Received: 5 May 2020 | Accepted: 14 October 2021

DOI: 10.1111/rssb.12482

ORIGINAL ARTICLE

The Barker proposal: Combining robustness and efficiency in gradient-based MCMC

Samuel Livingstone[1] | Giacomo Zanella[2]

[1] Department of Statistical Science, University College London, UK

[2] Department of Decision Sciences, BIDSA and IGIER, Bocconi University, Milan, Italy

Correspondence
Giacomo Zanella, Department of Decision Sciences, BIDSA and IGIER, Bocconi University, Via Roentgen 1, Milan, Italy.
Email: giacomo.zanella@unibocconi.it

Funding information
Engineering and Physical Sciences Research Council, Grant/Award Number: EP/K014463/1; European Research Council, Grant/Award Number: 306406 (N-BNP)

Abstract

There is a tension between robustness and efficiency when designing Markov chain Monte Carlo (MCMC) sampling algorithms. Here we focus on robustness with respect to tuning parameters, showing that more sophisticated algorithms tend to be more sensitive to the choice of step-size parameter and less robust to heterogeneity of the distribution of interest. We characterise this phenomenon by studying the behaviour of spectral gaps as an increasingly poor step-size is chosen for the algorithm. Motivated by these considerations, we propose a novel and simple gradient-based MCMC algorithm, inspired by the classical Barker accept-reject rule, with improved robustness properties. Extensive theoretical results, dealing with robustness to tuning, geometric ergodicity and scaling with dimension, suggest that the novel scheme combines the robustness of simple schemes with the efficiency of gradient-based ones. We show numerically that this type of robustness is particularly beneficial in the context of adaptive MCMC, giving examples where our proposed scheme significantly outperforms state-of-the-art alternatives.

KEYWORDS

adaptive tuning, Bayesian computation, MCMC, Metropolis–Hastings, spectral gap

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2022 The Authors. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* published by John Wiley & Sons Ltd on behalf of Royal Statistical Society.

1 | INTRODUCTION

The need to compute high-dimensional integrals is ubiquitous in modern statistical inference and beyond (e.g. Brooks et al., 2011; Krauth, 2006; Stuart, 2010). Markov chain Monte Carlo (MCMC) is a popular solution, in which the central idea is to construct a Markov chain with a certain limiting distribution and use ergodic averages to approximate expectations of interest. In the celebrated Metropolis–Hastings algorithm, the Markov chain transition is constructed using a combination of a 'candidate' kernel, to suggest a possible move at each iteration, together with an accept-reject mechanism (Hastings, 1970; Metropolis et al., 1953). Many different flavours of Metropolis–Hastings exist, with the most common difference being in the construction of the candidate kernel. In the Random walk Metropolis (RWM), proposed moves are generated using a symmetric distribution centred at the current point. Two more sophisticated methods are the Metropolis-adjusted Langevin algorithm (Roberts & Tweedie, 1996) and Hamiltonian/hybrid Monte Carlo (Duane et al., 1987; Neal, 2011). Both use gradient information about the distribution of interest (the *target*) to inform proposals. Gradient-based methods are widely considered to be state-of-the-art in MCMC, and much current work has been dedicated to their study and implementation (e.g. Beskos et al., 2013; Dalalyan, 2017; Durmus & Moulines, 2017).

Several measures of performance have been developed to help choose a suitable candidate kernel for a given task. One of these is high-dimensional scaling arguments, which compare how the efficiency of the method decays with d, the dimension of the state space. For the random walk algorithm this decay is of the order d^{-1} (Roberts et al., 1997), while for the Langevin algorithm the same figure is $d^{-1/3}$ (Roberts & Rosenthal, 1998) and for Hamiltonian Monte Carlo (HMC) it is $d^{-1/4}$ (Beskos et al., 2013). Another measure is to find general conditions under which a kernel will produce a geometrically ergodic Markov chain. For the random walk algorithm, this essentially occurs when the tails of the posterior decay at a faster than exponential rate and are suitably regular (more precise conditions are given in Jarner & Hansen, 2000). The same is broadly true of the Langevin and Hamiltonian schemes (Durmus et al., 2017a; Livingstone et al., 2019; Roberts & Tweedie, 1996), but here there is an additional restriction that the tails should not decay too quickly. This limitation is caused by the way in which gradients are used to construct the candidate kernel, which can result in the algorithm generating unreasonable proposals that are nearly always rejected in certain regions (Livingstone et al., 2019; Roberts & Tweedie, 1996).

There is clearly some tension between the different results presented above. According to the scaling arguments, gradient information is preferable. The ergodicity results, however, imply that gradient-based schemes are typically less *robust* than others, in the sense that there is a smaller class of limiting distributions for which the output will be a geometrically ergodic Markov chain. It is natural to wonder whether it is possible to incorporate gradient information in such a way that this measure of robustness is not compromised. Simple approaches to modifying the Langevin algorithm for this purpose have been suggested (based on the idea of truncating gradients, for example Atchade, 2006; Roberts & Tweedie, 1996), but these typically compromise the favourable scaling of the original method. In addition to this, it is often remarked that gradient-based methods can be difficult to tune. Algorithm performance is often highly sensitive to the choice of scale within the proposal (Neal, 2003, Figure 15), and if this is chosen to be too large in certain directions then performance can degrade rapidly. Because of this, practitioners must spend a long time adjusting the tuning parameters to ensure that the algorithm is running well, or develop sophisticated adaptation schemes for this purpose (e.g. Hoffman & Gelman, 2014), which can nonetheless still require a large number of iterations to find good tuning parameters (see Sections 5 and 6). We will refer to this issue as *robustness to tuning*.

In this article, we present a new gradient-based MCMC scheme, *the Barker proposal*, which combines favourable high-dimensional scaling properties with favourable ergodicity and robustness to tuning properties. To motivate the new scheme, in Section 2, we present a direct argument showing how the spectral gaps for the random walk, Langevin and Hamiltonian algorithms behave as the tuning parameters are chosen to be increasingly unsuitable for the problem at hand. In particular, we show that the spectral gaps for commonly used gradient-based algorithms decay to zero exponentially fast in the degree of mismatch between the scales of the proposal and target distributions, while for the random walk Metropolis the decay is polynomial. In Section 3, we derive the Barker proposal scheme beginning from a family of π-invariant continuous-time jump processes, and discuss its connections to the concept of 'locally balanced' proposals, introduced in (Zanella, 2020) for discrete state spaces. The name *Barker* comes from the particular choice of 'balancing function' used to uncover the scheme, which is inspired by the classical Barker accept-reject rule (Barker, 1965). In Section 4, we conduct a detailed analysis of the ergodicity, scaling and robustness properties of this new method, establishing that it shares the favourable robustness to tuning of the random walk algorithm, can be geometrically ergodic in the presence of very light tails, and enjoys the $d^{-1/3}$ scaling with dimension of the Langevin scheme. The theory is then supported by an extensive simulation study in Sections 5 and 6, including comparisons with state-of-the-art alternative sampling schemes, which highlights that this kind of robustness is particularly advantageous in the context of adaptive MCMC. The code to reproduce the experiments is available from the online repository at the link https://github.com/gzanella/barker. Proofs and further numerical simulations are provided in the supplement.

1.1 | Basic set-up and notation

Throughout we work on the Borel space $(\mathbb{R}^d, \mathcal{B})$, with $d \geq 1$ indicating the dimension. For $\lambda \in \mathbb{R}$, we write $\lambda \uparrow \infty$ and $\lambda \downarrow 0$ to emphasize the direction of convergence when this is important. For two functions $f, g : \mathbb{R} \to \mathbb{R}$, we use the Bachmann–Landau notation $f(t) = \mathcal{O}(g(t))$ if $\limsup_{t\to\infty} f(t)/g(t) < \infty$ and $f(t) = \Theta(g(t))$ if both $\liminf_{t\to\infty} f(t)/g(t) > 0$ and $f(t) = \mathcal{O}(g(t))$.

The Markov chains we consider will be of the Metropolis–Hastings type, meaning that the π-invariant kernel P is constructed as $P(x, dy) := \alpha(x,y)Q(x,dy) + r(x)\delta_x(dy)$, where $Q : \mathbb{R}^d \times \mathcal{B} \to [0,1]$ is a candidate kernel,

$$\alpha(x,y) := \min\left(1, \frac{\pi(dy)Q(y,dx)}{\pi(dx)Q(x,dy)}\right) \tag{1}$$

is the *acceptance rate* for a proposal y given the current point x (provided that the expression is well-defined, see Tierney, 1998 for details here), and $r(x) := 1 - \int \alpha(x,y)Q(x,dy)$ is the average probability of rejection given that the current point is x.

2 | ROBUSTNESS TO TUNING

In this section, we seek to quantify the robustness of the random walk, Langevin and Hamiltonian schemes with respect to the mismatch between the scales of $\pi(\cdot)$ and Q in a given direction.

Unlike other analyses in the MCMC literature (e.g. Beskos et al., 2018; Roberts & Rosenthal, 2001), we are interested in studying how MCMC algorithms perform when they are *not* optimally tuned, in order to understand how crucially performance depends on such design choices (e.g. the choice of proposal step-size or pre-conditioning matrix). The rationale for performing such an analysis is that achieving optimal or even close to optimal tuning can be extremely challenging in practice, especially when $\pi(\cdot)$ exhibits substantial heterogeneity. This is typically done using past samples in the chain to compute online estimates of the average acceptance rate and the covariance of π (or simply its diagonal terms for computational convenience), and then using those estimates to tune the proposal step-sizes in different directions (Andrieu & Thoms, 2008). If the degree of heterogeneity is large, it can take a long time for certain directions to be well-explored, and hence for the estimated covariance to be representative and the tuning parameters to converge.

In such settings, algorithms that are more robust to tuning are not only easier to use when such tuning is done manually by the user, but can also greatly facilitate the process of learning the tuning parameters *adaptively* within the algorithm. We show in Sections 5 and 6 that if an algorithm is robust to tuning then this adaptation process can be orders of magnitude faster than in the alternative case, drastically reducing the overall computational cost for challenging targets. The intuition for this is that more robust algorithms will start performing well (i.e. sampling efficiently) earlier in the adaptation process (when tuning parameters are not yet optimally tuned), which in turn will speed up the exploration of the target and the learning of the tuning parameters.

2.1 | Analytical framework

The most general scenario we consider is a family of target densities $\pi^{(\lambda,k)}$ indexed by $\lambda > 0$ and $k \in \{1, \dots, d\}$ defined as

$$\pi^{(\lambda,k)}(x) := \lambda^{-k}\pi(x_1/\lambda, \dots, x_k/\lambda, x_{k+1}, \dots, x_d), \qquad x = (x_1, \dots, x_d) \in \mathbb{R}^d, \tag{2}$$

where π is a density defined on $\mathbb{R}^d$ for which $\pi(x) > 0$ for all $x \in \mathbb{R}^d$ and $\log \pi \in C^1(\mathbb{R}^d)$. The set-up allows modification of the scale of the first k components of $\pi^{(\lambda,k)}$ through the parameter λ. Our main results are presented for the case $k = 1$, and we write $\pi^{(\lambda)} := \pi^{(\lambda,1)}$ for simplicity, before discussing extensions to the $k > 1$ setting in Section 2.5. We consider targeting $\pi^{(\lambda)}$ using a Metropolis–Hastings algorithm with fixed tuning parameters, and study performance as λ varies. Intuitively, we can think of λ as a parameter quantifying the level of heterogeneity in the problem. As a concrete example, consider a RWM algorithm in which given the current state $x^{(t)}$ the candidate move is $y = x^{(t)} + \sigma\xi$, with $\sigma > 0$ a fixed tuning parameter and $\xi \sim N(0, \mathbb{I}_d)$, where $\mathbb{I}_d$ is the $d \times d$ identity matrix. It is instructive to take σ as the optimal choice of global scale for π, meaning when λ is far from one then σ is no longer a suitable choice for the first coordinate of $\pi^{(\lambda)}$.

