



# Exploring the generalizability of the optimal 0.234 acceptance rate in random-walk metropolis and parallel tempering algorithms

Aidan Li, Liyan Wang, Tianye Dou, and Jeffrey S. Rosenthal

Department of Statistical Sciences, University of Toronto, Toronto, Canada

## ABSTRACT

For random-walk Metropolis (RWM) and parallel tempering (PT) algorithms, an asymptotic acceptance rate of around 0.234 is known to be optimal in certain high-dimensional limits. However, its practical relevance is uncertain due to restrictive derivation conditions. We synthesize previous theoretical advances in extending the 0.234 acceptance rate to more general settings, and demonstrate its applicability with a comprehensive empirical simulation study on examples examining how acceptance rates affect Expected Squared Jumping Distance (ESJD). Our experiments show the optimality of the 0.234 acceptance rate for RWM is surprisingly robust even in lower dimensions across various non-spherically symmetric proposal distributions, multimodal target distributions that may not have an i.i.d. product density, and curved Rosenbrock target distributions with nonlinear correlation structure. Parallel tempering experiments also show that the idealized 0.234 spacing of inverse temperatures may be approximately optimal for low dimensions and non i.i.d. product target densities, and that constructing an inverse temperature ladder with spacings given by a swap acceptance of 0.234 is a viable strategy.

## ARTICLE HISTORY

Received 16 October 2024  
Accepted 25 July 2025

## KEYWORDS

Acceptance rate; Markov chain Monte Carlo; Metropolis algorithm; optimal scaling; Parallel tempering

## 1. Introduction

Markov chain Monte Carlo (MCMC) methods, such as the random-walk Metropolis algorithm, are used to draw samples from complex high-dimensional target probability distributions. They enjoy strong theoretical asymptotic guarantees of accuracy, converging to the target distribution in stationarity (Brooks et al. 2011; Robert and Casella 2004). However, they may also be inefficient and prohibitively slow to provide a good approximation of the target distribution they are sampling (Rosenthal 1995). Therefore, the analysis of their running time is an important practical issue to consider.

Roberts, Gelman, and Gilks (1997) proved an important result for independently and identically distributed (i.i.d.) product target densities: in the state space high-dimensional limit  $d \rightarrow \infty$ , the asymptotic acceptance rate of a Metropolis algorithm is central to its efficiency. Under their assumptions, an asymptotic acceptance rate of approximately 0.234 maximizes the efficiency of a random-walk Metropolis (RWM) algorithm with a Gaussian proposal distribution starting at stationarity. However, their assumptions on the target and proposals are very restrictive. Many practical scenarios often involve finite and relatively small dimensions. Additionally, many probability distributions do not have densities of the i.i.d. product form, and for those that do, sampling a one-dimensional target is much simpler than using RWM. A burgeoning active research area is

dedicated to showing that this figure still applies beyond these assumptions, and much of this focuses on weakening the i.i.d. product assumption.

Many papers have shown that the optimal average acceptance rate of 0.234 for RWM holds in greater generality than the assumptions used in Roberts, Gelman, and Gilks (1997). For more general theoretical results, Sherlock (2013) replaces the restrictive i.i.d. product target and Gaussian proposal setting of Roberts, Gelman, and Gilks (1997) with a set of “shell,” relative variability, and eccentricity conditions on the target, eccentricity condition on the jump distance matrix, and shell condition on the proposal. This framework does not require a product form and is based on the geometric properties of the (log) target distribution in high dimensions. Yang, Roberts, and Rosenthal (2020) makes the 0.234 result applicable to targets with sparse dependency structures, such as those arising from Bayesian graphical models, using a set of sufficient conditions based on local dependencies and bounded asymptotic behavior of derivatives. We expand on these results in Sec. 2. The 0.234 acceptance rate in optimal scaling has also been shown to apply to the spacing and swapping between inverse temperatures in the parallel tempering MCMC method (Atchadé, Roberts, and Rosenthal 2011; Roberts and Rosenthal 2014); we describe this in greater detail in Sec. 4.

That being said, there are some cases in which 0.234 is not optimal, such as in the case of discontinuous targets (Neal, Roberts, and Yuen 2012) where the optimal acceptance rate is approximately 0.1353. Additionally, these results of optimality are still theoretical ideals, with required assumptions that could be further relaxed, and in the limit of  $d \rightarrow \infty$ , where the Markov chain converges to a diffusion process.

Other relaxations of the 0.234 assumptions include: non-Gaussian proposals (Neal and Roberts 2011), discrete hypercube target distributions (Roberts 1998), Gibbs random fields (Breyer and Roberts 2000), perturbations from product targets using pre-conditioned proposals (Beskos, Roberts, and Stuart 2009), independent target components with inhomogeneous scaling (Bédard 2007; Bédard and Rosenthal 2008; Roberts and Rosenthal 2001), partial updates where not all components are updated at once (Neal and Roberts 2006), elliptical symmetric unimodal target densities (Sherlock and Roberts 2009), and infinite-dimensional target distributions with non-trivial dependence structures (Mattingly, Pillai, and Stuart 2012).

These results are of great interest to practitioners who would like to understand how to tune their MCMC algorithms to maximize their efficiency. Previous simulation studies have been carried out on more general target distributions like products of independent, non-identical components (Bédard 2008; Roberts and Rosenthal 2001), and for the Markov modulated Poisson process (Sherlock, Fearnhead, and Roberts 2010), but there is still a lack of thorough experimentation and guidance for how applicable the 0.234 rule is on realistic target distributions that may have lower or higher dimensionality and/or multimodality. Our research examines how necessary these assumptions are for the theoretical result to remain relevant: where can we relax some assumptions in lower dimensions and still have the optimal acceptance rate of approximately 0.234, and where can we not do this? And in the end, how applicable is the 0.234 figure for the practitioner, who may want to use MCMC for complicated target distributions beyond the necessary assumptions in finite dimensions? We thoroughly dissect various aspects of the Metropolis algorithm and experiment with them, and in doing so, we show empirically where the theoretical ideal value can still align with more realistic scenarios.

We begin our paper with a description of the RWM and its optimal scaling framework, as well as formally stating sufficient theoretical conditions for the 0.234 rule that are most relevant to our work in Sec. 2. In Sec. 3, we describe RWM experiments in low dimensions that show where an acceptance rate of 0.234 can be optimal in the sense that it maximizes the Expected Squared Jumping Distance (ESJD) for various proposal and target densities which are not necessarily all i.i.d. product forms. Next, we describe the parallel tempering method and its own optimal scaling framework in Sec. 4, and show using experiments in Sec. 5 that

the 0.234 swap acceptance rate in parallel tempering may also be optimal in lower dimensions for the multivariate Gaussian target density and a Gaussian mixture target density which is not of an i.i.d. product form. Lastly, we discuss the implications of our findings in [Sec. 6](#).

## 2. Random-Walk Metropolis background

We first introduce the Metropolis algorithm and the Optimal Scaling framework. We then discuss measures to evaluate the efficiency of MCMC algorithms, such as the expected squared jumping distance, and explain how the asymptotic acceptance rate of a MCMC algorithm is tied to the algorithm's efficiency. Finally, we discuss some of the conditions for the 0.234 acceptance rate to be optimal that are most relevant to our work.

### 2.1. Metropolis algorithm

A Metropolis algorithm (Metropolis et al. 1953) constructs a Markov chain  $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \dots, \mathbf{x}^{(t)}$ . If the chain is constructed to be irreducible and aperiodic, it is guaranteed to have the target distribution  $\pi$  as its unique stationary distribution. This means that as the number of steps  $t$  becomes large, the distribution of  $\mathbf{x}^{(t)}$  converges to  $\pi$ . Therefore, after an initial burn-in period, the subsequent states of this single, long chain can be used as a sample from the target distribution  $\pi$ . When taking each step in the Markov chain, we generate a new state  $\mathbf{y}$  from a proposal distribution  $Q(\cdot|\mathbf{x}^{(t)})$  that is conditional on the current state  $\mathbf{x}^{(t)}$  and is symmetric, meaning  $Q(\mathbf{y}|\mathbf{x}^{(t)}) = Q(\mathbf{x}^{(t)}|\mathbf{y})$ . We accept this new state with probability  $\alpha(\mathbf{x}, \mathbf{y})$  given by  $\alpha = \min\left\{1, \frac{\pi(\mathbf{y})}{\pi(\mathbf{x})}\right\}$ , where, by a common abuse of notation, we let  $\pi(\cdot)$  also denote the probability density function of the target distribution  $\pi$ . If the proposed value is accepted, we set the next state  $\mathbf{x}^{(t+1)} = \mathbf{y}$ . Otherwise, we set  $\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)}$ .

### 2.2. Optimal scaling framework

Optimal Scaling (Roberts, Gelman, and Gilks 1997; Roberts and Rosenthal 2001) is one of the most successful frameworks for performing asymptotic analysis of high-dimensional MCMC methods, and provides mathematically-grounded guidance on how to best optimize MCMC performance by tuning the “scaling” parameter(s) of the proposal distribution. Typically, a multivariate Gaussian distribution centered at the current state is used as the proposal distribution. The scaling of the Gaussian proposal is then given by a scaling factor  $\sigma$ , and this is often used as a variance  $\sigma^2$  that is applied to the  $d \times d$  identity matrix  $I_d$ . So,  $\epsilon \sim \mathcal{N}(0, \sigma^2 I_d)$ . If the proposal's variance is too small, many proposed steps are accepted by the algorithm, but each step does not explore the state space much. Vice versa, if the proposal's variance is too large, too few proposed steps are accepted, leading to slow exploration of the state space since the algorithm stays at a state for too long. Therefore, the proposal's variance is crucial to the performance of the algorithm; the optimal scaling should balance the exploration of new areas in the state space with exploiting high-density areas of the target distribution.

We next discuss how to evaluate the efficiency of an MCMC algorithm for determining optimal scaling. A very popular measure of efficiency both theoretically (e.g. Sherlock (2013); Roberts and Rosenthal (2014); Yang, Roberts, and Rosenthal (2020)) and in practice (e.g. Pasarica and Gelman (2010); Sherlock et al. (2015)) is the Euclidean *Expected Squared Jumping Distance* (ESJD) metric which we use for our experiments. The expected squared jumping distance measures how far, in expectation, the MCMC chain moves in a single iteration. For the standard random-walk Metropolis algorithm, we define this as

$$\mathbb{E}[\|\mathbf{x}^{(t+1)} - \mathbf{x}^{(t)}\|^2] \approx \frac{1}{n-1} \sum_{i=1}^{n-1} \|\mathbf{x}^{(t+1)} - \mathbf{x}^{(i)}\|^2 \quad (1)$$

Where  $n$  is the total number of iterations of the algorithm. Maximizing the ESJD aligns with minimizing the first-order auto-correlation of the Markov chain and subsequently maximizes efficiency if each higher order auto-correlation is a monotonically increasing function of the first-order auto-correlation (Pasarica and Gelman 2010; Yang, Roberts, and Rosenthal 2020). It is worth mentioning that ESJD primarily measures local convergence performance. Because heavy-tailed targets disrupt the monotonic relationship between first-order and higher-order autocorrelations, achieving global convergence for heavy-tailed targets requires maximizing an adapted version of ESJD (Kamatani 2020). However, we do not consider heavy-tailed targets in this work, so the working definition of ESJD is sufficient.

There are other notions of efficiency of a Markov chain, but in the high-dimensional limit  $d \rightarrow \infty$ , if the moment conditions in [Assumption 2.1](#) hold, the chain converges to a diffusion process, and all efficiency measures are effectively equivalent (Roberts, Gelman, and Gilks 1997; Roberts and Rosenthal 2001). A key result of these referenced papers is that, given a Metropolis algorithm with a Gaussian proposal distribution  $\mathcal{N}(\mathbf{x}^{(t)}, \frac{\ell^2}{d} I_d)$  where  $\ell > 0$  is a fixed scaling constant and  $I_d$  is the identity matrix, maximizing the diffusion's speed measure  $h(\ell)$ , which is a function of the scaling constant  $\ell$ , yields the most efficient asymptotic diffusion. Furthermore, the speed measure has a clear relation to a much simpler quantity to tune for: the *asymptotic acceptance rate* of the proposed new states (moves) of the algorithm, defined as

$$a = \lim_{n \rightarrow \infty} \frac{\# \text{ accepted moves}}{n}.$$

Both the speed measure  $h(\ell)$  and asymptotic acceptance rate  $a(\ell)$  are functions of the scaling constant  $\ell$ . Given the restrictive conditions mentioned in [Sec. 2.3](#), the scaling constant  $\ell$  that maximizes the speed measure  $h(\ell)$  corresponds to an asymptotic acceptance rate  $a(\ell)$  of approximately 0.234. Hence, an asymptotic acceptance rate of approximately 0.234 should necessarily optimize a measure of efficiency such as the ESJD of the algorithm.

### 2.3. Theoretical assumptions for optimality of a 0.234 acceptance rate

Certain work has shown that under restrictive assumptions, a 0.234 acceptance rate is asymptotically optimal for random-walk Metropolis algorithms in certain high-dimensional limits. We briefly list the theoretical assumptions of these works that are most relevant to our simulations. We first describe the seminal result of Roberts, Gelman, and Gilks (1997) alluded to in the introduction in full.

**Assumption Set 2.1.** (Roberts, Gelman, and Gilks (1997)) Let  $\mathbf{x}$  be a random variable from the target density  $\pi$ . The target distribution has an i.i.d. product density  $\pi(\mathbf{x}) = \prod_{i=1}^d f(x_i)$ . The component density  $f : \mathbb{R} \rightarrow \mathbb{R}^+$  satisfies the following regularity conditions:

1.  $f$  is positive and twice continuously differentiable on its support.
2. The derivative of its logarithm,  $f'/f$ , is Lipschitz continuous.
3. The following moment conditions hold:

$$\mathbb{E}_f \left[ \left( \frac{f'(X)}{f(X)} \right)^8 \right] < \infty \quad \text{and} \quad \mathbb{E}_f \left[ \left( \frac{f''(X)}{f(X)} \right)^4 \right] < \infty \quad (2)$$

Under the conditions in [Assumption 2.1](#), Roberts, Gelman, and Gilks (1997) showed that for an RWM algorithm with a Gaussian proposal distribution starting in stationarity, when the whole

process is sped-up by a factor of  $d$ , the first component of the process converges to a limiting Langevin diffusion. Maximizing the speed of this limiting process, in turn, corresponds to tuning the algorithm to an asymptotic acceptance rate of approximately 0.234.

The Roberts, Gelman, and Gilks (1997) result works because an i.i.d. product target looks locally the same no matter which coordinate you observe. Then, the analysis of Sherlock (2013) shows that a similar homogeneity emerges for a much wider class of targets once the dimension is large.

**Assumption Set 2.2.** (Sherlock (2013)) Let  $\mathbf{x}$  be a random variable from the target density  $\pi$ ,  $M(\mathbf{x}) = \|\nabla \log \pi(\mathbf{x})\|$  denote the norm of the gradient of the log target,  $H(\mathbf{x}) = -\nabla^2 \log \pi(\mathbf{x})$  denote the negative Hessian matrix of the log target, and  $\Delta(\mathbf{x}, \mathbf{u}) = H(\mathbf{x} + \mathbf{u}) - H(\mathbf{x})$  denote the matrix of perturbations of the negative Hessian. The primary conditions are:

1. **Target weak shell.** There exist sequences  $\{M^{(d)}\}$  and  $\{H^{(d)}\}$ , as  $d \rightarrow \infty$  :

$$\frac{M^{(d)}(\mathbf{x}^{(d)})}{M^{(d)}} \xrightarrow{p} 1 \quad \text{and} \quad \frac{H^{(d)}(\mathbf{x}^{(d)})}{H^{(d)}} \xrightarrow{p} 1 \quad (3)$$

2. **Hessian relative variability.** The Hessian is locally stable, meaning it does not vary significantly on the scale of a typical proposal jump. Formally, for any fixed  $\mu > 0$  and  $\delta > 0$ , and for a standard Gaussian random vector  $\mathbf{z}^{(d)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$  independent of the target state  $\mathbf{x}^{(d)}$ , we require that as  $d \rightarrow \infty$ ,

$$\mathbb{P}_{\mathbf{x}, \mathbf{z}} \left( \frac{1}{H^{(d)}} \left| (\mathbf{z}^{(d)})^T \Delta \left( \mathbf{x}^{(d)}, t\mu \frac{M^{(d)}}{H^{(d)}} \mathbf{z}^{(d)} \right) \mathbf{z}^{(d)} \right| < \delta \quad \text{for all } t \in [0, 1] \right) \rightarrow 1. \quad (4)$$

3. **Target eccentricity.** The curvature is not disproportionately concentrated in any single direction. Denoting the eigenvalues of  $\mathbf{H}(\mathbf{x})$  by  $\beta_i(\mathbf{x})$  :

$$\frac{\max_i |\beta_i(\mathbf{x}^{(d)})|}{\sum_{j=1}^d \beta_j(\mathbf{x}^{(d)})} \xrightarrow{p} 0. \quad (5)$$

4. **Proposal shell.** There exists a sequence  $\{k_u^{(d)}\}$  such that the sequence of  $d$ -dimensional spherically symmetric proposals  $\{Q^{(d)}\}$  satisfies this as  $d \rightarrow \infty$  :

$$\frac{\|Q^{(d)}\|}{k_u^{(d)}} \xrightarrow{m.s.} 1. \quad (6)$$

5. **Jump distance matrix eccentricity.** If the ESJD is defined with respect to a positive definite symmetric matrix  $\mathbf{T}^{(d)}$  as  $\mathbb{E}[(\mathbf{Y}^{(d)})^T \mathbf{T}^{(d)} \mathbf{Y}^{(d)}]$ , then the eigenvalues of this jump distance matrix must be similarly well-behaved. Let  $\tau_i(d)$  be the eigenvalues of  $\mathbf{T}^{(d)}$ . When  $d \rightarrow \infty$  :

$$\frac{\max_i \tau_i(d)}{\sum_{j=1}^d \tau_j(d)} \rightarrow 0. \quad (7)$$

Put together, these geometric regularities recover the same result of the optimality of the 0.234 acceptance rate as Roberts, Gelman, and Gilks (1997) but with weaker assumptions on the target and proposal.

