Generalized parallel tempering on Bayesian inverse problems

Jonas Latz[1] · Juan P. Madrigal-Cianci[2] · Fabio Nobile[2] · Raúl Tempone[3,4]

Abstract

In the current work we present two generalizations of the Parallel Tempering algorithm in the context of discrete-time Markov chain Monte Carlo methods for Bayesian inverse problems. These generalizations use state-dependent swapping rates, inspired by the so-called continuous time Infinite Swapping algorithm presented in Plattner et al. (J Chem Phys 135(13):134111, 2011). We analyze the reversibility and ergodicity properties of our generalized PT algorithms. Numerical results on sampling from different target distributions, show that the proposed methods significantly improve sampling efficiency over more traditional sampling algorithms such as Random Walk Metropolis, preconditioned Crank–Nicolson, 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 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 compli-

cated target distributions arbitrarily long. Some recent works address these issues by employing Markov transitions kernels that use geometric information (Beskos et al. 2017); 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* (Dia 2019; Earl and Deem 2005; Latz et al. 2018; Miasojedow et al. 2013; Vrugt et al. 2009). Of particular importance for the work presented here is the Parallel Tempering (PT) algorithm (Earl and Deem 2005; Łącki and Miasojedow 2016; Miasojedow et al. 2013) (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 sam-

✉ Juan P. Madrigal-Cianci
juan.madrigalcianci@epfl.ch

[1] Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Cambridge, UK

[2] SB-MATH-CSQI, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland

[3] Computer, Electrical and Mathematical Sciences and Engineering, KAUST, Thuwal, Saudi Arabia

[4] Alexander von Humboldt professor in Mathematics of Uncertainty Quantification, RWTH Aachen University, Aachen, Germany

plers of getting "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 (Earl and Deem 2005).

Tempering ideas have been successfully used to sample from posterior distributions arising in different fields of science, ranging from astrophysics to machine learning (Desjardins et al. 2010; Earl and Deem 2005; Miasojedow et al. 2013; Van Der Sluys et al. 2008). The works (Madras and Randall 2002; Woodard et al. 2009) 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 (Marinari and Parisi 1992) 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 (Beskos et al. 2016, 2015; Kahle et al. 2019; Kantas et al. 2014; Latz et al. 2018) and Ensemble Kalman Inversion (Schillings and Stuart 2017).

A generalization over the PT approach, originating from the molecular dynamics community, is the so-called *Infinite Swapping (IS)* algorithm (Dupuis et al. 2012; Plattner et al. 2011). 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 Dupuis et al. (2012) 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 (Doll et al. 2012; Lu and Vanden-Eijnden 2013; Plattner et al. 2011); 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 Dupuis et al. (2012), 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 begin by presenting a generalized framework for discrete time PT in the context of MCMC for BIP, and then proceed to propose, analyze and implement two novel state-dependent PT algorithms inspired by the ideas presented in Dupuis et al. (2012).

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 an un-tempered, (single temperature), version of the underlying MCMC algorithm, and standard PT. For the base method to sample at the cold temperature level, we use Random Walk Metropolis (RWM) (Sects. 5.3–5.6) or preconditioned Crank-Nicolson (Sect. 5.7), however, we emphasize that our methods can be used together with any other, more efficient base sampler. Experimental results show improvements in terms of computational efficiency of GPT over un-tempered RWM and standard PT, thus making the proposed methods attractive from a computational perspective. From an implementation perspective, the swapping component of our proposed methods is rejection-free, thus effectively eliminating some tuning parameters on the PT algorithm, such as swapping frequency.

We notice that a PT algorithm with state-dependent swapping probabilities has been proposed in Łącki and Miasojedow (2016), 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 Rudolf (2012) 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 Woodard et al. (2009). 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 Sect. 3, we provide a brief review of (standard) PT (Sect. 3.2), and introduce the two versions of the GPT algorithm in Sects. 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 Sect. 4, we recall some of the standard theory of Markov chains in Sect. 4.1 and state the main theoretical results of the current work (Theorems 1 and 2) in Sect. 4.2. The proof of these Theorems is given by a series of Propositions in Sect. 4.2. We present some numerical experiments in Sect. 5, and draw some conclusions in Sect. 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 W_1, W_2 be two separable 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

$$
\boldsymbol{\mu}(A) = (\mu_1 \times \mu_2)(A)
$$
$$
= \iint_A \pi_{\mu_1}(\theta_1) \pi_{\mu_2}(\theta_2) \nu_{W_1}(d\theta_1) \nu_{W_2}(d\theta_2),
$$

for all $A \in \mathcal{B}(W_1 \times W_2)$. 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 $p : W \times \mathcal{B}(W) \to [0, 1]$ 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 on measures as $\nu \mapsto \nu P \in \mathcal{M}(W)$, and on functions as $f \mapsto Pf$, with Pf measurable on $(W, \mathcal{B}(W))$, such that

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

Let 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, $\boldsymbol{\nu}\mathbf{P}$ is the measure on $(W_1 \times W_2, \mathcal{B}(W_1 \times W_2))$ that satisfies

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

for all $A_1 \in \mathcal{B}(W_1)$ and $A_2 \in \mathcal{B}(W_2)$. Moreover, $(\mathbf{P}f) : W_1 \times W_2 \to \mathbb{R}$ 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 $f : W_1 \times W_2 \to \mathbb{R}$.

We say that a Markov operator P (resp. $\mathbf{P}$) is *invariant* with respect to a measure ν (resp. $\boldsymbol{\nu}$) if $\nu P = \nu$ (resp. $\boldsymbol{\nu}\mathbf{P} = \boldsymbol{\nu}$). 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

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

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 $P_c := P_1 P_2 \ldots P_K$, we say that P_c is *palindromic* if $P_1 = P_K$, $P_2 = P_{K-1}$, ..., $P_k = P_{K-k+1}$, $k = 1, 2 \ldots, K$. It is known (see, e.g., Brooks et al. (2011) 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:

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

Here ε 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. We will assume that the data y is finite dimensional, i.e., $Y = \mathbb{R}^{d_y}$ for some $d_y \geq 1$, and that $\theta \sim \mu_{\text{pr}}$. Furthermore, we define the *potential* function $\Phi(\theta; y) : \Theta \times Y \mapsto \mathbb{R}$ as

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

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 $\|y - \mathcal{F}(\theta)\|_Y$. The solution to the Bayesian

inverse problem is given by (see, for example, Latz et al. (2019) Theorem 2.5)

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

where μ (with corresponding ν_Θ-density π) is referred to as the *posterior measure* and

$$Z := \int_\Theta \exp(-\Phi(\theta; y)) \mu_{\mathrm{pr}}(\mathrm{d}\theta).$$

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 Eq. (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, for example, the Metropolis-Hastings algorithm, as detailed in the next section. Once samples $\{\theta^{(n)}\}_{n=0}^N$ have been obtained, 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

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

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

2.3 Metropolis-Hastings and tempering

We briefly recall the Metropolis-Hastings algorithm (Hastings 1970; Metropolis et al. 1953). Let $q_{\mathrm{prop}} : \Theta \times \mathcal{B}(\Theta) \mapsto [0, 1]$ be an auxiliary kernel. For $n = 1, 2, \ldots$, a candidate state θ^* is sampled from $q_{\mathrm{prop}}(\theta^n, \cdot)$, 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},

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

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* $p : \Theta \times \mathcal{B}(\Theta) \mapsto [0, 1]$

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

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 $q_{\mathrm{prop}}(\theta^n, \cdot)$. Such is the case for *random walk Metropolis* or *preconditioned Crank Nicolson*, where $q_{\mathrm{prop}}(\theta^n, \cdot) = \mathcal{N}(\theta^n, \Sigma)$ or $q_{\mathrm{prop}}(\theta^n, \cdot) = \mathcal{N}(\sqrt{1 - \rho^2}\theta^n, \rho\Sigma)$, $0 < \rho < 1$, 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 (Earl and Deem 2005). 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 $(\Theta, \mathcal{B}(\Theta))$, $k = 1, \ldots, K$, 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 $1 = T_1 < T_2 < \cdots < T_K \leq \infty$. In a Bayesian setting, μ_{pr} corresponds to the prior measure and μ_k, $k = 1, \ldots, K$ correspond to posterior measures associated to different temperatures. Denoting by π_k the μ_{pr}-density of μ_k, we set

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

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

We say that for $k = 2, \ldots, K$, 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 $T_k \to \infty$. 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}, $k \in \{1, \ldots, K-1\}$. 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 swaps states between two chains μ_i, μ_j, with i, j chosen randomly and uniformly from the index set $\{1, 2, \ldots, K\}$. In the next section we present the generalized PT algorithms which swap states according to a random

---

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

permutation of the indices drawn from a state dependent probability.

Algorithm 1 Standard PT.

function STANDARD PT($N, \{p_k\}_{k=1}^K, \{\pi_k\}_{k=1}^K, \mu_{\text{pr}}$)
  Sample $\theta_k^{(1)} \sim \mu_{\text{pr}}, \ k = 1, \ldots, K$
  \# Do one step of MH on each chain
  for $n = 1, 2, \ldots, N - 1$ **do**
    for $k = 1, \ldots, K$ **do**
      Sample $\theta_k^{(n+1)} \sim p_k(\theta_k^{(n)}, \cdot)$
    end for
    \# Swap states
    for $k = 1, 2, \ldots, K - 1$ **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 $\{1, \ldots, K\}$. 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 Dupuis et al. (2012) 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 associated product σ-algebra $\mathcal{B}^K := \bigotimes_{k=1}^K \mathcal{B}(\Theta)$, as well as the product measures on $(\Theta^K, \mathcal{B}^K)$

$$\boldsymbol{\mu} := \times_{k=1}^K \mu_k, \tag{6}$$

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

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

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 **p** and **q**, to some initial distribution $\boldsymbol{\nu}$, usually chosen to be the prior $\boldsymbol{\mu}_{\text{pr}}$. Each kernel acts as follows. Given the current state $\boldsymbol{\theta}^n = (\theta_1^n, \ldots, \theta_K^n)$, the kernel **p**, which we will call the *standard MCMC kernel*, proposes a new, intermediate state $\tilde{\boldsymbol{\theta}}^{n+1} = (\tilde{\theta}_1^{n+1}, \ldots, \tilde{\theta}_K^{n+1})$, possibly following the Metropolis-Hastings algorithm (or any other algorithm that generates a $\boldsymbol{\mu}$-invariant Markov operator). The Markov kernel **p** is a product kernel, meaning that each component $\tilde{\theta}_k^n, k = 1 \ldots, K$, is generated independently of the others. Then, the *swapping kernel* **q** proposes a new state $\boldsymbol{\theta}^{n+1} = (\theta_1^{n+1}, \ldots, \theta_K^{n+1})$ 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 **q**.

3.1 The swapping kernel q

Define $\mathscr{S}_K$ as the collection of all the bijective maps from $\{1, 2, \ldots, K\}$ to itself, i.e., the set of all $K!$ possible permutations of id$:=\{1, \ldots, K\}$. Let $\sigma \in \mathscr{S}_K$ be a permutation, and define the swapped state $\boldsymbol{\theta}_\sigma := (\theta_{\sigma(1)}, \ldots, \theta_{\sigma(K)})$, and the inverse permutation $\sigma^{-1} \in \mathscr{S}_K$ such that $\sigma \circ \sigma^{-1} = \sigma^{-1} \circ \sigma = $ id. In addition, let $S_K \subseteq \mathscr{S}_K$ be any subset of $\mathscr{S}_K$ 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 $|S_K|$.

Example 1 As a simple example, consider a Standard PT as in Algorithm 1 with $K = 4$. In this case, we attempt to swap two contiguous temperatures T_i and T_{i+1}, $i = 1, 2, 3$. Thus, S_K is the set of permutations $\{\sigma_{1,2}, \sigma_{2,3}, \sigma_{3,4}\}$ with:

$$\sigma_{1,2} = (2, 1, 3, 4),$$
$$\sigma_{2,3} = (1, 3, 2, 4),$$
$$\sigma_{3,4} = (1, 2, 4, 3).$$

Notice that each permutation is its own inverse; for example:

$$\sigma_{1,2}(\sigma_{1,2}) = \sigma_{1,2}((2, 1, 3, 4)) = (1, 2, 3, 4) = \text{id}.$$

To define the swapping kernel **q**, we first need to define the swapping ratio and swapping acceptance probability.

Definition 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 2 (Swapping acceptance probability)
Let $\boldsymbol{\theta} \in \Theta^K$ and σ, $\sigma^{-1} \in S_K$. 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 **q**.

Definition 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

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

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 1 *The Markov kernel* **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

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

Thus,

$$\int_A q(\boldsymbol{\theta}, B)\boldsymbol{\mu}(\mathrm{d}\boldsymbol{\theta}) = \underbrace{\sum_{\sigma \in S_K} \int_A r(\boldsymbol{\theta}, \sigma)\alpha_{\text{swap}}(\boldsymbol{\theta}, \sigma)\delta_{\boldsymbol{\theta}_\sigma}(B)\pi(\boldsymbol{\theta})\boldsymbol{\mu}_{\text{pr}}(\mathrm{d}\boldsymbol{\theta})}_{I}$$
$$+ \underbrace{\sum_{\sigma \in S_K} \int_A r(\boldsymbol{\theta}, \sigma)\big(1 - \alpha_{\text{swap}}(\boldsymbol{\theta}, \sigma)\big)\delta_{\boldsymbol{\theta}}(B)\pi(\boldsymbol{\theta})\boldsymbol{\mu}_{\text{pr}}(\mathrm{d}\boldsymbol{\theta})}_{II}.$$

Let $A_\sigma := \{z \in \Theta^K : z_{\sigma^{-1}} \in A\}$, and, for notational simplicity, write $\min\{a, b\} = \{a \wedge b\}$, $a, b \in \mathbb{R}$. From I, we have:

$$I = \sum_{\sigma \in S_K} \int_A \left\{1 \wedge \frac{\pi(\boldsymbol{\theta}_\sigma)r(\boldsymbol{\theta}_\sigma, \sigma^{-1})}{\pi(\boldsymbol{\theta})r(\boldsymbol{\theta}, \sigma)}\right\} r(\boldsymbol{\theta}, \sigma)\pi(\boldsymbol{\theta})\delta_{\boldsymbol{\theta}_\sigma}(B)\boldsymbol{\mu}_{\text{pr}}(\mathrm{d}\boldsymbol{\theta})$$
$$= \sum_{\sigma \in S_K} \int_A \left\{1 \wedge \frac{\pi(\boldsymbol{\theta})r(\boldsymbol{\theta}, \sigma)}{\pi(\boldsymbol{\theta}_\sigma)r(\boldsymbol{\theta}_\sigma, \sigma^{-1})}\right\} r(\boldsymbol{\theta}_\sigma, \sigma^{-1})\pi(\boldsymbol{\theta}_\sigma)\delta_{\boldsymbol{\theta}_\sigma}(B)\boldsymbol{\mu}_{\text{pr}}(\mathrm{d}\boldsymbol{\theta}).$$

