Reinforcement Learning for Adaptive MCMC

Congye Wang Newcastle University
Wilson Chen University of Sydney
Heishiro Kanagawa Newcastle University
Chris. J. Oates Newcastle University

Abstract

An informal observation, made by several authors, is that the adaptive design of a Markov transition kernel has the flavour of a reinforcement learning task. Yet, to-date it has remained unclear how to exploit modern reinforcement learning technologies for adaptive MCMC. The aim of this paper is to set out a general framework, called *Reinforcement Learning Metropolis–Hastings*, that is theoretically supported and empirically validated. Our principal focus is on learning fast-mixing Metropolis–Hastings transition kernels, which we cast as deterministic policies and optimise via a policy gradient. Control of the learning rate provably ensures conditions for ergodicity are satisfied. The methodology is used to construct a gradient-free sampler that out-performs a popular gradient-free adaptive Metropolis–Hastings algorithm on $\approx 90\%$ of tasks in the `PosteriorDB` benchmark.

1 Introduction

A vast literature on algorithms, tips, and tricks is testament to the success of *Markov chain Monte Carlo* (MCMC), which remains the most popular approach to numerical approximation of probability distributions characterised up to an intractable normalisation constant. Yet the breadth of methodology also presents a difficulty in selecting an appropriate algorithm for a specific task. The goal of *adaptive* MCMC is to automate, as much as possible, the design of a fast-mixing Markov transition kernel. To achieve this, one alternates between observing the performance of the current transition kernel, and updating the transition kernel in a manner that is expected to improve its future performance [Andrieu and Thoms, 2008]. Though the online adaptation of a Markov transition kernel in principle sacrifices the ergodicy of MCMC, there are several ways to prove that ergodicity is in fact retained if the transition kernel converges fast enough (in an appropriate sense) to a sensible limit.

There is at least a superficial relationship between adaptive MCMC and *reinforcement learning* (RL), with both attempting to perform optimisation in a *Markov decision process* (MDP) context. Several authors have noted this similarity, yet to-date it has remained unclear whether state-of-the-art RL technologies can be directly exploited for adaptive MCMC. The demonstrated success of RL in tackling diverse MDPs, including autonomous driving [Kiran et al., 2021], gaming [Silver et al., 2016], and natural language processing [Shinn et al., 2023], suggests there is a considerable untapped potential if RL can be brought to bear on adaptive MCMC.

This paper sets out a general framework in Section 2, called *Reinforcement Learning Metropolis–Hastings*, in which the parameters of Metropolis–Hastings transition kernels are iteratively optimised along the MCMC sample path via a policy gradient. In particular, we explore transition kernels parametrised by neural networks, and leverage state-of-the-art deterministic policy gradient algorithms from RL to learn suitable parameters for the neural network. Despite the apparent complexity of the set-up, at least compared to more standard methods in adaptive MCMC, it is shown in Section 3 how control of the learning rate and gradient clipping can be used to provably guarantee that diminishing adaptation and containment conditions, which together imply ergodicity of the resulting Markov process, are satisfied. The methodology was objectively stress-tested, with results on the `PosteriorDB` benchmark reported in Section 4.

1.1 Related Work

Before presenting our methodology, we provide a brief summary of the extensive literature on adaptive MCMC (Section 1.1.1), and a comprehensive review of existing work at the interface of MCMC and RL (Section 1.1.2). Let $\mathcal{X}$ be a topological space equipped with the Borel σ-algebra; henceforth the measurability of relevant func-

Proceedings of the 28[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258. Copyright 2025 by the author(s).

tions and sets will always be assumed. For notation, we mainly work with densities throughout and use $p(\cdot)$ denote the density of the target distribution, with respect to an appropriate reference measure on $\mathcal{X}$.

1.1.1 Adaptive MCMC

For the most part, research into adaptive MCMC has focused on the adaptive design of a fast-mixing Metropolis–Hastings transition kernel [Haario et al., 2006, Roberts and Rosenthal, 2009], though the adaptive design of other classes of MCMC method, such as Hamiltonian Monte Carlo, have also been considered [Wang et al., 2013, Hoffman and Gelman, 2014, Christiansen et al., 2023]. Recall that Metropolis–Hastings refers to a Markov chain $(x_n)_{n\in\mathbb{N}} \subset \mathcal{X}$ such that, to generate x_{n+1} from x_n, we first simulate a candidate state $x^\star_{n+1} \sim q(\cdot|x_n)$, where the collection $\{q(\cdot|x) : x \in \mathcal{X}\}$ is called the *proposal*, and then set $x_{n+1} = x^\star_{n+1}$ with probability

$$\alpha(x_n, x^\star_{n+1}) := \min\left\{1, \frac{p(x^\star_{n+1})}{p(x_n)} \frac{q(x_n|x^\star_{n+1})}{q(x^\star_{n+1}|x_n)}\right\}, \quad (1)$$

else we set $x_{n+1} = x_n$. To develop an adaptive MCMC method in this context three main ingredients are required:

Performance Criterion The first ingredient is a criterion to be (approximately) optimised. Standard choices include: the negative correlation between consecutive states x_n and x_{n+1}, or between x_n and x_{n+l} for some lag l; the average return time to a given set; the expected squared jump distance [Pasarica and Gelman, 2010]; or the asymptotic variance associated to a function $f(x_n)$ of interest [Andrieu and Robert, 2001]. For Metropolis–Hastings chains specifically, the average acceptance rate is a criterion that is widely-used, but alternatives include criteria based on raw acceptance probabilities [Titsias and Dellaportas, 2019], and using a divergence between the proposal and the target distributions [Andrieu and Moulines, 2006, Dharamshi et al., 2023].

Candidate Transition Kernels The second ingredient is a set of candidate transition kernels, among which an adaptive MCMC method aims to identify an element that is optimal with respect to the performance criterion. In the Metropolis–Hastings context, a popular approach is to use previous samples $(x_i)_{i=1}^n$ to construct a rough approximation p_n to the target distribution p, and then to exploit p_n in the construction of a Metropolis–Hastings proposal. For example, p_n might be a Gaussian approximation based on the mean and covariance of the states previously visited, and the proposal may be p_n itself. Several authors have proposed more flexible approximation methods, such as a mixture of t-distributions [Tran et al., 2016], local polynomial approximation of the target [Conrad et al., 2016], and normalising flows [Gabrié et al., 2022]. Such methods can suffer from substantial autocorrelation during the warm-up phase, but once a good approximation has been learned, samples generated using this approach can be almost independent [Davis et al., 2022].

Mechanism for Adaptation The final ingredient is a mechanism to adaptively select a suitable transition kernel, based on the current sample path $(x_i)_{i=1}^n$, in such a manner that the ergodicity of the Markov chain is still assured. General sufficient conditions for ergodicity of adaptive MCMC have been obtained, and these can guide the construction of a mechanism for selecting a transition kernel [Atchadé and Rosenthal, 2005, Andrieu and Moulines, 2006, Atchade et al., 2011]. Some of the more recent research directions in adaptive MCMC include the use of Bayesian optimisation [Mahendran et al., 2012], incorporating global jumps between modes as they are discovered [Pompe et al., 2020], local adaptation of the step size in the *Metropolis-adjusted Langevin algorithm* (MALA) [Biron-Lattes et al., 2024], tuning gradient-based MCMC by directly minimising a divergence between p and the MCMC output [Coullon et al., 2023], and the online training of diffusion models to approximate the target [Hunt-Smith et al., 2024]. Our contribution will, in effect, demonstrate that state-of-the-art methods from RL can be added to this list.

1.1.2 Reinforcement Learning Meets MCMC

Consider a MDP with a *state* set $\mathcal{S}$, an *action* set $\mathcal{A}$, and an *environment* which determines both how the state s_n is updated to s_{n+1} in response to an action a_n, and the (scalar) *reward* r_n that resulted. *Reinforcement learning* refers to a broad class of methods that attempt to learn a *policy* $\pi : \mathcal{S} \to \mathcal{A}$, meaning a mechanism to select actions $a_n = \pi(s_n)$, which aims to maximise the expected cumulative reward [Kaelbling et al., 1996]. A useful taxonomy of modern RL algorithms is based on whether the policy π is deterministic or stochastic, and whether the action space is discrete or continuous [François-Lavet et al., 2018]. Techniques from deep learning are widely used in modern RL, for example in *Deep Q-Learning* (for discrete actions) [Mnih et al., 2015] and in *Deep Deterministic Policy Gradient* (for continuous actions) [Lillicrap et al., 2015]. The impressive performance of modern RL serves as strong motivation for investigating if and how techniques from RL can be harnessed for adaptive MCMC. Several authors have speculated on this possibility, but to-date the problem remained unsolved:

Policy Guided Monte Carlo is an approach developed to sample configurations from discrete lattice models, proposed in Bojesen [2018]. The approach identifies the state s_n with the state x_n of the Markov chain, the (stochastic) policy π with the proposal distribution q in Metropolis–Hastings MCMC, and the (discrete) action a_n with the candidate $x^\star_{n+1}$ that is being proposed. At first sight this seems natural, but we contend that this set-up is not a true MDP, because the action $x^\star_{n+1}$ contains insufficient information to determine the acceptance probability for the proposed state $x^\star_{n+1}$; to achieve that, we would also need to know the ratio of probabilities $q(x_n|x^\star_{n+1})/q(x^\star_{n+1}|x_n)$ appearing in (1). A possible mitigation is to restrict attention to symmetric proposals, for which this ratio becomes identically 1, but symmetry places a limitation on the performance of MCMC. As a result, this set-up does not fit well with the usual formulation of RL, and modern RL methods cannot be directly deployed.

An alternative set-up is to again associate the state s_n with the state x_n of the Markov chain, but now associate the action set $\mathcal{A}$ with a set of Markov transition kernels, so that the policy determines which transition kernel to use to move from the current state to the next. This approach was called *Reinforcement Learning Accelerated MCMC* in Chung et al. [2020], where a finite set of Markov transition kernels were developed for a particular multi-scale inverse problem, and called *Reinforcement Learning-Aided MCMC* in Wang et al. [2021], where RL was used to learn frequencies for updating the blocks of a random-scan Gibbs sampler. The main challenge in extending this approach to general-purpose adaptive MCMC is the difficulty in parametrising a collection of transition kernels in such a manner that the gradient-based policy search will work well; perhaps as a consequence, existing work did not leverage state-of-the-art techniques from RL. In addition, it appears difficult to establish conditions for ergodicity with this set-up, with these existing methods being only empirically validated. Our contribution is to propose an alternative set-up, where we parametrise the *proposal* in Metropolis–Hastings MCMC instead of the transition kernel, showing how this enables both of these open challenges to be resolved.

2 Methods

Recall that we work with densities[1] and associate $p(\cdot)$ with a probability measure on $\mathcal{X}$. The task is to generate samples from $p(\cdot)$ using adaptive MCMC. Let Φ be a topological space, and for each $\varphi \in \Phi$ let $\{q_\varphi(\cdot|x) : x \in \mathcal{X}\}$ denote a proposal. Simple proposals will serve as building blocks for constructing more sophisticated Markov transition kernels, as we explain next.

2.1 ϕ-Metropolis–Hastings

Suppose now that we are given a map $\phi : \mathcal{X} \to \Phi$. This mapping defines a Markov chain such that, if the current state is x_n, the next state x_{n+1} is generated using a Metropolis–Hastings transition kernel with proposal $q_{\phi(x_n)}(\cdot|x_n)$. In other words, we have a transition kernel

$$x^\star_{n+1} \sim q_{\phi(x_n)}(\cdot|x_n),$$
$$x_{n+1} \leftarrow \begin{cases} x^\star_{n+1} & \text{with probability } \alpha_\phi(x_n, x^\star_{n+1}) \\ x_n & \text{otherwise} \end{cases}$$

with a ϕ-dependent acceptance probability

$$\alpha_\phi(x_n, x^\star_{n+1}) := \min\left\{1, \frac{p(x^\star_{n+1})}{p(x_n)} \frac{q_{\phi(x^\star_{n+1})}(x_n|x^\star_{n+1})}{q_{\phi(x_n)}(x^\star_{n+1}|x_n)}\right\}.$$

The design of a fast-mixing Markov chain can then be formulated as the design of a suitable map ϕ, characterising the state-dependent proposal. The Markov chain associated to ϕ will be called *ϕ-MH* in the sequel. Two illustrative examples are presented, where in each case we suppose $p(\cdot)$ is a distribution with (for illustrative purposes, *known*) mean μ and standard deviation σ on $\mathcal{X} = \mathbb{R}$:

Example 1 (ϕ-MH based on symmetric random walk proposal)**.** *Consider the case where $q_\varphi(\cdot|x)$ is Gaussian with mean x and standard deviation φ, corresponding to a* symmetric random walk*. Then ϕ-MH with $\phi(x) = \sigma + |x - \mu|$ is a Metropolis–Hastings chain that attempts to take larger steps when the chain is further away from the mean μ of the target.*

Example 2 (ϕ-MH based on independence sampler proposal)**.** *Consider the case where $q_\varphi(\cdot|x)$ is Gaussian with mean φ and standard deviation σ, corresponding to an* independence sampler*. Then ϕ-MH with $\phi(x) = 2\mu - x$ is a Metropolis–Hastings chain that induces anti-correlation by promoting jumps from one side of μ to the other.*

These examples of ϕ-MH can be analysed using existing techniques and their ergodicity can be established, but for general ϕ we will require some regularity to ensure ϕ-MH generates samples from the correct target. Lemma 1 below establishes weak conditions[2] under

[1] This choice sacrifices generality and is not required for our methodology, but serves to make the presentation more transparent.

[2] The conclusion of Lemma 1 can be extended to general topological spaces $\mathcal{X}$, by noting that the arguments used in the proof are all topological, but for the present purposes such a generalisation was not pursued.

which ϕ-MH transition is p-invariant and *ergodic*, the latter understood in this paper to mean convergence in total variation of $\mathrm{Law}(x_n)$ to $p(\cdot)$ is guaranteed from any initial state $x_0 \in \mathcal{X}$.

Lemma 1 (Ergodicity of ϕ-MH; $\mathcal{X} = \mathbb{R}^d$)**.** *Let ϕ be continuous, and let both $x \mapsto p(x)$ and $(\varphi, x, y) \mapsto q_\varphi(x, y)$ be positive and continuous. Then ϕ-MH is p-invariant and ergodic.*

The proof is contained in Appendix A.1. Of course, a suitable map ϕ is unlikely to be known *a priori* since $p(\cdot)$ is intractable, so instead a suitable ϕ will need to be *learned*. Armed with a guarantee of correctness for ϕ-MH, we now seek to cast the learning of a suitable map ϕ as a problem that can be addressed using modern techniques from RL.

2.2 ϕ-Metropolis–Hastings as Reinforcement Learning

Our main methodological contribution is to set out a correct framework for casting adaptive MCMC as RL, and represents a fundamental departure from the earlier attempts described in Section 1.1.2. To cast the learning of the map ϕ appearing in ϕ-MH as an MDP, the state set $\mathcal{S}$, action set $\mathcal{A}$, and the environment each need to be specified:

State set $\mathcal{S}$ The *state* in our MDP set-up is

$$s_n = \begin{bmatrix} x_n \\ x^\star_{n+1} \end{bmatrix} \in \mathcal{X} \times \mathcal{X} =: \mathcal{S},$$

consisting of the current state x_n of the Markov chain and the proposed state $x^\star_{n+1}$, which will either be accepted or rejected.

Action set $\mathcal{A}$ The *action* set in our set-up is $\mathcal{A} = \Phi \times \Phi$. A map $\phi : \mathcal{X} \to \Phi$ induces a (deterministic) policy of the form

$$\pi(s_n) = a_n = \begin{bmatrix} \phi(x_n) \\ \phi(x^\star_{n+1}) \end{bmatrix} \in \mathcal{A}. \tag{2}$$

The motivation for this set-up is that the environment will need to compute not just the probability $q_{\phi(x_n)}(x^\star_{n+1}|x_n)$ of sampling the candidate $x^\star_{n+1}$, but also the reverse probability $q_{\phi(x^\star_{n+1})}(x_n|x^\star_{n+1})$, to calculate the overall acceptance probability $\alpha_\phi(x_n, x^\star_{n+1})$. The action a_n therefore contains all the information that the environment will need to evolve the Markov chain, as we explain next.

Environment The *environment*, given the current state s_n and action a_n, executes three tasks: First, an accept/reject decision is made so that $x_{n+1} = x^\star_{n+1}$ with probability $\alpha_\phi(x_n, x^\star_{n+1})$, else $x_{n+1} = x_n$. Second, the environment simulates $x^\star_{n+2} \sim q_{\phi(x_{n+1})}(\cdot|x_{n+1})$ and returns the updated state $s_{n+1} = [x_{n+1}, x^\star_{n+2}]$. This is possible since $\phi(x_{n+1})$ is equal to either $\phi(x_n)$ or $\phi(x^\star_{n+1})$, each of which were provided (via a_n) to the environment. Third, the environment computes a reward r_n. For the case $\mathcal{X} = \mathbb{R}^d$, we define our reward as

$$r_n = 2 \log \|x_n - x^\star_{n+1}\| + \log \alpha_\phi(x_n, x^\star_{n+1}),$$

which is the logarithm of the *expected squared jump distance* (ESJD) from x_n to x_{n+1}, where the expectation is computed with respect to the randomness in the accept/reject step.

Remark 1 (Choice of action)**.** *Could we instead define the action to be ϕ, so that at each iteration the policy picks a transition kernel? In principle yes, but then the action space would be high- or infinite-dimensional, severely increasing the difficulty of the RL task. Our set-up allows us to flexibly parametrise ϕ using a neural network, while ensuring the number of parameters in this network has no bearing on the dimension of the action set.*

Remark 2 (Choice of reward)**.** *The ESJD is a popular criterion for use in adaptive MCMC, due to its close relationship with mixing times [Sherlock and Roberts, 2009] and the ease with which it can be computed. Our initial investigations found the logarithm of ESJD to be more useful for RL, as it provides the reward with a greater dynamic range to distinguish bad policies from very bad policies, leading to improved performance of methods based on policy gradient (see Section 2.3).*

2.3 Learning ϕ via Policy Gradient

The rigorous formulation of ϕ-MH as an MDP enables modern techniques from RL to be immediately brought to bear on adaptive MCMC. Assuming the Markov chain is initialised at stationarity, the standard RL objective (i.e. discounted cumulative reward) reduces to the expected reward at stationarity; $J(\phi) := \mathbb{E}_\phi[r]$, where the state s is sampled from the stationary distribution of the MDP and the action $a = \pi(s)$ is determined by the ϕ-MH policy π in (2). In practice one restricts attention to a parametric family ϕ_θ for some $\theta \in \mathbb{R}^p$, $p \in \mathbb{N}$. To optimise $J(\phi_\theta)$ we employ a (deterministic) *policy gradient* method [Silver et al., 2014], meaning in our case that we alternate between updating the Markov chain from x_n to x_{n+1} using ϕ_θ-MH with $\theta = \theta_n$ fixed, and updating the parameter θ using approximate gradient ascent

$$\theta_{n+1} \leftarrow \theta_n + \alpha_n \, \nabla_\theta J_n(\phi_\theta)|_{\theta=\theta_n} \tag{3}$$

where $\alpha_n \geq 0$ is called a *learning rate*. Here $\nabla_\theta J_n$ indicates an approximation to the *policy gradient* $\nabla_\theta J$,

Algorithm 1 Reinforcement Learning Metropolis–Hastings (RLMH)

Require: $x_0 \in \mathcal{X}$, $\theta_0 \in \mathbb{R}^p$, gradient threshold $\tau > 0$, learning rate $(\alpha_n)_{n\geq 0} \subset [0, \infty)$
Ensure: $\sum_{n\geq 0} \alpha_n < \infty$
  for $n = 0, 1, 2, \ldots$ **do**
    $x^\star_{n+1} \sim q_{\phi_{\theta_n}(x_n)}(\cdot|x_n)$ ▷ propose next state
    $x_{n+1} \leftarrow x^\star_{n+1}$ with probability $\alpha_{\phi_{\theta_n}}(x_n, x^\star_{n+1})$, else $x_{n+1} \leftarrow x_n$ ▷ accept/reject
    $g_n \leftarrow \nabla_\theta J_n(\phi_\theta)|_{\theta=\theta_n}$ ▷ approximate policy gradient
    if $\|g_n\| > \tau$ **then** $g_n \leftarrow \tau g_n / \|g_n\|$ ▷ gradient clipping
    $\theta_{n+1} \leftarrow \theta_n + \alpha_n g_n$ ▷ policy update
  end for

which may be constructed using any of the random variables generated up to that point. A considerable amount of research effort in RL has been devoted to approximating the policy gradient, and for the experiments reported in Section 4 we used the *deep deterministic policy gradient* (DDPG) method of Lillicrap et al. [2015]. DDPG is suitable for deterministic policies and a continuous action set, and operates by training a *critic* based on experience stored in a *replay buffer* to enable a stable approximation to the policy gradient. Since the details of DDPG are not a novel contribution of our work, we reserve them for Appendix B.

