MATHICSE
Institute of Mathematics
School of Basic Sciences

MATHICSE Technical Report

12 March 2020

Generalized Parallel Tempering on Bayesian Inverse Problems

Jonas Latz, Juan P. Madrigal-Cianci, Fabio Nobile, Raul Tempone.

https://mathicse.epfl.ch/

Address: EPFL - SB – INSTITUTE of MATHEMATICS – Math
(Bâtiment MA) Station 8 - CH-1015 - Lausanne - Switzerland

GENERALIZED PARALLEL TEMPERING ON BAYESIAN INVERSE PROBLEMS

BY JONAS LATZ[1], JUAN P. MADRIGAL-CIANCI[2,*], FABIO NOBILE[2,†] AND RAÚL TEMPONE[3]

[1] *Department of Applied Mathematics and Theoretical Physics, University of Cambridge.* jl2160@cam.ac.uk

[2] *SB-MATH-CSQI, École Polytechnique Fédérale de Lausanne.* [*]juan.madrigalcianci@epfl.ch; [†]fabio.nobile@epfl.ch

[3] *Computer, Electrical and Mathematical Sciences and Engineering, KAUST, and Alexander von Humboldt professor in Mathematics of Uncertainty Quantification, RWTH Aachen University.* raul.tempone@kaust.edu.sa, tempone@uq.rwth-aachen.de

In the current work we present two generalizations of the Parallel Tempering algorithm, inspired by the so-called continuous-time Infinite Swapping algorithm. Such a method, found its origins in the molecular dynamics community, and can be understood as the limit case of the continuous-time Parallel Tempering algorithm, where the (random) time between swaps of states between two parallel chains goes to zero. Thus, swapping states between chains occurs continuously. In the current work, we extend this idea to the context of time-discrete Markov chains and present two Markov chain Monte Carlo algorithms that follow the same paradigm as the continuous-time infinite swapping procedure. We analyze the convergence properties of such discrete-time algorithms in terms of their spectral gap, and implement them to sample from different target distributions. Numerical results show that the proposed methods significantly improve sampling efficiency over more traditional sampling algorithms such as Random Walk Metropolis and (traditional) Parallel Tempering.

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 which models 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), tend to encounter difficulties on the geometry of the posterior distribution. This in turn can make the computational time needed to properly *explore* these complicated target distributions arbitrarily long. Some recent works address these issues by employing Markov transitions kernels that use geometric information [4]; 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* [15, 23, 28, 35]. Of particular importance for the work presented here is the Parallel Tempering (PT) algorithm [15, 28]

MSC 2010 subject classifications: Primary 60J22, 60J20, 62F15; secondary 65C05

Keywords and phrases: Bayesian inversion, parallel tempering, infinite swapping, Markov chain Monte Carlo, uncertainty quantification

(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 *smoothed* (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 typically be able to better explore the parameter space. Thus, by proposing to exchange states between chains that target posteriors at different temperatures, it is possible for the chain of interest (i.e., the one targeting μ) to mix faster, and to avoid the undesirable behavior of some MCMC samplers, to get "stuck" in a mode. Moreover, the fact that such an exchange of states is accepted with the typical MH acceptance-rejection rule, will guarantee that the chain targeting μ remains invariant with respect to such probability measure [15].

Tempering ideas have been successfully used to sample from posterior distributions arising in different fields of science, ranging from astrophysics to machine learning [11, 15, 28, 34]. [25, 36] 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 [26] uses a single chain and varies the temperature within this chain. In addition, tempering forms the basis of efficient particle filtering methods for stationary model parameters in Sequential Monte Carlo settings [5, 6, 20, 21, 23] and Ensemble Kalman Inversion [8]. A generalization over the PT approach, originating from the molecular dynamics community, is the so-called *Infinite Swapping (IS)* algorithm [14, 29]. 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 [14] 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 estimators 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 [13, 24, 29]; 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 Markov chain Monte Carlo context, and analyzing the theoretical properties of such samplers. We will refer to these algorithms as *Generalized Parallel Tempering* (GPT). We remark, however, that these methods are *not* a time discretization of the continuous-time Infinite Swapping presented in [14], but, in fact, a discrete-time Markov process inspired by the ideas presented therein. We now summarize the main contributions of this work.

First, inspired by the work in [13], we propose two discrete-time MCMC generalizations of the PT algorithm in the Bayesian inverse problem setting. Indeed, we introduce a common MCMC framework for both PT the proposed methods.

Then, we analyze the convergence of both proposed algorithms and prove some of their theoretical properties, such as reversibility, existence of a positive L_2-spectral gap, and geometric ergodicity. While the reversibility guarantees that the chain is targeting the desired invariant probability measure, the existence of an L_2-spectral gap and geometric ergodicity quantify the speed of convergence of an MCMC chain to its invariant measure, and provide non-asymptotic error bounds for an ergodic estimator based on the samples from such a chain. We note that our estimates for convergence for the GPT algorithms presented herein are not based on temperature analysis or domain decomposition, as done for PT in [36], for instance. Improving on such analysis will be the subject of a future work.

Finally, we implement the proposed GPT algorithms for simple Bayesian inverse problems and compare their efficiency to that of Random walk Metropolis (RWM) and PT. Even for these simple experiments, we have achieved improvements in terms of computational efficiency of GPT over RWM and PT, thus making the proposed methods attractive from both a theoretical and computational perspective. The rest of this paper is organized as follows. Section 2 is devoted to the introduction of the notation, Bayesian inverse problems, and Markov chain Monte Carlo methods. In Section 3, we provide a brief review of (traditional) PT (Section 3.2), and introduce two versions of the GPT algorithm in Sections 3.3 and 3.4). In Section 4, we recall some of the standard theory of Markov chains in Section 4.1 and state the main theoretical result of the current work (Theorem 4.6) in Section 4.2. The proof of such a theorem is given by a series of Propositions and Lemmata in Section 4.2. We present some numerical experiments in Section 5, and draw some conclusions in Section 6.

2. Problem setting.

2.1. *Notation.* Let $(W, \|\cdot\|)$ be a separable Banach space with associated Borel σ-algebra $\mathcal{B}(W)$, and let ν_W be a σ-finite "reference" measure on W. For any measure μ on $(W, \mathcal{B}(W))$ that is absolutely continuous with respect to ν_W (in short $\mu \ll \nu_W$), we define the Radon-Nikodym derivative $\pi_\mu := \frac{\mathrm{d}\mu}{\mathrm{d}\nu_W}$.

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

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

Let now W_1, W_2 be two Banach spaces with reference measures ν_{W_1}, ν_{W_2}, and let $\mu_1 \ll \nu_{W_1}, \mu_2 \ll \nu_{W_2}$ be two probability measures, with corresponding densities (with respect to ν_{W_k}, for $k = 1, 2$) given by π_1, π_2. The *product* of these two measures is defined by

$$\boldsymbol{\mu}(A) = (\mu_1 \times \mu_2)(A) = \iint_A \pi_1(\theta_1)\pi_2(\theta_2)\nu_{W_1}(\mathrm{d}\theta_1)\nu_{W_2}(\mathrm{d}\theta_2), \quad \forall A \in \mathcal{B}(W_1 \times W_2).$$

In general, we will write product measures (and their respective product densities) with a bold symbol. Central to the work presented here is the concept of the Markov transition kernel, defined as follows:

DEFINITION 2.1 (Markov transition kernel, [32]). 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 $(W, \mathcal{B}(W))$.

Loosely speaking, $p(\theta, A)$ can be interpreted as the (conditional) probability of moving to a set $A \in \mathcal{B}(W)$ given that the chain is in a current state $\theta \in W$.

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 $(W, \mathcal{B}(W))$. Throughout this work, we will make the distinction between Markov kernel, denoted by lower case p or q, and Markov operator, written with an upper case P or Q. The Markov operator associated to a Markov kernel is defined as follows:

DEFINITION 2.2 (Markov operator, [30]). Let $p : W \times \mathcal{B}(W) \mapsto [0,1]$ be a Markov kernel on a Banach space W, let $f : W \mapsto \mathbb{R}$ be a measurable function on $(W, \mathcal{B}(W))$, and let $\nu \in \mathcal{M}(W)$. We denote by P the Markov transition operator, which acts to the left on measures, $\nu \mapsto \nu P \in \mathcal{M}(W)$, and to the right on functions, $f \mapsto Pf$, measurable on $(W, \mathcal{B}(W))$, such that

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

Additionally, throughout the work presented herein, we will consider the tensor product between Markov operators, defined as follows:

DEFINITION 2.3 (Tensor product Markov operator). Let W_1, W_2 be two separable Banach spaces and $P_k,\ k = 1, 2$, be Markov transition operators associated to kernels $p_k : W_k \times \mathcal{B}(W_k) \mapsto [0,1]$. We define the *tensor product Markov operator* $\mathbf{P} := P_1 \otimes P_2$ as the Markov operator associated with the product measure $\mathbf{p}(\boldsymbol{\theta}, \cdot) = p_1(\theta_1, \cdot) \times p_2(\theta_2, \cdot),\ \boldsymbol{\theta} = (\theta_1, \theta_2) \in W_1 \times W_2$. In particular, $\boldsymbol{\nu}\mathbf{P}$ is the measure on $(W_1 \times W_2, \mathcal{B}(W_1 \times W_2))$ that satisfies

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

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

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

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

In practice, $\mathbf{P}$ can be understood by independently applying two Markov kernels p_1, p_2 to the components represented by some measure $\boldsymbol{\nu}$.