### 3. Random-Walk Metropolis simulations

In this section, we explore the applicability of the 0.234 acceptance rate figure for RWM algorithms using several practical RWM examples in lower dimensions. We begin with simple examples where the target densities meet all the key conditions in Roberts, Gelman, and Gilks (1997) (provided as [Assumption 2.1](#)) required for the 0.234 acceptance rate to be optimal. Next, we investigate examples that do not use Gaussian proposal distributions, and then we explore other target densities that may lack these assumptions, such as smoothness or the i.i.d. product form. We estimate the Expected Squared Jumping Distance (ESJD) of the algorithm as a metric of the algorithm's efficiency as mentioned in [Sec. 2.2](#), and aim to see under what conditions an acceptance rate of approximately 0.234 maximizes the ESJD for various target and proposal distributions in lower dimensions.

Our experimental procedure is as follows: we run RWM simulations using (except for [Sec. 3.2](#)) a Gaussian proposal distribution

$$Q(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}|\mathbf{x}, \sigma^2 I_d) \quad (8)$$

Where  $\sigma$  is a proposal scaling factor, or  $\sigma^2$  can be thought of as a variance, and  $I_d$  is the  $d$ -dimensional identity matrix. We run 40 different RWM simulations where each simulation has a different scaling factor, and each simulation has a burn-in of at least 1000 steps before taking at least 200,000 steps. It is important to note that functions of acceptance rates are somewhat flat around the optimal values and that the values and figures reported are still prone to random error inherent in a Monte Carlo simulation. Therefore, we repeat this procedure over at least 20 computer seeds and average the results to reduce random error. This procedure is applied in each subsection and specific simulation details may be mentioned in a subsection if the experiment used more than these minimum values.

#### 3.1. Simpler examples: Gamma and Beta i.i.d. targets

We begin with simpler examples demonstrating that the 0.234 acceptance rate is still roughly optimal for lower dimensions with any i.i.d. product target distribution and a Gaussian proposal. As mentioned in [Sec. 2](#), Roberts, Gelman, and Gilks (1997) proved the 0.234 optimal acceptance rate for i.i.d. product target distributions with a standard Gaussian proposal distribution in the infinite-dimensional limit, and Roberts and Rosenthal (2001) and Bédard (2008) extended this with MCMC simulations showing that with a i.i.d. product target distribution and Gaussian proposal, the 0.234 acceptance rate seems approximately optimal for dimensions as low as 10. Our experiment provides further evidence for this being the case, and we extend this with simulations for  $d = 2, 5$ . We present two i.i.d. target densities  $\pi_1, \pi_2$  with different single-dimension component densities  $f_1, f_2$ :

$$\pi_1(\mathbf{x}) = \prod_{i=1}^n f_1(x_i), \quad f_1(x) = \text{Gamma}(x|3, 2), \quad (9)$$

$$\pi_2(\mathbf{x}) = \prod_{i=1}^n f_2(x_i), \quad f_2(x) = \text{Beta}(x|3, 2) \quad (10)$$

The first example sets  $f_1$  with shape parameter 3 and scale parameter 2, and the second example sets  $f_2$  with shape parameters 3 and 2. We use the multivariate Gaussian proposal distribution defined in (8) in both examples. We run this experiment in dimensions  $d \in \{2, 5, 10, 30, 50, 100\}$ .

The resulting ESJD-maximizing acceptance rates are summarized in [Table 1](#) and the i.i.d. Beta target has its trends of ESJD versus acceptance rate with different dimensions displayed in [Figure 1](#). For the Gamma target, the optimal acceptance rate drops from 0.304 at  $d = 2$  to around 0.21 once  $d \geq 10$ ; for the Beta target they fall from 0.390 to around 0.22 over the same range. [Figure 1](#) confirms that the ESJD curves have a moving plateau toward a center slightly below the classical 0.234

**Table 1.** Empirical optimal RWM acceptance rates (i.e. maximizing ESJD) for i.i.d. product targets with a Gaussian proposal.

Target Density	$d = 2$	$d = 5$	$d = 10$	$d = 30$	$d = 50$	$d = 100$
$\pi_1$ : i.i.d. Gamma(3, 2)	0.3036	0.2378	0.2199	0.2101	0.2141	0.2140
$\pi_2$ : i.i.d. Beta(3, 2)	0.3903	0.2937	0.2561	0.2319	0.2248	0.2159

value as the dimension grows. Our simulation provides further evidence that the optimal acceptance rate for maximizing ESJD is approximately 0.234 even in low dimensions.

Since most ESJD versus acceptance rate plots look fairly similar and have a similar progression as dimension changes, the full set of curves for ESJD versus acceptance rate plots for other sub-sections are in [Appendix A](#), including the results for the i.i.d. Gamma in [Figure A1](#). As an aside, in all dimensions, there is a relatively broad plateau where the acceptance rate is approximately optimal; thus, the coarse 40-point grid pinpoints the optimum to approximately  $\pm 0.01$ .

### 3.2. Non-Gaussian proposal densities

Next, we investigate the efficacy of different proposal distributions for the Metropolis algorithm. One of the conditions assumed by the 0.234 theorem of Roberts, Gelman, and Gilks (1997) is the use of a multivariate Gaussian proposal distribution  $Q(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}|\mathbf{x}, \sigma^2 I_d)$ . The additional results in Roberts and Rosenthal (2001) depend on the assumptions of light-tailed proposals. In contrast, Jarner and Roberts (2002) extend this analysis by examining the behavior of the algorithms with heavy-tailed proposals. They find that, unlike the diffusion behavior seen with light-tailed proposals, heavy-tailed proposals lead to a different dynamic characterized by abrupt movements followed by periods of inactivity. Furthermore, Neal and Roberts (2011) investigated optimal scaling results for heavy-tailed proposal distributions such as the Cauchy distribution, and spherical proposal distributions of fixed radius. In their RWM simulation study with three different continuous i.i.d. product target densities, both the Gaussian and Cauchy proposals had an estimated asymptotically optimal acceptance rate of approximately 0.234 in  $d = 100$ , and a spherical proposal with a fixed radius of 2.38 achieved an optimal acceptance rate of about 0.234 in  $d = 2$  and  $d \geq 10$ . Finally, Sherlock (2013) proved that spherically symmetric proposal distributions following a geometric “shell” condition from [Assumption 2.2 \(6\)](#) should yield an optimal acceptance rate of approximately 0.234 when sampling from a suitable target.

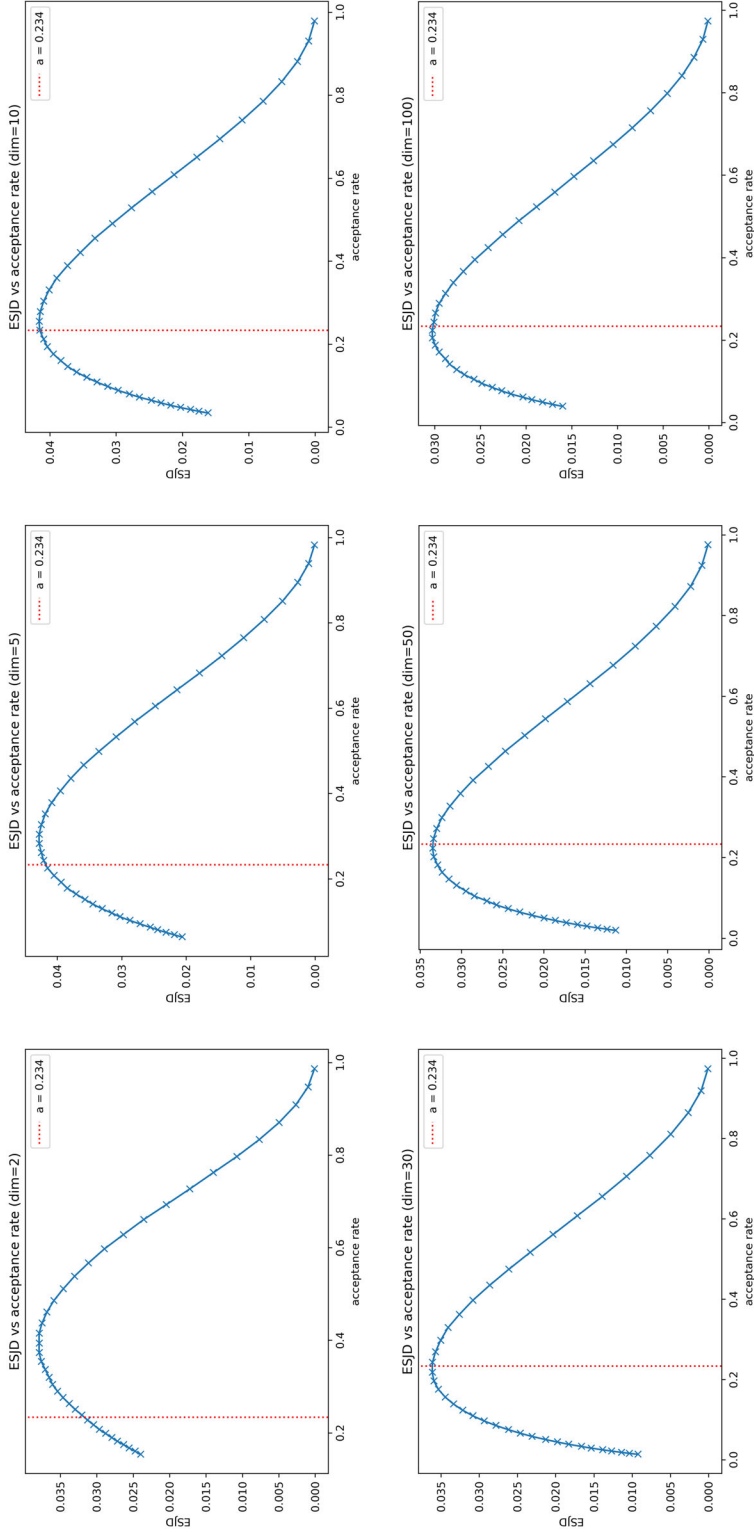
We experiment with two different proposal distributions in  $d \in \{2, 5, 10, 20, 50, 100\}$  and set the target density  $\pi_3$  to a standard multivariate Gaussian:

$$\pi_3(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\mathbf{0}, I_d)$$

$$Q_L(\mathbf{y}|\mathbf{x}) \propto \prod_{i=1}^d \exp\left(-\frac{|y_i - x_i|}{\sigma}\right), \quad Q_U(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^d \frac{\mathbb{I}\left[|y_i - x_i| \leq \frac{b}{2}\right]}{b} \quad (11)$$

The first proposal distribution we consider is the multivariate Laplace distribution, which may also be referred to as the double exponential distribution. We set the location parameter  $\mu = \mathbf{x}$  and scale parameter  $\sigma$  to be chosen to influence the acceptance rate of the algorithm. We also consider an example with the uniform proposal distribution within the closed interval  $\left[x_i - \frac{b}{2}, x_i + \frac{b}{2}\right]$  for each component  $i$ , where  $b > 0$  is the interval length.

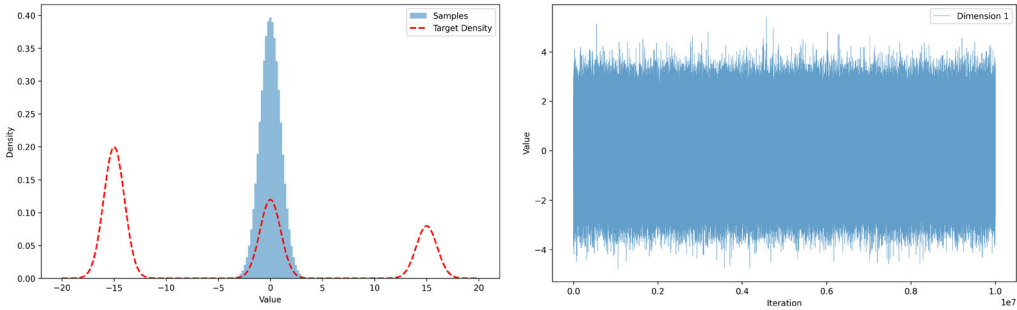
As observed in [Table 2](#) and the corresponding [Figures A2, A3](#) in [Appendix A](#), the optimal acceptance rates for both the Laplace and Uniform proposals exhibit a clear trend. In low dimensions, the optimal rate is significantly higher than 0.234, but converges toward 0.234 from above as the dimension increases. For the Uniform proposal, the optimal rate is already approximately 0.239 at  $d = 10$ , and for both proposals, the rates are close to the high-dimensional limit for  $d \geq 20$ .



**Figure 1.** ESJD vs. acceptance rate for the i.i.d. Beta(3, 2) target distribution  $\pi_2$  under RWM with a Gaussian proposal in dimensions  $d \in \{2, 5, 10, 30, 50, 100\}$  from top-left to bottom-right. Red dotted line indicates an acceptance rate of 0.234.

**Table 2.** Empirical optimal RWM acceptance rates across dimensions for different proposal distributions targeting a standard multivariate Gaussian.

Proposal	$d = 2$	$d = 5$	$d = 10$	$d = 20$	$d = 50$	$d = 100$
$Q_L$ : Laplace	0.3780	0.3036	0.2841	0.2570	0.2429	0.2377
$Q_U$ : Uniform	0.3194	0.2516	0.2391	0.2368	0.2365	0.2316

**Figure 2.** Histogram and traceplot of 10 million samples drawn from a multimodal single-dimensional target density. The true density values are traced with the dotted lines.

These empirical findings provide strong validation for the generalized theory of optimal scaling. While the foundational result of Roberts, Gelman, and Gilks (1997) was derived for Gaussian proposals, the more general criteria from Sherlock (2013), summarized in [Assumption 2.2](#), extend this theory. The key requirement on the proposal is the shell condition (6) that requires the norm of the proposal vector to concentrate around a constant in high dimensions. Interestingly, although both the multivariate Laplace and Uniform distributions used here are not spherically symmetric as required by (6), they still satisfy the spirit of the shell condition mentioned in the previous sentence. If these more relaxed conditions still satisfy the necessary requirements for the theory to work, the theory predicts that the 0.234 acceptance rate should indeed be asymptotically optimal when using these proposals on a suitable target. Our results confirm this prediction, and characterize the finite-dimensional behavior, demonstrating that convergence to the 0.234 optimum occurs from above for these proposals. The lack of spherical symmetry in our proposals also suggests that weaker conditions on the proposal than spherical symmetry may be sufficient for the 0.234 rate to be optimal.

### 3.3. Multimodal target densities

In this subsection, we investigate target distributions with multimodal densities. Many real-world probability and data distributions are multimodal, and standard RWM algorithms may struggle to explore all the modes of a multimodal distribution in a reasonable amount of time particularly when the modes are far apart. Additionally, multimodal densities may not necessarily fall under the i.i.d. product umbrella of target densities, and even if they do, it is debatable how relevant the 0.234 rule is in these cases. For example, a RWM algorithm may achieve a 0.234 acceptance rate in its run over many iterations and yet completely fail to escape a mode to explore other modes. We illustrate this with [Figure 2](#), where we attempt to draw 10 million samples from a multimodal distribution with three modes but fail to escape the central mode.