In the context of the above, the $\lambda \downarrow 0$ regime is representative of distributions in which one component (in this case the first) has a very small scale compared to all others. Conversely, the $\lambda \uparrow \infty$ regime reflects the case in which one component has a much larger scale than its counterparts. Studying robustness to tuning in the context of heterogeneity is particularly relevant, as highlighted above, as this is exactly the context in which tuning is more challenging. The $\lambda \downarrow 0$ regime is particularly interesting and has been recently considered in Beskos et al. (2018), where the authors study the behaviour of the RWM for 'ridged' densities for different values of k using a diffusion limit approach. The focus in that work, however, was on the finding optimal tuning parameters for the algorithm as a function of λ, whereas the present paper is concerned with the regime in which the tuning parameters are fixed (as discussed above).

The above framework could be equivalently formulated by keeping the target distribution π fixed and instead rescaling the first component of the candidate kernel by a factor $1/\lambda$. This is indeed the formulation we mostly use in the proofs of our theoretical results. A proof of the mathematical equivalence between the two formulations can be found in the supplement.

2.2 | Measure of performance

Our measure of performance for the various algorithms will be the spectral gap of the resulting Markov chains. Consider the space of functions

$$L^2_{0,1}(\pi) = \{f : \mathbb{R}^d \to \mathbb{R} | \mathbb{E}_\pi[f] = 0,\ Var_\pi[f] = 1\}.$$

Note that any function g with $\mathbb{E}_\pi g^2 < \infty$ can be associated with an $f \in L^2_{0,1}(\pi)$ through the map $f = (g - \mathbb{E}_\pi g)/\sqrt{Var_\pi g}$, and that if $X^{(t)} \sim \pi(\cdot)$ and $X^{(t+1)}|X^{(t)} \sim P(X^{(t)}, \cdot)$ then $\mathrm{Corr}\{g(X^{(t)}), g(X^{(t+1)})\} = \mathrm{Corr}\{f(X^{(t)}), f(X^{(t+1)})\}$. The (right) spectral gap of a π-reversible Markov chain with transition kernel P is

$$\mathrm{Gap}(P) = \inf_{f \in L^2_{0,1}(\pi)} \frac{1}{2} \int (f(y) - f(x))^2 \pi(dx) P(x, dy). \tag{3}$$

The expression inside the infimum is called a *Dirichlet form*, and can be thought of as the 'expected squared jump distance' for the function f provided the chain is stationary. This can in turn be re-written as $1 - \mathrm{Corr}\{f(X^{(t)}), f(X^{(t+1)})\}$. Maximising the spectral gap of a reversible Markov chain can therefore be understood as minimising the *worst-case* first-order autocorrelation among all possible square-integrable test functions.

The spectral gap allows to bound the variances of ergodic averages (see Proposition 1 of Rosenthal, 2003). Also, a direct connection between the spectral gap and mixing properties of the chain can be made if the operator $Pf(x) := \int f(y)P(x,dy)$ is positive on $L^2(\pi)$. This will always be the case if the chain is made lazy, which is the approach taken in Woodard et al. (2009), and the same adjustment can be made here if desired.

2.3 | The small λ regime

In this section, we assess the robustness to tuning of the random walk, Langevin and Hamiltonian schemes as $\lambda \downarrow 0$. This corresponds to the case in which the proposal scale is chosen to be too large in the first component of $\pi^{(\lambda)}$. The results in this section will support the idea that classical gradient-based schemes pay a very high price for any direction in which this tuning parameter is chosen to be too large, as already noted in the literature (e.g. Neal, 2003, p. 738), while the RWM is less severely affected by such issues.

2.3.1 | Random walk Metropolis

In the RWM, given a current point $x \in \mathbb{R}^d$, a proposal y is calculated using the equation

$$y = x + \sigma\xi, \tag{4}$$

with $\sigma > 0$ and $\xi \sim \mu(\cdot)$ for some centred symmetric distribution μ. The resulting candidate kernel Q^R is given by $Q^R(x, dy) = q^R(x, y)dy$ with $q^R(x, y) = \sigma^{-d}\mu((y - x)/\sigma)$, where $\mu(\xi)$ for $\xi \in \mathbb{R}^d$ denotes the density of μ. Following Section 2.1, we consider Metropolis–Hastings algorithms with proposal Q^R and target distribution $\pi^{(\lambda)}$ defined in Equation (2), and denote the resulting transition kernels as P^R_λ.

We impose the following mild regularity conditions on the density $\mu(\xi)$. These are satisfied for most popular choices of μ, as shown in the subsequent proposition.

Condition 1 There exists $\lambda_0 > 0$ such that for any $x, y \in \mathbb{R}^d$ and $\lambda < \lambda_0$ we have $\mu(\delta_\lambda) \geq \mu(\delta)$, where $\delta = y - x$ and

$$\delta_\lambda := (\lambda(y_1 - x_1), y_2 - x_2, \ldots, y_d - x_d). \tag{5}$$

In addition, $\sup_{\xi_1 \in \mathbb{R}} \mu_1(\xi_1) < \infty$, where $\mu_1(\xi_1) = \int_{\mathbb{R}^{d-1}} \mu(\xi_1, \xi_2, \ldots, \xi_d)d\xi_2 \ldots d\xi_d$ is the marginal distribution of ξ_1 under $\xi \sim \mu$.

Proposition 1 *Denoting the usual p-norm as* $\|x\|_p = \left(\sum_{i=1}^d x_i^p\right)^{1/p}$, *Condition* 1 *holds in each of the below cases:*

(i) $q^R(x, y) = (2\pi\sigma^2)^{-d/2} \exp(-\|x - y\|_2^2/(2\sigma^2))$ *(Gaussian)*
(ii) $q^R(x, y) = 2^{-d} \exp(-\|x - y\|_1)$ *(Laplace)*
(iii) $q^R(x, y) \propto (1 + \|y - x\|_2^2/\nu)^{-(\nu+d)/2}$ *for* $\nu \in \{1, 2, \ldots\}$ *(Student's t)*

We conclude the section with a characterization of the rate of convergence to zero of the spectral gap for the RWM as $\lambda \downarrow 0$.

Theorem 1 *Assume Condition* 1 *and* $\text{Gap}(P^R_1) > 0$. *Then it holds that*

$$\text{Gap}(P^R_\lambda) = \Theta(\lambda), \quad \text{as } \lambda \downarrow 0.$$

Note that Theorem 1 requires very few assumptions on the target π other than $\text{Gap}(P^R_1) > 0$. Note also that the lower bound is of the form $\text{Gap}(P^R_\lambda) \geq \lambda\text{Gap}(P^R_1)$, see proof of Theorem 1 for details. No dependence on the dimension of the problem other than that intrinsic to $\text{Gap}(P^R_1)$ is therefore introduced.

2.3.2 | The Langevin algorithm

In the Langevin algorithm (or more specifically the Metropolis-adjusted Langevin algorithm, MALA), given the current point $x \in \mathbb{R}^d$, a proposal y is generated by setting

$$y = x + \frac{\sigma^2}{2}\nabla \log \pi^{(\lambda)}(x) + \sigma\xi, \tag{6}$$

for some $\sigma > 0$ and $\xi \sim N(0, \mathbb{I}_d)$. In this case the proposal is no longer symmetric and so the full Hastings ratio (1) must be used. The proposal mechanism is based on the overdamped Langevin stochastic differential equation $dX_t = \nabla \log \pi^{(\lambda)}(X_t)dt + \sqrt{2}dB_t$. We write Q_λ^M for the corresponding candidate distribution and P_λ^M for the Metropolis–Hastings kernel with proposal Q_λ^M and target $\pi^{(\lambda)}$.

We present results for the Langevin algorithm in two settings. Initially, we consider more restrictive conditions under which our upper bound on the spectral gap depends on the tail behaviour of π in a particularly explicit manner, and then give a broader result.

Condition 2 Assume the following:

(i) π has a density of the form $\pi(x) = \pi_1(x_1)\pi_{2:n}(x_2, \dots, x_d)$, for some densities π_1 and $\pi_{2:n}$ on $\mathbb{R}$ and $\mathbb{R}^{d-1}$, respectively.

(ii) For some $q \in [0, 1)$, it holds that

$$\left|\frac{d}{dx_1} \log \pi_1(x_1)\right| = \Theta(|x_1|^q) \quad \text{as } |x_1| \uparrow \infty. \tag{7}$$

Theorem 2 *If Condition* 2 *holds, then there is a* $\gamma > 0$ *such that*

$$\text{Gap}(P_\lambda^M) = \mathcal{O}(e^{-\gamma\lambda^{-(1+q)} + q \log(\lambda)}) \quad as \ \lambda \downarrow 0.$$

When compared with the random walk algorithm, Theorem 2 shows that the Langevin scheme is much less robust to heterogeneity. Indeed, the spectral gap decays *exponentially fast* in $\lambda^{-(1+q)}$, meaning that even small errors in the choice of step-size can have a large impact on algorithm efficiency, and so practitioners must invest considerable effort tuning the algorithm for good performance, as shown through simulations in Sections 5 and 6. Theorem 2 also illustrates that the Langevin algorithm is more sensitive to λ when the tails of $\pi(\cdot)$ are lighter. This is intuitive, as in this setting gradient terms can become very large in certain regions of the state space.

Remark 1 Theorem 2 (and Theorem 4 below) could be extended to the case $q \geq 1$ in Equation (7); however, in these cases, samplers typically fail to be geometrically ergodic when λ is small (Livingstone et al., 2019; Roberts & Tweedie, 1996) meaning the spectral gap is typically 0 and the theorem becomes trivial.

Remark 2 Condition 2(ii) could be replaced with the simpler requirement that $|\nabla \log \pi_1(x_1)| \uparrow \infty$, with the corresponding bound $\text{Gap}(P_\lambda^M) = \mathcal{O}(e^{-1/\lambda})$.

A different set of conditions, which hold much more generally, and corresponding upper bound are presented below.

Condition 3 Assume the following:

(i) There is a $\gamma > 0$ such that

$$\liminf_{|x_1|\to\infty} \left(\inf_{(x_2,\dots,x_d)\in\mathbb{R}^{d-1}} \left|\frac{\partial \log \pi(x)}{\partial x_1}\right| \|x\|_2^\gamma \right) > 0, \tag{8}$$

(ii) Given $X \sim \pi$ there is a $\beta > 0$ such that

$$\mathbb{P}(\|X\|_2 > t) = \mathcal{O}(e^{-t^\beta}) \qquad \text{as } t \to \infty. \tag{9}$$

Theorem 3 *If Condition* 3 *holds, then*

$$\mathrm{Gap}(P_\lambda^M) = \mathcal{O}(e^{-\lambda^{-\alpha}}) \qquad \textit{as } \lambda \downarrow 0,$$

for some $\alpha > 0$, *which can be taken as* $\alpha = \min\{\beta/2, \beta/\gamma, 2/3\}$.

We expect Condition 3 to be satisfied in many commonly encountered scenarios, with the exception of particularly heavy-tailed models. In the exponential family class $\pi(x) \propto \exp\{-\alpha \|x\|_2^\beta\}$, for example, Condition 3 holds for any α and $\beta > 0$ (see proof in the supplement).

