

HAMILTONIAN MONTE CARLO IN INVERSE PROBLEMS. ILL-CONDITIONING AND MULTIMODALITY

I. Langmore,* M. Dikovsky, S. Geraedts, P. Norgaard, & R. von Behren

1600 Amphitheatre Pkwy, Mountain View, CA 94043, USA

*Address all correspondence to: I. Langmore, 1600 Amphitheatre Pkwy, Mountain View, CA 94043, USA,
E-mail: ianlangmore@gmail.com

Original Manuscript Submitted: 5/7/2021; Final Draft Received: 5/28/2022

The Hamiltonian Monte Carlo (HMC) method allows sampling from continuous densities. Favorable scaling with dimension has led to wide adoption of HMC by the statistics community. Modern autodifferentiating software should allow more widespread usage in Bayesian inverse problems. This paper analyzes two major difficulties encountered using HMC for inverse problems: poor conditioning and multimodality. Novel results on preconditioning and replica exchange Monte Carlo parameter selection are presented in the context of spectroscopy. Recommendations are given for the number of integration steps as well as step size, preconditioner type and fitting, annealing form, and schedule. These recommendations are analyzed rigorously in the Gaussian case and shown to generalize in a fusion plasma reconstruction.

KEY WORDS: *inverse problems, Markov chain Monte Carlo, preconditioning, replica exchange, parallel tempering*

1. INTRODUCTION

The goal of Bayesian inverse problems is to produce and characterize $p(x|y)$, the *posterior* distribution over possible state variables X , given measurements y . In principle, samples from the posterior can be used to determine the mean, quantiles, and other relevant statistics. These samples can be obtained using Markov chain Monte Carlo sampling, which requires only that the log density, $\log p(x|y)$ (assumed absolutely continuous with respect to Lebesgue measure), be available as a function (up to an additive constant). In practice, extracting samples can take prohibitively long, so people often resort to point estimates.

The most common Monte Carlo setup is the *random walk Metropolis-Hastings*. This requires, once burnt-in, $O(N)$ evaluations of $\log p(x|y)$ for each effective sample $X \in \mathbb{R}^N$. A more favorable scaling is obtained using the *Hamiltonian Monte Carlo*, or HMC, which requires only $O(N^{1/4})$ evaluations of $\nabla_z \log p(x|y)$ [1]. Due to the $O(N^{1/4})$ scaling, HMC has seen wide acceptance in the statistics community. However, thus far HMC has seen only minimal usage in the world of inverse problems [2–4]. The barriers to acceptance are real: The gradient evaluations, $\nabla_z \log p(x|y)$, required by HMC, translate to derivatives through a forward model. This task is made easier by recent advancements in autodifferentiating software. The next two barriers are geometrical and appear in somewhat predictable ways in inverse problems. In particular, the second barrier is ill-conditioned posterior covariance, often induced by low rank and/or low noise forward models. These introduce a large multiplier to the $O(N^{1/4})$ scaling that must be dealt with. Another barrier is that of posterior mass separated by regions of extremely low density. When present, this *multimodality* causes such difficulty that the highest priority is crossing these low-density regions.

Our efforts to mitigate ill-conditioned covariance led to reparameterization and linear preconditioning. We contribute rigorous analysis of reparameterization, diagonal, and “full covariance” preconditioning, including an algorithm to select burn-in size. *Replica Exchange Monte Carlo* (abbreviated REMC, also known as *parallel tempering*), is used to deal with multimodality. Novel criteria for selecting annealing form and schedule, number of integration

steps, as well as step size are given. In all cases, a generic Gaussian problem is used to extract concrete recommendations, which we then test on a spectroscopy-based inversion.

Our perspective is the result of Google's ongoing work in reconstruction of (fusion) plasma states [5]. In this industrial research setting, new experimental data arrives daily. Reconstructions must be done, and redone, for *thousands* of experiments. Unexpected artifacts may appear in new measurements that are not well represented by the current model. Physicists modify models weekly and need to understand changes. Sampling code must work well in the majority of reconstructions, and optimal tuning in each situation is not possible. For that reason, we emphasize the use of simple scaling laws and crude algorithmic decisions over intricate methods. With this perspective, we add to the body of inverse-problems-oriented HMC literature. See [2,4,6] for some applications. See also [7] for reparameterizations of HMC relevant to inverse problems in high dimension. See [8] for a lift-and-project approach applicable to inverse problems with low noise. An introduction to HMC can be found in [1]. For those with some HMC experience we recommend [9]. An introduction to Bayesian inverse problems can be found in [10], and an overview of Bayesian modeling in statistics in [11].

Section 3 is a prerequisite for Sections 4 and 5. All other sections can be read independently.

Section 2 describes how common features of inverse problems can lead to difficult sampling situations. Section 3 reviews HMC. Section 4 discusses preconditioning. Section 5 discusses REMC.

2. SOME FEATURES FOUND IN INVERSE PROBLEMS

In the Bayesian inverse problems setup we are interested in, the unknown X parameterizes a physical quantity of interest. For each unique X , a *single* measurement $Y \in \mathbb{R}^M$ is observed. The likelihood is chosen as a (hopefully very accurate) representation of the data generating processes. Often, little to no ground truth examples of X are available. A different situation is often found in the statistics community, where many independent Y are observed for the one and only X . For example, X could be a coefficient of effectiveness of a drug treatment and Y the outcome (recovery or not). The likelihood is often then chosen as a mathematically convenient representation of the distribution of possible Y given X [11]. Often, no attempt is made to describe details of the process by which Y emerges. Both of these contrast with the so-called “generative models” popular in the machine learning community. There, the usual goal is to learn a probabilistic relationship $X \rightarrow Y$ so as to generate new Y [12]. The likelihood there is often taken as a mathematically convenient and extremely flexible function with *learned* parameters.

These differences lead to a different prioritization of challenges for Bayesian inverse problems. The lack of flexibility in choosing a likelihood means we must deal with the covariance structure imposed by the measurement. The need for physically realistic representations of X , combined with often sparse measurements, means we often use parameterizations leading to multiple local maxima of $p(x | y)$. Contrast this with the “generative models” world. There, a deep network can often transform a simple posterior to the observed Y , thus reducing the need to model correlations [13]. Moreover, since recovering X is a nongoal, being stuck in one of many modes is okay, so long as the resultant distribution over Y is the same.

2.1 Poorly Conditioned Posterior Covariance

Consider the toy problem,

$$p(x) \propto \exp\left\{-\frac{1}{2}x^T C_{pr}^{-1}x\right\}, \quad p(y|x) \propto \exp\left\{-\frac{1}{2\sigma^2}\|Ax - y\|^2\right\}, \quad (1)$$

where $A \in \mathbb{R}^{M \times N}$ is the *forward matrix*, mapping the unknown state variable X to the measurement space \mathbb{R}^M . The prior correlation, C_{pr} , is the discretization of Eq. (5). Collecting terms quadratic in x , the posterior covariance is

$$C = [C_{pr}^{-1} + \sigma^{-2}A^T A]^{-1}. \quad (2)$$

Suppose the forward matrix A is low rank and so only constrains $M < N$ directions in \mathbb{R}^N . For small σ , these constrained directions will have small posterior variance asymptotic to σ^2 . The other directions will have variance governed by C_{pr} . The result is poorly conditioned posterior covariance (see Fig. 1). In that example, sampling is

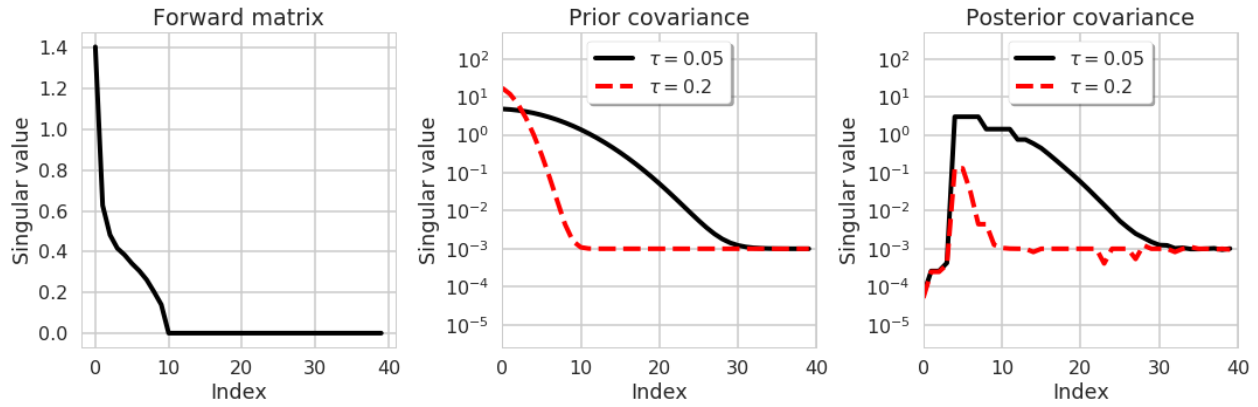


FIG. 1: Singular values of Gaussian problem. Spectra of relevant matrices in Eq. (1). The low rank forward matrix and/or long prior correlation length results in poor conditioning. In all cases, $M = 20$, $N = 40$, and the prior shrinkage $\delta = 0.001$. We sort spectra by their corresponding eigenvectors' (Pearson) correlation with prior eigenvectors. Frequency increases left to right. Left: the forward matrix $A \in \mathbb{R}^{M \times N}$. Center: prior covariance for two different values of prior correlation length τ , cf. Eq. (5). Right: posterior covariance (2) is poorly conditioned.

about 92 times less efficient than if the posterior was ideally conditioned. Section 3.4 details the relationship between sampling efficiency and conditioning.

2.2 Multimodal Posteriors

The posterior corresponding to the linear problem from Section 2.1 will always have exactly one local maximum, or *mode*. In practice, nonlinear parameterizations of state can easily lead to multiple modes. These tend to arise in an attempt to model details that are not fully constrained by the data.

A multimodal toy model is a Gaussian prior for $X \in \mathbb{R}^N$ and mixture of Gaussians likelihood corresponding to $M \leq N$ measurements:

$$p(x) \propto \prod_{n=1}^N e^{-x_n^2/2}, \quad p(y|x) \propto \prod_{m=1}^M \left[e^{-(x_m-1)^2/(2\sigma^2)} + e^{-(x_m+1)^2/(2\sigma^2)} \right]. \quad (3)$$

This leads to posterior density

$$p(x|y) \propto \left(\prod_{m=1}^M \left[e^{(x_m-\mu(\sigma))^2/(2v(\sigma)^2)} + e^{(x_m+\mu(\sigma))^2/(2v(\sigma)^2)} \right] \right) \left(\prod_{n=M+1}^N e^{-x_n^2/2} \right),$$

$$\mu(\sigma) := 1 + \sigma^2, \quad v(\sigma)^2 := \sigma^2/(1 + \sigma^2).$$

When $\sigma \ll 1$, standard HMC will find itself stuck in any of the 2^M modes. The probability of escaping one of these modes is, to leading order, less than $\exp\{-(1 + \sigma^2)/\sigma^2\}$ (see Section 3.5).

2.3 Model Problem: Reconstruction from Spectroscopy

Our model problem is the inference of amplitude, temperature, and velocity of an ion species from an emission spectrum. Photons are emitted from an N -point discretization of the square $[-1, 1] \times [-1, 1]$, and line-integrated emission is measured. Some details are given, so as to emphasize the ability of HMC to handle non-Gaussian posteriors. Other sections do *not* require understanding of these details.

For a set of (around 200) frequencies ν , we parameterize emissivity as a transformation, φ_ν , of an *a priori* Gaussian random variable $X \in \mathbb{R}^N$, $X \sim \mathcal{N}(0, C_{pr})$. At each frequency ν , an integration matrix $\mathcal{I} \in \mathbb{R}^{M \times N}$

projects the emissivity $\varphi_v(X)$ onto M measurements. This models M distinct view *chords* in the $[-1, 1] \times [-1, 1]$ square (Fig. 2):

$$Y_v = \mathcal{I}\varphi_v(X) + \eta_v.$$

The noise $\eta_v \in \mathbb{R}^M$ is, conditional on X , Gaussian, and independent for every frequency and chord. It takes the form

$$\eta_{v,m} \sim \mathcal{N}(0, \sigma^2 + \sigma_p^2 [\mathcal{I}\varphi_v(X)]_m^2).$$

The factor σ gives rise to the usual additive independent noise. σ_p gives us noise proportional to the signal, representing model error. This leads to the likelihood

$$p(y|x) = \prod_{m=1}^M \frac{\exp\left\{-\frac{1}{2\sigma(x)^2}(y_m - [\mathcal{I}\varphi_v(x)]_m)^2\right\}}{\sqrt{2\pi\sigma(x)^2_m}}, \quad (4)$$

$$\sigma(x)^2_m := \sigma^2 + \sigma_p^2 [\mathcal{I}\varphi_v(x)]_m^2.$$

The mapping φ_v is composed of two stages. First, X is divided into independent amplitude, temperature, and velocity components: $X = (X_A, X_T, X_V)$. These are mapped to amplitude \mathcal{A} , temperature \mathcal{T} , and velocity \mathcal{V} . The mapping is either *slab*, meaning constant on chord-aligned rectangles, or *shell*, meaning constant on circles of rotation, about a shifted center (Fig. 2).