When we have widely-separated modes, a RWM with the Gaussian proposal distribution struggles to reach the other modes in finite time regardless of the proposal scaling and the dimension of the target distribution. Therefore, for these multimodal target density experiments, we deliberately set the modes to be close enough to each other so that a RWM still mixes between the modes reasonably often. We list out the various target densities before detailing the experimental results.

### 3.3.1. The “Rough Carpet” target density

First, we examine a simpler i.i.d. product density. Our first multimodal distribution has the form

$$\pi_4(\mathbf{x}) = \prod_{i=1}^d f_4(x_i), \quad f_4(x) = 0.5 \mathcal{N}(x|\mu_1, 1) + 0.3 \mathcal{N}(x|\mu_2, 1) + 0.2 \mathcal{N}(x|\mu_3, 1) \quad (12)$$

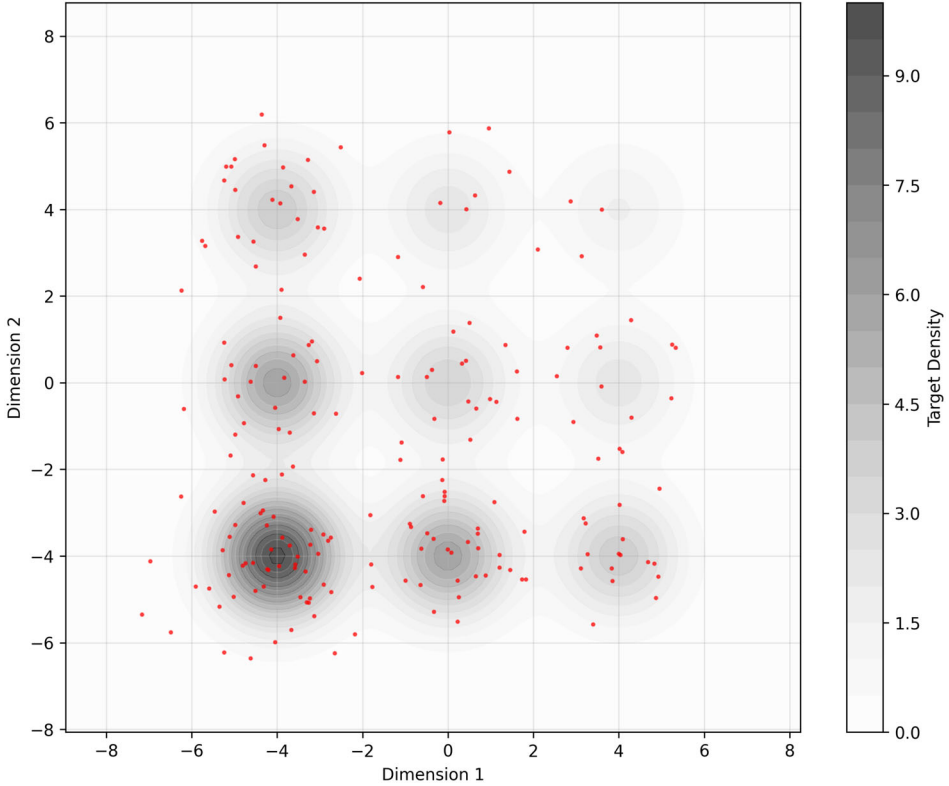
Where the single component density  $f_4$  is a one-dimensional density with three modes at  $(\mu_1, \mu_2, \mu_3) = (-5, 0, 5)$  and  $\mathcal{N}(x|\mu_i, 1)$  indicates the density of the univariate Gaussian distribution at point  $x$  with mean  $\mu_i$  and variance 1. This forms a “rough carpet”-like appearance for the target distribution when we take the product of this single-component density over many dimensions (see the visualization of the first two components in Figure 3 for an example). Since we have  $3^d$  modes, the density values at each mode may not be very high. Yet, since this is a i.i.d. product form, the theoretical result of 0.234 being the optimal acceptance rate should also apply.

### 3.3.2. The “Rough Carpet” target density: inhomogeneously scaled components

Next, we examine a multimodal distribution with a density of the form

$$\pi_5(\mathbf{x}) = \prod_{i=1}^d C_i f_5(C_i x_i), \quad f_5(x) = 0.5 \mathcal{N}(x|\mu_1, 1) + 0.3 \mathcal{N}(x|\mu_2, 1) + 0.2 \mathcal{N}(x|\mu_3, 1). \quad (13)$$

Note that  $f_5 = f_4$ , so this density  $\pi_5$  is a more general form of  $\pi_4$  by including inhomogeneous component-wise scaling factors  $C_i$ , where  $C_i > 0$  and  $\mathbb{E}(C_i) = 1$  and  $\text{Var}(C_i) < \infty$ . This is a special case of a class of inhomogeneously-scaled i.i.d. product target densities discussed by Roberts



**Figure 3.** Density of the first two dimensions of the “rough carpet” target distribution with MCMC samples from one RWM simulation plotted in red. One out of every 200 MCMC samples drawn are plotted.

and Rosenthal (2001), which proved the extension of the 0.234 asymptotically optimal acceptance rate to this class. We independently sample the component scaling factors  $C_i \sim \text{Uniform}[0.02, 1.98]$ .

### 3.3.3. A tale of three mixtures

The above “rough carpet” distributions are multimodal, but notice that they still follow the general product form  $\pi(\mathbf{x}) = \prod_{i=1}^d f(x_i)$ . Here, we examine a distribution that does not have this product assumption at all; instead, we have a weighted sum of densities as our target density. We examine a mixture of three Gaussians; a multimodal distribution with just three modes regardless of the number of dimensions. The density is

$$\pi_6(\mathbf{x}) = w_1 \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_1, \Sigma_1) + w_2 \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_2, \Sigma_2) + w_3 \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_3, \Sigma_3) \quad (14)$$

Where we have non-negative scalar weights  $w_i \geq 0$  such that  $w_1 + w_2 + w_3 = 1$ , and  $\mathcal{N}$  is the density function of a multivariate Gaussian with mean  $\boldsymbol{\mu}_i \in \mathbb{R}^d$  and covariance matrix  $\Sigma_i \in \mathbb{R}^{d \times d}$ .

For these experiments, we set the weights  $w_1 = w_2 = w_3 = \frac{1}{3}$ , the covariances  $\Sigma_1 = \Sigma_2 = \Sigma_3 = I_d$ , and set the means  $\boldsymbol{\mu}_1 = (\epsilon, 0, 0, \dots, 0)$ ,  $\boldsymbol{\mu}_2 = (0, 0, \dots, 0)$ ,  $\boldsymbol{\mu}_3 = (-\epsilon, 0, 0, \dots, 0)$  for some constant  $\epsilon \in \mathbb{R}$ , i.e. just varying the first component of the means and setting the other components of the means to 0. As mentioned (refer to Figure 2), we choose the constant  $\epsilon = 5.0$  in the first dimension after observing *via* traceplots that jumping between modes still happens frequently enough. We refer the reader to Figure 4 for an example histogram and traceplot to illustrate how the Markov chain explores the three modes.

### 3.3.4. A tale of three mixtures: inhomogeneous scaling factors

We repeat our experiment with inhomogeneous scaling factors applied to each Gaussian mode. Concretely, we instantiate our target density as

$$\pi_7(\mathbf{x}) = \sum_{k=1}^3 w_k \cdot \left( \prod_{j=1}^d C_j \right) \cdot \mathcal{N}(\mathbf{x} \odot \mathbf{C} | \mu_k, I_d) \quad (15)$$

Where  $C_i \sim \text{Uniform}[0.2, 1.8]$  and  $\mathbf{x} \odot \mathbf{C}$  denotes the Hadamard (element-wise) product  $(x_1 C_1, \dots, x_d C_d)$ . This allows each Gaussian mode to follow the functional form defined in Roberts and Rosenthal (2001) Sec. 6; each mode has its own density of the functional form  $\pi(\mathbf{x}) = \prod_{i=1}^d C_i f(C_i x_i)$ , where  $\{C_i\}$  are i.i.d. with  $\mathbb{E}(C_i) = 1$  and  $\mathbb{E}(C_i^2) < \infty$ .

### 3.3.5. Multimodal experiments and results

We conduct our experiments in  $d \in \{20, 30, 50\}$  using the default experimental procedure described. We summarize our results in Table 3, and refer the reader to Appendix A Figures A4–A7 for plots of the trend of ESJD with acceptance rate.

The results for our multimodal experiments provide strong empirical support for the generalizability of the 0.234 acceptance rate heuristic. We observe that for all four target distributions, the optimal acceptance rate converges toward 0.234 as the dimension increases, even for those that lack an i.i.d. product structure. Their optimal acceptance rates are already very close to 0.234 even in low dimensions ( $d = 10$ ). Theoretically, the 0.234 value is likely to be justified by the general criteria of Sherlock (2013) that are stated in Assumption 2.2. The three mixture targets  $(\pi_6, \pi_7)$ , while not product densities, are constructed from isotropic Gaussians. In high dimensions, they form a well-behaved geometric shell with low eccentricity, thus satisfying the target conditions for the 0.234 rule to apply. The rough carpet targets  $\pi_4, \pi_5$  are i.i.d. products of a more complex multimodal density. The component-wise multimodality creates a more intricate target landscape, which may require higher dimensions before the geometric properties average

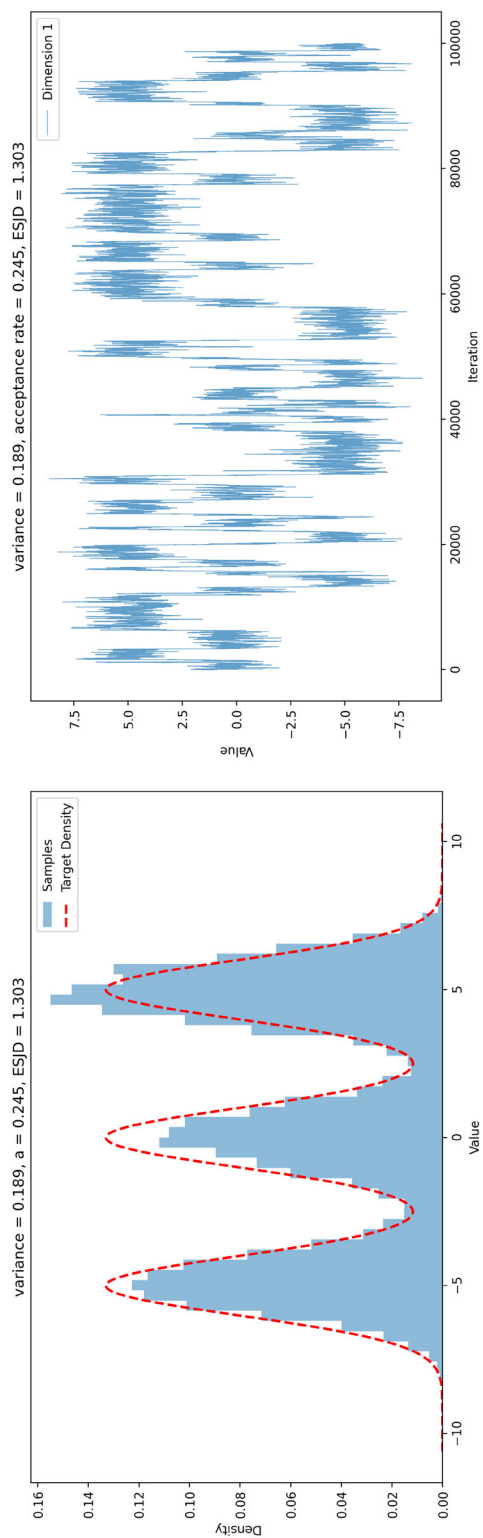


Figure 4. A histogram and traceplot of the first component from one of the simulations in the three mixture target distribution experiment after 100,000 iterations.

**Table 3.** Empirical optimal RWM acceptance rates for multimodal target distributions.

Target	$d = 20$	$d = 30$	$d = 50$
$\pi_4$ : Rough Carpet (homogeneous)	0.2133	0.2202	0.2291
$\pi_5$ : Rough Carpet (inhomogeneous)	0.2009	0.2174	0.2351
$\pi_6$ : 3-Mixture (homogeneous)	0.2489	0.2454	0.2394
$\pi_7$ : 3-Mixture (inhomogeneous)	0.2360	0.2334	0.2371

out to satisfy the shell and stability conditions as effectively; this may explain why their optimal rates approach 0.234 from below.

There is an interesting note that for both families of targets, introducing inhomogeneous scaling consistently pushes the optimal acceptance rate lower in smaller dimensions. This could be explained by the Target Eccentricity condition of Sherlock (2013) (mentioned in Assumption 2.2 (5)). The inhomogeneous scaling factors introduce anisotropy into the target's curvature, creating directions with much higher curvature than others. With a standard isotropic proposal such as the multivariate Gaussian, the optimal proposal scale must shrink to avoid constant rejections in these high-curvature directions. This makes the sampler inefficient in the less-curved directions, degrading the algorithm's overall performance. This less efficient dynamic alters the tradeoff between proposal size and acceptance probability, shifting the peak of the ESJD curve to the left. Our results show that for such low-dimensional inhomogeneously scaled targets, the optimal balance is achieved at a lower acceptance rate than the 0.234 ideal. Then, as  $d$  grows, the law of large numbers mitigates the effect of any single scaling factor, the target becomes effectively less eccentric, and the optimal acceptance rate recovers to 0.234.

### 3.4. Rosenbrock target densities

The multimodal experiments in Sec. 3.3 demonstrated the effectiveness of the 0.234 acceptance rate for multimodal distributions, where the main challenge is mode isolation. In practice, however, many posterior landscapes are not only multi-peaked but also have nonlinear correlation structure, providing challenges for local proposals even when there is only a single global mode. We examine this setting by sampling three different extensions of the two-dimensional Rosenbrock target density to  $d$ -dimensions (Dixon and Mills 1994; Goodman and Weare 2010; Pagani, Wiegand, and Nadarajah 2022), each of which have different curvature and therefore different difficulty from a MCMC perspective.

Rosenbrock (1960) originally introduced the Rosenbrock function as an optimization problem: a non-convex function which has a global minimum inside a long, narrow parabolic canyon. The Rosenbrock function can then be converted into a probability density that preserves these properties. Because the local correlation structure changes rapidly along the ridge, step sizes that work near the mode are either rejected or become inefficient elsewhere. Consequently, Rosenbrock targets have become common benchmarks to test MCMC algorithms (Haario, Saksman, and Tamminen 2001; Huijser, Goodman, and Brewer 2022; Hunt-Smith et al. 2024; Militzer 2025). Including these targets in our empirical study extends our findings to targets with curved correlation structure.

We first describe the unnormalized Rosenbrock target densities, then describe our experimental procedure and results. We use the Rosenbrock kernels described by Pagani, Wiegand, and Nadarajah (2022).

#### 3.4.1. Full Rosenbrock

The  $d$ -dimensional Full Rosenbrock distribution (Goodman and Weare 2010) has the following kernel for  $\mathbf{x} \in \mathbb{R}^d$ :

$$\pi_8(\mathbf{x}) \propto \exp \left\{ - \sum_{i=1}^{d-1} \left[ 100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2 \right] / 20 \right\} \quad (16)$$

The original 2-dimensional Rosenbrock resembles a long banana-shaped valley whose floor follows the parabola  $x_2 = x_1^2$ . As stated in Pagani, Wiegand, and Nadarajah (2022), as  $d$  increases, the long, narrow ridge that makes sampling challenging becomes more concentrated around the mode, decreasing the problem's difficulty. However, the variance of  $x_d$  increases steeply as each new random variable is directly dependent on the squared value of the previous variable.

### 3.4.2. Even Rosenbrock

The  $d$ -dimensional Even Rosenbrock distribution extended from the optimization function in Dixon and Mills (1994) has the following kernel for  $\mathbf{x}, \boldsymbol{\mu} \in \mathbb{R}^d$ , where  $d$  must be even:

$$\pi_9(\mathbf{x}) \propto \exp \left\{ -\sum_{i=1}^{d/2} \left[ (x_{2i-1} - \mu_{2i-1})^2 + 100(x_{2i} - x_{2i-1}^2)^2 \right] / 20 \right\} \quad (17)$$

The Even Rosenbrock partitions the state into  $\frac{d}{2}$  independent 2-dimensional Rosenbrock blocks. Because the blocks do not interact, the global target is the product of smaller bananas; the joint density still exhibits the same curved valleys within each block but lacks the long-range, serial dependence of the Full Rosenbrock. Consequently, we expect higher ESJD (and hence slightly lower optimal acceptance rates) than in the fully coupled case.