Then, noticing that $\boldsymbol{\mu}_{\text{pr}}$ is permutation invariant, we get

$$I = \sum_{\sigma \in S_K} \int_{A_\sigma} \left\{1 \wedge \frac{\pi(\boldsymbol{\theta}_{\sigma^{-1}})r(\boldsymbol{\theta}_{\sigma^{-1}}, \sigma)}{\pi(\boldsymbol{\theta})r(\boldsymbol{\theta}, \sigma^{-1})}\right\}$$
$$\times r(\boldsymbol{\theta}, \sigma^{-1})\pi(\boldsymbol{\theta})\delta_{\boldsymbol{\theta}}(B)\boldsymbol{\mu}_{\text{pr}}(\mathrm{d}\boldsymbol{\theta})$$
$$= \sum_{\sigma \in S_K} \int_{A_\sigma \cap B} \left\{1 \wedge \frac{\pi(\boldsymbol{\theta}_{\sigma^{-1}})r(\boldsymbol{\theta}_{\sigma^{-1}}, \sigma)}{\pi(\boldsymbol{\theta})r(\boldsymbol{\theta}, \sigma^{-1})}\right\}$$
$$\times r(\boldsymbol{\theta}, \sigma^{-1})\pi(\boldsymbol{\theta})\delta_{\boldsymbol{\theta}}(B)\boldsymbol{\mu}_{\text{pr}}(\mathrm{d}\boldsymbol{\theta})$$
$$= \sum_{\sigma \in S_K} \int_B \left\{1 \wedge \frac{\pi(\boldsymbol{\theta}_{\sigma^{-1}})r(\boldsymbol{\theta}_{\sigma^{-1}}, \sigma)}{\pi(\boldsymbol{\theta})r(\boldsymbol{\theta}, \sigma^{-1})}\right\}$$
$$\times r(\boldsymbol{\theta}, \sigma^{-1})\pi(\boldsymbol{\theta})\delta_{\boldsymbol{\theta}}(A_\sigma)\boldsymbol{\mu}_{\text{pr}}(\mathrm{d}\boldsymbol{\theta})$$
$$= \sum_{\sigma \in S_K} \int_B \left\{1 \wedge \frac{\pi(\boldsymbol{\theta}_{\sigma^{-1}})r(\boldsymbol{\theta}_{\sigma^{-1}}, \sigma)}{\pi(\boldsymbol{\theta})r(\boldsymbol{\theta}, \sigma^{-1})}\right\}$$
$$\times r(\boldsymbol{\theta}, \sigma^{-1})\pi(\boldsymbol{\theta})\delta_{\boldsymbol{\theta}_{\sigma^{-1}}}(A)\boldsymbol{\mu}_{\text{pr}}(\mathrm{d}\boldsymbol{\theta})$$
$$= \sum_{\sigma \in S_K} \int_B r(\boldsymbol{\theta}, \sigma^{-1})\pi(\boldsymbol{\theta})\alpha_{\text{swap}}(\boldsymbol{\theta}, \sigma^{-1})\delta_{\boldsymbol{\theta}_{\sigma^{-1}}}(A)\boldsymbol{\mu}_{\text{pr}}(\mathrm{d}\boldsymbol{\theta})$$
$$= \sum_{\sigma \in S_K} \int_B r(\boldsymbol{\theta}, \sigma)\pi(\boldsymbol{\theta})\alpha_{\text{swap}}(\boldsymbol{\theta}, \sigma)\delta_{\boldsymbol{\theta}_\sigma}(A)\boldsymbol{\mu}_{\text{pr}}(\mathrm{d}\boldsymbol{\theta}).$$

For the second term II we simply have

$$II = \sum_{\sigma \in S_K} \int_A r(\boldsymbol{\theta}, \sigma)(1 - \alpha_{\text{swap}}(\boldsymbol{\theta}, \sigma))\delta_{\boldsymbol{\theta}}(B)\boldsymbol{\pi}(\boldsymbol{\theta})\boldsymbol{\mu}_{\text{pr}}(\text{d}\boldsymbol{\theta})$$

$$= \sum_{\sigma \in S_K} \int_{A \cap B} r(\boldsymbol{\theta}, \sigma)(1 - \alpha_{\text{swap}}(\boldsymbol{\theta}, \sigma))\delta_{\boldsymbol{\theta}}(B)\boldsymbol{\pi}(\boldsymbol{\theta})\boldsymbol{\mu}_{\text{pr}}(\text{d}\boldsymbol{\theta})$$

$$= \sum_{\sigma \in S_K} \int_B r(\boldsymbol{\theta}, \sigma)(1 - \alpha_{\text{swap}}(\boldsymbol{\theta}, \sigma))\delta_{\boldsymbol{\theta}}(A)\boldsymbol{\pi}(\boldsymbol{\theta})\boldsymbol{\mu}_{\text{pr}}(\text{d}\boldsymbol{\theta}).$$

$\square$

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 ith 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 $(1, 2, \ldots, K)$, which only permutes the ith 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 $S_K = \{\sigma_{i,j}, \ i, j = 1, \ldots, K\}$ 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:

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

$$\alpha_{\text{swap}}^{(\text{PT})}(\boldsymbol{\theta}, \sigma) = 0, \ \sigma \neq \sigma_{i,j}.$$

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

Definition 4 (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 ith and j^{th} chains as $\mathbf{q}_{i,j}^{(\text{PT})} : \Theta^K \times \mathcal{B}^K \mapsto [0, 1]$ given by

$$\mathbf{q}_{i,j}^{(\text{PT})}(\boldsymbol{\theta}, B) = \sum_{\sigma \in S_K} r_{i,j}^{(\text{PT})}(\boldsymbol{\theta}, \sigma) \left((1 - \alpha_{\text{swap}}^{(\text{PT})}(\boldsymbol{\theta}, \sigma))\delta_{\boldsymbol{\theta}}(B) \right.$$

$$\left. + \alpha_{\text{swap}}^{(\text{PT})}(\boldsymbol{\theta}, \sigma)\delta_{\boldsymbol{\theta}_\sigma}(B) \right)$$

$$= \left(1 - \min \left\{ 1, \frac{\boldsymbol{\pi}(\boldsymbol{\theta}_{\sigma_{i,j}})}{\boldsymbol{\pi}(\boldsymbol{\theta})} \right\} \delta_{\boldsymbol{\theta}}(B) \right)$$

$$+ \min \left\{ 1, \frac{\boldsymbol{\pi}(\boldsymbol{\theta}_{\sigma_{i,j}})}{\boldsymbol{\pi}(\boldsymbol{\theta})} \right\} \delta_{\boldsymbol{\theta}_{\sigma_{i,j}}}(B), \quad \forall B \in \mathcal{B}^K.$$

In practice, however, the PT algorithm considers various sequential swaps between chains, which can be understood by applying the composition of kernels $\mathbf{q}_{i,j}^{(\text{PT})}\mathbf{q}_{k,\ell}^{(\text{PT})} \ldots$ at every swapping step. In its most common form Brooks et al. (2011); Earl and Deem (2005); Miasojedow et al. (2013), 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

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

Moreover, the algorithm described in Earl and Deem (2005), proposes to swap states every $N_s \geq 1$ steps of MCMC. The complete kernel for the PT kernel is then given by Brooks et al. (2011); Earl and Deem (2005); Miasojedow et al. (2013)

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

where $\mathbf{p}$ is a standard reversible Markov transition kernel used to evolve the individual chains independently.

Remark 1 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 $\boldsymbol{\mu}$ is to consider the palindromic form

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

where RPT stands for *Reversible Parallel Tempering*. In practice, there is not much difference between $\mathbf{p}^{(\text{RPT})}$ and $\mathbf{p}^{(\text{PT})}$, however, under the additional assumption of geometric ergodicity of the chain (c.f Sect. 4) having a reversible kernel is useful to compute explicit error bounds on the non-asymptotic mean square error of an ergodic estimator (Rudolf 2012).

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 $r^{(\mathrm{UW})} : \Theta^K \times S_K \mapsto [0, 1]$ be a function defined as

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

Clearly, (9) is a swapping ratio according to Definition 1. As such, given some state $\boldsymbol{\theta} \in \Theta^K$, $r^{(\mathrm{UW})}(\boldsymbol{\theta}, \sigma)$ 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 $\alpha_{\mathrm{swap}}^{(\mathrm{UW})}(\boldsymbol{\theta}, \sigma)$, given by

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

Thus, we can define the swapping kernel for the UGPT algorithm, which takes the form of (7), with the particular choice of $r(\boldsymbol{\theta}, \sigma) = r^{(\mathrm{UW})}(\boldsymbol{\theta}, \sigma)$ and

$$\alpha_{\mathrm{swap}}(\boldsymbol{\theta}, \sigma) = \alpha_{\mathrm{swap}}^{(\mathrm{UW})}(\boldsymbol{\theta}, \sigma).$$

Notice that $\alpha_{\mathrm{swap}}^{(\mathrm{UW})}(\boldsymbol{\theta}, \sigma) = 1, \forall \sigma \in S_K$. Indeed, if we further examine Eq. (10), we see that

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

$$= \frac{\pi(\boldsymbol{\theta}_\sigma)}{\pi(\boldsymbol{\theta})} \cdot \frac{\pi(\boldsymbol{\theta})}{\pi(\boldsymbol{\theta}_\sigma)} = 1.$$

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

$$\mathbf{q}^{(\mathrm{UW})}(\boldsymbol{\theta}, B) = \sum_{\sigma \in S_K} r^{(\mathrm{UW})}(\boldsymbol{\theta}, \sigma) \delta_{\boldsymbol{\theta}_\sigma}(B),$$

$\forall \boldsymbol{\theta} \in \Theta^K$, $B \in \mathcal{B}^K$. Applying this swapping kernel successively with the kernel $\mathbf{p} = p_1 \times p_2 \times \ldots p_K$ in the order $\mathbf{q}^{(\mathrm{UW})} \mathbf{p} \mathbf{q}^{(\mathrm{UW})} =: \mathbf{p}^{(\mathrm{UW})}$ gives what we call *Unweighted Generalized Parallel Tempering kernel* $\mathbf{p}^{(\mathrm{UW})}$. Lastly, we write the UGPT in operator form as

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

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

Proposition 2 *Suppose that, for any $k = 1, 2, \ldots, K$, p_k is μ_k-reversible. Then, $\mathbf{p} = p_1 \times \cdots \times p_K$ is reversible with respect to μ.*

Proof We prove reversibility by confirming that Eq. (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 $A_1, \ldots, A_K, B_1, \ldots, B_K \in \mathcal{B}(\Theta)$. Then,

$$\int_A \pi(\boldsymbol{\theta}) \mathbf{p}(\boldsymbol{\theta}, B) \mathrm{d}\boldsymbol{\theta} = \prod_{k=1}^K \int_{A_k} \pi(\theta_k) p(\theta_k, B_k) \mathrm{d}\theta_k$$

$$= \prod_{k=1}^K \int_{B_k} \pi(\theta_k) p(\theta_k, A_k) \mathrm{d}\theta_k$$

$$= \int_B \pi(\boldsymbol{\theta}) \mathbf{p}(\boldsymbol{\theta}, A) \mathrm{d}\boldsymbol{\theta}.$$

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 Ash (2000) section 1.3.10, 2.6.3, for details. □

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

Proposition 3 *(Reversibility of the UGPT chain) Suppose that, for any $k = 1, 2, \ldots, K$, p_k is μ_k-reversible. Then, the Markov chain generated by $\mathbf{p}^{(\mathrm{UW})}$ is $\boldsymbol{\mu}$-reversible.*

Proof It follows from Propositions 1 and 2 that the kernels $\mathbf{q}^{(\mathrm{UW})}$ and $\mathbf{p}$ are $\boldsymbol{\mu}$-reversible. Furthermore, since $\mathbf{p}^{(\mathrm{UW})}$ is a *palindromic* composition of kernels, each of which is reversible with respect to $\boldsymbol{\mu}$, then, $\mathbf{p}^{(\mathrm{UW})}$ is reversible with respect to $\boldsymbol{\mu}$ Brooks et al. (2011). □

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

Algorithm 2 Unweighted Generalized Parallel Tempering.

function GENERALIZED PARALLEL TEMPERING($\mathbf{p}, N, \boldsymbol{\nu}$)
    Sample $\boldsymbol{\theta}^{(1)} \sim \boldsymbol{\nu}$
    for $n = 1, 2, \ldots, N - 1$ **do**
        # First swapping kernel
        Sample $\boldsymbol{\theta}_\sigma^{(n)} \sim \mathbf{q}^{(\mathrm{UW})}(\boldsymbol{\theta}^{(n)}, \cdot)$
        # Markov transition kernel $\mathbf{p}$
        Sample $z^{(n+1)} \sim \mathbf{p}(\boldsymbol{\theta}_\sigma^{(n)}, \cdot)$ kernel
        # Second swapping kernel
        Sample $\boldsymbol{\theta}^{(n+1)} \sim \mathbf{q}^{(\mathrm{UW})}(z^{(n+1)}, \cdot)$
    end for
    Output $\{\theta_1^{(n)}\}_{n=1}^N$
end function

Remark 2 In practice, one does not need to perform $|S_K|$ posterior evaluations when computing $r^{(\mathrm{UW})}(\boldsymbol{\theta}^n, \cdot)$, rather "just" K of them. Indeed, since $\pi_j(\theta_k^n) \propto \pi(\theta_k)^{T_j}$, $k, j = 1, 2, \ldots, K$, we just need to store the values of $\pi(\theta_k^n)$, $k =$

$1, 2, \ldots, K$, 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 $\widehat{\mathcal{Q}}_{(\text{UW})}$:

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

3.3.1 A comment on the pairwise state-dependent PT method of Łącki and Miasojedow (2016)

The work Łącki and Miasojedow (2016) 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 $(1, 2, \ldots, K)$. Given a current state $\boldsymbol{\theta} \in \Theta^K$, the PSDPT algorithm samples a pairwise permutation $\boldsymbol{\theta}_{\sigma_{i,j}} \in S_{K,\text{pairwise}}$ with probability $r_{i,j}^{(\text{PSDPT})}(\boldsymbol{\theta}, \sigma_{i,j})$ given by

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

and then accepts this swap with probability

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

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 Dupuis et al. (2012); Plattner et al. (2011), 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:

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

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

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

Moreover, we define the swapping weights

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

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

Definition 6 We define the *Weighted Generalized Parallel Tempering* kernel $\mathbf{p}^{(\text{W})} : \Theta^K \times B^K \mapsto [0, 1]$ 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 4 that the resulting Markov chain has invariant measure

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

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

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

and a similar average and then tensorization representation applies to $\boldsymbol{\pi}_{\text{W}}$. 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 4 (*Reversibility of the WGPT chain*) *Suppose that, for any* $k = 1, 2, \ldots, K$ p_k *is* μ_k-*reversible. Then, the Markov chain generated by* $\mathbf{p}^{(\text{W})}$ *is* $\boldsymbol{\mu}_W$-*reversible.*

Proof We show reversibility by showing that (1) holds true. Thus, for $\boldsymbol{\theta} \in \Theta^K$, $A, B \in \mathcal{B}^K$, 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:

$$
\int_A \mathbf{p}^{(\mathrm{W})}(\boldsymbol{\theta}, B)\boldsymbol{\pi}_{\mathrm{W}}(\boldsymbol{\theta})\boldsymbol{\mu}_{\mathrm{pr}}(\mathrm{d}\boldsymbol{\theta})
$$

$$
= \int_A \left[\sum_{\sigma \in S_K} w_\sigma(\boldsymbol{\theta})\mathbf{p}_\sigma(\boldsymbol{\theta}, B) \right] \frac{\sum_{\rho \in S_K} \boldsymbol{\pi}_\rho(\boldsymbol{\theta})}{|S_K|}\boldsymbol{\mu}_{\mathrm{pr}}(\mathrm{d}\boldsymbol{\theta})
$$

$$
= \int_A \left[\sum_{\sigma \in S_K} \frac{\boldsymbol{\pi}_\sigma(\boldsymbol{\theta})}{\sum_{\sigma' \in S_K} \boldsymbol{\pi}_{\sigma'}(\boldsymbol{\theta})}\mathbf{p}_\sigma(\boldsymbol{\theta}, B) \right]
$$

$$
\times \frac{\sum_{\rho \in S_K} \boldsymbol{\pi}_\rho(\boldsymbol{\theta})}{|S_K|}\boldsymbol{\mu}_{\mathrm{pr}}(\mathrm{d}\boldsymbol{\theta})
$$

$$
= \frac{1}{|S_K|} \sum_{\sigma \in S_K} \int_A \boldsymbol{\pi}_\sigma(\boldsymbol{\theta})\mathbf{p}_\sigma(\boldsymbol{\theta}, B)\boldsymbol{\mu}_{\mathrm{pr}}(\mathrm{d}\boldsymbol{\theta}) = I.
$$