The proposed *Reinforcement Learning Metropolis–Hastings* (RLMH) method is stated in Algorithm 1, and is compatible with *any* approach to approximation of the policy gradient, not just DDPG. Indeed, the theoretical results that we present in Section 3 leverage the summability of the learning rate sequence $(\alpha_n)_{n\geq 0}$ and norm-based *gradient clipping* in Algorithm 1 to prove that sufficient conditions for ergodicity are satisfied.

Remark 3 (On-policy requirement)**.** *Our set-up is* on-policy*, meaning that only actions specified by the policy are used to evolve the Markov chain; this is necessary for maintaining detailed balance in ϕ-MH, which in turn ensures the Markov transition kernels are p-invariant. On the other hand, off-policy exploration can be concurrently conducted to aid in approximating the policy gradient [for example, as training data for the critic in DDPG; Ladosz et al., 2022].*

3 Theoretical Assessment

Such is the generality of ϕ-MH that it is possible to develop diverse gradient-free and gradient-based adaptive MCMC algorithms using RLMH. Indeed, the building block proposals for ϕ-MH could range from simple proposals (e.g. random walks) to complicated proposals (e.g. involving higher-order gradients of the target). This section establishes sufficient conditions for the ergodicity of RLMH, and to this end it is necessary to be specific about what building block proposals will be employed. The remainder of this paper develops a *gradient-free* sampling scheme in detail; our motivation is a tractable end-to-end theoretical analysis, guaranteeing correctness of the proposed method.

An accessible introduction to the theory of adaptive MCMC is provided in Andrieu and Moulines [2006]. At a high-level, the parameter θ of a Markov transition kernel is being allowed to depend on the sample path, i.e. $\theta_n \equiv \theta_n(x_0, \ldots, x_n)$; intuitively, the hope is that θ_n will converge to a fixed limiting value $\theta_\star$, and that ergodicity properties of the adaptive chain will be inherited from those of the chain associated with $\theta_\star$. However, carried out naïvely, the sequence θ_n can fail to converge, or converge to a value $\theta_\star$ for which the chain is no longer ergodic; see Section 2 in Andrieu and Thoms [2008]. In brief, the first issue can be solved by *diminishing adaptation*; ensuring that the differences between θ_n and θ_{n+1} are decreasing, or even zero after a finite number of iterations have been performed. The second issue can be resolved by *containment*; restricting θ to e.g. compact subsets for which all transition kernels are well-behaved.

Our strategy below is to force diminishing adaptation via control of the learning rate and gradient clipping, and then to demonstrate containment via careful parametrisation of the ϕ-MH Markov transition kernel. The particular form of containment that we consider is motivated by the analytical framework of Roberts and Rosenthal [2007], and requires us to establish that ϕ-MH is *simultaneously strongly aperiodically geometrically ergodic* (SSAGE); this condition is precisely stated in Definition 1. For notation, recall that the *transition kernel* $P(x, \cdot)$ of a Markov chain specifies the distribution of x_{n+1} if the current state is $x_n = x$. For a function $f : \mathcal{X} \to \mathbb{R}$ we write $(Pf)(x) = \int f(y)P(x, \mathrm{d}y)$. Let 1_S denote the indicator function associated to a set S.

Definition 1 (SSAGE; Roberts and Rosenthal, 2007)**.** *A family of p-invariant Markov transition kernels $\{P_\theta\}_{\theta\in\Theta}$ is* SSAGE *if there exists $S \subset \mathcal{X}$, $V : \mathcal{X} \to [1, \infty)$, $\delta > 0$, $\lambda < 1$, and $b < \infty$, such that $\sup_{x\in S} V(x) < \infty$ and the following hold:*

1. (minorisation) *For each $\theta \in \Theta$, there exists a probability measure ν_θ on S with $P_\theta(x, \cdot) \geq \delta\nu_\theta(\cdot)$ for all $x \in S$.*
2. (simultaneous drift) $(P_\theta V)(x) \leq \lambda V(x) + b1_S(x)$ *for all $x \in \mathcal{X}$ and $\theta \in \Theta$.*

To the best of our knowledge, existing SSAGE results for Metropolis–Hastings focused on the case of a random walk proposal [Roberts and Tweedie, 1996b, Roberts and Rosenthal, 2007, Jarner and Hansen, 2000, Bai et al., 2009, Saksman and Vihola, 2010]. However, the the proposal of ϕ-MH can in principle be (much) more sophisticated than a random walk. Our first contribution is therefore to generalise existing sufficient conditions for SSAGE beyond the case of a random walk proposal, and in this sense Theorem 1 below may be of independent interest.

Let $\mathcal{P}(\mathbb{R}^d)$ denote the set of probability distributions $p(\cdot)$ on $\mathbb{R}^d$ that are positive, continuously differentiable, and *sub-exponential*, i.e.

$$\lim_{\|x\|\to\infty} n(x) \cdot (\nabla \log p)(x) = -\infty \tag{4}$$

where $n(x) := x/\|x\|$. To simplify notation, we use $q_\theta(\cdot|x)$ as shorthand for $q_{\phi_\theta(x)}(\cdot|x)$ and $\alpha_\theta(x, y)$ as shorthand for $\alpha_{\phi_\theta}(x, y)$. Let $A_\theta(x) = \{y \in \mathbb{R}^d : \alpha_\theta(x, y) = 1\}$ denote the region where proposals are always accepted, and let $\partial A^\delta_\theta(x) := \{y + sn(y) : y \in \partial A_\theta(x), |s| \leq \delta\}$ denote a tube of radial width $\delta > 0$ around the boundary $\partial A_\theta(x)$ of $A_\theta(x)$.

Theorem 1 (SSAGE for general Metropolis–Hastings)**.** *Let $p \in \mathcal{P}(\mathbb{R}^d)$. Consider a family of p-invariant Metropolis–Hastings transition kernels $\{P_\theta\}_{\theta\in\Theta}$, where P_θ corresponds to a proposal $\{q_\theta(\cdot|x) : x \in \mathbb{R}^d\}$, and denote $Q_\theta(S|x) := \int_S q_\theta(y|x)\mathrm{d}y$ for $S \subseteq \mathbb{R}^d$. Let $(\theta, x, y) \mapsto q_\theta(y|x)$ be positive and continuous, and assume further that:*

(i) (quasi-symmetry)
$$\rho := \sup_{x,y\in\mathbb{R}^d,\theta\in\Theta} q_\theta(y|x)/q_\theta(x|y) < \infty.$$
(ii) (regular acceptance boundary) *For all $\epsilon > 0$ there exists $\delta > 0$ and $R > 0$ such that, for all $\|x\| \geq R$ and $\theta \in \Theta$, we have $Q_\theta(\partial A^\delta_\theta(x)|x) < \epsilon$.*
(iii) (minimum performance level)
$$\liminf_{\|x\|\to\infty} \inf_{\theta\in\Theta} Q_\theta(A_\theta(x)|x) > 0.$$

Then $\{P_\theta\}_{\theta\in\Theta}$ is SSAGE.

The proof is contained in Appendix A.2, and follows the general strategy used to establish SSAGE for random walk proposals in Jarner and Hansen [2000], but with additional technical work to relax the random walk requirement (which corresponds to the case $\rho = 1$).

Quasi-symmetry imposes a non-trivial constraint on the form of the ϕ-MH chains that can be analysed. For both our theoretical and empirical assessment we focus on a setting that can be considered a multi-dimensional extension of Example 2, where the mean φ of the proposal is specified as the output of a map $\phi_\theta(\cdot)$ with parameters $\theta \in \mathbb{R}^p$. To be precise, let the set of $d \times d$ symmetric positive-definite matrices be denoted S^+_d and, for $\Sigma \in \mathrm{S}^+_d$, let $\|x\|_{1,\Sigma} := \|\Sigma^{-1/2}x\|_1$ denote an associated norm on $\mathbb{R}^d$. Then, for the remainder, we consider a Laplace proposal

$$q_\theta(y|x) \propto \exp\left(-\|y - \phi_\theta(x)\|_{1,\Sigma}\right), \tag{5}$$

for which sufficient conditions for ergodicity, including quasi-symmetry, can be shown to hold under appropriate assumptions on $\{\phi_\theta\}_{\theta\in\mathbb{R}^p}$. This is the content of Theorem 2, which we present next.

To set notation, recall that a function $f : \mathbb{R}^p \to \mathbb{R}$ is said to be *locally bounded* if, for all $\theta \in \mathbb{R}^p$, there is an open neighbourhood of θ on which f is bounded. Such a function is said to be *locally Lipschitz* at $\theta \in \mathbb{R}^p$ if

$$\mathrm{LocLip}_\theta(f) := \limsup_{\vartheta\to\theta} \frac{|f(\vartheta) - f(\theta)|}{\|\vartheta - \theta\|} < \infty,$$

and, when this is the case, $\mathrm{LocLip}_\theta(f)$ is called the *local Lipschitz constant* of f at $\theta \in \mathbb{R}^p$. Such a function is called *Lipschitz* if $\mathrm{Lip}(f) := \sup_\theta \mathrm{LocLip}_\theta(f) < \infty$ and, when this is the case, $\mathrm{Lip}(f)$ is called the *Lipschitz constant*. Let $\mathcal{P}_0(\mathbb{R}^d) \subset \mathcal{P}(\mathbb{R}^d)$ denote the subset of distributions for which the *interior cone condition*

$$\limsup_{\|x\|\to\infty} n(x) \cdot \frac{(\nabla p)(x)}{\|(\nabla p)(x)\|} < 0$$

is also satisfied.

Theorem 2 (Ergodicity of RLMH)**.** *Let $p \in \mathcal{P}_0(\mathbb{R}^d)$, $\Sigma \in \mathrm{S}^+_d$, $\Theta = \mathbb{R}^p$. Consider ϕ-MH with proposal (5). Assume that each $x \mapsto \phi_\theta(x)$ is Lipschitz and $\sup_{x\in\mathbb{R}^d} \|x - \phi_\theta(x)\| < \infty$. Further assume the parametrisation of ϕ_θ in terms of θ is* regular*, meaning that the following maps are well-defined and locally bounded:*

(i) $\theta \mapsto \sup_{x\in\mathbb{R}^d} \|x - \phi_\theta(x)\|$;
(ii) $\theta \mapsto \mathrm{Lip}(x \mapsto \phi_\theta(x))$;
(iii) $\theta \mapsto \sup_{x\in\mathbb{R}^d} \mathrm{LocLip}_\theta(\vartheta \mapsto \phi_\vartheta(x))$.

Then RLMH in Algorithm 1 is p-invariant and ergodic, irrespective of the approach used to approximate the policy gradient.

The proof is contained in Appendix A.3, where containment is established using Theorem 1 and diminishing adaptation follows from control of the learning rate and

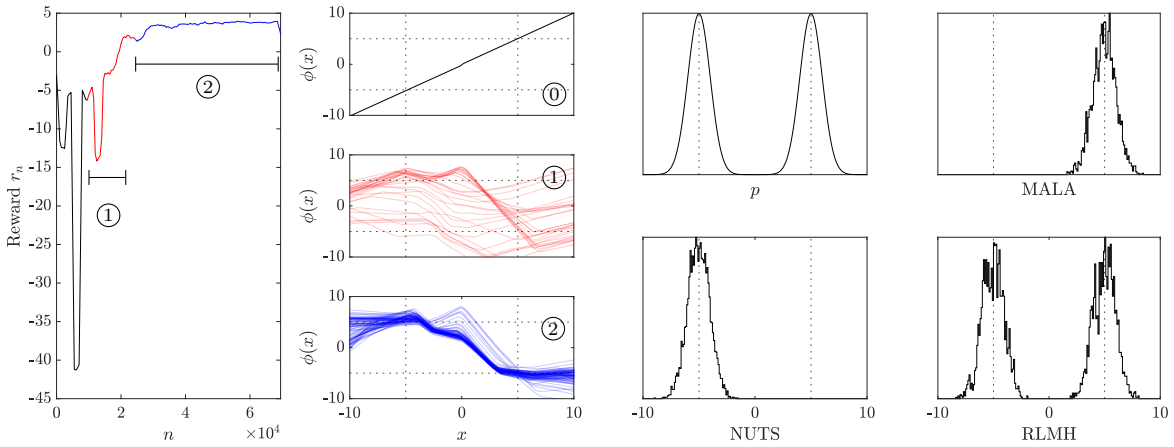


Figure 1: *Reinforcement Learning Metropolis–Hastings* (RLMH), illustrated. Here the task is to sample from the Gaussian mixture model $p(\cdot)$ whose equally-weighted components are $\mathcal{N}(\pm 5, 1)$. Left: The reward sequence $(r_n)_{n\geq 0}$, where r_n is the logarithm of the expected squared jump distance corresponding to iteration n of RLMH. Middle: Proposal mean functions $x \mapsto \phi(x)$, at initialisation in ⓪, and corresponding to the rewards indicated in ① and ②. Right: The density $p(\cdot)$, and histograms of the last $n = 5,000$ samples produced using MALA, *no U-turn sampler* (NUTS), and RLMH. [A smoothing window of length 5 was applied to the reward sequence to improve clarity of this plot.]

gradient clipping, as in Algorithm 1. The conditions (i)-(iii) in Theorem 2 are satisfied in the experiments reported in Section 4 through careful choice of the family of maps $\{\phi_\theta\}_{\theta\in\mathbb{R}^p}$; full details are reserved for Appendix B.3.

4 Empirical Assessment

The aim of this section is to illustrate the behaviour of the gradient-free RLMH algorithm that was analysed in Section 3, and to compare performance against an established and popular gradient-free adaptive Metropolis–Hastings algorithm using a community benchmark.

Implementation of RLMH An established adaptive sampling scheme was used to *warm start* RLMH. For the purposes of this paper, we first run $m = 10^4$ iterations $(x_i)_{i=-m+1}^{0}$ of a popular *adaptive random walk Metropolis–Hastings* (ARWMH) algorithm based on a gradient-free random walk proposal, and let Σ denote an estimate of the covariance of $p(\cdot)$ so-obtained; full details are contained in Appendix B.2. The matrix Σ is then used to define the norm appearing in (5), and the final iteration x_0 is used to initialise RLMH. To control for the benefit of a warm start, we use ARWMH as a comparator in the subsequent empirical assessment. The proposal mean ϕ_θ was taken as a neural network designed to satisfy the assumptions of Theorem 2; see Appendix B.3. Sensitivity to the neural architecture was explored in Appendix C.1. The policy gradient was approximated using DDPG with default settings as described in Appendix B.4. Code to reproduce our experiments is available at `https://github.com/congyewang/Reinforcement-Learning-for-Adaptive-MCMC`.

Illustration: Learning a Global Proposal To understand the behaviour of RLMH, consider the (toy) task of sampling from a two-component Gaussian mixture model in dimension $d = 1$ as presented in Figure 1. The proposal mean was initialised such that $\phi_\theta(x) \approx x$, corresponding to a random walk; this is visually represented in panel ⓪. The sequence of rewards $(r_n)_{n\geq 0}$ obtained during training is displayed in the left panel. Considerable exploration is observed in the period from initialisation to $n = 10^4$, while between 10^4 and 2.5×10^4 the policy gradient directs the algorithm to an effective policy; see panel ①. From then onward, the policy appears to have essentially converged; see panel ②. The final learned policy performs *global mode-hopping*; if the chain is currently in the effective support of one mixture component, then it will propose to move to the other mixture component. This behaviour is consistent with seeking a large ESJD. The samples generated from RLMH are displayed as a histogram in the right hand panel of Figure 1, and are seen to form a good approximation of the Gaussian mixture target. These can be visually compared against samples generated using MALA [Roberts and Tweedie, 1996a] and the *no U-turn sampler* (NUTS) [Hoffman and Gelman, 2014], each of which struggled to escape from the mixture component where the Markov chain was initialised. Further illustrations of RLMH, which can be visualised in dimensions $d \in \{1, 2\}$, can be found

in Appendix C.2.

Benchmarking on PosteriorDB `PosteriorDB` is a community benchmark for performance assessment in Bayesian computation, consisting of a collection of posteriors to be numerically approximated [Magnusson et al., 2022]. Here we use `PosteriorDB` to compare gradient-free ARWMH (using default settings detailed in Appendix B.2) against RLMH (default settings in Appendix B.4). A plethora of other algorithms exist, but ARWMH represents arguably the most widely-used gradient-free algorithm for adaptive MCMC. As an additional point of reference, we also present results for an adaptive version of MALA (AMALA; Appendix C.3), for which gradient information on $p(\cdot)$ is required. For higher-dimensional problems gradient information is usually essential. For assessment purposes, RLMH was run for $n = 5 \times 10^4$ iterations, while ARWMH and AMALA were each run for $n = 6 \times 10^4$ iterations; this equates computational cost when one accounts for the warm start of RLMH. At the end, all algorithms were then run for an additional 5×10^3 iterations with no adaptation permitted, and it was on these final samples that performance was assessed. Two metrics are reported: (i) ESJD, and (ii) the *maximum mean discrepancy* (MMD) relative to a gold-standard provided as part of `PosteriorDB`. Both performance metrics are precisely defined in Appendix C.4. Partial results are shown in Table 2, and full results are provided in Appendix C.5. These results show the gradient-free version of RLMH out-performed the natural gradient-free comparator, ARWMH, on 86% of tasks in terms of ESJD, and 93% of tasks in terms of MMD. Remarkably, the gradient-free version of RLMH also out-performed AMALA on the majority of low-dimensional tasks, while AMALA demonstrated predictably superior performance on higher-dimensional tasks where gradient information is well-known to be essential. RLMH and AMALA both failed on the most challenging 66-dimensional task, with RLMH converging to a policy for which all subsequent samples were rejected.

5 Discussion

This paper provided, for the first time, a correct framework that enables modern RL to be brought to bear on adaptive MCMC. Though the framework is general, for the purposes of end-to-end theoretical analysis we focused on a gradient-free sampling algorithm whose state-dependent proposal mean function is actively learned. Even in this context, an astonishing level of performance was observed on the `PosteriorDB` benchmark when we consider that we did not exploit gradient information on the target, and that an off-the-shelf implementation of DDPG was used. Of course, gradient-free samplers are limited to tasks that are low-dimensional, and a natural next step is to investigate the extent to which performance can be improved by exploiting gradient information in the proposal.

More broadly, our contribution comes at a time of increasing interest in exploiting RL for adaptive Monte Carlo methods, with recent work addressing adaptive importance sampling [El-Laham and Bugallo, 2021] and the adaptive design of control variates [Bras and Pagès, 2023]. It would be interesting to see whether the approach we have set out can be extended to the design of other related Monte Carlo algorithms, such as multiple-try Metropolis [Liu et al., 2000], and delayed acceptance MCMC [Christen and Fox, 2005].

Acknowledgements CW was supported by the China Scholarship Council under Grant Number 202208890004. HK and CJO were supported by EP/W019590/1.

References

C. Andrieu and É. Moulines. On the ergodicity properties of some adaptive MCMC algorithms. *The Annals of Applied Probability*, 16(1):1462–1505, 2006.

C. Andrieu and C. P. Robert. Controlled MCMC for optimal sampling. Technical Report 33, Center for Research in Economics and Statistics, 2001.

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

Y. Atchade, G. Fort, E. Moulines, and P. Priouret. *Adaptive Markov chain Monte Carlo: Theory and methods*. Bayesian Time Series Models. Cambridge University Press, 2011.

