{d}\mathbf{y}). \end{aligned} \quad (25)$$

Notice that equation (25) implies that

$$\int_{\Theta^K} \int_{\Theta^K} (f(\boldsymbol{\theta}) - f(\mathbf{y}))^2 \boldsymbol{\eta}_{ij}(\boldsymbol{\theta}) \boldsymbol{\varphi}_i(\boldsymbol{\theta}) \boldsymbol{\mu}_{\text{prior}}(\mathrm{d}\boldsymbol{\theta}) \boldsymbol{\mu}_{\text{prior}}(\mathrm{d}\mathbf{y}) \leq \frac{\mathbb{V}_{\boldsymbol{\mu}_{\sigma_i}}[f] - \Lambda_{ij}^2 \mathbb{V}_{\boldsymbol{\eta}_{ij}}[f]}{\Lambda_{ij}(1 - \Lambda_{ij})}. \quad (26)$$

As for the non-diagonal entries of C , we have

$$\begin{aligned}
C_{ij} &= \int_{\Theta^K} \int_{\Theta^K} (f(\boldsymbol{\theta}) - f(\mathbf{y}))^2 [\Lambda_{ij} \boldsymbol{\eta}_{ij}(\boldsymbol{\theta}) \\
&\quad + (1 - \Lambda_{ij}) \boldsymbol{\varphi}_i(\boldsymbol{\theta})] (\Lambda_{ij} \boldsymbol{\eta}_{ij}(\mathbf{y}) + (1 - \Lambda_{ij}) \boldsymbol{\psi}_j(\mathbf{y})) \boldsymbol{\mu}_{\text{prior}}(d\boldsymbol{\theta}) \boldsymbol{\mu}_{\text{prior}}(d\mathbf{y}) \\
&= 2\Lambda_{ij}^2 \mathbb{V}_{\boldsymbol{\eta}_{ij}}[f] + (1 - \Lambda_{ij})^2 \int_{\Theta^K} \int_{\Theta^K} (f(\boldsymbol{\theta}) - f(\mathbf{y}))^2 \boldsymbol{\varphi}_i(\boldsymbol{\theta}) \boldsymbol{\psi}_j(\mathbf{y}) \boldsymbol{\mu}_{\text{prior}}(d\boldsymbol{\theta}) \boldsymbol{\mu}_{\text{prior}}(d\mathbf{y}) \\
&\quad + \Lambda_{ij}(1 - \Lambda_{ij}) \int_{\Theta^K} \int_{\Theta^K} (f(\boldsymbol{\theta}) - f(\mathbf{y}))^2 (\boldsymbol{\eta}_{ij}(\boldsymbol{\theta}) \boldsymbol{\psi}_j(\mathbf{y}) + \boldsymbol{\eta}_{ij}(\mathbf{y}) \boldsymbol{\varphi}_i(\boldsymbol{\theta})) \boldsymbol{\mu}_{\text{prior}}(d\boldsymbol{\theta}) \boldsymbol{\mu}_{\text{prior}}(d\mathbf{y}).
\end{aligned} \tag{27}$$

We can bound the second term in the previous expression using Cauchy-Schwarz. Let $\mathbf{z} \in \Theta^K$. Then,

$$\begin{aligned}
&\int_{\Theta^K} \int_{\Theta^K} (f(\boldsymbol{\theta}) - f(\mathbf{y}))^2 \boldsymbol{\varphi}_i(\boldsymbol{\theta}) \boldsymbol{\psi}_j(\mathbf{y}) \boldsymbol{\mu}_{\text{prior}}(d\boldsymbol{\theta}) \boldsymbol{\mu}_{\text{prior}}(d\mathbf{y}) \\
&= \int_{\Theta^K} \int_{\Theta^K} \int_{\Theta^K} (f(\boldsymbol{\theta}) - f(\mathbf{z}) + f(\mathbf{z}) - f(\mathbf{y}))^2 \boldsymbol{\varphi}_i(\boldsymbol{\theta}) \boldsymbol{\psi}_j(\mathbf{y}) \boldsymbol{\eta}_{ij}(\mathbf{z}) \boldsymbol{\mu}_{\text{prior}}(d\boldsymbol{\theta}) \boldsymbol{\mu}_{\text{prior}}(d\mathbf{y}) \boldsymbol{\mu}_{\text{prior}}(d\mathbf{z}) \\
&\leq 2 \int_{\Theta^K} \int_{\Theta^K} \int_{\Theta^K} \left((f(\boldsymbol{\theta}) - f(\mathbf{z}))^2 + (f(\mathbf{z}) - f(\mathbf{y}))^2 \right) \boldsymbol{\varphi}_i(\boldsymbol{\theta}) \boldsymbol{\psi}_j(\mathbf{y}) \boldsymbol{\eta}_{ij}(\mathbf{z}) \boldsymbol{\mu}_{\text{prior}}(d\boldsymbol{\theta}) \boldsymbol{\mu}_{\text{prior}}(d\mathbf{y}) \boldsymbol{\mu}_{\text{prior}}(d\mathbf{z}) \\
&= 2 \int_{\Theta^K} \int_{\Theta^K} (f(\boldsymbol{\theta}) - f(\mathbf{z}))^2 \boldsymbol{\varphi}_i(\boldsymbol{\theta}) \boldsymbol{\eta}_{ij}(\mathbf{z}) \boldsymbol{\mu}_{\text{prior}}(d\boldsymbol{\theta}) \boldsymbol{\mu}_{\text{prior}}(d\mathbf{z}) \\
&\quad + 2 \int_{\Theta^K} \int_{\Theta^K} (f(\mathbf{y}) - f(\mathbf{z}))^2 \boldsymbol{\psi}_j(\mathbf{y}) \boldsymbol{\eta}_{ij}(\mathbf{z}) \boldsymbol{\mu}_{\text{prior}}(d\mathbf{y}) \boldsymbol{\mu}_{\text{prior}}(d\mathbf{z}).
\end{aligned} \tag{28}$$