Our experiments only use a single scaling factor applied to the whole covariance matrix of the proposal, but it could be useful for a practitioner to note that it is possible to locally tune the proposal for each 2-dimensional Rosenbrock block to make the RWM algorithm more efficient on this distribution. For example, when using a Gaussian proposal, the covariance matrix could be set as a block diagonal matrix where each  $2 \times 2$  block on the diagonal of the Gaussian covariance matrix is tailored to and achieves optimality on its own 2-dimensional Rosenbrock.

### 3.4.3. Hybrid Rosenbrock

The  $d$ -dimensional Hybrid Rosenbrock distribution introduced by Pagani, Wiegand, and Nadarajah (2022) has the following kernel for  $\mathbf{x} \in \mathbb{R}^d$ , where  $\mu, x_{j,i} \in \mathbb{R}; a, b_{j,i} \in \mathbb{R}^+(\forall j, i)$ :

$$\pi_{10}(\mathbf{x}) \propto \exp \left\{ -a(x_1 - \mu)^2 - \sum_{j=1}^{n_2} \sum_{i=2}^{n_1} b_{j,i} (x_{j,i} - x_{j,i-1}^2)^2 \right\} \quad (18)$$

The integers  $n_1$  and  $n_2$  describe a grid of blocks: there are  $n_2$  vertical stacks, each containing  $n_1 - 1$  Rosenbrock-type links and a shared root coordinate  $x_1$ . The total dimension is therefore  $d = (n_1 - 1)n_2 + 1$ . Within every column  $j$ , the curvature coefficients  $b_{j,i}$  can vary with depth  $i$ , producing steeper or flatter valleys as we move away from the root. Horizontally, however, columns are independent, so the Hybrid form interpolates between the serial dependence of the Full Rosenbrock (one long chain:  $n_2 = 1$ ) and the independent blocks of the Even Rosenbrock ( $n_1 = 3, n_2 = d/2$ ). This allows the difficulty of the distribution to be modified by setting  $n_1, n_2$ ; larger  $n_1$  increases non-linearity within a column, while larger  $n_2$  raises overall dimension without adding extra curvature.

### 3.4.4. Simulations and results

For our Rosenbrock simulations, we found that our initial settings of 200,000 Markov chain steps and 20 seeds were insufficient to produce meaningful trends, so we ran more simulations for more steps with the Rosenbrock targets. In this section, we continue using a Gaussian proposal with 40 different variance values. However, we run each simulation for 1,000,000 steps and we repeated each experiment with 30 different seeds instead of the usual 20. For the Full Rosenbrock, we select  $d \in \{5, 10, 20, 30\}$ . For the Even Rosenbrock, we select  $d \in \{4, 10, 20, 30\}$ .

**Table 4.** Empirical optimal RWM acceptance rates for three different  $d$ -dimensional Rosenbrock target distributions.

Target	$d = 4$ or $5$	$d = 9$ or $10$	$d = 19$ or $20$	$d = 29$ or $30$
$\pi_8$ : Full Rosenbrock	0.2047	0.2359	0.2384	0.2337
$\pi_9$ : Even Rosenbrock	0.0646	0.1430	0.2003	0.2107
$\pi_{10}$ : Hybrid Rosenbrock	0.1394	0.2272	0.2349	0.2340

For the Hybrid Rosenbrock, we select  $(n_1, n_2) \in \{(3, 2), (5, 2), (7, 3), (8, 4)\}$  which gives total dimension  $d \in \{5, 9, 19, 29\}$  respectively.

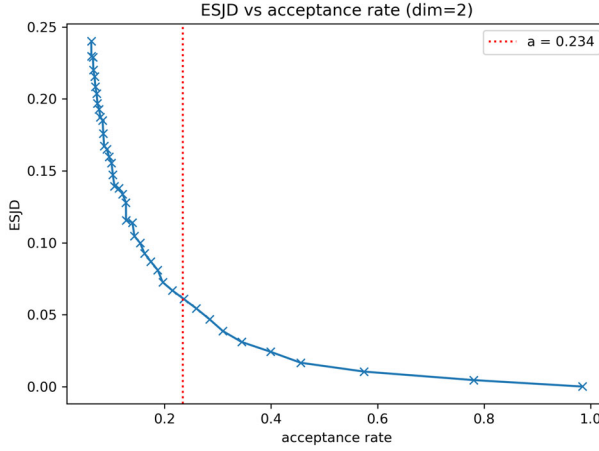
We also provide a plot of the ESJD versus acceptance rate trend for the 2-dimensional Rosenbrock. Note that for  $d = 2$ , each of the three Rosenbrocks reduce to the standard Rosenbrock 2-dimensional density, assuming  $\mu_1 = 1$  for the Even Rosenbrock and  $n_1 = 2, n_2 = 1, a = \frac{1}{20}, b = \frac{100}{20}, \mu = 1$  for the Hybrid Rosenbrock. All other plots of ESJD versus acceptance rate trends for these distributions are in [Appendix A Figures A8–A10](#) for plots of the trend of ESJD with acceptance rate.

Our experiments with the Rosenbrock family of targets, presented in [Table 4](#), demonstrate that the 0.234 heuristic remains effective even for distributions with non-linear dependencies and curvature. For the Full Rosenbrock ( $\pi_8$ ) and Hybrid Rosenbrock ( $\pi_{10}$ ) targets, the optimal acceptance rate robustly converges to approximately 0.234 as the dimension increases. Although these targets are not i.i.d. and possess a highly anisotropic banana-shaped geometry, they should satisfy the general criteria from [Sherlock \(2013\)](#) (stated as [Assumption 2.2](#)) in the high-dimensional limit. As  $d$  grows, the complex local correlations become less influential relative to the global structure, allowing the target’s geometric properties to stabilize in the manner required by [Assumption 2.2](#). On the other hand, the Even Rosenbrock ( $\pi_9$ ) has a significantly lower optimal acceptance rate in low dimensions than the other Rosenbrocks, and the optimal acceptance rate increases slowly toward 0.234 with increasing dimension. This seems reasonable and the analysis of [Beskos et al. \(2018\)](#) is useful here.

The Even Rosenbrock is a product of independent 2-dimensional Rosenbrock blocks, and for a small number of blocks, the standard high-dimensional scaling theory is less applicable. We elaborate on this in the next paragraph about the 2-dimensional Rosenbrock, but the key relevant message from [Beskos et al. \(2018\)](#) is that in low dimensions, large proposed jumps and an acceptance rate closer to zero are most efficient, explaining the low optimal acceptance rate. As the dimension  $d$  and thus the number of independent blocks grow, the dynamics transition from a Markov jump process toward a limiting diffusion. While the theory for such targets suggests a position-dependent diffusion speed and thus a “local” 0.234 rule ([Beskos et al. 2018](#), see point (iv), p. 2968), the growing number of independent blocks and therefore the growing number of “large” components may induce a law of large numbers effect where position-dependent variations in diffusion speed average out across the many blocks. This would result in an approximately constant global diffusion speed and therefore a global 0.234 acceptance rate being approximately optimal in high dimensions for the Even Rosenbrock.

The result for the 2-dimensional Rosenbrock target, shown in [Figure 5](#), provides empirical evidence for the low-dimensional sampling phenomena analyzed by [Beskos et al. \(2018\)](#). Unlike the higher-dimensional cases, the ESJD does not exhibit a clear peak around a non-zero acceptance rate. Instead, it appears to increase monotonically as the acceptance rate approaches zero.

As theoretically established by [Beskos et al. \(2018\)](#), the optimal sampling strategy for this low-dimensional ridged target is not a diffusion process. They state that efficiency is maximized by proposing extremely large jumps using a large proposal scaling  $\sigma$  and allowing the acceptance rate to go to zero. A heuristic explanation for this is that in this case, the squared distance of a proposal increases proportionally to  $\sigma^2$  but the fraction of accepted proposals decreases proportionally to  $\frac{1}{\sigma}$ , and therefore ESJD increases proportionally to  $\sigma$  until the proposal scale becomes comparable to the characteristic scale of the target distribution itself, at which point the



**Figure 5.** ESJD vs. acceptance rate for the 2-dimensional Rosenbrock target distribution  $\pi_8$  under RWM with a Gaussian proposal. Red dotted line indicates an acceptance rate of 0.234.

**Table 5.** Empirical optimal RWM acceptance rates for the continuous hypercube target.

Target	$d = 2$	$d = 5$	$d = 10$	$d = 20$	$d = 30$	$d = 50$	$d = 100$
$\pi_{11}$ : Hypercube $[0,1]^d$	0.4316	0.2767	0.2027	0.1670	0.1525	0.1439	0.1423

acceptance probability begins to decay more rapidly. Taking the limit, as the thickness (scale  $\varepsilon$  in Beskos et al. (2018)) of the support of the target distribution goes to zero, the algorithm's dynamics converge to a continuous-time Markov jump process rather than a diffusion. Our observation for the 2-dimensional Rosenbrock target, and by extension the low-dimensional Even Rosenbrock target (e.g. for  $d = 4$ , which consists of two independent 2-dimensional blocks), is an instance of this theoretically-predicted behavior.

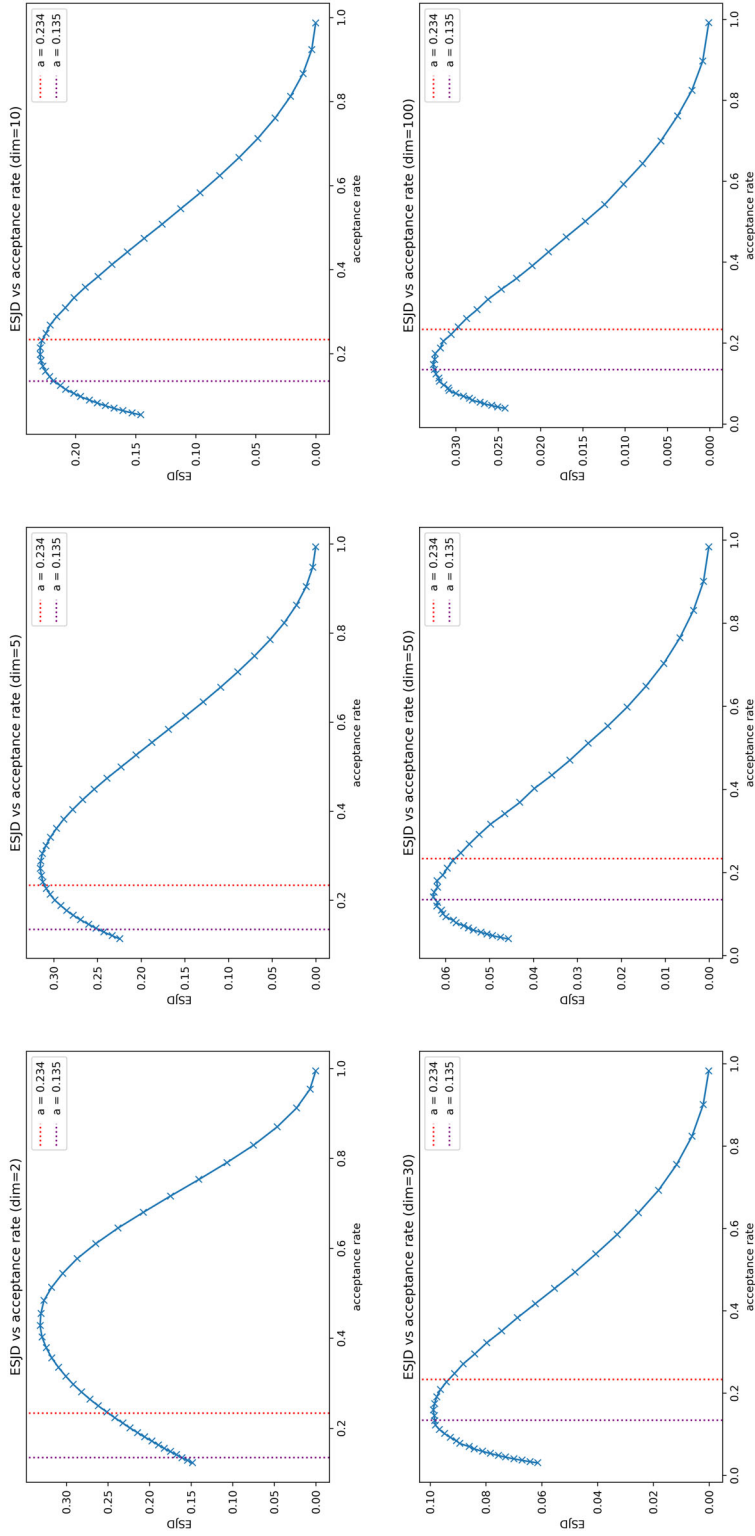
### 3.5. Non-smooth target densities: continuous hypercube

Now, consider a RWM using a Gaussian proposal and a target density still of the product form  $\pi(\mathbf{x}) = \prod_{i=1}^d f(x_i)$ , but where the individual component density  $f$  is Uniform $[a, b]$ .

$$\pi_{11}(\mathbf{x}) = \prod_{i=1}^d f_{11}(x_i), \quad f_{11}(x) = \frac{\mathbb{I}[a \leq x \leq b]}{b - a} \quad (19)$$

Unlike the previous examples, the component density  $f$  and its logarithm are not continuously differentiable on  $\mathbb{R}$  due to the discontinuities at the boundaries  $x_i = a$  and  $x_i = b$ . Since this is one of the assumptions in [Assumption 2.1](#), the asymptotic results where the optimal acceptance rate is 0.234 may not apply. This discontinuous target density case has been explored by Neal, Roberts, and Yuen (2012), which found that for product densities with a discontinuity at the boundary, such as the unit hypercube  $[0,1]^d$ , the efficiency of the RWM is maximized with an asymptotic optimal acceptance rate of 0.1353 as  $d \rightarrow \infty$ . We provide simulation studies in various dimensions to investigate the applicability of this particular 0.1353 acceptance rate figure in lower dimensions on the hypercube  $[0,1]^d$ . We use 300,000 iterations instead of the typical 200,000 for this target to make the plots sufficiently smooth.

Compared to the previous examples (see [Appendix A](#) for their ESJD versus acceptance rate curves, which tend to be smooth and gentle at the plateau), which seem to be fairly consistently optimal with an acceptance rate of approximately 0.234, we find (refer to [Table 5](#) and [Figure 6](#)) that the optimal acceptance rate for the hypercube target changes significantly as the dimension



**Figure 6.** ESJD vs. acceptance rate for the i.i.d. Hypercube  $[0, 1]^d$  target distribution  $\pi_{11}$  under RWM with a Gaussian proposal in dimensions  $d \in \{2, 5, 10, 30, 50, 100\}$  from top-left to bottom-right. Red dotted line indicates an acceptance rate of 0.234. Purple dotted line indicates an acceptance rate of 0.135.

increases, and notice that the plateau of the curve tends to get quite sharp as the dimension increases. This is because discontinuities introduce additional complexity in the target distribution, affecting the behavior and performance of the RWM algorithm.

As the dimension  $d$  increases, the algorithm's performance converges to the theoretical 0.1353 value. In this specific case, the shrinking optimal acceptance rate might be because of the unique zero or non-zero form of the target density beyond the hypercube boundary. As discussed by Neal, Roberts, and Yuen (2012), whether a proposed move is accepted depends completely on whether the proposed state is inside the hypercube. Subsequently, when using a Gaussian proposal, the probability that at least one proposed component will lie outside the boundary increases as dimension increases, leading to a higher probability of instant rejection when the proposal scaling factor independent of dimension is kept constant.

The trends in Figure 6 offer practical guidance for tuning. In high dimensions (e.g.  $d = 100$ ), the ESJD curve peaks sharply near the theoretical optimum of 0.1353, and tuning to 0.234 would result in a significant loss of efficiency. However, in lower dimensions (e.g.  $d = 2, 5, 10$ ), the optimal acceptance rate is much closer to 0.234 than to 0.1353. The ESJD curve is also flatter, meaning the penalty for slight mistuning is less severe. This suggests a mini-max strategy for practitioners unsure of the dimensionality effects: tuning to an acceptance rate of approximately 0.234 is a robust choice across all dimensions. While it could be suboptimal for very high dimensions, it is nearly empirically optimal for low dimensions, whereas tuning to 0.1353 in these instances would be highly inefficient even though theory supports it over the 0.234 rate for this specific target distribution.