Y. F. Atchadé and J. S. Rosenthal. On adaptive Markov chain Monte Carlo algorithms. *Bernoulli*, 11(5):815–828, 2005.

Y. Bai, G. O. Roberts, and J. S. Rosenthal. On the containment condition for adaptive Markov chain Monte Carlo algorithms. Technical report, University of Warwick, 2009.

M. Biron-Lattes, N. Surjanovic, S. Syed, T. Campbell, and A. Bouchard-Côté. autoMALA: Locally adaptive Metropolis-adjusted Langevin algorithm. In *Proceedings of the 27th International Conference on Artificial Intelligence and Statistics*, pages 4600–4608. PMLR, 2024.

T. A. Bojesen. Policy-guided Monte Carlo: Reinforcement-learning Markov chain dynamics. *Physical Review E*, 98(6):063303, 2018.

P. Bras and G. Pagès. Policy gradient optimal correlation search for variance reduction in Monte Carlo

Task	d	RLMH		ARWMH		AMALA	
		ESJD	MMD	ESJD	MMD	ESJD	MMD
earnings-earn_height	3	**4.5(0.6)E3**	**1.8(0.1)E-1**	2.1(0.0)E3	1.5(0.0)E0	1.3(0.9)E3	1.8(0.3)E0
earnings-log10earn_height	3	**1.4(0.0)E-1**	**1.6(0.0)E-1**	4.4(0.0)E-2	1.4(0.0)E0	1.4(0.0)E-1	1.6(0.0)E-1
earnings-logearn_height	3	**3.2(0.0)E-1**	**1.6(0.0)E-1**	1.0(0.0)E-1	1.5(0.0)E0	3.3(0.0)E-1	1.7(0.0)E-1
gp_pois_regr-gp_regr	3	**3.7(0.1)E-1**	**1.2(0.0)E-1**	1.2(0.0)E-1	1.1(0.0)E0	3.6(0.0)E-1	1.2(0.0)E-1
kidiq-kidscore_momhs	3	**1.3(0.2)E0**	**1.5(0.0)E-1**	7.2(0.0)E-1	1.3(0.0)E0	2.3(0.0)E0	1.4(0.0)E-1
kidiq-kidscore_momiq	3	**3.6(0.2)E0**	**1.7(0.0)E-1**	1.3(0.0)E0	1.5(0.0)E0	4.2(0.0)E0	1.6(0.0)E-1
kilpisjarvi_mod-kilpisjarvi	3	**1.3(0.1)E1**	**1.7(0.0)E-1**	6.5(0.0)E0	1.5(0.0)E0	6.1(3.1)E0	1.2(0.2)E0
mesquite-logmesquite_logvolume	3	**1.2(0.0)E-1**	**1.3(0.0)E-1**	3.7(0.0)E-2	1.1(0.0)E0	1.1(0.0)E-1	1.3(0.0)E-1
arma-arma11	4	**6.4(0.0)E-2**	**1.2(0.0)E-1**	1.9(0.0)E-2	1.1(0.0)E0	3.7(1.0)E-2	9.2(3.3)E-1
earnings-logearn_height_male	4	**4.1(0.1)E-1**	**1.6(0.0)E-1**	1.2(0.0)E-1	1.5(0.0)E0	4.2(0.1)E-1	1.6(0.0)E-1
earnings-logearn_logheight_male	4	**1.7(0.1)E0**	**1.6(0.0)E-1**	5.3(0.0)E-1	1.5(0.0)E0	1.8(0.0)E0	1.6(0.0)E-1
garch-garch11	4	**8.0(0.2)E-1**	**1.4(0.0)E-1**	2.8(0.0)E-1	1.2(0.0)E0	7.1(0.1)E-1	1.4(0.0)E-1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
nes2000-nes	10	**4.2(0.1)E-1**	**1.2(0.0)E-1**	1.6(0.0)E-1	9.9(0.0)E-1	6.5(0.1)E-1	1.1(0.0)E-1
gp_pois_regr-gp_pois_regr	13	2.6(1.4)E-2	1.5(0.2)E0	**1.9(0.0)E-1**	**1.2(0.0)E0**	4.9(0.2)E-1	1.1(0.0)E-1
diamonds-diamonds	26	4.1(2.1)E-4	2.0(0.1)E0	**7.6(0.5)E-2**	**1.5(0.0)E0**	3.4(0.4)E-1	3.1(2.0)E-1
mcycle_gp-accel_gp	66	0	1.9(0.0)E0	**3.2(0.2)E-1**	**1.3(0.0)E0**	0	1.8(0.0)E0

Table 1: Benchmarking using `PosteriorDB`. Here, we compared a gradient-free version of RLMH to the gradient-free ARWMH, and also the gradient-based *adaptive Metropolis-adjusted Langevin algorithm* (AMALA). Performance was measured using the *expected squared jump distance* (ESJD) and the *maximum mean discrepancy* (MMD), and $d = \dim(\mathcal{X})$. Results are based on an average of 10 replicates, with standard errors (in parentheses) reported. The best performing gradient-free method is highlighted in **bold**. Shaded rows indicate situations where RLMH out-performed *adaptive Metropolis-adjusted Langevin algorithm* (AMALA) for either ESJD or *maximum mean discrepancy* (MMD). [Table shortened due to space constraint; full results can be found in Appendix C.5.]

simulation and maximum optimal transport. *arXiv preprint arXiv:2307.12703*, 2023.

J. A. Christen and C. Fox. Markov chain Monte Carlo using an approximation. *Journal of Computational and Graphical statistics*, 14(4):795–810, 2005.

H. Christiansen, F. Errica, and F. Alesiani. Self-tuning Hamiltonian Monte Carlo for accelerated sampling. *The Journal of Chemical Physics*, 159(23), 2023.

E. Chung, Y. Efendiev, W. T. Leung, S.-M. Pun, and Z. Zhang. Multi-agent reinforcement learning accelerated MCMC on multiscale inversion problem. *arXiv preprint arXiv:2011.08954*, 2020.

P. R. Conrad, Y. M. Marzouk, N. S. Pillai, and A. Smith. Accelerating asymptotically exact MCMC for computationally intensive models via local approximations. *Journal of the American Statistical Association*, 111(516):1591–1607, 2016.

J. Coullon, L. South, and C. Nemeth. Efficient and generalizable tuning strategies for stochastic gradient MCMC. *Statistics and Computing*, 33(3):66, 2023.

A. D. Davis, Y. Marzouk, A. Smith, and N. Pillai. Rate-optimal refinement strategies for local approximation MCMC. *Statistics and Computing*, 32(4):60, 2022.

B. Delyon and A. Juditsky. Accelerated stochastic approximation. *SIAM Journal on Optimization*, 3 (4):868–881, 1993.

A. Dharamshi, V. Ngo, and J. S. Rosenthal. Sampling by divergence minimization. *Communications in Statistics*, pages 1–25, 2023.

Y. El-Laham and M. F. Bugallo. Policy gradient importance sampling for Bayesian inference. *IEEE Transactions on Signal Processing*, 69:4245–4256, 2021.

V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare, and J. Pineau. An introduction to deep reinforcement learning. *Foundations and Trends® in Machine Learning*, 11(3-4):219–354, 2018.

M. Gabrié, G. M. Rotskoff, and E. Vanden-Eijnden. Adaptive Monte Carlo augmented with normalizing flows. *Proceedings of the National Academy of Sciences*, 119(10):e2109420119, 2022.

D. Garreau, W. Jitkrittum, and M. Kanagawa. Large sample analysis of the median heuristic. *arXiv preprint arXiv:1707.07269*, 2017.

A. Gelman, W. R. Gilks, and G. O. Roberts. Weak convergence and optimal scaling of random walk Metropolis algorithms. *The Annals of Applied Probability*, 7(1):110–120, 1997.

H. Haario, M. Laine, A. Mira, and E. Saksman. DRAM: Efficient adaptive MCMC. *Statistics and Computing*, 16:339–354, 2006.

M. D. Hoffman and A. Gelman. The No-U-Turn Sampler: Adaptively setting path lengths in Hamiltonian

Monte Carlo. *Journal of Machine Learning Research*, 15(1):1593–1623, 2014.

N. T. Hunt-Smith, W. Melnitchouk, F. Ringer, N. Sato, A. W. Thomas, and M. J. White. Accelerating Markov chain Monte Carlo sampling with diffusion models. *Computer Physics Communications*, 296: 109059, 2024.

S. F. Jarner and E. Hansen. Geometric ergodicity of Metropolis algorithms. *Stochastic Processes and Their Applications*, 85(2):341–361, 2000.

L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

B. R. Kiran, I. Sobh, V. Talpaert, P. Mannion, A. A. Al Sallab, S. Yogamani, and P. Pérez. Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(6):4909–4926, 2021.

P. Ladosz, L. Weng, M. Kim, and H. Oh. Exploration in deep reinforcement learning: A survey. *Information Fusion*, 85:1–22, 2022.

T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

J. S. Liu, F. Liang, and W. H. Wong. The multiple-try method and local optimization in Metropolis sampling. *Journal of the American Statistical Association*, 95(449):121–134, 2000.

S. Livingstone and G. Zanella. The Barker proposal: Combining robustness and efficiency in gradient-based MCMC. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 84(2):496–523, 2022.

M. Magnusson, P. Bürkner, and A. Vehtari. `PosteriorDB`: A set of posteriors for Bayesian inference and probabilistic programming. 2022. URL `https://github.com/stan-dev/posteriordb`.

N. Mahendran, Z. Wang, F. Hamze, and N. De Freitas. Adaptive MCMC with Bayesian optimization. In *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics*, pages 751–760. PMLR, 2012.

K. L. Mengersen and R. L. Tweedie. Rates of convergence of the Hastings and Metropolis algorithms. *The Annals of Statistics*, 24(1):101–121, 1996.

V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

C. Pasarica and A. Gelman. Adaptively scaling the Metropolis algorithm using expected squared jumped distance. *Statistica Sinica*, pages 343–364, 2010.

M. Plappert, R. Houthooft, P. Dhariwal, S. Sidor, R. Y. Chen, X. Chen, T. Asfour, P. Abbeel, and M. Andrychowicz. Parameter space noise for exploration. In *Proceedings of the 6th International Conference on Learning Representations*, 2018.

E. Pompe, C. Holmes, and K. Łatuszyński. A framework for adaptive MCMC targeting multimodal distributions. *Annals of Statistics*, 48(5):2930–2952, 2020.

G. O. Roberts and J. S. Rosenthal. Optimal scaling of discrete approximations to Langevin diffusions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 60(1):255–268, 1998.

G. O. Roberts and J. S. Rosenthal. Coupling and ergodicity of adaptive Markov chain Monte Carlo algorithms. *Journal of Applied Probability*, 44(2): 458–475, 2007.

G. O. Roberts and J. S. Rosenthal. Examples of adaptive MCMC. *Journal of Computational and Graphical Statistics*, 18(2):349–367, 2009.

G. O. Roberts and R. Tweedie. Exponential convergence of Langevin diffusions and their discrete approximation. *Bernoulli*, 2:341–363, 1996a.

G. O. Roberts and R. L. Tweedie. Geometric convergence and central limit theorems for multidimensional Hastings and Metropolis algorithms. *Biometrika*, 83(1):95–110, 1996b.

E. A. Roualdes, B. Ward, B. Carpenter, A. Seyboldt, and S. D. Axen. BridgeStan: Efficient in-memory access to the methods of a stan model. *Journal of Open Source Software*, 8(87):5236, 2023.

E. Saksman and M. Vihola. On the ergodicity of the adaptive Metropolis algorithm on unbounded domains. *Annals of Applied Probability*, 20(6):2178–2203, 2010.

C. Sherlock and G. O. Roberts. Optimal scaling of the random walk Metropolis on elliptically symmetric unimodal targets. *Bernoulli*, 15(3):774–798, 2009.

N. Shinn, F. Cassano, A. Gopinath, K. Narasimhan, and S. Yao. Reflexion: Language agents with verbal reinforcement learning. In *Proceedings of the 36th Conference on Neural Information Processing Systems*, 2023.

D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller. Deterministic policy gradient algorithms. In *Proceedings of the 31st International Conference on Machine Learning*, pages 387–395. PMLR, 2014.

D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

L. Tierney. Markov chains for exploring posterior distributions. *The Annals of Statistics*, 22(4):1701–1728, 1994.

M. Titsias and P. Dellaportas. Gradient-based adaptive Markov chain Monte Carlo. In *Proceedings of the 32nd Conference on Neural Information Processing Systems*, 2019.

M.-N. Tran, M. K. Pitt, and R. Kohn. Adaptive Metropolis–Hastings sampling using reversible dependent mixture proposals. *Statistics and Computing*, 26(1-2):361–381, 2016.

C. Wang, Y. Chen, H. Kanagawa, and C. J. Oates. Stein π-importance sampling. In *Proceedings of the 37th Conference on Neural Information Processing Systems*, 2023.

Z. Wang, S. Mohamed, and N. Freitas. Adaptive Hamiltonian and Riemann manifold Monte Carlo. In *Proceedings of the 30th International Conference on Machine Learning*, pages 1462–1470. PMLR, 2013.

Z. Wang, Y. Xia, S. Lyu, and C. Ling. Reinforcement learning-aided Markov chain Monte Carlo for lattice Gaussian sampling. In *Proceedings of the IEEE Information Theory Workshop*. IEEE, 2021.

J. Yang, G. O. Roberts, and J. S. Rosenthal. Optimal scaling of random-walk Metropolis algorithms on general target distributions. *Stochastic Processes and their Applications*, 130(10):6094–6132, 2020.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Not Applicable; all algorithms are $O(n)$ time and $O(n)$ space complexity.]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes; `PosteriorDB` properly attributed]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. Not Applicable]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Appendices

Appendix A contains the proofs for all theoretical results stated in the main text. Appendix B contains full details of our implementation, so that the empirical results we report can be reproduced. Full empirical results are contained in Appendix C.

A Proofs

Appendix A.1 contains the proof of Theorem 2. Appendix A.2 contains the prof of Theorem 1. Appendix A.3 contains the proof of Theorem 2. Auxiliary lemmas used for these proofs are contained in Appendix A.4.

A.1 Proof of Lemma 1

Proof of Lemma 1. Since ϕ-MH is a Metropolis–Hastings chain it is automatically p-invariant. The proposal distribution of ϕ-MH has a density $q_{\phi(x)}(\cdot|x)$. Under our assumptions, $(x, y) \mapsto q_{\phi(x)}(y|x)$ is positive and continuous over $x, y \in \mathcal{X}$. It follows that ϕ-MH is both (a) aperiodic, and (b) p-irreducible; in addition, from Corollary 2 of Tierney [1994], ϕ-MH is Harris recurrent. The conclusion then follows from Theorem 1 of Tierney [1994]. □

A.2 Proof of Theorem 1

The minorisation and drift conditions in Definition 1 must be established. These are the content, respectively, of Lemma 2 and Lemma 3. In the sequel we let $\lambda_{\text{Leb}}(C)$ denote the Lebesgue measure of a set $C \subset \mathbb{R}^d$, and let $R_\theta(x) = \mathbb{R}^d \setminus A_\theta(x)$ denote the region where proposals may be rejected. The following argument builds on relatively standard arguments used to establish ergodicity of Metropolis–Hastings, such as Lemma 1.2 of Mengersen and Tweedie [1996], but includes an additional dependence on the parameter $\theta \in \Theta$:

Lemma 2 (Simultaneous minorisation condition for Metropolis–Hastings)**.** *Let $\mathcal{X} = \mathbb{R}^d$. Let Θ be a topological space and let $\Theta_0 \subset \Theta$ be compact. Consider a family of p-invariant Metropolis–Hastings transition kernels P_θ, and corresponding proposals $\{q_\theta(\cdot|x) : x \in \mathbb{R}^d\}$, indexed by $\theta \in \Theta$. Let $x \mapsto p(x)$ be positive and continuous, let $(\theta, x, y) \mapsto q_\theta(y|x)$ be positive and continuous over $\theta \in \Theta$, $x, y \in \mathbb{R}^d$, and let $C \subset \mathbb{R}^d$ be compact with $\lambda_{\text{Leb}}(C) > 0$. Then there exists $\delta > 0$ and a probability measure ν on C, such that $P_\theta(x, \cdot) \geq \delta\nu(\cdot)$ holds simultaneously for all $\theta \in \Theta_0$ and all $x \in C$.*

Proof. From positivity, continuity and compactness $p_{\text{sup}} := \sup_{x \in C} p(x) \in (0, \infty)$ and $q_{\text{inf}} := \inf_{x,y \in C, \theta \in \Theta_0} q_\theta(y|x) \in (0, \infty)$. Let $p(C) := \int_C p(x)\mathrm{d}x$, which is positive since $p(\cdot)$ is bounded away from 0 on C and $\lambda_{\text{Leb}}(C) > 0$. For the result we will take $\delta = (q_{\text{inf}}/p_{\text{sup}})p(C)$ and $\nu(\cdot) = p(\cdot)/p(C)$. Then for all $\theta \in \Theta_0$, $x \in C$, and $S \subseteq C$,

$$
\begin{aligned}
P_\theta(x, S) = & \underbrace{\int_S q_\theta(y|x)\alpha_\theta(x, y)\mathrm{d}y}_{\text{probability of moving from } x \text{ to a different state in } S} + 1_S(x) \underbrace{\int_{\mathcal{X}} q_\theta(y|x)[1 - \alpha_\theta(x, y)]\mathrm{d}y}_{\text{probability of rejecting a proposal}} \\
& \geq \int_S q_\theta(y|x)\alpha_\theta(x, y)\mathrm{d}y \\
& = \int_{S \cap A_\theta(x)} q_\theta(y|x)\alpha_\theta(x, y)\mathrm{d}y + \int_{S \cap R_\theta(x)} q_\theta(y|x)\alpha_\theta(x, y)\mathrm{d}y \\
& = \int_{S \cap A_\theta(x)} q_\theta(y|x)\mathrm{d}y + \int_{S \cap R_\theta(x)} \frac{p(y)}{p(x)} q_\theta(x|y)\mathrm{d}y \\
& \geq \int_{S \cap A_\theta(x)} \frac{p(y)}{p_{\text{sup}}} q_{\text{inf}}\mathrm{d}y + \int_{S \cap R_\theta(x)} \frac{p(y)}{p_{\text{sup}}} q_{\text{inf}}\mathrm{d}y = \int_S \frac{p(y)}{p_{\text{sup}}} q_{\text{inf}}\mathrm{d}y = \delta\nu(S),
\end{aligned}
$$

as required. □

Note that the conclusion of Lemma 2 is slightly stronger than what we are minimally required to establish for minorisation in Definition 1, since it provides a probability measure ν that applies simultaneously for all θ in a compact set Θ.

Our focus now turns to the drift condition. For $x \in \mathbb{R}^d$ and $R \geq 0$, let $B_R(x) := \{y \in \mathbb{R}^d : \|y - x\| \leq R\}$ denote the x-centred, radius R ball. The following is based on an argument used in the proof of Theorem 4.1 of Jarner and Hansen [2000], but generalised beyond the case of a random walk proposal. For the proof we will use a technical lemma on the tail properties of sub-exponential distributions, provided as Lemma 6 in Appendix A.4.