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:

DEFINITION 2.4 (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

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

Clearly, if a Markov kernel is reversible with respect to a measure ν, then the associated Markov operator P has ν as an invariant measure. The reverse is not true, in general. For two given ν-invariant Markov operators P_1, P_2, we say that P_1P_2 is a *composition* of Markov operators. We remark that, in general, $P_1P_2 \neq P_1 \otimes P_2$. Furthermore, given a composition of K ν-invariant Markov operators $P_c := P_1P_2 \dots P_K$, we say that P_c is *palindromic* if $P_1 = P_K$, $P_2 = P_{K-1}, \dots, P_k = P_{K-k+1},\ k = 1, 2 \dots, K$. It is known (see, e.g., [7, 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 σ-algebras $\mathcal{B}(\Theta)$, $\mathcal{B}(Y)$, and let us define the forward operator $\mathcal{F} : \Theta \to Y$. In inverse problems, we use some data $y \in Y$, usually polluted by some random noise $\eta \sim \mu_{\text{noise}}$, $\eta \in Y$, to determine a possible state $\theta \in \Theta$ that may have generated the data. Assuming an additive noise model, the relationship between θ and y is given by:

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

for some measure μ_{noise} assumed to have a density π_{noise} with respect to some reference measure ν_Y on Y. Here, θ can be a set of parameters of a possibly non-linear Partial Differential Equation (PDE) modeled by $\mathcal{F}$, for example. On a Bayesian setting, we consider the parameter θ to be uncertain and model it as a random variable with a given *prior* measure μ_{prior} on $(\Theta, \mathcal{B}(\Theta))$. Such a prior measure models the knowledge we have on the uncertainty in θ, before observing the data y. If we further assume that the noise η and θ are statistically independent (when seen as random variables on their respective spaces), then, we have that $\mathbb{P}(y - \mathcal{F}(\theta) \in \cdot|\theta) = \mathbb{P}(\eta \in \cdot)$, i.e., $y - \mathcal{F}(\theta)$ conditioned on θ has the same distribution as η). Thus, we define the *likelihood* function

$$\pi(y|\theta) := \pi_{\text{noise}}(y - \mathcal{F}(\theta)).$$

Throughout this work, we assume that the likelihood is strictly positive μ_{prior}-a.s. and often write its density in terms of a non-negative *potential* function $\Phi(\theta; y) : \Theta \times Y \mapsto [0, \infty)$:

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

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 $\left\|y - \mathcal{F}(\theta)\right\|_Y$. Assuming that the prior measure μ_{prior} has a density π_{prior} with respect to some σ-finite measure ν_Θ, we have from Bayes' Theorem (see, for example, [22, Theorem 2.5]) that

$$\pi(\theta) := \pi(\theta|y) = \frac{1}{Z}\pi_{\text{noise}}(y - \mathcal{F}(\theta))\pi_{\text{prior}}(\theta), \quad \text{with} \quad Z := \int_\Theta \exp(-\Phi(\theta; y))\mu_{\text{prior}}(\mathrm{d}\theta). \tag{4}$$

where μ (with corresponding ν_Θ-density π) is referred to as the *posterior measure*. The Bayesian approach to the inverse problem consists of updating our knowledge concerning the parameter θ, i.e., the prior, given the information that we observed in Equation (2). One way of doing so is to generate samples from the posterior measure μ. However, it is generally not possible to directly sample from μ given that the normalization constant Z is usually not known and intractable to compute. A common method for performing such a task is to use Markov chains Monte Carlo (MCMC) algorithms, as detailed in the next section. Once samples $\{\theta^n\}_{n=1}^N$ drawn approximately from μ have been obtained by some MCMC algorithm, the posterior expectation $\mathbb{E}_\mu[\mathcal{Q}]$ of some μ-integrable quantity of interest $\mathcal{Q} : \Theta \mapsto \mathbb{R}$ can be approximated by the following ergodic estimator

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

2.3. *Markov Chain Monte Carlo and tempering.* The main idea behind using Markov chain Monte Carlo methods to sample a measure of interest μ on $(\Theta, \mathcal{B}(\Theta))$, is to create a Markov chain whose initial state θ^0 has some distribution $\nu \in \mathcal{M}(\Theta)$ and whose Markov operator P is invariant with respect to μ, i.e., $\mu P = \mu$. The Markov chain $\{\theta^n\}_{n=0}^N$ is then generated by sampling $\theta^n \sim p(\theta^{n-1}, \cdot), \forall n \in \mathbb{N}$. One of the most common approaches for performing such a task is the Metropolis-Hastings algorithm [19, 27]. Let $q_{\text{prop}} : \Theta \times \mathcal{B}(\Theta) \mapsto$

$[0,1]$ be an auxiliary kernel. The Metropolis-Hastings algorithm works as follows. For $n = 1, 2, \ldots,$ a candidate state θ^* is sampled from $q_{\text{prop}}(\theta^n, \cdot)$, and proposed as the new state of the chain at step $n+1$. Such a state is then accepted (i.e., we set $\theta^{n+1} = \theta^*$), with probability α_{MH},

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

otherwise the current state is retained, i.e., $\theta^{n+1} = \theta^n$. The Metropolis-Hastings algorithm induces the *Markov transition kernel* $p : \Theta \times \mathcal{B}(\Theta) \mapsto [0,1]$

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

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 $q_{\text{prop}}(\theta^n, \cdot)$. Such is the case for *random walk Metropolis* or *preconditioned Crank Nicolson*, where $q_{\text{prop}}(\theta^n, \cdot) = \mathcal{N}(\theta^n, \Sigma)$ or $q_{\text{prop}}(\theta^n, \cdot) = \mathcal{N}(\sqrt{1-\rho^2}\theta^n, \rho\Sigma)$, $0 < \rho < 1$, respectively. However, these types of *localized* proposals tend to present some undesirable behaviors when sampling from certain *difficult* measures, which are, for example, concentrated over a manifold or are multi-modal [15]. In the first case, in order to avoid a large rejection rate, the "step-size" $\left\|\Sigma^{1/2}\right\|$ 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 $\mu_k, \mu_{\text{prior}}$ be probability measures on $(\Theta, \mathcal{B}(\Theta))$, $k = 1, \ldots, K$, such that all μ_k are absolutely continuous with respect to μ_{prior}, 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_{prior} corresponds to the prior measure and $\mu_k, k = 1, \ldots, K$ correspond to posterior measures associated to different temperatures. Denoting by π_k the μ_{prior}-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 $Z_k := \int_\Theta e^{-\Phi(\theta;y)/T_k} \mu_{\text{prior}}(\mathrm{d}\theta)$, and with $\Phi(\theta; y)$ as the potential function defined in (3). In the case where $T_K = \infty$, we set $\mu_K = \mu_{\text{prior}}$. 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" 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 $\mu_k, \mu_{k+1},\ k \in \{1, \ldots, K-1\}$. 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 $\{1, 2, \ldots, K\}$. Notice that Algorithm 1 only considers pairwise swaps. In the GPT framework we effectively consider *all* $K!$ possible swaps, and accept the proposed swap with probability 1. The construction of the GPT framework will be discussed in the next section.

Algorithm 1 Simple PT.

function SIMPLE PT($N, \{p_k\}_{k=1}^N, \{\pi_k\}_{k=1}^N, \mu_{\text{prior}}$)
  Sample $\theta_k^{(1)} \sim \mu_{\text{prior}},\ k = 1, \dots, K$
  for $n = 1, 2, \dots, N-1$ **do** ▷ Do one step of MH on each chain
    for $k = 1, \dots, K$ **do**
      Sample $\theta_k^{(n+1)} \sim p_k(\theta_k^{(n)}, \cdot)$
    end for
    for $k = 1, 2, \dots, K-1$ **do** ▷ Swap states
      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. Here, we use PT to, for instance, simulate a system's energy at different temperatures and to prevent a *critical slow down* if the temperature is small. In continuous-time PT, the swapping of the states is controlled by a Poisson process on the set $\{1, \dots, K\}$. Infinite Swapping is the limiting algorithm obtained by letting the waiting times of this Poisson process go to zero. Hence, we swap the states of the chain infinitely often over a finite time interval. We refer to [14] 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 $\boldsymbol{\mu}$. We refer to this approach as *Weighted Generalized Parallel Tempering (WGPT)*. We begin by introducing a common framework to both PT and both 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 $\Theta^K := \bigtimes_{k=1}^K \Theta$,with associated product σ-algebra $\mathcal{B}^K := \bigotimes_{k=1}^K \mathcal{B}(\Theta)$, as well as the product measures on $(\Theta^K, \mathcal{B}^K)$

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

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

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

with $\pi_i(\theta)$ added subscript given as in (5). The idea behind the tempering methods presented herein is to sample from $\boldsymbol{\mu}$ (as opposed to solely sampling from μ_1) by creating a Markov

chain obtained from the successive application of two $\boldsymbol{\mu}$-invariant Markov kernels $\mathbf{p}$ and $\mathbf{q}$, to some initial distribution $\boldsymbol{\nu}$, usually chosen to be the prior $\boldsymbol{\mu}^0$. Each kernel acts as follows. Given the current state added subscript $\boldsymbol{\theta}^n = (\theta_1^n, \dots, \theta_K^n)$, the kernel $\mathbf{p}$, which we will call the *standard MCMC kernel*, proposes a new, intermediate state $\tilde{\boldsymbol{\theta}}^{n+1} = (\tilde{\theta}_1^{n+1}, \dots, \tilde{\theta}_K^{n+1})$, possibly following the Metropolis-Hastings algorithm (or any other algorithm that generates a $\boldsymbol{\mu}$-invariant Markov operator). Typically, $\mathbf{p}$ is a product kernel, meaning that each component $\tilde{\theta}_k^n$, $k = 1 \dots, K$, is generated independently of the others. Then, the *swapping kernel* $\mathbf{q}$ proposes a new state $\boldsymbol{\theta}^{n+1} = (\theta_1^{n+1}, \dots, \theta_K^{n+1})$ by introducing an "interaction" between the components of $\tilde{\boldsymbol{\theta}}^{(n+1)}$. This interaction step can be achieved, e.g., in the case of PT, by proposing to swap two components at two consecutive temperatures, i.e., components k and $k+1$, and accepting this swap with a certain probability given by the usual Metropolis-Hastings acceptance-rejection rule. In general, the swapping kernel is usually applied every N_s steps of the chain. We will devote the following subsection to the construction of the swapping kernel $\mathbf{q}$.

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

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

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

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

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

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

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

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

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 $\alpha_{\mathrm{swap}}(\boldsymbol{\theta}, \sigma)$. Notice that the swapping kernel follows a Metropolis-Hastings-like procedure with "proposal" distribution $r(\boldsymbol{\theta}, \sigma)$ and acceptance probability $\alpha_{\mathrm{swap}}(\boldsymbol{\theta}, \sigma)$. Moreover, such a kernel is reversible with respect to $\boldsymbol{\mu}$, since it is a Metropolis-Hastings type kernel.

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

PROOF. See Appendix A.1. □

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 $\sigma_{i,j} \in S_K$ 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 $\sigma(k) = k$, $k \neq i, k \neq j$). 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:

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

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

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

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

In practice, however, the PT algorithm considers various sequential swaps between chains, which can be understood by applying the composition of kernels $\mathbf{q}_{i,j}^{(\text{PT})}\mathbf{q}_{k,\ell}^{(\text{PT})}\ldots$ at every swapping step. In its most common form [7, 15, 28], the PT algorithm, hereafter referred to as canonical 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 [15], proposes to swap states every $N_s \geq 1$ steps of MCMC. The complete kernel for the PT kernel is then given by [7, 15, 28]

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

where $\mathbf{p}$ is a standard reversible Markov transition kernel used to evolve the individual chains independently. Although the kernel $\mathbf{p}$ as well as each of the $\mathbf{q}_{i,i+1}$ are $\boldsymbol{\mu}$-reversible, notice that (8) does not have a palindromic structure, and as such it is not necessarily $\boldsymbol{\mu}$-reversible. One way of making the PT algorithm reversible with respect to $\boldsymbol{\mu}$ (although not very common in practice, to the best of the authors' knowledge) is to consider the palindromic form

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

3.3. *Unweighted Generalized Parallel Tempering.* The idea behind the Unweighted Generalized Parallel Tempering algorithm is to generalize PT so that (i) $N_s = 1$ provides a proper mixing of the chains, (ii) the algorithm is reversible with respect to $\boldsymbol{\mu}$, and (iii) the algorithm considers all possible swaps, 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

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

Clearly, (9) is a swapping ratio according to Definition 3.1. 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 σ in S_K. This permutation σ is then accepted with probability $\alpha_{\mathrm{swap}}^{(\mathrm{UW})}$, given by

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

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 $\alpha_{\mathrm{swap}}(\boldsymbol{\theta},\sigma) = \alpha_{\mathrm{swap}}^{(\mathrm{UW})}(\boldsymbol{\theta},\sigma)$ so that $\alpha_{\mathrm{swap}}^{(\mathrm{UW})}(\boldsymbol{\theta},\sigma) = 1$. Indeed, if we further examine Equation (10), we can see that

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

In practice, this means that the proposed permuted state is accepted with probability 1. We define the swapping kernel for this process.