From proposition 2, and multiplying and dividing by

$$
\sum_{\rho \in S_K} \boldsymbol{\pi}_\rho(\boldsymbol{\theta})
$$

we obtain

$$
I = \frac{1}{|S_K|} \sum_{\sigma \in S_K} \int_B \boldsymbol{\pi}_\sigma(\boldsymbol{\theta})\mathbf{p}_\sigma(\boldsymbol{\theta}, A)\boldsymbol{\mu}_{\mathrm{pr}}(\mathrm{d}\boldsymbol{\theta}) \quad \text{(by Prop. 2)}
$$

$$
= \frac{1}{|S_K|} \sum_{\sigma \in S_K} \int_B \frac{\boldsymbol{\pi}_\sigma(\boldsymbol{\theta})\mathbf{p}_\sigma(\boldsymbol{\theta}, A)}{\sum_{\sigma' \in S_K} \boldsymbol{\pi}_{\sigma'}(\boldsymbol{\theta})} \sum_{\rho \in S_K} \boldsymbol{\pi}_\rho(\boldsymbol{\theta})\boldsymbol{\mu}_{\mathrm{pr}}(\mathrm{d}\boldsymbol{\theta})
$$

$$
= \sum_{\sigma \in S_K} \int_B w_\sigma(\boldsymbol{\theta})\mathbf{p}_\sigma(\boldsymbol{\theta}, A)\boldsymbol{\pi}_{\mathrm{W}}(\boldsymbol{\theta})\boldsymbol{\mu}_{\mathrm{pr}}(\mathrm{d}\boldsymbol{\theta})
$$

$$
= \int_B \mathbf{p}^{(\mathrm{W})}(\boldsymbol{\theta}, A)\boldsymbol{\pi}_{\mathrm{W}}(\boldsymbol{\theta})\boldsymbol{\mu}_{\mathrm{pr}}(\mathrm{d}\boldsymbol{\theta}).
$$

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 $\boldsymbol{\mu}_{\mathrm{W}}$ is not of interest per se. However, we can use importance sampling to compute posterior expectations. Let $\mathcal{Q}(\boldsymbol{\theta}) := \mathcal{Q}(\theta_1)$ be a $\boldsymbol{\mu}$-integrable quantity of interest. We can write

$$
\mathbb{E}_{\mu_1}[\mathcal{Q}] = \mathbb{E}_{\boldsymbol{\mu}}[\mathcal{Q}(\theta_1)] = \mathbb{E}_{\boldsymbol{\mu}_{\mathrm{W}}} \left[\mathcal{Q}(\theta_1)\frac{\boldsymbol{\pi}(\boldsymbol{\theta})}{\boldsymbol{\pi}_{\mathrm{W}}(\boldsymbol{\theta})} \right]
$$

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

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

$$
\mu(\mathcal{Q}) \approx
$$

$$
\widehat{\mathcal{Q}}_{(\mathrm{W})} = \frac{1}{|S_K|}\frac{1}{N} \sum_{\sigma \in S_K} \sum_{n=1}^N \frac{\boldsymbol{\pi}(\boldsymbol{\theta}_\sigma^{(n)})}{\boldsymbol{\pi}_{\mathrm{W}}(\boldsymbol{\theta}_\sigma^{(n)})}\mathcal{Q}(\theta_{\sigma(1)}^{(n)})
$$

$$
= \frac{1}{|S_K|}\frac{1}{N} \sum_{\sigma \in S_K} \sum_{n=1}^N \widehat{w}(\boldsymbol{\theta}^{(n)}, \sigma)\mathcal{Q}(\theta_{\sigma(1)}^{(n)}), \tag{13}
$$

where we have denoted the importance sampling weights by $\widehat{w}(\boldsymbol{\theta}, \sigma) := \frac{\boldsymbol{\pi}(\boldsymbol{\theta}_\sigma)}{\boldsymbol{\pi}_{\mathrm{W}}(\boldsymbol{\theta}_\sigma)} = \frac{\mathrm{d}\boldsymbol{\mu}}{\mathrm{d}\boldsymbol{\mu}_{\mathrm{W}}}(\boldsymbol{\theta}_\sigma)$ and where N is the number of samples in the chain. Notice that $w(\boldsymbol{\theta}, \sigma) = \widehat{w}(\boldsymbol{\theta}, \sigma^{-1})$. 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 $\mathcal{Q}(\theta_k^{(n)})$, $k = 1, 2, \ldots, K$.

Remark 3 Although it is known that, in some cases, an importance sampling estimator can be negatively affected by the dimensionality of the parameter space Θ (see e.g., Asmussen and Glynn (2007) Remark 1.17 or Owen (AB) Examples 9.1-9.3), we argue that this is not the case for our estimator. Indeed, notice that the importance-sampling weights $\widehat{w}(\boldsymbol{\theta}, \sigma)$ are always upper bounded by $|S_K|$, and do not blow up when the dimension goes to infinity. In Sect. 5.7 we present a numerical example on a high-dimensional problem. The results on that section evidence the robustness of WGPT with respect to the dimension of θ.

The Weighted Generalized Parallel Tempering procedure is shown in Algorithm 3. To reiterate, we remark that sampling from $\mathbf{p}_\sigma(\boldsymbol{\theta}^{(n)}, \cdot)$ involves a swap of dynamics, i.e., kernels and temperatures.

Algorithm 3 Weighted Generalized Parallel Tempering.

---

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

Just as in Remark 2, one only needs to evaluate the posterior K times (instead of $|S_K|$) to compute $w_{(\cdot)}(\boldsymbol{\theta}^n)$.

4 Ergodicity of generalized parallel tempering

4.1 Preliminaries

We assume that the chains generated by the MCMC kernels p_k, for $k = 1, \ldots, K$, are aperiodic, μ_k-irreducible (Asmussen and Glynn 2007), 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

$$L_r = L_r(\Theta, \mu) = \{f : \Theta \mapsto \mathbb{R}, \ \mu\text{-measurable},$$
$$\text{s.t } \|f\|_r^r := \int |f(\theta)|^r \mu(\mathrm{d}\theta) < \infty \},$$
$$L_r^0 = L_r^0(\Theta, \mu) = \{f \in L_r(\Theta, \mu),$$
$$\text{s.t } \mu(f) := \int_\Theta f(\theta)\mu(\mathrm{d}\theta) = 0 \}.$$

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

$$\mathcal{M}_r(\Theta, \mu) := \{\nu \in \mathcal{M}(\Theta) \text{ s.t. } \nu \ll \mu, \ \|\nu\|_{L_r(\Theta,\mu)} < \infty\},$$
$$\text{where} \quad \|\nu\|_{L_r(\Theta,\mu)} := \left\| \frac{\mathrm{d}\nu}{\mathrm{d}\mu} \right\|_{L_r(\Theta,\mu)}.$$

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

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

for $f \in L_r(\Theta, \mu)$. It is well-known (see, e.g., Rudolf (2012)) 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

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

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

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

The next Proposition, whose proof can be found e.g., in Rudolf (2012), relates the existence of an L_2-spectral gap to the geometric ergodicity of the chain (with $\beta \leq \alpha$, in general).

Proposition 5 *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 Rudolf (2012) Proposition 3.17 and Appendix A.4. It is also shown in Rudolf (2012) that, in general, $\beta \leq \alpha$, with α, β given as in Eqs. (14) and (15). □

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 Theorems 1 and 2, 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 7 (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

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

An analogous definition holds for $\boldsymbol{\pi}_\sigma, \boldsymbol{\pi}_\rho$, with $\rho, \sigma \in S_K$.

Assumption 1 For $k = 1, \ldots, K$, let $\mu_k \in \mathcal{M}_1(\Theta, \mu_{\mathrm{pr}})$ be given as in (5), $p_k : \Theta \times \mathcal{B}(\Theta) \mapsto [0, 1]$ be the Markov kernel associated to the kth 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 $\boldsymbol{\mu}_\sigma, \boldsymbol{\mu}_\rho \in \mathcal{M}(\Theta^K)$ as in Eq. (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 $\sigma, \rho \in S_K$, $\Lambda_{\sigma,\rho} := \eta_{\boldsymbol{\mu}_{\mathrm{pr}}}(\boldsymbol{\pi}_\sigma, \boldsymbol{\pi}_\rho) > 0$, with $\boldsymbol{\pi}_\sigma, \boldsymbol{\pi}_\rho$ 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 Asmussen and Glynn (2007); Hairer et al. (2014). Assumption C3 holds true given the construction of the product measures in Sect. 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 6 *Suppose that Assumption 1 holds and let* $\mathbf{P} := \bigotimes_{k=1}^{K} P_k : L_2(\Theta^K, \boldsymbol{\mu}) \mapsto L_2(\Theta^K, \boldsymbol{\mu})$, *with invariant measure* $\boldsymbol{\mu} = \mu_1 \times \cdots \times \mu_K$. *Then,* $\mathbf{P}$ *has an* $L_2(\Theta^K, \boldsymbol{\mu})$-*spectral gap, i.e.,* $\|\mathbf{P}\|_{L_2^0(\Theta^K, \boldsymbol{\mu}) \mapsto L_2^0(\Theta^K, \boldsymbol{\mu})} < 1$. *Moreover, the Markov chain obtained from* $\mathbf{P}$ *is* L_r *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 $f \in L_2(\Theta^2, \boldsymbol{\mu})$. 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 $\{\phi_k\}_{i \in \mathbb{N}}$ 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

$$\|(P_1 \otimes I) f_0\|_{L_2(\Theta^2, \boldsymbol{\mu})}^2 = \left\| \sum_{(k,j) \neq (0,0)} (P_1 \phi_k) \psi_j c_{k,j} \right\|_{L_2(\Theta^2, \boldsymbol{\mu})}^2$$

$$= \left\| \sum_{j=0}^{\infty} \left(\sum_{k=1}^{\infty} P_1 \phi_k c_{k,j} \right) \psi_j + \sum_{j=1}^{\infty} c_{0,j} P_1 \phi_0 \psi_j \right\|_{L_2(\Theta^2, \boldsymbol{\mu})}^2 .$$
(16)

Splitting the sum, we get from the orthonormality of the basis that:

$$(16) = \sum_{j=1}^{\infty} \left\| \sum_{k=1}^{\infty} P_1 \phi_k c_{k,j} + c_{0,j} P_1 \phi_0 \right\|_{L_2(\Theta, \mu_1)}^2$$

$$+ \left\| \sum_{k=1}^{\infty} P_1 \phi_k c_{k,0} \right\|_{L_2(\Theta, \mu_1)}^2$$

$$= \sum_{j=1}^{\infty} \left\| P_1 \left(\sum_{k=1}^{\infty} \phi_k c_{k,j} \right) \right\|_{L_2(\Theta, \mu_1)}^2 + \sum_{j=1}^{\infty} \|c_{0,j} \phi_0\|_{L_2(\Theta, \mu_1)}^2$$

$$+ \left\| P_1 \left(\sum_{k=1}^{\infty} \phi_k c_{k,0} \right) \right\|_{L_2(\Theta, \mu_1)}^2$$

$$\leq \sum_{j=1}^{\infty} \left(\|P_1\|^2 \sum_{k=1}^{\infty} c_{k,j}^2 + c_{0,j}^2 \right) + \|P_1\|^2 \sum_{k=1}^{\infty} c_{k,0}^2$$

$$= \|P_1\|^2 \|f_0\|_{L_2(\Theta^2, \boldsymbol{\mu})}^2 + (1 - \|P_1\|^2) \sum_{j=1}^{\infty} (c_{0,j})^2 .$$

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 $\|\mathbf{P}\|_{L_2(\Theta^2, \boldsymbol{\mu}) \mapsto L_2(\Theta^2, \boldsymbol{\mu})}^2$:

$$\|\mathbf{P} f_0\|_{L_2(\Theta^2, \boldsymbol{\mu})}^2 = \|(P_1 \otimes P_2) f_0\|_{L_2(\Theta^2, \boldsymbol{\mu})}^2$$

$$= \|(P_1 \otimes I)(I \otimes P_2) f_0\|_{L_2(\Theta^2, \boldsymbol{\mu})}^2$$

$$\leq \|P_1\|^2 \|(I \otimes P_2) f_0\|_{L_2(\Theta^2, \boldsymbol{\mu})}^2$$

$$+ (1 - \|P_1\|^2)$$

$$\times \left(\sum_{j=1}^{\infty} \left((I \otimes P_2) \sum_{(\ell,k) \neq (0,0)} c_{\ell,k} \phi_\ell \psi_k, \phi_0 \psi_j \right)^2 \right)$$

$$= \|P_1\|^2 \|(I \otimes P_2) f_0\|_{L_2(\Theta^2, \boldsymbol{\mu})}^2$$

$$+ (1 - \|P_1\|^2) \left(\sum_{j=1}^{\infty} \left(\sum_{k=1}^{\infty} c_{0,k}(P_2 \psi_k), \psi_j \right)^2 \right)$$

$$\leq \|P_1\|^2 \|(I \otimes P_2) f_0\|_{L_2(\Theta^2, \boldsymbol{\mu})}^2$$

$$+ (1 - \|P_1\|^2) \left\| P_2 \left(\sum_{k=1}^{\infty} c_{0,k} \psi_k \right) \right\|_{L_2(\Theta, \mu_2)}^2$$

$$\leq \|P_1\|^2 \|P_2\|^2 \|f_0\|_{L_2(\Theta^2, \boldsymbol{\mu})}^2$$

$$+ \|P_1\|^2 (1 - \|P_2\|^2) \left(\sum_{j=1}^{\infty} c_{j,0}^2 \right)$$

$$+ (1 - \|P_1\|^2) \|P_2\|^2 \left(\sum_{k=1}^{\infty} c_{0,k}^2 \right)$$

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

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

Thus, we have that

$$
\|\mathbf{P}\|_{L_2^0(\Theta^2,\boldsymbol{\mu}) \mapsto L_2^0(\Theta^2,\boldsymbol{\mu})} \leq \max_{k=1,2} \{ \|P_k\|_{L_2^0(\Theta,\mu_k) \mapsto L_2^0(\Theta,\mu_k)} \} < 1.
$$

The previous result can easily be extended to $K > 2$. Lastly, $L_r(\Theta^K, \boldsymbol{\mu})$-geometric ergodicity $\forall r \in [1, \infty]$ follows from proposition 5. □

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

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

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

$$
\begin{aligned}
&\left\| \mathbf{P}^{(\mathrm{UW})} \right\|_{L_2^0(\Theta^K,\boldsymbol{\mu}) \mapsto L_2^0(\Theta^K,\boldsymbol{\mu})} \\
&\leq \left\| \mathbf{Q}^{(\mathrm{UW})} \right\|_{L_2^0(\Theta^K,\boldsymbol{\mu}) \mapsto L_2^0(\Theta^K,\boldsymbol{\mu})}^2 \|\mathbf{P}\|_{L_2^0(\Theta^K,\boldsymbol{\mu}) \mapsto L_2^0(\Theta^K,\boldsymbol{\mu})} \\
&\leq \|\mathbf{P}\|_{L_2^0(\Theta^K,\boldsymbol{\mu}) \mapsto L_2^0(\Theta^K,\boldsymbol{\mu})} < 1,
\end{aligned}
$$

where the previous line follows from Proposition 6 and the fact that $\mathbf{Q}^{(\mathrm{UW})}$ is a weak contraction in $L_2(\Theta^K, \boldsymbol{\mu})$ (see, e.g., Baxter and Rosenthal (1995) Proposition 1). Lastly, $L_r(\Theta^K, \boldsymbol{\mu})$-geometric ergodicity $\forall r \in [1, \infty]$ follows from Proposition 5 and the fact that $\mathbf{P}^{(\mathrm{UW})}$ is $\boldsymbol{\mu}$-reversible by Proposition 3. □