2.3.3 | Hamiltonian Monte Carlo

In Hamiltonian Monte Carlo, we write the current point $x \in \mathbb{R}^d$ as $x(0)$, and construct the proposal $y := x(L)$ for some prescribed integer L using the update

$$x(L) = x(0) + \sigma^2 \left(\frac{L}{2} \nabla \log \pi^{(\lambda)}(x(0)) + \sum_{j=1}^{L-1} (L-j) \nabla \log \pi^{(\lambda)}(x(j)) \right) + L\sigma\xi(0), \tag{10}$$

where each $x(j)$ is defined recursively in the same manner, and $\xi(0) \sim N(0, \mathbb{I}_d)$. The transition is based on numerically solving Hamilton's equations for the Hamiltonian system $H(x,\xi) = -\log \pi^{(\lambda)}(x) + \xi^T\xi/2$ for $L\sigma$ units of time. The decision of whether or not the proposal is accepted is taken using the acceptance probability $\min(1, \pi^{(\lambda)}(y)/\pi^{(\lambda)}(x) \times e^{-\xi(L)^T\xi(L)/2 + \xi(0)^T\xi(0)/2})$, where

$$\xi(L) = \xi(0) + \frac{\sigma}{2} \left(\nabla \log \pi^{(\lambda)}(x(0)) + \nabla \log \pi^{(\lambda)}(x(L)) \right) + \sigma \sum_{j=1}^{L-1} \nabla \log \pi^{(\lambda)}(x(j)).$$

A more detailed description is given in Neal (2011). We write P_λ^H for the corresponding Metropolis–Hastings kernel with proposal mechanism as above and target $\pi^{(\lambda)}$. Here we present a heterogeneity result under Condition 2 of the previous subsection.

Theorem 4 *If Condition* 2 *holds, then there is a* $\gamma > 0$ *such that*

$$\mathrm{Gap}(P_\lambda^H) = \mathcal{O}\left(e^{-\gamma \lambda^{-(1+q)} + q \log(\lambda)} \right) \quad \textit{as } \lambda \downarrow 0.$$

It is no surprise that Theorem 4 is comparable to Theorem 2, since setting $L = 1$ equates the Langevin and Hamiltonian methods.

2.4 | The large λ regime

In this section, we briefly discuss the $\lambda \uparrow \infty$ regime, where σ is chosen to be too small for the first component of $\pi^{(\lambda)}$, arguing that all samplers under consideration behave similarly in this regime and pay a similar price for too small tuning parameters in a given direction. The intuition for this is that as $\lambda \uparrow \infty$ the gradient-based proposal mechanisms discussed here all tend towards that of the random walk sampler in the first coordinate. For example, if we consider one-dimensional models, for any $x \in \mathbb{R}$ we can write $\nabla \log \pi^{(\lambda)}(x) = \lambda^{-1} \nabla \log \pi(x/\lambda)$, meaning as $\lambda \uparrow \infty$ the amount of gradient information in the proposal is reduced provided π is suitably regular. The following result makes this intuition precise. To avoid repetitions, we state here the result for both the Langevin and the Barker proposal that we will introduce in the next section.

Proposition 2 *Fix $x \in \mathbb{R}$ and let the density π be such that $\nabla \log \pi$ is bounded in some neighbourhood of zero. Then the Langevin and Barker candidate kernels Q_λ^M and Q_λ^B, defined in Equations* (6) *and* (16) *respectively, both satisfy*

$$\|Q_\lambda^{M/B}(x, \cdot) - Q^R(x, \cdot)\|_{TV} = \mathcal{O}(1/\lambda),$$

where Q^R is the (Gaussian) random walk candidate kernel.

The same intuition applies to the Hamiltonian case provided L is fixed, since each gradient term in the proposal is also $\Theta(1/\lambda)$. While there are many well-known measures of distance between two distributions, we argue that total variation is an appropriate choice here, since it has an explicit focus on how much the two kernels overlap and is invariant under bijective transformations of the state space (including re-scaling coordinates).

While the above statements provide useful heuristic arguments, in order to obtain more rigorous results one should prove that the spectral gaps decay to 0 at the same rate as $\lambda \uparrow \infty$, which we leave to future work. We note, however, that the conjecture that the algorithms behave similarly for large values of λ is supported by the simulations of Section 5.1.

2.5 | Extensions

The lower bound of Theorem 1 extends naturally to the $k > 1$ setting, becoming instead $\Theta(\lambda^k)$, and so the rate of decay remains polynomial in λ for any k. Analogously, we expect the corresponding upper bound for gradient-based schemes to remain exponential and become $\mathcal{O}(e^{-k(\gamma\lambda^{-(1+q)}+q\ \log(\lambda))})$, although the details of this are left for future work. We explore examples of this nature through simulations in Section 5 and find empirically that the single component case is informative also of more general cases. Further extensions in which a different λ_i is chosen in each of the k directions can also be considered, with each $\lambda_i \downarrow 0$ at a different rate. We conjecture that in this setting the λ_i that decays most rapidly will dictate the behaviour of spectral gaps, though such an analysis is beyond the scope of the present work. One could consider using a mixture of the MALA/HMC and random walk kernels in an attempt to achieve both robustness to tuning and favourable scaling properties. While this may seem promising in theory, in practice we believe that it would be difficult to achieve *both* robustness to tuning *and* favourable high-dimensional performance from such an approach. In the next section, we consider a scheme for which the two goals can be achieved simultaneously.

3 | COMBINING ROBUSTNESS AND EFFICIENCY

The results of Section 2 show that the two gradient-based samplers considered there are much less robust to heterogeneity than the random walk algorithm. In this section, we introduce a novel and simple to implement gradient-based scheme that shares the superior scaling properties of the Langevin and Hamiltonian schemes, but also retains the robustness of the random walk sampler, both in terms of geometric ergodicity and robustness to tuning.

3.1 | Locally balanced Metropolis–Hastings

Consider a continuous-time Markov jump process on $\mathbb{R}^d$ with associated generator

$$\mathcal{L}f(x) = \int [f(y) - f(x)]g\left(\frac{\pi(y)q(y,x)}{\pi(x)q(x,y)}\right) Q(x, dy), \tag{11}$$

for some suitable function $f : \mathbb{R}^d \to \mathbb{R}$, where $\pi(x)$ is a probability density, $Q(x, dy) := q(x, y)dy$ is a transition kernel and the *balancing* function $g : (0, \infty) \to (0,\infty)$ satisfies

$$g(t) = tg(1/t). \tag{12}$$

A discrete state-space version of this process with symmetric Q was introduced in Power and Goldman (2019). The dynamics of the process are such that if the current state $X_t = x$, the next jump will be determined by a Poisson process with intensity

$$Z(x) := \int g\left(\frac{\pi(y)q(y,x)}{\pi(x)q(x,y)}\right) Q(x, dy), \tag{13}$$

and the next state is drawn from the kernel

$$Q^{(g)}(x, dy) := Z(x)^{-1}g\left(\frac{\pi(y)q(y,x)}{\pi(x)q(x,y)}\right) Q(x, dy).$$

It is straightforward to show that $\mathcal{L}$ is a self-adjoint operator on the Hilbert space $L^2(\pi)$ using Equation (12), implying that the process is π-reversible and can therefore serve as a starting point for designing MCMC algorithms.

In the ‘locally balanced’ framework for discrete state-space Metropolis–Hastings introduced in Zanella (2020), candidate kernels are of the form

$$\tilde{Q}(x, dy) = \tilde{Z}(x)^{-1}g\left(\frac{\pi(y)}{\pi(x)}\right) \mu_\sigma(y - x)dy, \tag{14}$$

meaning the *embedded Markov chain* of Equation (11) with the choice $Q(x, dy) := \mu_\sigma(y - x)dy$, where $\mu_\sigma(y - x) := \sigma^{-d}\mu((y - x)/\sigma)$ for some symmetric density μ. It is well-known that the invariant density of the embedded chain does not coincide with that of the process when jumps are not of constant intensity, in this case becoming proportional to $Z(x)\pi(x)$, as shown in Zanella (2020). As a result a Metropolis–Hastings step is employed to correct for the discrepancy. In Power and Goldman (2019) it is suggested that as an alternative the jump process can be simulated exactly.

The challenge with employing either of these strategies on a continuous state space is that the integral (13) will typically be intractable. To overcome this issue we take two steps, and for simplicity, we first describe these on $\mathbb{R}$ (there are two options on $\mathbb{R}^d$ for $d > 1$, which are discussed in Section 3.3). The first step is to consider a first-order Taylor series expansion of $\log \pi$ within g (again with a symmetric choice of Q), leading to the family of processes with generator

$$Lf(x) = \int [f(y) - f(x)]g(e^{\nabla \log \pi(x)(y-x)})\mu_\sigma(y - x)dy.$$

We refer to candidate kernels in Metropolis–Hastings algorithms that are constructed using the embedded Markov chain of this new process as *first-order* locally balanced proposals, taking the form

$$Q^{(g)}(x, dy) = Z(x)^{-1} g(e^{\nabla \log \pi(x)(y-x)})\mu_\sigma(y - x)dy, \tag{15}$$

where $Z(x) := \int g(e^{\nabla \log \pi(x)(y-x)})\mu_\sigma(y - x)dy$.

Remark 3 One can also think at Equation (12) as a requirement to ensure that the proposals in Equation (15) are exact (i.e. π-reversible) at the first order. In particular, in the supplement, it is shown that a proposal $Q^{(g)}$ defined as in Equation (15) is π-reversible for π log-linear if and only if Equation (12) holds.

The second step is to note that, if particular choices of g are made, then $Z(x)$ becomes tractable. In fact, if the balancing function $g(t) = \sqrt{t}$ and a Gaussian kernel μ_σ are chosen, then the result is the Langevin proposal

$$Q^M(x, dy) \propto e^{\nabla \log \pi(x)(y-x)/2}\mu_\sigma(y - x)dy.$$

Thus, MALA can be viewed as a particular instance of this class. Other choices of g are, however, also possible, and give rise to different gradient-based algorithms. In the next section we explore what a sensible choice of g might look like.

3.2 | The Barker proposal on $\mathbb{R}$

The requirement for the balancing function g to satisfy $g(t) = tg(1/t)$ is in fact also imposed on the acceptance rate of a Metropolis–Hastings algorithm to produce a π-reversible Markov chain. Indeed, setting $t := \pi(y)q(y, x)/(\pi(x)q(x, y))$ and assuming $\alpha(x, y) := \alpha(t)$, then the detailed balance equations can be written $\alpha(t) = t\alpha(1/t)$. Possible choices of g can therefore be found by considering suggestions for α in the literature. One choice proposed in Barker (1965) is

$$g(t) = \frac{t}{1 + t}.$$

The work of Peskun (1973) and Tierney (1998) showed that this choice of α is inferior to the more familiar Metropolis–Hasting rule $\alpha(t) = \min(1, t)$ in terms of asymptotic variance. The same conclusion cannot, however, be drawn when considering the choice of balancing function g.

In fact, the choice $g(t) = t/(1+t)$ was shown to minimize asymptotic variances of Markov chain estimators in some discrete settings in Zanella (2020). In addition, as shown below, this particular choice of g leads to a fully tractable candidate kernel that can be easily sampled from. For this reason, we focus on this choice of g here, and leave the question of optimality in general for future work.

Proposition 3 *If* $g(t) = t\,/\,(1 + t)$, *then the normalising constant* $Z(x)$ *in Equation* (15) *is* $1/2$.