To make amplitude and temperature positive, we use the *softplus* function, $\mathcal{S}(u) := \log[1 + \exp\{u\}]$:

$$\mathcal{A} := c_A \mathcal{S}(\tilde{X}_A), \quad \mathcal{T} := c_T \mathcal{S}(\tilde{X}_T), \quad \mathcal{V} := c_V \tilde{X}_V.$$

Above, (c_A, c_T, c_V) are scaling factors, and $\tilde{X} := (\tilde{X}_A, \tilde{X}_T, \tilde{X}_V)$ is a transformation of X to impart radial correlation and a shifted center. That is, the amplitude, temperature, and velocity components of \tilde{X} have radial values approximating a zero mean Gaussian random field with correlation

$$\Gamma_{pr}(r, r') := \exp\left\{-\frac{|r - r'|^2}{2\tau^2}\right\} + \delta. \quad (5)$$

Above, $\delta > 0$ is a *shrinkage* parameter preventing small eigenvalues from creating numerical difficulties. The coordinate r is the radius about a center shifted by an *a priori* 2D normal.

The second step of φ_v is the dimensionless discrete emissivity spectrum,

$$\mathcal{E}_v(\mathcal{A}, \mathcal{T}, \mathcal{V}) := \frac{\mathcal{A}}{\sqrt{2\pi}w} \exp\left(-\frac{(\nu^{-1} - \tilde{\nu}^{-1})^2}{2w^2}\right), \quad (6)$$

where $\tilde{\nu} = \nu_0/(1 - \mathcal{V})$ is the Doppler-shifted center frequency, and $w = \sqrt{\mathcal{T}}/\nu_0$ is the Doppler-broadened bandwidth.

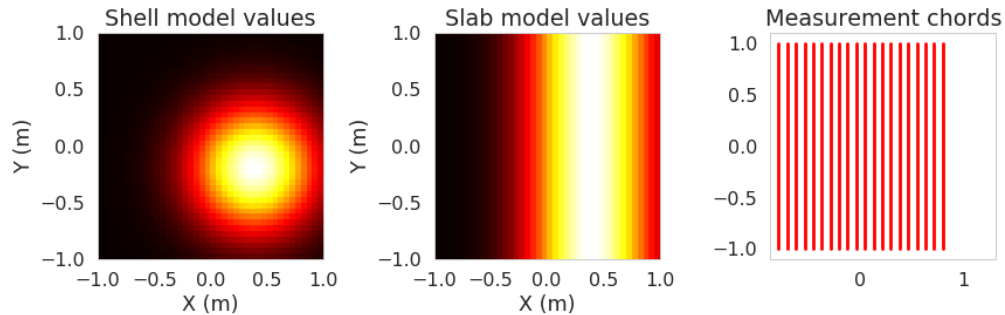


FIG. 2: Left: example shell model values for arbitrary plasma state variable (amplitude, temperature, or density). They are radially smooth and circularly symmetric about a shifted center. Center: example slab model values are constant along rectangles aligned with the measurement chords. Right: 20 rays through a disk shaped body, representing chords of measurement.

In its shell parameterization, the spectroscopy problem can suffer from ill-conditioned posterior covariance. The Gaussian toy problem (1) is a linearization, achieved by assuming amplitude \mathcal{A} is the only unknown, ignoring the softplus, and setting $\sigma_p = 0$ and $c_{\mathcal{A}} = 1$.

Multimodality can also occur. Suppose the measured emissivity Y_ν has two spectral peaks at ν_1 and ν_2 , as a result of carbon III and an additional, unexpected pollutant. As a result, we are trying to fit two peaks with a model capable of producing only one. This results in five different posterior modes (Fig. 3). Changing velocity determines the Doppler shift, and therefore the mode a sample is closest to. In a neighborhood of each mode, the log-posterior is concave. In this way our spectroscopy problem is similar to the toy model (3) with the number of velocity variables, M , equal to $N/3$.

3. REVIEW OF BAYESIAN INFERENCE USING MCMC

Common to all Bayesian inversion is the goal of inferring a probabilistic description of an unobserved quantity $X \in \mathbb{R}^N$, given the observed $Y \in \mathbb{R}^M$. This *posterior*, $p(x|y)$, is proportional to the product of the *prior* $p(x)$ and the *likelihood* $p(y|x)$. That is, $p(x|y) \propto p(x)p(y|x)$. The goal of this inference can be point estimates, such as $\mathbb{E}\{X|Y=y\}$, or a quantification of uncertainty, such as $\text{Var}\{X|Y=y\}$.

Markov chains have been used since 1953 as a means to generate sequences of samples X^1, X^2, \dots from target distributions [14]. These samples can be viewed by themselves or used to approximate expectations $\mathbb{E}\{f(X)\} \approx S^{-1} \sum_{s=1}^S f(X^s)$.

3.1 Statistical Efficiency

Samples from the Markov chain have two major deficiencies. First, only in the limit $k \rightarrow \infty$ can we say that $X^k \sim \pi$. Often a chain is initialized with a draw from the prior, and an initial *burn-in* period is used to allow the chain to “forget” this initial position and move toward the *typical set* of the posterior [15]. Second, draws are correlated. Roughly speaking, we say that identically distributed samples $\{X^1, \dots, X^K\}$ have *effective sample size* (ESS) S if the variance of $K^{-1} \sum_{k=1}^K X^k$ is $\text{Var}\{X^1\}/S$, in other words, if, for purposes of computing the mean, they are “as effective” as S i.i.d. samples. Usually $S \leq K$. The most straightforward computation of ESS assumes the $\{X^k\}$ are identically distributed but correlated draws from the target, with mean μ and variance ν^2 . Then, for dimensions $n = 1, \dots, N$, ESS is built from the t -lag autocorrelation ρ_t ,

$$\text{ESS}_n := \frac{K}{1 + 2 \sum_{t=1}^{\infty} \rho_t}, \quad \rho_t := \frac{\mathbb{E}\{(X_n^1 - \mu_n)(X_n^{1+t} - \mu_n)\}}{\nu_n^2}.$$

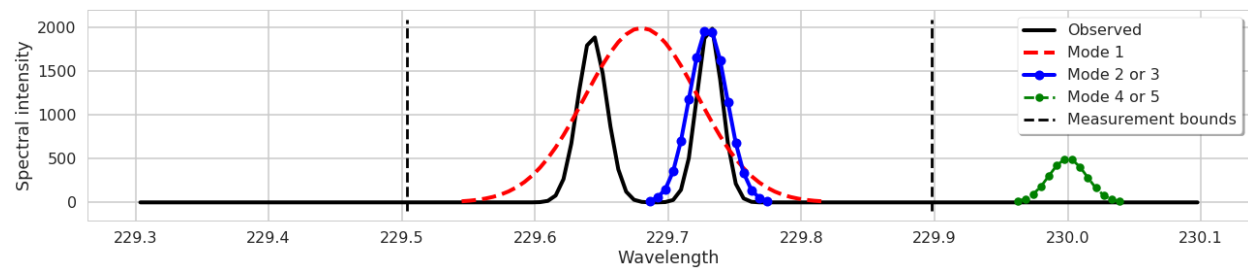


FIG. 3: Multiple modes: observed two-peak measurements, along with hypothetical *posterior predictive* samples ($Y \sim p(y|X)$ where $X \sim p(x|y)$) demonstrating different posterior modes. Mode 1: if the temperature prior allows for large values and/or the modeled noise is large enough, a local maximum is a hot plasma with bulk velocity near zero, giving rise to a single wide spectra that covers both peaks. Mode 2 or 3: the next two modes arise if the temperature is *a priori* small enough, with velocity values either positive or negative. These correspond to modeled spectral peaks that cover one of the measured peaks. Mode 4 or 5: if the velocity prior is permissive and the amplitude prior discourages tiny values, we have two more local maxima. These have velocity values so large that the modeled peak is shifted far to the left or right, outside the measured range of frequencies.

We are usually most interested in the worst-performing dimension and therefore $\min_n \{\text{ESS}_n\}$. Due to insufficient burn-in, or being stuck in local modes, the assumption that X^k are draws from the target is often not satisfied. For that reason, our version of ESS will be lower if the mean differs between multiple chains (see Appendix A of [16]). A more direct method of checking deviation between chains is the *potential scale reduction*, or \hat{R} [16].

3.2 Computational Efficiency

In our experience, obtaining Markov chain Monte Carlo (MCMC) samples on production models is a computationally intensive task that can take prohibitively long. Graphical processing units (GPUs) provide a significant speedup, as they excel at parallel (SIMD) operations. This allows concurrent running of multiple Markov chains [17]. The slow-down as more chains are added is sublinear, until GPU memory is exhausted. We therefore only consider algorithms that take advantage of GPU parallelism. In particular, our replica-exchange implementation uses the same number of leapfrog steps for each replica (see Section 5.3.4).

The majority of HMC computation time is spent in the numerical (leapfrog) integration of Hamilton's equations. Accordingly, we track the number of leapfrog steps per effective sample ($\min_n \{\text{ESS}_n\}$ actually) as a performance metric. This allows platform-independent comparisons.

A secondary performance consideration is time spent compiling and optimizing the computational graph [18,19]. This takes a fixed amount of time proportional to the graph complexity. In our model problem, where ill conditioning and/or multimodality was present, graph optimization was a small fraction of total runtime. In well-conditioned HMC sampling problems, avoiding many graph optimizations may be important.

3.3 Description of HMC

The Hamiltonian Monte Carlo (HMC) method was introduced in 1987 as ‘‘Hybrid Monte Carlo’’ for use in lattice field theory simulations [20]. Since then, it has been recognized as an efficient alternative to the random walk Metropolis-Hastings method, well suited for higher-dimensional problems. Implementations are available for a variety of languages [17,21,22]. A comprehensive introduction to HMC can be found in [1].

To sample $X \in \mathbb{R}^N$, distributed according to a smooth (with respect to Lebesgue measure) density $\pi(x)$, HMC augments state space with a fictitious momentum $\xi \in \mathbb{R}^N$. This defines the joint density

$$\pi(x, \xi) \propto \exp\{-H(x, \xi)\}, \quad \text{where} \quad H(x, \xi) := -\log \pi(x) + \frac{\|\xi\|^2}{2},$$

and $\|\xi\|$ is the Euclidean norm. In the physics setting, the *Hamiltonian* H is total energy, whereas $-\log \pi(x)$, $\|\xi\|^2/2$ are potential and kinetic energies. It is not uncommon to model the kinetic energy $\sqrt{\xi \cdot A^{-1} \xi}/2$, where $A \in \mathbb{R}^{N \times N}$ is the *mass matrix*. However, as shown in Section 4 of [1], this is equivalent to the linear preconditioning $X \mapsto LX$ (where $A^{-1} = LL^T$) in conjunction with the Euclidean norm. Kinetic energy can be non-Gaussian, e.g., depend on position [23] (see also [15]).

Sampling proceeds by (a numerical approximation to) the following iteration from point (x^j, ξ^j) :

1. Draw $\tilde{\xi} \sim \mathcal{N}(0, I_N)$.
2. Let $(x(t), \xi(t))$ be the time t solution to the Hamilton's equations of motion: $\dot{x} = \xi$, $\dot{\xi} = \nabla \log \pi(x)$, with initial condition $(x^j, \tilde{\xi})$.
3. Set $(x^{j+1}, \xi^{j+1}) = (x(t), \xi(t))$ for integration time t .

Each integration path lives on a single level set of the Hamiltonian. The resampling step $\tilde{\xi} \sim \mathcal{N}(0, I_N)$ is necessary to jump between level sets and is thus necessary for ergodicity [24].

In practice, Hamilton's equations must be solved numerically over ℓ steps with step size h . Denote this solution by Ψ^ℓ . The integration error means we can no longer just accept the move in step 3, which is replaced by a Metropolis correction:

$$(x^{j+1}, \xi^{j+1}) = \Psi^\ell, \quad \text{with probability} \quad a(x^j, \xi^j \rightarrow \Psi^\ell),$$

and

$$(x^{j+1}, \xi^{j+1}) = (x^j, \xi^j), \quad \text{with probability } 1 - a(x^j, \xi^j \rightarrow \Psi^\ell),$$

for acceptance probability

$$a(x^j, \xi^j \rightarrow \Psi^\ell) := \min(1, \exp\{H(x^j, \xi^j) - H(\Psi^\ell)\}). \quad (7)$$

Since Hamilton's equations preserve the Hamiltonian, if numerical integration was perfect, $H(x^j, \xi^j) = H(\Psi^\ell)$ and every step would be accepted. In practice, finite step size leads to some rejections and wasted effort. We also note that for the Metropolis correction to be symmetric, the final momentum should be negated before evaluating H . This makes no difference since our Hamiltonian is symmetric in momentum.

The numerical integration is usually done with ℓ steps of the *Störmer-Verlet* or *leapfrog integrator*. This symplectic integrator ensures that the Hamiltonian does not diverge, provided h is sufficiently small (see [25], Theorem 8.1).

3.4 HMC Step Size Scaling Laws

Here we review existing scaling laws for HMC step size. These results always inform our choice of HMC step size and integration time. They are directly used in Section 4.

The step size h , along with the number of leapfrog steps ℓ , are two important parameters to choose. Usually, h is chosen to achieve some desired acceptance rate, and ℓ is set to the desired integration time divided by h . If h is too large, the average acceptance probability, $P[\text{Accept}]$, will tend to zero. On the other hand, if h is too small, then too many steps ℓ will be required, and computational effort will be wasted. In [26,27], $X \in \mathbb{R}^N = \mathbb{R}^K \times \dots \times \mathbb{R}^K$ with density $\pi(x_1, \dots, x_{N/K}) = f(x_1) \dots f(x_{N/K})$ is studied in the limit $N \rightarrow \infty$. This so-called *i.i.d. limit* yields the result that h should be tuned until $0.6 \leq P[\text{Accept}] \leq 0.9$.

In the Gaussian case, we can extract more precise conclusions, resulting in useful analysis and algorithms that can be used, even in non-Gaussian problems. Consider a target $X \sim \mathcal{N}(0, C)$, where C has eigenvalues $\lambda_1^2 \geq \lambda_2^2 \geq \dots \geq \lambda_N^2 > 0$. If we want the integration trajectories to travel a distance comparable to the largest scale λ_1 , we must have $h\ell = O(\lambda_1)$. On the other hand, to avoid instability we must have $h \leq 2\lambda_N$. This leads to a naive scaling $\ell = O(\lambda_1/\lambda_N)$. The ratio λ_1/λ_N is the common, *spectral*, condition number of any matrix L such that $C = LL^T$.