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

Proposition 7 *Let $f \in L_2^0(\Theta^K, \boldsymbol{\mu}_{\mathrm{W}})$ be a $\boldsymbol{\mu}_{\mathrm{W}}$-integrable function such that $\|f\|_{L_2(\Theta^K,\boldsymbol{\mu}_{\mathrm{W}})} = 1$, and denote by $\mathbb{V}_{\boldsymbol{\mu}_{\mathrm{W}}}[f]$, $\mathbb{V}_{\boldsymbol{\mu}_\sigma}[f]$ the variance of f with respect to $\boldsymbol{\mu}_{\mathrm{W}}, \boldsymbol{\mu}_\sigma$,*

respectively with $\sigma \in S_K$. In addition, suppose Assumption 1 holds. Then, it can be shown that

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

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 presented in Appendix A. □

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

Theorem 2 *(Ergodicity of WGPT) Suppose Assumption 1 holds for some $r \in [1, \infty]$ and denote by $\boldsymbol{\mu}_{\mathrm{W}}$ the invariant measure of the WGPT Markov operator $\mathbf{P}^{(\mathrm{W})}$. Then, $\mathbf{P}^{(\mathrm{W})}$ has an $L_2(\Theta^K, \boldsymbol{\mu}_{\mathrm{W}})$-spectral gap. Moreover, the chain generated by $\mathbf{P}^{(\mathrm{W})}$ is $L_r(\Theta^K, \boldsymbol{\mu}_{\mathrm{W}})$ geometrically ergodic for any $r \in [1, \infty]$.*

Proof Let $\mathcal{L} := \{ f \in L_2^0(\Theta^K, \boldsymbol{\mu}_{\mathrm{W}}) : \|f\|_{L_2^0(\Theta^K,\boldsymbol{\mu}_{\mathrm{W}})} = 1 \}$, and, for notational clarity, write

$$
\|\mathbf{P}_\sigma\|_{L_2^0} := \left\| \mathbf{P}_\sigma^2 \right\|_{L_2^0(\Theta^K,\boldsymbol{\mu}_\sigma) \mapsto L_2^0(\Theta^K,\boldsymbol{\mu}_\sigma)}.
$$

Then, from the definition of operator norm,

$$
\begin{aligned}
&\left\| \mathbf{P}^{(\mathrm{w})} \right\|_{L_2^0(\Theta^K,\boldsymbol{\mu}_{\mathrm{W}}) \mapsto L_2^0(\Theta^K,\boldsymbol{\mu}_{\mathrm{W}})}^2 \\
&= \sup_{f \in \mathcal{L}} \left\| \mathbf{P}^{(\mathrm{w})} f \right\|_{L_2(\Theta^K,\boldsymbol{\mu}_{\mathrm{W}})}^2 \\
&= \sup_{f \in \mathcal{L}} \int_{\Theta^K} \left| \sum_{\sigma \in S_K} w_\sigma(\boldsymbol{\theta}) \int_{\Theta^K} f(\boldsymbol{y}) \mathbf{p}_\sigma(\boldsymbol{\theta}, \mathrm{d}\boldsymbol{y}) \right|^2 \boldsymbol{\mu}_{\mathrm{W}}(\mathrm{d}\boldsymbol{\theta}) \\
&\leq \sup_{f \in \mathcal{L}} \int_{\Theta^K} \sum_{\sigma \in S_K} w_\sigma(\boldsymbol{\theta}) \left| \int_{\Theta^K} f(\boldsymbol{y}) \mathbf{p}_\sigma(\boldsymbol{\theta}, \mathrm{d}\boldsymbol{y}) \right|^2 \boldsymbol{\mu}_{\mathrm{W}}(\mathrm{d}\boldsymbol{\theta}) \\
&= \sup_{f \in \mathcal{L}} \frac{1}{|S_K|} \sum_{\sigma \in S_K} \int_{\Theta^K} \left| \int_{\Theta^K} f(\boldsymbol{y}) \mathbf{p}_\sigma(\boldsymbol{\theta}, \mathrm{d}\boldsymbol{y}) \right|^2 \boldsymbol{\mu}_\sigma(\mathrm{d}\boldsymbol{\theta}),
\end{aligned}
\tag{17}
$$

where the second to last line follows from the convexity of $(\cdot)^2$ and the last line follows from the definition of w_σ and $\boldsymbol{\mu}_{\mathrm{W}}$. Now, let $\bar{f}_\sigma := \boldsymbol{\mu}_\sigma(f)$. Notice that we have

$$
\begin{aligned}
&\int_{\Theta^K} \left| \int_{\Theta^K} f(\boldsymbol{y}) \mathbf{p}_\sigma(\boldsymbol{\theta}, \mathrm{d}\boldsymbol{y}) \right|^2 \boldsymbol{\mu}_\sigma(\mathrm{d}\boldsymbol{\theta}) \\
&= \int_{\Theta^K} \left| \int_{\Theta^K} (f(\boldsymbol{y}) - \bar{f}_\sigma + \bar{f}_\sigma) \mathbf{p}_\sigma(\boldsymbol{\theta}, \mathrm{d}\boldsymbol{y}) \right|^2 \boldsymbol{\mu}_\sigma(\mathrm{d}\boldsymbol{\theta}) \\
&= \int_{\Theta^K} \left(\left| \int_{\Theta^K} (f(\boldsymbol{y}) - \bar{f}_\sigma) \mathbf{p}_\sigma(\boldsymbol{\theta}, \mathrm{d}\boldsymbol{y}) \right|^2 + \left| \int_{\Theta^K} \bar{f}_\sigma \mathbf{p}_\sigma(\boldsymbol{\theta}, \mathrm{d}\boldsymbol{y}) \right|^2 \right.
\end{aligned}
$$

$$+ 2\bar{f}_\sigma \int_{\Theta^K} (f(\mathbf{y}) - \bar{f}_\sigma) \mathbf{p}_\sigma(\boldsymbol{\theta}, d\mathbf{y}) \bigg) \boldsymbol{\mu}_\sigma(d\boldsymbol{\theta})$$

$$= \underbrace{\int_{\Theta^K} \left(\int_{\Theta^K} (f(\mathbf{y}) - \bar{f}_\sigma) \mathbf{p}_\sigma(\boldsymbol{\theta}, d\mathbf{y}) \right)^2 \boldsymbol{\mu}_\sigma(d\boldsymbol{\theta})}_{\text{I}} + (\bar{f}_\sigma)^2$$

$$+ 2\bar{f}_\sigma \underbrace{\int_{\Theta^K} \int_{\Theta^K} (f(\mathbf{y}) - \bar{f}_\sigma) \mathbf{p}_{\boldsymbol{\alpha}}(\boldsymbol{\theta}, d\mathbf{y}) \boldsymbol{\mu}_{\boldsymbol{\alpha}}(d\boldsymbol{\theta})}_{=} \, 0 \text{ by stationarity}$$

$$\tag{18}$$

Thus, multiplying and dividing I by

$$\left(\int_{\Theta^K} \left(f(\boldsymbol{\theta}) - \bar{f}_\sigma \right)^2 \boldsymbol{\mu}_\sigma(d\boldsymbol{\theta}) \right),$$

we obtain from the definition of $\|\mathbf{P}_\sigma\|^2_{L_2^0}$ that:

$$(18) = \left(\frac{\int_{\Theta^K} \left(\int_{\Theta^K} (f(\mathbf{y}) - \bar{f}_\sigma) \mathbf{p}_\sigma(\boldsymbol{\theta}, d\mathbf{y}) \right)^2 \boldsymbol{\mu}_\sigma(d\boldsymbol{\theta})}{\int_{\Theta^K} \left(f(\boldsymbol{\theta}) - \bar{f}_\sigma \right)^2 \boldsymbol{\mu}_\sigma(d\boldsymbol{\theta})} \right)$$

$$\times \left(\int_{\Theta^K} \left(f(\boldsymbol{\theta}) - \bar{f}_\sigma \right)^2 \boldsymbol{\mu}_\sigma(d\boldsymbol{\theta}) \right) + (\bar{f}_\sigma)^2$$

$$\leq \|\mathbf{P}_\sigma\|^2_{L_2^0} \left(\int_{\Theta^K} \left(f(\boldsymbol{\theta}) - \bar{f}_\sigma \right)^2 \boldsymbol{\mu}_\sigma(d\boldsymbol{\theta}) \right) + (\bar{f}_\sigma)^2$$

$$= \|\mathbf{P}_\sigma\|^2_{L_2^0} \left(\int_{\Theta^K} f(\boldsymbol{\theta})^2 \boldsymbol{\mu}_\sigma(d\boldsymbol{\theta}) \right) \tag{19}$$

$$+ \left(1 - \|\mathbf{P}_\sigma\|^2_{L_2^0} \right) (\bar{f}_\sigma)^2$$

$$= \left(\int_{\Theta^K} f(\boldsymbol{\theta})^2 \boldsymbol{\mu}_\sigma(d\boldsymbol{\theta}) \right) - \underbrace{\left(1 - \|\mathbf{P}_\sigma\|^2_{L_2^0} \right)}_{:=\gamma}, \text{ with } \gamma \in (0, 1)$$

$$\times \left(\int_{\Theta^K} \left(f(\boldsymbol{\theta}) - \bar{f}_\sigma \right)^2 \boldsymbol{\mu}_\sigma(d\boldsymbol{\theta}) \right).$$

Replacing Eq. (19) into Eq. (17), we get

$$\left\| \mathbf{P}^{(\mathrm{W})} \right\|^2_{L_2^0(\Theta^K, \boldsymbol{\mu}_{\mathrm{W}}) \mapsto L_2^0(\Theta^K, \boldsymbol{\mu}_{\mathrm{W}})}$$

$$\leq \sup_{f \in \mathcal{L}} \left(\int_{\Theta^K} f(\boldsymbol{\theta})^2 \boldsymbol{\mu}_{\mathrm{W}}(d\boldsymbol{\theta}) \right) - \frac{\gamma}{|S_K|} \sum_{\sigma \in S_K} \mathbb{V}_{\boldsymbol{\mu}_\sigma}[f]$$

$$\leq 1 - \gamma \left(\frac{\Lambda_m}{2 - \Lambda_m} \right) < 1 \quad \text{(by Proposition 7)}.$$

Thus, $\boldsymbol{P}^{(\mathrm{w})}$ has an $L_2(\Theta^K, \boldsymbol{\mu}_{\mathrm{W}})$ spectral gap. Once again, $L_r(\Theta^K, \boldsymbol{\mu}_{\mathrm{W}})$-geometric ergodicity (with $r \in [1, \infty]$) follows from Proposition 5 and the fact that $\mathbf{P}^{(\mathrm{W})}$ is $\boldsymbol{\mu}_{\mathrm{W}}$-reversible by Proposition 4. □

4.2.1 Discussion and comparison to similar theoretical result

Theorems 1 and 2 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 that 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 (Miasojedow et al. 2013) and (Woodard et al. 2009). In particular, the work (Miasojedow et al. 2013) has proved geometric ergodicity for the pairwise parallel tempering algorithm using the standard drift condition construction of Meyn and Tweedie (2012). 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 Woodard et al. (2009) 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 that 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 Woodard et al. (2009) for the WGPT algorithm seems more problematic.

5 Numerical experiments

We now present four academic 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 different algorithms in their simplest version that uses random walk Metropolis as a base transition kernel. The only exception is in Sect. 5.7, which presents a high-dimensional BIP for which the preconditioned Crank-Nicolson Cotter et al. (2013) is used as the base method in all algorithms instead of RWM. More advanced samplers, such as Adaptive metropolis (Haario et al. 2006, 2001), or other transition kernels, could be used as well to replace RWM or pCN. Experiments 5.3, 5.4 and 5.5 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 Sects. 5.3 and 5.5 were run on a single thread, while the numerical simulations in Sect. 5.4 were run on an *embarrassingly parallel* fashion over 8 threads using the Message Passing Interface (MPI) and the Python package MPI4py (Dalcín et al. 2005). Lastly, Experiments 5.6 and 5.7 were 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 https://doi.org/10.5281/zenodo.3700048.

5.1 Implementation remarks

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., $S_K = \mathscr{S}_K$), 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 $\pi(\theta_k^{(n)})$, $k = 1, \ldots, K$, $n = 1, \ldots N$, 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.

There is, however, a clear trade-off between the choice of K (which has a direct impact on the efficiency of the method), and the computational cost associated to (G)PT. Intuitively, a large K would provide a better mixing, however, it requires a larger number of forward model evaluations, which tends to be costly. We remark that such a trade-off between efficiency and number of function evaluations is also present in some advanced MCMC methods, such as Hamiltonian Monte Carlo, where one needs to choose a number of time steps for the time integration (see, e.g., Beskos et al. (2017)). Furthermore, there is an additional constraint when choosing $S_K = \mathscr{S}_K$, and it is the *permutation cost* associated to computing $r^{(\mathrm{UW})}(\boldsymbol{\theta}, \sigma)$ and $w_\sigma(\boldsymbol{\theta})$. In particular, the computation of either of those quantities has a complexity of $K!$ thus, this cost will eventually surpass the cost of evaluating the forward model K times. This is illustrated in Fig. 1, where we plot the cost per sample of two different posteriors vs K. These posteriors are taken from the numerical examples in Sects. 5.5 and 5.7. The posterior in Sect. 5.5 is rather inexpensive to evaluate, since one can compute the forward map $\mathcal{F}$ analytically (the difficulty associated to sampling from that posterior comes from its high multi-modality). On the contrary, evaluating the posterior in Sect. 5.7 requires numerically approximating the solution to a time-dependent, second-order PDE, and as such, evaluating such a posterior is costly. As we can see for $K \leq 6$, the computational cost in both cases is dominated by the forward model evaluation. Notice that for $K \leq 9$, the cost per sample from posterior (27) is dominated by the evaluation of the forward model.

Thus, for high values of K, it is advisable to only consider the union of properly chosen semi-groups A, B of $\mathscr{S}_K$,
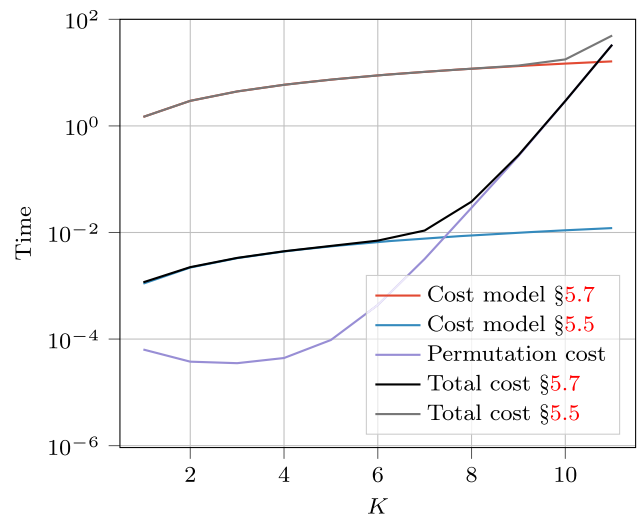

Fig. 1 Cost per sample vs K for $S_K = \mathscr{S}_K$ for the forward model in Sect. 5.5 and the forward model in 5.7

with $A \cap B \neq \emptyset$, such that A, B generates $\mathscr{S}_K$ (i.e., if the smallest semi-groups that contains A and B is $\mathscr{S}_K$ itself), and $|A \cup B| < |\mathscr{S}_K| = K!$, which is referred to as partial Infinite Swapping in the continuous case (Dupuis et al. 2012). 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 Dupuis et al. (2012) 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 $\mu_{i+1}, \mu_{i+2}, \ldots, \mu_{i+\ell}$ for some user-defined $\ell \geq 1$ and any given $i = 1, 2, \ldots, K - 1$. 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.