Lemma 3 (Simultaneous drift condition for Metropolis–Hastings)**.** *In the setting of Theorem 1, there exists $V : \mathbb{R}^d \to [1, \infty)$, $\lambda < 1$, $b < \infty$, and $R > 0$ such that, letting $S = B_R(0)$, we have $\sup_{x \in S} V(x) < \infty$ and $(P_\theta V)(x) \leq \lambda V(x) + b 1_S(x)$ for all $x \in \mathbb{R}^d$.*

Proof. First note that our quasi-symmetry assumption implies that

$$\left\{ y : \frac{p(y)}{p(x)} \geq \rho \right\} \subseteq A_\theta(x) \subseteq \left\{ y : \frac{p(y)}{p(x)} \geq \frac{1}{\rho} \right\} \tag{6}$$

$$\left\{ y : \frac{p(y)}{p(x)} < \frac{1}{\rho} \right\} \subseteq R_\theta(x) \subseteq \left\{ y : \frac{p(y)}{p(x)} < \rho \right\} \tag{7}$$

and also that $x \in A_\theta(x)$ will always hold. For the proof we will take $V(x) = cp(x)^{-1/2}$ for c such that $V \geq 1$, which we can do since p is continuous, positive, and vanishing in the tail. It follows from continuity and compactness that $\sup_{x \in S} V(x) < \infty$ is satisfied. It then suffices to show that

$$\limsup_{\|x\| \to \infty} \sup_{\theta \in \Theta} \frac{(P_\theta V)(x)}{V(x)} < 1 \tag{8}$$

$$\sup_{x \in \mathbb{R}^d} \sup_{\theta \in \Theta} \frac{(P_\theta V)(x)}{V(x)} < \infty. \tag{9}$$

Indeed, if (8) holds then there exists $R > 0$, $\lambda < 1$ such that $P_\theta V(x) \leq \lambda V(x)$ for all $x \notin S$ and all $\theta \in \Theta$. Further, if (9) holds then, since V is continuous, we can set

$$b = \sup_{x \in S} \sup_{\theta \in \Theta} (P_\theta V)(x) \leq \sup_{x \in S} V(x) \sup_{x' \in \mathbb{R}^d} \sup_{\theta \in \Theta} \frac{(P_\theta V)(x')}{V(x')} < \infty$$

since V is bounded on the compact set S. Then the drift condition will have been established.

Establishing (9)*:* Decompose and then bound the integral as

$$\begin{aligned}
(P_\theta V)(x) &= \underbrace{\int_{A_\theta(x)} q_\theta(y|x) V(y) \mathrm{d}y}_{\text{always accept}} + \underbrace{\int_{R_\theta(x)} q_\theta(y|x) \left[\frac{q_\theta(x|y)p(y)}{q_\theta(y|x)p(x)} V(y) + \left(1 - \frac{q_\theta(x|y)p(y)}{q_\theta(y|x)p(x)} \right) V(x) \right] \mathrm{d}y}_{\text{possibly reject}} \\
&\leq \int_{A_\theta(x)} q_\theta(y|x) V(y) \mathrm{d}y + \int_{R_\theta(x)} \frac{q_\theta(x|y)p(y)}{p(x)} V(y) + q_\theta(y|x) V(x) \mathrm{d}y
\end{aligned}$$

so

$$\frac{(P_\theta V)(x)}{V(x)} \leq \int_{A_\theta(x)} q_\theta(y|x) \frac{V(y)}{V(x)} \mathrm{d}y + \int_{R_\theta(x)} \frac{q_\theta(x|y)p(y)}{p(x)} \frac{V(y)}{V(x)} + q_\theta(y|x) \mathrm{d}y.$$

(The first of these two inequalities is in fact strict, but we do not need a strict inequality for our argument.) Then, plugging in our choice of V, we have

$$\frac{(P_\theta V)(x)}{V(x)} \leq \int_{A_\theta(x)} q_\theta(y|x) \frac{p(x)^{1/2}}{p(y)^{1/2}} \mathrm{d}y + \int_{R_\theta(x)} q_\theta(x|y) \frac{p(y)^{1/2}}{p(x)^{1/2}} + q_\theta(y|x) \mathrm{d}y. \tag{10}$$

From (6) and (7), together with $q_\theta(x|y) \leq \rho q_\theta(y|x)$, we see that

$$\begin{aligned}
\frac{(P_\theta V)(x)}{V(x)} &\leq \int_{A_\theta(x)} q_\theta(y|x) \rho^{1/2} \mathrm{d}y + \int_{R_\theta(x)} \rho q_\theta(y|x) \rho^{1/2} + q_\theta(y|x) \mathrm{d}y \\
&= \rho^{1/2} Q_\theta(A_\theta(x)|x) + (\rho^{3/2} + 1) Q_\theta(R_\theta(x)|x) = \rho^{1/2} + \rho^{3/2} + 1 < \infty
\end{aligned}$$

where the final bound is x- and θ-independent, so that (9) is established.

Establishing (8)*:* Fix $\epsilon > 0$. By the regular acceptance boundary assumption, there exists $\delta > 0$ small enough and $R_1 > 0$ large enough that $Q_\theta(\partial A_\theta^\delta(x)|x) < \epsilon$ for all θ and all $\|x\| \geq R_1$. From Lemma 6, there exists $R_2 > 0$ large enough that, for all $\|x\| \geq R_2$, $y \in A_\theta(x) \cap (\partial A_\theta^\delta(x)^\mathsf{c})$ implies that $p(x)/p(y) \leq \epsilon$ and $y \in R_\theta(x) \cap \partial A_\theta^\delta(x)^c$ implies that $p(y)/p(x) \leq \epsilon$. So, for $\|x\| \geq \max\{R_1, R_2\}$, and using (6),

$$\begin{aligned}\int_{A_\theta(x)} q_\theta(y|x)\frac{p(x)^{1/2}}{p(y)^{1/2}}\mathrm{d}y &= \int_{A_\theta(x)\cap\partial A_\theta^\delta(x)} q_\theta(y|x)\frac{p(x)^{1/2}}{p(y)^{1/2}}\mathrm{d}y + \int_{A_\theta(x)\cap(\partial A_\theta^\delta(x)^\mathsf{c})} q_\theta(y|x)\frac{p(x)^{1/2}}{p(y)^{1/2}}\mathrm{d}y \\ &\leq \rho^{1/2} Q_\theta(\partial A_\theta^\delta(x)|x) + \epsilon^{1/2} Q_\theta(\mathbb{R}^d|x) \leq \rho^{1/2}\epsilon + \epsilon^{1/2}.\end{aligned}$$

Since $\epsilon > 0$ was arbitrary, we have shown that

$$\limsup_{\|x\|\to\infty} \sup_{\theta\in\Theta} \int_{A_\theta(x)} q_\theta(y|x)\frac{p(x)^{1/2}}{p(y)^{1/2}}\mathrm{d}y = 0. \tag{11}$$

A similar argument shows that, for $\|x\| \geq \max\{R_1, R_2\}$, and using (7),

$$\begin{aligned}\int_{R_\theta(x)} q_\theta(x|y)\frac{p(y)^{1/2}}{p(x)^{1/2}}\mathrm{d}y &\leq \int_{R_\theta(x)\cap\partial A_\theta^\delta(x)} q_\theta(x|y)\frac{p(y)^{1/2}}{p(x)^{1/2}}\mathrm{d}y + \int_{R_\theta(x)\cap(\partial A_\theta^\delta(x)^\mathsf{c})} q_\theta(x|y)\frac{p(y)^{1/2}}{p(x)^{1/2}}\mathrm{d}y \\ &\leq \rho Q_\theta(\partial A_\theta^\delta(x)|x)\rho^{1/2} + \rho Q_\theta(\mathbb{R}^d|x)\epsilon^{1/2} \leq \rho^{3/2}\epsilon + \rho\epsilon^{1/2}\end{aligned}$$

and since $\epsilon > 0$ was arbitrary, we have shown that

$$\limsup_{\|x\|\to\infty} \sup_{\theta\in\Theta} \int_{R_\theta(x)} q_\theta(x|y)\frac{p(y)^{1/2}}{p(x)^{1/2}}\mathrm{d}y = 0. \tag{12}$$

Thus, substituting (11) and (12) into (10) yields

$$\limsup_{\|x\|\to\infty} \sup_{\theta\in\Theta} \frac{P_\theta V(x)}{V(x)} = \limsup_{\|x\|\to\infty} \sup_{\theta\in\Theta} \int_{R_\theta(x)} q_\theta(y|x)\mathrm{d}y = 1 - \liminf_{\|x\|\to\infty} \inf_{\theta\in\Theta} Q_\theta(A_\theta(x)|x) < 1,$$

where we have used the minimum performance assumption to obtain the inequality. The claim (8) has been established. □

Proof of Theorem 1. From Definition 1 we must show that minorisation and simultaneous drift conditions are satisfied. The simultaneous drift condition is established in Lemma 3 with $S = B_R(0)$ for some $R > 0$. Since S is compact with $\lambda_{\mathrm{Leb}}(S) > 0$, the minorisation condition follows from Lemma 2. □

A.3 Proof of Theorem 2

The proof of Theorem 2 exploits the framework for analysis of adaptive MCMC advocated in Roberts and Rosenthal, 2007. Auxiliary lemmas used in this proof are deferred to Appendix A.4. For a (possibly signed) measure ν on $\mathcal{X}$, denote the total variation norm $\|\nu\|_{\mathrm{TV}} = \sup_{S\subset\mathcal{X}} |\nu(S)|$. Then we aim to make use of the following well-known result:

Theorem 3 (Theorem 3 of Roberts and Rosenthal, 2007)**.** *Let Θ_0 be a set and consider an adaptive MCMC algorithm $\theta_n \equiv \theta_n(x_0, \ldots, x_n) \in \Theta_0$ with Markov transition kernels $\{P_\theta\}_{\theta\in\Theta_0}$ initialised at a fixed $x_0 \in \mathcal{X}$ and $\theta_0 \in \Theta_0$. Suppose $\{P_\theta\}_{\theta\in\Theta_0}$ is SSAGE and that the* diminishing adaptation *condition, meaning*

$$\sup_{x\in\mathcal{X}} \|P_{\theta_{n+1}}(x,\cdot) - P_{\theta_n}(x,\cdot)\|_{\mathrm{TV}} \to 0$$

in probability as $n \to \infty$, is satisfied. Then $\|\mathrm{Law}(x_n) - p\|_{\mathrm{TV}} \to 0$.

The easier of these two conditions to establish is diminishing adaptation, which is satisfied due to our control of the learning rate and clipping of the gradient, as demonstrated in Lemma 4. A useful fact that we will use in the proof is that the norms $\|\cdot\|$ and $\|\cdot\|_{1,\Sigma}$ are equivalent with

$$\lambda_{\max}^{-1/2}(\Sigma)\|x\| \leq \|x\|_{1,\Sigma} \leq \sqrt{d}\lambda_{\min}^{-1/2}(\Sigma)\|x\| \tag{13}$$

for all $x \in \mathbb{R}^d$, where $\lambda_{\min}(\Sigma)$ and $\lambda_{\max}(\Sigma)$ denote, respectively, the minimum and maximum eigenvalues of the matrix $\Sigma \in \mathrm{S}_d^+$.

Lemma 4 (Diminishing adaptation for RLMH)**.** *The gradient clipping with threshold $\tau > 0$ and summable learning rate $(\alpha_n)_{n\geq 0} \subset [0,\infty)$, appearing in Algorithm 1, ensure that*

$$(\theta_n)_{n\geq 0} \subset \Theta_0 := B_T(\theta_0), \qquad T := \tau \sum_{n=0}^{\infty} \alpha_n < \infty. \tag{14}$$

From local boundedness it follows that

$$B := \sup_{\theta\in\Theta_0} \mathrm{Lip}(\phi_\theta) < \infty, \tag{15}$$

since Θ_0 is compact. In particular, in the setting of Theorem 2, diminishing adaptation is satisfied.

Proof. The set-up in Algorithm 1 implies that

$$\|\theta_n - \theta_0\| \leq \sum_{i=0}^{n-1} \|\theta_{i+1} - \theta_i\| \leq \sum_{i=0}^{n-1} \alpha_i \tau \leq \tau \sum_{i=0}^{\infty} \alpha_i < \infty$$

where the final bound is n-independent and finite, since the summability of the learning rate $(\alpha_n)_{n\geq 0} \subset [0,\infty)$ was assumed. From this, (14) is immediately established.

The main idea of this proof is to exploit the triangle inequality and the definition of the total variation norm, as follows:

$$\begin{aligned}
\|P_\theta(x,\cdot) - P_\vartheta(x,\cdot)\|_{\mathrm{TV}} &\leq \Big\| \int q_\theta(y|x)[\delta_y(\cdot)\alpha_\theta(x,y) + \delta_x(\cdot)(1-\alpha_\theta(x,y))]\,\mathrm{d}y \\
&\qquad - \int q_\vartheta(y|x)[\delta_y(\cdot)\alpha_\vartheta(x,y) + \delta_x(\cdot)(1-\alpha_\vartheta(x,y))]\,\mathrm{d}y \Big\|_{\mathrm{TV}} \\
&\leq \Big\| \int \delta_y(\cdot)[q_\theta(y|x)\alpha_\theta(x,y) - q_\vartheta(y|x)\alpha_\vartheta(x,y)]\,\mathrm{d}y \Big\|_{\mathrm{TV}} \\
&\qquad + \Big\| \delta_x(\cdot) \int [q_\theta(y|x)\alpha_\theta(x,y) - q_\vartheta(y|x)\alpha_\vartheta(x,y)]\,\mathrm{d}y \Big\|_{\mathrm{TV}} \\
&\leq \int |q_\theta(y|x)\alpha_\theta(x,y) - q_\vartheta(y|x)\alpha_\vartheta(x,y)|\,\mathrm{d}y,
\end{aligned}$$

where δ_x denotes the probability distribution that puts all mass at $x \in \mathbb{R}^d$. From the triangle inequality again,

$$\begin{aligned}
\|P_\theta(x,\cdot) - P_\vartheta(x,\cdot)\|_{\mathrm{TV}} &\leq \int |q_\theta(y|x) - q_\vartheta b(y|x)|\,\alpha_\theta(x,y) + |\alpha_\theta(x,y) - \alpha_\vartheta(x,y)| q_\vartheta(y|x)\,\mathrm{d}y \\
&\leq \int |q_\theta(y|x) - q_\vartheta(y|x)|\,\mathrm{d}y + \int |\alpha_\theta(x,y) - \alpha_\vartheta(x,y)|\,q_\vartheta(y|x)\,\mathrm{d}y,
\end{aligned} \tag{16}$$

and we seek to bound the two integrals appearing in (16).

Fix $x \in \mathbb{R}^d$. Then the map $\vartheta \mapsto \phi_\vartheta(x)$ is locally Lipschitz, and since Θ_0 is connected and compact it follows that $\vartheta \mapsto \phi_\vartheta(x)$ is Lipschitz on Θ_0, with Lipschitz constant

$$L_x := \sup_{\theta\in\Theta_0} \mathrm{LocLip}_\theta(\vartheta \mapsto \phi_\vartheta(x)).$$

Further, this Lipschitz constant can be uniformly bounded over $x \in \mathbb{R}^d$:

$$L := \sup_{x\in\mathbb{R}^d} L_x = \sup_{x\in\mathbb{R}^d} \sup_{\theta\in\Theta_0} \mathrm{LocLip}_\theta(\vartheta \mapsto \phi_\vartheta(x)) = \sup_{\theta\in\Theta_0} \sup_{x\in\mathbb{R}^d} \mathrm{LocLip}_\theta(\vartheta \mapsto \phi_\vartheta(x)) < \infty,$$

where finiteness follows since we assumed local boundedness of $\theta \mapsto \sup_{x\in\mathbb{R}^d} \mathrm{LocLip}_\theta(\vartheta \mapsto \phi_\vartheta(x))$ and Θ_0 is compact.

Fix $\theta, \vartheta \in \Theta_0$. Now,

$$\begin{aligned}|q_\theta(y|x) - q_\vartheta(y|x)| &= q_\vartheta(y|x)\left|1 - \frac{q_\theta(y|x)}{q_\vartheta(y|x)}\right| \\ &= q_\vartheta(y|x)\left|1 - \exp\left(-\|y-\phi_\theta(x)\|_{1,\Sigma} + \|y-\phi_\vartheta(x)\|_{1,\Sigma}\right)\right|.\end{aligned}$$

From the reverse triangle inequality, the aforementioned Lipschitz property, and (13),

$$\left|-\|y-\phi_\theta(x)\|_{1,\Sigma} + \|y-\phi_\vartheta(x)\|_{1,\Sigma}\right| \leq \|\phi_\theta(x) - \phi_\vartheta(x)\|_{1,\Sigma} \leq \sqrt{d}\lambda_{\min}^{1/2}(\Sigma)\, L\, \|\theta - \vartheta\|. \tag{17}$$

Further, our assumptions imply that the right hand side of (17) is bounded since $\theta, \vartheta \in \Theta_0$ and Θ_0 is compact. Since the exponential function is Lipschitz on any compact set, there exists an (x, θ, ϑ)-independent constant $C > 0$ such that

$$\left|1 - \exp\left(-\|y-\phi_\theta(x)\|_{1,\Sigma} + \|y-\phi_\vartheta(x)\|_{1,\Sigma}\right)\right| \leq C\left|-\|y-\phi_\theta(x)\|_{1,\Sigma} + \|y-\phi_\vartheta(x)\|_{1,\Sigma}\right|$$

and hence

$$\int |q_\theta(y|x) - q_\vartheta(y|x)|\, \mathrm{d}y \leq C\, \sqrt{d}\lambda_{\min}^{1/2}(\Sigma)\, L\, \|\theta - \vartheta\| \tag{18}$$

holds for all $\theta, \vartheta \in \Theta_0$ and all $x \in \mathbb{R}^d$.

Next, from Lemma 7 the proposal $\{q_\theta(\cdot|x) : x \in \mathbb{R}^d\}$ is quasi-symmetric on Θ_0 with constant ρ defined as in (22). From (6), if $p(y)/p(x) \geq \rho$ then $y \in A_\theta(x) \cap A_\vartheta(x)$ and $\alpha_\theta(x,y) = \alpha_\vartheta(x,y) = 1$, so that $|\alpha_\theta(x,y) - \alpha_\vartheta(x,y)| = 0$. Otherwise, we have $p(y)/p(x) < \rho$ and from quasi-symmetry again,

$$\begin{aligned}|\alpha_\theta(x,y) - \alpha_\vartheta(x,y)| &\leq \frac{p(y)}{p(x)}\left|\frac{q_\theta(x|y)}{q_\theta(y|x)} - \frac{q_\vartheta(x|y)}{q_\vartheta(y|x)}\right| \\ &= \frac{p(y)}{p(x)}\frac{q_\theta(x|y)}{q_\theta(y|x)}\left|1 - \frac{q_\theta(y|x)}{q_\theta(x|y)}\frac{q_\vartheta(x|y)}{q_\vartheta(y|x)}\right| \\ &\leq \rho^2\left|1 - \exp\begin{pmatrix} -\|y-\phi_\theta(x)\|_{1,\Sigma} + \|x-\phi_\theta(y)\|_{1,\Sigma} \\ -\|x-\phi_\vartheta(y)\|_{1,\Sigma} + \|y-\phi_\vartheta(x)\|_{1,\Sigma}\end{pmatrix}\right|.\end{aligned}$$

From (17) and an analogous argument to before involving the fact that the exponential function is Lipschitz on a compact set, we obtain that

$$\int |\alpha_\theta(x,y) - \alpha_\vartheta(x,y)|\, q_\vartheta(y|x)\mathrm{d}y \leq 2C\rho^2\, \sqrt{d}\lambda_{\min}^{1/2}(\Sigma)\, L\, \|\theta - \vartheta\| \tag{19}$$

for all $\theta, \vartheta \in \Theta_0$ and $x \in \mathbb{R}^d$.

Substituting (18) and (19) into (16),

$$\|P_\theta(x,\cdot) - P_\vartheta(x,\cdot)\|_{\mathrm{TV}} \leq C(1+2\rho^2)\, \sqrt{d}\lambda_{\min}^{1/2}(\Sigma)\, L\, \|\theta - \vartheta\|$$

for all $\theta, \vartheta \in \Theta_0$ and $x \in \mathbb{R}^d$. It follows (a.s.) that

$$\sup_{x \in \mathbb{R}^d} \|P_{\theta_n}(x,\cdot) - P_{\theta_{n+1}}(x,\cdot)\|_{\mathrm{TV}} \leq C(1+2\rho^2)\, \sqrt{d}\lambda_{\min}^{1/2}(\Sigma)\, L\, \|\theta_n - \theta_{n+1}\| \to 0$$

since $(\theta_n)_{n \geq 0}$ is (a.s.) convergent. Since almost sure convergence implies convergence in probability, diminishing adaptation is established. □

The main technical effort required to prove Theorem 2 occurs in establishing conditions under which ϕ-MH is SSAGE. This result is the content of Lemma 5. For the proof we will use technical lemmas on the tail properties of sub-exponential distributions, provided as Lemmas 6 to 8, and a technical lemma on the interior cone condition, provided as Lemma 10, all of which can be found in Appendix A.4.