The problem with the spectral condition number is that it takes into account only the largest and smallest scales. The largest scale does set the required integration length, but *all* dimensions contribute to integration error. A condition number taking these considerations into account was introduced in [28]:

$$\kappa(L) := \left(\sum_{n=1}^N \left(\frac{\lambda_1}{\lambda_n} \right)^4 \right)^{1/4} = \|L\|_2 \|L^{-1}\|_{S^4}. \quad (8)$$

Above, $\|\cdot\|_2$ is the *spectral* norm, and $\|\cdot\|_{S^4}$ is the fourth *Schatten* norm [29]. For matrix A with singular values $\lambda_1 \geq \dots \geq \lambda_N \geq 0$, the k th Schatten norm is

$$\|A\|_{S^k} := \left[\sum_{n=1}^N \lambda_n^k \right]^{1/k}. \quad (9)$$

In [28] it is shown that under some regularity conditions, a Gaussian density requires $O(\kappa)$ leapfrog steps for efficient sampling. κ may be approximated using the largest scale λ_1 , step size h , and average acceptance probability $P[\text{Accept}]$:

$$\kappa \approx \frac{\lambda_1}{h} 2^{7/4} \sqrt{\Phi^{-1} \left(1 - \frac{P[\text{Accept}]}{2} \right)}. \quad (10)$$

Above, Φ is the normal cumulative distribution function. Thus κ is the number of steps needed to traverse the largest scale, λ_1/h , times a correction factor depending on acceptance probability. In non-Gaussian problems, we recommend using large (small) λ_1/h as evidence of *possible* (efficient) sampling.

3.5 Conductance in Multimodal HMC

Let $A \subset \mathbb{R}^N$, and $c(A)$ be the average probability of the chain escaping A in one step. The *conductance* of the chain, $\mathcal{C} := \min_{\pi(A) < 1/2} \{c(A)\}$, quantifies its ability to escape local modes. In [30] it is shown that the conductance of HMC is no better than that of the random walk Metropolis-Hastings method. An upper bound is also given for conductance of HMC chains:

$$c(A) \leq t \frac{1}{2} \frac{\int_{\partial A} \pi(x) dx}{\pi(A)}. \quad (11)$$

From Eq. (11) it is clear that an increasing integration time can increase conductance at most linearly. Since computational effort also increases linearly with t , this does not help overall.

Equation (11) also shows that conductance can be constricted by a single low-probability (density) surface. For Gaussian noise models (e.g., Section 2.3), this can lead to conductance decreasing exponentially in $1/\sigma^2$. Indeed, for the bi-modal normal $0.5\mathcal{N}(-1, \sigma^2) + 0.5\mathcal{N}(1, \sigma^2)$, the leading-order term of conductance is proportional to $\exp\{-1/(2\sigma^2)\}$ (see [30] Theorem 3).

Tempering methods need to use a highest temperature T_{\max} such that the corresponding “hottest density” has nonvanishing conductance (see Section 5.3.2).

4. PRECONDITIONING OF THE POSTERIOR COVARIANCE

The HMC (as well as standard Metropolis Hastings) method works better when sampling from a unit Gaussian. The preconditioning techniques below sample a transformed variable $Z = F^{-1}(X)$, which (hopefully) looks more like a unit Gaussian. These samples Z^j are transformed back into $X^j = F(Z^j) \sim p(x|y)$.

4.1 Transformation by Diffeomorphism

Here we write some relations involving pushforwards by diffeomorphisms. They will be needed in later sections.

Let us start with $X \sim p(\cdot|y)$, and a diffeomorphism F , which transforms $X \mapsto Z = F^{-1}(X)$. Equivalently, the density $p(\cdot|y)$ is transformed by the *pushforward* operator, $F_{\#}^{-1}$, into g :

$$g(z) = (F_{\#}^{-1}p(\cdot|y))(z) := |\det(DF(z))| p(F(z)|y). \quad (12)$$

Above, DF is the matrix of partial derivatives, $(DF)_{ij} = \partial F_j / \partial z_i$. Using HMC, we sample from the transformed density g , producing Z^1, \dots, Z^K . Transforming back, $X^k := F(Z^k)$, and we have samples from $p(\cdot|y)$ as desired.

In the Gaussian case, $p(\cdot|y) \sim \mathcal{N}(\mu, C)$, with $C = LL^T$, the linear preconditioner induced by a matrix F transforms $L \mapsto F^{-1}L$, and likewise the covariance and κ as

$$\begin{aligned} LL^T &\mapsto (F^{-1}L)(F^{-1}L)^T = (F^{-1})LL^T(F^{-1})^T \\ \kappa(L) &\mapsto \|F^{-1}L\|_2 \|(F^{-1}L)^{-1}\|_{S^4}. \end{aligned} \quad (13)$$

4.2 Preconditioning by Prior Reparameterization

If the support of $p(x|y)$ is bounded, sampling directly from it will suffer from boundary issues. It is therefore standard practice to transform $X \rightarrow Z := F^{-1}(X)$ (as in Section 4.1) such that $\text{supp}(Z) = \mathbb{R}^N$. We know of no existing work analyzing the effects of this reparameterization on conditioning, so we include it here.

In our case, the prior $p(x)$ is a transformation of a standard Gaussian by a diffeomorphism G . That is, $p(x) = (G_{\#}\phi)(x)$. This means $Z = G^{-1}(X)$ is a useful transformation. If the prior results in difficult posterior covariance (see, e.g., Section 2.1), using G as a preconditioner will often improve conditioning. To see this, note that

$$G_{\#}^{-1}p(x|y) = p(G(x)|y)|\det(DG(x))| \propto \phi(x)p_Y(y|G(x)). \quad (14)$$

Thus, preconditioning with G is equivalent to a reparameterization that uses a standard Gaussian prior and inserts G inside the likelihood (probably achieved by using $G(x)$ in the forward model). In our work, we sometimes explicitly

preconditioned with G and at other times reparameterized a unit normal. Not only are these mathematically equivalent, but, due to caching in TensorFlow probability, they are computationally equivalent ($G \circ G^{-1}$ is replaced by the identity) [31]. Doing one of these is necessary, as it removes much of the nonlinearity and allows the methods of Section 4.3 to work efficiently.

Organizing code around reparameterizations has some advantages: First, the benefits of preconditioning by the prior are realized without the programmatic complexity of specifying a preconditioner. Second, the function G does not need to be a diffeomorphism (see [5] for an example). On the other hand, there are benefits to letting the prior encode the state directly. First, in this setup the MCMC variables being sampled are the state variables you care about. Second, one can use a prior *not* easily described as a transformation of a Gaussian.

The sampling benefits/degradation of prior reparameterization will vary. Consider the Gaussian example (1). If $C_{pr} = L_{pr}L_{pr}^T$, then the reparameterized posterior covariance becomes

$$[I + \sigma^{-2}(AL_{pr})^T(AL_{pr})]^{-1}. \quad (15)$$

We see in Fig. 4 that reparameterization can help when prior covariance is the major contributor to κ , but can hurt if low noise is the major source of small eigenvalues.

4.3 Linear Preconditioning for Gaussian Problems

Here we examine linear preconditioning of Gaussian distributions. This simplification allows for precise results and inspires techniques. These techniques are applied to non-Gaussian problems in Section 4.4.

The setup here is like other diffeomorphisms (Section 4.1), except we assume $X \sim \mathcal{N}(0, C)$ is Gaussian, and the preconditioning transformation $F \in \mathbb{R}^{N \times N}$ is linear. If FF^T is a scalar multiple of C , then Eq. (13) shows the transformed covariance is a multiple of the identity, which minimizes κ . Subsequent sections will use approximations of this factor built from samples obtained during burn-in.

4.3.1 Diagonal Preconditioning

A simple approximation of a covariance factor is the diagonal matrix \hat{D} made with sample standard deviations. This approximates D , the diagonal matrix of *actual* standard deviations. The next three propositions are original, and illustrate why we like this preconditioner.

First, preconditioning with D should work well in the diagonally dominant case. The proof follows from the Gershgorin circle theorem [29].

Proposition 1. Suppose $C = LL^T$ is diagonally dominant with, for every i , $\sum_{j \neq i} |C_{ij}/C_{ii}| \leq \delta < 1$. Then,

$$\kappa(D^{-1}L) \leq N^{1/4} \sqrt{\frac{1+\delta}{1-\delta}}.$$

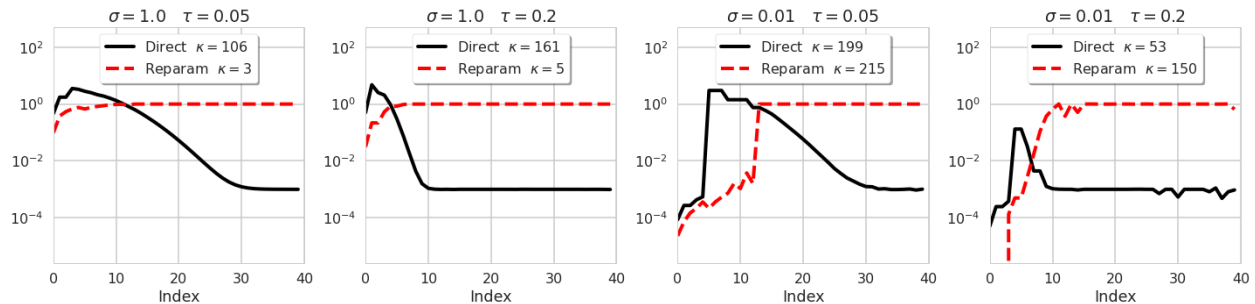


FIG. 4: Covariance spectra. Eigenvalues indexed by correlation with the prior eigenvectors. From the posterior before reparameterization (2), and after (15). Reparameterization “lifts” the high-frequency posterior scales. Different noise levels σ and prior correlation lengths τ result in reparameterization helping or hurting. In all cases, $M = 20$, $N = 40$, and the prior shrinkage $\delta = 0.001$. The minimal condition number κ [see Eq. (8)] is $40^{1/4} \approx 2.5$.

Second, D is close to the ideal diagonal preconditioner, especially in the near-diagonal case.

Proposition 2. *Let D_{opt} be a preconditioner minimizing $\kappa(G^{-1}L)$ over all diagonal matrices G . Then,*

$$\kappa(D^{-1}L) \leq \sqrt{N} \kappa(D_{\text{opt}}^{-1}L).$$

Furthermore, if at most K entries in each row of LL^T are nonzero, then

$$\kappa(D^{-1}L) \leq \sqrt{K} \kappa(D_{\text{opt}}^{-1}L).$$

The proof is found in Appendix A.1. A corollary is that preconditioning with D can hurt by at most a factor of \sqrt{N} (or \sqrt{K}).

Third, in practice, we use the diagonal matrix of sample standard deviations, \hat{D} , rather than D . If we use S i.i.d. samples from $\mathcal{N}(0, LL^T)$, then S need only grow logarithmically with N .

Proposition 3. *Given $\varepsilon, p \in (0, 1)$,*

$$\kappa(\hat{D}^{-1}L) \leq \kappa(D^{-1}L) \sqrt{\frac{1+\varepsilon}{1-\varepsilon}},$$

with probability p , as soon as the number of i.i.d. samples S satisfies

$$S \geq \frac{25}{\varepsilon^2} \log\left(\frac{3N}{p}\right).$$

Proof. We have the distributional equality, $\hat{D} \stackrel{d}{=} \sqrt{(I + \Delta)} D$, where $\Delta \in \mathbb{R}^{N \times N}$ is diagonal,

$$\Delta_{nn} := \frac{1}{S} \sum_{s=1}^S Z_{s,n}^2 - 1 - \left(\frac{1}{S} \sum_{s=1}^S Z_{s,n} \right)^2,$$

and $Z_{s,n}$ are i.i.d. normal variates. Then, using Lemma 3,

$$\kappa(\hat{D}^{-1}L) = \|\hat{D}^{-1}L\|_2 \|L^{-1}\hat{D}\|_{S^4} \leq \sqrt{\frac{1 + \max_n \{\Delta_{nn}\}}{1 + \min_n \{\Delta_{nn}\}}} \kappa(D^{-1}L).$$

The proof will be complete once we show that our condition on S implies $\max_n |\Delta_{nn}| \leq \varepsilon$ with probability less than p . This follows from Lemma 6 and the fact that $\mathbb{P}[\max_n |\Delta_{nn}| \geq \varepsilon] \leq N \mathbb{P}[|\Delta_{11}| \geq \varepsilon]$. \square

4.3.2 Full Covariance Preconditioning

By *full covariance preconditioning*, we mean starting with the sample covariance, \hat{C} , factorizing as $\hat{C} = \hat{L}\hat{L}^T$, then preconditioning with \hat{L} . After stating two results on full covariance preconditioning, we discuss a scheme for implementing it. The results and the scheme were discussed in our previous work [28]. The scheme is discussed in more detail and implemented here for the first time (Section 4.4).

Remarkably, if the samples are independent, the condition number does *not* depend on the true covariance.

Lemma 1. *Suppose (X^1, \dots, X^S) are i.i.d. samples of $X \sim \mathcal{N}(0, C)$, and we precondition sampling of X with the S -sample factor \hat{L} . Then the preconditioned κ follows the law of $\kappa(B)$, for $BB^T \sim \text{InverseWishart}(S, N)$.*

In the high-dimensional limit, κ for inverse Wishart matrices has a simple expression.