### 3.6. Neal's funnel target density

To probe the limits of the 0.234 heuristic, we now examine a target distribution that is canonical for its challenging geometry: the funnel distribution introduced by Neal (2003). Unlike the Rosenbrock target, which has localized curvature, Neal's funnel possesses a global, non-linear dependency structure that presents a severe challenge to standard MCMC algorithms. In this distribution, a single variable controls the variance of all other variables, creating an extreme, state-dependent anisotropy. This structure should violate the geometric regularity conditions outlined in Assumption 2.2, particularly the target eccentricity and Hessian stability.

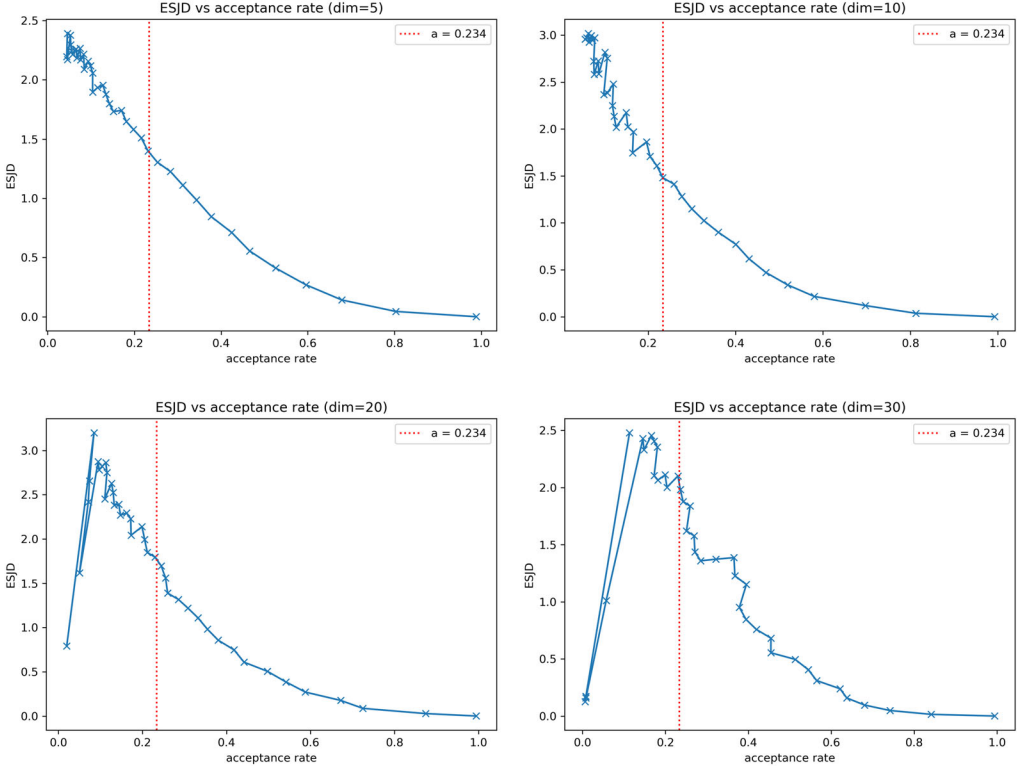
The  $d$ -dimensional Neal's funnel target density, which we denote  $\pi_{12}$ , is defined for  $\mathbf{x} = (x_1, x_2, \dots, x_d)$ . The density is given by the product of a marginal density for the first component  $x_1$  and conditional densities for the other components:

$$\pi_{12}(\mathbf{x}) = \mathcal{N}(x_1 | 0, 3^2) \prod_{i=2}^d \mathcal{N}(x_i | 0, \exp(x_1)) \quad (20)$$

The key feature is that the variance of each  $x_i$  after  $x_1$  is  $\exp(x_1)$ . This creates a funnel shape in the state space: when  $x_1$  is large and negative, the funnel is a very narrow neck where the  $x_i$ 's are tightly constrained; when  $x_1$  is positive, the funnel opens into a wide mouth where the  $x_i$ 's can have enormous variance.

For our experiments, we follow the standard parameterization from Neal (2003). We test this target in dimensions  $d \in \{5, 10, 20, 30\}$ . Due to the difficulty of sampling this target, we increased our simulation budget significantly, using 1,000,000 iterations per simulation (across 40 seeds, 40 million total Metropolis steps), a burn-in of 50,000 steps, and averaging the results over 30 random seeds for each data point. We continue using the Gaussian proposal with variance as described in (8). Our simulation results are shown in Figure 7.

Similarly to the 2-dimensional Rosenbrock, we find that the ESJD increases monotonically as the acceptance rate reduces in lower dimensions. In higher dimensions ( $d = 20, 30$ ), we observe a similar trend, but at the extreme ends of the curve, we observe a complete failure of the RWM algorithm to accept any reasonable number of steps as the average acceptance rate decreases



**Figure 7.** ESJD vs. acceptance rate for the Neal's funnel target  $\pi_{12}$  under RWM with a Gaussian proposal in dimensions  $d \in \{5, 10, 20, 30\}$  from top-left to bottom-right. Red dotted line indicates an acceptance rate of 0.234.

toward zero, leading to sudden random drops in ESJD. Notice that the curvature of the graph gets steeper with increasing dimension: it is nearly a straight line at  $d = 5$  and is much more curved by  $d = 30$ .

Overall, this behavior indicates an extreme sensitivity to the proposal scaling in higher dimensions. The practical implication is that even minor mistuning of the proposal variance will cause the sampler's efficiency to collapse to zero. This is a direct consequence of the target's geometry violating the conditions of [Assumption 2.2](#); the target has extreme eccentricity, and the Hessian is not locally stable. Consequently, the 0.234 acceptance rate is not a relevant or useful heuristic for this target. The more important conclusion, which Neal (2003) himself meant to show with this example, is that for targets with strong non-linear dependencies, no single proposal scale can be efficient in both the narrow neck and the wide mouth of the funnel, leading to a sampler that inevitably fails as dimensionality grows. This experiment effectively demonstrates a hard boundary for the class of problems on which the 0.234 RWM heuristic can be successfully applied.

## 4. Parallel tempering background

This section offers a brief introduction to the parallel tempering method and its own optimal scaling framework that we conduct experiments on in [Sec. 5](#). Many of the terms in [Sec. 2](#) are re-used but have slightly different definitions for the parallel tempering context.

### 4.1. Parallel tempering method

The parallel tempering method (Geyer 1991), also known as the Metropolis-coupled Markov chain Monte Carlo algorithm or the replica exchange method, is a version of the Metropolis

algorithm that specializes in sampling from multimodal distributions. In this case, we sample from a target distribution with density  $\pi(\mathbf{x})$  on a state space  $\mathcal{X}$ . We define a sequence of tempered target densities  $\pi^{\beta_j}(\mathbf{x})$  where  $0 \leq \beta_n < \beta_{n-1} < \dots < \beta_1 < \beta_0 = 1$  are the inverse temperature values. We require that  $\pi^{\beta_0}(\mathbf{x}) = \pi(\mathbf{x})$  (that is, that  $\beta_0 = 1$ ). Typically, the tempered target densities are simply powers of the original density, and we assume this for all parallel tempering experiments. Then, we run a Markov chain for each of the  $n + 1$  values of  $\beta$ , each with its own tempered target density. Eventually, the  $\beta_0 = 1$  chain, also called the “cold” chain, with its stationary density  $\pi^{\beta_0}(\mathbf{x}_0)$  should correspond to the original target density of interest  $\pi(\mathbf{x})$ , and the idea is that using information from other chains with higher temperatures (lower  $\beta$ ) will speed up the mixing in the cold chain.

In each iteration of the algorithm, we alternate between *within temperature* and *temperature swap* moves. The former indicates that each chain takes a standard RWM update within its respective  $\pi^{\beta_j}$ . The latter indicates that for two chains with  $\beta_j, \beta_k$  inverse temperature values that are adjacent in the sequence of  $\beta$  values, switch the chain values of  $\mathbf{x}_j$  and  $\mathbf{x}_k$  with probability

$$\min \left\{ 1, \frac{\pi^{\beta_j}(\mathbf{x}_k) \pi^{\beta_k}(\mathbf{x}_j)}{\pi^{\beta_j}(\mathbf{x}_j) \pi^{\beta_k}(\mathbf{x}_k)} \right\}. \quad (21)$$

## 4.2. Optimal scaling for parallel tempering

Since the hot chains can explore the state space more quickly, we would like to maximize how frequently we can swap values from the hottest chain to the coldest chain so that the cold chain can mix faster and escape local modes. To do this, we maximize the effective speed with which the chain values move along in the inverse temperature domain. The spacings of the inverse temperatures  $\beta_i$ ’s are crucial to this efficiency. If  $\beta_j$  and  $\beta_k$  are too far apart, we usually reject these swaps described by (21), but if they are too close, the swaps will not improve mixing. We would like to swap just the right amount to maximize mixing in the temperature domain.

The *expected squared jumping distance* (ESJD) for parallel tempering thus refers to the expected squared jump in inverse temperatures. Formalizing this, when we attempt to swap the chain values between the inverse temperatures  $\beta$  and  $\gamma := \beta + \epsilon$  where  $\beta, \epsilon > 0$  and  $\beta, \gamma \leq 1$ , the swap is either accepted, in which case the values move a squared distance of  $(\gamma - \beta)^2 = \epsilon^2$ , or the swap is rejected, in which case the distance moved is 0. This leads to a very natural definition for ESJD, which is

$$ESJD = \mathbb{E}[(\gamma - \beta)^2] = \epsilon^2 \times \mathbb{E}[\text{Pr}(\text{swap accepted})]. \quad (22)$$

Maximizing the asymptotic ESJD effectively maximizes the efficiency of the attempted swap moves in providing mixing in the temperature space, or in other words, maximizes the speed with which the chain values move in the inverse temperature space.

There is also a 0.234 swap acceptance rate theoretical result for parallel tempering. The required assumptions for the result are the following:

**Assumption Set 4.1.** (Atchadé, Roberts, and Rosenthal (2011)) Let  $\mathbf{x}$  be a random variable from the target density  $\pi$ .

1. The target density is an i.i.d. product  $\pi(\mathbf{x}) = \prod_{i=1}^d f(x_i)$  where  $f$  is a single-dimensional component density. There are other smoothness and regularity restrictions on the target density taken from the main results of Roberts, Gelman, and Gilks (1997); Roberts and Rosenthal (2001).
2. The tempered distributions are powers of the original density, so  $\pi^{(\beta)}(\mathbf{x}) = \prod_{i=1}^d f^{(\beta)}(x_i) = \left( \prod_{i=1}^d f(x_i) \right)^\beta$ .
3. The inverse-temperature spread  $\epsilon = d^{-1/2} \ell$  for some positive constant  $\ell$ .

Under these assumptions, choosing  $\ell$  such that the inverse temperature spacing yields a swap acceptance probability of approximately 0.234 is also optimal in the sense that it maximizes the ESJD of a parallel tempering algorithm.

## 5. Parallel tempering simulations

In this section, we provide parallel tempering experiments that investigate how well the 0.234 optimal swap acceptance rate holds for the various multimodal target distributions in [Sec. 3.3](#) in finite dimensions. The parallel tempering method is highly effective for exploring multimodal distributions (Atchadé, Roberts, and Rosenthal 2011; Hastings 1970) and thus we aim our experiments at these distributions to learn more about their applicability in practice.

### 5.1. Target distributions

#### 5.1.1. Multivariate Gaussian distribution

We first give an example of the standard multivariate Gaussian distribution to verify the correctness of our implementation. Note that the standard multivariate Gaussian as a target distribution satisfies the necessary conditions of Atchadé, Roberts, and Rosenthal (2011) that are stated in [Assumption 4.1](#).

#### 5.1.2. Rough carpet distribution

In this section, we use the same target distribution  $\pi_4$  from (12) as described in [Sec. 3.3.1](#). Here, we examine the simpler case with homogeneous scaling factors  $f_d(\mathbf{x}) = \prod_{i=1}^n f(x_i)$ . In the individual component density  $f(x)$  definition, we set  $(\mu_1, \mu_2, \mu_3) = (-15, 0, 15)$ . We provide an example traceplot of the cold chain from one of the experiment simulations in [Appendix A Figure A11](#) to demonstrate what mixing looks like in the first three components (all components are i.i.d.). Unlike the standard random-walk Metropolis which cannot escape from the central mode in this setting of the target distribution, the values in the cold chain swap between modes considerably often due to the swapping of values between chains.

#### 5.1.3. Three mixture distribution

In this section, we use the same target distribution as described in [Sec. 3.3.3 \(14\)](#). We set weights  $w_1 = w_2 = w_3 = \frac{1}{3}$ , means  $\mu_1 = (-15, 0, 0, \dots, 0)$ ,  $\mu_2 = (0, 0, \dots, 0)$ ,  $\mu_3 = (15, 0, 0, \dots, 0)$  and covariances  $\Sigma_1 = \Sigma_2 = \Sigma_3 = I_d$ . We provide an example traceplot of the cold chain in [Appendix A Figure A12](#) to demonstrate what mixing looks like in the first three components.

### 5.2. Methodology

We conduct experiments in dimension  $d \in \{10, 20, 30\}$  that examine the trend of ESJD with swap acceptance rate on these distributions. For each distribution, we run many simulations where each simulation is a parallel tempering algorithm with 30 different average swap acceptance rate values. We explain how we set up these algorithms to match these swap rates (through constructing a specific inverse temperature ladder) in [Sec. 5.2.1](#). Each parallel tempering algorithm runs for 500,000 iterations, so every individual chain in the algorithm takes 500,000 steps. We attempt a temperature swap every 100 steps. Furthermore, we ran each algorithm instance over 20 different seeds and average the results to reduce the effects of randomness caused by a particular seed.

For each experiment, we set the tempered target distribution to simply be the original distribution raised to the power of  $\beta$ . We set the modes to be far away enough such that the other modes were not reached by a standard random-walk Metropolis algorithm after 100,000 iterations, but still reasonably close so that the parallel tempering method would show the cold chain values would swap between modes somewhat frequently. The within-chain proposal distribution for each chain is given by the multivariate Gaussian (8) with  $\sigma^2 = 2.38^2/d$ , which is suggested by Roberts, Gelman, and Gilks (1997) to be optimal for high dimensions. We calculate the optimal swap acceptance rate by finding each simulation's specified swap acceptance rate used to construct the iterative temperature ladder that maximizes the ESJD, and take the average of the maximizing acceptance rates over 20 seeds.

### 5.2.1. Constructing an inverse temperature ladder iteratively

The most common method to construct an inverse temperature ladder selects the inverse temperatures using a geometric series spacing. However, since we are examining the 0.234 swap acceptance rule, we need to have a way of constructing the inverse temperature spacings such that the probability of a swap between adjacent chains is approximately 0.234 (or any other value). To construct an inverse temperature ladder with a desired swap acceptance rate  $s$ , we use an iterative procedure adapted from (Atchadé, Roberts, and Rosenthal 2011, Section 2.2). Starting with  $\beta_0 = 1$  and a minimum value  $\beta_{\min} = 0.01$ , we iteratively add  $\beta$ 's to the ladder. Let  $\beta_{\text{curr}}$  denote the most recent  $\beta$  in the ladder. We initialize  $\rho_n = 0.5$  ( $n = 1$  initially) and set  $\beta^* = \beta_{\text{curr}}(1 + e^{\rho_n})^{-1}$ . To determine if  $\beta^*$  should be added to the ladder, we draw  $N$  samples (our GPU implementation uses  $N = 1,000,000$ ) from the target distribution tempered by  $\beta^*$  and  $\beta_{\text{curr}}$ , and calculate the average swap probability  $a$ . If  $a$  is within  $s \pm 0.0001$ , we add  $\beta^*$  to the ladder and set  $\beta_{\text{curr}} = \beta^*$ . Otherwise, we update  $\rho_{n+1} = \rho_n + n^{-0.25}(a - s)$  and recalculate  $\beta^*$ . This process continues until  $\beta^* \leq \beta_{\min}$ , at which point we add  $\beta_{\min}$  to the ladder and terminate.