The resulting proposal distribution is

$$Q^B(x, dy) = 2\frac{\mu_\sigma(y - x)}{1 + e^{-\nabla \log \pi(x)(y-x)}}dy. \tag{16}$$

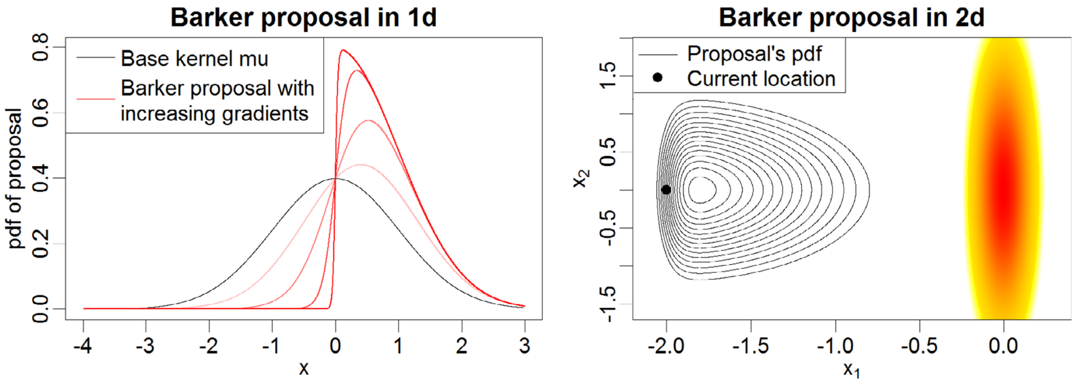

FIGURE 1 Left: density of the Barker proposal in one dimension. Current location is $x = 0$ and the four lines with increasing red intensity correspond to $\nabla \log \pi(x)$ equal to 1, 3, 10 and 50. Right: density of the Barker proposal in two dimensions. Solid lines display the proposal density contours, heat colours refer to the target density, and the current location is $x = (-2, 0)$

We refer to Q^B as the *the Barker proposal*. A simple sampling strategy to generate $y \sim Q^B(x, \cdot)$ is given in Algorithm 1.

```
Algorithm 1 Generating a Barker proposal on $\mathbb{R}$
Require: the current point $x \in \mathbb{R}$.
  (a) Draw $z \sim \mu_\sigma(\cdot)$
  (b) Calculate $p(x, z) = 1/(1 + e^{-z\nabla \log \pi(x)})$
  (c) Set $b(x, z) = 1$ with probability $p(x, z)$, and $b(x, z) = -1$ otherwise
  (d) Set $y = x + b(x, z) \times z$
Output: the resulting proposal $y$.
```

Proposition 4 *Algorithm 1 produces a sample from* Q^B on $\mathbb{R}$.

Algorithm 1 shows that the magnitude $|y–x| = |z|$ of the proposed move does not depend on the gradient $\nabla \log \pi(x)$ here, it is instead dictated only by the choice of symmetric kernel μ_σ. The *direction* of the proposed move is, however, informed by both the magnitude and direction of the gradient. Examining the form of $p(x, z)$, it becomes clear that if the signs of z and $\nabla \log \pi(x)$ are in agreement, then $p(x, z) > 1/2$, and indeed as $z\nabla \log \pi(x) \uparrow \infty$ then $e^{-z\nabla \log \pi(x)} \downarrow 0$ and so $p(x, z) \uparrow 1$. Hence, if the indications from $\nabla \log \pi(x)$ are that $\pi(x + z) \gg \pi(x)$, then it is highly likely that $b(x, z)$ will be set to 1 and $y = x + z$ will be the proposed move. Conversely, if $z\nabla \log \pi(x) < 0$, then there is a larger than 50% chance that the proposal will instead be $y = x - z$. As $\nabla \log \pi(x) \uparrow \infty$ the Barker proposal converges to μ_σ truncated on the right, and similarly to μ_σ truncated on the left as $\nabla \log \pi(x) \downarrow -\infty$. See Figure 1 for an illustration.

The multiplicative term $1/(1 + e^{-\nabla \log \pi(x)(y-x)})$ in Equation (16), which incorporates the gradient information, injects skewness into the base kernel μ_σ (as can be clearly seen in the left-hand plot of Figure 1). Indeed, the resulting distribution Q^B is an example of a *skew-symmetric* distribution (Azzalini, 2013, Eq. (1.3)). Skew-symmetric distributions are a tractable family of (skewed) probability density functions that are obtained by multiplying a symmetric base density function with the cumulative distribution function (cdf) of a symmetric random variable. We refer to (Azzalini 2013 Ch. 1) for more details, including a more general version of Propositions 3 and 4. In the context of skewed distributions, the Gaussian cdf is often used, leading to the skew-normal distribution introduced in Azzalini (1985). In our context, however, the Barker proposal (which leads to the logistic cdf $p(x, z)$ in Algorithm 1) is the only skew-symmetric distribution that can be obtained from Equation (15) using a balancing function g satisfying $g(t) = tg(1/t)$. See the supplement for more detail.

3.3 | The Barker proposal on $\mathbb{R}^d$

There are two natural ways to extend the Barker proposal to $\mathbb{R}^d$, for $d > 1$. The first is to treat each coordinate separately, and generate the proposal $y = (y_1, \dots, y_d)$ by applying Algorithm 1 independently to each coordinate. This corresponds to generating a z_i and $b_i(x, z_i)$ for each $i \in \{1, \dots, d\}$, and choosing the sign of each b_i using

$$p_i(x, z_i) = \frac{1}{1 + e^{-z_i \partial_i \log \pi(x)}},$$

where $\partial_i \log \pi(x)$ denotes the partial derivative of $\log \pi(x)$ with respect to x_i. Writing $Q_i^B(x, dy_i)$ to denote the resulting Barker proposal candidate kernel for the ith coordinate, the full candidate kernel Q^B can then be written

$$Q^B(x, dy) = \prod_{i=1}^{d} Q_i^B(x, dy_i). \tag{17}$$

The full Metropolis–Hastings scheme using the Barker proposal mechanism for a target distribution is given in Algorithm 2 (see the supplement for more details and variations of the algorithm, such as a pre-conditioned version). Note that the computational cost of each iteration of the algorithm is essentially equivalent to that of MALA and will be typically dominated by the cost of computing the gradient and density of the target.

```
Algorithm 2 Metropolis–Hastings with the Barker proposal on $\mathbb{R}^d$
Require: starting point for the chain $x^{(0)} \in \mathbb{R}^d$, and scale $\sigma > 0$.
Set $t = 0$ and do the following:
  (a) Given $x^{(t)} = x$, draw $y_i$ using Algorithm 1 independently for $i \in \{1, ..., d\}$
  (b) Set $x^{(t+1)} = y$ with probability $\alpha^B(x, y)$, where
        $\alpha^B(x, y) = \min\left(1, \frac{\pi(y)}{\pi(x)} \times \prod_i \frac{1 + e^{(x_i - y_i)\partial_i \log \pi(x)}}{1 + e^{(y_i - x_i)\partial_i \log \pi(y)}}\right).$   (18)
      Otherwise set $x^{(t+1)} = x$
  (c) If $t + 1 < N$, set $t \leftarrow t + 1$ and return to step 1, otherwise stop.
Output: the Markov chain $\{x^{(0)}, \dots, x^{(N)}\}$.
```

The second approach to deriving a multivariate Barker proposal consists of sampling $z \in \mathbb{R}^d$ from a d-dimensional symmetric distribution, and then choosing whether or not to flip the sign of *every* coordinate at the same time, using a single global $\check{b}(x,z) \in \{-1,1\}$, to produce the global proposal $y = x + \check{b}(x,z) \times z$. In this case, the probability that $\check{b}(x,z) = 1$ will be

$$\check{p}(x,z) = \frac{1}{1 + e^{-z^T\nabla \log \pi(x)}}. \tag{19}$$

This second approach does not allow gradient information to feed into the proposal as effectively as in the first case. Specifically, only the global inner product $z^T\nabla \log \pi(x)$ is considered, and the decision to alter the sign of every component of z is taken based solely on this value. In other words, once $z \sim \mu_\sigma$ has been sampled, gradient information is only used to make a single binary decision of choosing between the two possible proposals $x + z$ and $x - z$, while in the first strategy gradient information is used to choose between 2^d possible proposals $\{x + b \cdot z : b \in \{-1,1\}^d\}$ (where $b \cdot z := (b_1z_1, \dots, b_dz_d)$). Indeed, the following proposition shows that the second strategy cannot improve over the RWM by more than a factor of two.

Proposition 5 *Let* $\check{P}^B$ *denote the modified Barker proposal on* $\mathbb{R}^d$ *using Equation* (19). *Then* $\text{Gap}(P^R) \geq \text{Gap}(\check{P}^B)/2$.

One can also make a stronger statement than the above proposition, namely that if this strategy is employed, only a constant factor improvement over the RWM can be achieved in terms of asymptotic variance, for any $L^2(\pi)$ function of interest. Given Proposition 5 we choose to use the first strategy described to produce Barker proposals on $\mathbb{R}^d$, and the multi-dimensional candidate kernel given in Equation (17). In the following sections, we will show both theoretically and empirically that this choice does indeed have favourable robustness and efficiency properties.

4 | ROBUSTNESS, SCALING AND ERGODICITY RESULTS FOR THE BARKER PROPOSAL

In this section, we establish results concerning robustness to tuning, scaling with dimension and geometric ergodicity for the Barker proposal scheme. As we will see, the method not only enjoys the superior efficiency of gradient-based algorithms in terms of scaling with dimension, but also shares the favourable robustness properties of the RWM when considering both robustness to tuning and geometric ergodicity.

4.1 | Robustness to tuning

We now examine the robustness to tuning of the Barker proposal using the framework introduced in Section 2. We write Q^B_λ and P^B_λ to denote the candidate and Metropolis–Hastings kernels for the Barker proposal targeting the distribution $\pi^{(\lambda)}$ defined therein, and P^B for the case $\lambda = 1$. The following result characterizes the behaviour of the spectral gap of P^B_λ as $\lambda \downarrow 0$.

Theorem 5 *Assume Condition* 1 *and* $\text{Gap}(P^B) > 0$. *Then it holds that*

$$\text{Gap}(P^B_\lambda) = \Theta(\lambda), \quad \textit{as } \lambda \downarrow 0.$$

Comparing Theorem 5 with Theorems 1–4 from Section 2.3, we see that the Barker proposal inherits the robustness to tuning of random walk schemes and is significantly more robust than the Langevin and Hamiltonian algorithms. In the next section, we establish general conditions under which $\mathrm{Gap}(P^B) > 0$.

4.2 | Geometric ergodicity

In this section, we study the class of target distributions for which the Barker proposal produces a geometrically ergodic Markov chain. We show that geometric ergodicity can be obtained even when the gradient term in the proposal grows faster than linearly, which is typically not the case for MALA and HMC.

Recall that a Markov chain is called *geometrically ergodic* if

$$\|P^t(x,\cdot) - \pi(\cdot)\|_{TV} \leq CV(x)\rho^t, \qquad t \geq 1, \tag{20}$$

for some $C < \infty$, Lyapunov function $V : \mathbb{R}^d \to [1,\infty)$, and $\rho < 1$, where $\|\mu(\cdot) - \nu(\cdot)\|_{TV} := \sup_{A\in\mathcal{B}} |\mu(A) - \nu(A)|$ for probability measures μ and ν. When such a condition can be established for a reversible Markov chain, then a central limit theorem exists for any square-integrable function (Roberts & Rosenthal, 2004).

We prove geometric ergodicity results for generic proposals as in Equation (15), assuming g to be bounded and monotone, and μ_σ to have lighter than exponential tails. Following the discussion in Section 3.3, we consider proposals that are independent across components, leading to

$$Q^{(g)}(x, dy) = \prod_{i=1}^{d} Q_i^{(g)}(x, dy_i) = \prod_{i=1}^{d} \frac{g(e^{\partial_i \log \pi(x)(y_i - x_i)})\mu_\sigma(y_i - x_i)dy_i}{Z_i(x)}, \tag{21}$$

where $Z_i(x) := \int_{\mathbb{R}} g(e^{\partial_i \log \pi(x)(y_i-x_i)})\mu_\sigma(y_i - x_i)dy_i$. With a slight abuse of notation, we use μ_σ to represent one and d-dimensional densities. The Barker proposal in Equation (17) is the special case obtained by taking $g(t) = t/(1+t)$.

For the results of this section, we make the simplifying assumption that π is spherically symmetric outside a ball of radius $R < \infty$.

Condition 4 There exists $R < \infty$ and a differentiable function $f : (0,\infty) \to (0,\infty)$ with $\lim_{r\to\infty} f'(r) = -\infty$ and $f'(r)$ non-increasing for $r > R$ such that $\log \pi(x) = f(\|x\|)$ for $r > R$.