Proposition 4. *If $BB^T \sim \text{InverseWishart}(N, S)$, and $N \rightarrow \infty$ with $S/N \rightarrow \omega \in (1, \infty)$, then*

$$\frac{\kappa(B)}{N^{1/4}} \rightarrow \frac{(1 + \omega^{-1})^{1/4}}{1 - \omega^{-1/2}},$$

almost surely.

See Fig. 5 for a visualization of Lemma 1 and Proposition 4. Due to the “universality” of random matrices, we expect these results to hold for linear transformations of a wide variety of i.i.d. random variables [32,33].

These results lead to a useful preconditioning scheme. Let κ_0 be the condition number before preconditioning, and κ_S be the condition number after preconditioning with S i.i.d. samples. Assuming the sampling rate is proportional to $1/\kappa$, the time to obtain S_f “final” samples is proportional to $S\kappa_0 + S_f\kappa_S$. On the other hand, without preconditioning, the time is proportional to $S_f\kappa_0$. This means the speedup from preconditioning is

$$\frac{S_f\kappa_0}{S\kappa_0 + S_f\kappa_S}. \quad (16)$$

Estimating κ_0 using Eq. (10), and using Proposition 4 as an expression for κ_S , we can compute speedup for various S . If the maximal speedup (using S^* samples) is > 1 , we proceed with drawing S^* burn-in samples, precondition with \hat{L} , then draw our S_f final samples. If not we draw S_f samples without preconditioning.

Mentioning some practicalities is in order. Burn-in samples obtained using standard HMC are far from independent. As a remedy, we use the no-U-turn sampler (NUTS) [34] to obtain the S^* preconditioning burn-in samples and stop sampling when the mean (across dimensions) effective sample size is S^* . In our experience, obtaining NUTS samples takes around three times longer than standard HMC samples. This happens since NUTS sampling involves doubling the trajectory length and resampling within these long trajectories. Moreover, the additional preconditioning stage requires another step-size adaptation stage. We therefore replace κ_0 with $4\kappa_0$ in the denominator of Eq. (16). This leads to Algorithm 1. See also plots of samples in different stages in Fig. 6.

Each step-size adaptation in Algorithm 1 is done via an iterative scheme [35], invoked after preemptively adjusting step size using Eq. (10). Step size adjustment comprises around 30% of runtime. This could often be shorter, but this longer adaptation makes the algorithm more robust to stuck chains.

4.4 Application to a Weakly Non-Gaussian Problem

Here we compare preconditioning schemes as applied to the shell model from Section 2.3. This problem is non-Gaussian. In particular, the noise level depends on the signal, the temperature and amplitude are constrained to be positive via a Softplus, the coordinate system is shifted, and emissivity is a nonlinear function. This is still “weakly” non-Gaussian, since observed skew and kurtosis levels of transformed samples Z were close to that of a Normal.

The schemes compared are referred to as “full,” “diag,” and “none.” “Full” uses Algorithm 1. “Diag” uses Algorithm 1 but skips the NUTS sampling and uses diagonal rather than sample covariance preconditioning. “None” does not precondition. Code was run on Tesla P100 GPUs.

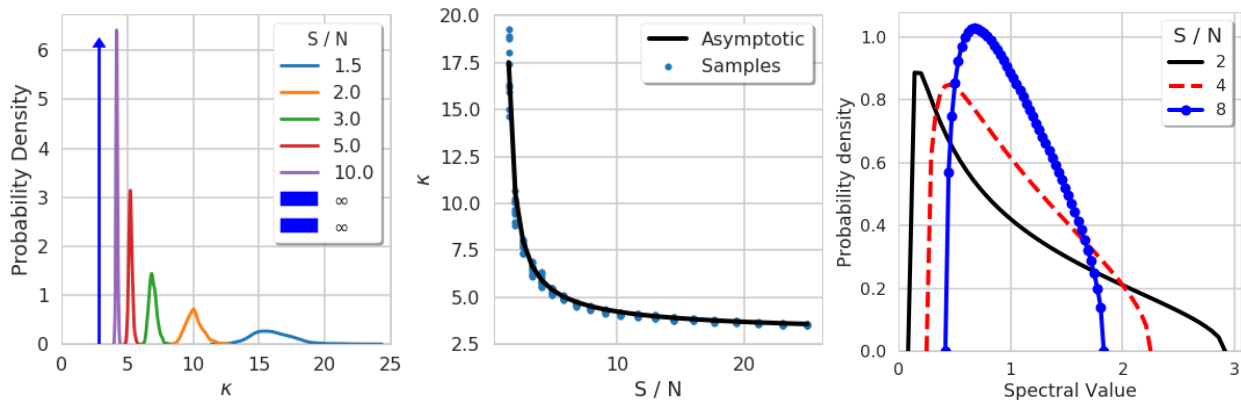


FIG. 5: Density and asymptotic $\kappa(B)$, when $BB^T \sim \text{InverseWishart}(N, S)$ with $N = 64$. Left: density plots of sample values of $\kappa(B)$ for different S/N . $\kappa(B) \rightarrow N^{1/4}$ as $S/N \rightarrow \infty$. Center: asymptotic estimate (from Proposition 4) vs. samples of κ . Once $S/N \approx 20$, κ is close to the ideal value of $N^{1/4} \approx 2.8$. Right: the Marčenko-Pastur density, which is the limiting spectral density of $\text{Wishart}(N, S)$.

Algorithm 1: Sampling stages for unimodal problems. Abandon and restart using REMC if \hat{R} fails to reduce fast enough

Initialize 20 chains by sampling from the prior;
 Start h small enough so $P[\text{Accept}] \approx 1.0$, then adapt h until $P[\text{Accept}] \approx 0.9$. Use number of leapfrog steps $\ell = 5$. When done, set $\ell = (1/h)(\pi/2)$;
 Draw stage 1 samples. Use them to compute the largest scale λ_1 , then set $\ell = (\lambda_1/h)(\pi/2)$;
 Draw stage 2 samples. Use them to recompute λ_1 and $P[\text{Accept}]$. Compute κ_0 using Eq. (10) and maximal speedup using Eq. (16);
if *maximal speedup* > 1 **then**
 while $N^{-1} \sum_{n=1}^N \{\text{ESS}_n(\text{stage 3 samples})\} < S^*$ **do**
 Draw more stage 3 samples using NUTS;
 end
 Precondition using the stage 3 sample covariance;
 Adapt step size until $P[\text{Accept}] \approx 0.9$, and set $\ell = (1/h)(\pi/2)$;
end
while $\text{ESS}_n(\text{final stage samples}) < S_f$ **do**
 Draw more *final* stage samples;
end
Result: S_f “final” samples

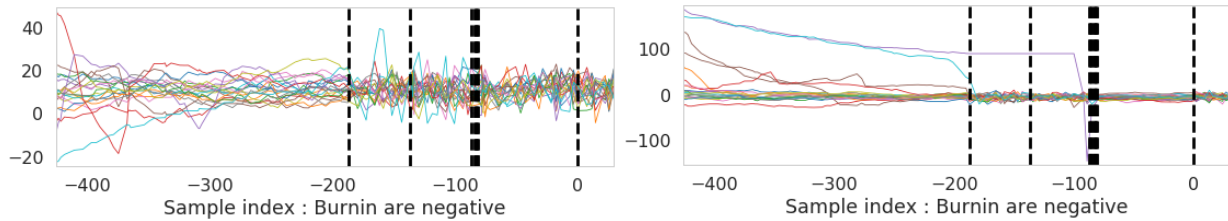


FIG. 6: Traces of Algorithm 1. Plots of coordinate samples (Z_n^s) are the most important diagnostic tool. Left: stages of Algorithm 1 are divided by dotted lines. Leftmost is the initial step-size adaptation phase. Using a large number of samples here allows chains to reach the typical set. The low ℓ value and low initial h means sampling proceeds slowly but is likely to be stable. Second from the left are the “stage 1” samples. These have ℓ large enough to get a reasonable estimate of λ_1 and also allow chains more time to reach the typical set. Next are the “stage 2” samples, used to compute κ_0 and the number of preconditioning samples needed, S^* . Next are four very closely spaced dashed lines, within which NUTS sampling was used to obtain S^* effective samples. These are used to form the sample covariance factor \hat{L} used for preconditioning. Second stage from the right is step-size adaptation done after preconditioning. The final stage, starting at 0, includes the first 25 “final” samples. Right: same stages, in a case where preconditioning hurt. The reason here is that some chains did not reach the typical set before preconditioning samples were taken. This led to a bad preconditioner.

Twenty plasmas to reconstruct were drawn from the prior. The reconstruction model used a variety of noise levels from $\sigma = 1.25$ to 15. Each reconstruction was run until $\min_n \{\text{ESS}_n\} = S_f$, for $S_f \in \{400, 1600, 6400\}$. A total of 282 reconstruction configurations were attempted for each of the three schemes. Eleven configurations were thrown out, since at least one model failed to reduce \hat{R} fast enough. This is usually the result of being stuck in a local mode due to poor initialization. If this happens in production, our algorithm restarts with REMC. The traces in Fig. 6 help visualize different stages of Algorithm 1. Figure 7 shows κ correlates well with sampling efficiency, and can be predicted from Proposition 4, even in this non-Gaussian problem. Figure 8 shows that the full preconditioner significantly speeds up sampling, at the cost of a more expensive burn-in. Diagonal preconditioning helps only a little.

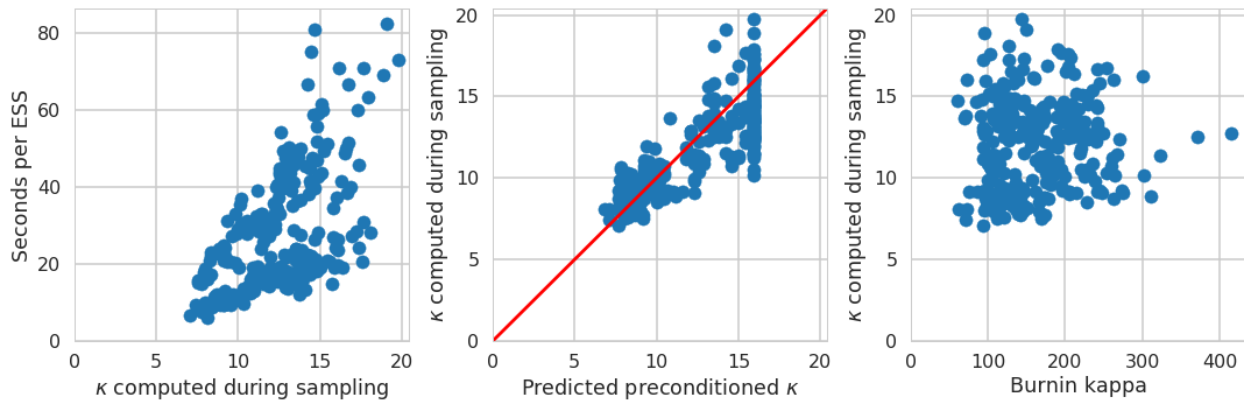


FIG. 7: Using Eq. (10), we compute κ at various sampling stages to show our formulas apply, even in non-Gaussian problems. Experiments when the burn-in κ was huge (upper 2%) are not shown, as these distort the plot. Preconditioning often made the situation worse for these. Left: the relationship between κ and the seconds required for effective samples is somewhat close to linear. This validates (16) as a measure of speedup from preconditioning. Center: the predicted post-preconditioning value of κ matches nicely with the actual value obtained by preconditioning, validating Proposition 4. Right: the post-preconditioning κ_S , is plotted against the burn-in value, κ_0 . This shows a significant reduction in κ due to Algorithm 1.

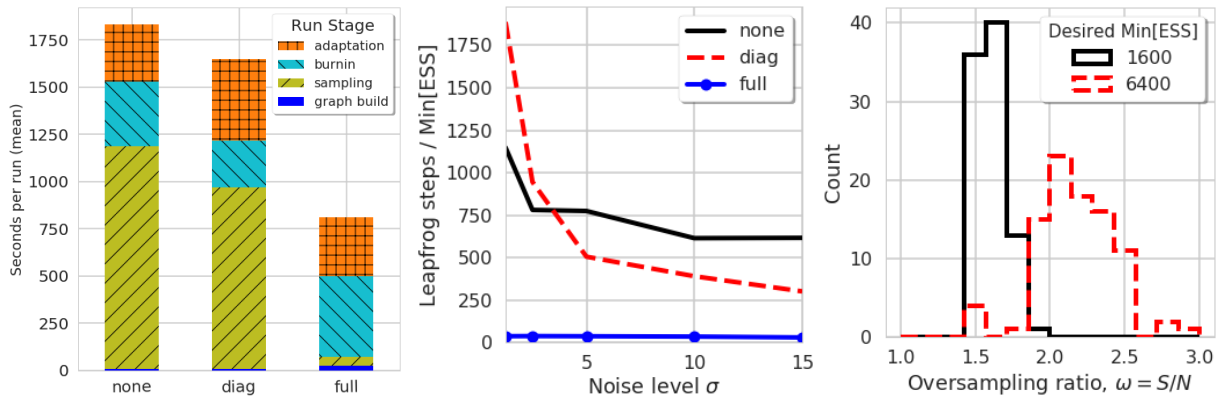


FIG. 8: Preconditioner comparison from the study outlined in Section 4.4. Left: average runtime breakdown ($S_f = 1600$) shows that while the burn-in time is longer for full preconditioning, the sampling time is significantly shorter. Center: full covariance preconditioning results in sampling efficiency around 36, which is around $30\times$ better than not preconditioning. Right: histogram of oversampling ratio $\omega = S/N$ selected by Eq. (16) for two values of desired $\min_n \{ESS_n\}$. When the desired $\min_n \{ESS_n\}$ is larger, a larger ω is selected.