## 5.3. Results

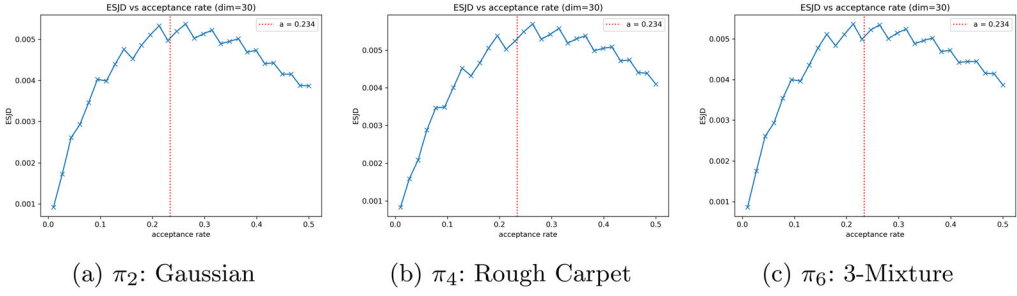
The empirical optimal swap acceptance rates for the three target distributions are summarized in Table 6, and sample ESJD curves for  $d = 30$  shown in Figure 8. The ESJD curves for PT are visibly more jagged than their RWM counterparts. This might be because the estimation of ESJD in this context is more susceptible to Monte Carlo error even with a large number of simulation steps, since the efficiency or acceptance rate of a swap move depends on the joint stationary distribution of two adjacent chains, and this is repeated for each temperature construction. Another possible reason might be the determinism in the way  $\rho_n$  changes. Future work could investigate alternative algorithms to compute a fixed discrete temperature ladder that provides a certain swap rate prior to running the PT algorithm, perhaps by computing each change in  $\rho_n$  from a distribution. Nonetheless, clear trends emerge from the results.

For both the standard Gaussian ( $\pi_2$ ) and the non-product Three-Mixture ( $\pi_6$ ) targets, the optimal swap acceptance rate is consistently in the neighborhood of 0.234. For the Three-Mixture target, the rate converges cleanly toward the theoretical value, from 0.261 in  $d = 10$  to 0.237 in  $d = 30$ . The Gaussian target shows a similar pattern, with an optimal rate of 0.247 at  $d = 30$ .

**Table 6.** Empirical swap acceptance rates that maximize ESJD for various PT target densities.

Target	$d = 10$	$d = 20$	$d = 30$
$\pi_2$ : Standard Multivariate Gaussian	0.2618	0.2685	0.2474
$\pi_4$ : Rough Carpet	0.2660	0.2803	0.2761
$\pi_6$ : 3-Mixture	0.2609	0.2660	0.2372

Based on Fig 8, the jagged pattern of the ESJD curves suggests that the true optimal acceptance rates are in a neighborhood of these values. See Section 5.2 for calculation of optimal swap acceptance rates.



**Figure 8.** ESJD vs. swap acceptance rate for the parallel tempering algorithm with the  $\pi_2, \pi_4, \pi_6$  target distributions in dimension  $d = 30$ . Red dotted line indicates a swap acceptance rate of 0.234.

The standard Gaussian result provides a successful validation of our experimental setup against the theory of Atchadé, Roberts, and Rosenthal (2011), as this target satisfies the i.i.d. product form assumption. More importantly, the result for the Three-Mixture target suggests that the 0.234 heuristic for PT swaps is robust to violations of the i.i.d. product assumption, much like its RWM counterpart. Although the target is a sum of densities, its underlying geometric regularity (a shell composed of three isotropic modes) seems sufficient for the optimal scaling 0.234 result to hold. For a practitioner, this suggests that constructing a temperature ladder with a swap acceptance probability of 0.234 is a viable and likely near-optimal strategy, even for complex, non-product multimodal targets, provided they are not pathologically structured.

Notably, the Rough Carpet target ( $\pi_4$ ) shows a deviation. The optimal swap acceptance rate is consistently higher than 0.234 and increases away from the theoretical value with increasing dimension. This is surprising because the Rough Carpet target does satisfy the i.i.d. product form and other conditions of Assumption 4.1 by Atchadé, Roberts, and Rosenthal (2011).

There are two plausible explanations that we favor here. The failure of the 0.234 rule here could be because the extreme multimodality of the target could reduce the practical utility of the “hot” chains. The target possesses  $3^d$  modes, and its probability mass is spread thinly across this vast landscape. For a hot chain where the inverse temperature  $\beta$  is close to 0, the tempered density becomes extremely flat and loses the slight peak that a unimodal target would have. Consequently, it fails to efficiently sample from the specific modal regions that are of interest to the colder chains; the information flow from hot to cold chains is broken. To swap successfully more often, the temperature steps must be made smaller, which forces the swap acceptance rate to be higher.

An alternative and more simple explanation is that, due to the jaggedness of the ESJD vs swap acceptance rate trend in Figure 8, the discrete temperature ladder is only approximately optimal, and the true empirically optimal swap rate is somehow confounded by determinism in the ladder construction algorithm. As mentioned previously, alternative methods to construct temperature ladders yielding certain swap acceptance rates prior to running a PT algorithm could be an area for future work.

## 6. Discussion

We have presented an extensive empirical investigation into the generalizability of the 0.234 acceptance rate heuristic for both Random-Walk Metropolis (RWM) and Parallel Tempering (PT) algorithms, using a variety of target and proposal distributions. Our results demonstrate that the 0.234 rule’s utility extends beyond the classical i.i.d. product setting of Roberts, Gelman, and Gilks (1997). The heuristic remains approximately optimal for: (i) non-Gaussian Laplace and Uniform proposal distributions that satisfy a “shell” condition with a less stringent symmetry requirement than spherical symmetry; (ii) multimodal targets such as the “rough carpet” i.i.d

product with  $3^d$  modes, and non-product targets constructed as a sum of isotropic components, such as our Three-Mixture model; and (iii) even for highly correlated, non-linear targets like the Full and Hybrid Rosenbrock in sufficiently high dimensions ( $d \geq 20$ ). These findings provide empirical validation for the generalized theory of Sherlock (2013), which predicts that as long as the target’s geometry is sufficiently regular and isotropic in the high-dimensional limit, the 0.234 rule should hold. Our results in Sec. 3.2 even suggest that weaker conditions on the symmetry of the proposal than those required by Sherlock (2013) could be sufficient.

That being said, there are cases where 0.234 does not seem as relevant. The main case is in very low dimensions; in all our experiments with varying target densities, the optimal acceptance rate strays from 0.234 below a certain threshold which varies for each target density. In particular, targets composed of independent, low-dimensional anisotropic blocks (Even Rosenbrock) or those that are globally anisotropic in low dimensions (2D Rosenbrock), have an optimal acceptance rate significantly below 0.234. In these cases, we hypothesize the high-dimensional averaging effect that stabilizes the target’s geometry has not occurred, and the sampler’s performance is dictated by the challenging local geometry. As the dimension increases, the law of large numbers mitigates the effect of any single scaling factor or local correlation, so the target becomes effectively less eccentric, complying with the theory of Sherlock (2013), and the optimal acceptance rate recovers toward 0.234. On the other hand, the continuous hypercube and Neal’s funnel completely fail the 0.234 heuristic. Although the hypercube is continuously differentiable everywhere except on its boundary, which has zero measure, an intuitive argument is that as  $d$  increases, the probability that the proposal has all the values inside the hypercube decreases. As for the funnel, no single proposal scaling can be globally efficient, which is just a limitation of the RWM algorithm.

Next, we discuss the optimal swap acceptance rate figure for parallel tempering and compare it with the optimal acceptance rate for random-walk Metropolis. On one hand, the results for the multivariate Gaussian and three-mixture density examples show that the 0.234 swap acceptance rate figure may be optimal even in lower dimensions. Although costly, if the practitioner is willing and able to construct the inverse temperature ladder iteratively with a spacing given by a swap acceptance probability, the spacing dictated by a 0.234 swap acceptance probability may be optimal for the practitioner. However, the theoretical 0.234 figure may not always be applicable: despite satisfying the i.i.d. conditions of the PT optimal scaling theory (Atchadé, Roberts, and Rosenthal 2011), the extreme multimodality ( $3^d$  modes) of the “rough carpet” product density appears to render the hot chains ineffective in our experiments, with the caveat that the apparent failure of the 0.234 rule could partly be due to the jaggedness of our experimental ESJD vs swap acceptance rate curves. One hypothesis for this observed failure is that because the probability mass is spread so thinly, the hot chains fail to sample from meaningful modal regions, breaking the flow of information to the cold chain. This forces the algorithm to use smaller temperature steps (and thus higher swap acceptance rates) to achieve mixing, representing a practical failure of the 0.234 PT rule.

Not only that, constructing a temperature ladder iteratively with the “optimal spacing” is challenging and time-consuming. It requires the ability to draw samples from the target distribution, which is itself the main goal of the parallel tempering algorithm. Without a direct way to sample from the target, a practitioner could run an MCMC algorithm with the  $\beta$  value being proposed for the new addition to the ladder, but this may be time-consuming with many attempts required until a suitable  $\beta$  is found for each step of the ladder.

Since this optimal spacing may not yield the optimal ESJD in practice for finite, lower-dimensional targets and with a finite number of runs, an alternative approach is to adaptively adjust the spacings of the temperature ladder during the algorithm’s runs based on its recent ESJD performance over a recent window as guidance to practitioners. This is known as adaptive parallel tempering (Miasojedow, Moulines, and Vihola 2013). Previous literature shows that adaptive parallel tempering algorithms can achieve substantial efficiency gains over a standard

geometric temperature spacing (Vousden, Farr, and Mandel 2016) while still converging to the target distribution (Miasojedow, Moulines, and Vihola 2013) under the assumptions that the target distribution is sufficiently regular and has tails decaying faster than exponentially. Adaptive parallel tempering may suffice to be “good enough” for most practitioners whilst avoiding the costly temperature construction algorithm.

Lastly, while we have established that the 0.234 acceptance rate for RWM is still fairly good even in lower dimensions and on some target distributions that may not have the i.i.d. product form, we provide some precautions to the practitioner. The 0.234 acceptance rate is a good heuristic to tune the algorithm to, yet there are still many cases where it is not optimal in terms of efficiency, such as the continuous hypercube example in Sec. 3.5. Even if the 0.234 acceptance rate is in fact optimal, the acceptance rate alone does not necessarily guarantee good samples, as illustrated by Figure 2. What literature (Roberts and Rosenthal 2001; Sherlock, Fearnhead, and Roberts 2010) indicates is that another very important thing in practice beyond just the acceptance rate is defining a good proposal covariance matrix  $\Sigma_d$  that is approximately proportional to the target covariance matrix  $\Sigma_\pi$ ; doing so can significantly increase the asymptotic relative efficiency. Realistically, it may be impossible to know  $\Sigma_\pi$  in advance or even provide a good estimate for it. In fact, understanding how we can extend this result to correlated targets is an ongoing challenge (Yang, Roberts, and Rosenthal 2020) and immediately relevant to the practitioner since real-world data tends to have correlations between variables. Practitioners may consider using the adaptive Metropolis algorithm (Haario, Saksman, and Tamminen 2001) to dynamically update the algorithm’s proposal covariance matrix. This adaptive algorithm is very useful and yields significant speed benefits in low and high dimensions (Craiu, Rosenthal, and Yang 2009; Roberts and Rosenthal 2009), but an important potential pitfall to note is that adaptive methods are not guaranteed to converge to the correct target density; in many cases, they may fail to converge or converge to something completely different (Roberts and Rosenthal 2007, 2009).

## Acknowledgements

Resources used in preparing this research were provided, in part, by the Province of Ontario, the Government of Canada through CIFAR, and companies sponsoring the Vector Institute. We also thank the anonymous reviewer(s) for their helpful comments on earlier versions of this paper.

## Author contributions

CRediT: **Aidan Li**: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Project administration, Software, Validation, Visualization, Writing – original draft, Writing – review & editing; **Liyan Wang**: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Writing – original draft; **Tianye Dou**: Conceptualization, Software, Writing – original draft; **Jeffrey S. Rosenthal**: Conceptualization, Project administration, Supervision, Writing – original draft, Writing – review & editing.

## Disclosure statement

The authors did not receive any funding for this submitted work and have no relevant financial or non-financial competing interests. The code implementing the experiments and generating the figures used in this work are publicly available on GitHub at <https://github.com/aidanmrli/rwm-pt-pytorch>.

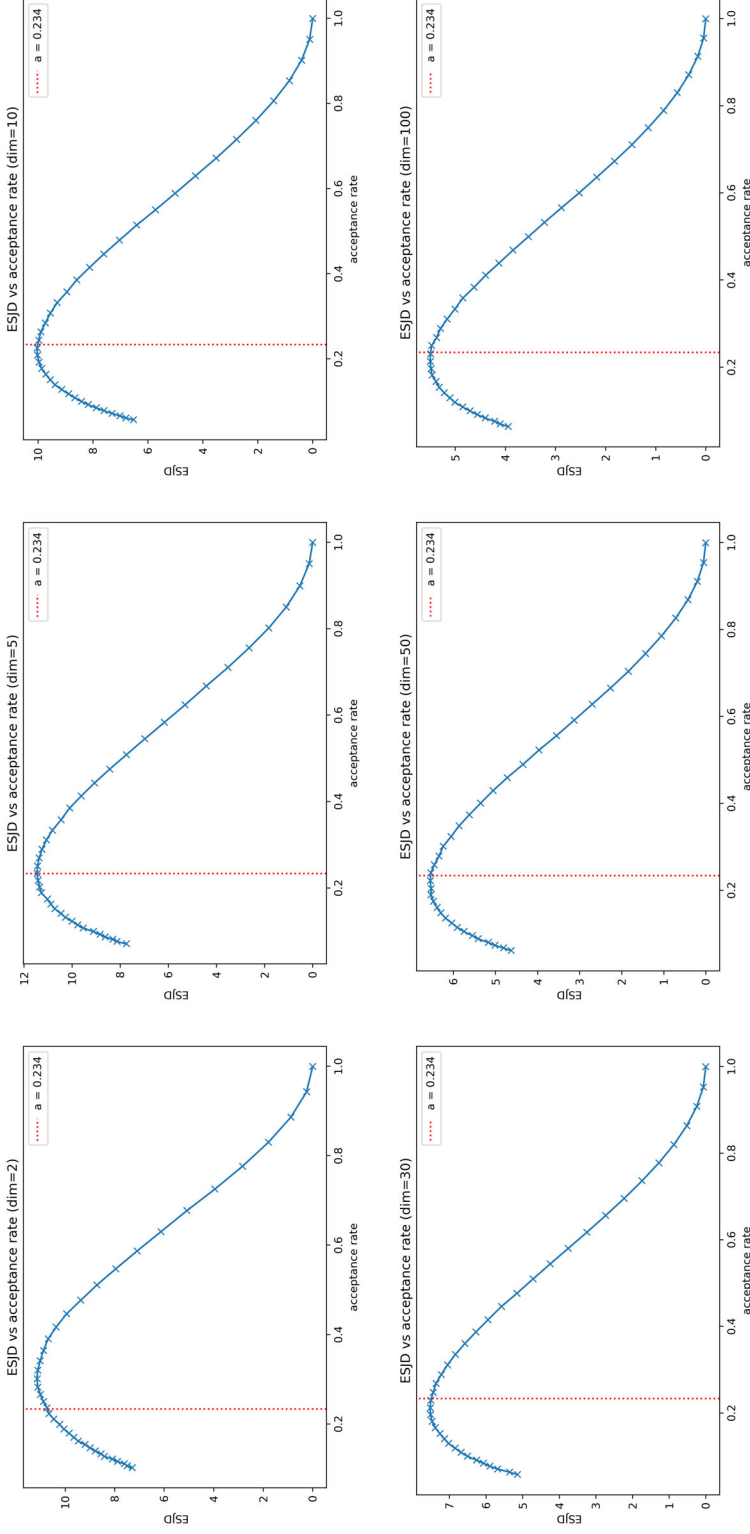
During the preparation of revisions, the authors used generative AI large language models (Gemini 2.5 Pro Preview 05-06, Claude 4.0 Sonnet) to assist with coding and manuscript revision. Generative AI was utilized to help generate, debug, and optimize code for the simulations. The new generated code based on PyTorch allows for the use of Graphics Processing Units (GPUs) when running RWM and PT algorithms to significantly speed up simulations and run more simulations at once. For manuscript revision, Generative AI was used to improve phrasing and clarity on text written by the authors. The authors thoroughly reviewed and edited all AI-generated output and take full responsibility for the scientific integrity and final content of this publication.