For a subset $C \subset \mathbb{R}^d$, let $C \cong \mathbb{S}^{d-1}$ denote that $C = \{r(\xi)\xi : \xi \in \mathbb{S}^{d-1}\}$ for some continuous function $r : \mathbb{S}^{d-1} \to (0, \infty)$, meaning that C is a hyper-surface that can be parametrised using the $(d-1)$-dimensional sphere $\mathbb{S}^{d-1}$. For $\epsilon > 0$, let $C_\epsilon := \{x \in \mathbb{R}^d : p(x) = \epsilon\}$ be the ϵ-level set of $p(\cdot)$ and, for $\delta \geq 0$, let $C_\epsilon^\delta := \{x + sn(x) : x \in C_\epsilon, |s| \leq \delta\}$.

Lemma 5 (Flexible mean ϕ-MH is SSAGE)**.** *Let $\Theta_0 \subset \Theta = \mathbb{R}^p$ be the compact set from* (4)*. In the setting of Theorem 2, $\{P_\theta\}_{\theta\in\Theta_0}$ is SSAGE.*

Proof. The conditions of Theorem 1 need to be checked. Let $M := \sup_{\theta\in\Theta_0} \sup_{x\in\mathbb{R}^d} \|x - \phi_\theta(x)\|_{1,\Sigma}$, which exists by the assumed local boundedness of $\theta \mapsto \sup_{x\in\mathbb{R}^d} \|x - \phi_\theta(x)\|$, compactness of Θ_0, and the norm-equivalence in (13). The form of our flexible mean Laplace proposal (5) ensures that $q_{\sup} := \sup_{\theta\in\Theta_0, x, y\in\mathbb{R}^d} q_\theta(y|x) < \infty$.

Quasi-symmetry: From Lemma 7, the proposal $\{q_\theta(\cdot|x) : x \in \mathbb{R}^d\}$ is quasi-symmetric on Θ_0, with the constant ρ defined as in (22).

Regular acceptance boundary: Given $\epsilon > 0$, we can pick $R_1 > 0$ such that for all $x \in \mathbb{R}^d$ and all $\theta \in \Theta_0$,

$$Q_\theta(B_{R_1}(\phi_\theta(x))^{\mathsf{c}}|x) < \epsilon$$

due to the form of our Laplace proposal in (5). From the definition of M, we can pick $R_2 \geq R_1$ such that $B_{R_1}(\phi_\theta(x)) \subseteq B_{R_2}(x)$ holds simultaneously for all $x \in \mathbb{R}^d$ and all $\theta \in \Theta_0$. Thus in particular

$$Q_\theta(B_{R_2}(x)^{\mathsf{c}}|x) < \epsilon \tag{20}$$

holds simultaneously for all $x \in \mathbb{R}^d$ and all $\theta \in \Theta_0$.

From Lemmas 6 and 8, the assumption that $p \in \mathcal{P}(\mathbb{R}^d)$ implies there is an $R_3 > R_2$ such that for all $\|x\| \geq R_3$ we have $C_{\rho p(x)}, \partial A_\theta(x), C_{\rho^{-1}p(x)} \cong \mathbb{S}^{d-1}$. Further, from Lemma 6 with $r = \frac{4}{\epsilon}\log\rho$, and the fact $\rho \geq 1$, we may assume that R_3 is large enough that

$$\frac{p(x + sn(x))}{p(x)} \geq \frac{1}{\rho^2} \qquad \Longrightarrow \qquad s \leq \frac{\epsilon}{2}$$

so that the radial distance between $C_{\rho p(x)}$ and $C_{\rho^{-1}p(x)}$ is uniformly less than $\epsilon/2$ for all $\|x\| \geq R_3$. From (6), both $C_{p(x)}$ and $\partial A_\theta(x)$ is contained in the region bounded by $C_{\rho p(x)}$ and $C_{\rho^{-1}p(x)}$. It follows that, if $\delta \in [0, \epsilon/2)$, then $\partial A^\delta_\theta(x) \subset C^\epsilon_{p(x)}$ uniformly in $\|x\| \geq R_3$ and $\theta \in \Theta_0$.

From Lemma 9, for all $\|x\| \geq R_3$ ($> R_2$) we have the bound

$$\begin{aligned}\lambda_{\mathrm{Leb}}\left(C^\epsilon_{p(x)} \cap B_{R_2}(x)\right) &\leq \epsilon \left(\frac{\|x\| + R_2}{\|x\| - R_2}\right)^{d-1} \frac{\lambda_{\mathrm{Leb}}(B_{3R_2}(x))}{R_2} \\ &\leq \epsilon \underbrace{\left(\frac{R_3 + R_2}{R_3 - R_2}\right)^{d-1} \frac{\lambda_{\mathrm{Leb}}(B_{3R_2}(x))}{R_2}}_{=:D}\end{aligned} \tag{21}$$

where the last inequality is obtained by maximising the x-dependent term over $\|x\| \geq R_3$.

Putting these results together, we have the bound

$$Q_\theta(\partial A^\delta_\theta(x)|x) = Q_\theta(\partial A^\delta_\theta(x) \cap (B_{R_2}(x)^{\mathsf{c}})|x) + Q_\theta(\partial A^\delta_\theta(x) \cap B_{R_2}(x)|x) \leq \epsilon + q_{\sup}\, D\, \epsilon$$

for all $\|x\| \geq R_3$ and all $\theta \in \Theta_0$, where for the first term we have used (20) and for the second term we have used (21). Since $\epsilon > 0$ was arbitrary and our bound is θ-independent, the regular acceptance boundary condition is established.

Minimum performance level: From Lemmas 6 and 8, the assumption that $p \in \mathcal{P}(\mathbb{R}^d)$ implies there is an $R > 0$ such that for all $\|x\| \geq R$ we have $C_{\rho p(x)}, \partial A_\theta(x), C_{\rho^{-1}p(x)} \cong \mathbb{S}^{d-1}$. From (6), $\partial A_\theta(x)$ is contained in the region bounded by $C_{\rho p(x)}$ and $C_{\rho^{-1}p(x)}$. For $x \neq 0$, let y_x denote the intersection of the line $\{sx : s \geq 0\}$ with the set $C_{\rho p(x)}$. From Lemma 6 and the norm-equivalence in (13), we may consider R sufficiently large that $\|y_x - x\|_{1,\Sigma} \leq 1$ for all $\|x\| \geq R$, and in particular this leads to the bound $\|y_x - \phi_\theta(x)\|_{1,\Sigma} \leq M + 1$.

From Lemma 10, since $p \in \mathcal{P}_0(\mathbb{R}^d)$ we may assume R is sufficiently large that for all $\|x\| \geq R$ there exists a radius-1 cone $K_1(y_x)$ with x-independent Lebesgue measure $\zeta_1 > 0$, whose apex is y_x, contained in the interior of the compact region bounded by $C_{\rho p(x)}$ ($\subset A_\theta(x)$). In particular, $K_1(y_x)$ is contained in a $\|\cdot\|_{1,\Sigma}$-ball of radius $M + 2$ centred at $\phi_\theta(x)$.

Thus for all $\|x\| \geq R$ and $\theta \in \Theta_0$,

$$\begin{aligned} Q_\theta(A_\theta(x)|x) \geq Q_\theta(A_\theta(x) \cap K_1(y_x)|x) &= Q_\theta(K_1(y_x)|x) \\ &\geq \lambda_{\text{Leb}}(K_1(y_x)) \times \inf_{\|y-\phi_\theta(x)\|_{1,\Sigma} \leq M+2} q_\theta(y|x) \\ &\geq \zeta_1 \times \frac{1}{Z} \exp(-(M+2)) > 0 \end{aligned}$$

where $Z > 0$ is the (x, θ)-independent normalisation constant of the proposal (5). Thus the final condition of Theorem 1 is satisfied. □

At last we have established Theorem 2:

Proof of Theorem 2. From Lemma 4, under our stated assumptions diminishing adaptation is satisfied and the sequence $(\theta_n)_{n\geq 0}$ is contained in a compact set Θ_0. From Lemma 5, under our stated assumptions, the ϕ-MH Markov transition kernels $\{P_\theta\}_{\theta\in\Theta_0}$ are SSAGE. The ergodicity of RLMH then follows from Theorem 3. □

A.4 Auxiliary Lemmas

The following auxiliary lemmas were used in the proofs of Theorems 1 and 2. The first, Lemma 6, is a well-known result that sub-exponential distributions decay uniformly quickly in the tail:

Lemma 6 (Tails of sub-exponential distributions)**.** *Let $p \in \mathcal{P}(\mathbb{R}^d)$. Then for all $r > 0$ there exists $R > 0$ such that*

$$\frac{p(x + sn(x))}{p(x)} \leq \exp(-sr), \qquad \forall \|x\| \geq R, s \geq 0$$

and $C_{p(x)} = \{y : p(y) = p(x)\} \cong \mathbb{S}^{d-1}$.

Proof. See Sec. 4. of Jarner and Hansen, 2000. □

The next technical lemma, Lemma 7, is a simple but novel cocntribution of this work, establishing sufficient conditions for quasi-symmetry to be satisfied:

Lemma 7 (Quasi-symmetry for ϕ-MH)**.** *Let Θ_0 be compact. In the setting of Theorem 2, the proposal $\{q_\theta(\cdot|x) : x \in \mathbb{R}^d\}$ is quasi-symmetric on Θ_0, meaning that*

$$\rho := \sup_{\theta\in\Theta_0} \sup_{x,y\in\mathbb{R}^d} \frac{q_\theta(y|x)}{q_\theta(x|y)} < \infty. \tag{22}$$

Proof. For this proposal,

$$\frac{q_\theta(y|x)}{q_\theta(x|y)} = \exp\left(-\|y - \phi_\theta(x)\|_{1,\Sigma} + \|x - \phi_\theta(y)\|_{1,\Sigma}\right).$$

Let $z = x - y$ and $\eta_x^\theta = x - \phi_\theta(x)$. Let $M := \sup_{\theta\in\Theta_0} \sup_{x\in\mathbb{R}^d} \|x - \phi_\theta(x)\|_{1,\Sigma}$, which exists by the assumed local boundedness of $\theta \mapsto \sup_{x\in\mathbb{R}^d} \|x - \phi_\theta(x)\|$, compactness of Θ_0, and the norm-equivalence in (13). Then we have a bound

$$\begin{aligned} |-\|y - \phi_\theta(x)\|_{1,\Sigma} + \|x - \phi_\theta(y)\|_{1,\Sigma}| &= \left|\|\eta_y^\theta - z\|_{1,\Sigma} - \|\eta_x^\theta - z\|_{1,\Sigma}\right| \\ &\leq \|\eta_x^\theta - \eta_y^\theta\|_{1,\Sigma} \leq \|\eta_x^\theta\|_{1,\Sigma} + \|\eta_y^\theta\|_{1,\Sigma} \leq 2M. \end{aligned}$$

where we have used the reverse triangle inequality. The result is then established with $\rho \leq \exp(2M)$ being an explicit bound on the quasi-symmetry constant. □

The next technical lemma, Lemma 8, concerns the geometry of the acceptance set $A_\theta(x)$. In the case of a random walk proposal, $\rho = 1$ and the first part of Lemma 8 reduces to the existing Lemma 6. A novel contribution of this paper is to study the geometry of the acceptance set in the case of a more general Metropolis–Hastings proposal.

Lemma 8 (Geometry of $\partial A_\theta(x)$)**.** *Let $p \in \mathcal{P}(\mathbb{R}^d)$ and $\Theta_0 \subset \Theta = \mathbb{R}^p$. Suppose that $(\theta, x) \mapsto \phi_\theta(x)$ is continuous and $B = \sup_{\theta\in\Theta_0} \mathrm{Lip}(\phi_\theta) < \infty$. Assume quasi-symmetry with parameter ρ. Then for all x large enough, and all $\theta \in \Theta_0$, $\partial A_\theta(x) \cong \mathbb{S}^{d-1}$.*

Proof. From Lemma 6, the assumption that $p \in \mathcal{P}(\mathbb{R}^d)$ implies there exists $R_1 > 0$ such that, for all $\|x\| \geq R_1$, we have $C_{\rho p(x)} \cong \mathbb{S}^{d-1}$. From (4), there exists $R_2 \geq R_1$ sufficiently large that

$$\sup_{\|x\|\geq R_2} n(x) \cdot \nabla \log p(x) < -\sqrt{d}\lambda_{\min}^{-1/2}(\Sigma)(1 + B). \tag{23}$$

From Lemma 6 with $r = \log \rho$, and recalling the fact that $\rho \geq 1$, there exists $R_3 \geq R_2 + 1$ large enough that

$$\frac{p(x + sn(x))}{p(x)} \geq \frac{1}{\rho} \qquad \Longrightarrow \qquad s \leq 1$$

so that the radial distance between $C_{p(x)}$ and $C_{\rho p(x)}$ is uniformly at most 1 for all $\|x\| \geq R_3$. Finally, since $p(\cdot)$ is positive and continuous with $p(x) \to 0$ as $\|x\| \to \infty$ (from Lemma 6), there exists $R_4 \geq R_3$ such that for each $\|x\| \geq R_4$, we have $\|z\| \geq R_3$ for all $z \in C_{\rho p(x)}$. For $x, y \in \mathbb{R}^d$ with $\|x\| \geq R_4$, $y \neq 0$, and $p(y) \leq \rho p(x)$, let $r_{x,y}$ denote the (unique) positive constant such that $r_{x,y} n(y) \in C_{\rho p(x)}$. Then, for all $\|x\| \geq R_4$,

$$\sup_{y:p(y)\leq\rho p(x)} \partial_s \log p(y + sn(y)) < -\sqrt{d}\lambda_{\min}^{-1/2}(\Sigma)(1 + B) \tag{24}$$

since for any y with $p(y) \leq \rho p(x)$, we have that $\|y\| \geq \|r_{x,y} n(y)\| - 1 \geq R_3 - 1 \geq R_2$, and thus (23) will hold. In the sequel we assume that $\|x\| \geq R_4$.

From (6) the set $\partial A_\theta(x)$ is contained in the region bounded by $C_{\rho p(x)}$ and $C_{\rho^{-1} p(x)}$. Let $\xi \in \mathbb{S}^{d-1}$ and consider the intersection of the set $\partial A_\theta(x)$ with the line segment $\gamma(\xi) = \{r\xi : r_{\min} \leq r \leq r_{\max}\}$ where $p(r_{\min}\xi) = \rho p(x)$ and $p(r_{\max}\xi) = \rho^{-1} p(x)$. Our first task is to establish that this intersection is a singleton set.

Let $f_{x,\xi}(r) := \log p(r\xi) - \log p(x)$ and $g_{x,\xi,\theta}(r) := \log q_\theta(r\xi|x) - \log q_\theta(x|r\xi)$, so that we seek solutions to $f_{x,\xi}(r) = g_{x,\xi,\theta}(r)$ with $r \in [r_{\min}, r_{\max}]$. Now $f_{x,\xi}$ is continuous with $f_{x,\xi}(r_{\min}) = \log \rho$ and $f_{x,\xi}(r_{\max}) = -\log \rho$. On the other hand, $g_{x,\xi,\theta} : [r_{\min}, r_{\max}] \to \mathbb{R}$ is continuous and takes values only in $[-\log \rho, \log \rho]$, so the intersection $\partial A_\theta(x) \cap \gamma(\xi)$ is a non-empty set, and we can pick $r_\theta(\xi)\xi \in \partial A_\theta(x) \cap \gamma(\xi)$. Since the region bounded by $C_{\rho p(x)}$ and $C_{\rho^{-1} p(x)}$ does not contain 0, it follows that $r_\theta(\xi) \in (0, \infty)$.

Our next task to argue that $r_\theta(\xi)$ is the only element of this set. Now,

$$g_{x,\xi,\theta}(r) = \log q_\theta(r\xi|x) - \log q_\theta(x|r\xi) = -\|r\xi - \phi_\theta(x)\|_{1,\Sigma} + \|x - \phi_\theta(r\xi)\|_{1,\Sigma}$$

from which it follows that $\mathrm{Lip}(g_{x,\xi,\theta}) \leq \sqrt{d}\lambda_{\min}^{-1/2}(\Sigma)(1 + B)$, where we have used the norm-equivalence in (13). So $f_{x,\xi}(r)$ and $g_{x,\xi,\theta}(r)$ are equal at $r = r_\theta(\xi)$ and cannot be equal again on $r \in [r_{\min}, r_{\max}]$ since from (24) the gradient of $f_{x,\xi}(r)$ is everywhere lower than the Lipschitz constant of $g_{x,\xi,\theta}(r)$ on $[r_{\min}, r_{\max}]$. Thus $r = r_\theta(\xi)$ is the only solution to $f_{x,\xi}(r) = g_{x,\xi,\theta}(r)$.

Lastly we note that continuity of the map $\xi \mapsto r_\theta(\xi)$ follows from continuity of the maps $\xi \mapsto f_{x,\xi}$ and $\xi \mapsto g_{x,\xi,\theta}$, which in turn follows from the continuity of $x \mapsto p(x)$ and $(\theta, x) \mapsto \phi_\theta(x)$ that we assumed. □

The next technical lemma, Lemma 9, is a basic bound on the Lebesgue measure of the intersection of any set $C \cong \mathbb{S}^{d-1}$ with a Euclidean ball. The proof closely follows an argument used within the proof of Theorem 4.1 in Jarner and Hansen [2000], but we present it here to keep the paper self-contained:

Lemma 9. *Suppose that $C \cong \mathbb{S}^{d-1}$ and let $C^\epsilon := \{x + sn(x) : |s| \leq \epsilon\}$. Then, for all $R > 0$ and all $\|x\| > R$,*

$$\lambda_{\mathrm{Leb}}\left(C^\epsilon \cap B_R(x)\right) \leq \epsilon \left(\frac{\|x\| + R}{\|x\| - R}\right)^{d-1} \frac{\lambda_{\mathrm{Leb}}(B_{3R}(x))}{R}.$$

Proof. Recall that $C \cong \mathbb{S}^{d-1}$ means that we can parametrise C as $C = \{r(\xi)\xi \ : \ \xi \in \mathbb{S}^{d-1}\}$ for some function $r : \mathbb{S}^{d-1} \to (0, \infty)$. Let $T(x) := \{\xi \in \mathbb{S}^{d-1} : r\xi \in B_R(x) \text{ for some } r \geq 0\}$ and $S(x) := \{r\xi : \xi \in T(x), \|x\| - R \leq$

$r \leq \|x\| + R\}$. Then $B_R(x) \subset S(x) \subset B_{3R}(x)$. Let ω_d denote the surface measure on $\mathbb{S}^{d-1}$. The first inclusion leads to the bound

$$\begin{aligned}\lambda_{\text{Leb}}(C^\epsilon \cap B_R(x)) = \int 1_{C^\epsilon \cap B_R(x)}(y)\mathrm{d}y &= \int_{T(x)} \left(\int_0^\infty 1_{C^\epsilon \cap B_R(x)}(r\xi)\; r^{d-1}\mathrm{d}r \right) \omega_d(\mathrm{d}\xi) \\ &\leq \int_{T(x)} \left(\int_{\|x\|-R}^{\|x\|+R} 1_{C^\epsilon}(r\xi)\; r^{d-1}\mathrm{d}r \right) \omega_d(\mathrm{d}\xi) \\ &\leq 2\epsilon(\|x\| + R)^{d-1}\omega_d(T(x)) \end{aligned} \tag{25}$$

where for the final inequality we have used the fact that $C \cong \mathbb{S}^{d-1}$. The second inclusion leads to the bound

$$\lambda_{\text{Leb}}(B_{3R}(x)) \geq \lambda_{\text{Leb}}(S(x)) = \omega_d(T(x)) \int_{\|x\|-R}^{\|x\|+R} r^{d-1}\mathrm{d}r \geq \omega_d(T(x))2R(\|x\| - R)^{d-1}. \tag{26}$$

Combining (25) and (26) leads to the stated result. □

Our final auxiliary lemma is a standard property of distributions satisfying an interior cone condition, which appears in the standard convergence analysis of Metropolis–Hastings:

Lemma 10 (Interior cone condition for $C_{p(x)}$)**.** *Let $p \in \mathcal{P}_0(\mathbb{R}^d)$ and fix $\epsilon > 0$. Then there exists $\delta > 0$ and $R > 0$ such that, for all $\|x\| \geq R$, the cones*

$$K_\epsilon(x) = \{x - s\xi : 0 < s < \epsilon, \xi \in \mathbb{S}^{d-1}, \|\xi - n(x)\| \leq \delta/2\}$$

lie in the interior of the compact region bounded by $C_{p(x)}$ and each have a Lebesgue measure $\zeta_\epsilon > 0$ that is x-independent.