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

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

Applying this swapping kernel successively with the kernel $\mathbf{p}$ 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* $\mathbf{p}^{(\mathrm{UW})}$. Notice that $\mathbf{p}^{(\mathrm{UW})}$ is a *palindromic* composition of kernels, which is reversible with respect to $\boldsymbol{\mu}$, and as such, $\mathbf{p}^{(\mathrm{UW})}$ will also be reversible with respect to $\boldsymbol{\mu}$ [7]. 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.

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**
    Sample $\boldsymbol{\theta}_\sigma^{(n)} \sim \mathbf{q}^{(\mathrm{UW})}(\boldsymbol{\theta}^{(n)}, \cdot)$ ▷ first swapping kernel
    Sample $\boldsymbol{z}^{(n+1)} \sim \mathbf{p}(\boldsymbol{\theta}_\sigma^{(n)}, \cdot)$ ▷ Markov transition kernel $\mathbf{p}$ kernel
    Sample $\boldsymbol{\theta}^{(n+1)} \sim \mathbf{q}^{(\mathrm{UW})}(\boldsymbol{z}^{(n+1)}, \cdot)$ ▷ second swapping kernel
  end for
  Output $\{\theta_1^{(n)}\}_{n=1}^N$
end function

REMARK 3.7. 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 sample mean estimator $\widehat{\mathcal{Q}}_{(\mathrm{UW})}$:

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

3.4. *Weighted Generalized Parallel Tempering.* Following the intuition of the continuous-time Infinite Swapping approach of [14, 29],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 stationary 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, $w_\sigma(\boldsymbol{\theta}) \neq r^{(\mathrm{UW})}(\boldsymbol{\theta}, \sigma)$, with w_σ defined as in (12).

DEFINITION 3.8. We define the *Weighted Generalized Parallel Tempering* kernel $\mathbf{p}^{(\mathrm{W})} : \Theta^K \times B^K \mapsto [0, 1]$ as the following state-dependent, convex combination of kernels:

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

Thus, the WGPT chain is obtained by iteratively applying $\mathbf{p}^{(\mathrm{W})}$. We show in Lemma 4.9 that the resulting Markov chain has invariant measure

$$\boldsymbol{\mu}_{\mathrm{W}} = \frac{1}{|S_K|} \sum_{\sigma \in S_K} \boldsymbol{\mu}_\sigma = \tilde{\mu} \times \cdots \times \tilde{\mu}, \tag{13}$$

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

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

and a similar average and then tensorization representation applies to $\boldsymbol{\pi}_{\mathrm{W}}$. We remark that this measure 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) estimator of the posterior mean $\widehat{\mathcal{Q}}_{(\mathrm{W})}$ of a quantity of interest $\mathcal{Q}$ by

$$\begin{aligned} \mu(\mathcal{Q}) \approx \widehat{\mathcal{Q}}_{(\mathrm{W})} &= \frac{1}{|S_K|}\frac{1}{N}\sum_{\sigma \in S_K}\sum_{n=1}^{N} \frac{\boldsymbol{\pi}(\boldsymbol{\theta}_\sigma^{(n)})}{\boldsymbol{\pi}_{\mathrm{W}}(\boldsymbol{\theta}_\sigma^{(n)})}\mathcal{Q}(\theta_{\sigma(1)}^{(n)}) \\ &= \frac{1}{|S_K|}\frac{1}{N}\sum_{\sigma \in S_K}\sum_{n=1}^{N} \widehat{w}(\boldsymbol{\theta}^{(n)}, \sigma)\mathcal{Q}(\theta_{\sigma(1)}^{(n)}), \end{aligned} \tag{14}$$