Thus, from equations (26), (27), and (28) we get

$$\begin{aligned}
C_{ij} &\leq 2\Lambda_{ij}^2 \mathbb{V}_{\boldsymbol{\eta}_{ij}}[f] + (2(1 - \Lambda_{ij})^2 + \Lambda_{ij}(1 - \Lambda_{ij})) \left(\int_{\Theta^K} \int_{\Theta^K} (f(\boldsymbol{\theta}) - f(\mathbf{y}))^2 (\boldsymbol{\eta}_{ij}(\boldsymbol{\theta}) \boldsymbol{\psi}_j(\mathbf{y}) \right. \\
&\quad \left. + \boldsymbol{\eta}_{ij}(\mathbf{y}) \boldsymbol{\psi}_i(\boldsymbol{\theta})) \boldsymbol{\mu}_{\text{prior}}(d\boldsymbol{\theta}) \boldsymbol{\mu}_{\text{prior}}(d\mathbf{y}) \right) \\
&= 2\Lambda_{ij}^2 \mathbb{V}_{\boldsymbol{\eta}_{ij}}[f] + (2 - \Lambda_{ij})(1 - \Lambda_{ij}) \left(\mathbb{V}_{\boldsymbol{\mu}_{\sigma_i}}[f] - \Lambda_{ij}^2 \mathbb{V}_{\boldsymbol{\eta}_{ij}}[f] + \mathbb{V}_{\boldsymbol{\mu}_{\sigma_j}}[f] - \Lambda_{ij}^2 \mathbb{V}_{\boldsymbol{\eta}_{ij}}[f] \right) / \Lambda_{ij}(1 - \Lambda_{ij}) \\
&= \frac{2 - \Lambda_{ij}}{\Lambda_{ij}} \left(V_{\boldsymbol{\mu}_{\sigma_i}}[f] + V_{\boldsymbol{\mu}_{\sigma_j}}[f] \right) - 4\Lambda_{ij}(1 - \Lambda_{ij}) \mathbb{V}_{\boldsymbol{\eta}_{ij}}[f] \\
&\leq \frac{2 - \Lambda_{ij}}{\Lambda_{ij}} \left(V_{\boldsymbol{\mu}_{\sigma_i}}[f] + V_{\boldsymbol{\mu}_{\sigma_j}}[f] \right),
\end{aligned} \tag{29}$$

since $\Lambda_{ij} \in (0, 1) \forall i, j$. Finally, from equations (24) and (29) we get that

$$1 = V_{\boldsymbol{\mu}_w}[f] = \frac{1}{2} \sum_{i,j} \frac{1}{|S_K|^2} C_{ij} \leq \frac{1}{2} \frac{1}{|S_K|^2} \sum_{i,j=1}^{|S_K|} \frac{2 - \Lambda_{ij}}{\Lambda_{ij}} \left(V_{\boldsymbol{\mu}_{\sigma_j}}[f] + V_{\boldsymbol{\mu}_{\sigma_i}}[f] \right) \leq \frac{2 - \Lambda_m}{\Lambda_m} \left(\frac{1}{|S_K|} \sum_{i=1}^{|S_K|} \mathbb{V}_{\boldsymbol{\mu}_{\sigma_i}}[f] \right),$$

with $\Lambda_m := \min_{i,j=1,2,\dots,|S_K|} \{\Lambda_{ij}\} > 0$, and $\Lambda_{i,j}$ as in Assumption C3. Notice that we have used (29) for the first inequality, including the case $i = j$, in the previous

equation. This in turn yields the lower bound

$$0 < \frac{\Lambda_m}{2 - \Lambda_m} \leq \left(\frac{1}{|S_K|} \sum_{i \in S_K} \mathbb{V}_{\mu_i}[f] \right).$$