Proof. This is the content of the proof of Theorem 4.3 in Jarner and Hansen [2000]. □

B Implementation Detail

This section explains how all algorithms referred to in the main text were implemented. Appendix B.1 contains a brief introduction to DDPG; an established approach to approximating the policy gradient. Appendix B.2 contains full details for the ARWMH algorithm that was discussed in the main text. Appendix B.3 explains how our proposal was parametrised for RLMH to ensure that the regularity conditions of Theorem 2 were satisfied. The specific implementational details that are required to reproduce our results, together with a discussion of the associated computational costs, are contained in Appendix B.4.

B.1 Deep Deterministic Policy Gradient

This section describes the DDPG algorithm for maximising $J(\phi_\theta)$ based on the *deterministic policy gradient theorem* of Silver et al. [2014]. Since the algorithm itself is quite detailed, we simply aim to landmark the main aspects of DDPG and refer the reader to the original paper of Lillicrap et al. [2015] for further detail.

The deterministic policy gradient theorem states that

$$\nabla_\theta J(\phi_\theta) = \mathbb{E}_{s\sim D_\pi}\left[\nabla_\theta \pi(s)\; \nabla_a \mathrm{Q}_\pi(s,a)|_{a=\pi(s)} \right], \tag{27}$$

where the expectation here is taken with respect to the stationary distribution $s \sim D_\pi$ of the MDP when the policy π is fixed. The *action-value function* $\mathrm{Q}_\pi(s,a)$ gives the expected discounted cumulative reward from taking an action a in state s and following policy π thereafter[3].

Under DDPG, the policy π and the action-value function Q are parameterised by neural networks whose parameters are updated via an *actor-critic algorithm* based on Silver et al. [2014]. Specifically, an *actor network* $\pi_\theta(s)$ is

[3]Here, the notation for the action-value function Q is not to be confused with the notation Q, which denotes the Metropolis–Hastings proposal distribution in the main text.

updated in the direction of the policy gradient in (27), and a *critic network* that approximates the action-value function, $\mathrm{Q}_w(s,a) \approx \mathrm{Q}_\pi(s,a)$, is updated by solving the *Bellman equation* via stochastic approximation with off-policy samples from a *replay buffer*. The Bellman equation is the name given to the recursive relationship

$$\mathrm{Q}_\pi(s_n, a_n) = \mathbb{E}_{s_{n+1}\sim D_\pi}\left[r_n + \gamma \mathrm{Q}_\pi(s_{n+1}, \pi(s_{n+1}))\right],$$

and in DDPG the critic network is trained by solving the optimisation problem

$$\arg\min_w \mathbb{E}_{s_n\sim D_{\tilde{\pi}}, a_n\sim\tilde{\pi}}\left[(\mathrm{Q}_w(s_n,a_n) - y_n)^2\right], \tag{28}$$

where $y_n = r_n + \gamma \mathrm{Q}_w(s_{n+1}, \pi(s_{n+1}))$, a_n is generated from a stochastic *behaviour policy* $\tilde{\pi}$, $D_{\tilde{\pi}}$ is the stationary state distribution according to $\tilde{\pi}$, and s_{n+1} is resulted from interacting with the environment using $a_n \sim \tilde{\pi}(s_n)$. The expectation in (28) is approximated by sampling a mini-batch from a replay buffer that stores the experience tuples $\mathcal{R} := \{(s_n, a_n, r_n, s_{n+1})\}_{n=1}^H$. A common choice for $\tilde{\pi}$ is a noisy version of the deterministic policy π, e.g., $\tilde{\pi}(s) = \pi(s) + \varepsilon$, where $\varepsilon \sim \mathcal{N}(0, \Sigma_{\tilde{\pi}})$ with the covariance matrix $\Sigma_{\tilde{\pi}}$ that must be specified. Another popular choice is for ε to follow an Ornstein–Uhlenbeck process. Plappert et al. [2018] noticed that DDPG is still capable of learning successful policies even when run *on-policy*, meaning that $\varepsilon = 0$. This relates to the fact that the replay buffer is naturally off-policy, by keeping experiences from policies that were previously visited. The implementation of DDPG is summarised in Algorithm 2, where d_w and d_θ are determined by the architecture of the corresponding neural networks, and $\texttt{Env}(\cdot)$ denotes a function that encapsulates the environment.

There are several aspects of DDPG that are non-trivial, such as the distinction between *target networks* $\mathrm{Q}_{w'}$ and $\pi_{\theta'}$ and the current actor and critic networks Q_w and π_θ, and how these networks interact via the *taming factor* τ; see Lillicrap et al. [2015] for further detail. For the purpose of this work we largely relied on default settings for DDPG as implemented in `Matlab` R2024a; full details are contained in Appendix B.4. The specific design of RL methods for use in adaptive MCMC was not explored, but might be an interesting avenue for future work.

Algorithm 2 DDPG; Algorithm 1 in Lillicrap et al., 2015

Require: $s_1 \in \mathcal{S}$ (initial state), $w_1 \in \mathbb{R}^{d_w}$ (initial critic parameters), $\theta_1 \in \mathbb{R}^{d_\theta}$ (initial actor parameters), $T \in \mathbb{N}$ (number of iterations), $M \in \mathbb{N}$ (mini-batch size), $\gamma \in (0,1)$ (discount factor), $(\varsigma_i)_{i=1}^T \subset [0,\infty)$ (critic learning rate), $(\varrho_i)_{i=1}^T \subset [0,\infty)$ (actor learning rate), $\tau \in (0,1)$ (taming factor)

Initialise: $w'_1 = w_1$, $\theta'_1 = \theta_1$, $\mathcal{R}_1 = \{\}$

for $n = 1$ **to** T **do**

- $a_n \sim \tilde{\pi}(s_n)$ ▷ sample action from behaviour policy
- $(r_n, s_{n+1}) \leftarrow \texttt{Env}(a_n)$ ▷ interact with environment
- $\mathcal{R}_{n+1} \leftarrow \mathcal{R}_n \cup \{(s_n, a_n, r_n, s_{n+1})\}$ ▷ append experience to replay buffer
- $\{(s_{(i)}, a_{(i)}, r_{(i)}, s_{(i+1)})\}_{i=1}^M \sim \mathcal{R}_{n+1}$ ▷ sample a mini-batch uniformly from replay buffer
- $\{y_i \leftarrow r_{(i)} + \gamma \mathrm{Q}_{w'}(s_{(i+1)}, \pi_{\theta'}(s_{(i+1)}))\}_{i=1}^M$ ▷ compute target values using target networks
- $w_{n+1} \leftarrow w_n - \varsigma_n \frac{1}{M}\sum_{i=1}^M \nabla_w (\mathrm{Q}_w(s_{(i)}, a_{(i)}) - y_i)^2\big|_{w=w_n}$ ▷ update critic network
- $\theta_{n+1} \leftarrow \theta_n + \varrho_n \frac{1}{M}\sum_{i=1}^M \nabla_\theta \pi_\theta(s)|_{\substack{s=s_{(i)}\\ \theta=\theta_n}} \nabla_a \mathrm{Q}_w(s,a)|_{\substack{s=s_{(i)}, a=\pi(s_{(i)})\\ w=w_{n+1}}}$ ▷ update actor network
- $w'_{n+1} \leftarrow \tau w_{n+1} + (1-\tau) w'_n$ ▷ update target critic network
- $\theta'_{n+1} \leftarrow \tau \theta_{n+1} + (1-\tau)\theta'_n$ ▷ update target actor network

end for

B.2 Adaptive Metropolis–Hastings

This appendix describes the classical ARWMH algorithm was used to warm-start RLMH, and as a comparator in the empirical assessment reported in Section 4. The algorithm we used is formally called an *adaptive Metropolis algorithm with global adaptive scaling* in Andrieu and Thoms [2008], and full pseudocode is presented in Algorithm 3.

Algorithm 3 ARWMH; Algorithm 4 in Andrieu and Thoms, 2008

Require: $\alpha_\star \in (0,1)$ (target acceptance rate, here 0.234), $m \in \mathbb{N}$ (number of iterations), $(\gamma_i)_{i\geq 0} \subset [0,\infty)$ (learning rate)
  Initialise: $x_0 = 0$, $\mu_0 = 0$, $\Sigma_0 = I$, $\lambda_0 = 1$
  for $i = 1$ **to** m **do**
    $x_i^\star \sim \mathcal{N}(x_{i-1}, \lambda_{i-1}\Sigma_{i-1})$ ▷ propose next state
    $\alpha_i \leftarrow \min(1, p(x_i^\star)/p(x_{i-1}))$ ▷ acceptance probability
    $x_i \leftarrow x_i^\star$ with probability α_i, else $x_i \leftarrow x_{i-1}$ ▷ accept/reject
    $\log(\lambda_i) \leftarrow \log(\lambda_{i-1}) + \gamma_{i-1}(\alpha_i - \alpha_\star)$ ▷ refine proposal scale
    $\mu_i \leftarrow \mu_{i-1} + \gamma_{i-1}(x_i - \mu_{i-1})$ ▷ update mean approximation
    $\Sigma_i \leftarrow \Sigma_{i-1} + \gamma_{i-1}\left[(x_i - \mu_{i-1})(x_i - \mu_{i-1})^\top - \Sigma_{i-1}\right]$ ▷ update covariance approximation
  end for

In brief, Algorithm 3 constructs a sequence of approximations Σ_n to the covariance matrix of the target $p(\cdot)$, and then proposes new states $x_{n+1}^\star$ using a Gaussian distribution centred at the current state x_n with covariance $\lambda_n\Sigma_n$. The prefactor λ_n is selected in such a manner that the proportion of accepted proposals aims to approach 0.234, a value that is theoretically supported [Gelman et al., 1997, Yang et al., 2020].

Remark 4 (AMH as ϕ-MH)**.** *The ARWMH algorithm in Algorithm 3 can be viewed as an instance of ϕ-MH with the building block proposals $q_\varphi(\cdot|x)$ being Gaussian with mean x and covariance $\varphi \in \mathrm{S}_d^+$, where the algorithm attempts to learn a* constant *function $\phi : \mathbb{R}^d \to \mathrm{S}_d^+$.*

For warm starting of RLMH we performed $m = 10^4$ iterations of ARWMH to obtain samples $(x_i)_{i=-m+1}^0$, and we took the matrix Σ appearing in (5) to be the matrix Σ_m obtained as the sample average of the final third of samples generates from ARWMH. This approach enables us to exploit a rough approximation of the covariance of $p(\cdot)$, which is important in higher-dimensional problems, while removing the burden of simultaneously learning a proposal covariance, in addition to a proposal mean, in our set-up for RLMH.

B.3 Parametrisation of the Policy

The aim of this appendix is to explain how the maps ϕ_θ were parametrised for the experiments that we performed, and to explain how our choice of parametrisation ensured the conditions of Theorem 2 were satisfied.

Let $\psi_\theta : \mathbb{R}^d \to \mathbb{R}^d$ be a collection of functions indexed by $\theta \in \mathbb{R}^p$ such that $(\theta, x) \mapsto \psi_\theta(x)$ is locally Lipschitz over $(\theta, x) \in \mathbb{R}^p \times \mathbb{R}^d$. Let $C \subset \mathbb{R}^d$ be a compact set and let $\gamma_C : \mathbb{R}^d \to \mathbb{R}$ be a smooth function with $\gamma_C(x) = 1$ on $x \in C^{\mathsf{c}}$. Armed with these tools, we propose to set

$$\phi_\theta(x) := \psi_\theta(x) + \gamma_C(x)[x - \psi_\theta(x)] \tag{29}$$

for all $x \in \mathbb{R}^d$ and $\theta \in \mathbb{R}^p$. The construction in (29) ensures that the regularity conditions (i)-(iii) of Theorem 2 are satisfied, so that the ergodicity of RLMH can be guaranteed. Intuitively, the proposal (5) will default to a random walk proposal $\phi_\theta(x) = x$ when the state x is outside of the set C, as a consequence $\gamma_C(x) = 1$ in (29), while when $x \in C$ there is an opportunity to learn a flexible mean $\psi_\theta(x)$ for the proposal (5). First we establish that the regularity requirements (i)-(iii) of Theorem 2 are indeed satisfied, and then specific choices for ψ_θ, C, and γ_C will be presented.

Property (i): For any compact $\Theta_0 \subset \mathbb{R}^p$,

$$\sup_{\theta\in\Theta_0}\sup_{x\in\mathbb{R}^d} \|x - \phi_\theta(x)\| \leq \sup_{\theta\in\Theta_0}\sup_{x\in C} \|x - \phi_\theta(x)\| < \infty,$$

where the first inequality holds since $x - \phi_\theta(x)$ vanishes when $x \in C^{\mathsf{c}}$, and the second inequality holds since $(\theta, x) \mapsto x - \phi_\theta(x)$ is continuous, and hence bounded on the compact set $\Theta_0 \times C$.

Property (ii): For any compact $\Theta_0 \subset \mathbb{R}^p$,

$$\begin{aligned}\sup_{\theta\in\Theta_0} \mathrm{Lip}(x \mapsto \phi_\theta(x)) &= \sup_{\theta\in\Theta_0} \sup_{x\in\mathbb{R}^d} \mathrm{LocLip}_x(y \mapsto \phi_\theta(y)) \\ &\leq 1 + \sup_{\theta\in\Theta_0} \sup_{x\in C} \mathrm{LocLip}_x(y \mapsto \phi_\theta(y)) \\ &\leq 1 + \underbrace{\sup_{\theta\in\Theta_0} \sup_{x\in C} \mathrm{LocLip}_{(\theta,x)}((\vartheta, y) \mapsto \phi_\vartheta(y))}_{(*)} < \infty, \end{aligned} \tag{30}$$

where the first equality is the definition of the Lipschitz constant, the first inequality holds since $\phi_\theta(x)$ is the identity on $x \in C^{\mathsf{c}}$ with unit Lipschitz constant, the second inequality holds via set inclusion, and the final inequality holds since $\Theta_0 \times C$ is compact and therefore $(\theta, x) \mapsto \phi_\theta(x)$ is Lipschitz when restricted to $\Theta_0 \times C$, with $(*)$ the corresponding Lipschitz constant.

Property (iii): For any compact $\Theta_0 \subset \mathbb{R}^p$,

$$\begin{aligned}\sup_{\theta\in\Theta_0} \sup_{x\in\mathbb{R}^d} \mathrm{LocLip}_\theta(\vartheta \mapsto \phi_\vartheta(x)) &= \sup_{\theta\in\Theta_0} \sup_{x\in C} \mathrm{LocLip}_x(y \mapsto \phi_\theta(y)) \\ &\leq \sup_{\theta\in\Theta_0} \sup_{x\in C} \mathrm{LocLip}_{(\theta,x)}((\vartheta, y) \mapsto \phi_\vartheta(y)) < \infty,\end{aligned}$$

where the equality holds since $\phi_\theta(x)$ is constant in θ when $x \in C^{\mathsf{c}}$, the first inequality holds by set inclusion, and the final inequality was established in (30).

To implement the construction in (29) we need to specify a parametric map $\psi_\theta : \mathbb{R}^d \to \mathbb{R}^d$, a compact set $C \subset \mathbb{R}^d$, and a smooth function γ_C that vanishes on C^{c}. For the experiments that we report in this manuscript we took:

- $\psi_\theta(x) = \bar{x} + \Sigma^{1/2}\nu_\theta(x)$, where $\bar{x}$ is the mean and $\Sigma \in \mathrm{S}_d^+$ is the covariance matrix obtained from the warm-up samples $(x_i)_{i=-m+1}^0$ as explained in Section 4, and $\nu_\theta : \mathbb{R}^d \to \mathbb{R}^d$ is a neural network whose architecture and initialisation are detailed in Appendix B.4.
- the set C was taken to be an ellipsoid

  $$C = \{x \in \mathbb{R}^d : \eta(x) \leq 1\}, \qquad \eta(x) := \frac{\|\Sigma^{-1/2}(x - \bar{x})\|^2}{\ell^2},$$

  where $\ell > 0$ is a radius to be specified.
- the map γ_C was taken to be the smooth transition function $\gamma_C(x) = \gamma(\eta(x))$ where

  $$\gamma(\eta) := \begin{cases} 0 & \eta \in [0, 1/2] \\ \left[1 + \exp\left(-\frac{4\eta-3}{4\eta^2-6\eta+2}\right)\right]^{-1} & \eta \in [1/2, 1] \\ 1 & \eta \in [1, \infty) \end{cases}$$

  which satisfies the smoothness requirement and is identically one when $x \in C^{\mathsf{c}}$.

The radius ℓ of the ellipsoid C was set to $\ell = 10$, representing approximately 10 standard deviations from the mean of $p(\cdot)$, which ensures that the symmetric random walk behaviour (that occurs under (29) when $x \in C^{\mathsf{c}}$) is rarely encountered.

B.4 Training Details

This section contains the implementational details required to reproduce the experimental results reported in Section 4.

Parametrisation of ϕ_θ As explained in Appendix B.3, the function ϕ_θ was parametrised using a flexible parametric map $\nu_\theta : \mathbb{R}^d \to \mathbb{R}^d$. For our experiments we took ν_θ to be a fully-connected two-layer neural network with the ReLU activation function and 32 features in the hidden layer; a total of $p = (32 + d)(d + 1)$ parameters to be inferred.

Pre-training of ϕ_θ The parameters θ of the neural network ν_θ were initialised by pre-training against the loss

$$\theta \mapsto \frac{1}{m}\sum_{i=1}^{m}\left\|\Sigma^{-\frac{1}{2}}(\bar{x}-x_{i-m})-\nu_\theta(x_{i-m})\right\|^2,$$

computed over the warm-up samples $(x_i)_{i=-m+1}^{0}$ generated from ARWMH, so that the proposal corresponding to ν_θ approximates the anti-correlated behaviour illustrated in Example 2 of the main text.

Optimisation was conducted using the Deep Learning toolbox in `Matlab` R2024a, with the ADAM optimisation method [Kingma and Ba, 2014]. The default settings of the toolbox were employed. Pre-training was terminated once either the mean squared error validation loss (computed using a held out subset of 30% of the dataset) fell below the threshold '1', or a maximum of 2000 epochs was reached. Upon termination, the network with the best validation loss was returned.

Training of ϕ_θ RLMH was performed using the implementation of DDPG provided in the RL toolbox of `Matlab` R2024a. The default settings for training ϕ_θ using this toolbox were employed with:

- 100 episodes, each consisting of 500 iterations of MCMC
- standard deviation of the Ornstein–Uhlenbeck process noise is 0 (c.f. Remark 3)
- actor learning rate = 10^{-6}
- gradient clipping threshold = $\frac{d}{\|\Sigma\|_F^2} \wedge 10^{-5}$
- experience buffer length = 10^6,

where $\|\Sigma\|_F$ denotes the Frobenius norm of $\Sigma \in \mathrm{S}_d^+$.

Parametrisation of the Critic Q For all experiments we took $\mathsf{Q} : \mathbb{R}^{2d} \times \mathbb{R}^{2d} \to \mathbb{R}$ to be a fully-connected two-layer neural network with the ReLU activation function and 8 features in the hidden layer; a total of $(8+4d)(4d+1)$ parameters to be inferred.

Training of the Critic Q The default settings for training the critic Q using the RL toolbox in `Matlab` R2024a were employed, except for the maximum size of the replay buffer which was set to be 10^6; large enough to retain the full sample path of the Markov chain.

Computation All computation was performed on a desktop PC with a 12th generation Intel i9-12900F (24) @ 2.419GHz CPU, 32GB RAM, and NVIDIA GeForce RTX 3060 Ti GPU. The (median) average time required to perform RLMH on a single task from the `PosteriorDB` benchmark was 132 seconds.

Note that such specifications are not required to run the experiments that we report; in particular it is not required to have access a GPU.