where we have denoted the importance sampling weights by $\widehat{w}(\boldsymbol{\theta}, \sigma) := \frac{\boldsymbol{\pi}(\boldsymbol{\theta}_\sigma)}{\boldsymbol{\pi}_{\mathrm{W}}(\boldsymbol{\theta}_\sigma)}$ 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 $\mathcal{Q}(\theta_k^{(n)})$, $k = 1, 2, \ldots, K$. 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({p_σ}_{σ∈S_K}, N, ν)
    Sample θ^(1) ~ ν
    for n = 1, 2, ..., N − 1 do
        Sample σ ~ {w_σ'(θ^n)}_{σ'∈S_K}        ▷ sample the permutation σ with probability w_σ(θ^n)
        Sample θ^(n+1) ~ p_σ(θ^(n), ·)          ▷ Sample state with the swapped Markov kernel
    end for
    Output {θ^(n)}_{n=1}^N, {{w_σ'(θ^n)}_{σ'∈S_K}}_{n=1}^N.
end function
```

Just as in Remark 3.7, one only needs to evaluate the posterior K times (instead of $|S_K|$) to compute $w_{(\cdot)}(\boldsymbol{\theta}^n)$.

4. Convergence theory of Generalized Parallel Tempering.

4.1. *Preliminaries.* In this section, we briefly review some of the concepts related to the convergence of MCMC chains, which in turn will be used to prove some of the desirable theoretical properties of both Weighted and Unweighted GPT algorithms. We rely heavily on the theory developed in [18, 30, 32]. We assume that the chains generated by the MCMC kernels p_k, for $k = 1, \dots, K$, are aperiodic, μ_k-irreducible [2], and have invariant measure μ_k on the measurable space $(\Theta, \mathcal{B}(\Theta))$.

Let $r \in [1, \infty)$ and $\mu \in \mathcal{M}(\Theta)$ be a "reference" probability measure. On a BIP setting, this reference measure is considered to be the posterior. We define the following spaces

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

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

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

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

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. We remark that the functional space $L_r(\Theta, \mu)$ is isometrically isomorphic to the space of measures $\mathcal{M}_r(\Theta, \mu)$ [32].

A Markov operator $P : L_r(\Theta, \mu) \mapsto L_r(\Theta, \mu)$ with invariant measure μ is a bounded linear operator. Let $f \in L_r(\Theta, \mu)$. The operator norm of P is given by

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

Let $r, s \in [1, \infty]$, such that $r^{-1} + s^{-1} = 1$. If $P^* : L_s(\Theta, \mu) \mapsto L_s(\Theta, \mu)$ denotes the adjoint operator of P acting on $L_r(\Theta, \mu)$, it can be shown (see, e.g., [32]) that

$$\|P\|_{L_r(\Theta,\mu) \mapsto L_r(\Theta,\mu)} = \left\|P^*\right\|_{L_s(\Theta,\mu) \mapsto L_s(\Theta,\mu)}.$$

It is also shown in [32] that if $P : L_2(\Theta, \mu) \mapsto L_2(\Theta, \mu)$ is μ-reversible, then, P is a μ-self-adjoint operator, i.e., $P^* = P$. It is well-known (see, e.g., [32]) that any Markov operator P 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.

DEFINITION 4.1 (L_r-geometric ergodicity [30]). 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

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

A related concept to L_r-geometric ergodicity is that of L_2-spectral gap.

DEFINITION 4.2 (L_2-spectral gap [30]). 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 for any signed measure $\nu \in \mathcal{M}_2(\Theta, \mu)$ with $\nu(\Theta) = 0$, the following holds

$$\|\nu P\|_{L_2(\Theta,\mu)} \leq \beta \|\nu\|_{L_2(\Theta,\mu)} .$$

Note that this is equivalent to having $\|P\|_{L_2^0(\Theta,\mu)\mapsto L_2^0(\Theta,\mu)} \leq \beta$.

The following result follows from [32], and relates the concepts of $L_r(\Theta, \mu)$-geometric ergodicity and $L_2(\Theta, \mu)$-spectral gap.

LEMMA 4.3. *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 [32, Proposition 3.17 and Appendix A.4]. It is also shown in [32] that, in general, $\beta \leq \alpha$, with α given as in Definition 4.1. □

We remark that some of the most widely used Metropolis-Hastings type algorithms, such as independent Metropolis, random Walk Metropolis and preconditioned Crank-Nicolson, among others, are known to be both reversible and to have an L_2-spectral gap under very mild conditions [18]. We will make use of these concepts when discussing the theoretical properties of the GPT procedures in the following subsection. In particular, we will show that under some mild assumptions on each of the K Markov transition kernels p_k, $k = 1, \ldots K$, the chains generated by both the Weighted and Unweighted GPT algorithms are (i) reversible with respect to either $\boldsymbol{\mu}$ (for Unweighted GPT) or $\boldsymbol{\mu}_{\mathrm{W}}$ (for Weighted GPT), (ii) their corresponding Markov operators have an L_2-spectral gap, and as such (iii) they are L_r-geometrically ergodic for $r \in [1, \infty]$.

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

DEFINITION 4.4 (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

$$\eta_{\nu_\Theta}(\pi_k, \pi_j) = \int_\Theta \min\{\pi_k(\theta), \pi_j(\theta)\}\nu_\Theta(\mathrm{d}\theta).$$

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

ASSUMPTION 4.5. For $k = 1, \ldots, K$, let $\mu_k \in \mathcal{M}_1(\Theta, \mu_{\text{prior}})$ be given as in (5), $p_k : \Theta \times \mathcal{B}(\Theta) \mapsto [0, 1]$ be the Markov kernel associated to the ith 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 Equation (6). Throughout this work it is assumed that:

C1. The Markov kernel p_k is μ_k-reversible.
C2. The Markov operator P_k has an $L_2(\Theta, \mu_k)$-spectral gap.

C3. For any $\sigma, \rho \in S_K$, $\Lambda_{\sigma,\rho} := \eta_{\boldsymbol{\mu}_{\text{prior}}}(\boldsymbol{\pi}_\sigma, \boldsymbol{\pi}_\rho) > 0$, with $\boldsymbol{\pi}_\sigma, \boldsymbol{\pi}_\rho$ defined as in (11).

We now proceed to state the main result of this section. Assumption C3 holds true given the construction of the product measures in Section 3.

THEOREM 4.6 (Main theoretical result). *Suppose that Assumption 4.5 holds, let $\boldsymbol{\mu}, \boldsymbol{\mu}_{\mathrm{W}} \in \mathcal{M}(\Theta^K)$ be the measures defined in (6) and (13), and denote by $\mathbf{P}^{(\mathrm{UW})} : L_2(\Theta^K, \boldsymbol{\mu}) \mapsto L_2(\Theta^K, \boldsymbol{\mu})$ and $\mathbf{P}^{(\mathrm{W})} : L_2(\Theta^K, \boldsymbol{\mu}_{\mathrm{W}}) \mapsto L_2(\Theta^K, \boldsymbol{\mu}_{\mathrm{W}})$ the Markov operators associated to the Unweighted and Weighted GPT algorithms, respectively. Then:*

(i) $\mathbf{P}^{(\mathrm{UW})}$ *is $\boldsymbol{\mu}$-reversible and has an $L_2(\Theta^K, \boldsymbol{\mu})$-spectral gap.*
(ii) $\mathbf{P}^{(\mathrm{W})}$ *is $\boldsymbol{\mu}_{\mathrm{W}}$-reversible and has an $L_2(\Theta^K, \boldsymbol{\mu}_{\mathrm{W}})$-spectral gap.*

The following corollary is a direct consequence of Theorem 4.6 and Proposition 4.3.

COROLLARY 4.7. Under the same assumptions as in *Theorem 4.6*, the Markov kernels $\mathbf{p}^{(UW)} : \Theta^K \times \mathcal{B}^K \mapsto [0,1]$, and $\mathbf{p}^{(W)} : \Theta^K \times \mathcal{B}^K \mapsto [0,1]$, associated with the Unweighted and Weighted GPT algorithms, are $L_r(\Theta^K, \boldsymbol{\mu})$-geometrically ergodic and $L_r(\Theta^K, \boldsymbol{\mu}_{\mathrm{W}})$-geometrically ergodic for any $r \in [1, \infty]$.

The proof of Theorem 4.6 is decomposed in several propositions and lemmata. We begin by studying reversibility.

PROPOSITION 4.8. Suppose Assumption C1 holds. Then, $\mathbf{p} = p_1 \times \cdots \times p_K$ (resp. $\mathbf{p}_\sigma = p_{\sigma(1)} \cdots \times p_{\sigma(k)}$) is reversible with respect to $\boldsymbol{\mu}$ (resp. $\boldsymbol{\mu}_\sigma$).

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

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

Showing that the previous equality holds for sets A, B that tensorize is indeed sufficient to show that the claim holds for any $A, B \in \mathcal{B}^K$. This follows from Carathéodory's Extension Theorem applied as in the proof of uniqueness of product measures; see [1, §1.3.10, 2.6.3], for details. □

LEMMA 4.9 (Reversibility of the Generalized Parallel Tempering chain). *Under Assumption C1, the Markov chains generated by $\mathbf{p}^{(\mathrm{UW})}$ and $\mathbf{p}^{(\mathrm{W})}$ are reversible with respect to $\boldsymbol{\mu}$ and $\boldsymbol{\mu}_{\mathrm{W}}$, respectively.*

PROOF. We begin with the Unweighted GPT algorithm. Since $\mathbf{p}^{(\mathrm{UW})}$ is a palindromic composition of reversible kernels (with respect to the same measure $\boldsymbol{\mu}$), i.e., $\mathbf{p}^{(\mathrm{UW})} = \mathbf{q}^{(\mathrm{UW})}\mathbf{p}\mathbf{q}^{(\mathrm{UW})}$, reversibility follows from [7, chapter 1.12.7]. For the Weighted case, 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:

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

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

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

$$= \frac{1}{|S_K|}\sum_{\sigma\in S_K}\int_B \boldsymbol{\pi}_\sigma(\boldsymbol{\theta})\mathbf{p}_\sigma(\boldsymbol{\theta}, A)\boldsymbol{\mu}_{\mathrm{prior}}(\mathrm{d}\boldsymbol{\theta}) \quad \text{(by Proposition 4.8)}$$

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

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

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

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