5. TEMPERING TO SAMPLE WITH MULTIMODALITY

Best practices for sampling from multimodal distributions are not as easy to come by as for their unimodal counterparts. For example, linear preconditioning usually does not help. A popular family of techniques involves using a number of modifications of the target, each *tempered* by temperature T . The terminology and history is rooted in statistical mechanics [36]. These techniques make use of the fact that if π is a probability density, and temperature $T > 1$, the density proportional to $\pi^{1/T}$ will have lower peaks and higher troughs. Hence it will be better able to jump between modes.

5.1 Replica Exchange Monte Carlo (REMC)

This section reviews REMC (also known as *parallel tempering*). Related techniques, such as *annealed importance sampling*, also deserve consideration [37].

Given posterior $p(x|y) \propto p(x)p(y|x)$, and sequence of temperatures $1 = T_1 < T_2 < \dots < T_R \leq \infty$, we form the *replica densities* π_r in one of two ways:

$$\pi_r(x) \propto \begin{cases} p(x)^{1/T_r} p(y|x)^{1/T_r}, & \text{posterior tempering (requires } T_R < \infty), \\ p(x)p(y|x)^{1/T_r}, & \text{likelihood tempering.} \end{cases} \quad (17)$$

To gain intuition, consider the unimodal example where the prior $p(x) \sim \mathcal{N}(0, I)$ and the likelihood $p(y|x) \sim \mathcal{N}(\mu, \Gamma)$. The posterior covariance after tempering with T will be

$$\Gamma_{\text{post}}(T) := \begin{cases} T[I + \Gamma^{-1}]^{-1}, & \text{posterior tempering,} \\ [I + T^{-1}\Gamma^{-1}]^{-1}, & \text{likelihood tempering.} \end{cases} \quad (18)$$

Posterior tempering increases posterior variance without changing the shape or condition number. Likelihood tempering distorts the posterior covariance to make it look like the prior (see Fig. 9).

Together, these form the joint density $\pi(x_1, \dots, x_R) := \pi_1(x_1) \cdots \pi_R(x_R)$. Samples from $(X_1^k, \dots, X_R^k) \sim \pi$ are generated, but only the target samples $X_1^k \sim \pi_1 = p(\cdot|y)$ are kept. REMC repeats two alternating steps. In the *exploration* step, each replica progresses independently, $X_r^{k-1} \rightarrow X_r^k$. HMC, or another sampling method, can be used here. In the *communication* step, a number of *swaps* are proposed between adjacent replicas. For example, the $(1, 2)$ swap proposes that replicas 1 and 2 exchange position, $X_1^{k+1} = X_2^k$ and $X_2^{k+1} = X_1^k$. The set of swaps proposed at each turn are either the set of even swaps, $\{(1, 2), (3, 4), \dots\}$, or odd swaps, $\{(2, 3), (4, 5), \dots\}$. In both

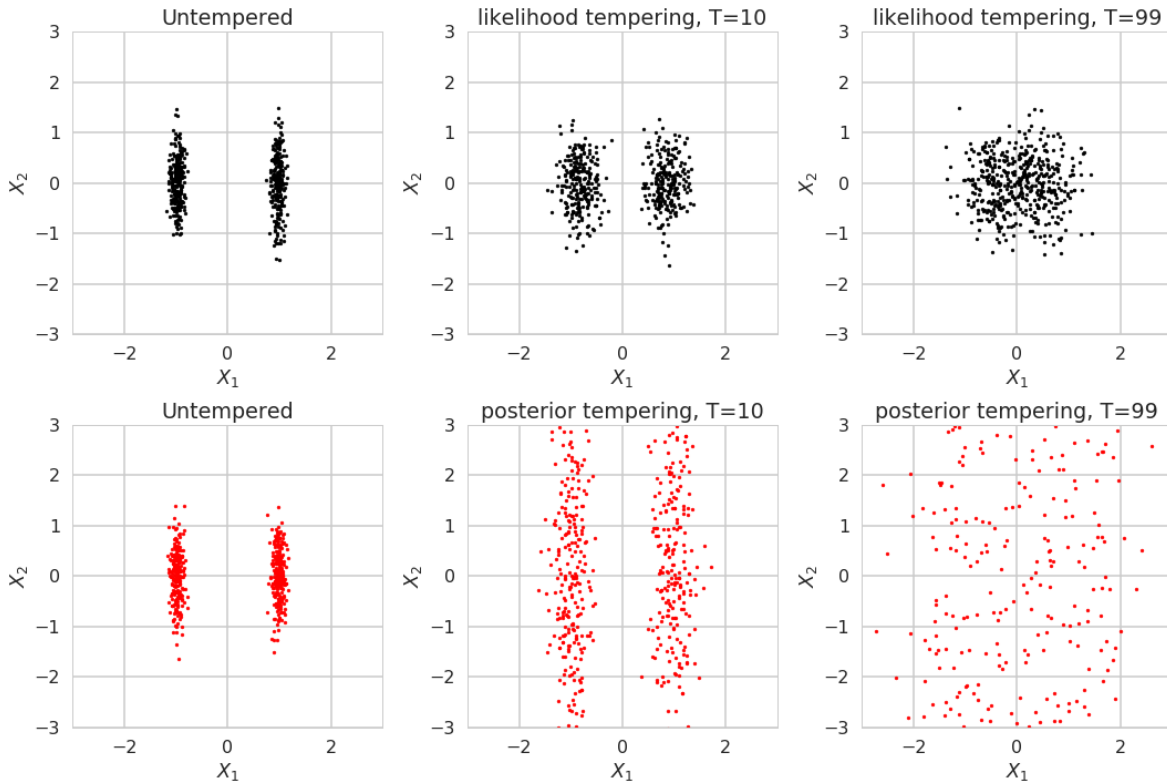


FIG. 9: Likelihood and posterior tempering. Sampling from a tempered bi-modal normal [an $M = 1$ version of Eq. (3)]. Top: likelihood tempering means the hottest ($T = 99$) replica samples come from a (nearly) isotropic Gaussian. Bottom: posterior tempering means the $T = 10$ replica samples come from a (nearly) bimodal normal, with each mode being (nearly) as poorly conditioned as the posterior ($T = 1$) modes.

cases, each swap is independently accepted or rejected according to the standard Metropolis-Hastings criteria. For example,

$$\begin{aligned} \mathbb{P}[\text{Swap}_{(1,2)} | x_1, x_2] &= \min\{1, \alpha_{(1,2)}(x_1, x_2)\}, \\ \alpha_{(1,2)} &:= \frac{\pi(x_2, x_1, x_3, \dots)}{\pi(x_1, x_2, x_3, \dots)} = \frac{\pi_1(x_2)\pi_2(x_1)}{\pi_1(x_1)\pi_2(x_2)}. \end{aligned} \quad (19)$$

The computational cost of swapping is negligible compared with leapfrog integration. We therefore propose swaps between every exploration step. In most of the literature, swaps are proposed using a stochastic even-odd (SEO) scheme, whereby a coin flip chooses between even and odd swaps. More recently, deterministically alternating between even and odd swaps was proposed [38]. This deterministic even-odd (DEO) scheme has superior scaling characteristics [39]. We use DEO in all experiments.

5.2 Swapping, Dimension Laws, and Underconstrained Problems

Effective REMC requires the mean swap probabilities,

$$\mathbb{P}[\text{Swap}_{(r,r+1)}] = \mathbb{E}\left\{\mathbb{P}[\text{Swap}_{(r,r+1)} | X_r, X_{r+1}]\right\},$$

be large enough to allow information from the hottest replica (the replica using the highest temperature) to make its way to the target. This section shows that, unlike conductance, swap probability is strongly related to dimension.

5.2.1 Existing Fundamental Results

In higher dimensions, samples concentrate in a thin neighborhood of the *typical set* [15]. For example, if π is log-concave, the typical set is $\{x : \log \pi(x) = \mathbb{E}\{\log \pi(X)\}\}$, and the neighborhood grows (relatively) thinner as dimension increases [40]. Since swapping of replicas r and $r+1$ must lead to valid samples from their respective densities, they must swap about as often as these neighborhoods overlap. This overlap is made explicit by Proposition 5.

Proposition 5. *Let X_r be a sample from replica π_r defined by Eq. (17). The mean swap probability can be written in terms of the untempered posterior/likelihood:*

$$\mathbb{P}[\text{Swap}_{(r,r+1)}] = \begin{cases} 2 \mathbb{P}[p(X_r | y) < p(X_{r+1} | y)], & \text{posterior tempering,} \\ 2 \mathbb{P}[p(y | X_r) < p(y | X_{r+1})], & \text{likelihood tempering.} \end{cases}$$

Proof. The proof has likely been shown many times before (e.g., [41]), for the case of posterior tempering. Consider likelihood tempering, and write $\pi_r(x) \propto \exp\{-V(x)/T_r - V_0(x)\}$. The Metropolis criteria (19) gives

$$\mathbb{P}[\text{Swap}_{(r,r+1)}] = \int \min\{\pi_r(x_r)\pi_{r+1}(x_{r+1}), \pi_r(x_{r+1})\pi_{r+1}(x_r)\} dx_r dx_{r+1}. \quad (20)$$

Since $T_r < T_{r+1}$,

$$\begin{aligned} \pi_r(x_r)\pi_{r+1}(x_{r+1}) < \pi_r(x_{r+1})\pi_{r+1}(x_r) &\iff \frac{V(x_r)}{T_r} + \frac{V(x_{r+1})}{T_{r+1}} > \frac{V(x_r)}{T_{r+1}} + \frac{V(x_{r+1})}{T_r} \\ &\iff V(x_r) > V(x_{r+1}). \end{aligned}$$

This leads us to split the integral (20) up over regions $\{V(x_r) > V(x_{r+1})\}$ and $\{V(x_r) < V(x_{r+1})\}$. A switch of the dummy variables x_r, x_{r+1} in the second shows that both integrals are equal to $\mathbb{P}[V(X_r) > V(X_{r+1})]$, which gives to the desired result. The case of posterior tempering is similar. \square

For REMC to work well, information must propagate from the hottest replica (replica using the highest temperature) to the target. To study this, one can keep track of the *index process* of temperatures. For example, chain k may start by sampling from π_R , then swap and sample from π_{R-1} , then π_{R-2} , π_{R-1} and so on. The corresponding indices would be $(R, R-1, R-2, R-1, \dots)$. A *round trip* occurs when a chain starts at index k , reaches R , then 1, then back to k . The average number of round trips, starting from all replicas, after S swap attempts, is the *round trip rate*. To derive round trip rates for likelihood tempering when $T_R = \infty$, [39] makes three assumptions: First, *stationarity*, $X_r \sim \pi_r$, which is reasonable after burn-in. Second, *efficient local exploration* (ELE). ELE means that if $X \sim \pi_r$ and X' is the result of local exploration (e.g., HMC integration) starting from X , then the potential energy is independent. In the case of posterior tempering, this means $\log[p(X|y)]$ and $\log[p(X'|y)]$ are independent, and for likelihood tempering, $\log[p(y|X)]$ and $\log[p(y|X')]$ are independent. Third, they assume integrability of the cubed log likelihood. This leads to round-trip rates for the SEO and DEO swapping schemes:

$$\tau_{\text{SEO}} = \frac{1}{2R + 2\gamma}, \quad \tau_{\text{DEO}} = \frac{1}{2 + 2\gamma}, \quad (21)$$

where γ is the *schedule inefficiency*

$$\gamma := \sum_{r=1}^{R-1} \frac{1 - \text{P}[\text{Swap}_{(r,r+1)}]}{\text{P}[\text{Swap}_{(r,r+1)}]}.$$

This justifies using DEO rather than SEO. Importantly for us, 2τ is the fraction of samples, starting from π_R , that make their way down to the target π_1 .

Note that ELE will be violated if chains are stuck in different modes, and the modes do not have identical energy surfaces. In other words, we expect ELE to be violated in most multimodal problems. Nonetheless, [39] finds that the results of this section roughly held in a variety of problems despite ELE being violated.

As $\max_r |T_r^{-1} - T_{r+1}^{-1}| \rightarrow 0$, the swap probabilities are governed by the increasing function, $\Lambda(T)$, which satisfies the Theorem 1.

Theorem 1 ([39] Theorem 2). *For annealing schedule $1 = T_1 < T_2 < \dots < T_R \leq \infty$,*

$$1 - \text{P}[\text{Swap}_{(r,r+1)}] = \Lambda(T_{r+1}) - \Lambda(T_r) + O(\max_r |T_r^{-1} - T_{r+1}^{-1}|^3).$$

When $T_R = T_{\max}$ is fixed, Λ is understood to mean $\Lambda(T_{\max})$, the *global communication barrier*.

Ignoring the error term in Theorem 1, the round-trip rate τ_{DEO} is optimized when

$$1 - \text{P}[\text{Swap}_{(r,r+1)}] \equiv \Lambda/R. \quad (22)$$

Consider running k copies (chains, in our terminology) of REMC independently, with a total computational budget of \bar{R} . In other words, $\bar{R} = kR$. In this setup, [39] derives the optimal number of chains k^* , number of replicas R^* , and round-trip rate τ_{DEO}^* :

$$R^* = 2\Lambda + 1, \quad k^* = \frac{\bar{R}}{R^*} = \frac{\bar{R}}{2\Lambda + 1}, \quad \tau_{\text{DEO}}^* = \frac{k^*}{2 + 4\Lambda} = \frac{\bar{R}}{2(2\Lambda + 1)^2}. \quad (23)$$

This optimum is achieved when $\text{P}[\text{Swap}_{(r,r+1)}] \equiv \Lambda/R \approx 0.5$, although they recommend $\text{P}[\text{Swap}_{(r,r+1)}] > 0.5$ to reduce the ELE violation.