C Additional Empirical Details and Results

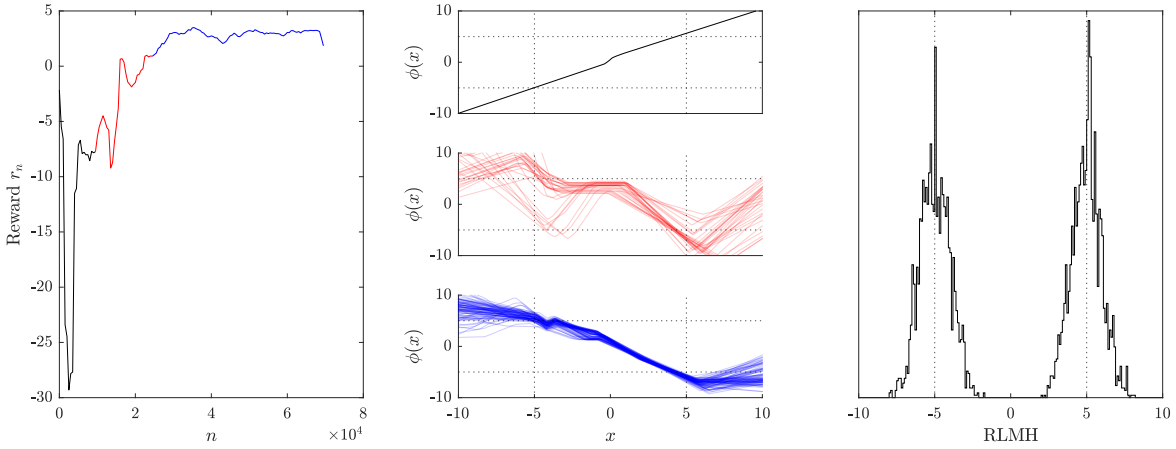
The sensitivity of our experiments to the choice of the neural network architecture is examined in Appendix C.1. A selection of additional illustrations of RLMH are presented in Appendix C.2. Appendix C.3 describes how MALA was implemented. The performance measures that we used for assessment are precisely defined in Appendix C.4. Full results for `PosteriorDB` are contained in Appendix C.5.

C.1 Choice of Neural Network

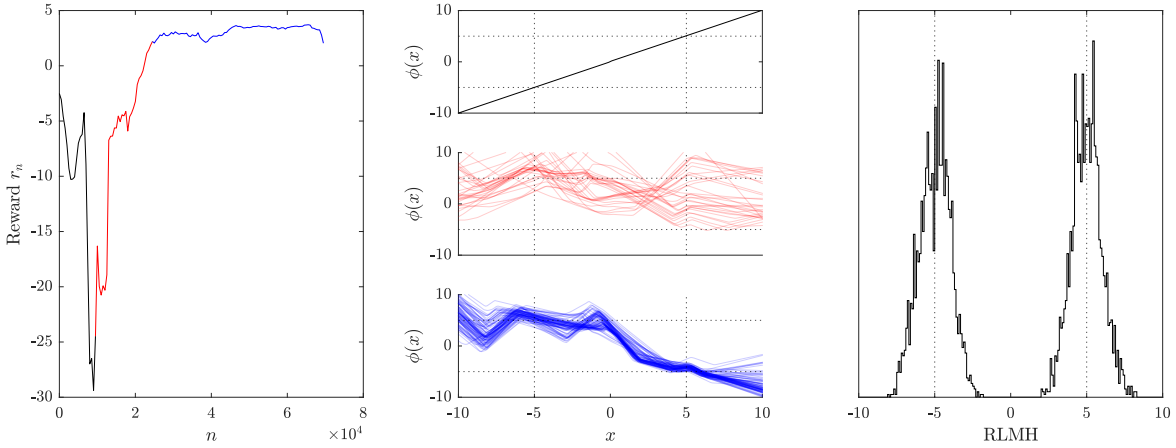
The sophistication of modern RL methodologies, such as DDPG, means that in practice there are several design choices to be specified. For the present work we largely relied on the default settings provided in the RL toolbox of `Matlab` R2024a, but it is still necessary for us to select the neural architectures that are used. The aim of this

appendix is to briefly explore the consequences of varying the neural architecture for ϕ_θ in the context of the simple example from Figure 1, to understand the sensitivity of RLMH to the choice of neural network.

Results are displayed in Figure 2. For these experiments all settings were identical to that of Figure 1, with the exception of gradient clipping; since the number of parameters $\dim(\theta)$ in the neural network ϕ_θ depends on the architecture of the neural network, the gradient clipping threshold τ in Algorithm 1 was adjusted accordingly. These results broadly indicate an insensitivity to the architecture of the neural network used to construct the proposal mean ϕ_θ in RLMH. Specifically, both narrower and wider architectures, and also deeper architectures, all led to the same global mode-hopping proposal reported in Figure 1 of the main text.



(a) $h = 1$, $w = 16$

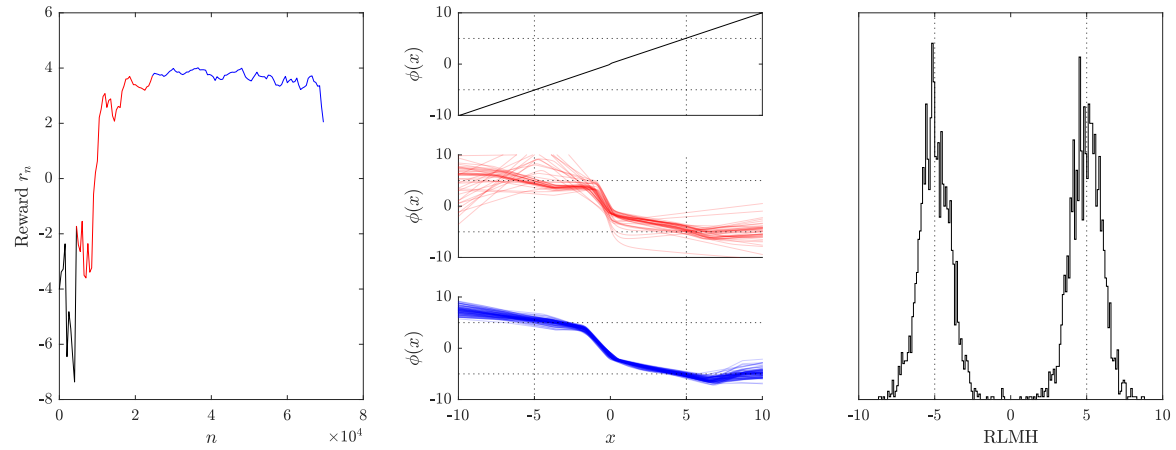


(b) $h = 1$, $w = 64$

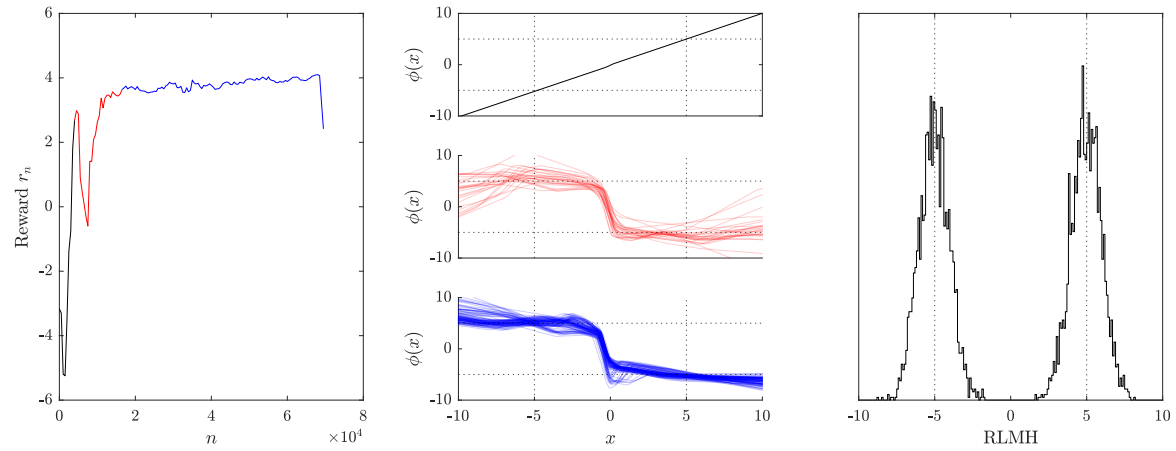
Figure 2: Investigating sensitivity to the architecture of the neural network ϕ in RLMH. For the experiment presented in Figure 1 of the main text we employed a two layer (i.e. $h = 1$ hidden layer) neural network with width $w = 32$. The same experiment was performed with the architecture dimensions (h, w) changed to (a) (1,16), (b) (1,64), (c) (1,256), (d) (2,32), and (e) (3,32); in all cases similar conclusions were obtained. [The colour convention and the interpretation of each panel is identical to that of Figure 1 in the main text.]

C.2 Additional Illustrations in 1D and 2D

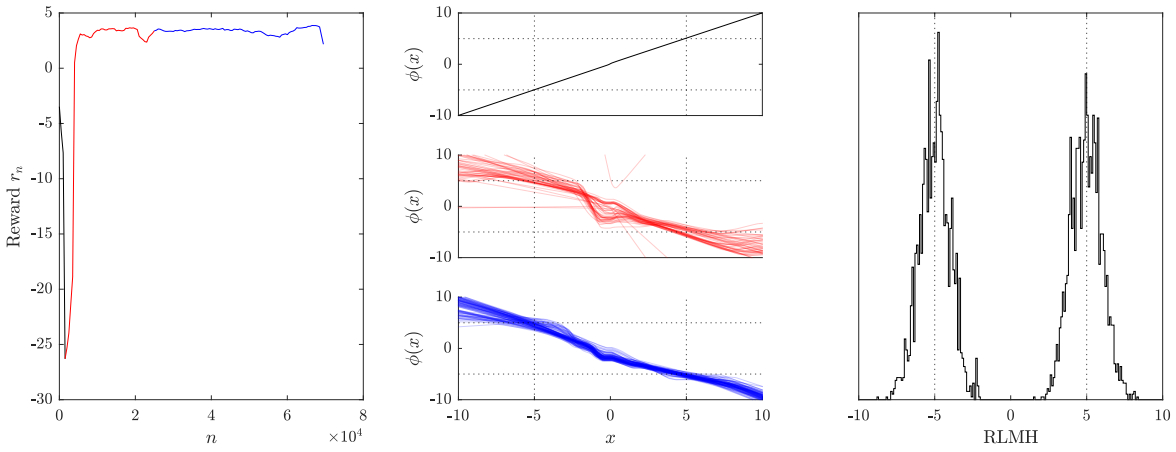
This appendix supplements Figure 1 in the main text with additional illustrations, corresponding to different target distributions $p(\cdot)$ in dimensions $d = 1$ and $d = 2$. Specifically, we consider (a) a skewed target, (b) a skewed multimodal target, and (c) an unequally-weighted mixture model target in dimension $d = 1$, and a Gaussian mixture model target in dimension $d = 2$. Results are reported in Figure 3 (for $d = 1$) and Figure 4 (for $d = 2$).



(c) $h = 1$, $w = 256$



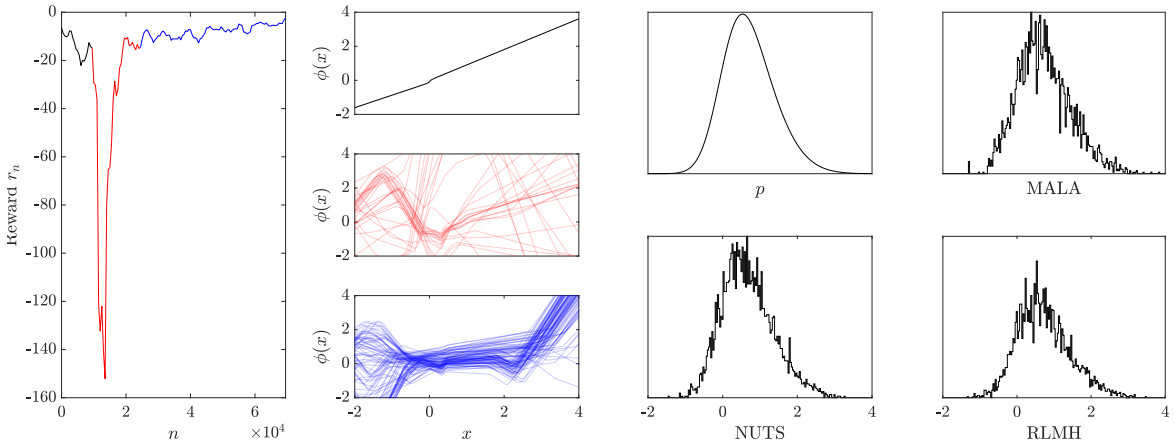
(d) $h = 2$, $w = 32$



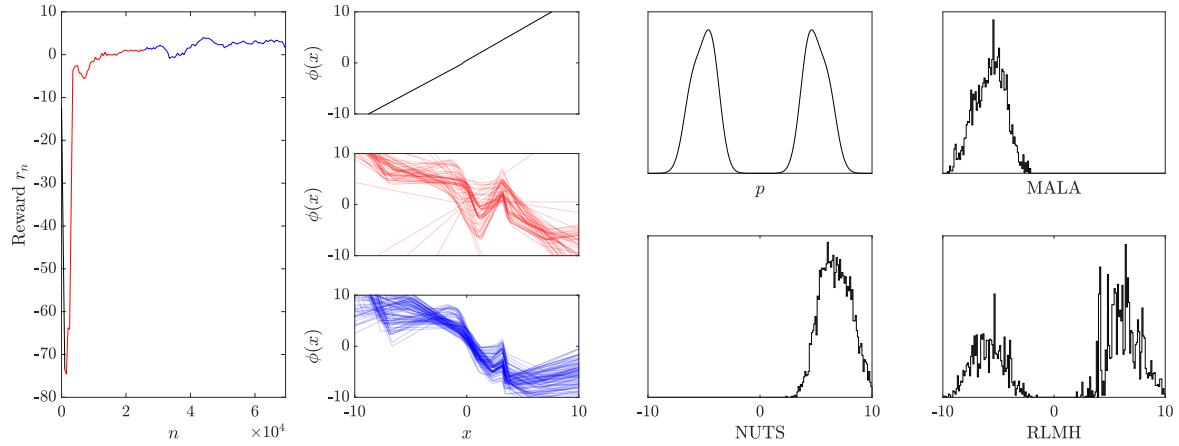
(e) $h = 3$, $w = 32$

Figure 2: Investigating sensitivity to the architecture of the neural network ϕ in RLMH, continued.

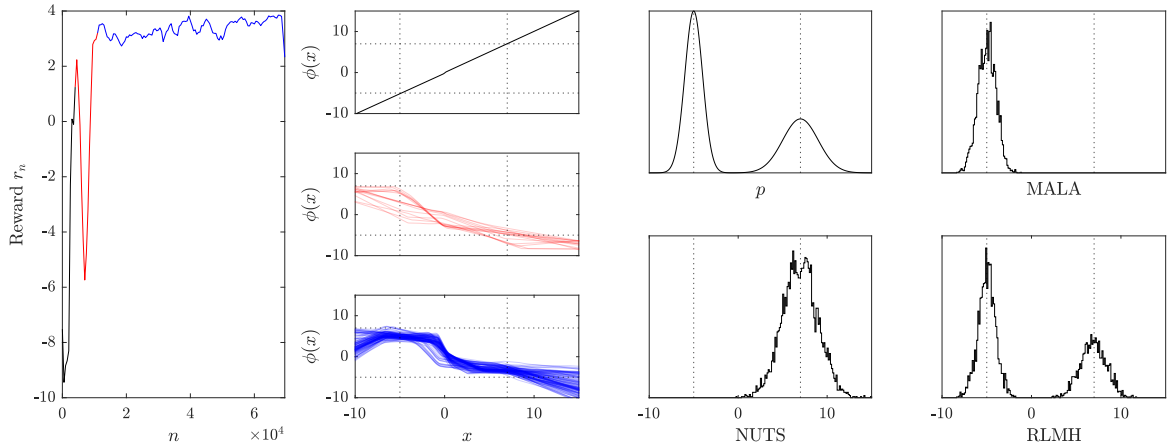
These examples suggest that the gradient-free version of RLMH can learn rapidly mixing Markov transition kernels for a range of different targets.



(a) Skewed



(b) Skewed multimodal



(c) Unequal mixture model

Figure 3: RLMH, illustrated. Here we considered (a) a skewed target, (b) a skewed multimodal target, and (c) an unequally-weighted mixture model target. Left: The reward sequence $(r_n)_{n\geq 0}$, where r_n is the logarithm of the expected squared jump distance corresponding to iteration n of RLMH. Middle: Proposal mean functions $x \mapsto \phi(x)$, at initialisation (top), and corresponding to the rewards indicated in red (middle) and blue (bottom). Right: The density $p(\cdot)$, and histograms of the last $n = 5,000$ samples produced using MALA, NUTS, and RLMH. [A smoothing window of length 5 was applied to the reward sequence to improve clarity of this plot.]

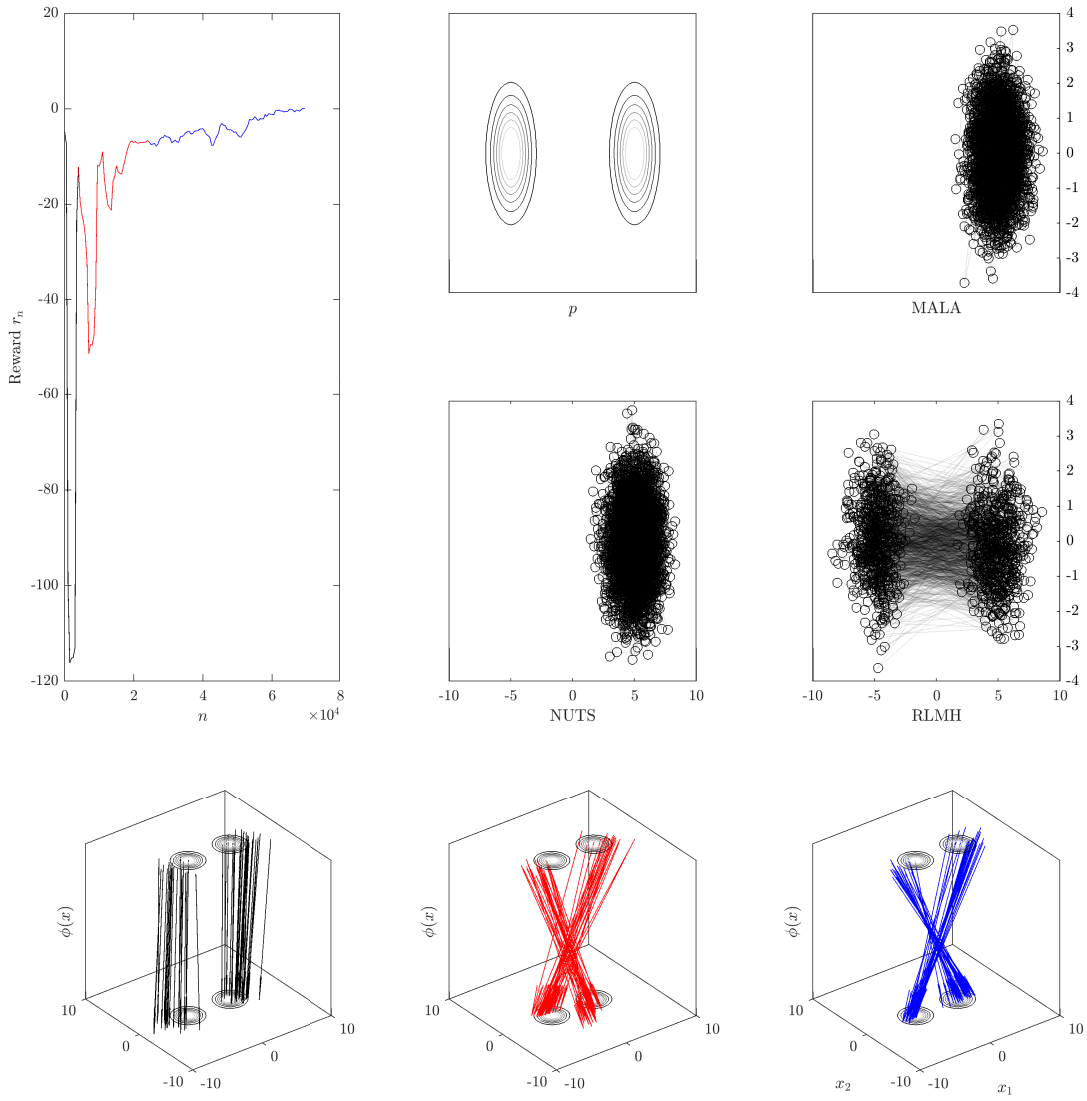


Figure 4: RLMH, illustrated. Here we considered a two-dimensional Gaussian mixture model target. Top Left: The reward sequence $(r_n)_{n\geq 0}$, where r_n is the logarithm of the expected squared jump distance corresponding to iteration n of RLMH. Top Right: The density $p(\cdot)$, and histograms of the last $n = 5,000$ samples produced using MALA, NUTS, and RLMH. Bottom: Proposal mean functions $x \mapsto \phi(x)$, at initialisation (left), and corresponding to a typical policy from the period whose rewards are indicated in red (middle) and blue (right). [A smoothing window of length 5 was applied to the reward sequence to improve clarity of this plot. In the bottom row we indicate the current state x_n and the proposed state $x^\star_{n+1}$ of the Markov chain using a directed arrow from x_n to $x^\star_{n+1}$.]