We reiterate that the total number of temperatures K depends heavily on the problem and the computational budget available (Doll et al. 2012; Van Der Sluys et al. 2008; Yu et al. 2016) For the experiments considered in the work we chose $K = 4$ or $K = 5$, which provide an acceptable compromise between acceleration and cost.

5.2 Experimental setup

We now present an experimental setup common to all the numerical examples presented in the following subsections. In particular, all the experiments presented in this work utilize a *base* method given by either RWM (for Experiments 5.3 through 5.6) or pCN (used in Experiment 5.7) for the Markov transition kernels p. Furthermore, we take $S_K = \mathscr{S}_K$ for all experiments, where $K = 5$ for Experiment 5.5 and $K = 4$ for the other 4 experiments. In addition, we follow the *rule of thumb* of Earl and Deem (2005) for the choice of temperatures, setting, for each experiment, $T_k = a^{k-1}$, $k = 1, \ldots, K$, for some positive constant $a > 1$. The particular choice of a is problem-dependent and it is generally chosen so that μ_K becomes sufficiently simple to explore. For each experiment we implement 5 MCMC algorithms to sample from a given posterior $\mu = \mu_1$, namely, the base (untempered) method (either RWM or pCN), and such a method combined with the standard PT algorithm (PT) with $N_s = 1$, the PSDPT algorithm of Łącki and Miasojedow (2016), and both versions of GPT. For our setting, the tempered algorithms have a cost (in terms of number of likelihood evaluations) that is K times larger than the base method. Thus, to obtain a fair comparison across all algorithms, we run the chain for the base method K times longer. Lastly, given some problem-dependent quantity of interest $\mathcal{Q}$, we assess the efficiency of our proposed algorithms to compute the posterior expectation of $\mathcal{Q}$ by comparing the mean square error (Experiments 5.3–5.5), for which the exact value of $\mathbb{E}_\mu[\mathcal{Q}]$ is known, or the variance (Experiments 5.6–5.7) of the ergodic estimator $\hat{\mathcal{Q}}$ obtained over N_{runs} independent runs of each algorithm.

5.3 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 $\mu_{\text{pr}} = \mathcal{U}([0, 1]^2)$ given by

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

where $\theta = (\theta_1, \theta_2)$, 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 Fig. 2. 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.
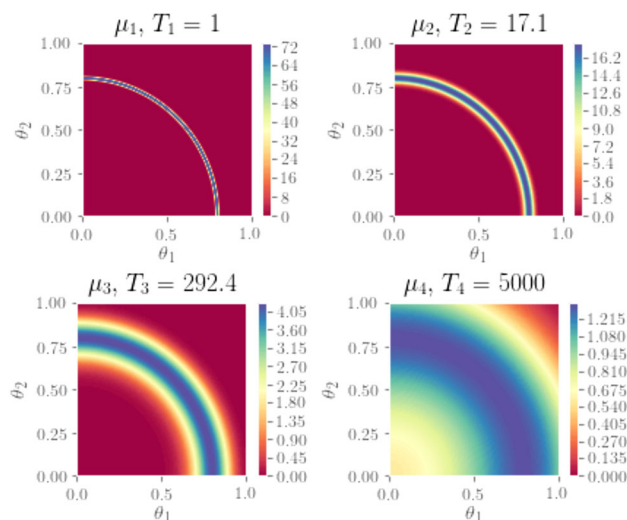

Fig. 2 Tempered densities (with $T_1 = 1$, $T_2 = 17.1$, $T_3 = 292.4$, $T_4 = 5000$) for the density concentrated around a quarter circle-shaped manifold example. As we can see, the density becomes less concentrated as the temperature increases, which allows us to use RWM proposals with larger step sizes

We aim at estimating $\hat{\mathcal{Q}}_i = \mathbb{E}_\mu[\theta_i] \approx \hat{\theta}_i$, for $i = 1, 2$. For the tempered algorithms (PT, PSDPT, UGPT, and WGPT), we consider $K = 4$ temperatures and choose $T_4 = 5000$, so that the tempered density π_4 becomes sufficiently simple to explore the target distribution. This gives $T_1 = 1$, $T_2 = 17.1$, $T_3 = 292.4$, $T_4 = 5000$.

We compare the quality of our algorithms by examining the variance of the estimators $\hat{\theta}_i$, $i = 1, 2$ computed over $N_{\text{runs}} = 100$ independent MCMC runs of each algorithm. For the tempered algorithms, each estimator is obtained by running the inversion experiment for $N = 25,000$ samples per run, discarding the first 20% of the samples (5000) as a burn-in. Accordingly, we run the single-chain random walk Metropolis algorithm for $N_{\text{RWM}} = KN = 100,000$ iterations, per run, and discard the first 20% of the samples obtained with the RWM algorithm (20,000) as a burn-in.

The untempered RWM algorithm uses Gaussian proposals with covariance matrix $\Sigma_{\text{RWM}} = \rho_1^2 I_{2\times2}$, where $I_{2\times2}$ is the identity matrix in $\mathbb{R}^{2\times2}$, and $\rho_1^2 = 0.022$ is chosen in order to obtain an acceptance rate of around 0.23. 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 $q_{\text{prop},k}(\theta_k^{(n)}, \cdot) = \mathcal{N}(\theta_k^{(n)}, \rho_k^2 I_{2\times2})$, where ρ_k is shown in Table 1. This choice of ρ_k gives an acceptance rate for each chain of around 0.23. Notice that ρ_1 corresponds to the "step-size" of the single-temperature RWM algorithm.

Experimental results for the ergodic run are shown in Table 2. We can see how both GPT algorithms provide a gain over 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 Fig. 3. Here, the

Table 1 Step size of the RWM proposal distribution for the manifold experiment

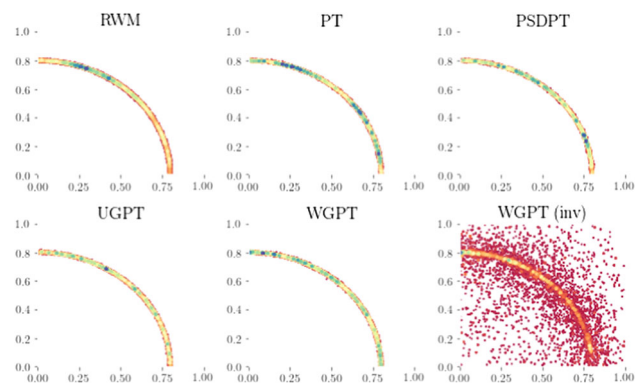|	$k = 1$	$k = 2$	$k = 3$	$k = 4$
ρ_k	0.022	0.090	0.310	0.650

Fig. 3 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

subplot titled "WGPT" (bottom row, middle) corresponds to weighted samples from $\boldsymbol{\mu}_W$, with weight $\hat{w}$ as in (13), while the one titled "WGPT (inv)" (bottom row, right) corresponds to samples from $\boldsymbol{\mu}_W$ without any post-processing. Notice how the samples from the latter concentrates over a *thicker* manifold, which in turn makes the target density easier to explore when using state-dependent Markov transition kernels.

5.4 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. 20), based on some noisy measured data. Let $(\Theta, \mathcal{B}(\Theta), \mu_{pr})$ be the measure space, set $\Theta = \bar{D} := [0, 1]^2$, with Lebesgue (uniform) measure μ_{pr}, and consider the following Poisson's equation with homogeneous boundary conditions:

$$\begin{cases} \Delta u(x, \theta) = f(x, \theta), & x \in D, \ \theta \in \Theta, \\ u(x, \theta) = 0, & x \in \partial D. \end{cases} \tag{20}$$

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 (20) with a forcing term given by

$$f(x) = \sum_{i=1}^{4} e^{-1000[(x_1 - s_1^{(i)})^2 + (x_2 - s_2^{(i)})^2]}, \tag{21}$$

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 with distribution $\mathcal{N}(0, \eta^2 I_{64 \times 64})$, with $\eta = 3.2 \times 10^{-6}$, (which corresponds to a 1% noise) and where $I_{64 \times 64}$ is the 64-dimensional identity matrix. Thus, we set $(Y, \|\cdot\|_Y) = (\mathbb{R}^{64 \times 64}, \|\cdot\|)$, with $\|A\| = (64\eta)^{-2} \|A\|_F^2$, 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. (21). That, is, we construct our forward operator $\mathcal{F} : \Theta \mapsto Y$ by solving (20) with a source term given by

$$f(x, \theta) = e^{-1000[(x_1 - \theta_1)^2 + (x_2 - \theta_2)^2]}. \tag{22}$$

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 $\mu_{pr} = \mathcal{U}(\bar{D})$ 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 (20) with a forcing term of the form of (22) 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 Fig. 4).

We follow a similar experimental setup to the previous example, and aim at estimating $\hat{Q}_i = \mathbb{E}_\mu[\theta_i] \approx \hat{\theta}_i$, for $i = 1, 2$. We use $K = 4$ temperatures and $N_{runs} = 100$. 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 $T_4 = 400$. 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

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

| | Mean | | MSE | | MSE$_{\text{RWM}}$/MSE | |
	$\hat{\theta}_1$	$\hat{\theta}_2$	$\hat{\theta}_1$	$\hat{\theta}_2$	$\hat{\theta}_1$	$\hat{\theta}_2$
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
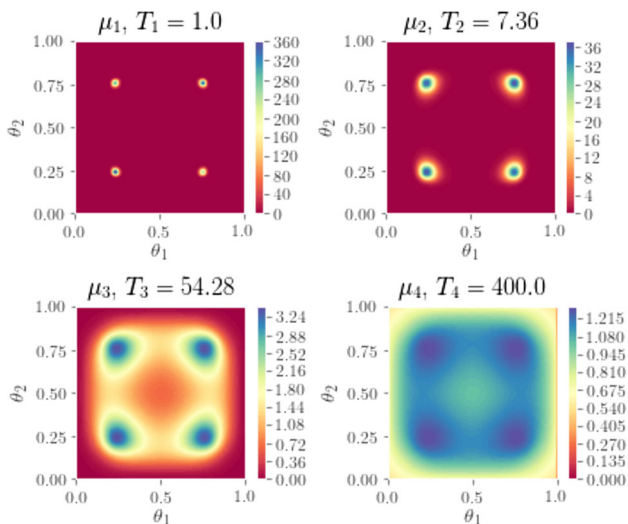
Fig. 4 True tempered densities for the elliptic BIP example. Notice that the density is not symmetric, due to the additional random noise

Fig. 5 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

5.5 1D wave source inversion

We consider a small variation of example 5.1 in Motamed and Appelo (2019). Let $(\Theta, \mathcal{B}(\Theta), \mu_{\text{pr}})$ be a measure space, with $\Theta = [-5, 5]$ and uniform (Lebesgue) measure μ_{pr}, and let $I = (0, T)$ be a time interval. Consider the following Cauchy problem for the 1D wave equation:

$$\begin{cases} u_{tt}(x, t, \theta) - u_{xx}(x, t, \theta) = 0, & (x, t, \theta) \in \mathbb{R} \times I \times \Theta, \\ u(x, 0, \theta) = h(x, \theta), & (x, t, \theta) \in \mathbb{R} \times \{0\} \times \Theta, \\ u_t(x, 0, \theta) = 0, & (x, t, \theta) \in \mathbb{R} \times \{0\} \times \Theta. \end{cases}$$
(23)

Here, $h(x, \theta)$ acts as a source term generating a initial wave pulse. Notice that Eq. (23) can be easily solved using d'Alembert's formula, namely

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

Synthetic data y is generated by solving Eq. (23) with initial data

Table 3 Step size of the RWM proposal distribution for the elliptic BIP experiment

	$k = 1$	$k = 2$	$k = 3$	$k = 4$
$\rho_{k,\text{Tempered}}$	0.030	0.100	0.400	0.600
$\rho_{k,\text{RWM}}$	0.160	–	–	–

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 ($\rho_{\text{RWM}} = 0.16$) 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 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 Fig. 4.
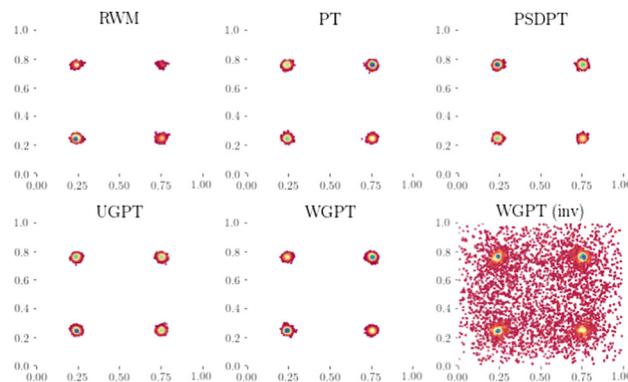
Table 4 Results for the elliptic BIP problem. The computational cost is comparable across all algorithms, given that the cost of each iteration is dominated by the cost of solving the underlying PDE

	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

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

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 instants between $t = 0$ and $T = 5$. The signal recorded by each receiver is assumed to be polluted by additive Gaussian noise $\mathcal{N}(0, \eta^2 I_{1000 \times 1000})$, with $\eta = 0.01$, which corresponds to roughly 1% noise. We set $(Y, \|\|\|_Y) = (\mathbb{R}^{11 \times 1000}, \|\cdot\|_\Sigma)$, with

$$\|A\|_\Sigma^2 = (\sqrt{N_R} \eta)^{-2} \sum_{i=1}^{N_R} \sum_{j=1}^{N_T} A_{i,j}^2,$$

$A \in \mathbb{R}^{11 \times 1000}$. Once again, we assume a misspecified model where we construct our forward operator $\mathcal{F} : \Theta \mapsto Y$ by solving (23) with a source term given by

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

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 $\mu_{pr} = \mathcal{U}([-5, 5])$ given by

$$\pi(\theta) = \frac{1}{Z} \exp\left(-\frac{1}{2} \|y - \mathcal{F}(\theta)\|_\Sigma^2 \right)$$
$$= \frac{1}{Z} \exp\left(-\Phi(\theta; y) \right), \qquad (24)$$

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 Fig. 6. This, shape of $\Phi(\theta; y)$ makes the posterior difficult to explore using local proposals.

In this case, we consider $K = 5$, and set $T_1 = 1$, $T_2 = 5$, $T_3 = 25$, $T_4 = 125$ and $T_5 = 625$. Notice that from Fig. 1, the computational cost per sample is dominated by the evaluation of (24) for values of $K \leq 6$. Once again, we
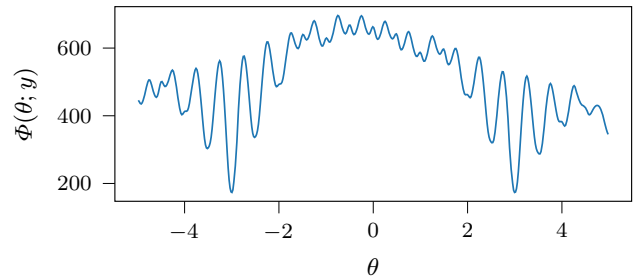
Fig. 6 Multi-modal potential for the Cauchy problem. Notice the minima around $\theta = -3$ and $\theta = 3$

Table 5 Step size of the RWM proposal distribution for the Cauchy BIP experiment

	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$
$\rho_{k, \text{Tempered}}$	0.02	0.05	0.10	0.50	2.0
$\rho_{k, \text{RWM}}$	0.5	–	–	–	–

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

We consider $\mathcal{Q} = \theta$ as a quantity of interest. Experimental results are shown in Table 6. Once again, we can see how both GPT algorithms provide a gain over RWM and both variations of the PT algorithm, with the WGPT algorithm providing the largest gain. Notice that, given the high muti-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 $\mathbb{E}_\mu[\theta] = 0.08211$.