An asymptotic expression for $\Lambda(T_R)$ is also provided in the i.i.d. regime. Here, one adds a parameter d , and with $N = d \cdot N'$, assumes the prior and likelihood acts in an i.i.d. manner on each of the d copies of $\mathbb{R}^{N'}$. In other words,

$$\log p(x|y) = \sum_{i=1}^d V(x_i), \quad \log p(x) = \sum_{i=1}^d \tilde{V}_0(x_i). \quad (24)$$

We restate their proposition, extending it to posterior tempering.

Proposition 6 ([39] Proposition 4). *Given 24, as $d \rightarrow \infty$, we have asymptotic convergence*

$$\Lambda_d(T_R) \stackrel{asy.}{\sim} \sqrt{\frac{1}{\pi}} \int_1^{T_R} \frac{\sigma(T)}{T^2} dT,$$

where, with $X(T)$ the tempered state,

$$\sigma^2(T) = \begin{cases} \text{Var}\{\log p(X(T) | y)\}, & \text{posterior tempering,} \\ \text{Var}\{\log p(y | X(T))\}, & \text{likelihood tempering.} \end{cases}$$

In this i.i.d. regime, $\sigma^2 = O(d)$ and hence $\Lambda = O(\sqrt{d})$. It follows that $\bar{R}/\tau_{DEO}^* = O(d)$ units of work are done to produce each sample making its way from the hottest replica to the target.

5.2.2 Number of Replicas and Its Relation to Underconstrained Problems

This section makes its point by example, although results should apply more generally. The example is a linear/Gaussian problem, with $A \in \mathbb{R}^{M \times N}$ and $\text{Rank}(A) = M < N$:

$$p(x) \propto \exp\left\{-\frac{1}{2}x^T C_{pr}^{-1}x\right\}, \quad p(y|x) \propto \exp\left\{-\frac{1}{2\sigma^2}\|Ax - y\|^2\right\}.$$

In this case, the log posterior is a sum of N terms, whereas the log likelihood a sum of M . If these terms are independent enough, we expect Proposition 6 and Eq. (23) to show the optimal number of replicas, R^* , is $= O(\sqrt{N})$ for posterior tempering, and $= O(\sqrt{M})$ for likelihood tempering. Figure 10 shows this relation holds for a toy problem.

More precise results can be obtained for unimodal distributions. This approach is partially justified by noting that if R replicas are needed to exchange in unimodal density π , at least R should be needed to exchange in a multimodal distribution where one mode looks like π .

Consider an arbitrary Gaussian posterior, and posterior tempering. Proposition 5 shows that with $\chi_N^2, \tilde{\chi}_N^2$ two independent chi-square random variables,

$$\text{P}[\text{Swap}_{(r,r+1)}] = \text{P}[T_r \chi_N^2 - T_{r+1} \tilde{\chi}_N^2 > 0].$$

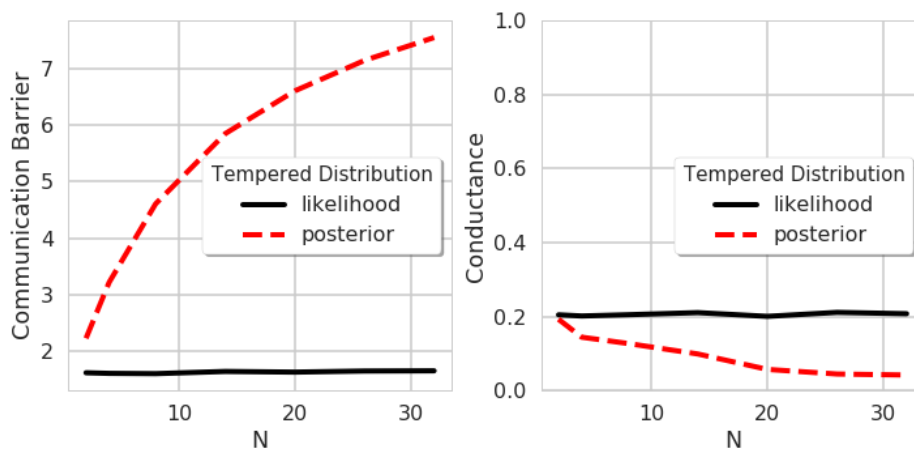


FIG. 10: Tempering comparison. The communication barrier, Λ , and conductance in the bimodal normal of Eq. (3) are shown, and N is increased while fixing $M = 1$. These show the \sqrt{N} , \sqrt{M} scaling of the communication barrier for posterior tempering discussed in Section 5.2.2. The resultant conductance is much better for likelihood tempering.

Since the mean of $T_r \chi_N^2 - T_{r+1} \tilde{\chi}_N^2$ is $N(T_r - T_{r+1}) < 0$, the probability is non-vanishing as $N \rightarrow \infty$ only if the standard deviation is of the same order. This is satisfied if $T_{r+1}/T_r = 1 + c/\sqrt{N}$, for some $c > 0$ depending only on the desired acceptance probability. If $T_{\max} < \infty$ is chosen ahead of time, $T_{\max} = (T_2/T_1)^{R-1} = (1 + c/\sqrt{N})^{R-1}$. It follows that $R \propto \sqrt{N} \log T_{\max}$. This line of reasoning can be extended to any distribution with constant heat capacity [41].

To analyze the case of likelihood tempering, it will help to rewrite the variance of the potentials from Proposition 6 as specific heatlike quantities:

Lemma 2. *Let $X(T) \sim \pi^{1/T}$ be a tempered state. Then, for posterior tempering,*

$$\text{Var}\{\log p(X(T) | y)\} = -T^2 \frac{d}{dT} \mathbb{E}\{\log p(X(T) | y)\},$$

and for likelihood tempering,

$$\text{Var}\{\log p(y | X(T))\} = -T^2 \frac{d}{dT} \mathbb{E}\{\log p(y | X(T))\}.$$

With this in hand, suppose the prior covariance $C_{pr} = I$ in Eq. (1), and $\text{Rank}(A) = M \leq N$. Suppose A has singular values $\{\alpha_n\}$. If the nonzero α_n are all equal, one can use Lemma 2 and Proposition 6 to derive an asymptotic relation (as $M \rightarrow \infty$) for likelihood tempering. Assuming the relation holds when α_n is nonconstant, we have

$$\Lambda \stackrel{asy.}{\sim} \sqrt{\frac{1}{2\pi}} \int_0^1 \sqrt{\sum_{n=1}^N \frac{\alpha_n^4}{(\beta \alpha_n^2 + \sigma^2)^2}} d\beta \leq \sqrt{\frac{M}{2\pi}} \log \left[\frac{1 + (\sigma/\alpha_1)^2}{(\sigma/\alpha_1)^2} \right].$$

Equation (23) now indicates that the optimal number of replicas should be $= O(\sqrt{M})$.

5.3 Selecting Parameters for REMC with HMC

The work of previous sections allow us to give concrete recommendations.

5.3.1 Likelihood vs. Posterior Tempering

We usually prefer likelihood tempering. The reasons are that (i) it has better scaling properties for underconstrained problems (Section 5.2.2), (ii) it is stable no matter how large T_{\max} is, and (iii) the hottest replica is close to the prior, facilitating the leapfrog heuristic of Section 5.3.4.

Likelihood tempering is limited in that the highest temperature replica is approximately the prior. For that reason, if there is a possibility of modes far outside the typical set of the prior, posterior tempering should be used.

5.3.2 Selecting the Annealing Schedule

The annealing schedule is $1 = T_1 < T_2 < \dots < T_R = T_{\max}$. We select this in two stages. First we find T_{\max} , and second we adjust the other temperatures.

T_{\max} must be hot enough so that the hottest replica can mix well and explore the proximity of all posterior modes. The dimension N need not play a role. For example, with our spectroscopy model, tempering changes the noise term $\sigma(x)^2$ [in Eq. (4)] to $T\sigma(x)^2$. We therefore expect $T_{\max} \propto \sigma^{-2}$ to be hot enough for nonvanishing conductance (see Section 3.5). The constant of proportionality will depend on the maximum residual $|y - [\mathcal{I}\varphi_v(x)]_m|$ on some path between modes. We cannot expect to know this *a priori*. Instead, during burn-in we start with geometrically increasing temperatures. We then monitor \hat{R} ([16]) to see which replicas mix *without* swapping. T_{\max} is set to the coldest temperature that was mixing during this burn-in phase. For likelihood tempering, we *could have* set $T_{\max} = \infty$, but we found that to be less efficient.

After $T_R = T_{\max}$ is chosen, we adjust T_2, \dots, T_{R-1} until $P[\text{Swap}_{(r,r+1)}]$ is close to constant, as justified by Eq. (22). We use an interpolation scheme as in [39].

5.3.3 Selecting Step Sizes for Each Replica

A good initial guess for the r th replica's step size, h_r , is $h_r \propto \sqrt{T_r}$, since this would be ideal for a Gaussian. After every temperature change, step sizes can be adjusted by an iterative scheme [35]. It helps to have an initial guess for the new step sizes. This can be done by finding an interpolating function, $h = f(T)$, that is piecewise linear in \sqrt{T} .

5.3.4 Selecting the Number of Leapfrog Steps

To take advantage of batch operations on a single GPU, the number of leapfrog steps ℓ should be the same for all replicas (see Section 3.2). Given likelihood tempering, the hottest replica will be similar to the prior, and should be much better conditioned than the target. We therefore expect the hottest replica will need far fewer leapfrog steps than the target. Since we choose T_{\max} so that only the hottest replica is mixing well (without swapping), we should choose ℓ to facilitate mixing of this hottest replica. This allows $\ell \propto 1/h_R$, which is less than the $O(1/h_1)$ needed for the target to mix well (without swapping).

As it turns out, we can reduce ℓ even further. Equation (21) indicates that only one out of $1 + \gamma$ samples produced by the hottest replica make their way to the target. These $1 + \gamma$ intermediate samples give additional time for the hottest replica to mix. To use this opportunity, note that leapfrog integration travels for a time $\approx h_R \ell$, and if $h_R \ell$ is small, sampling approximates a random walk. The expected time traveled by these $1 + \gamma$ intermediate samples is therefore proportional to $h_R \ell \sqrt{1 + \gamma}$. Assuming the hottest replica has largest scale of $\lambda_{R,1}$, we should choose ℓ so that $h_R \ell \sqrt{1 + \gamma} \approx \lambda_{R,1} \pi/2$, or

$$\ell \approx \lambda_{R,1} \frac{\pi}{2} \frac{1}{h_R \sqrt{1 + \gamma}}. \quad (25)$$

Reducing ℓ according to this *leapfrog heuristic* means the integration length is less than ideal for every replica. As a result, state samples X^j will certainly not be independent. We also expect the potential energy samples, $-\log[p(y | X^j)]$, to lack independence as well. In other words, the ELE assumption of [39] (see Section 5.2.1) will be violated. This can be visualized by plotting chain traces (see Fig. 11). Swapping is turned off for these plots, since we want to visualize *local* exploration.

The toy model in Fig. 11 implemented Eq. (3) with $N = 10$, $M = 5$, 55 replicas, and noise $\sigma = 0.025$. We also set $T_{\max} = \sigma^{-2}$ to ensure the hottest replica was barely mixing. Other temperatures were geometrically distributed. We estimated $\lambda_{R,1}$ using an exact formula for the largest scale of the tempered unimodal version of Eq. (3).

We next used the same toy model but with swapping turned on ($P[\text{Swap}_{(r,r+1)}] \approx 0.75$). The noise level σ was swept from 0.00075 to 0.05 while keeping the number of replicas R fixed at 55. This varies the schedule inefficiency γ . See Fig. 12, where following our leapfrog heuristic exactly (*leapfrog multiplier* = 1) led to more efficient sampling than shorter (*leapfrog multiplier* < 1) or longer (*leapfrog multiplier* > 1) integration times.

In a problem where modes are not Gaussian, the optimal integration time will no longer be given by the $\pi/2$ heuristic, and the energy distribution may be such that an ELE violation is more problematic. To test these effects, we used the multimodal version of our spectroscopy model from Section 2.2. Since our REMC implementation runs R concurrent chains on GPU at once, memory usage increases R times. We therefore limited R to 150. We used the parameter choices outlined in Sections 5.3.2 and 5.3.3, and assumed $\lambda_{R,1} = 1$. Figure 13 shows potential energy traces, similar to the toy-model traces of Fig. 11. Again, we see correlated samples. However, with the spectroscopy model, different modes have different average potential energy for $T = 1$.

As with the toy model, we used a *leapfrog multiplier* to test variations on the leapfrog heuristic (25). Sampling is once again most efficient if the heuristic is followed exactly (Fig. 14). We also compared setting $T_{\max} = \infty$ in addition to the coldest temperature that mixes without swapping. $T_{\max} = \infty$ was much less efficient, supporting the recommendations given in Section 5.3.2.

ACKNOWLEDGMENTS

The authors would like to acknowledge the entire TensorFlow probability team at Google. In particular, Colin Carroll's feedback was helpful in preparation. Also, Saifuddin Syed was helpful in confirming the correctness of our reframing of results from [39].