Theorem 6 *Let* $g : (0, \infty)\to(0, \infty)$ *be a bounded and non-decreasing function,* $\int_{\mathbb{R}} \exp(sw)\mu_\sigma(w) dw < \infty$ *for every* $s > 0$, *and* $\inf_{w\in(-\delta,\delta)}\mu_\sigma(w) > 0$ *for some* $\delta > 0$. *If the target density* π *satisfies* Condition 4, *then the Metropolis–Hastings chain with proposal* $Q^{(g)}$ *is* π*-a.e. geometrically ergodic.*

We note that tail regularity assumptions such as Condition 4 are common in this type of analysis (e.g. Durmus et al., 2017a; Jarner & Hansen, 2000). As an intuitive example, the condition is satisfied in the exponential family $\pi(x) \propto \exp(-\alpha\|x\|^\beta)$ for all $\beta > 1$. As a contrast, for MALA and HMC it is known that for $\beta > 2$ the sampler fails to be geometrically ergodic (Livingstone et al., 2019; Roberts & Tweedie, 1996). We expect the Barker proposal to be geometrically ergodic also for the case $\beta = 1$, although we do not prove it in this work. It is worth noting that for the MALA choice $g(t) = \sqrt{t}$ is unbounded above, which is a central reason for the lack of stability compared to bounded choices such as $g(t) = t/(1+t)$ employed in the Barker scheme.

4.3 | Scaling with dimensionality

In this section, we provide preliminary results suggesting that the Barker proposal enjoys scaling behaviour analogous to that of MALA in high-dimensional settings, meaning that under appropriate assumptions it requires the number of iterations per effective sample to grow as $\Theta(d^{1/3})$ with the number of dimensions d as $d \to \infty$. Similarly to Section 4.2, we prove results for general proposals $Q^{(g)}$ as in Equation (21) with balancing functions g satisfying $g(t) = t\,g(1/t)$. The Barker proposal is a special case of the latter family.

We perform an asymptotic analysis for $d \to \infty$ using the framework introduced in Roberts et al. (1997). The main idea is to study the rate at which the proposal step size σ needs to decrease as $d \to \infty$ to obtain well-behaved limiting behaviour for the MCMC algorithm under consideration (such as a $\Theta(1)$ acceptance rate and convergence to a non-trivial diffusion process after appropriate time re-scaling). Based on the rate of decrease of σ, one can infer how the number of MCMC iterations required for each effective sample increases as $d \to \infty$. For example, in the case of the RWM, σ^2 must be scaled as $\Theta(d^{-1})$ as $d \to \infty$ to have a well-behaved limit (Roberts et al., 1997), which leads to RWM requiring $\Theta(d)$ iterations for each effective sample. By contrast, for MALA it is sufficient to take $\sigma^2 = \Theta(d^{-1/3})$ as $d \to \infty$, which leads to only $\Theta(d^{1/3})$ iterations for each effective sample (Roberts & Rosenthal, 1998). While these analyses are typically performed under simplifying assumptions, such as having a target distribution with i.i.d. components, the results have been extended in many ways (e.g. removing the product-form assumption, see Mattingly et al., 2012) obtaining analogous conclusions. See also Beskos et al. (2013) for optimal scaling analysis of HMC and Roberts and Rosenthal (2016) for rigorous connections between optimal scaling results and computational complexity statements.

In this section, we focus on the scaling behaviour of Metropolis–Hastings algorithms with proposal $Q^{(g)}$ as in Equation (21), when targeting distributions of the form $\pi(x) = \prod_{i=1}^{d} f(x_i)$, where f is a one-dimensional smooth density function. Given the structure of $Q^{(g)}$ and $\pi(\cdot)$, the acceptance rate takes the form $\alpha(x,y) = \min\{1, \prod_{i=1}^{d} \alpha_i(x_i, y_i)\}$, where

$$\alpha_i(x_i, y_i) = \frac{f(y_i)}{f(x_i)} \frac{g(e^{\phi'(y_i)(x_i - y_i)})}{g(e^{\phi'(x_i)(y_i - x_i)})} \frac{Z_i(x_i)}{Z_i(y_i)}, \tag{22}$$

and $\phi = \log f$. In such a context, the scaling properties of the MCMC algorithms under consideration are typically governed by the behaviour of $\log(\alpha_i(x_i, y_i))$ as y_i gets close to x_i, or more precisely by degree of the leading term in the Taylor series expansion of $\log(\alpha_i(x_i, x_i + \sigma u_i))$ in powers of σ as $\sigma \to 0$ for fixed x_i and u_i. For example, in the case of the RWM one has $\log(\alpha_i(x_i, x_i + \sigma u_i)) = \Theta(\sigma)$ as $\sigma \to 0$, which in fact implies the proposal variance σ^2 must decrease at a rate $\Theta(d^{-1})$ to obtain a non-trivial limit. By contrast, when the MALA proposal is used, one has $\log(\alpha_i(x_i, x_i + \sigma u_i)) = \Theta(\sigma^3)$ as $\sigma \to 0$, which in turn leads to $\sigma^2 = \Theta(d^{-1/3})$. See Sections 2.1–2.2 of Durmus et al. (2017b) for a more detailed and rigorous discussion on the connection between the Taylor series expansion of $\log(\alpha_i(x_i, y_i))$ and MCMC scaling results. The following proposition shows that the condition $g(t) = t\ g(1/t)$, when combined with some smoothness assumptions, is sufficient to ensure that the proposals $Q^{(g)}$ lead to $\log(\alpha_i(x_i, x_i + \sigma u_i)) = \mathcal{O}(\sigma^3)$ as $\sigma \to 0$.

Proposition 6 *Let $g : (0, \infty) \to (0, \infty)$ and $g(t) = t\, g(1/t)$ for all t. If g is three times continuously differentiable and $\int_{\mathbb{R}} g^{(j)}(e^{sw})\mu(w)dw < \infty$ for all $s > 0$ and $j \in \{0, 1, 2, 3\}$, where $g^{(j)} : (0, \infty) \to (0, \infty)$ is the jth derivative of g, then*

$$\log(\alpha_i(x_i, x_i + \sigma u_i)) = \mathcal{O}(\sigma^3) \qquad as\ \sigma \to 0, \tag{23}$$

for any x_i and u_i in $\mathbb{R}$.

Proposition 6 suggests that Metropolis–Hastings algorithms with proposals $Q^{(g)}$ such that $g(t) = t\, g(1/t)$ have scaling behaviour analogous to MALA, meaning that $\sigma^2 = \Theta(d^{-1/3})$ is sufficient to ensure a non-trivial limit and thus $\Theta(d^{1/3})$ iterations are required for each effective sample. To make these arguments rigorous, one should prove weak convergence results for $d \to \infty$, as in Roberts and Rosenthal (1998). Proving such a result for a general g would require a significant amount of technical work, thus going beyond the scope of this section. In this paper we rather support the conjecture of $\Theta(d^{1/3})$ scaling for $Q^{(g)}$ by means of simulations (see Section 5.2). While Proposition 6 only shows $\log(\alpha_i(x_i, x_i + \sigma u_i)) = \mathcal{O}(\sigma^3)$, it is possible to show that $\log(\alpha_i(x_i, x_i + \sigma u_i)) = \Theta(\sigma^3)$ with some extra assumptions on ϕ to exclude exceptional cases (see the supplement for more detail).

5 | SIMULATIONS WITH FIXED TUNING PARAMETERS

Throughout Sections 5 and 6, we choose the symmetric density μ_σ within the random walk and Barker proposals to be $N(0, \sigma^2 \mathbb{I}_d)$ for simplicity. Note, however, that any symmetric density μ_σ could in principle be used. It would be interesting to explore the impact of different choices of μ_σ to the performances of the Barker algorithm, and we leave such a comparison to future work.

5.1 | Illustrations of robustness to tuning

We first provide an illustration of the robustness to tuning of the random walk, Langevin and Barker algorithms in three simple one-dimensional settings. In each case we approximate the expected squared jump distance (ESJD) using 10^4 Monte Carlo samples and standard Rao–Blackwellisation techniques, across of range of different proposal step-sizes between 0.01 and 100. As is clearly shown in Figure 2, all algorithms perform similarly when the step-size is smaller than optimal, as suggested in Section 2.4. As the step-size increases beyond this optimum, however, behaviours begin to differ. In particular, the ESJD for MALA rapidly decays to zero, whereas in the random walk and Barker cases the reduction is much less pronounced. In fact, the rate of decay is similar for the two schemes, which is to be expected following the results of Sections 4.1 and 2.3. See the supplement for a similar illustration on a 20-dimensional example.

5.2 | Comparison of efficiency on isotropic targets

Next we compare the expected squared jump distance of the random walk, Langevin and Barker schemes when sampling from isotropic distributions of increasing dimension, with optimised proposal scale (chosen to maximise expected squared jumping distance). This set-up is favourable to MALA, which is the least robust scheme among the three, as the target distribution is homogeneous and the proposal step-size optimally chosen. We consider target distributions with

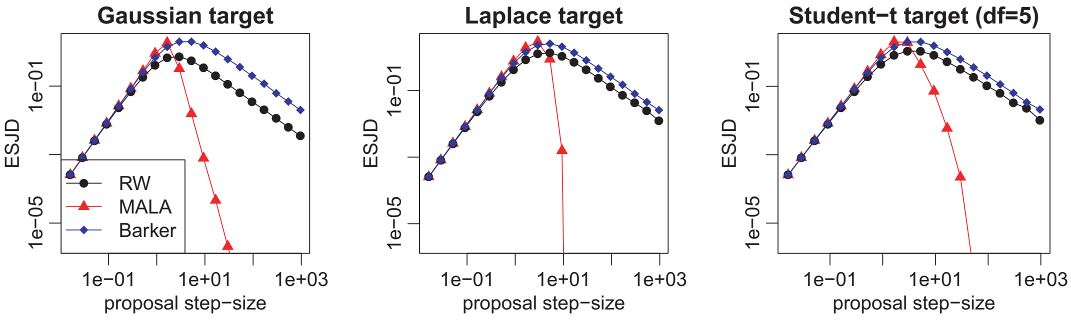

FIGURE 2 Expected squared jump distance against proposal step-size for random walk Metropolis, Metropolis-adjusted Langevin algorithm and Barker on different one-dimensional targets

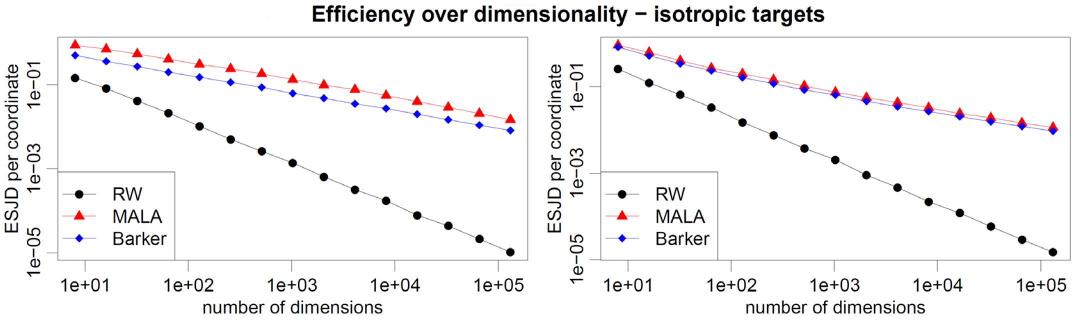

FIGURE 3 Expected squared jump distance against dimensionality for random walk Metropolis, Metropolis-adjusted Langevin algorithm and Barker schemes with optimally-tuned step size. The target distribution has i.i.d. coordinates following either a Gaussian distribution (left plot) or a hyperbolic one (right plot)

independent and identically distributed (i.i.d.) components, corresponding to the scenario studied theoretically in Section 4.3. We set the distribution of each coordinate to be either a standard normal distribution or a hyperbolic distribution, corresponding to $\log \pi(x) = -\sum_{i=1}^{d} x_i^2/2 + \text{const}$ and $\log \pi(x) = -\sum_{i=1}^{d}(0.1 + x_i^2)^{1/2} + \text{const}$, respectively. Figure 3 shows how the ESJD per coordinate decays as dimension increases for the three algorithms. For MALA and Barker, the ESJD appears to decrease at the same rate as d increases, which is in accordance with the preliminary results in Section 4.3. In the Gaussian case, MALA outperforms Barker roughly by a factor of 2 regardless of dimension (more precisely, the ESJD ratio lies between 1.7 and 2.5 for all values of d in Figure 3), while in the hyperbolic case the same factor is around 1.2, again independently of dimension (ESJD ratio between 1.1 and 1.25 for all values of d in Figure 3). The rate of decay for the RWM is faster, as predicted by the theory.