□

Acknowledgements

This publication was supported by funding from King Abdullah University of Science and Technology (KAUST) Office of Sponsored Research (OSR) under award numbers URF/1/2281-01-01 and URF/1/2584-01-01 in the KAUST Competitive Research Grants Programs- Round 3 and 4, respectively, and the Alexander von Humboldt Foundation. Jonas Latz acknowledges support by the Deutsche Forschungsgemeinschaft (DFG) through the TUM International Graduate School of Science and Engineering (IGSSE) within the project 10.02 BAYES. Juan P. Madrigal-Cianci and Fabio Nobile also acknowledge support from the Swiss Data Science Center (SDSC) Grant p18-09.

References

- [1] Christophe Andrieu and Gareth O. Roberts. The pseudo-marginal approach for efficient Monte Carlo computations. *The Annals of Statistics*, 37(2):697–725, 2009.
- [2] R. B. Ash. *Probability and measure theory*. Harcourt/Academic Press, Burlington, MA, 2000.
- [3] Søren Asmussen and Peter W Glynn. *Stochastic simulation: algorithms and analysis*, volume 57. Springer Science & Business Media, 2007.
- [4] John Robert Baxter and Jeffrey S Rosenthal. Rates of convergence for everywhere-positive Markov chains. *Statistics & probability letters*, 22(4):333–338, 1995.
- [5] A. Beskos, A. Jasra, N. Kantas, and A. Thiery. On the convergence of adaptive sequential Monte Carlo methods. *Ann. Appl. Probab.*, 26(2):1111–1146, 2016.
- [6] A. Beskos, A. Jasra, E. Muzaffer, and A.M. Stuart. Sequential Monte Carlo methods for Bayesian elliptic inverse problems. *Stat. Comp.*, 25:727–737, 2015.
- [7] Alexandros Beskos, Mark Girolami, Shiwei Lan, Patrick E Farrell, and Andrew M Stuart. Geometric MCMC for infinite-dimensional inverse problems. *Journal of Computational Physics*, 335:327–351, 2017.

- [8] Steve Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng. *Handbook of Markov chain Monte Carlo*. CRC press, 2011.
- [9] C.Schillings and A.M.Stuart. Analysis of the ensemble Kalman filter for inverse problems. *SINUM*, 55:1264–1290, 2017.
- [10] Tiangang Cui, Kody JH Law, and Youssef M Marzouk. Dimension-independent likelihood-informed MCMC. *Journal of Computational Physics*, 304:109–137, 2016.
- [11] Lisandro Dalcín, Rodrigo Paz, and Mario Storti. MPI for python. *Journal of Parallel and Distributed Computing*, 65(9):1108–1115, 2005.
- [12] Guillaume Desjardins, Aaron Courville, Yoshua Bengio, Pascal Vincent, and Olivier Delalleau. Tempered Markov chain Monte Carlo for training of restricted boltzmann machines. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 145–152, 2010.
- [13] Ben Mansour Dia. A continuation method in Bayesian inference. *arXiv preprint arXiv:1911.11650*, 2019.
- [14] Tim J Dodwell, Chris Ketelsen, Robert Scheichl, and Aretha L Teckentrup. A hierarchical multilevel markov chain monte carlo algorithm with applications to uncertainty quantification in subsurface flow. *SIAM/ASA Journal on Uncertainty Quantification*, 3(1):1075–1108, 2015.
- [15] JD Doll, Nuria Plattner, David L Freeman, Yufei Liu, and Paul Dupuis. Rare-event sampling: Occupation-based performance measures for parallel tempering and infinite swapping Monte Carlo methods. *The Journal of chemical physics*, 137(20):204112, 2012.
- [16] Paul Dupuis, Yufei Liu, Nuria Plattner, and Jimmie D Doll. On the infinite swapping limit for parallel tempering. *Multiscale Modeling & Simulation*, 10(3):986–1022, 2012.
- [17] David J Earl and Michael W Deem. Parallel tempering: Theory, applications, and new perspectives. *Physical Chemistry Chemical Physics*, 7(23):3910–3916, 2005.
- [18] Heikki Haario, Marko Laine, Antonietta Mira, and Eero Saksman. DRAM: efficient adaptive MCMC. *Statistics and computing*, 16(4):339–354, 2006.
- [19] Heikki Haario, Eero Saksman, and Johanna Tamminen. An adaptive Metropolis algorithm. *Bernoulli*, 7(2):223–242, 2001.
- [20] Martin Hairer, Andrew M Stuart, Sebastian J Vollmer, et al. Spectral gaps for a Metropolis–Hastings algorithm in infinite dimensions. *The Annals of Applied Probability*, 24(6):2455–2490, 2014.