## References

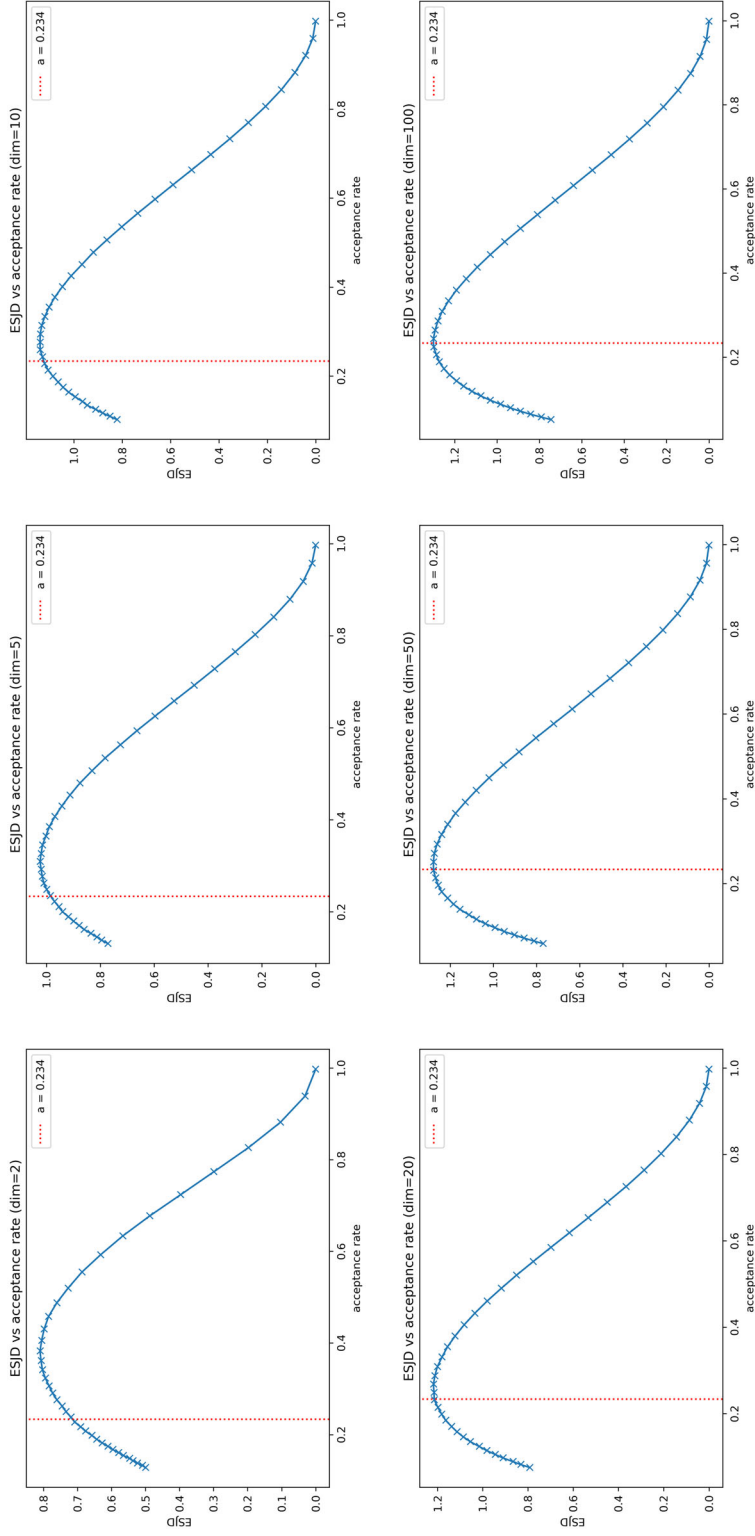
- Atchadé, Y. F., G. O. Roberts, and J. S. Rosenthal. 2011. Towards optimal scaling of Metropolis-coupled Markov chain Monte Carlo. *Statistics and Computing* 21 (4):555–68. doi:10.1007/s11222-010-9192-1.
- Brooks, S., A. Gelman, G. Jones, and X.-L. Meng, eds. 2011. *Handbook of Markov Chain Monte Carlo*. New York: Chapman and Hall/CRC. doi:10.1201/b10905.
- Breyer, L. A., and G. O. Roberts. 2000. From Metropolis to diffusions: Gibbs states and optimal scaling. *Stochastic Processes and Their Applications* 90 (2):181–206. doi:10.1016/S0304-4149(00)00041-7.
- Bédard, M., and J. S. Rosenthal. 2008. Optimal scaling of Metropolis algorithms: Heading toward general target distributions. *Canadian Journal of Statistics* 36 (4):483–503. doi:10.1002/cjs.5550360401.
- Beskos, A., G. Roberts, and A. Stuart. 2009. Optimal scalings for local Metropolis-Hastings chains on nonproduct targets in high dimensions. *The Annals of Applied Probability* 19 (3):863–98. doi:10.1214/08-AAP563.
- Beskos, A., G. Roberts, A. Thiery, and N. Pillai. 2018. Asymptotic analysis of the random walk Metropolis algorithm on ridged densities. *The Annals of Applied Probability* 28 (5):2966–3001. doi:10.1214/18-AAP1380.
- Bédard, M. 2007. Weak convergence of Metropolis algorithms for non-i.i.d. target distributions. *The Annals of Applied Probability* 17 (4):1222–44. doi:10.1214/105051607000000096.
- Bédard, M. 2008. Efficient sampling using metropolis algorithms: Applications of optimal scaling results. *Journal of Computational and Graphical Statistics* 17 (2):312–32. doi:10.1198/108571108X319970.
- Craiu, R. V., J. Rosenthal, and C. Yang. 2009. Learn from thy neighbor: Parallel-chain and regional adaptive MCMC. *Journal of the American Statistical Association* 104 (488):1454–66. doi:10.1198/jasa.2009.tm08393.
- Dixon, L. C. W., and D. J. Mills. 1994. Effect of rounding errors on the variable metric method. *Journal of Optimization Theory and Applications* 80 (1):175–9. doi:10.1007/BF02196600.
- Geyer, C. J. 1991. Markov chain Monte Carlo maximum likelihood. *Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface*, 156–63.
- Goodman, J., and J. Weare. 2010. Ensemble samplers with affine invariance. *Communications in Applied Mathematics and Computational Science* 5 (1):65–80. doi:10.2140/camcos.2010.5.65.
- Hastings, W. K. 1970. Monte Carlo sampling Methods Using Markov chains and their applications. *Biometrika* 57 (1):97–109. doi:10.2307/2334940.
- Huijser, D., J. Goodman, and B. J. Brewer. 2022. Properties of the affine-invariant ensemble sampler’s ‘stretch move’ in high dimensions. *Australian & New Zealand Journal of Statistics* 64 (1):1–26. doi:10.1111/anzs.12358.
- Hunt-Smith, N. T., W. Melnitchouk, F. Ringer, N. Sato, A. W. Thomas, and M. J. White. 2024. Accelerating Markov Chain Monte Carlo sampling with diffusion models. *Computer Physics Communications* 296:109059. doi:10.1016/j.cpc.2023.109059.
- Haario, H., E. Saksman, and J. Tamminen. 2001. An adaptive Metropolis algorithm. *Bernoulli* 7 (2):223. doi:10.2307/3318737.
- Jarner, S., and G. Roberts. 2002. Polynomial convergence rates of Markov chains. *The Annals of Applied Probability* 12 (1):224–47. doi:10.1214/aoap/1015961162.
- Kamatani, K. 2020. Random walk metropolis algorithm in high dimension with non-gaussian target distributions. *Stochastic Processes and Their Applications* 130 (1):297–327. doi:10.1016/j.spa.2019.03.002.
- Miltzer, B. 2025. Ensemble Monte Carlo calculations with five novel moves. *Computer Physics Communications* 307:109424. doi:10.1016/j.cpc.2024.109424.
- Miasojedow, B., E. Moulines, and M. Vihola. 2013. An adaptive parallel tempering algorithm. *Journal of Computational and Graphical Statistics* 22 (3):649–64. doi:10.1080/10618600.2013.778779.
- Mattingly, J. C., N. S. Pillai, and A. M. Stuart. 2012. Diffusion limits of the random walk Metropolis algorithm in high dimensions. *The Annals of Applied Probability* 22 (3):881–930. doi:10.1214/10-AAP754.
- Metropolis, N., A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. 1953. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics* 21 (6):1087–92. doi:10.1063/1.1699114.
- Neal, R. M. 2003. Slice sampling. *The Annals of Statistics* 31 (3):705–41. doi:10.1214/aos/1056562461.
- Neal, P., and G. Roberts. 2006. Optimal scaling for partially updating MCMC algorithms. *The Annals of Applied Probability* 16 (2):475–515. doi:10.1214/105051605000000791.
- Neal, P., and G. Roberts. 2011. Optimal scaling of random walk metropolis algorithms with non-Gaussian proposals. *Methodology and Computing in Applied Probability* 13 (3):583–601. doi:10.1007/s11009-010-9176-9.
- Neal, P., G. Roberts, and W. K. Yuen. 2012. Optimal scaling of random walk Metropolis algorithms with discontinuous target densities. *The Annals of Applied Probability* 22 (5):1880–1927. doi:10.1214/11-AAP817.
- Pasarica, C., and A. Gelman. 2010. Adaptively scaling the metropolis algorithm using expected squared jumped distance. *Statistica Sinica* 20 (1):343–64.
- Pagani, F., M. Wiegand, and S. Nadarajah. 2022. An n-dimensional Rosenbrock distribution for Markov chain Monte Carlo testing. *Scandinavian Journal of Statistics* 49 (2):657–80. doi:10.1111/sjos.12532.
- Robert, C. P., and G. Casella. 2004. Monte Carlo statistical methods. *Springer Texts in Statistics*. New York, NY: Springer. doi:10.1007/978-1-4757-4145-2.

- Roberts, G. O., A. Gelman, and W. R. Gilks. 1997. Weak convergence and optimal scaling of random walk Metropolis algorithms. *The Annals of Applied Probability* 7 (1):110–20. doi:[10.1214/aoap/1034625254](https://doi.org/10.1214/aoap/1034625254).
- Roberts, G. O. 1998. Optimal metropolis algorithms for product measures on the vertices of a hypercube. *Stochastics and Stochastic Reports* 62 (3–4):275–83. doi:[10.1080/17442509808834136](https://doi.org/10.1080/17442509808834136).
- Rosenbrock, H. H. 1960. An automatic method for finding the greatest or least value of a function. *The Computer Journal* 3 (3):175–84. doi:[10.1093/comjnl/3.3.175](https://doi.org/10.1093/comjnl/3.3.175).
- Rosenthal, J. S. 1995. Minorization conditions and convergence rates for Markov Chain Monte Carlo. *Journal of the American Statistical Association* 90 (430):558–66. doi:[10.2307/2291067](https://doi.org/10.2307/2291067).
- Roberts, G. O., and J. S. Rosenthal. 2001. Optimal scaling for various Metropolis-Hastings algorithms. *Statistical Science* 16 (4):351–67. doi:[10.1214/ss/1015346320](https://doi.org/10.1214/ss/1015346320).
- Roberts, G. O., and J. S. Rosenthal. 2007. Coupling and ergodicity of adaptive Markov Chain Monte Carlo algorithms. *Journal of Applied Probability* 44 (2):458–75. doi:[10.1239/jap/1183667414](https://doi.org/10.1239/jap/1183667414).
- Roberts, G. O., and J. S. Rosenthal. 2009. Examples of adaptive MCMC. *Journal of Computational and Graphical Statistics* 18 (2):349–67. doi:[10.1198/jcgs.2009.06134](https://doi.org/10.1198/jcgs.2009.06134).
- Roberts, G., and J. Rosenthal. 2014. Minimising MCMC variance via diffusion limits, with an application to simulated tempering. *The Annals of Applied Probability* 24 (1):131–149. doi:[10.1214/12-AAP918](https://doi.org/10.1214/12-AAP918).
- Sherlock, C., P. Fearnhead, and G. O. Roberts. 2010. The Random Walk metropolis: Linking theory and practice through a case study. *Statistical Science* 25 (2):172–90. doi:[10.1214/10-STS327](https://doi.org/10.1214/10-STS327).
- Sherlock, C. 2013. Optimal scaling of the Random Walk Metropolis: General criteria for the 0.234 acceptance rule. *Journal of Applied Probability* 50 (1):1–15. doi:[10.1239/jap/1363784420](https://doi.org/10.1239/jap/1363784420).
- Sherlock, C., and G. Roberts. 2009. Optimal scaling of the random walk Metropolis on elliptically symmetric unimodal targets. *Bernoulli* 15 (3):774–98. doi:[10.3150/08-BEJ176](https://doi.org/10.3150/08-BEJ176).
- Sherlock, C., A. H. Thiery, G. O. Roberts, and J. S. Rosenthal. 2015. On the efficiency of pseudo-marginal random walk Metropolis algorithms. *The Annals of Statistics* 43 (1):238–75. doi:[10.1214/14-AOS1278](https://doi.org/10.1214/14-AOS1278).
- Vousden, W. D., W. M. Farr, and I. Mandel. 2016. Dynamic temperature selection for parallel tempering in Markov chain Monte Carlo simulations. *Monthly Notices of the Royal Astronomical Society* 455 (2):1919–37. doi:[10.1093/mnras/stv2422](https://doi.org/10.1093/mnras/stv2422).
- Yang, J., G. O. Roberts, and J. S. Rosenthal. 2020. Optimal scaling of random-walk Metropolis algorithms on general target distributions. *Stochastic Processes and Their Applications* 130 (10):6094–132. doi:[10.1016/j.spa.2020.05.004](https://doi.org/10.1016/j.spa.2020.05.004).

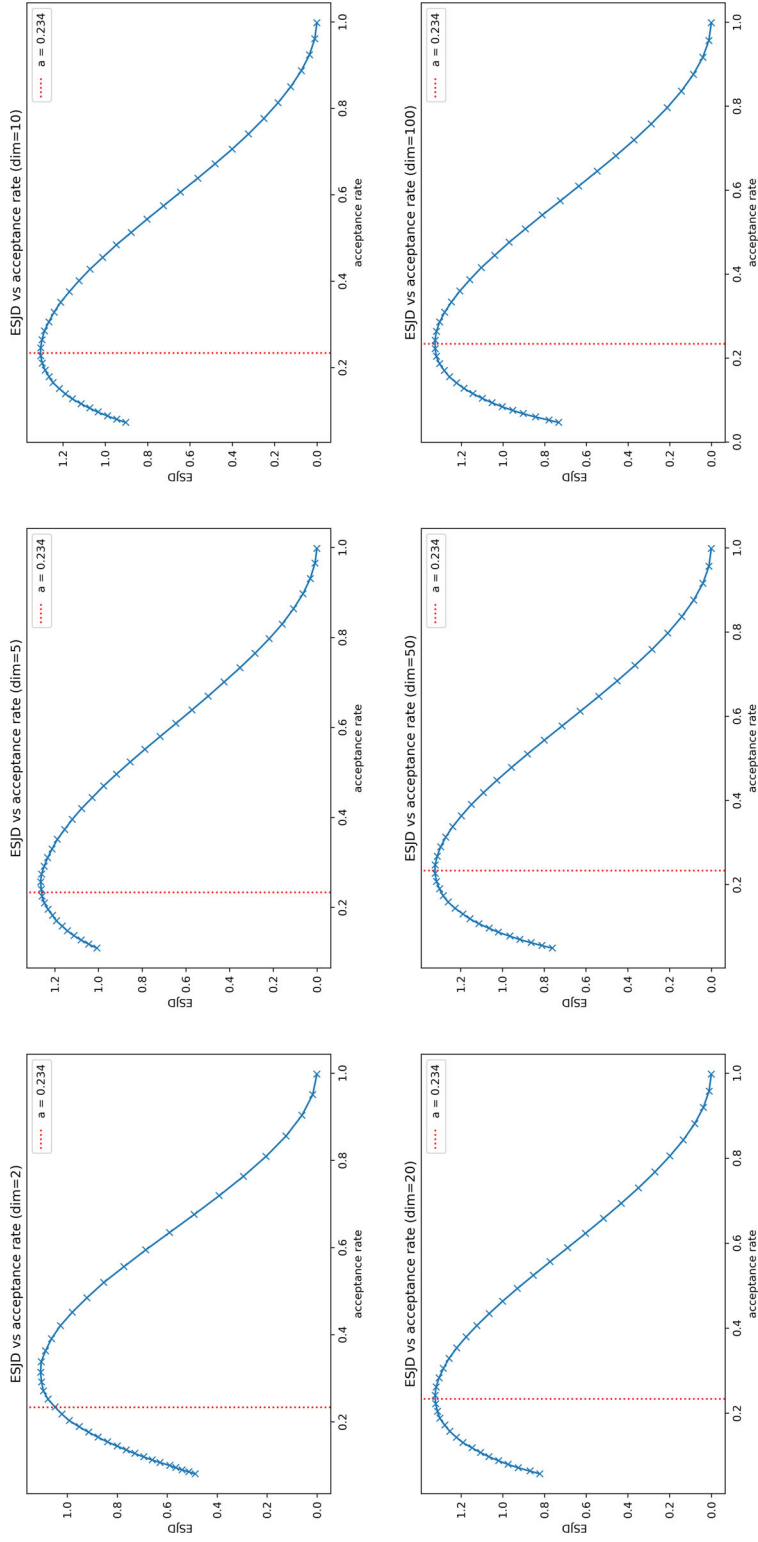
## A Plots for RWM simulations



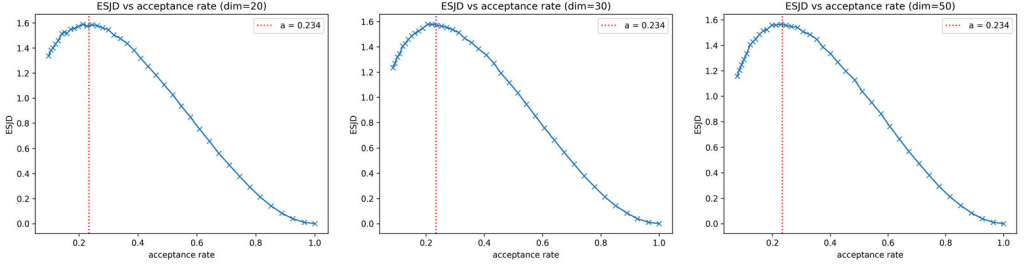
**Figure A1.** ESJD vs. acceptance rate for the i.i.d. Gamma(3, 2) target distribution  $\pi_2$  (Equation 10) under RWM with a Gaussian proposal in dimensions  $d \in \{2, 5, 10, 30, 50, 100\}$  from top-left to bottom-right. Red dotted line indicates an acceptance rate of 0.234.