Since reversibility with respect to a measure implies that the Markov kernel is invariant with respect to such measure, the previous result shows that both GPT algorithms considered herein sample from the desired measures, $\boldsymbol{\mu}$ and $\boldsymbol{\mu}_{\mathrm{W}}$, for the Unweighted and the Weighted GPT, respectively.

Next, we focus on studying the ergodicity of the samplers. We begin with an auxiliary result that we will use to bound the convergence of both the Weighted and Unweighted GPT algorithms.

LEMMA 4.10. *Suppose that Assumption 4.5 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* $L_r(\Theta^K, \boldsymbol{\mu})$*-geometrically ergodic, for any* $r \in [1, \infty]$.

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

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

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

Proceeding similarly, we can obtain an equivalent bound for $\left\|(I \otimes P_2) f_0\right\|^2_{L_2(\Theta^2, \boldsymbol{\mu})}$. We are now ready to bound $\|\mathbf{P}\|^2_{L_2(\Theta^2, \boldsymbol{\mu}) \mapsto L_2(\Theta^2, \boldsymbol{\mu})}$ as

$$
\begin{aligned}
&\|\mathbf{P}\|^2_{L_2(\Theta^2, \boldsymbol{\mu}) \mapsto L_2(\Theta^2, \boldsymbol{\mu})} \leq \left\|(P_1 \otimes P_2) f_0\right\|^2_{L_2(\Theta^2, \boldsymbol{\mu})} = \left\|(P_1 \otimes I)(I \otimes P_2) f_0\right\|^2_{L_2(\Theta^2, \boldsymbol{\mu})} \\
&\leq \|P_1\|^2 \left\|(I \otimes P_2) f_0\right\|^2_{L_2(\Theta^2, \boldsymbol{\mu})} + (1 - \|P_1\|^2) \left(\sum_{j=1}^{\infty} \left((I \otimes P_2) \sum_{\ell,k}^{\infty} c_{\ell,k} \phi_\ell \psi_k, \phi_0 \psi_j \right)^2 \right) \\
&= \|P_1\|^2 \left\|(I \otimes P_2) f_0\right\|^2_{L_2(\Theta^2, \boldsymbol{\mu})} + (1 - \|P_1\|^2) \left(\sum_{j=1}^{\infty} \left(\sum_{k=1}^{\infty} c_{0,k} (P_2 \psi_k), \psi_j \right)^2 \right) \\
&\leq \|P_1\|^2 \left\|(I \otimes P_2) f_0\right\|^2_{L_2(\Theta^2, \boldsymbol{\mu})} + (1 - \|P_1\|^2) \left\| P_2 \left(\sum_{k=1}^{\infty} c_{0,k} \psi_k \right) \right\|^2_{L_2(\Theta, \mu_2)} \\
&\leq \|P_1\|^2 \|P_2\|^2 \|f_0\|^2_{L_2(\Theta^2, \boldsymbol{\mu})} + \|P_1\|^2 (1 - \|P_2\|^2) \left(\sum_{j=1}^{\infty} c_{j,0}^2 \right) + (1 - \|P_1\|^2) \|P_2\|^2 \left(\sum_{k=1}^{\infty} c_{0,k}^2 \right)
\end{aligned}
$$

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

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

Thus, we have that $\|\boldsymbol{P}\|_{L_2^0(\Theta^2,\boldsymbol{\mu})\mapsto L_2^0(\Theta^2,\boldsymbol{\mu})} \leq \max_{k=1,2}\{\|P_k\|_{L_2^0(\Theta,\mu_k)\mapsto L_2^0(\Theta,\mu_k)}\} < 1$. 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 Lemma 4.3. □

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

LEMMA 4.11 (Convergence of UGPT). *Suppose* Assumption 4.5 *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

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

where the previous line follows from Proposition 4.10 and the fact that $\mathbf{Q}^{(\mathrm{UW})}$ is a weak contraction in $L_2(\Theta^K, \boldsymbol{\mu})$ (see, e.g., [3, Lemma 1]). Lastly, $L_r(\Theta^K, \boldsymbol{\mu})$-geometric ergodicity $\forall r \in [1, \infty]$ follows from Lemma 4.3. □

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 $f \in L_2(\Theta^K, \boldsymbol{\mu}_{\mathrm{W}})$.

LEMMA 4.12. *Let* $f \in L_2^0(\Theta^K, \boldsymbol{\mu}_{\mathrm{W}})$ *be a* $\boldsymbol{\mu}_{\mathrm{W}}$*-integrable function such that* $\|f\|_{L_2^0(\Theta^K,\boldsymbol{\mu}_{\mathrm{W}})} = 1$, *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* 4.5 *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. See Appendix A.2. □

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

LEMMA 4.13 (Convergence of WGPT). *Suppose Assumption 4.5 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 $f : \Theta^K \mapsto \mathbb{R}$ be an $L_2(\Theta^K, \boldsymbol{\mu}_{\mathrm{W}})$-integrable function with $\boldsymbol{\mu}_{\mathrm{W}}(f) = 0$. Moreover, let $\mathcal{L} := \{f \in L_2^0(\Theta^K, \boldsymbol{\mu}_{\mathrm{W}}) : \|f\|_{L_2^0(\Theta^K,\boldsymbol{\mu}_{\mathrm{W}})} = 1\}$. Then, from the definition of operator norm,

$$\left\|\mathbf{P}^{(\mathrm{w})}\right\|^2_{L_2^0(\Theta^K,\boldsymbol{\mu}_{\mathbf{W}})\mapsto L_2^0(\Theta^K,\boldsymbol{\mu}_{\mathbf{W}})} = \sup_{f\in\mathcal{L}} \left\|\mathbf{P}^{(\mathrm{w})} f\right\|^2_{L_2(\Theta^K,\boldsymbol{\mu}_{\mathbf{W}})}$$

$$= \sup_{f\in\mathcal{L}} \int_{\Theta^K} \left| \sum_{\sigma\in S_K} w_\sigma(\boldsymbol{\theta}) \int_{\Theta^K} f(\boldsymbol{y}) \mathbf{p}_\sigma(\boldsymbol{\theta}, \mathrm{d}\boldsymbol{y}) \right|^2 \boldsymbol{\mu}_{\mathrm{W}}(\mathrm{d}\boldsymbol{\theta})$$

$$\leq \sup_{f\in\mathcal{L}} \int_{\Theta^K} \sum_{\sigma\in S_K} w_\sigma(\boldsymbol{\theta}) \left| \int_{\Theta^K} f(\boldsymbol{y}) \mathbf{p}_\sigma(\boldsymbol{\theta}, \mathrm{d}\boldsymbol{y}) \right|^2 \boldsymbol{\mu}_{\mathrm{W}}(\mathrm{d}\boldsymbol{\theta}) \quad \text{(from convexity of } (\cdot)^2\text{)}$$

(15)

$$= \sup_{f\in\mathcal{L}} \frac{1}{|S_K|} \sum_{\sigma\in S_K} \int_{\Theta^K} \left| \int_{\Theta^K} f(\boldsymbol{y}) \mathbf{p}_\sigma(\boldsymbol{\theta}, \mathrm{d}\boldsymbol{y}) \right|^2 \boldsymbol{\mu}_\sigma(\mathrm{d}\boldsymbol{\theta}) \quad \text{(from the definition of } w_\sigma \text{ and } \boldsymbol{\mu}_{\mathrm{W}}\text{).}$$

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

$$\int_{\Theta^K} \left| \int_{\Theta^K} f(\boldsymbol{y}) \mathbf{p}_\sigma(\boldsymbol{\theta}, \mathrm{d}\boldsymbol{y}) \right|^2 \boldsymbol{\mu}_\sigma(\mathrm{d}\boldsymbol{\theta})$$

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

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

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

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

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

$$\leq \|\mathbf{P}_\sigma\|^2_{L^0_2(\Theta^K, \boldsymbol{\mu}_\sigma) \mapsto L^0_2(\Theta^K, \boldsymbol{\mu}_\sigma)} \left(\int_{\Theta^K} \left(f(\boldsymbol{\theta}) - \bar{f}_\sigma \right)^2 \boldsymbol{\mu}_\sigma(\mathrm{d}\boldsymbol{\theta}) \right) + (\bar{f}_\sigma)^2$$

$$= \|\mathbf{P}_\sigma\|^2_{L^0_2(\Theta^K, \boldsymbol{\mu}_\sigma) \mapsto L^0_2(\Theta^K, \boldsymbol{\mu}_\sigma)} \left(\int_{\Theta^K} f(\boldsymbol{\theta})^2 \boldsymbol{\mu}_\sigma(\mathrm{d}\boldsymbol{\theta}) \right) + \left(1 - \|\mathbf{P}_\sigma\|^2_{L^0_2(\Theta^K, \boldsymbol{\mu}_\sigma) \mapsto L^0_2(\Theta^K, \boldsymbol{\mu}_\sigma)} \right) (\bar{f}_\sigma)^2$$

(16)

$$= \left(\int_{\Theta^K} f(\boldsymbol{\theta})^2 \boldsymbol{\mu}_\sigma(\mathrm{d}\boldsymbol{\theta}) \right) - \underbrace{\left(1 - \|\mathbf{P}_\sigma\|^2_{L^0_2(\Theta^K, \boldsymbol{\mu}_\sigma) \mapsto L^0_2(\Theta^K, \boldsymbol{\mu}_\sigma)} \right)}_{:= \gamma, \text{ with } \gamma \in (0,1)} \left(\int_{\Theta^K} \left(f(\boldsymbol{\theta}) - \bar{f}_\sigma \right)^2 \boldsymbol{\mu}_\sigma(\mathrm{d}\boldsymbol{\theta}) \right).$$

Replacing Equation (16) into Equation (15), we get

$$\left\| \mathbf{P}^{(\mathrm{W})} \right\|^2_{L^0_2(\Theta^K, \boldsymbol{\mu}_{\mathrm{W}}) \mapsto L^0_2(\Theta^K, \boldsymbol{\mu}_{\mathrm{W}})} \leq \sup_{\substack{f \in L^0_2(\Theta^K, \boldsymbol{\mu}_{\mathrm{W}}) \\ \|f\|_{L^0_2(\Theta^K, \boldsymbol{\mu}_{\mathrm{W}})=1}}} \left(\int_{\Theta^K} f(\boldsymbol{\theta})^2 \boldsymbol{\mu}_{\mathrm{W}}(\mathrm{d}\boldsymbol{\theta}) \right) - \gamma \mathbb{V}_{\boldsymbol{\mu}_\sigma}[f]$$

$$\leq 1 - \gamma \left(\frac{\Lambda_m}{2 - \Lambda_m} \right) < 1 \quad \text{(by Lemma 4.12).}$$

Thus, $\mathbf{P}^{(\mathrm{w})}$ 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 Lemma 4.3.

□

The proof of Theorem 4.6 then follows immediately from Lemmata 4.9, 4.11, 4.13. We remark that, we have not used temperature information in our estimates, and as such, we believe that our estimates can thus be improved. These potential improvements will be the focus of a future work. Furthermore, we remark that the framework presented herein is, in principle, dimension independent, and as such, can be applied to infinite-dimensional BIP [33], provided proper Markov kernels are used on each chain (as for instance, those discussed on [18]). Extending the results of the current work to infinite dimensional BIP will also be the subject of a future work.

5. Numerical experiments. We now present two *academic* examples to illustrate the efficiency of both GPT algorithms discussed herein and compare them to the more traditional random walk Metropolis and PT algorithms. Notice that we compare the examples with respect to the "simplest" version of these methods, since more efficient variations, such as Adaptive Metropolis [17, 16], for example, can also be extended into the GPT framework. The following experiments were run in a Dell (R) Precision (TM) T3620 workstation with Intel(R) Core(TM) i7-7700 CPU with 32 GB of RAM. Numerical simulations in Section 5.1 were run on a single thread, while the numerical simulations in Section 5.2 were run on an *embarrassingly parallel* fashion over 8 threads using the Message Passing Interface (MPI) and the Python package MPI4py [10]. The scripts used to generate the results presented in this section were written in Python 3.6, and can be found in `DOI:10.5281/zenodo.3700049`.