6 | SIMULATIONS WITH ADAPTIVE MARKOV CHAIN MONTE CARLO

In this section, we illustrate how robustness to tuning affects the performance of adaptive MCMC methods.

6.1 | Adaptation strategy and algorithmic set-up

We use Algorithm 4 in Section 5 of Andrieu and Thoms (2008) to adapt the tuning parameters within each scheme. Specifically, in each case, a Markov chain is initialised using a chosen global proposal scale σ_0 and an identity pre-conditioning matrix $\Sigma_0 = \mathbb{I}_d$, and at each iteration the global scale and pre-conditioning matrix are updated using the equations

$$\log(\sigma_t) = \log(\sigma_{t-1}) + \gamma_t \times (\alpha(X^{(t)}, Y^{(t)}) - \overline{\alpha}_*) \tag{24}$$

$$\mu_t = \mu_{t-1} + \gamma_t \times (X^{(t)} - \mu_{t-1}) \tag{25}$$

$$\Sigma_t = \Sigma_{t-1} + \gamma_t \times ((X^{(t)} - \mu_t)(X^{(t)} - \mu_t)^T - \Sigma_{t-1}). \tag{26}$$

Here $X^{(t)}$ denotes the current point in the Markov chain, $Y^{(t)}$ is the proposed move, $\mu_0 = 0$, $\overline{\alpha}_*$ denotes some ideal acceptance rate for the algorithm and the parameter γ_t is known as the learning rate. We set $\overline{\alpha}_*$ to be 0.23 for RWM, 0.57 for MALA and 0.40 for Barker. We tried changing the value of $\overline{\alpha}_*$ for Barker in the range [0.2, 0.6] without observing major differences. In our simulations, we constrain Σ_t to be diagonal (i.e., all off-diagonal terms in Equation (26) are set to 0). This is often done in practice to avoid having to learn a dense pre-conditioning matrix, which has both a high computational cost and would require a large number of MCMC samples. See the supplement for full details on the pre-conditioned Barker schemes obtained with both diagonal and dense matrix Σ_t, including pseudo-code of the resulting algorithms.

We set the learning rate to $\gamma_t := t^{-\kappa}$ with $\kappa \in (0.5, 1)$, as for example suggested in (Shaby & Wells, 2010). Small values of κ correspond to more aggressive adaptation, and for example Shaby and Wells (2010) suggest using $\kappa = 0.8$. In the simulations of Section 6.2, we use $\kappa = 0.6$ as this turned out to be a good balance between fast adaptation and stability for MALA ($\kappa = 0.8$ resulted in too slow adaptation, while values of κ lower than 0.6 led to instability). The adaptation of RWM and Barker was not very sensitive to the value of κ. Unless specified otherwise, all algorithms are randomly initialized with each coordinate sampled independently from a normal distribution with standard deviation 10. Following the results from the optimal scaling theory (Roberts & Rosenthal, 2001), we set the starting value for the global scale as $\sigma_0^2 = 2.4^2/d$ for RWM and $\sigma_0^2 = 2.4^2/d^{1/3}$ for MALA. For Barker we initialize σ_0 to the same values as MALA.

6.2 | Performance on target distributions with heterogeneous scales

In this section, we compare the adaptive algorithms described above when sampling from target distributions with significant heterogeneity of scales across their components. We consider 100-dimensional target distributions with different types of heterogeneity, tail behaviour and degree of skewness according to the following four scenarios:

1. (*One coordinate with small scale; Gaussian target*) In the first scenario, we consider a Gaussian target with zero mean and diagonal covariance matrix. We set the standard deviation of the first coordinate to 0.01 and that of the other coordinates to 1. This scenario mirrors the theoretical framework of Sections 2 and 4.1 in which a single coordinate is the source of heterogeneity.

2. (*Coordinates with random scales; Gaussian target*) Here we modify scenario 1 by generating the standard deviations of each coordinate randomly, sampling them independently from a log-normal distribution. More precisely, we sample $\log(\eta_i) \sim N(0, 1)$ independently for $i = 1, \dots, 100$, where η_i is the standard deviation of the ith component.
3. (*Coordinates with random scales; Hyperbolic target*) In the third scenario, we change the tail behaviour of the target distribution, replacing the Gaussian with a hyperbolic distribution (a smoothed version of the Laplace distribution to ensure $\log \pi \in C^1(\mathbb{R}^d)$). In particular, we set $\log \pi(x) = -\sum_{i=1}^{d} (\epsilon + (x_i/\eta_i)^2)^{1/2} + c$, with $\epsilon = 0.1$ and c being a normalizing constant. The scale parameters $(\eta_i)_i$ are generated randomly as in scenario 2.
4. (*Coordinates with random scales; Skew-normal target*) Finally, we consider a non-symmetric target distribution, which represents a more challenging and realistic situation. We assume that the ith coordinate follows a skew-normal distribution with scale η_i and skewness parameter α, meaning that $\log \pi(x) = -\frac{1}{2}\sum_{i=1}^{d} (x_i/\eta_i)^2 + \sum_{i=1}^{d} \log \Phi(\alpha x_i/\eta_i) + c$, with c being a normalizing constant. We set $\alpha = 4$ and generate the η_i's randomly as in scenario 2.

First we provide an illustration of the behaviour of the three algorithms by plotting the trace plots of tuning parameters and MCMC trajectories—see Figure 4 for the results in scenario 1. The adaptation of tuning parameters for the Barker scheme stabilises within a few hundred iterations, after which the algorithm performance appears to be stable and efficient. On the contrary both RWM and MALA struggle to learn the heterogeneous scales and the adaptation process has either just stabilized or not yet stabilized after 10^4 iterations. Looking at the behaviour of MALA in Figure 4 we see that, in order for the algorithm to achieve a non-zero acceptance rate, the global scale parameter σ_t must first be reduced considerably to accommodate the smallest scale of $\pi(\cdot)$. At this point the algorithm can slowly begin to learn the components of the pre-conditioning matrix Σ_t, but this learning occurs very slowly because the comparatively small value for σ_t results in poor mixing across all other dimensions than the first. Analogous plots for scenarios 2, 3 and 4 are given in the supplement and display comparable behaviour.

We then compare algorithms in a more quantitative way, by looking at the average mean squared error (MSE) of MCMC estimators of the first moment of each coordinate, which is a standard metric in MCMC. For any $h : \mathbb{R}^d \to \mathbb{R}$, define the corresponding MSE as $\mathbb{E}[(\hat{h}^{(t)} - \mathbb{E}_\pi[h])^2]$ where $\hat{h}^{(t)} = (t - t_{burn})^{-1} \sum_{i=t_{burn}+1}^{t} h(X^{(i)})$ is the MCMC estimator of $\mathbb{E}_\pi[h]$ after t iterations of the algorithm. Here t_{burn} is a burn-in period, which we set to $t_{burn} = \lfloor t/2 \rfloor$, where $\lfloor \cdot \rfloor$ denotes the floor function. Below, we report the average MSE for the collection of test functions given by $h(x) = x_i/\eta_i$ for $i = 1, \dots, d$ after t MCMC iterations (rescaling by η_i is done to give equal importance to each coordinate).

In addition, we also monitor the rate at which the pre-conditioning matrix Σ_t converges to the covariance of π, denoted as Σ, in order to measure how quickly the adaptation mechanism learns suitable local tuning parameters. We consider the l^2-distance between the diagonal elements of Σ_t and Σ on the log scale. This leads to the following measure of convergence of the tuning parameters after t MCMC iterations:

$$d_t = \mathbb{E}\left[\frac{1}{\sqrt{d}} \left(\sum_{i=1}^{d} (\log(\Sigma_{t,ii}) - \log(\Sigma_{ii}))^2 \right)^{1/2} \right], \tag{27}$$

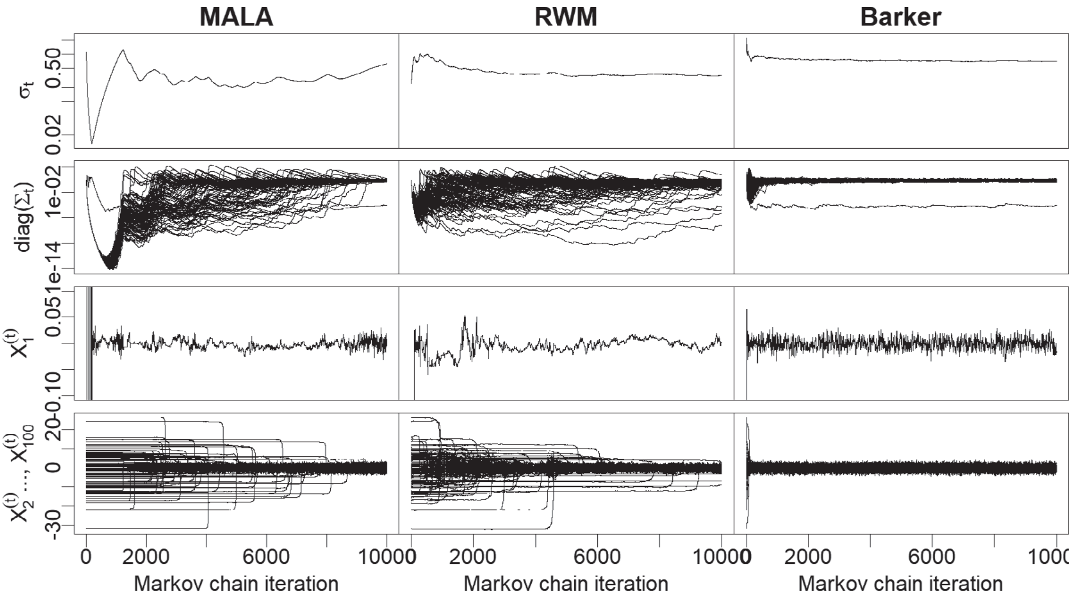

FIGURE 4 Random walk Metropolis, Metropolis-adjusted Langevin algorithm and Barker schemes with adaptive tuning as in Equations (24)–(26) and learning rate set to $\gamma_t = t^{-\kappa}$ with $\kappa = 0.6$. The target distribution is a 100-dimensional Gaussian in which the first component has standard deviation 0.01 and all others have unit scale. First row: adaptation of the global scale σ_t; second row: adaptation of the local scales $\text{diag}(\Sigma_t) = (\Sigma_{t,ii})_{i=1}^{100}$; third row: trace plot of first coordinate; fourth row: trace plots of coordinates from 2 to 100 (superposed)

where the expectation is with respect the Markov chain $(X^{(t)})_{t\geq 1}$. We use the log scale as it is arguably more appropriate than the natural one when comparing step-size parameters, and we focus on diagonal terms as both Σ_t and Σ are diagonal here. Monitoring the convergence of d_t to 0 we can compare the speed at which good tuning parameters are found during the adaptation process for different schemes.

Figure 5 displays the evolution of d_t and the MSE defined above over 4×10^4 iterations of each algorithms, where d_t and the MSE are estimated by averaging over 100 independent runs of each algorithm. The results are in accordance with the illustration in Figure 4, and suggest that the Barker scheme is robust to different types of targets and heterogeneity and results in very fast adaptation, while both MALA and RWM require significantly more iterations to find good tuning parameters. The tuning parameters of MALA appear to exhibit more unstable behaviour than RWM in the first few thousands iterations (larger d_t), while after that they converge more quickly, which again is in accordance with the behaviour observed in Figure 5 and with the theoretical considerations of Sections 2 and 4.1. To further quantify the tuning period, we define the time to reach a stable level of tuning as $\tau_{adapt}(\epsilon) = \inf\{t \geq 1 : d_t \leq \epsilon\}$ for some $\epsilon > 0$. We take ϵ=1 and report the resulting values in Table 1, denoting $\tau_{adapt}(1)$ simply as τ_{adapt}. The results show that in these examples Barker always has the smallest adaptation time, with a speed-up compared to RWM of at least 34x in all four scenarios, and a speed-up compared to MALA ranging between 3x (scenario 3) and 30x (scenario 2). The adaptation times τ_{adapt} tend to increase from scenarios 1 to 4, suggesting that the target distribution becomes more challenging as we move from scenarios 1 to 4. The hardest case for Barker seems to be the hyperbolic target, although even there the tuning stabilized in roughly 3000 iterations, while the hardest case for MALA is the skew-normal, in which tuning stabilized in roughly 30,000 iterations.