C.3 Adaptive Metropolis-Adjusted Langevin Algorithm

Algorithm 4 contains pseudocode for an adaptive version of the (preconditioned) *Metropolis-adjusted Langevin algorithm* (MALA) algorithm of Roberts and Tweedie [1996a]. For the experiments that we report, we implemented this AMALA in `Matlab` R2024a.

Algorithm 4 $\texttt{MALA}(x_0, \epsilon, \Sigma, n)$; Roberts and Tweedie, 1996a

Require: $x_0 \in \mathbb{R}^d$ (initial state), $\epsilon > 0$ (scale of proposal), $\Sigma \in \mathrm{S}_d^+$ (preconditioner matrix), $n \in \mathbb{N}$ (number of iterations)

for $i = 1$ **to** n **do**

$x_i^\star \leftarrow \underbrace{x_{i-1} + \epsilon\Sigma(\nabla \log p)(x_{i-1})}_{=:\nu(x_{i-1})} + (2\epsilon\Sigma)^{1/2} Z_i$ ▷ propose new state; $Z_i \sim \mathcal{N}(0, I)$

$\alpha_i \leftarrow \min\left(1, \frac{p(x_i^\star)}{p(x_{i-1})} \frac{\exp\left(-\frac{1}{4\epsilon}\|\Sigma^{-1/2}(x_{i-1} - \nu(x_i^\star))\|^2\right)}{\exp\left(-\frac{1}{4\epsilon}\|\Sigma^{-1/2}(x_i^\star - \nu(x_{i-1}))\|^2\right)}\right)$ ▷ acceptance probability

$x_i \leftarrow x_i^\star$ with probability α_i, else $x_i \leftarrow x_{i-1}$ ▷ accept/reject

end for

Return: $\{x_1, \ldots, x_n\}$

For implementation of (non-adaptive) MALA, we are required to specify a step size $\epsilon > 0$ and a preconditioner matrix $\Sigma \in \mathrm{S}_d^+$ in Algorithm 4. In general, suitable values for both of these parameters will be problem-dependent, and eliciting suitable values can be difficult [Livingstone and Zanella, 2022]. Standard practice is to perform some form of manual or automated tuning to arrive at parameter values for which the average acceptance rate is close to 0.574, motivated by the asymptotic analysis of Roberts and Rosenthal [1998]. For the purpose of this work we implemented a particular adaptive version of MALA used in recent work such as Wang et al. [2023], which for completeness is described in Algorithm 5.

Algorithm 5 Adaptive MALA

Require: $x_{0,0} \in \mathbb{R}^d$ (initial state), $\epsilon_0 > 0$ (initial scale of proposal), $\Sigma_0 \in \mathrm{S}_d^+$ (initial preconditioner matrix), $\{n_i\}_{i=0}^{h-1}$ (epoch lengths), $(\alpha_i)_{i=1}^{h-1} \subset (0, \infty)$ (learning schedule), $E \in \mathbb{N}$ (number of epochs)

$\{x_{0,1} \ldots, x_{0,n_0}\} \leftarrow \texttt{MALA}(x_{0,0}, \epsilon_0, \Sigma_0, n_0)$

for $i = 1, \ldots, E-1$ **do**

  $x_{i,0} \leftarrow x_{i-1,n_{i-1}}$

  $\rho_{i-1} \leftarrow \frac{1}{n_{i-1}} \sum_{j=1}^{n_{i-1}} 1_{x_{i-1,j} \neq x_{i-1,j-1}}$ ▷ average acceptance rate for epoch i

  $\epsilon_i \leftarrow \epsilon_{i-1} \exp(\rho_{i-1} - 0.574)$ ▷ update scale of propsoal

  $\Sigma_i \leftarrow \alpha_i \Sigma_i + (1 - \alpha_i)\mathrm{cov}(\{x_{i-1,1} \ldots, x_{i-1,n_{i-1}}\})$ ▷ update preconditioner matrix

  $\{x_{i,1} \ldots, x_{i,n_i}\} \leftarrow \texttt{MALA}(x_{i,0}, \epsilon_i, \Sigma_i, n_i)$

end for

Return: $\{x_{0,1}, \ldots, x_{E-1,n_{E-1}}\}$

For the pseudocode in Algorithm 5, we use $\texttt{MALA}(x, \epsilon, \Sigma, n)$ to denote the output from Algorithm 4, and we use $\mathrm{cov}(\cdot)$ to denote the sample covariance matrix. The algorithm monitors the average acceptance rate and increases or decreases it according to whether it is below or above, respectively, the 0.574 target. For the preconditioner matrix, the sample covariance matrix of samples obtained from the previous run of MALA is used. For all experiments that we report using AMALA, we employed identical settings to those used in Wang et al. [2023]. Namely, we set $\epsilon_0 = 1$, $\Sigma_0 = I_d$, $E = 10$, and $\alpha_1 = \cdots = \alpha_9 = 0.3$. The warm-up epoch lengths were $n_0 = \cdots = n_8 = 1,000$ and the final epoch length was $n_9 = 10^5$. The samples $\{x_{E-1,1}, \ldots, x_{E-1,n_{E-1}}\}$ from the final epoch were returned, and constituted the output from AMALA that was used for our experimental assessment.

C.4 Performance Measures

This section precisely defines the performance measures that were used as part of our assessment; *expected squared jump distance* (ESJD), and *maximum mean discrepancy* (MMD). For the comparison of adaptive MCMC methods,

we disabled adaptation after the initial training period in order to generate additional pairs $\{(x_{i-1}, x_i^\star)\}_{i=1}^n$ with $n = 5,000$, from which the ESJD and MMD were calculated.

Expected Squared Jump Distance The ESJD in each case was consistently estimated using

$$\frac{1}{n}\sum_{i=1}^{n} \|x_i - x_{i-1}\|^2,$$

the actual squared jump distance averaged over the sample path. In principle a Rao–Blackwellised estimator could also be used, analogous to how the rewards r_n are calculated in RLMH, but we preferred to use the above simpler estimator as it generalises as a performance metric beyond Metropolis–Hastings MCMC.

Maximum Mean Discrepancy The MMD $D(P_m, Q_n)$ between a pair of empirical distributions $P_m = \frac{1}{m}\sum_{i=1}^m \delta_{x_i}$ and $Q_n = \frac{1}{n}\sum_{j=1}^n \delta_{y_j}$ is defined via the formula

$$\mathrm{MMD}(P_m, Q_n)^2 := \frac{1}{m^2}\sum_{i=1}^{m}\sum_{i'=1}^{m} k(x_i, x_{i'}) - \frac{2}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} k(x_i, y_j) + \frac{1}{n^2}\sum_{j=1}^{n}\sum_{j'=1}^{n} k(y_j, y_{j'}),$$

and for this work we took the kernel k to be the Gaussian kernel

$$k(x, y) := \exp\left(-\frac{\|x - y\|^2}{\ell^2}\right)$$

where the length-scale $\ell > 0$ was set according to the *median heuristic*

$$\ell := \frac{1}{2}\mathrm{median}\{\|y_i - y_j\| : 1 \le i, j \le n\}$$

following Garreau et al. [2017]. Here P_m represents the approximation to the target $p(\cdot)$ produced using an adaptive MCMC algorithm, and Q_n represents a gold-standard set of $n = 10^4$ samples from the target, which are provided in `PosteriorDB`.

C.5 Full Results on `PosteriorDB`

`PosteriorDB` is an attempt toward standardised benchmarking, consisting of a collection of posteriors to be numerically approximated [Magnusson et al., 2022]. The test problems are defined in the `Stan` probabilistic programming language, and `BridgeStan` [Roualdes et al., 2023] was used to directly access posterior densities and their gradients as required. At the time we conducted our research, `PosteriorDB` was at version 0.5.0 and contained 120 models, all of which came equipped with a gold-standard sample of size $n = 10^4$, generated from a long run of NUTS. Of these models, a subset of 44 were found to be compatible with `BridgeStan`, which was at version 2.4.1 at the time this research was performed. The version of `Stan` that we used was `Stanc3` version 2.34.0 (Unix). Thus we used a total of 44 test problems for our empirical assessment.

To implement ARWMH it is required to specify the learning rate $(\gamma_i)_{i\ge 0}$ appearing in Algorithm 3. Following Section 4.2.2 of Andrieu and Thoms [2008], we implemented a learning rate of the form

$$\gamma_i = \frac{1}{2 \cdot (i+1)^\beta} \tag{31}$$

where the exponent $\beta \in (0, 1)$ was manually selected on a per-task basis to deliver the best performance for each of the 44 tasks from `PosteriorDB`. Though it is possible to determine γ_i at runtime, using techniques such as those described in Delyon and Juditsky [1993], the simple schedule in (31) is most widely-used and performed reasonably well for most of the 44 tasks we considered. The task-specific exponents β that we used are included in the code used to produce these results, included as part of the electronic supplement.

Full results are contained in Table 2, which is an expanded version of Table 1 in the main text.

Task	d	RLMH		ARWMH		AMALA	
		ESJD	MMD	ESJD	MMD	ESJD	MMD
earnings-earn_height	3	**4.5(0.6)E3**	**1.8(0.1)E-1**	2.1(0.0)E3	1.5(0.0)E0	1.3(0.9)E3	1.8(0.3)E0
earnings-log10earn_height	3	**1.4(0.0)E-1**	**1.6(0.0)E-1**	4.4(0.0)E-2	1.4(0.0)E0	1.4(0.0)E-1	1.6(0.0)E-1
earnings-logearn_height	3	**3.2(0.0)E-1**	**1.6(0.0)E-1**	1.0(0.0)E-1	1.5(0.0)E0	3.3(0.0)E-1	1.7(0.0)E-1
gp_pois_regr-gp_regr	3	**3.7(0.1)E-1**	**1.2(0.0)E-1**	1.2(0.0)E-1	1.1(0.0)E0	3.6(0.0)E-1	1.2(0.0)E-1
kidiq-kidscore_momhs	3	**1.3(0.2)E0**	**1.5(0.0)E-1**	7.2(0.0)E-1	1.3(0.0)E0	2.3(0.0)E0	1.4(0.0)E-1
kidiq-kidscore_momiq	3	**3.6(0.2)E0**	**1.7(0.0)E-1**	1.3(0.0)E0	1.5(0.0)E0	4.2(0.0)E0	1.6(0.0)E-1
kilpisjarvi_mod-kilpisjarvi	3	**1.3(0.1)E1**	**1.7(0.0)E-1**	6.5(0.0)E0	1.5(0.0)E0	6.1(3.1)E0	1.2(0.2)E0
mesquite-logmesquite_logvolume	3	**1.2(0.0)E-1**	**1.3(0.0)E-1**	3.7(0.0)E-2	1.1(0.0)E0	1.1(0.0)E-1	1.3(0.0)E-1
arma-arma11	4	**6.4(0.0)E-2**	**1.2(0.0)E-1**	1.9(0.0)E-2	1.1(0.0)E0	3.7(1.0)E-2	9.2(3.3)E-1
earnings-logearn_height_male	4	**4.1(0.1)E-1**	**1.6(0.0)E-1**	1.2(0.0)E-1	1.5(0.0)E0	4.2(0.1)E-1	1.6(0.0)E-1
earnings-logearn_logheight_male	4	**1.7(0.1)E0**	**1.6(0.0)E-1**	5.3(0.0)E-1	1.5(0.0)E0	1.8(0.0)E0	1.6(0.0)E-1
garch-garch11	4	**8.0(0.2)E-1**	**1.4(0.0)E-1**	2.8(0.0)E-1	1.2(0.0)E0	7.1(0.1)E-1	1.4(0.0)E-1
hmm_example-hmm_example	4	**4.6(0.1)E-1**	**1.3(0.0)E-1**	1.4(0.0)E-1	1.2(0.0)E0	4.6(0.1)E-1	1.3(0.0)E-1
kidiq-kidscore_momhsiq	4	**2.7(0.2)E0**	**1.4(0.0)E-1**	1.3(0.0)E0	1.3(0.0)E0	4.4(0.1)E0	1.4(0.0)E-1
earnings-logearn_interaction	5	**8.8(0.3)E-1**	**1.4(0.0)E-1**	2.9(0.0)E-1	1.2(0.0)E0	1.1(0.0)E0	1.4(0.0)E-1
earnings-logearn_interaction_z	5	**8.7(0.1)E-2**	**1.2(0.0)E-1**	2.7(0.0)E-2	1.1(0.0)E0	9.1(0.1)E-2	1.2(0.0)E-1
kidiq-kidscore_interaction	5	**5.5(0.5)E0**	**1.7(0.1)E-1**	3.8(0.0)E0	1.3(0.0)E0	1.4(0.0)E1	1.4(0.0)E-1
kidiq_with_mom_work-kidscore_interaction_c	5	**7.2(0.5)E-1**	**1.6(0.0)E-1**	5.2(0.0)E-1	1.3(0.0)E0	1.8(0.0)E0	1.3(0.0)E-1
kidiq_with_mom_work-kidscore_interaction_c2	5	**7.3(0.6)E-1**	**1.7(0.0)E-1**	5.3(0.0)E-1	1.3(0.0)E0	1.9(0.0)E0	1.4(0.0)E-1
kidiq_with_mom_work-kidscore_interaction_z	5	**1.0(0.1)E0**	**1.5(0.1)E-1**	1.0(0.0)E0	1.1(0.0)E0	3.5(0.0)E0	1.2(0.0)E-1
kidiq_with_mom_work-kidscore_mom_work	5	9.6(1.3)E-1	**1.8(0.1)E-1**	**1.2(0.0)E0**	1.1(0.0)E0	4.2(0.0)E0	1.2(0.0)E-1
low_dim_gauss_mix-low_dim_gauss_mix	5	**6.7(0.0)E-2**	**1.1(0.0)E-1**	2.1(0.0)E-2	9.9(0.0)E-1	7.0(0.1)E-2	1.1(0.0)E-1
mesquite-logmesquite_logva	5	**2.5(0.0)E-1**	**1.2(0.0)E-1**	7.5(0.0)E-2	1.1(0.0)E0	2.6(0.0)E-1	1.2(0.0)E-1
bball_drive_event_0-hmm_drive_0	6	**4.6(0.5)E-1**	**1.6(0.3)E-1**	1.8(0.0)E-1	1.1(0.0)E0	6.2(0.3)E-1	1.4(0.1)E-1
sblrc-blr	6	**4.2(0.1)E-2**	**1.7(0.0)E-1**	1.4(0.0)E-2	1.5(0.0)E0	4.6(0.0)E-2	1.6(0.0)E-1
sblri-blr	6	**4.2(0.1)E-2**	**1.7(0.0)E-1**	1.3(0.0)E-2	1.5(0.0)E0	4.5(0.1)E-2	1.6(0.0)E-1
arK-arK	7	**1.2(0.0)E-1**	**1.1(0.0)E-1**	3.5(0.0)E-2	9.5(0.0)E-1	1.4(0.0)E-1	1.1(0.0)E-1
mesquite-logmesquite_logvash	7	**3.6(0.1)E-1**	**1.1(0.0)E-1**	1.1(0.0)E-1	9.9(0.0)E-1	4.1(0.0)E-1	1.1(0.0)E-1
mesquite-logmesquite	8	**3.3(0.0)E-1**	**1.1(0.0)E-1**	1.1(0.0)E-1	9.5(0.0)E-1	4.1(0.1)E-1	1.1(0.0)E-1
mesquite-logmesquite_logvas	8	**3.4(0.1)E-1**	**1.1(0.0)E-1**	1.1(0.0)E-1	9.6(0.0)E-1	4.1(0.0)E-1	1.1(0.0)E-1
mesquite-mesquite	8	2.0(0.4)E1	**5.1(1.1)E-1**	**6.5(0.0)E1**	9.6(0.0)E-1	2.5(0.0)E2	1.1(0.0)E-1
eight_schools-eight_schools_centered	10	2.3(0.5)E-1	**7.7(1.2)E-1**	**1.4(0.0)E0**	1.1(0.0)E0	4.1(0.3)E0	1.5(0.2)E-1
eight_schools-eight_schools_noncentered	10	**8.0(0.5)E-1**	**1.2(0.0)E0**	6.2(0.0)E-1	1.2(0.0)E0	2.5(0.1)E0	1.2(0.0)E0
nes1972-nes	10	**2.4(0.0)E-1**	**1.1(0.0)E-1**	9.2(0.0)E-2	9.6(0.0)E-1	3.6(0.0)E-1	1.0(0.0)E-1
nes1976-nes	10	**2.5(0.0)E-1**	**1.1(0.0)E-1**	9.2(0.0)E-2	9.6(0.0)E-1	3.7(0.0)E-1	1.1(0.0)E-1
nes1980-nes	10	**3.1(0.1)E-1**	**1.1(0.0)E-1**	1.2(0.0)E-1	9.6(0.0)E-1	4.9(0.1)E-1	1.1(0.0)E-1
nes1984-nes	10	**2.4(0.0)E-1**	**1.2(0.0)E-1**	9.5(0.1)E-2	9.5(0.0)E-1	3.7(0.0)E-1	1.1(0.0)E-1
nes1988-nes	10	**2.5(0.1)E-1**	**1.1(0.0)E-1**	1.0(0.0)E-1	9.7(0.0)E-1	3.9(0.0)E-1	1.0(0.0)E-1
nes1992-nes	10	**2.3(0.0)E-1**	**1.1(0.0)E-1**	8.4(0.0)E-2	9.6(0.0)E-1	3.3(0.1)E-1	1.0(0.0)E-1
nes1996-nes	10	**2.6(0.0)E-1**	**1.1(0.0)E-1**	9.6(0.1)E-2	9.9(0.0)E-1	3.9(0.0)E-1	1.1(0.0)E-1
nes2000-nes	10	**4.2(0.1)E-1**	**1.2(0.0)E-1**	1.6(0.0)E-1	9.9(0.0)E-1	6.5(0.1)E-1	1.1(0.0)E-1
gp_pois_regr-gp_pois_regr	13	2.6(1.4)E-2	1.5(0.2)E0	**1.9(0.0)E-1**	**1.2(0.0)E0**	4.9(0.2)E-1	1.1(0.0)E-1
diamonds-diamonds	26	4.1(2.1)E-4	2.0(0.1)E0	**7.6(0.5)E-2**	**1.5(0.0)E0**	3.4(0.4)E-1	3.1(2.0)E-1
mcycle_gp-accel_gp	66	0	1.9(0.0)E0	**3.2(0.2)E-1**	**1.3(0.0)E0**	0	1.8(0.0)E0

Table 2: Benchmarking using `PosteriorDB`. Here, we compared a gradient-free version of RLMH to the gradient-free ARWMH, and also the gradient-based AMALA. Performance was measured using the *expected squared jump distance* (ESJD) and the *maximum mean discrepancy* (MMD) relative to the gold-standard, and $d = \dim(\mathcal{X})$. Results are based on an average of 10 replicates, with standard errors (in parentheses) reported. The best performing gradient-free method is highlighted in **bold**. Shaded rows indicate situations where RLMH outperformed AMALA for either ESJD or MMD.