REMARK 5.1. In most Bayesian inverse problems, particularly those dealing with large-scale computational models, the computational cost is dominated by the evaluation of the forward operator, which can be, for example, the numerical approximation of a possibly non-linear partial differential equation. In the case where all possible permutations are considered (i.e., $S_K = \boldsymbol{S}_K$), 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. For higher values of K, it is advisable to only consider the union of properly chosen semi-groups A, B of $\boldsymbol{S}_K$, with $A \cap B \neq \emptyset$, such that A, B generates $\boldsymbol{S}_K$ (i.e., if the smallest semi-groups that contains A and B is $\boldsymbol{S}_K$ itself), and $|A \cup B| < |\boldsymbol{S}_K| = K!$, which is referred to as partial Infinite Swapping in the continuous case [14]. 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 [14, 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.

5.1. *Density concentrated over a quarter circle-shaped manifold.* Let μ be a probability measure that has density π with respect to the uniform Lebesgue measure on the unit square $\mu_{\text{prior}} = \mathcal{U}([0,1]^2)$ given by

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

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

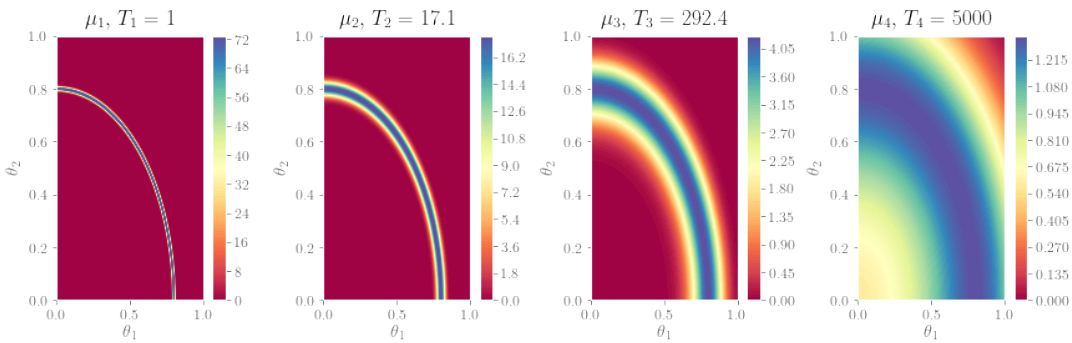

FIG 1. *Tempered densities (with $T_1 = 1,\ T_2 = 17.1,\ T_3 = 292.4,\ T_4 = 5000$) for the density concentrated around a quarter circle-shaped manifold example. As we can see, the density becomes less concentrated as the temperature increases, which allows us to use RWM proposals with larger step sizes.*

We aim at estimating $\mathcal{Q}_k = \mathbb{E}_{\mu_1}[\theta_k] \approx \hat{\theta}_k$, for $k = 1, 2$. To do so, we implement four MCMC algorithms to sample from μ_1, namely Random Walk Metropolis (RWM), the canonical PT (PT) with $N_s = 1$, as described in Section 3.2, and both versions of the GPT algorithm. We compare the quality of our algorithms by examining the variance of the estimators $\hat{\theta}_k,\ k = 1, 2$ computed over 100 independent MCMC runs of each algorithm, which we describe as follows. For the tempered algorithms (PT, UGPT, and WGPT), we consider $K = 4$ temperatures. A *rule of thumb* [15] for the choice of temperatures is to set $T_i = a^{i-1}$, $k = 1, \ldots, K$, for some positive constant $a > 1$. In particular, we choose $T_4 = 5000$, so that the tempered density π_4 becomes a sufficiently simple to explore target distribution. This gives $T_1 = 1, T_2 = 17.1, T_3 = 292.4, T_4 = 5000$. Moreover, for both GPT algorithms, we set $S_K = \bar{S}_K$, i.e., we consider all possible $K!$ permutations of $\{1, 2, \ldots, K\}$. Notice that since this is a relatively small value of K, the computational time is dominated by the transition operator $\mathbf{P}$, rather than by the computation of the swapping ratio. In the current setting, the computational cost of PT is comparable to that of both GPT algorithms discussed herein. Each estimator is obtained by running the inversion experiment for $N = 25,000$ samples, discarding the first 20% of the samples (5000) as a burn-in. Notice that the tempering algorithms (i.e., PT, UGPT and WGPT) have a K-times larger computational cost than RWM, since such algorithms need to run a total of K chains. To account for this computational cost, we run the single-chain random walk Metropolis algorithm for $N_{\text{RWM}} = KN = 100,000$ iterations, and discard the first 20% of the samples obtained with the RWM algorithm (20,000) as a burn-in.

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

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

TABLE 1

Step size of the RWM proposal distribution for the manifold experiment. This choice of step size provides an average acceptance rate for each chain, at each temperature, of around 0.23 for all the algorithms tested. Such values are relatively close to the "optimal" value of 0.234 in [31].

Experimental results for the ergodic run are shown in Table 5.1. We can see how both GPT algorithms provide a gain over both RWM and the (standard) PT algorithms, with the WGPT algorithm providing a larger gain. Scatter plots of the samples obtained with each method are presented in Figure 2. Here, the subplot titled "WGPT" (second from right to left) corresponds to weighted samples from $\boldsymbol{\mu}_{\mathrm{W}}$, with weight $\hat{w}$ as in (14), while the one titled "WGPT (inv)" (rightmost) corresponds to samples from $\boldsymbol{\mu}_{\mathrm{W}}$ without any post-processing. Notice how the samples from the latter concentrates over a *wider* manifold, which in turn makes the target density easier to explore when using state-dependent Markov transition kernels.

	Mean		MSE		$\mathrm{MSE}_{\mathrm{RWM}}/\mathrm{MSE}$	
	θ_1	θ_2	θ_1	θ_2	θ_1	θ_2
RWM	0.50996	0.50657	0.002521	0.00236	1.00	1.00
PT	0.50978	0.51241	0.000460	0.00051	5.50	4.70
UGPT	0.50986	0.50987	0.000370	0.00035	6.80	6.70
WGPT	0.51062	0.50838	0.000220	0.00023	11.5	10.2

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 and RWM. The computational cost is comparable across all algorithms.

5.2. *Multiple source elliptic BIP.* We now consider a slightly more challenging problem, for which we try to recover the probability distribution of the location of a source term in a Poisson equation (Eq. (17)), based on some noisy measured data. Let $(\Theta, \mathcal{B}(\Theta), \mu_{\mathrm{prior}})$ be the measure space, set $\Theta = \bar{D} := [0,1]^2$, with Lebesgue (uniform) measure μ_{prior}, and consider the following Poisson's equation with homogeneous boundary conditions:

$$\begin{cases} \Delta u(x,\theta) = f(x,\theta), & x \in D,\ \theta \in \Theta, \\ u(x,\theta) = 0, & x \in \partial D. \end{cases} \tag{17}$$

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

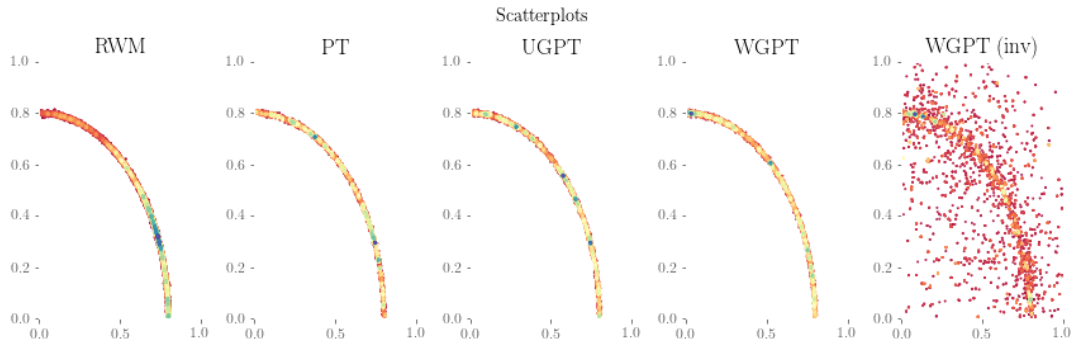

FIG 2. *Scatter-plots of the samples from μ_1 obtained with each algorithm. From left to right: random walk Metropolis, PT, UGPT, WGPT (after re-weighting the samples), and WGPT, before re-weighting the samples.*

recorded on an array of 64×64 equally-spaced points in D by solving (17) with a forcing term given by

$$f(x) = \sum_{i=1}^{4} e^{-1000[(x_1 - s_1^{(i)})^2 + (x_2 - s_2^{(i)})^2]}, \tag{18}$$

where the true source locations $s^{(i)}$, $i = 1, 2, 3, 4$, are given by $s^{(1)} = (0.2, 0.2)$, $s^{(2)} = (0.2, 0.8)$, $s^{(3)} = (0.8, 0.2)$, and $s^{(4)} = (0.8, 0.8)$. Such data is assumed to be polluted by an additive Gaussian noise $\eta \sim \mathcal{N}(0, \sigma^2 I_{64\times 64})$, with $\sigma = 3.2 \times 10^{-6}$, (which corresponds to a 1% noise) and where $I_{64\times 64}$ is the 64-dimensional identity matrix. Thus, we set $(Y, \|\cdot\|_Y) = (\mathbb{R}^{64\times 64}, \|\cdot\|_\Sigma)$, with $\|A\|_\Sigma = (64\sigma)^{-2} \left\| A^T A \right\|_F$, for some arbitrary matrix $A \in \mathbb{R}^{64\times 64}$, where $\|\cdot\|_F$ is the Frobenius norm. We assume a misspecified model where we only consider a single source in Eq. (18). That, is, we construct our forward operator $\mathcal{F} : \Theta \mapsto Y$ by solving (17) with a source term given by

$$f(x, \theta) = e^{-1000[(x_1 - \theta_1)^2 + (x_2 - \theta_2)^2]}. \tag{19}$$

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 by sampling from the posterior measure μ, which has a density with respect to the prior $\mu_{\text{prior}} = \mathcal{U}(\bar{D})$ given by

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

for some (intractable) normalization constant Z as in (4). We remark that the solution to (17) with a forcing term of the form of (19) is approximated using a second-order accurate finite difference approximation with grid-size $h = 1/64$ on each spatial component.

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

We follow a similar experimental setup as in the previous example, by implementing RWM, PT (with $N_s = 1$), and both versions of the GPT algorithms. For the PT and GPT algorithms, four different temperatures are used, with $T_1 = 1$, $T_2 = 7.36$, $T_3 = 54.28$, and $T_4 = 400$. Once again, we set $S_K = \bar{S}_K = 4!$ for both GPT algorithms. Given that 41 is a moderately small number, the computational cost of evaluating the forward model is much higher than the cost associated with computing the swapping ratio.

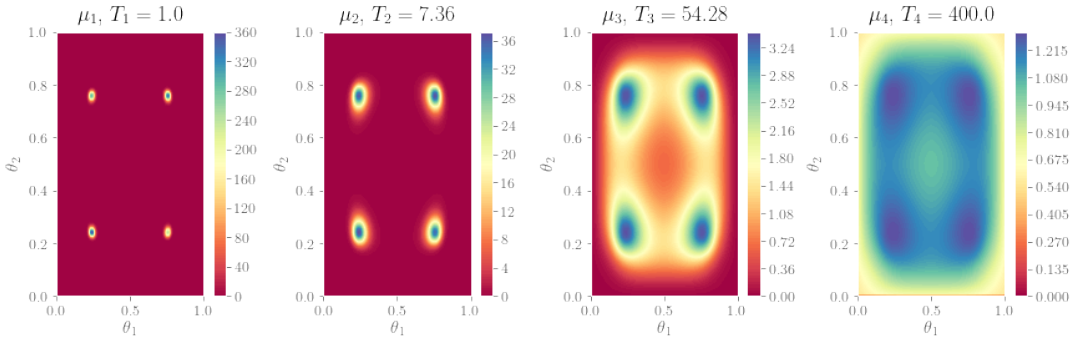

FIG 3. *True tempered densities for the elliptic BIP example. Notice that the density is not symmetric, due to the additional random noise.*

Since we have $K = 4$ temperatures, we run the RWM algorithm for K-times longer, so that the computational cost of all algorithms tested is comparable. For each run, we obtain $N = 25,000$ samples with the PT and 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.

For the tempered algorithms, we run each simulation for a total of $N = 25,000$ samples, and a total of $100,000$ samples with RWM. We discard the first $5,000$ the samples for the PT and GPT algorithms and the first $20,000$ for the RWM algorithm as a burn-in. On each of the tempered chains, we use RWM proposals, with step-sizes shown in table 3. This choice of step size provides an acceptance rate of about 0.24 across all tempered chains and all tempered algorithms. For the single-temperature RWM run, we choose a larger step size ($\sigma_{\text{RWM}} = 0.16$) so that the RWM algorithm is able to explore the whole distribution. Such a choice, however, provides a smaller acceptance rate of about 0.01 for the single-chain RWM.