Table 6 Results for the 1D Cauchy BIP problem. The computational cost is comparable across all algorithms

	Mean	MSE	MSE$_{RWM}$/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

5.6 Acoustic wave source inversion

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. 25), based on some noisy measured data. We begin by describing the mathematical model of such wave phenomena. Let $(\Theta, \mathcal{B}(\Theta), \mu_{pr})$ be the measure space , with Lebesgue (uniform) measure μ_{pr}, set $\bar{D} := [0, 3] \times [0, 2]$, $\partial D = \bar{\Gamma}_N \cup \bar{\Gamma}_{Abs}, \mathring{\Gamma}_N \cap \mathring{\Gamma}_{Abs} = 0, |\Gamma_N|, |\Gamma_{Abs}| > 0$, 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 $\theta := (s_1, s_2, \alpha, \beta) \in \Theta$. Consider the following acoustic wave equation with absorbing boundary conditions:

$$\begin{cases} \alpha^2 u_{tt} - \nabla \cdot (\beta^2 \nabla u) = f, & \text{in } D \times (0, T) \times \Theta, \\ u = u_t = 0, & \text{in } D \times \{0\} \times \Theta, \\ \beta^2 \nabla u \cdot \hat{n} = 0, & \text{on } \Gamma_N \times (0, T) \times \Theta, \\ \beta^2 \nabla u \cdot \hat{n} = -\alpha \beta u_t, & \text{on } \Gamma_{Abs} \times (0, T) \times \Theta, \end{cases} \quad (25)$$

where $u = u(x, t, \theta)$, and $f = f(x, t, \theta)$. 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 Eq. (25) with a force term given by

$$f(x, t, \theta) = 10^{11} e^{-\frac{1}{2 \cdot 0.1^2} \left[(x_1 - s_1)^2 + (x_2 - s_2)^2 \right]}$$
$$\times (1 - 2 \cdot 1000 \pi^2 t^2) e^{-2 \cdot 1000^2 \pi^2 t^2}, \quad (26)$$

with a true set of parameters $\Theta \ni \theta^* := (s_1, s_2, \alpha, \beta)$ given by $s_1 = 1.5, s_2 = 1.0, \alpha = 10, \beta = 5000$, 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 s_1, s_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 $3.14159\ldots$ in Eq. (26) and it should not be confused with the symbol for density. The data measured by each receiver is polluted by additive Gaussian noise $\mathcal{N}(0, \eta^2 I_{117 \times 117})$, with $\eta = 0.013$, which corresponds to roughly a 2% noise. Thus, we have that $(Y, \|\cdot\|_Y) = (\mathbb{R}^{3 \times 117}, \|\cdot\|_\Sigma)$, where $\|A\|_\Sigma^2 := (\sqrt{N_R} \eta)^{-2} \sum_{i=1}^{N_R} \sum_{j=0}^{N_T} A_{i,j}^2$. Thus, the forward mapping operator $\mathcal{F} : \Theta \mapsto Y$ can be understood as the numerical solution of Eq. (25) 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-Friedrichs-Lewy condition (Quarteroni and Quarteroni 2009). This numerical solution is implemented using the Python library FEniCS (Logg et al. 2012), 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 $\mu_{pr} = \mathcal{U}(\Theta)$ 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, Eq. (25) 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 Fig. 7, where we plot of the log-likelihood for different source locations (s_1, s_2) and for fixed values of the material properties $\alpha = 10, \beta = 5000$. More precisely, we plot $\tilde{\Phi}((s_1, s_2); y) := -\frac{1}{2} \|y - \mathcal{F}(s_1, s_2, 10, 5000)\|_\Sigma^2$ on a grid of 100×100 equally spaced points (s_1, s_2) in D. It can be seen that, even though the log-likelihood has a clear peak around the true value of (s_1, s_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 $\hat{Q}_i = \mathbb{E}_\mu[\theta_i] \approx \hat{\theta}_i$, for $i = 1, 2$. Once again, we consider $K = 4$ temperatures for the tempered algorithms (PT, PSDPT, UGPT, and WGPT), and set temperatures to $T_1 = 1, T_2 = 7.36, T_3 = 54.28, T_4 = 400$. We compare the quality of our algorithms by examining the variance of the estimators $\hat{\theta}_i, i = 1, 2$ computed over $N_{runs} = 50$ independent MCMC runs of each algorithm. For each run, we run the tempered algorithms obtaining $N = 7,000$ samples, discarding the first 20% of the samples (1400) as a burn-in. For the RWM algorithm, we run the inversion experiment
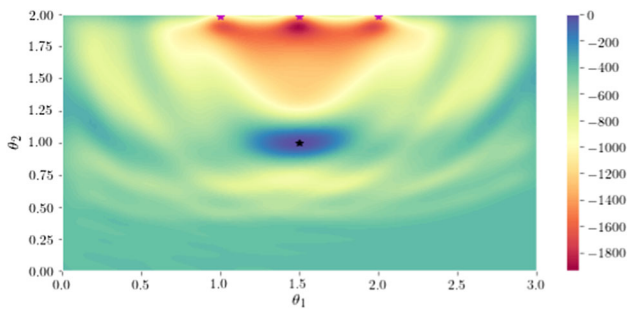
Fig. 7 Plot of the log-likelihood for different values of s_1, s_2 and fixed values of $\alpha = 10$ and $\beta = 5000$. The magenta points represent the reciever locations R_1, R_2, R_3. The black point represents the true location of the source $(s_1, s_2) = (1.5, 1.0)$

Table 7 Step size of the RWM proposal distribution for the acoustic BIP experiment. Here $\mathrm{Diag}(d_1, d_2, \ldots, d_N)$ is to be understood as the $N \times N$ diagonal matrix with entries $d_1, d_2, \ldots, d_N$

	$\mathcal{C}_{k,\mathrm{Tempered}}^{1/2}$	$\mathcal{C}_{k,\mathrm{RWM}}^{1/2}$
$k = 1$	$\mathrm{Diag}(0.01, 0.01, 0.2, 5)$	$\mathrm{Diag}(0.02, 0.02, 0.2, 5)$
$k = 2$	$\mathrm{Diag}(0.06, 0.06, 0.4, 14)$	–
$k = 3$	$\mathrm{Diag}(0.3, 0.3, 0.6, 20)$	–
$k = 4$	$\mathrm{Diag}(1, 1, 1, 50)$	–

for $N_{\mathrm{RWM}} = KN = 28,000$ iterations, and discard the first 20% of the samples obtained (5600) as a burn-in.

Each individual chain is constructed using Gaussian RWM proposals $q_{\mathrm{prop},k}(\theta_k^n, \cdot) = \mathcal{N}(\theta_k^n, \mathcal{C}_k)$, $k = 1, 2, 3, 4$, 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 0.2. 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 Łącki and Miasojedow (2016), as well as some moderate computational gains when compared to the standard PT.

5.7 High-dimensional acoustic wave inversion

Lastly, we present a high-dimensional example for which we try to invert for the material properties β^2 in (25). For simplicity, we will consider fixed values of $\alpha = 1$, $s_1 = 1.5$, and $s_2 = 1$. In this case, we set $\beta^2 = 10 + \hat{\beta}^2(x)$, where $\hat{\beta}(x)$ is taken to be a realization of a random field discretized on a mesh of $N_x \times N_y$ triangular elements. This modeling choice ensures that β^2 is lower bounded. In this case, we will invert for the nodal values of (the finite element discretization of) $\hat{\beta}$, which will naturally result in a high-dimensional problem. We remark that one is usually interested in including the randomness in β^2, instead of $\hat{\beta}$; however, we purposely choose to do so to induce an explicitly multi-modal posterior, and

as such, to better illustrate the advantages of our proposed methods when sampling from these types of distributions.

We begin by formalizing the finite-element discretization of the parameter space (see e.g., Bui-Thanh and Nguyen (2016) for a more detailed discussion).

Let $\bar{D} = [0, 3] \times [0, 2]$, denote the physical space of the problem and let V_h be a finite-dimensional subspace of $L_2(D)$ arising from a given finite element discretization. We write the finite element approximation $\hat{\beta}_h \in V_h$ of $\hat{\beta}$ as

$$\hat{\beta}(x) \approx \hat{\beta}_h(x) = \sum_{n=1}^{N_v} b_n \phi_n(x),$$

where $\{\phi\}_{n=1}^{N_v}$ are the Lagrange basis functions corresponding to the nodal points $\{x_n\}_{n=1}^{N_v}$, $(b_1, \ldots, b_{N_v})^T =: \theta \in \mathbb{R}^{N_v}$ is the set of nodal parameters and N_v corresponds to the number of vertices used in the FE discretization. Thus, the problem of inferring the distribution of β given some data y, can be understood as inferring the distribution of θ given y. For our particular case, we will discretize D using 28×28 (non-overlapping) piece-wise linear finite elements, which results in $N_v = 841$ and as such $\Theta = \mathbb{R}^{841}$. We consider a Gaussian prior $\mu_{\mathrm{pr},\infty} = \mathcal{N}(0, \mathcal{A}^{-2})$, where $\mathcal{A}$ is a differential operator acting on $L_2(D)$ of the form

$$\mathcal{A} := - a \nabla \cdot (H \nabla) + dI, \quad a, d > 0,$$

together with Robin boundary conditions $\nabla(\cdot) \cdot \hat{n} + \sqrt{ad}(\cdot) = 0$, where, following Villa et al. (2021), H is taken of the form

$$H := \begin{pmatrix} e_1 \sin^2(\ell) + e_2 \cos^2(\ell) & (e_1 - e_2) \sin(\ell) \cos(\ell) \\ (e_1 - e_2) \sin(\ell) \cos(\ell) & e_1 \cos^2(\ell) + e_2 \sin^2(\ell) \end{pmatrix}.$$

Here H models the spatial anisotropy of a Gaussian Random field sampled from $\mu_{\mathrm{pr},\infty}$. It is known that for a two-dimensional (spatial) space, the covariance operator $\mathcal{A}^{-2}$ is symmetric and trace-class (Bui-Thanh and Nguyen 2016), and as such, the (infinite-dimensional) prior measure is well-defined. Thus, we set

$$\hat{\beta}(x) \sim \mu_{\mathrm{pr},\infty},$$

which in turn induces the discretized prior:

$$\hat{\beta}_h(x) \sim \mu_{\mathrm{pr}} := \mathcal{N}(0, \mathcal{A}_h^{-2}),$$

where $\mathcal{A}_h^{-2}$ is a finite-element approximation of $\mathcal{A}$ using 28×28 (non-overlapping) piece-wise linear finite elements. Samples from μ_{pr} are obtained using the FEniCS package (Logg et al. 2012) and the hIPPYlib library Villa et al. (2021).

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

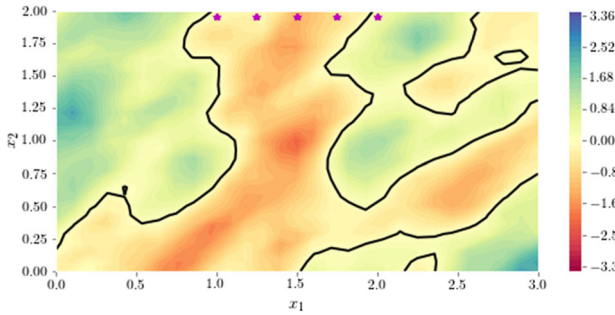|	Mean		Var		$\text{Var}_{\text{RWM}}/\text{Var}$	
	s_1	s_2	s_1	s_2	s_1	s_2
RWM	1.33801	1.54293	9.86×10^{-1}	8.21×10^{-2}	1.000000	1.000
PT	1.50121	1.00829	6.61×10^{-6}	2.77×10^{-4}	149136.1	296.2
PSDPT	1.39775	1.23119	2.48×10^{-1}	6.54×10^{-2}	3.900000	1.200
UGPT	1.50177	1.00711	2.72×10^{-6}	2.38×10^{-4}	361744.5	345.0
WGPT	1.50174	1.00601	2.08×10^{-6}	1.46×10^{-4}	472133.2	558.6

Fig. 8 True field $\hat{\beta}_h^*(x)$. Notice the anisotropy on the field. The magenta points represent the receiver locations. The black line represents the zero-level set of the field

We follow an approach similar to our previous example. We collect data $y \in Y$ by solving Eq. (25) with a force term given by (26) and a true field $\hat{\beta}_h^* \sim \mu_{\text{pr}}$ with $a = 0.1$, $d = 0.5$, $\ell = \pi/4$, $e_1 = 2$ and $e_2 = 0.5$. Such a realization of $\hat{\beta}_h^*$ is shown in Fig. 8.

Furthermore, data is observed at $N_R = 5$ different receiver locations $R_1 = (1.0, 2.0)$, $R_2 = (1.25, 2.0)$, $R_3 = (1.5, 2.0)$, $R_4 = (1.75, 2.0)$, and $R_5 = (2.0, 2.0)$ at $N_T = 600$ equally-spaced time instants between $t = 0$ and $t = 0.6$. The data measured by each receiver is polluted by an (independent) additive Gaussian noise $\eta \sim \mathcal{N}(0, \sigma_{\text{noise}}^2 I_{600 \times 600})$, with $\sigma = 0.021$, which corresponds to roughly a 0.5% noise. Thus, we have that $(Y, \|\cdot\|_Y) = (\mathbb{R}^{5 \times 600}, \|\cdot\|_\Sigma)$. Similarly as in Sect. 5.6, the forward mapping operator $\mathcal{F} : \Theta \mapsto Y$ can be understood as the numerical solution of Eq. (25) evaluated at 600 discrete time instants at each of the 5 receiver locations. Numerical implementation follows a similar setup as in Sect. 5.6, however, for simplicity, we use 28×28 triangular elements to approximate the forward operator $\mathcal{F}$. The Bayesian inverse problem at hand can thus be understood as sampling from the posterior measure μ, which has a Radon-Nikodym derivative with respect to the prior μ_{pr} given by

$$\pi(\theta) = \frac{\mathrm{d}\mu}{\mathrm{d}\mu_{\text{pr}}}(\theta) = \frac{1}{Z} \exp\left(-\frac{1}{2} \|y - \mathcal{F}(\theta)\|_\Sigma^2\right). \qquad (27)$$

The previous BIP has several difficulties; clearly, it is a high-dimensional posterior. Furthermore, just as in the previous example, the underlying mathematical model for the forward operator is a costly time-dependent PDE. By choosing to invert for $\hat{\beta}_h \sim \mu_{\text{pr}}$ (instead of $\beta_h^2 \approx \beta^2$), and since μ_{pr} is centered at zero, we induce a multi-modal posterior, indeed, if the posterior concentrates around $\hat{\beta}_h^*$ it will also have peaks at any other $\hat{\beta}^j$ obtained by flipping the sign of $\hat{\beta}_h^*$ in a concentrated region separated by the zero level set of $\hat{\beta}_h^*$ (we identify 7 regions in Fig. 8). This can be seen in Fig. 9, where we plot 4 posterior samples $\hat{\beta}_h \sim \mu$. Notice the change in sign between some regions. Lastly, as a quantities of interest, we will consider $\mathcal{Q}_1 = \int_D \exp(\hat{\beta}(x))\mathrm{d}x$ and $\mathcal{Q}_2 = \exp(\hat{\beta}(1.5, 1))$. We remark that, although these quantities of interest do not have any meaningful physical interpretation, they are, however, affected by the multi-modality of the posterior, and as such, well suited to exemplify the capabilities of our method.