- [21] W Keith Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Oxford University Press*, 1970.
- [22] C. Kahle, K. Lam, J. Latz, and E. Ullmann. Bayesian parameter identification in Cahn–Hilliard models for biological growth. *SIAM/ASA Journal on Uncertainty Quantification*, 7(2):526–552, 2019.
- [23] N. Kantas, A. Beskos, and A. Jasra. Sequential Monte Carlo Methods for High-Dimensional Inverse Problems: A case study for the Navier–Stokes equations. *SIAM/ASA J. Uncertain. Quantif.*, 2(1):464–489, 2014.
- [24] Mateusz Krzysztof Łącki and Błażej Miasojedow. State-dependent swap strategies and automatic reduction of number of temperatures in adaptive parallel tempering algorithm. *Statistics and Computing*, 26(5):951–964, 2016.
- [25] Jonas Latz. On the well-posedness of Bayesian inverse problems. *arXiv e-prints*, page arXiv:1902.10257, 2019.
- [26] Jonas Latz, Iason Papaioannou, and Elisabeth Ullmann. Multilevel Sequential² Monte Carlo for Bayesian inverse problems. *Journal of Computational Physics*, 368:154 – 178, 2018.
- [27] Anders Logg, Kent-Andre Mardal, and Garth Wells. *Automated solution of differential equations by the finite element method: The FEniCS book*, volume 84. Springer Science & Business Media, 2012.
- [28] Jianfeng Lu and Eric Vanden-Eijnden. Infinite swapping replica exchange molecular dynamics leads to a simple simulation patch using mixture potentials. *The Journal of chemical physics*, 138(8):084105, 2013.
- [29] Neal Madras, Dana Randall, et al. Markov chain decomposition for convergence rate analysis. *The Annals of Applied Probability*, 12(2):581–606, 2002.
- [30] E Marinari and G Parisi. Simulated tempering: A new Monte Carlo scheme. *Europhysics Letters (EPL)*, 19(6):451–458, jul 1992.
- [31] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.
- [32] Sean P Meyn and Richard L Tweedie. *Markov chains and stochastic stability*. Springer Science & Business Media, 2012.
- [33] Błażej Miasojedow, Eric Moulines, and Matti Vihola. An adaptive parallel tempering algorithm. *Journal of Computational and Graphical Statistics*, 22(3):649–664, 2013.

- [34] Mohammad Motamed and Daniel Appelo. Wasserstein metric-driven bayesian inversion with applications to signal processing. *International Journal for Uncertainty Quantification*, 9(4), 2019.
- [35] Nuria Plattner, JD Doll, Paul Dupuis, Hui Wang, Yufei Liu, and JE Gubernatis. An infinite swapping approach to the rare-event sampling problem. *The Journal of chemical physics*, 135(13):134111, 2011.
- [36] Alfio Quarteroni and Silvia Quarteroni. *Numerical models for differential problems*, volume 2. Springer, 2009.
- [37] Gareth O Roberts and Jeffrey S Rosenthal. Optimal scaling of discrete approximations to Langevin diffusions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 60(1):255–268, 1998.
- [38] Daniel Rudolf. Explicit error bounds for Markov chain Monte Carlo. *Dissertationes Math.*, 485:1–93, 2012.
- [39] Marc Van Der Sluys, Vivien Raymond, Ilya Mandel, Christian Röver, Nelson Christensen, Vicky Kalogera, Renate Meyer, and Alberto Vecchio. Parameter estimation of spinning binary inspirals using Markov chain Monte Carlo. *Classical and Quantum Gravity*, 25(18):184011, 2008.
- [40] Jasper A Vrugt, CJF Ter Braak, CGH Diks, Bruce A Robinson, James M Hyman, and Dave Higdon. Accelerating Markov chain Monte Carlo simulation by differential evolution with self-adaptive randomized subspace sampling. *International Journal of Nonlinear Sciences and Numerical Simulation*, 10(3):273–290, 2009.
- [41] Dawn B Woodard, Scott C Schmidler, Mark Huber, et al. Conditions for rapid mixing of parallel and simulated tempering on multimodal distributions. *The Annals of Applied Probability*, 19(2):617–640, 2009.