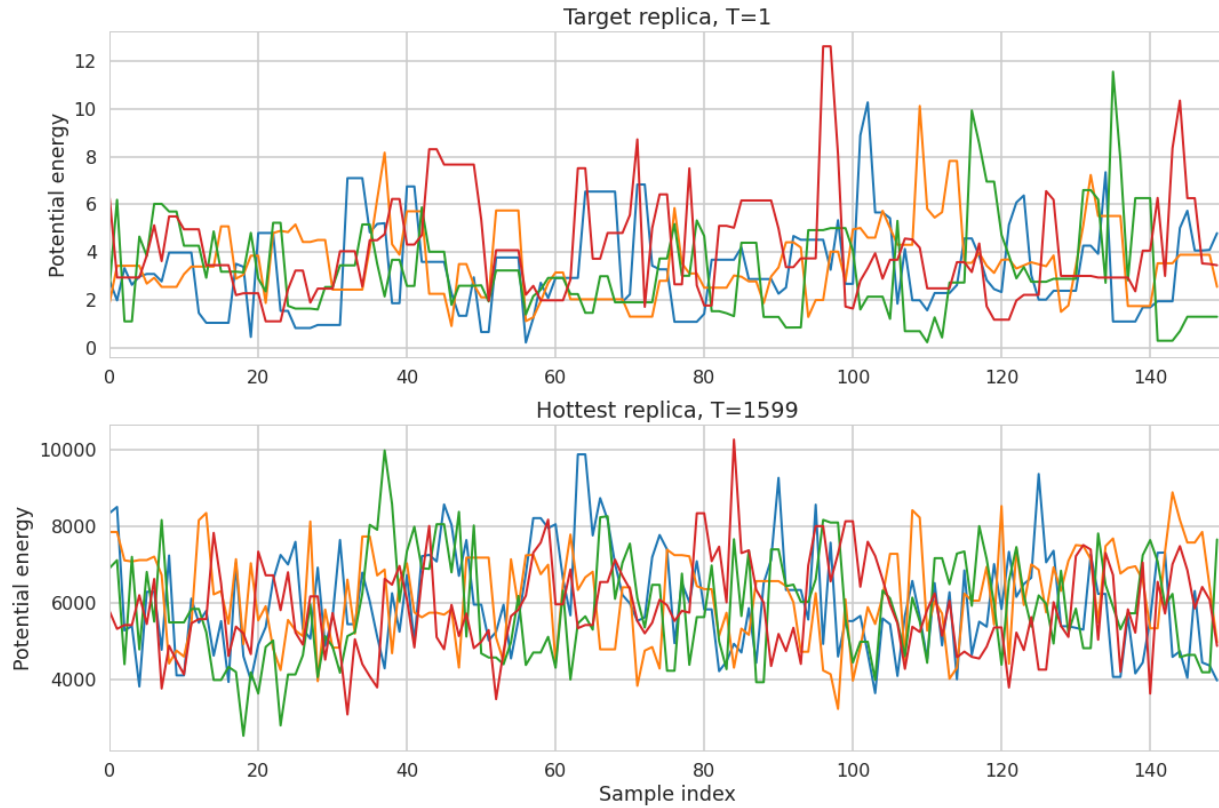


FIG. 11: Potential energy traces from the toy model (3). Three chain traces of $-\log[p(y | X)]$ are shown for the target $T = 1$ and hottest $T = 1599$ replicas. Autocorrelation length is longer in the target. In neither case are samples independent. Since the two modes in the toy model are identical, one cannot tell (by potential energy alone) if the chains are in different modes.

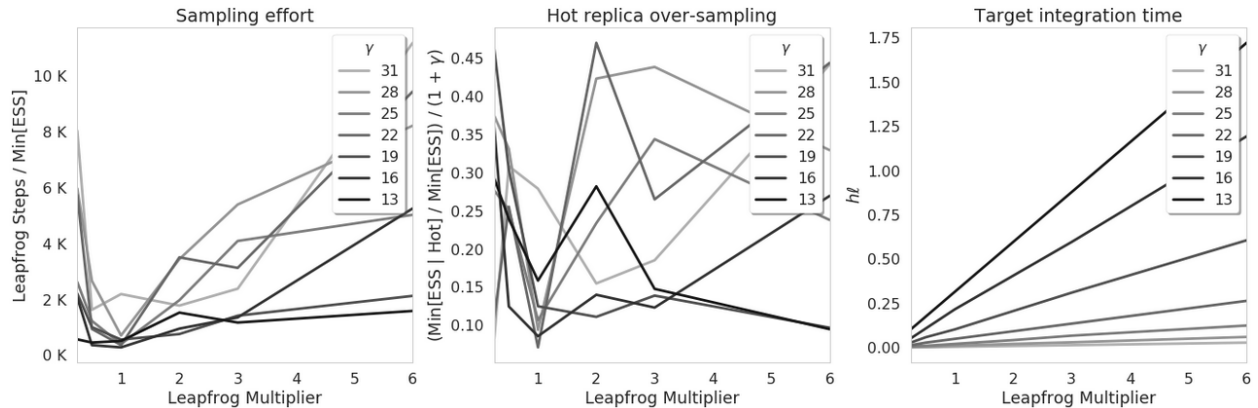


FIG. 12: Optimal number of leapfrog steps and γ , toy model. The (likelihood) tempered posterior from Eq. (3) was run for a number of different σ . Each σ corresponds to a different schedule inefficiency γ . The x-axis is a *leapfrog multiplier*. The number of leapfrog steps taken is this multiplier times the heuristic (25). Left: sampling effort, as measured in leapfrog steps per $\min_n\{\text{ESS}_n\}$, is lowest when the leapfrog multiplier = 1. So following the heuristic (25) exactly was most efficient. Center: the number of effective samples obtained by the hottest replica (in relation to the target, and γ) was 1/2 to 1/10 that predicted by Eq. (21). Perhaps extra effective samples were picked up along the “hot to target” trip. Right: the target integration time, $h_1 \ell$, is significantly less than would be required to efficiently explore the largest target scale ($= 1$) without REMC. So (i) the leapfrog heuristic is saving a significant number of steps, (ii) the state samples are correlated, hinting that ELE is likely violated, and (iii) even so, following our leapfrog heuristic was the most efficient choice.

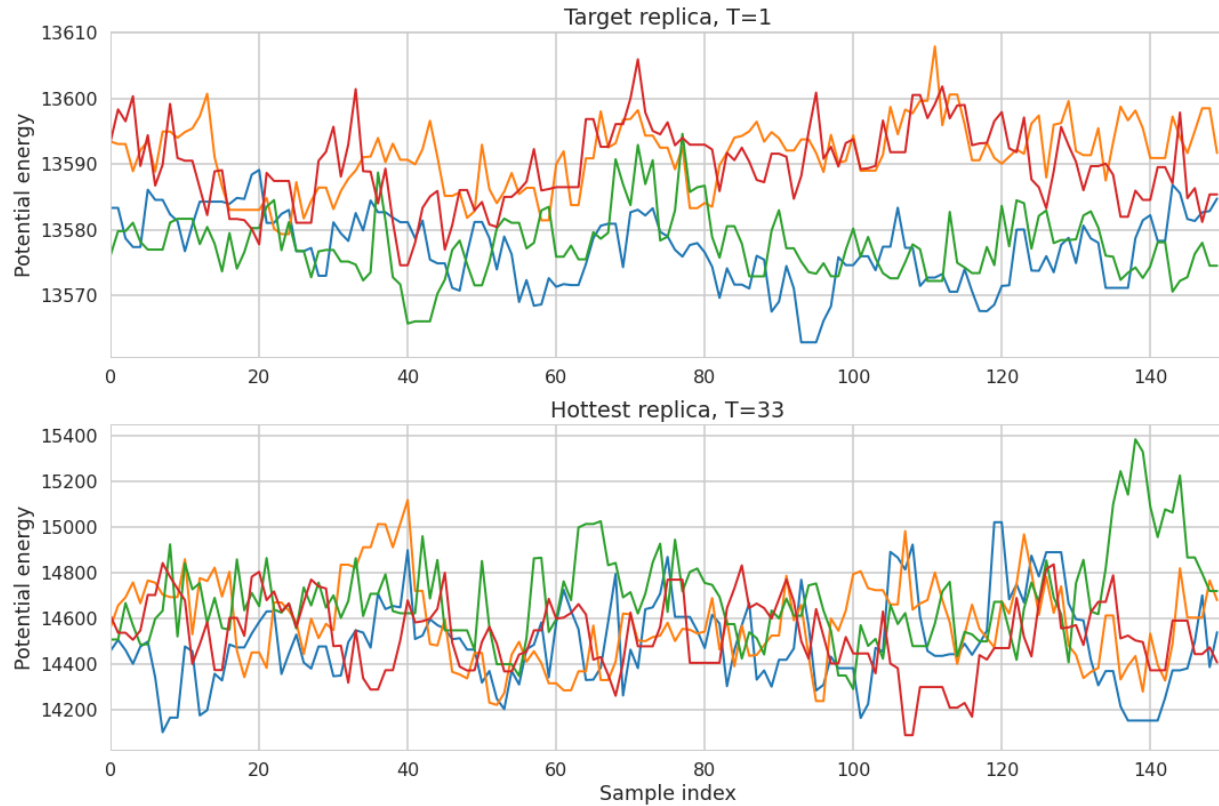


FIG. 13: Potential energy traces from the spectroscopy model. Three chain traces shown for the target $T = 1$ and hottest $T = 33$ replicas. Both replicas exhibit autocorrelation within each chain. The target replica's chains also exhibit different average potential energy. This happens since the target replica is unable to jump between modes, and the modes have different potential energy. The autocorrelation of each chain (not shown) is longer for the target $T = 1$ than hottest replica $T = 33$. This is expected since both use the same number of leapfrog steps, but the target must use a smaller step size.

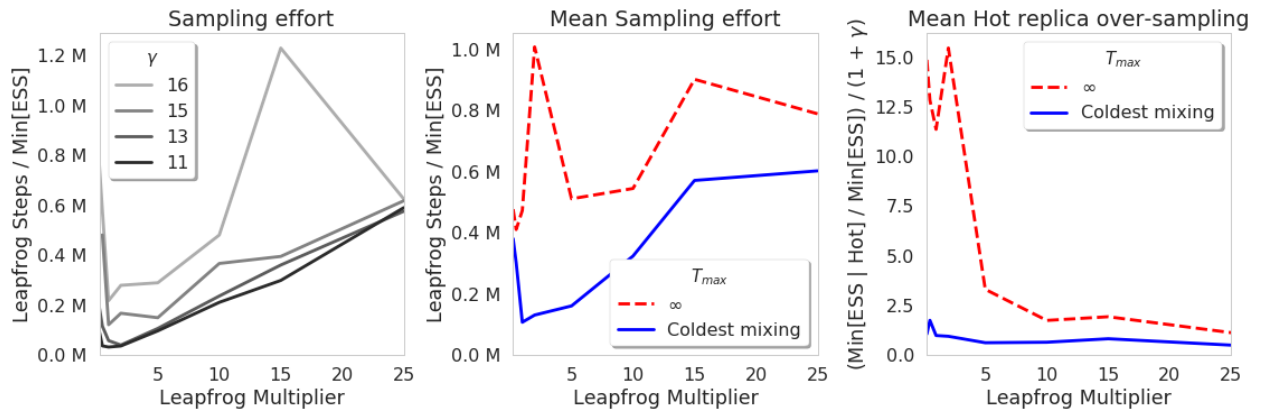


FIG. 14: Efficiency, number of leapfrog steps, and γ in the spectroscopy model. The (likelihood) tempered multimodal posterior from Section 2.2. X and Y axis are as in Fig. 12. The “Mean” is taken over experiments with different γ . Left: following the leapfrog heuristic was the most efficient choice. Center: experiments were also done with $T_{\max} = \infty$ in addition to the coldest temperature that mixes without swapping (recommended in Section 5.3.2). When $T_{\max} = \infty$, sampling was less efficient, and the leapfrog heuristic selected too few steps to be efficient. Right: when $T_{\max} = \infty$, the ratio of (effective) samples drawn by the hottest replica to that of the target was greater than $1 + \gamma$. This is unexpected and unexplained. For both T_{\max} values, once ℓ is high enough, Eq. (21) is satisfied (the Y value is ≈ 1), presumably because ELE is no longer violated.