Given the high-dimensionality of the posterior, we present a slightly different experimental setup in order to estimate $\mathbb{E}_\mu[\mathcal{Q}_i] \approx \hat{\mathcal{Q}}_i$, $i = 1, 2$. In particular, we will use the pre-conditioned Crank-Nicolson (pCN) as a base method, instead of RWM, for the transition kernel p. We compare the quality of our algorithms by examining the variance of the estimators $\hat{\mathcal{Q}}_i$ computed over $N_{\text{runs}} = 50$ independent MCMC runs of each algorithm, with $K = 4$ temperatures for the tempered algorithms given by $T_1 = 1$, $T_2 = 4.57$, $T_3 = 20.89$, $T_4 = 100$. For the tempered algorithms, each estimator is obtained by running the inversion experiment for $N = 4,800$ samples, discarding the first 20% of the samples (800) as a burn-in. For the untempered pCN algorithm, we run the inversion experiment for $N_{\text{pCN}} = KN = 19,200$ iterations, and discard the first 20% of the samples obtained (3840) as a burn-in.

Each individual chain is constructed using pCN proposals $q_{\text{prop},k}(\theta_k^n, \cdot) = \mathcal{N}(\sqrt{1 - \rho_k^2}\theta_k^n, \rho_k^2 \mathcal{A}_h^{-2})$, $k = 1, 2, 3, 4$, with ρ_k described in Table 9. The simple, un-tempered pCN algorithm is run with a step size given by $\rho = \rho_1$. The values of ρ_k are tuned in such a way that the acceptance rate of each chain is around 0.3 and are reported in Table 9. The variance of the estimators obtained with each method is presented in Table 10. Once again, even for this high-dimensional, highly
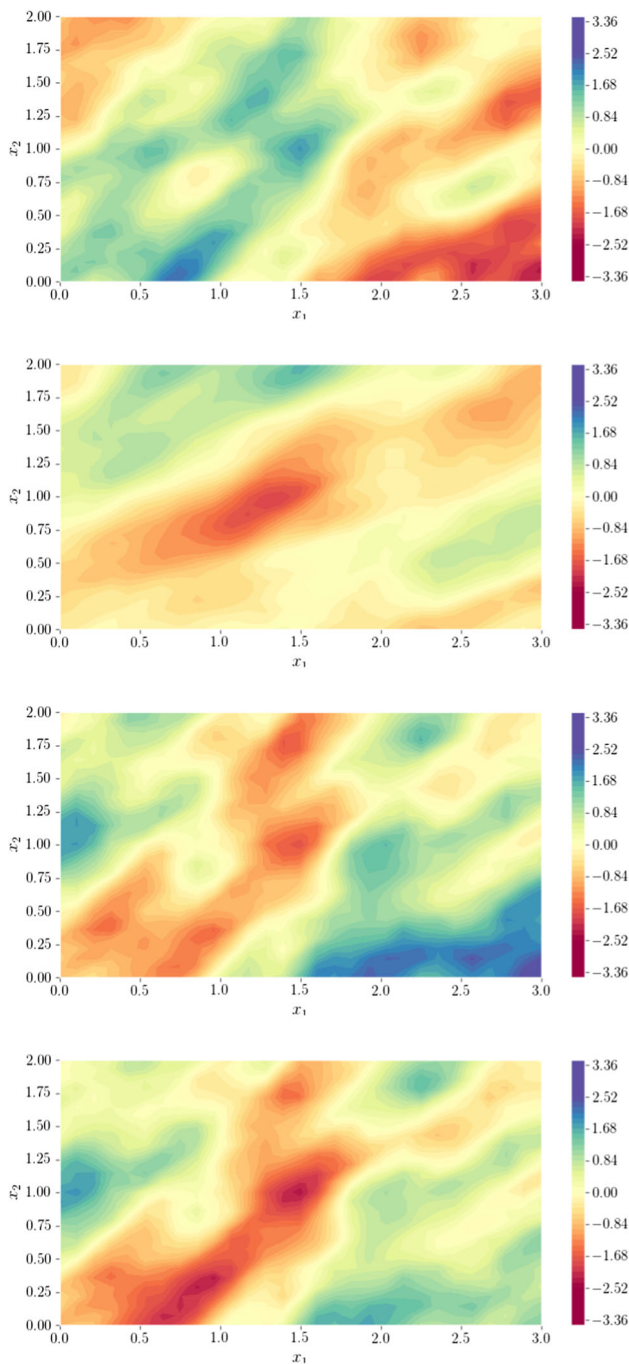
Table 9 Values of ρ_k for the pCN kernel for the high-dimensional wave inversion problem

	$k = 1$	$k = 2$	$k = 3$	$k = 4$
ρ_k	0.1	0.2	0.4	0.8

ods to a BIP context. We demonstrate that such algorithms produce reversible and geometrically-ergodic chains under relatively mild conditions. As shown in Sect. 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 Haario et al. (2006).

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 Dodwell et al. (2015); Madrigal-Cianci et al. (2021). 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 Woodard et al. (2009). 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., id $\notin S_K$) on the UGPT, one could obtain a swapping kernel which is *better* in the so-called Peskun sense, see (Andrieu and Roberts 2009) 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 Beskos et al. (2017); Cui et al. (2016), 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.

Fig. 9 Posterior samples $\hat{\beta}_h \sim \mu$ obtained with the UW GPT algorithm. Notice the resemblance to Fig. 8

multi-modal case, our proposed methods perform considerably better than the other algorithms.

6 Conclusions and future work

In the current work, we have proposed, implemented, and analyzed two versions of the GPT, and applied these meth-

Table 10 Results for the high-dimensional acoustic BIP problem. As for the previous examples, The computational cost is comparable across all algorithms

| | Mean | | Var | | Var$_{\text{pCN}}$/Var | |
	$\hat{\mathcal{Q}}_1$	$\hat{\mathcal{Q}}_2$	$\hat{\mathcal{Q}}_1$	$\hat{\mathcal{Q}}_2$	$\hat{\mathcal{Q}}_1$	$\hat{\mathcal{Q}}_2$
pCN	8.8665	1.5255	5.7362	0.6029	1.00	1.00
PT +pCN	8.7710	1.5311	1.3308	0.1380	4.31	4.36
PSDPT +pCN	8.5546	1.4453	2.1289	0.2666	2.69	2.26
UGPT +pCN	8.7983	1.4614	1.0543	0.1051	5.49	5.73
WGPT +pCN	8.6464	1.4643	1.0126	0.1016	5.74	5.93

Appendix

A Proof of Proposition 7

Proof This proof is partially based on the proof of Theorem 1.2 in Madras and Randall (2002). Let $\boldsymbol{\theta}, \boldsymbol{y} \in \Theta^K$ and define $\bar{f}_\sigma := \boldsymbol{\mu}_\sigma(f)$. The right-most inequality follows from the fact that

$$1 = \mathbb{V}_{\boldsymbol{\mu}_{\text{W}}}[f] = \int_{\Theta^K} f(\boldsymbol{\theta})^2 \boldsymbol{\mu}_{\text{W}}(\mathrm{d}\boldsymbol{\theta})$$

$$= \frac{1}{|S_K|} \sum_{\sigma \in S_K} \int_{\Theta^K} f^2(\boldsymbol{\theta}) \boldsymbol{\mu}_\sigma(\mathrm{d}\boldsymbol{\theta})$$

$$= \frac{1}{|S_K|} \sum_{\sigma \in S_K} \left(\mathbb{V}_{\boldsymbol{\mu}_\sigma}[f] + \bar{f}_\sigma^2 \right) \geq \frac{1}{|S_K|} \sum_{\sigma \in S_K} \mathbb{V}_{\boldsymbol{\mu}_\sigma}[f]$$

We follow a procedure similar to the proof of Madras and Randall (2002) Theorem 1.2 for the lower bound on the variance. We introduce an ordering on $S_K = \sigma_1, \sigma_2, \ldots, \sigma_{|S_K|}$, define the matrix $C \in \mathbb{R}^{|S_K| \times |S_K|}$ as the matrix with entries

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

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

$$2 = 2\mathbb{V}_{\boldsymbol{\mu}_{\text{W}}}[f] = \int_{\Theta^K} \int_{\Theta^K} (f(\boldsymbol{\theta}) - f(\boldsymbol{y}))^2 \left(\frac{1}{|S_K|} \sum_{i=1}^{|S_K|} \boldsymbol{\mu}_{\sigma_i}(\mathrm{d}\boldsymbol{\theta}) \right)$$

$$\times \left(\frac{1}{|S_K|} \sum_{j=1}^{|S_K|} \boldsymbol{\mu}_{\sigma_j}(\mathrm{d}\boldsymbol{y}) \right)$$

$$= \sum_{i,j} \frac{1}{|S_K|^2} C_{ij}. \tag{28}$$

We thus aim at finding an upper bound of Eq. (28) 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 $\boldsymbol{\mu}_{\text{pr}}$) 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, integrating over Θ^K, we get for the diagonal entries of the C matrix:

$$C_{ii} = 2\mathbb{V}_{\boldsymbol{\mu}_{\sigma_i}}[f]$$

$$= \iint (f(\boldsymbol{\theta}) - f(\boldsymbol{y}))^2 \left(\Lambda_{ij} \boldsymbol{\eta}_{ij}(\boldsymbol{\theta}) + (1 - \Lambda_{ij}) \boldsymbol{\varphi}_i(\boldsymbol{\theta}) \right)$$

$$\times \left(\Lambda_{ij} \boldsymbol{\eta}_{ij}(\boldsymbol{y}) + (1 - \Lambda_{ij}) \boldsymbol{\varphi}_i(\boldsymbol{y}) \right) \boldsymbol{\mu}_{\text{pr}}(\mathrm{d}\boldsymbol{\theta}) \boldsymbol{\mu}_{\text{pr}}(\mathrm{d}\boldsymbol{y})$$

$$= \iint (f(\boldsymbol{\theta}) - f(\boldsymbol{y}))^2 \Lambda_{ij}^2 \boldsymbol{\eta}_{ij}(\boldsymbol{\theta}) \boldsymbol{\eta}_{ij}(\boldsymbol{y}) \boldsymbol{\mu}_{\text{pr}}(\mathrm{d}\boldsymbol{\theta}) \boldsymbol{\mu}_{\text{pr}}(\mathrm{d}\boldsymbol{y})$$

$$+ \iint (f(\boldsymbol{\theta}) - f(\boldsymbol{y}))^2 \Lambda_{ij}(1 - \Lambda_{ij}) \boldsymbol{\varphi}_i(\boldsymbol{y}) \boldsymbol{\eta}_{ij}(\boldsymbol{\theta}) \boldsymbol{\mu}_{\text{pr}}(\mathrm{d}\boldsymbol{\theta}) \boldsymbol{\mu}_{\text{pr}}(\mathrm{d}\boldsymbol{y})$$

$$+ \iint (f(\boldsymbol{\theta}) - f(\boldsymbol{y}))^2 \Lambda_{ij}(1 - \Lambda_{ij}) \boldsymbol{\varphi}_i(\boldsymbol{\theta}) \boldsymbol{\eta}_{ij}(\boldsymbol{y}) \boldsymbol{\mu}_{\text{pr}}(\mathrm{d}\boldsymbol{\theta}) \boldsymbol{\mu}_{\text{pr}}(\mathrm{d}\boldsymbol{y})$$

$$+ \iint (f(\boldsymbol{\theta}) - f(\boldsymbol{y}))^2 (1 - \Lambda_{ij})^2 \boldsymbol{\varphi}_i(\boldsymbol{y}) \boldsymbol{\varphi}_i(\boldsymbol{\theta}) \boldsymbol{\mu}_{\text{pr}}(\mathrm{d}\boldsymbol{\theta}) \boldsymbol{\mu}_{\text{pr}}(\mathrm{d}\boldsymbol{y})$$

$$= 2\Lambda_{ij}^2 \mathbb{V}_{\boldsymbol{\eta}_{ij}}[f] + 2(1 - \Lambda_{ij})^2 \mathbb{V}_{\boldsymbol{\varphi}_i}[f] + 2\Lambda_{ij}(1 - \Lambda_{ij})$$

$$\times \iint (f(\boldsymbol{\theta}) - f(\boldsymbol{y}))^2 \boldsymbol{\eta}_{ij}(\boldsymbol{\theta}) \boldsymbol{\varphi}_i(\boldsymbol{\theta}) \boldsymbol{\mu}_{\text{pr}}(\mathrm{d}\boldsymbol{\theta}) \boldsymbol{\mu}_{\text{pr}}(\mathrm{d}\boldsymbol{y}). \tag{29}$$

Notice that Eq. (29) implies that

$$\iint (f(\boldsymbol{\theta}) - f(\boldsymbol{y}))^2 \boldsymbol{\eta}_{ij}(\boldsymbol{\theta}) \boldsymbol{\varphi}_i(\boldsymbol{\theta}) \boldsymbol{\mu}_{\text{pr}}(\mathrm{d}\boldsymbol{\theta}) \boldsymbol{\mu}_{\text{pr}}(\mathrm{d}\boldsymbol{y})$$

$$\leq \frac{\mathbb{V}_{\boldsymbol{\mu}_{\sigma_i}}[f] - \Lambda_{ij}^2 \mathbb{V}_{\boldsymbol{\eta}_{ij}}[f]}{\Lambda_{ij}(1 - \Lambda_{ij})}. \tag{30}$$

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

$$
\begin{aligned}
C_{ij} &= \iint (f(\boldsymbol{\theta}) - f(\boldsymbol{y}))^2 \big[\Lambda_{ij}\boldsymbol{\eta}_{ij}(\boldsymbol{\theta}) \\
&\quad + (1 - \Lambda_{ij})\boldsymbol{\varphi}_i(\boldsymbol{\theta})\big] \big(\Lambda_{ij}\boldsymbol{\eta}_{ij}(\boldsymbol{y}) \\
&\quad + (1 - \Lambda_{ij})\boldsymbol{\psi}_j(\boldsymbol{y})\big)\boldsymbol{\mu}_{\mathrm{pr}}(\mathrm{d}\boldsymbol{\theta})\boldsymbol{\mu}_{\mathrm{pr}}(\mathrm{d}\boldsymbol{y}) \\
&= 2\Lambda_{ij}^2 \mathbb{V}_{\boldsymbol{\eta}_{ij}}[f] \\
&\quad + (1 - \Lambda_{ij})^2 \iint (f(\boldsymbol{\theta}) - f(\boldsymbol{y}))^2 \boldsymbol{\varphi}_i(\boldsymbol{\theta})\boldsymbol{\psi}_j(\boldsymbol{y})\boldsymbol{\mu}_{\mathrm{pr}}(\mathrm{d}\boldsymbol{\theta})\boldsymbol{\mu}_{\mathrm{pr}}(\mathrm{d}\boldsymbol{y}) \\
&\quad + \Lambda_{ij}(1 - \Lambda_{ij}) \iint (f(\boldsymbol{\theta}) - f(\boldsymbol{y}))^2 \\
&\quad \times \big(\boldsymbol{\eta}_{ij}(\boldsymbol{\theta})\boldsymbol{\psi}_j(\boldsymbol{y}) + \boldsymbol{\eta}_{ij}(\boldsymbol{y})\boldsymbol{\varphi}_i(\boldsymbol{\theta})\big)\boldsymbol{\mu}_{\mathrm{pr}}(\mathrm{d}\boldsymbol{\theta})\boldsymbol{\mu}_{\mathrm{pr}}(\mathrm{d}\boldsymbol{y}). \tag{31}
\end{aligned}
$$