Experimental results are shown in Table 5.1. Once again, we can see how both GPT algorithms provide a gain over both RWM and the PT algorithms, with the WGPT algorithm providing a larger gain. Scatter-plots of the obtained samples are shown in Figure 3.

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

TABLE 3

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

6. Conclusions and future work. In the current work, we have proposed, implemented, and analyzed two versions of the GPT, and applied these methods to a BIP context. We demonstrate that such algorithms produce reversible and geometrically-ergodic chains under relatively mild conditions. As shown in Section 5, such sampling algorithms provide an attractive alternative to the more standard Parallel Tempering when sampling from *difficult* (i.e., multi-modal or concentrated around a manifold) posteriors. We remark that the framework considered here-in can be combined with other, more advanced MCMC algorithms, such as, e.g., the Metropolis-adjusted Langevin algorithm (MALA), or the Delayed Rejection Adaptive Metropolis (DRAM), for example [16].

	Mean		MSE		$\text{MSE}_{\text{RWM}}/\text{MSE}$	
	θ_1	θ_2	θ_1	θ_2	θ_1	θ_2
RWM	0.41143	0.52954	0.01099	0.01270	1.00	1.00
PT	0.39262	0.53690	0.00062	0.00089	17.7	14.2
UGPT	0.39169	0.53338	0.00050	0.00079	21.9	12.8
WGPT	0.39345	0.53074	0.00048	0.00077	22.9	16.5

TABLE 4

Results for the elliptic BIP problem. Once again, we can see that both GPT provide an improvement over RWM and PT. The computational cost GPT comparable across all algorithms.

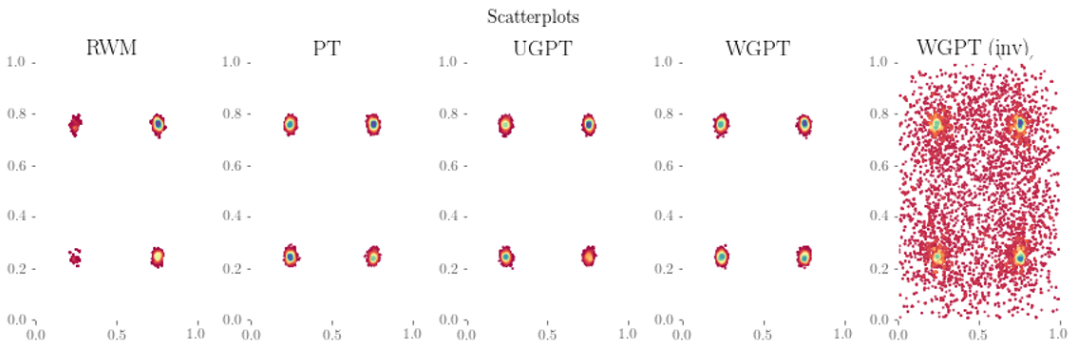

FIG 4. *Scatterplots of the samples from μ_1 obtained with different algorithms on a single run. From left to right: random walk Metropolis, PT, UGPT, WGPT (after re-weighting the samples), and WGPT, before re-weighting the samples. As we can see, WGPT is able to "connect" the parameter space.*

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 [12]. 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 [36]. Improving on the estimates presented here would likely be the focus of future work. 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 [4, 9], 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.

APPENDIX: AUXILIARY RESULTS

A.1. Proof Proposition 3.4.

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

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

Thus,

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

Let $A_\sigma := \{\boldsymbol{z}\in X^K : \boldsymbol{z}_{\sigma^{-1}}\in A\}$. From I, we get

$$\begin{aligned}
I &= \sum_{\sigma\in S_K}\int_A \min\left\{1, \frac{\boldsymbol{\pi}(\boldsymbol{\theta}_\sigma)r(\boldsymbol{\theta}_\sigma,\sigma^{-1})}{\boldsymbol{\pi}(\boldsymbol{\theta})r(\boldsymbol{\theta},\sigma)}\right\} r(\boldsymbol{\theta},\sigma)\boldsymbol{\pi}(\boldsymbol{\theta})\delta_{\boldsymbol{\theta}_\sigma}(B)\boldsymbol{\mu}_{\text{prior}}(\mathrm{d}\boldsymbol{\theta})\\
&= \sum_{\sigma\in S_K}\int_A \min\left\{1, \frac{\boldsymbol{\pi}(\boldsymbol{\theta})r(\boldsymbol{\theta},\sigma)}{\boldsymbol{\pi}(\boldsymbol{\theta}_\sigma)r(\boldsymbol{\theta}_\sigma,\sigma^{-1})}\right\} r(\boldsymbol{\theta}_\sigma,\sigma^{-1})\boldsymbol{\pi}(\boldsymbol{\theta}_\sigma)\delta_{\boldsymbol{\theta}_\sigma}(B)\boldsymbol{\mu}_{\text{prior}}(\mathrm{d}\boldsymbol{\theta})\\
&= \sum_{\sigma\in S_K}\int_{A_\sigma} \min\left\{1, \frac{\boldsymbol{\pi}(\boldsymbol{\theta}_{\sigma^{-1}})r(\boldsymbol{\theta}_{\sigma^{-1}},\sigma)}{\boldsymbol{\pi}(\boldsymbol{\theta})r(\boldsymbol{\theta},\sigma^{-1})}\right\} r(\boldsymbol{\theta},\sigma^{-1})\boldsymbol{\pi}(\boldsymbol{\theta})\delta_{\boldsymbol{\theta}}(B)\boldsymbol{\mu}_{\text{prior}}(\mathrm{d}\boldsymbol{\theta})
\end{aligned}$$

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

For the second term II we simply have

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

□

A.2. Proof of Lemma 4.12.

PROOF. This proof is partially based on the proof of Theorem 1.2 in [25]. 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

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

$$\geq \frac{1}{|S_K|} \sum_{\sigma \in S_K} \mathbb{V}_{\boldsymbol{\mu}_\sigma}[f]$$

We follow a procedure similar to the proof of [25, 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

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

$$= \sum_{i,j} \frac{1}{|S_K|^2} C_{ij}. \tag{20}$$

We thus aim at finding an upper bound of Equation (20) 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 $\boldsymbol{\mu}^0$) have an overlap $\Lambda_{ij} > 0$. Thus, we can find densities $\boldsymbol{\eta}_{ij} := \Lambda_{ij}^{-1} \min_{\boldsymbol{\theta} \in \Theta^K} \{\boldsymbol{\pi}_{\sigma_i}(\boldsymbol{\theta}), \boldsymbol{\pi}_{\sigma_j}(\boldsymbol{\theta})\}, \boldsymbol{\varphi}_i, \boldsymbol{\psi}_j$ such that $\boldsymbol{\pi}_{\sigma_i} = \Lambda_{ij}\boldsymbol{\eta}_{ij} + (1 - \Lambda_{ij})\boldsymbol{\varphi}_i$, and $\boldsymbol{\pi}_{\sigma_j} = \Lambda_{ij}\boldsymbol{\eta}_{ij} + (1 - \Lambda_{ij})\boldsymbol{\psi}_j$. Thus, we get for the diagonal entries of the C matrix:

$$\begin{aligned}
C_{ii} &= 2\mathbb{V}\boldsymbol{\mu}_{\sigma_i}[f] \\
&= \int_{\Theta^K} \int_{\Theta^K} (f(\boldsymbol{\theta}) - f(\boldsymbol{y}))^2 \left(\Lambda_{ij}\boldsymbol{\eta}_{ij}(\boldsymbol{\theta}) + (1 - \Lambda_{ij})\boldsymbol{\varphi}_i(\boldsymbol{\theta})\right) \left(\Lambda_{ij}\boldsymbol{\eta}_{ij}(\boldsymbol{y}) + (1 - \Lambda_{ij})\boldsymbol{\varphi}_i(\boldsymbol{y})\right) \boldsymbol{\mu}^0(\mathrm{d}\boldsymbol{\theta})\boldsymbol{\mu}^0(\mathrm{d}\boldsymbol{y}) \\
&= \int_{\Theta^K} \int_{\Theta^K} (f(\boldsymbol{\theta}) - f(\boldsymbol{y}))^2 \Lambda_{ij}^2 \boldsymbol{\eta}_{ij}(\boldsymbol{\theta})\boldsymbol{\eta}_{ij}(\boldsymbol{y})\boldsymbol{\mu}^0(\mathrm{d}\boldsymbol{\theta})\boldsymbol{\mu}^0(\mathrm{d}\boldsymbol{y}) \\
&+ \int_{\Theta^K} \int_{\Theta^K} (f(\boldsymbol{\theta}) - f(\boldsymbol{y}))^2 \Lambda_{ij}(1 - \Lambda_{ij})\boldsymbol{\varphi}_i(\boldsymbol{y})\boldsymbol{\eta}_{ij}(\boldsymbol{\theta})\boldsymbol{\mu}^0(\mathrm{d}\boldsymbol{\theta})\boldsymbol{\mu}^0(\mathrm{d}\boldsymbol{y}) \\
&+ \int_{\Theta^K} \int_{\Theta^K} (f(\boldsymbol{\theta}) - f(\boldsymbol{y}))^2 \Lambda_{ij}(1 - \Lambda_{ij})\boldsymbol{\varphi}_i(\boldsymbol{\theta})\boldsymbol{\eta}_{ij}(\boldsymbol{y})\boldsymbol{\mu}^0(\mathrm{d}\boldsymbol{\theta})\boldsymbol{\mu}^0(\mathrm{d}\boldsymbol{y}) \\
&+ \int_{\Theta^K} \int_{\Theta^K} (f(\boldsymbol{\theta}) - f(\boldsymbol{y}))^2 (1 - \Lambda_{ij})^2 \boldsymbol{\varphi}_i(\boldsymbol{y})\boldsymbol{\varphi}_i(\boldsymbol{\theta})\boldsymbol{\mu}^0(\mathrm{d}\boldsymbol{\theta})\boldsymbol{\mu}^0(\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}) \int_{\Theta^K} \int_{\Theta^K} (f(\boldsymbol{\theta}) - f(\boldsymbol{y}))^2 \boldsymbol{\eta}_{ij}(\boldsymbol{\theta})\boldsymbol{\varphi}_i(\boldsymbol{\theta})\boldsymbol{\mu}^0(\mathrm{d}\boldsymbol{\theta})\boldsymbol{\mu}^0(\mathrm{d}\boldsymbol{y}).
\end{aligned} \tag{21}$$

Notice that equation (21) implies that

$$\int_{\Theta^K} \int_{\Theta^K} (f(\boldsymbol{\theta}) - f(\boldsymbol{y}))^2 \boldsymbol{\eta}_{ij}(\boldsymbol{\theta})\boldsymbol{\varphi}_i(\boldsymbol{\theta})\boldsymbol{\mu}^0(\mathrm{d}\boldsymbol{\theta})\boldsymbol{\mu}^0(\mathrm{d}\boldsymbol{y}) \leq \frac{\mathbb{V}_{\boldsymbol{\mu}_{\sigma_i}}[f] - \Lambda_{ij}^2 \mathbb{V}_{\boldsymbol{\eta}_{ij}}[f]}{\Lambda_{ij}(1 - \Lambda_{ij})}. \tag{22}$$

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

$$\begin{aligned}
C_{ij} &= \int_{\Theta^K} \int_{\Theta^K} (f(\boldsymbol{\theta}) - f(\boldsymbol{y}))^2 \left[\Lambda_{ij}\boldsymbol{\eta}_{ij}(\boldsymbol{\theta})\right. \\
&\quad \left. + (1 - \Lambda_{ij})\boldsymbol{\varphi}_i(\boldsymbol{\theta})\right] \left(\Lambda_{ij}\boldsymbol{\eta}_{ij}(\boldsymbol{y}) + (1 - \Lambda_{ij})\boldsymbol{\psi}_j(\boldsymbol{y})\right) \boldsymbol{\mu}^0(\mathrm{d}\boldsymbol{\theta})\boldsymbol{\mu}^0(\mathrm{d}\boldsymbol{y})
\end{aligned} \tag{23}$$

$$= 2\Lambda_{ij}^2 \mathbb{V}_{\boldsymbol{\eta}_{ij}}[f] + (1-\Lambda_{ij})^2 \int_{\Theta^K}\int_{\Theta^K} (f(\boldsymbol{\theta}) - f(\boldsymbol{y}))^2 \boldsymbol{\varphi}_i(\boldsymbol{\theta})\boldsymbol{\psi}_j(\boldsymbol{y})\boldsymbol{\mu}^0(\mathrm{d}\boldsymbol{\theta})\boldsymbol{\mu}^0(\mathrm{d}\boldsymbol{y})$$

$$+ \Lambda_{ij}(1-\Lambda_{ij}) \int_{\Theta^K}\int_{\Theta^K} (f(\boldsymbol{\theta}) - f(\boldsymbol{y}))^2 \left(\boldsymbol{\eta}_{ij}(\boldsymbol{\theta})\boldsymbol{\psi}_j(\boldsymbol{y}) + \boldsymbol{\eta}_{ij}(\boldsymbol{y})\boldsymbol{\varphi}_i(\boldsymbol{\theta})\right) \boldsymbol{\mu}^0(\mathrm{d}\boldsymbol{\theta})\boldsymbol{\mu}^0(\mathrm{d}\boldsymbol{y}).$$

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