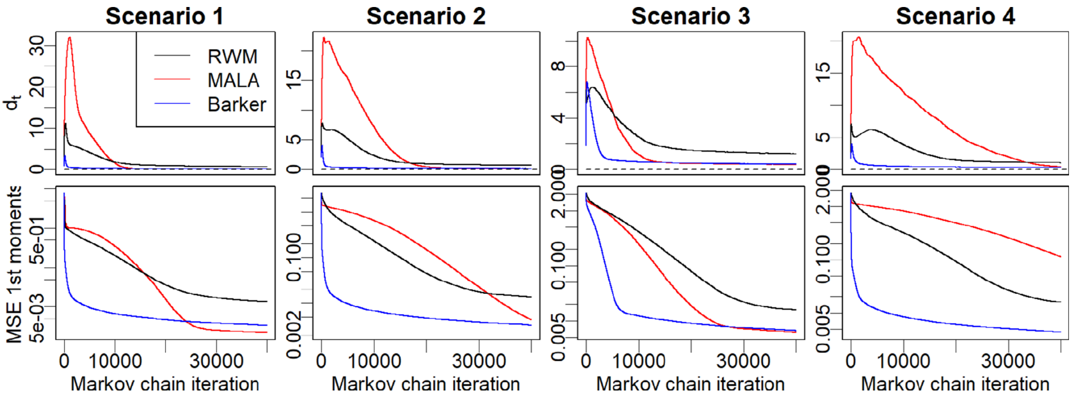

FIGURE 5 Comparison of random walk Metropolis, Metropolis-adjusted Langevin algorithm and Barker on the four target distributions (scenarios 1 to 4) described in Section 6.2, averaging over ten repetitions of each algorithm. First row: convergence of tuning parameters, measured by d_t defined in Equation (27). Second row: Mean Square Error of Markov chain Monte Carlo estimators of first moments averaged over all coordinates

TABLE 1 Adaptation times (τ_{adapt}) and mean squared errors (MSE) from 10 k, 20 k and 40 k iterations of the random walk Metropolis (RWM), Metropolis-adjusted Langevin algorithm (MALA) and Barker algorithms under each of the four heterogeneous scenarios described in Section 6.2

	Method	τ_{adapt}	MSE_{10k}	MSE_{20k}	MSE_{40k}
1	RWM	18,757	0.200	0.036	0.013
	MALA	10,785	0.348	0.016	0.002
	Barker	524	0.007	0.005	0.003
2	RWM	19,163	0.228	0.045	0.013
	MALA	17,298	0.644	0.147	0.004
	Barker	542	0.007	0.005	0.003
3	RWM	>40 k	0.409	0.080	0.016
	MALA	10,630	0.248	0.019	0.006
	Barker	3294	0.012	0.009	0.007
4	RWM	>40 k	0.315	0.092	0.016
	MALA	34,340	0.813	0.488	0.112
	Barker	1427	0.008	0.006	0.004

The differences in the adaptation times have a direct implication on the resulting MSE of MCMC estimators, which is intuitive because the Markov chain will typically start sampling efficiently from π only once good tuning parameters are found. As we see from the second row of Figure 5 and the second part of Table 1, the MSE of Barker is already quite low (between 0.007 and 0.012) after 10^4 iterations in all scenarios, while RWM and MALA need significantly more iterations to achieve the same MSE. After finding good tuning parameters and having sampled enough, MALA is slightly more efficient than Barker for the Gaussian target in scenario 1 and equally efficient in the hyperbolic target of scenario 3, which is consistent with the simulations of Section 5.2 under optimal tuning.

6.3 | Comparison on a Poisson random effects model

In this section, we consider a Poisson hierarchical model of the form

$$\begin{aligned} y_{ij}|\eta_i &\overset{ind}{\sim} \text{Poisson}(\exp(\eta_i)) \qquad && j = 1, \dots, n_i, \\ \eta_i|\mu &\overset{ind}{\sim} \text{N}(\mu, \sigma_\eta^2) && i = 1, \dots, I, \\ \mu &\sim \text{N}(0, 10^2), && \end{aligned} \tag{28}$$

and test the algorithms on the task of sampling from the resulting posterior distribution $p(\mu, \eta_1, \dots, \eta_I|\mathbf{y})$, where $\mathbf{y} = (y_{ij})_{ij}$ denotes the observed data. In our simulations, we set $I = 50$ and $n_i = 5$ for all i, leading to 51 unknown parameters and 250 observations.

The model in Equation (28) is an example of a generalized linear model that induces a posterior distribution with light tails and potentially large gradients of $\log \pi$, which creates a challenge for gradient-based algorithms. In particular, the task of sampling from the posterior becomes harder when either the observations $(y_{ij})_{ij}$ contain large values or they are heterogeneous across values of $i \in \{1, \dots, I\}$. The former case results in a more peaked posterior distribution with larger gradients, while the latter induces heterogeneity across the posterior distributions of the parameters η_i.

In our simulations, we consider three scenarios, corresponding to increasingly challenging target distributions:

1. In the first scenario, we take $\sigma_\eta = 1$ and generate the data $\mathbf{y}$ from the model in Equation (28) assuming the data-generating value of μ to be $\mu^* = 5$ and sampling the data-generating values of $\eta_1, \dots, \eta_I$ from their prior distribution.
2. In the second scenario, we increase the value of σ_η to 3, which induces more heterogeneity across the parameters $\eta_1, \dots, \eta_I$.
3. In the third scenario, we keep $\sigma_\eta = 3$ and increase the values of μ^* to 10, thus inducing larger gradients.

In each scenario, we run the algorithm directly on the joint parameter space $(\mu, \eta_1, \dots, \eta_I)$. Similarly to Section 6.2, we first provide an illustration of the behaviour of the tuning parameters and MCMC trace plots for RWM, MALA and Barker in Figure 6. Here all algorithms are run for 5×10^4 iterations, with the target defined in the first scenario. We use the adaptation strategy of Section 6.2 for tuning, following Equations (24)–(26) with $\kappa = 0.6$ and Σ_t constrained to be diagonal, and initialize the chains from a random configuration sampled from the prior distribution of the model. In this example, the random walk converges to stationarity in roughly 10,000 iterations while the Barker scheme takes a few hundreds. By contrast MALA struggles to converge and exhibits unstable behaviour even after 5×10^4 iterations. Note that the first 3×10^4 iterations of MALA, in which the parameter μ appears to be constant, do not correspond to rejections but rather to moves with very small increments in the μ component.

We then provide a more systematic comparison between the algorithms under consideration in Table 2. In addition to RWM, MALA and Barker, we also consider a state-of-the-art

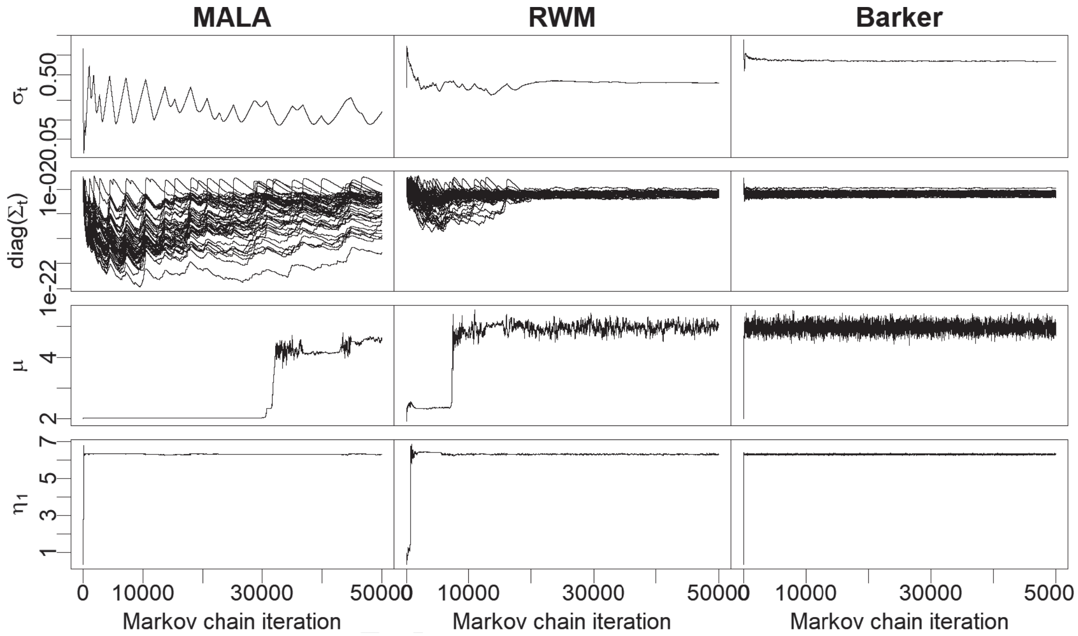

FIGURE 6 Behaviour of random walk Metropolis, Metropolis-adjusted Langevin algorithm and Barker on the posterior distribution from the Poisson hierarchical model in Equation (28). Data are generated as in the first scenario of Section 6.3. First row: adaptation of the global scale σ_t; second row: adaptation of the local scales $\text{diag}(\Sigma_t) = (\Sigma_{t,ii})_{i=1}^{100}$; third row: trace plot of the parameter μ; fourth row: trace plots of the parameter η_1

implementation of adaptive HMC, namely the Stan (Stan Development Team, 2020) implementation of the No-U-Turn Sampler (NUTS) (Hoffman & Gelman, 2014) as well as of standard HMC (referred to as 'static HMC' in the Stan package). The NUTS algorithm is a variant of standard HMC in which the number of leapfrog iterations, that is, the parameter L in Equation (10), is allowed to depend on the current state (using a 'No-U-Turn' criterion). The resulting number of leapfrog steps (and thus log-posterior gradient evaluations) per iteration is not fixed in advance but rather tuned adaptively depending on the hardness of the problem. This is also the case for the static HMC algorithm implementation in Stan, as in that case the total integration time in Equation (10) is fixed and the step-size and mass matrix are adapted. For both algorithms, we use the default Stan version that learns a diagonal covariance/mass matrix during the adaptation process. This is analogous to constraining the preconditioning matrix Σ_t for RWM, MALA and Barker to be diagonal, as we are doing here.