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

$$
\begin{aligned}
&\iint (f(\boldsymbol{\theta}) - f(\boldsymbol{y}))^2 \boldsymbol{\varphi}_i(\boldsymbol{\theta})\boldsymbol{\psi}_j(\boldsymbol{y})\boldsymbol{\mu}_{\mathrm{pr}}(\mathrm{d}\boldsymbol{\theta})\boldsymbol{\mu}_{\mathrm{pr}}(\mathrm{d}\boldsymbol{y}) \\
&= \iiint (f(\boldsymbol{\theta}) - f(\boldsymbol{z}) + f(\boldsymbol{z}) - f(\boldsymbol{y}))^2 \boldsymbol{\varphi}_i(\boldsymbol{\theta})\boldsymbol{\psi}_j(\boldsymbol{y})\boldsymbol{\eta}_{ij}(\boldsymbol{z}) \\
&\quad \times \boldsymbol{\mu}_{\mathrm{pr}}(\mathrm{d}\boldsymbol{\theta})\boldsymbol{\mu}_{\mathrm{pr}}(\mathrm{d}\boldsymbol{y})\boldsymbol{\mu}_{\mathrm{pr}}(\mathrm{d}\boldsymbol{z}) \\
&\leq 2 \iiint \big((f(\boldsymbol{\theta}) - f(\boldsymbol{z}))^2 + (f(\boldsymbol{z}) - f(\boldsymbol{y}))^2\big)\boldsymbol{\varphi}_i(\boldsymbol{\theta})\boldsymbol{\psi}_j(\boldsymbol{y})\boldsymbol{\eta}_{ij}(\boldsymbol{z}) \\
&\quad \times \boldsymbol{\mu}_{\mathrm{pr}}(\mathrm{d}\boldsymbol{\theta})\boldsymbol{\mu}_{\mathrm{pr}}(\mathrm{d}\boldsymbol{y})\boldsymbol{\mu}_{\mathrm{pr}}(\mathrm{d}\boldsymbol{z}) \\
&= 2 \iint (f(\boldsymbol{\theta}) - f(\boldsymbol{z}))^2 \boldsymbol{\varphi}_i(\boldsymbol{\theta})\boldsymbol{\eta}_{ij}(\boldsymbol{z})\boldsymbol{\mu}_{\mathrm{pr}}(\mathrm{d}\boldsymbol{\theta})\boldsymbol{\mu}_{\mathrm{pr}}(\mathrm{d}\boldsymbol{z}) \\
&\quad + 2 \iint (f(\boldsymbol{y}) - f(\boldsymbol{z}))^2 \boldsymbol{\psi}_j(\boldsymbol{y})\boldsymbol{\eta}_{ij}(\boldsymbol{z})\boldsymbol{\mu}_{\mathrm{pr}}(\mathrm{d}\boldsymbol{y})\boldsymbol{\mu}_{\mathrm{pr}}(\mathrm{d}\boldsymbol{z}). \tag{32}
\end{aligned}
$$

Thus, from Eqs. (30), (31), and (32) we get

$$
\begin{aligned}
C_{ij} &\leq 2\Lambda_{ij}^2 \mathbb{V}_{\boldsymbol{\eta}_{ij}}[f] + (2(1 - \Lambda_{ij})^2 + \Lambda_{ij}(1 - \Lambda_{ij})) \\
&\quad \times \bigg(\iint (f(\boldsymbol{\theta}) - f(\boldsymbol{y}))^2 \big(\boldsymbol{\eta}_{ij}(\boldsymbol{\theta})\boldsymbol{\psi}_j(\boldsymbol{y}) \\
&\quad + \boldsymbol{\eta}_{ij}(\boldsymbol{y})\boldsymbol{\psi}_i(\boldsymbol{\theta})\big)\boldsymbol{\mu}_{\mathrm{pr}}(\mathrm{d}\boldsymbol{\theta})\boldsymbol{\mu}_{\mathrm{pr}}(\mathrm{d}\boldsymbol{y})\bigg) \\
&= 2\Lambda_{ij}^2 \mathbb{V}_{\boldsymbol{\eta}_{ij}}[f] + (2 - \Lambda_{ij})(1 - \Lambda_{ij}) \\
&\quad \times \frac{\big(\mathbb{V}_{\boldsymbol{\mu}_{\sigma_i}}[f] - \Lambda_{ij}^2 \mathbb{V}_{\boldsymbol{\eta}_{ij}}[f] + \mathbb{V}_{\boldsymbol{\mu}_{\sigma_j}}[f] - \Lambda_{ij}^2 \mathbb{V}_{\boldsymbol{\eta}_{ij}}[f]\big)}{\Lambda_{ij}(1 - \Lambda_{ij})} \\
&= \frac{2 - \Lambda_{ij}}{\Lambda_{ij}}\big(V_{\boldsymbol{\mu}_{\sigma_i}}[f] + V_{\boldsymbol{\mu}_{\sigma_j}}[f]\big) - 4\Lambda_{ij}(1 - \Lambda_{ij})\mathbb{V}_{\boldsymbol{\eta}_{ij}}[f] \\
&\leq \frac{2 - \Lambda_{ij}}{\Lambda_{ij}}\big(V_{\boldsymbol{\mu}_{\sigma_i}}[f] + V_{\boldsymbol{\mu}_{\sigma_j}}[f]\big), \tag{33}
\end{aligned}
$$

since $\Lambda_{ij} \in (0, 1) \; \forall i, j$. Finally, from Eqs. (28) and (33) we get that

$$1 = V_{\boldsymbol{\mu}_{\mathrm{W}}}[f] = \frac{1}{2}\sum_{i,j}\frac{1}{|S_K|^2}C_{ij}$$

$$\leq \frac{1}{2}\frac{1}{|S_K|^2}\sum_{i,j=1}^{|S_K|}\frac{2 - \Lambda_{ij}}{\Lambda_{ij}}\big(V_{\boldsymbol{\mu}_{\sigma_j}}[f] + V_{\boldsymbol{\mu}_{\sigma_j}}[f]\big)$$

$$\leq \frac{2 - \Lambda_m}{\Lambda_m}\bigg(\frac{1}{|S_K|}\sum_{i=1}^{|S_K|}\mathbb{V}_{\boldsymbol{\mu}_{\sigma_i}}[f]\bigg),$$

with $\Lambda_m := \min\limits_{i,j=1,2,\ldots,|S_K|}\{\Lambda_{ij}\} > 0$, and $\Lambda_{i,j}$ as in Assumption C3. Notice that we have used (33) for the first inequality, including the case $i = j$, in the previous equation. This in turn yields the lower bound

$$0 < \frac{\Lambda_m}{2 - \Lambda_m} \leq \bigg(\frac{1}{|S_K|}\sum_{i \in S_K}\mathbb{V}_{\boldsymbol{\mu}_i}[f]\bigg).$$

$\square$

References

Andrieu, C., Roberts, G.O.: The pseudo-marginal approach for efficient Monte Carlo computations. Ann. Stat. **37**(2), 697–725 (2009)

Ash, R.B.: Probability and Measure Theory. Harcourt/Academic Press, Burlington (2000)

Asmussen, S., Glynn, P.W.: Stochastic Simulation: Algorithms and Analysis, 57th edn. Springer, Berlin (2007)

Baxter, J.R., Rosenthal, J.S.: Rates of convergence for everywhere-positive Markov chains. Stat. Probab. Lett. **22**(4), 333–338 (1995)

Beskos, A., Girolami, M., Lan, S., Farrell, P.E., Stuart, A.M.: Geometric MCMC for infinite-dimensional inverse problems. J. Comput. Phys. **335**, 327–351 (2017)

Beskos, A., Jasra, A., Kantas, N., Thiery, A.: On the convergence of adaptive sequential Monte Carlo methods. Ann. Appl. Probab. **26**(2), 1111–1146 (2016). https://doi.org/10.1214/15-AAP1113

Beskos, A., Jasra, A., Muzaffer, E., Stuart, A.: Sequential Monte Carlo methods for Bayesian elliptic inverse problems. Stat. Comp. **25**, 727–737 (2015). https://doi.org/10.1007/s11222-015-9556-7

Brooks, S., Gelman, A., Jones, G., Meng, X.L.: Handbook of Markov Chain Monte Carlo. CRC Press, Boca Raton (2011)

Bui-Thanh, T., Nguyen, Q.P.: Fem-based discretization-invariant mcmc methods for pde-constrained Bayesian inverse problems. Inverse Probl. Imag. **10**(4), 943 (2016)

Cotter, S.L., Roberts, G.O., Stuart, A.M., White, D.: Mcmc methods for functions: modifying old algorithms to make them faster. Stat. Sci. **28**, 424–446 (2013)

Cui, T., Law, K.J., Marzouk, Y.M.: Dimension-independent likelihood-informed MCMC. J. Comput. Phys. **304**, 109–137 (2016)

Dalcín, L., Paz, R., Storti, M.: MPI for python. J. Parallel Distrib. Comput. **65**(9), 1108–1115 (2005)

Desjardins, G., Courville, A., Bengio, Y., Vincent, P., Delalleau, O.: Tempered Markov chain Monte Carlo for training of restricted boltzmann machines. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, pp. 145–152 (2010)

Dia, B.M.: A continuation method in Bayesian inference. arXiv:1911.11650 (2019)

Dodwell, T.J., Ketelsen, C., Scheichl, R., Teckentrup, A.L.: A hierarchical multilevel Markov chain Monte Carlo algorithm with applications to uncertainty quantification in subsurface flow. SIAM/ASA J. Certain. Quantif. **3**(1), 1075–1108 (2015)

Doll, J., Plattner, N., Freeman, D.L., Liu, Y., Dupuis, P.: Rare-event sampling: occupation-based performance measures for parallel tempering and infinite swapping Monte Carlo methods. J. Chem. Phys. **137**(20), 204112 (2012)

Dupuis, P., Liu, Y., Plattner, N., Doll, J.D.: On the infinite swapping limit for parallel tempering. Multiscale Model. Simul. **10**(3), 986–1022 (2012)

Earl, D.J., Deem, M.W.: Parallel tempering: theory, applications, and new perspectives. Phys. Chem. Chem. Phys. **7**(23), 3910–3916 (2005)

Haario, H., Laine, M., Mira, A., Saksman, E.: DRAM: efficient adaptive MCMC. Stat. Comput. **16**(4), 339–354 (2006)

Haario, H., Saksman, E., Tamminen, J.: An adaptive Metropolis algorithm. Bernoulli **7**(2), 223–242 (2001)

Hairer, M., Stuart, A.M., Vollmer, S.J., et al.: Spectral gaps for a Metropolis-Hastings algorithm in infinite dimensions. Ann. Appl. Probab. **24**(6), 2455–2490 (2014)

Hastings, W.K.: Monte Carlo Sampling Methods Using Markov Chains and Their Applications. Oxford University Press, Oxford (1970)

Kahle, C., Lam, K., Latz, J., Ullmann, E.: Bayesian parameter identification in Cahn-Hilliard models for biological growth. SIAM/ASA J. Uncertain. Quantif. **7**(2), 526–552 (2019). https://doi.org/10.1137/18M1210034

Kantas, N., Beskos, A., Jasra, A.: 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). https://doi.org/10.1137/130930364

Łącki, M.K., Miasojedow, B.: State-dependent swap strategies and automatic reduction of number of temperatures in adaptive parallel tempering algorithm. Stat. Comput. **26**(5), 951–964 (2016)

Latz, J.: On the well-posedness of Bayesian inverse problems. arXiv:1902.10257 (2019)

Latz, J., Papaioannou, I., Ullmann, E.: Multilevel Sequential^2 Monte Carlo for Bayesian inverse problems. J. Comput. Phys. **368**, 154–178 (2018) https://doi.org/10.1016/j.jcp.2018.04.014. http://www.sciencedirect.com/science/article/pii/S0021999118302286

Logg, A., Mardal, K.A., Wells, G.: Automated solution of differential equations by the finite element method: the FEniCS book, vol. 84. Springer (2012)

Lu, J., Vanden-Eijnden, E.: Infinite swapping replica exchange molecular dynamics leads to a simple simulation patch using mixture potentials. J. Chem. Phys. **138**(8), 084105 (2013)

Madras, N., Randall, D., et al.: Markov chain decomposition for convergence rate analysis. Ann. Appl. Probab. **12**(2), 581–606 (2002)

Madrigal-Cianci, J.P., Nobile, F., Tempone, R.: Analysis of a class of Multi-Level Markov Chain Monte Carlo algorithms based on Independent Metropolis-Hastings. arXiv:2105.02035 (2021)

Marinari, E., Parisi, G.: Simulated tempering: a new Monte Carlo scheme. Europhys. Lett. (EPL) **19**(6), 451–458 (1992). https://doi.org/10.1209/0295-5075/19/6/002

Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E.: Equation of state calculations by fast computing machines. J. Chem. Phys. **21**(6), 1087–1092 (1953)

Meyn, S.P., Tweedie, R.L.: Markov Chains and Stochastic Stability. Springer, Berlin (2012)

Miasojedow, B., Moulines, E., Vihola, M.: An adaptive parallel tempering algorithm. J. Comput. Graph. Stat. **22**(3), 649–664 (2013)

Motamed, M., Appelo, D.: Wasserstein metric-driven bayesian inversion with applications to signal processing. Int. J. Uncertain. Quantif. **9**(4), 395–414 (2019)

Owen, A.B.: Monte Carlo theory, methods and examples (2013). https://statweb.stanford.edu/~owen/mc/

Plattner, N., Doll, J., Dupuis, P., Wang, H., Liu, Y., Gubernatis, J.: An infinite swapping approach to the rare-event sampling problem. J. Chem. Phys. **135**(13), 134111 (2011)

Quarteroni, A., Quarteroni, S.: Numerical Models for Differential Problems, vol. 2. Springer, Berlin (2009)

Rudolf, D.: Explicit error bounds for Markov chain Monte Carlo. Dissertationes Math. **485**, 1–93 (2012). https://doi.org/10.4064/dm485-0-1

Schillings, C., Stuart, A.M.: Analysis of the ensemble Kalman filter for inverse problems. SINUM **55**, 1264–1290 (2017). https://doi.org/10.1137/16M105959X

Van Der Sluys, M., Raymond, V., Mandel, I., Röver, C., Christensen, N., Kalogera, V., Meyer, R., Vecchio, A.: Parameter estimation of spinning binary inspirals using Markov chain Monte Carlo. Class. Quantum Gravity **25**(18), 184011 (2008)

Villa, U., Petra, N., Ghattas, O.: HIPPYlib: an extensible software framework for large-scale inverse problems governed by PDEs: part I: deterministic inversion and linearized Bayesian inference. ACM Trans. Math. Softw. (2021). https://doi.org/10.1145/3428447

Vrugt, J.A., Ter Braak, C., Diks, C., Robinson, B.A., Hyman, J.M., Higdon, D.: Accelerating Markov chain Monte Carlo simulation by differential evolution with self-adaptive randomized subspace sampling. Int. J. Nonlinear Sci. Numer. Simul. **10**(3), 273–290 (2009)

Woodard, D.B., Schmidler, S.C., Huber, M., et al.: Conditions for rapid mixing of parallel and simulated tempering on multimodal distributions. Ann. Appl. Probab. **19**(2), 617–640 (2009)

Yu, T.Q., Lu, J., Abrams, C.F., Vanden-Eijnden, E.: Multiscale implementation of infinite-swap replica exchange molecular dynamics. Proc. Natl. Acad. Sci. **113**(42), 11744–11749 (2016). https://doi.org/10.1073/pnas.1605089113