REFERENCES

1. Radford, N., MCMC Using Hamiltonian Dynamics, in *Handbook of Markov Chain Monte Carlo*, S. Brooks, A. Gelman, G. Jones, and X.L. Meng, Eds., Vol. 54, Boca Raton, FL: CRC Press, pp. 113–162, 2011.
2. Fichtner, A., Zunino, A., and Gebräad, L., Hamiltonian Monte Carlo Solution of Tomographic Inverse Problems, *Geophys. J. Int.*, **216**(2):1344–1363, 2019.
3. Bui-Thanh, T. and Girolami, M., Solving Large-Scale PDE-Constrained Bayesian Inverse Problems with Riemann Manifold Hamiltonian Monte Carlo, *Inverse Probl.*, **30**(11):114014, 2014.
4. Nagel, J.B. and Sudret, B., Hamiltonian Monte Carlo and Borrowing Strength in Hierarchical Inverse Problems, *ASCE-ASME J. Risk Uncertainty Eng. Syst., Part A: Civ. Eng.*, **2**(3):B4015008, 2016.
5. Dikovskiy, M., Baltz, E.A., Von Behren, R., Geraedts, S., Kast, A., Langmore, I., Madams, T., Norgaard, P., Platt, J.C., Romero, J., and Roche, T., Multi-Instrument Bayesian Reconstruction of Plasma Shape Evolution in the C-2W Experiment, *Phys. Plasmas*, **28**(6):062503, 2021.
6. Conway, D., Simpson, J., Didana, Y., Rugari, J., and Heinson, G., Probabilistic Magnetotelluric Inversion with Adaptive Regu-Larisation Using the No-U-Turns Sampler, *Pure Appl. Geophys.*, **175**(8):2881–2894, 2018.
7. Beskos, A., Pinski, F.J., Sanz-Serna, J.-M., and Stuart, A.-M., Hybrid Monte Carlo on Hilbert Spaces, *Stochastic Process. Appl.*, **121**(10):2201–2230, 2011.
8. Au, K.X., Graham, M.M., and Thiery, A.H., Manifold Lifting: Scaling MCMC to the Vanishing Noise Regime, *Stat. Comput.*, arXiv:2003.03950, 2020.
9. Betancourt, M., Byrne, S., Livingstone, S., and Girolami, M., The Geometric Foundations of Hamiltonian Monte Carlo, *Bernoulli*, **23**(4A):2257–2298, 2017.
10. Calvetti, D. and Somersalo, E., Inverse Problems: From Regularization to Bayesian Inference, *WIREs Comput. Stat.*, **10**(3):e1427, 2018.
11. Gelman, A., Vehtari, A., Simpson, D., Margossian, D.C., Carpenter, B., Yao, Y., Kennedy, L., Gabry, J., Bürkner, P.-C., and Modrak, M., Bayesian Workflow, *Stat. Methodol.*, arXiv:2011.01808, 2020.
12. Goodfellow, I., Bengio, Y., and Courville, A., *Deep Learning*, Cambridge, MA: MIT Press, 2016.
13. Farquhar, S., Smith, L., and Gal, Y., Liberty or Depth: Deep Bayesian Neural Nets Do Not Need Complex Weight Posterior Approximations, in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, Eds., pp. 4346–4357, 2020.
14. Robert, C. and Casella, G., A Short History of Markov Chain Monte Carlo: Subjective Recollections from Incomplete Data, *Stat. Sci.*, **26**(1):102–115, 2011.
15. Betancourt, M., A Conceptual Introduction to Hamiltonian Monte Carlo, *Stat. Methodol.*, arXiv:1701.02434, 2017.
16. Vehtari, A., Gelman, A., Simpson, D., Carpenter, B., and Bürkner, P.-C., Rank-Normalization, Folding, and Localization: An Improved \hat{R} for Assessing Convergence of MCMC (with Discussion), *Bayesian Anal.*, **16**(2):667–718, 2021.
17. Lao, J., Suter, C., Langmore, I., Chimisov, C., Sax-ena, A., Sountsov, P., Moore, D., Saurous, R.A., Hoffman, M.D., and Dillon, J.V., tfp.mcmc: Modern Markov Chain Monte Carlo Tools Built for Modern Hardware, *Stat. Comput.*, arXiv:2002.01184, 2020.
18. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., and Ghemawat, S., TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, *Comput. Sci. Distrib. Paral. Cluster Comput.*, arXiv:1603.04467, 2015.
19. Frostig, R., Johnson, M.J., and Leary, C., Compiling Machine Learning Programs via High-Level Tracing, Accessed Jan. 2, 2022, from <https://mlsys.org/Conferences/2019/doc/2018/146.pdf>, 2022.
20. Duane, S., Kennedy, A.D., Pendleton, B.J., and Roweth, D., Hybrid Monte Carlo, *Phys. Lett. B*, **195**(2):216–222, 1987.
21. Carpenter, B., Gelman, A., Hoffman, M.D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P., and Riddell, A., Stan: A Probabilistic Programming Language, *J. Stat. Software*, **76**(1):1–32, 2017.
22. Salvatier, J., Wiecki, T.V., and Fonnesbeck, C., Probabilistic Programming in Python Using PyMC3, *PeerJ. Comput. Sci.*, **2**:e55, 2016.
23. Girolami, M. and Calderhead, B., Riemann Manifold Langevin and Hamiltonian Monte Carlo Methods, *J. R. Stat. Soc. B*, **73**(2):123–214, 2011.

24. Livingstone, S., Betancourt, M., Byrne, S., and Girolami, M., On the Geometric Ergodicity of Hamiltonian Monte Carlo, *Stat. Comput.*, arXiv:1601.08057, 2016.
25. Hairer, E., Lubich, C., and Wanner, G., *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*, New York: Springer, 2006.
26. Beskos, A., Pillai, N., Roberts, G., Sanz-Serna, J.-M., and Stuart, A., Optimal Tuning of the Hybrid Monte Carlo Algorithm, *Bernoulli*, **19**(5A):1501–1534, 2013.
27. Betancourt, M., Byrne, S., and Girolami, M., Optimizing the Integrator Step Size for Hamiltonian Monte Carlo, *Stat. Methodol.*, arXiv:1411.6669, 2014.
28. Langmore, I., Dikovsky, M., Geraedts, S., Norgaard, P., and Von Behren, R., A Condition Number for Hamiltonian Monte Carlo, *Stat. Comput.*, arXiv:1905.09813, 2020.
29. Horn, R.A., Horn, R.A., and Johnson, C.R., *Matrix Analysis*, Oxford, U.K.: Cambridge University Press, 1990.
30. Mangoubi, O., Pillai, N.S., and Smith, A., Does Hamiltonian Monte Carlo Mix Faster Than a Random Walk on Multimodal Densities, *Math. Probab.*, arXiv:1808.03230, 2018.
31. Dillon, J.V., Langmore, I., Tran, D., Brevdo, E., Va-Sudevan, S., Moore, D., Patton, B., Alemi, A., Hoffman, M., and Saurous, R.A., TensorFlow Distributions, *Comput. Sci. Mach. Learn.*, arXiv:1711.10604, 2017.
32. Bai, Z., Methodologies in Spectral Analysis of Large Dimensional Random Matrices: A Review, *Stat. Sinica*, **9**(3):611–662, 1999.
33. Bose, A., Gangopadhyay, S., and Sen, A., Limiting Spectral Distribution of XX' Matrices, *Ann. Inst. Henri Poincaré Probab. Stat.*, **46**(3):677–707, 2010.
34. Hoffman, M.D., The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo, *J. Mach. Learn. Res.*, **15**, 2014.
35. Andrieu, C. and Thoms, J., A Tutorial on Adaptive MCMC, *Stat. Comput.*, **18**(4):343–373, 2008.
36. Swendsen, R.H. and Wang, J.S., Replica Monte Carlo Simulation of Spin Glasses, *Phys. Rev. Lett.*, **57**(21):2607–2609, 1986.
37. Neal, R.M., Annealed Importance Sampling, *Stat. Comput.*, **11**(2):125–139, 2001.
38. Okabe, T., Kawata, M., Okamoto, Y., and Mikami, M., Replica-Exchange Monte Carlo Method for the Isobaric–Isothermal Ensemble, *Chem. Phys. Lett.*, **335**(5):435–439, 2001.
39. Syed, S., Bouchard-Cote, A., Deligiannidis, G., and Doucet, A., Non-Reversible Parallel Tempering: A Scalable Highly Parallel MCMC Scheme, *J. R. Stat. Soc. B*, **84**(2):321–350, 2022.
40. Bobkov, S. and Madiman, M., Concentration of the Information in Data with Log-Concave Distributions, *Ann. Probab.*, **39**(4):1528–1543, 2011.
41. Kofke, D.A., On the Acceptance Probability of Replica-Exchange Monte Carlo Trials, *J. Chem. Phys.*, **117**(15):6911–6914, 2002.
42. van der Sluis, A., Condition Numbers and Equilibration of Matrices, *Numer. Math.*, **14**(1):14–23, 1969.
43. Laurent, B. and Massart, P., Adaptive Estimation of a Quadratic Functional by Model Selection, *Ann. Stat.*, **28**(5):1302–1338, 2000.

APPENDIX A. SUPPORTING LEMMAS

APPENDIX A.1 Equilibration of Matrices

The equilibration of a matrix A is defined in [42] as a rescaling of the rows A_j such that $\|A_j\| \equiv 1$, in some vector norm. Since the preconditioned covariance $(D^{-1}L)(D^{-1}L)^T$ has ones on the diagonal, $D^{-1}L$ is equilibrated in the L2 norm. They go on to prove that equilibration is a near optimal diagonal scaling. In this section we follow their lead, reproving pieces from scratch in order to avoid unwinding their more general, and technical, results.

Below, $\|x\|_2$ is the L2 norm on vectors, and $\|A\|_2 := \max_{\|x\|_2=1} x \cdot Ax$ is the spectral norm on matrices. This is upper bounded by the Frobenius norm $\|A\|_F^2 = \sum_{i,j} A_{ij}^2$. In the following, A_j denotes the j th row of matrix A . A key to [42] establishing their results is the notion of monotonicity of some norms. We prove here the only case we need.

Lemma 3. Let $G \in \mathbb{R}^{N \times N}$ be diagonal with non-negative entries. For any $A \in \mathbb{R}^{N \times N}$,

$$\|AG\|_{S^4} \geq \|A\|_{S^4} \min_n \{G_{nn}\}.$$

Proof. We have

$$\|AG\|_{S^4}^4 = \text{Trace}\{(AG)(AG^T)(AG)(AG^T)\} = \text{Trace}\{G^2 A^T A G^2 A^T A\}.$$

The result then follows by repeatedly applying (to positive semidefinite U)

$$\text{Trace}\{GU\} = \sum_n G_{nn} U_{nn} \geq \min_n \{G_{nn}\} \text{Trace}\{U\}.$$

□

Lemma 4. Let $A \in \mathbb{R}^{N \times N}$ be equilibrated in the L2 norm. Suppose AA^T has K nonzero elements in every row/column. Then $\|A\|_2 \leq \sqrt{K}$.

Proof. Since $(AA^T)_{ij} = A_i \cdot A_j$, our condition on AA^T implies that, for every i , there are at most K rows A_j such that $A_i \cdot A_j \neq 0$. For these rows, Cauchy-Schwarz gives us $A_i \cdot A_j \leq 1$. Therefore,

$$\|A\|_2^4 = \|AA^T\|_2^2 \leq \|AA^T\|_F^2 = \sum_{i,j} (A_i \cdot A_j)^2 \leq \sum_{\{(i,j): A_i \cdot A_j \neq 0\}} 1 \leq K^2.$$

The result follows by taking fourth roots. □

We now have our main equilibration result.

Lemma 5. Suppose A is equilibrated in the L2 norm, with AA^T having at most K nonzero elements in every row. Then for any diagonal matrix G ,

$$\kappa(A) = \|A\|_2 \|A^{-1}\|_{S^4} \leq \sqrt{K} \|G^{-1}A\|_2 \|A^{-1}G\|_{S^4} = \sqrt{K} \kappa(G^{-1}A).$$

Proof. For any matrix F , $\|F\|_2 \geq \max_j \|F_j\|_2$. Therefore, using Lemma 4,

$$\|G^{-1}A\|_2 \geq \max_j \|(G^{-1}A)_j\|_2 = \max_j |G_{jj}^{-1}| = \max_j |G_{jj}^{-1}| \frac{\|A\|_2}{\sqrt{K}}.$$

Likewise, Lemma 3 tells us $\|A^{-1}G\|_{S^4} \geq \|A^{-1}\|_{S^4} \min_j |G_{jj}|$. Therefore,

$$\|G^{-1}A\|_2 \|A^{-1}G\|_{S^4} \geq \max_j |G_{jj}^{-1}| \frac{\|A\|_2}{\sqrt{K}} \|A\|_{S^4} \min_j |G_{jj}| = \frac{\|A\|_2}{\sqrt{K}} \|A\|_{S^4}.$$

Rearranging, we have proved Lemma 5. □

APPENDIX A.2 Chi-Square Bounds

Lemma 6. If $Z_s \sim \mathcal{N}(0, 1)$ are i.i.d., then for $\varepsilon \in (0, 1)$,

$$P \left[\left| \frac{1}{S} \sum_{s=1}^S Z_s^2 - 1 - \left(\frac{1}{S} \sum_{s=1}^S Z_s \right)^2 \right| \geq \varepsilon \right] \leq 3 \exp \left\{ \frac{-S\varepsilon^2}{25} \right\}.$$

Proof. In the corollary to Lemma 1 in [43], they establish for χ_S^2 a chi-square random variable with S degrees of freedom, and $c > 0$:

$$\begin{aligned} \mathbb{P}[\chi_S^2 - S \geq 2\sqrt{St} + 2t] &\leq e^{-t}, \\ \mathbb{P}[\chi_S^2 - S \leq -2\sqrt{St}] &\leq e^{-t}. \end{aligned} \quad (\text{A.1})$$

The second inequality in Eq. (A.1) directly gives us, if $\varepsilon/2 = 2\sqrt{c/S}$,

$$\mathbb{P}\left[\frac{1}{S} \sum_{s=1}^S Z_s^2 - 1 \leq -\frac{\varepsilon}{2}\right] = \mathbb{P}\left[\chi_S^2 - S \leq -2\sqrt{Sc}\right] \leq e^{-c} = e^{-S\varepsilon^2/16}.$$

To use the first inequality in (A.1), we start with the substitution $\varepsilon/2 = (5/2)\sqrt{c/S}$, which implies $\sqrt{c/S} = \varepsilon/5 < 1/4$, so that $(5/4) \geq 1 + \sqrt{c/S}$:

$$\begin{aligned} \mathbb{P}\left[\frac{1}{S} \sum_{s=1}^S Z_s^2 - 1 \geq \frac{\varepsilon}{2}\right] &= \mathbb{P}\left[\frac{\chi_S^2 - S}{S} \geq 2\frac{5}{4}\sqrt{\frac{c}{S}}\right] \\ &\leq \mathbb{P}\left[\frac{\chi_S^2 - S}{S} \geq 2\left(1 + \sqrt{\frac{c}{S}}\right)\sqrt{\frac{c}{S}}\right] \\ &= \mathbb{P}\left[\chi_S^2 - S \geq 2\sqrt{Sc} + 2c\right] \\ &\leq e^{-c} \\ &= e^{-S\varepsilon^2/25}. \end{aligned}$$

Combining, we have

$$\mathbb{P}\left[\left|\frac{1}{S} \sum_{s=1}^S Z_s^2 - 1\right| \geq \frac{\varepsilon}{2}\right] \leq 2e^{-S\varepsilon^2/25}. \quad (\text{A.2})$$

Next, a Bernstein inequality gives the standard Normal tail bound

$$\mathbb{P}\left[\left(\frac{1}{S} \sum_{s=1}^S Z_s\right)^2 \geq \frac{\varepsilon}{2}\right] \leq e^{-S\varepsilon/4}. \quad (\text{A.3})$$

Combining Eqs. (A.2) and (A.3), we have the result. \square