$$\int_{\Theta^K}\int_{\Theta^K} (f(\boldsymbol{\theta}) - f(\boldsymbol{y}))^2 \boldsymbol{\varphi}_i(\boldsymbol{\theta})\boldsymbol{\psi}_j(\boldsymbol{y})\boldsymbol{\mu}^0(\mathrm{d}\boldsymbol{\theta})\boldsymbol{\mu}^0(\mathrm{d}\boldsymbol{y})$$

$$= \int_{\Theta^K}\int_{\Theta^K}\int_{\Theta^K} (f(\boldsymbol{\theta}) - f(\boldsymbol{z}) + f(\boldsymbol{z}) - f(\boldsymbol{y}))^2 \boldsymbol{\varphi}_i(\boldsymbol{\theta})\boldsymbol{\psi}_j(\boldsymbol{y})\boldsymbol{\eta}_{ij}(\boldsymbol{z})\boldsymbol{\mu}^0(\mathrm{d}\boldsymbol{\theta})\boldsymbol{\mu}^0(\mathrm{d}\boldsymbol{y})\boldsymbol{\mu}^0(\mathrm{d}\boldsymbol{z})$$

$$\leq 2\int_{\Theta^K}\int_{\Theta^K}\int_{\Theta^K} \left(\left(f(\boldsymbol{\theta}) - f(\boldsymbol{z})\right)^2 + (f(\boldsymbol{z}) - f(\boldsymbol{y}))^2\right) \boldsymbol{\varphi}_i(\boldsymbol{\theta})\boldsymbol{\psi}_j(\boldsymbol{y})\boldsymbol{\eta}_{ij}(\boldsymbol{z})\boldsymbol{\mu}^0(\mathrm{d}\boldsymbol{\theta})\boldsymbol{\mu}^0(\mathrm{d}\boldsymbol{y})\boldsymbol{\mu}^0(\mathrm{d}\boldsymbol{z})$$

$$= 2\int_{\Theta^K}\int_{\Theta^K} (f(\boldsymbol{\theta}) - f(\boldsymbol{z}))^2 \boldsymbol{\varphi}_i(\boldsymbol{\theta})\boldsymbol{\eta}_{ij}(\boldsymbol{z})\boldsymbol{\mu}^0(\mathrm{d}\boldsymbol{\theta})\boldsymbol{\mu}^0(\mathrm{d}\boldsymbol{z})$$

(24)

$$+ 2\int_{\Theta^K}\int_{\Theta^K} (f(\boldsymbol{y}) - f(\boldsymbol{z}))^2 \boldsymbol{\psi}_j(\boldsymbol{y})\boldsymbol{\eta}_{ij}(\boldsymbol{z})\boldsymbol{\mu}^0(\mathrm{d}\boldsymbol{y})\boldsymbol{\mu}^0(\mathrm{d}\boldsymbol{z}).$$

Thus, from equations (22), (23), and (24) we get

$$C_{ij} \leq 2\Lambda_{ij}^2 \mathbb{V}_{\boldsymbol{\eta}_{ij}}[f] + (2(1-\Lambda_{ij})^2 + \Lambda_{ij}(1-\Lambda_{ij})) \left(\int_{\Theta^K}\int_{\Theta^K} (f(\boldsymbol{\theta}) - f(\boldsymbol{z}))^2 \left(\boldsymbol{\eta}_{ij}(\boldsymbol{\theta})\boldsymbol{\psi}_j(\boldsymbol{y})\right.\right.$$

$$\left.\left. + \boldsymbol{\eta}_{ij}(\boldsymbol{y})\psi_i(\boldsymbol{\theta})\right) \boldsymbol{\mu}^0(\mathrm{d}\boldsymbol{\theta})\boldsymbol{\mu}^0(\mathrm{d}\boldsymbol{y})\right)$$

$$= 2\Lambda_{ij}^2 \mathbb{V}_{\boldsymbol{\eta}_{ij}}[f] + (2-\Lambda_{ij})(1-\Lambda_{ij}) \left(\mathbb{V}_{\boldsymbol{\mu}_{\sigma_i}}[f] - \Lambda_{ij}^2 \mathbb{V}_{\boldsymbol{\eta}_{ij}}[f] + \mathbb{V}_{\boldsymbol{\mu}_{\sigma_j}}[f] - \Lambda_{ij}^2 \mathbb{V}_{\boldsymbol{\eta}_{ij}}[f]\right) / \Lambda_{ij}(1-\Lambda_{ij})$$

$$= \frac{2-\Lambda_{ij}}{\Lambda_{ij}} \left(V_{\boldsymbol{\mu}_{\sigma_i}}[f] + V_{\boldsymbol{\mu}_{\sigma_j}}[f]\right) - 4\Lambda_{ij}(1-\Lambda_{ij}) \mathbb{V}_{\boldsymbol{\eta}_{ij}}[f]$$

(25)

$$\leq \frac{2-\Lambda_{ij}}{\Lambda_{ij}} \left(V_{\boldsymbol{\mu}_{\sigma_i}}[f] + V_{\boldsymbol{\mu}_{\sigma_j}}[f]\right),$$

since $\Lambda_{ij} \in (0,1)\ \forall i,j$. Finally, from equations (20) and (25) we get that

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

with $\Lambda_m := \min_{i,j=1,2,\ldots,|S_K|}\{\Lambda_{ij}\} > 0$, and $\Lambda_{i,j}$ as in Assumption C3. This in turn yields the lower bound

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

□

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

REFERENCES

[1] ASH, R. B. (2000). *Probability and measure theory*. Harcourt/Academic Press, Burlington, MA.

[2] ASMUSSEN, S. and GLYNN, P. W. (2007). *Stochastic simulation: algorithms and analysis* **57**. Springer Science & Business Media.

[3] BAXTER, J. R. and ROSENTHAL, J. S. (1995). Rates of convergence for everywhere-positive Markov chains. *Statistics & probability letters* **22** 333–338.

[4] BESKOS, A., GIROLAMI, M., LAN, S., FARRELL, P. E. and STUART, A. M. (2017). Geometric MCMC for infinite-dimensional inverse problems. *Journal of Computational Physics* **335** 327–351.

[5] BESKOS, A., JASRA, A., KANTAS, N. and THIERY, A. (2016). On the convergence of adaptive sequential Monte Carlo methods. *Ann. Appl. Probab.* **26** 1111–1146.

[6] BESKOS, A., JASRA, A., MUZAFFER, E. and STUART, A. M. (2015). Sequential Monte Carlo methods for Bayesian elliptic inverse problems. *Stat. Comp.* **25** 727-737.

[7] BROOKS, S., GELMAN, A., JONES, G. and MENG, X.-L. (2011). *Handbook of Markov chain Monte Carlo*. CRC press.

[8] SCHILLINGS, C. and STUART, A. M. (2017). Analysis of the ensemble Kalman filter for inverse problems. *SINUM* **55** 1264-1290.

[9] CUI, T., LAW, K. J. and MARZOUK, Y. M. (2016). Dimension-independent likelihood-informed MCMC. *Journal of Computational Physics* **304** 109–137.

[10] DALCÍN, L., PAZ, R. and STORTI, M. (2005). MPI for Python. *Journal of Parallel and Distributed Computing* **65** 1108–1115.

[11] DESJARDINS, G., COURVILLE, A., BENGIO, Y., VINCENT, P. and DELALLEAU, O. (2010). Tempered Markov chain Monte Carlo for training of restricted Boltzmann machines. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics* 145–152.

[12] DODWELL, T. J., KETELSEN, C., SCHEICHL, R. and TECKENTRUP, A. L. (2015). A hierarchical multilevel Markov chain Monte Carlo algorithm with applications to uncertainty quantification in subsurface flow. *SIAM/ASA Journal on Uncertainty Quantification* **3** 1075–1108.

[13] DOLL, J., PLATTNER, N., FREEMAN, D. L., LIU, Y. and DUPUIS, P. (2012). Rare-event sampling: Occupation-based performance measures for parallel tempering and infinite swapping Monte Carlo methods. *The Journal of chemical physics* **137** 204112.

[14] DUPUIS, P., LIU, Y., PLATTNER, N. and DOLL, J. D. (2012). On the infinite swapping limit for parallel tempering. *Multiscale Modeling & Simulation* **10** 986–1022.

[15] EARL, D. J. and DEEM, M. W. (2005). Parallel tempering: Theory, applications, and new perspectives. *Physical Chemistry Chemical Physics* **7** 3910–3916.

[16] HAARIO, H., LAINE, M., MIRA, A. and SAKSMAN, E. (2006). DRAM: efficient adaptive MCMC. *Statistics and computing* **16** 339–354.

[17] HAARIO, H., SAKSMAN, E. and TAMMINEN, J. (2001). An adaptive Metropolis algorithm. *Bernoulli* **7** 223–242.

[18] HAIRER, M., STUART, A. M., VOLLMER, S. J. et al. (2014). Spectral gaps for a Metropolis–Hastings algorithm in infinite dimensions. *The Annals of Applied Probability* **24** 2455–2490.

[19] HASTINGS, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Oxford University Press*.

[20] KAHLE, C., LAM, K., LATZ, J. and ULLMANN, E. (2019). Bayesian parameter identification in Cahn–Hilliard models for biological growth. *SIAM/ASA Journal on Uncertainty Quantification* **7** 526-552.

[21] KANTAS, N., BESKOS, A. and JASRA, A. (2014). Sequential Monte Carlo Methods for High-Dimensional Inverse Problems: A case study for the Navier–Stokes equations. *SIAM/ASA J. Uncertain. Quantif.* **2** 464–489.

[22] LATZ, J. (2019). On the well-posedness of Bayesian inverse problems. *arXiv e-prints* arXiv:1902.10257.

[23] LATZ, J., PAPAIOANNOU, I. and ULLMANN, E. (2018). Multilevel Sequential^2 Monte Carlo for Bayesian inverse problems. *Journal of Computational Physics* **368** 154 - 178.

[24] LU, J. and VANDEN-EIJNDEN, E. (2013). Infinite swapping replica exchange molecular dynamics leads to a simple simulation patch using mixture potentials. *The Journal of chemical physics* **138** 084105.

[25] MADRAS, N., RANDALL, D. et al. (2002). Markov chain decomposition for convergence rate analysis. *The Annals of Applied Probability* **12** 581–606.

[26] MARINARI, E. and PARISI, G. (1992). Simulated Tempering: A New Monte Carlo Scheme. *Europhysics Letters (EPL)* **19** 451–458.

[27] METROPOLIS, N., ROSENBLUTH, A. W., ROSENBLUTH, M. N., TELLER, A. H. and TELLER, E. (1953). Equation of state calculations by fast computing machines. *The journal of chemical physics* **21** 1087–1092.

[28] MIASOJEDOW, B., MOULINES, E. and VIHOLA, M. (2013). An adaptive parallel tempering algorithm. *Journal of Computational and Graphical Statistics* **22** 649–664.

[29] PLATTNER, N., DOLL, J., DUPUIS, P., WANG, H., LIU, Y. and GUBERNATIS, J. (2011). An infinite swapping approach to the rare-event sampling problem. *The Journal of chemical physics* **135** 134111.

[30] ROBERTS, G. O. and ROSENTHAL, J. S. (1997). Geometric ergodicity and hybrid Markov chains. *Electron. Comm. Probab* **2** 13–25.

[31] ROBERTS, G. O. and ROSENTHAL, J. S. (1998). Optimal scaling of discrete approximations to Langevin diffusions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **60** 255–268.

[32] RUDOLF, D. (2012). Explicit error bounds for Markov chain Monte Carlo. *Dissertationes Math.* **485** 1–93. MR2977521

[33] STUART, A. M. (2010). Inverse problems: a Bayesian perspective. *Acta Numerica* **19** 451–559.

[34] VAN DER SLUYS, M., RAYMOND, V., MANDEL, I., RÖVER, C., CHRISTENSEN, N., KALOGERA, V., MEYER, R. and VECCHIO, A. (2008). Parameter estimation of spinning binary inspirals using Markov chain Monte Carlo. *Classical and Quantum Gravity* **25** 184011.

[35] VRUGT, J. A., TER BRAAK, C., DIKS, C., ROBINSON, B. A., HYMAN, J. M. and HIGDON, D. (2009). Accelerating Markov chain Monte Carlo simulation by differential evolution with self-adaptive randomized subspace sampling. *International Journal of Nonlinear Sciences and Numerical Simulation* **10** 273–290.

[36] WOODARD, D. B., SCHMIDLER, S. C., HUBER, M. et al. (2009). Conditions for rapid mixing of parallel and simulated tempering on multimodal distributions. *The Annals of Applied Probability* **19** 617–640.