Table 2 reports the results of the simulations for the five algorithms in each of the three scenarios. For each algorithm, we report the number of log-posterior gradient evaluations and the minimum and median effective sample size (ESS) across the 51 unknown parameters. The ESS values are computed with the `effectiveSize` function from the `coda` R package (Plummer et al., 2006), discarding the first half of the samples as burn-in. The RWM, MALA and Barker schemes are run for 5×10^4 iterations, and the HMC and NUTS schemes for 2×10^3 iterations. The latter is the default value in the Stan package and in this example corresponds to a number of gradient evaluations between 1.7×10^4 and 1.6×10^7. All numbers in Table 2 are averaged over ten independent replications of each algorithm. We use the minimum ESS per gradient

TABLE 2 Comparison of sampling schemes on the posterior distribution arising from the Poisson hierarchical model in Equation (28)

	Method	Iterations (n)	Leapfrog steps/n	Gradient calls (g)	ESS	$ESS/g \times 100$
1	RWM	5×10^4	–	–	(49, 66)	–
	MALA	5×10^4	–	5×10^4	(648, 727)	1.30 ± 2.73
	Barker	5×10^4	–	5×10^4	(1445, 1587)	2.89 ± 0.07
	HMC	2×10^3	89.5	1.8×10^5	(285, 1954)	0.25 ± 0.78
	NUTS	2×10^3	8.5	1.7×10^4	(1175, 1822)	6.95 ± 1.68
2	RWM	5×10^4	–	–	(0.4, 10.6)	–
	MALA	5×10^4	–	5×10^4	(0.0, 8.0)	< 0.01
	Barker	5×10^4	–	5×10^4	(1365, 1563)	2.73 ± 0.13
	HMC	2×10^3	797	1.6×10^6	(25, 1949)	< 0.01
	NUTS	2×10^3	57.7	1.2×10^5	(942, 1826)	1.19 ± 1.14
3	RWM	5×10^4	–	–	(0.0, 5.3)	–
	MALA	5×10^4	–	5×10^4	(0.0, 0.2)	< 0.01
	Barker	5×10^4	–	5×10^4	(1301, 1594)	2.60 ± 0.92
	HMC	2×10^3	8103	1.6×10^7	(3.3, 899)	< 0.01
	NUTS	2×10^3	179	3.5×10^5	(137, 348)	0.012 ± 0.14

Blocks of rows from 1 to 3 refer to the three data-generating scenarios described in Section 6.3. All numbers are averaged across ten repetitions of each algorithm. For each algorithm we report: number of iterations; number of leapfrog steps per iteration and total number of gradient evaluations (when applicable); estimated effective sample size (ESS) (minimum and median across parameters); minimum ESS per hundred gradient evaluations (with standard deviation across the ten repetitions).

evaluation as an efficiency metric, of which we report the mean and standard deviation across the 10 replicates (multiplied by 100 to facilitate readability).

According to Table 2, NUTS is the most efficient scheme in scenario 1, while Barker is the most efficient one in scenarios 2 and 3. This is in accordance with the intuition of Barker being a more robust scheme, as the target distribution becomes more challenging as we move from scenarios 1 to 3. MALA struggles to converge to stationarity in scenarios 2 and 3 (with an estimated ESS around zero), while it performs better in scenario 1, although with a high variability across different runs (shown by the large standard deviation in the last column). The RWM displays low ESS values for all three scenarios, although with a less dramatic deterioration going from scenarios 1 to 3. Interestingly, the performances of Barker are remarkably stable across scenarios (with an ESS of around 1400), as well as across parameters for which ESS is computed (in all cases the minimum and median ESS are close to each other) and across repetitions (shown by the relatively small standard deviation in the last column). We note that NUTS is also remarkably effective taking into consideration that it is not an algorithm designed with a major emphasis on robustness, but that performance does degrade when moving from scenarios 1 to 3. As in the MALA case, static HMC struggles to converge in scenarios 2 and 3 and is not very efficient in scenario 1. Note that NUTS, and in particular HMC, compensate for the increasing difficulty of the target by increasing the number of leapfrog steps per iteration. For example, the drop in efficiency of NUTS between scenarios 1 and 2 is mostly due to the increase in average number of leapfrog iterations from 8.5 to 57.7 rather than in a decrease in ESS. Somewhat surprisingly, in static HMC the number of leapfrog steps per iteration is increased significantly more than NUTS, which could either be due to genuine algorithmic differences or to variations in the details of the adaptation strategy implemented in Stan. Overall, Barker and NUTS are the two most efficient algorithms in these simulation, with a relative efficiency that depends on the scenario under consideration: NUTS being roughly 2.4 times more efficient in scenario 1, Barker 2.3 times more efficient in scenario 2 and Barker 40 times more efficient in scenario 3.

6.4 | Additional simulations reported in the supplement

In the supplement, we report additional simulations for some of the above experiments. As a sensitivity check, we also performed simulations using the tamed Metropolis-adjusted Langevin algorithm (Brosse et al., 2018) and the truncated Metropolis-adjusted Langevin algorithm (Atchade, 2006; Roberts & Tweedie, 1996), two more robust modifications to MALA in which large gradients are controlled by monitoring the size of $\|\nabla \log \pi(x)\|$. The schemes do offer some added stability compared to MALA in terms of controlling large gradients, but ultimately are still very sensitive to heterogeneity of the target distribution and to the choice of the truncation level, and do not exhibit the same robustness observed in the case of the Barker scheme. See the supplement for implementation details, results and further discussion.

7 | DISCUSSION

We have introduced a new gradient-based MCMC method, *the Barker proposal*, and have demonstrated both analytically and numerically that it shares the favourable scaling properties of other gradient-based approaches, along with an increased level of robustness, both in terms of geometric ergodicity and robustness to tuning (as defined in the present paper). The most striking benefit of the method appears to be in the context of adaptive MCMC. Evidence suggests that combining the efficiency of a gradient-based proposal mechanism with a method that exhibits robustness to tuning gives a combination of stability and speed that is very desirable in this setting, and can lead to efficient sampling that requires minimal practitioner input.

The theoretical results in this paper could be extended by studying in greater depth the large λ regime (Section 2.4) and the high-dimensional scaling of the Barker proposal (Section 4.3). Of course, there are many other algorithms that could be considered under the robustness to tuning framework, and it is worthwhile future work to explore which features of a scheme result in either robustness to tuning or a lack of it. Extensions to the Barker proposal that incorporate momentum and exhibit the $d^{-1/4}$ decay in efficiency with dimension enjoyed by HMC may be possible, as well as the development of other methods within the first-order locally balanced proposal framework introduced in Section 3, or indeed schemes that are exact at higher orders.

ACKNOWLEDGEMENTS

The authors thank Marco Frangi for preliminary work in his Master's thesis related to the ergodicity arguments in the paper. SL acknowledges support from the Engineering and Physical Sciences Research Council through grant number EP/K014463/1 (i-like). GZ acknowledges support from the European research council starting grant 306406 (N-BNP), and by the Italian ministry of education, universities and research PRIN Project 2015SNS29B.

ORCID

Giacomo Zanella https://orcid.org/0000-0001-6727-9884

REFERENCES

Andrieu, C. & Thoms, J. (2008) A tutorial on adaptive MCMC. *Statistics and Computing*, 18, 343–373.

Atchade, Y.F. (2006) An adaptive version for the Metropolis adjusted Langevin algorithm with a truncated drift. *Methodology and Computing in Applied Probability*, 8, 235–254.

Azzalini, A. (1985) A class of distributions which includes the normal ones. *Scandinavian Journal of Statistics*, 12, 171–178.

Azzalini, A. (2013) *The skew-normal and related families*. Institute of Mathematical Statistics Monographs. Cambridge: Cambridge University Press.

Barker, A.A. (1965) Monte Carlo calculations of the radial distribution functions for a proton-electron plasma. *Australian Journal of Physics*, 18, 119–134.

Beskos, A., Pillai, N., Roberts, G., Sanz-Serna, J.-M. & Stuart, A. (2013) Optimal tuning of the hybrid Monte Carlo algorithm. *Bernoulli*, 19, 1501–1534.

Beskos, A., Roberts, G., Thiery, A. & Pillai, N. (2018) Asymptotic analysis of the random walk metropolis algorithm on ridged densities. *The Annals of Applied Probability*, 28, 2966–3001.

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

Brosse, N., Durmus, A., Moulines, É. & Sabanis, S. (2018) The tamed unadjusted Langevin algorithm. *Stochastic Processes and their Applications*, 129(10), 3638–3663.

Dalalyan, A.S. (2017) Theoretical guarantees for approximate sampling from smooth and log-concave densities. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79, 651–676.

Duane, S., Kennedy, A.D., Pendleton, B.J. & Roweth, D. (1987) Hybrid Monte Carlo. *Physics Letters B*, 195, 216–222.

Durmus, A. & Moulines, E. (2017) Nonasymptotic convergence analysis for the unadjusted Langevin algorithm. *The Annals of Applied Probability*, 27, 1551–1587.

Durmus, A., Moulines, E. & Saksman, E. (2017a) On the convergence of Hamiltonian Monte Carlo. *arXiv preprint arXiv:1705.00166*.

Durmus, A., Roberts, G.O., Vilmart, G. & Zygalakis, K.C. (2017b) Fast Langevin based algorithm for MCMC in high dimensions. *The Annals of Applied Probability*, 27, 2195–2237.

Hastings, W.K. (1970) Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1), 97–109.

Hoffman, M.D. & Gelman, A. (2014) The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15, 1593–1623.

Jarner, S.F. & Hansen, E. (2000) Geometric ergodicity of Metropolis algorithms. *Stochastic Processes and their Applications*, 85, 341–361.

Krauth, W. (2006) *Statistical mechanics: algorithms and computations*, vol. 13. Oxford: OUP.

Livingstone, S., Betancourt, M., Byrne, S. & Girolami, M. (2019) On the geometric ergodicity of Hamiltonian Monte Carlo. *Bernoulli*, 25, 3109–3138. To appear.

Mattingly, J.C., Pillai, N.S. & Stuart, A.M. (2012) Diffusion limits of the random walk Metropolis algorithm in high dimensions. *The Annals of Applied Probability*, 22, 881–930.

Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H. & Teller, E. (1953) Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21, 1087–1092.

Neal, R.M. (2003) Slice sampling. *The Annals of Statistics*, 31, 705–767.

Neal, R.M. (2011) MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2, 2.

Peskun, P.H. (1973) Optimum Monte-Carlo sampling using Markov chains. *Biometrika*, 60, 607–612.

Plummer, M., Best, N., Cowles, K. & Vines, K. (2006) Coda: convergence diagnosis and output analysis for MCMC. *R News*, 6, 7–11. Available from: https://journal.r-project.org/archive/.

Power, S. & Goldman, J.V. (2019) Accelerated sampling on discrete spaces with non-reversible Markov processes. *arXiv preprint arXiv:1912.04681*.

Roberts, G.O. & 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.

Roberts, G.O. & Rosenthal, J.S. (2001) Optimal scaling for various Metropolis-Hastings algorithms. *Statistical Science*, 16, 351–367.

Roberts, G.O. & Rosenthal, J.S. (2004) General state space Markov chains and MCMC algorithms. *Probability Surveys*, 1, 20–71.

Roberts, G.O. & Rosenthal, J.S. (2016) Complexity bounds for Markov chain Monte Carlo algorithms via diffusion limits. *Journal of Applied Probability*, 53, 410–420.

Roberts, G.O. & Tweedie, R.L. (1996) Exponential convergence of Langevin distributions and their discrete approximations. *Bernoulli*, 2, 341–363.

Roberts, G.O., Gelman, A. & Gilks, W.R. (1997) Weak convergence and optimal scaling of random walk Metropolis algorithms. *The Annals of Applied Probability*, 7, 110–120.

Rosenthal, J.S. (2003) Asymptotic variance and convergence rates of nearly-periodic Markov chain Monte Carlo algorithms. *Journal of the American Statistical Association*, 98, 169–177.

Shaby, B. & Wells, M.T. (2010) Exploring an adaptive Metropolis algorithm. Technical report.

Stan Development Team (2020) RStan: the R interface to Stan. R package version 2.19.3. Available from: http://mc-stan.org/

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

Tierney, L. (1998) A note on Metropolis-Hastings kernels for general state spaces. *The Annals of Applied Probability*, 8, 1–9.

Woodard, D., Schmidler, S. & Huber, M. (2009) Sufficient conditions for torpid mixing of parallel and simulated tempering. *Electronic Journal of Probability*, 14, 780–804.

Zanella, G. (2020) Informed proposals for local MCMC in discrete spaces. *Journal of the American Statistical Association*, 115, 852–865.

SUPPORTING INFORMATION

Additional supporting information may be found in the online version of the article at the publisher's website.

How to cite this article: Livingstone, S. & Zanella, G. (2022) The Barker proposal: Combining robustness and efficiency in gradient-based MCMC. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 1–28. Available from: https://doi.org/10.1111/rssb.12482