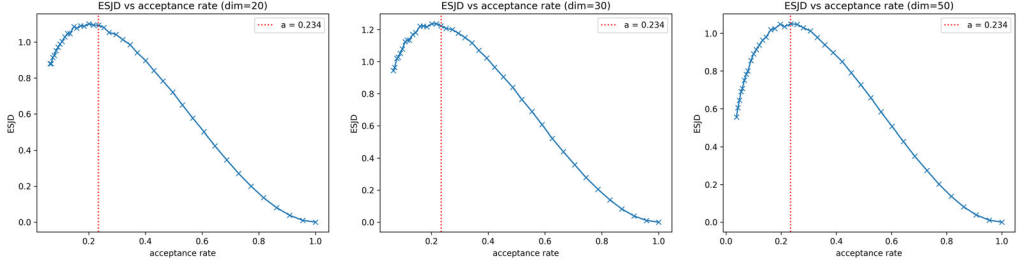
**Figure A2.** ESJD vs. acceptance rate for the standard multivariate Gaussian target distribution  $\pi_3$  under RWM with a multivariate Laplace proposal  $Q_L$  (Equation 11) in dimensions  $d \in \{2, 5, 10, 30, 50, 100\}$  from top-left to bottom-right. Red dotted line indicates an acceptance rate of 0.234.



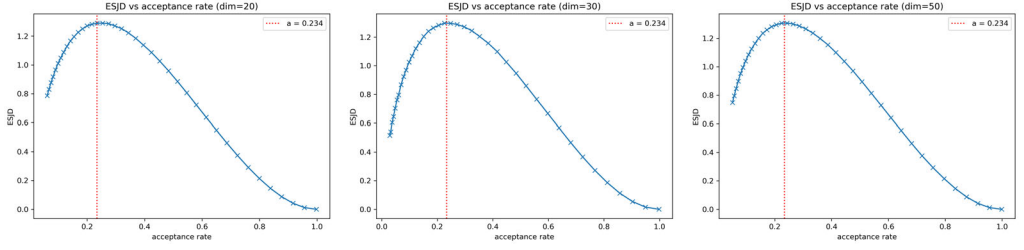
**Figure A3.** ESJD vs. acceptance rate for the standard multivariate Gaussian target distribution  $\pi_3$  under RWM with a multivariate Uniform proposal  $Q_J$  (Equation 11) in dimensions  $d \in \{2, 5, 10, 30, 50, 100\}$  from top-left to bottom-right. Red dotted line indicates an acceptance rate of 0.234.



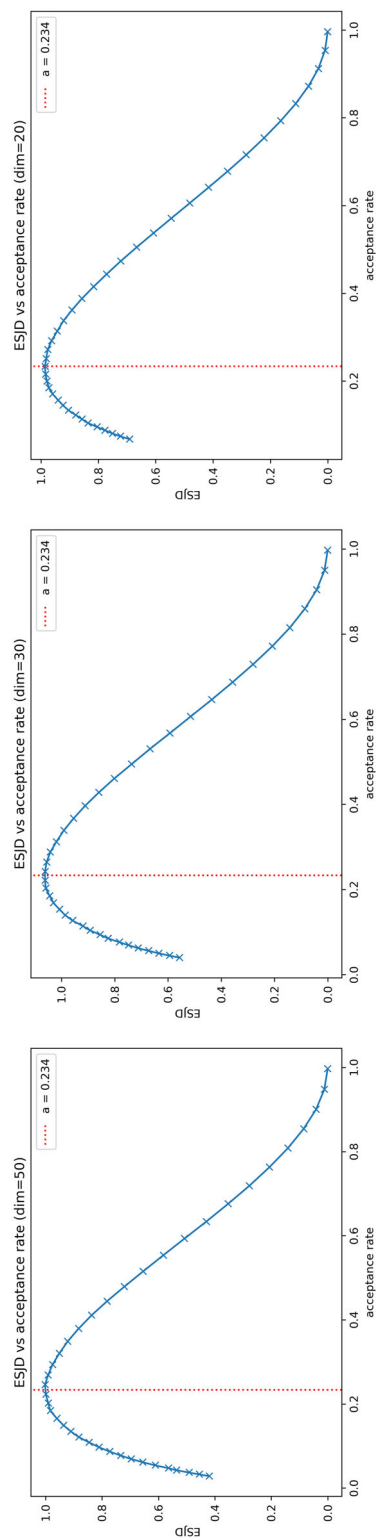
**Figure A4.** ESJD vs. acceptance rate for the i.i.d. rough carpet target distribution  $\pi_4$  (Equation 12) under RWM with a Gaussian proposal in dimensions  $d \in \{20, 30, 50\}$  from left to right. Red dotted line indicates an acceptance rate of 0.234.



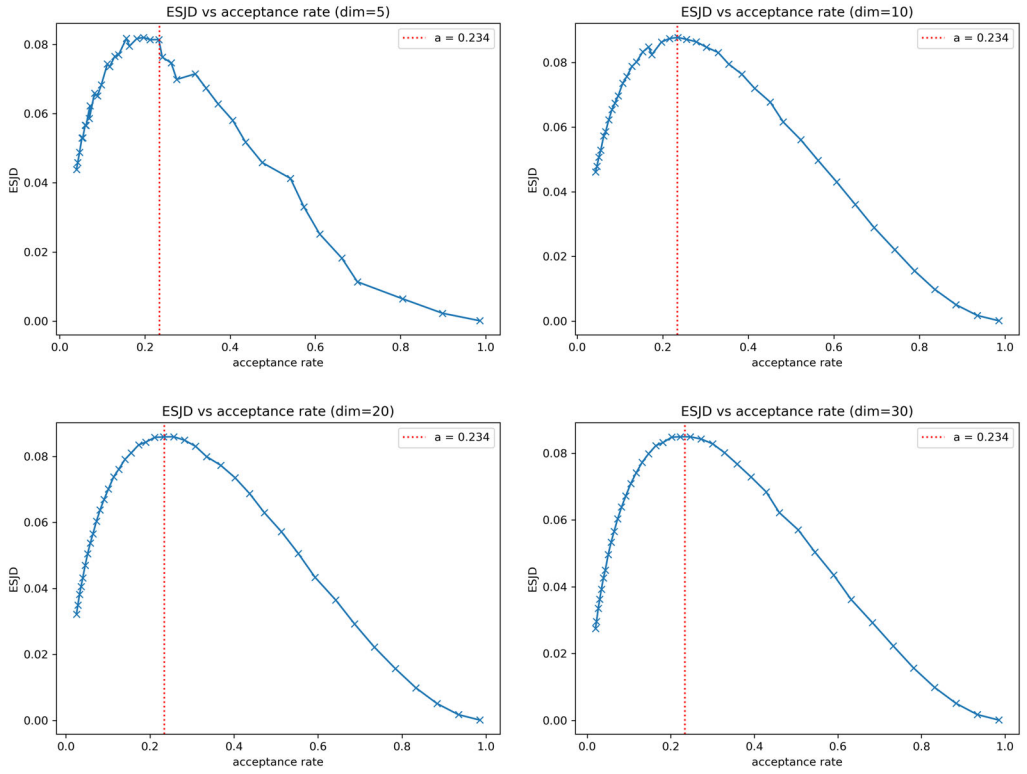
**Figure A5.** ESJD vs. acceptance rate for the inhomogeneously scaled i.i.d. rough carpet target distribution  $\pi_5$  (Equation 13) under RWM with a Gaussian proposal in dimensions  $d \in \{20, 30, 50\}$  from left to right. Red dotted line indicates an acceptance rate of 0.234.



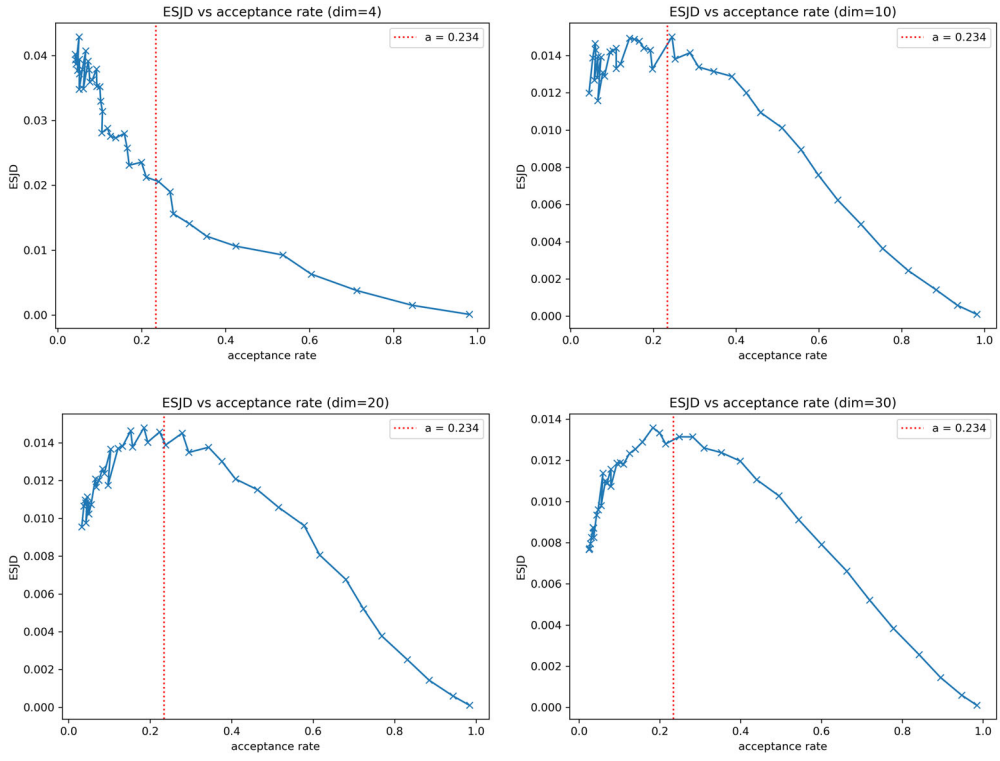
**Figure A6.** ESJD vs. acceptance rate for the three mixture target distribution  $\pi_6$  (Equation 14) under RWM with a Gaussian proposal in dimensions  $d \in \{20, 30, 50\}$  from left to right. Red dotted line indicates an acceptance rate of 0.234.



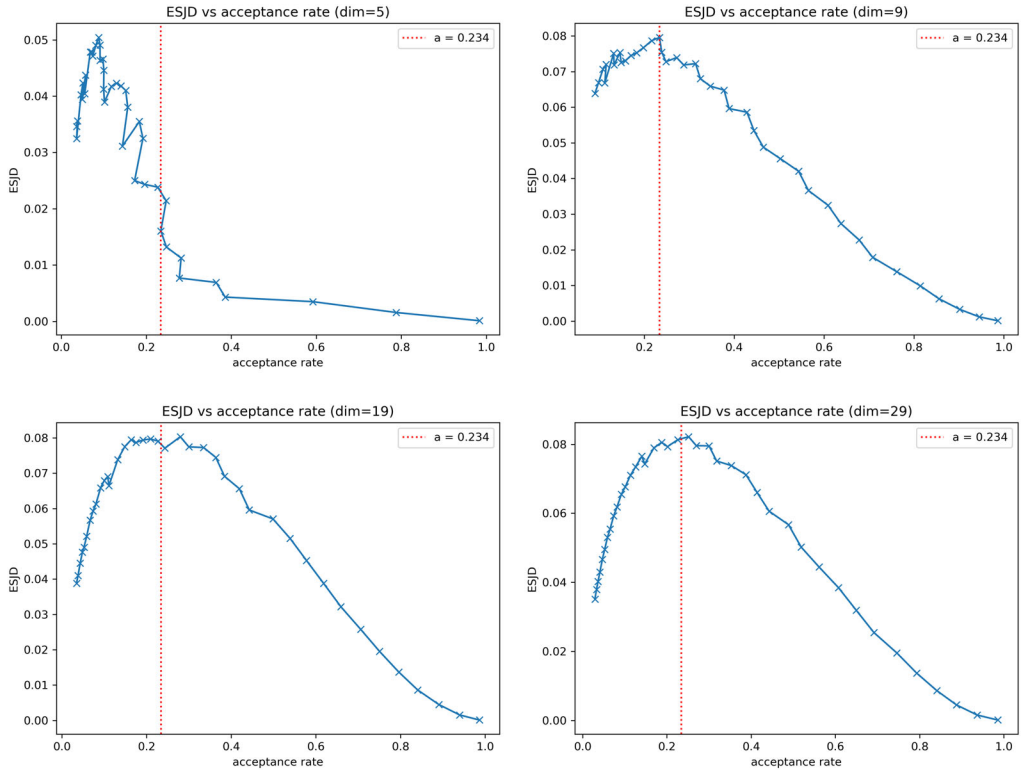
**Figure A7.** ESJD vs. acceptance rate for the inhomogeneously scaled three mixture target distribution  $\pi_7$  (Equation 14) under RWM with a Gaussian proposal in dimensions  $d \in \{20, 30, 50\}$  from left to right. Red dotted line indicates an acceptance rate of 0.234.



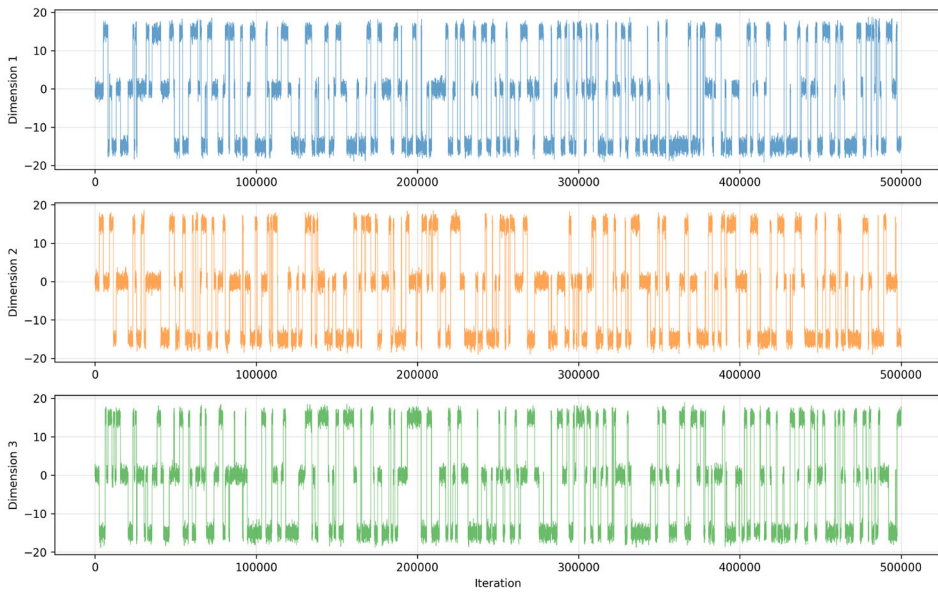
**Figure A8.** ESJD vs. acceptance rate for the Full Rosenbrock target  $\pi_g$  (Equation 16) under RWM with a Gaussian proposal in dimensions  $d \in \{5, 10, 20, 30\}$  from top-left to bottom-right. Red dotted line indicates an acceptance rate of 0.234.



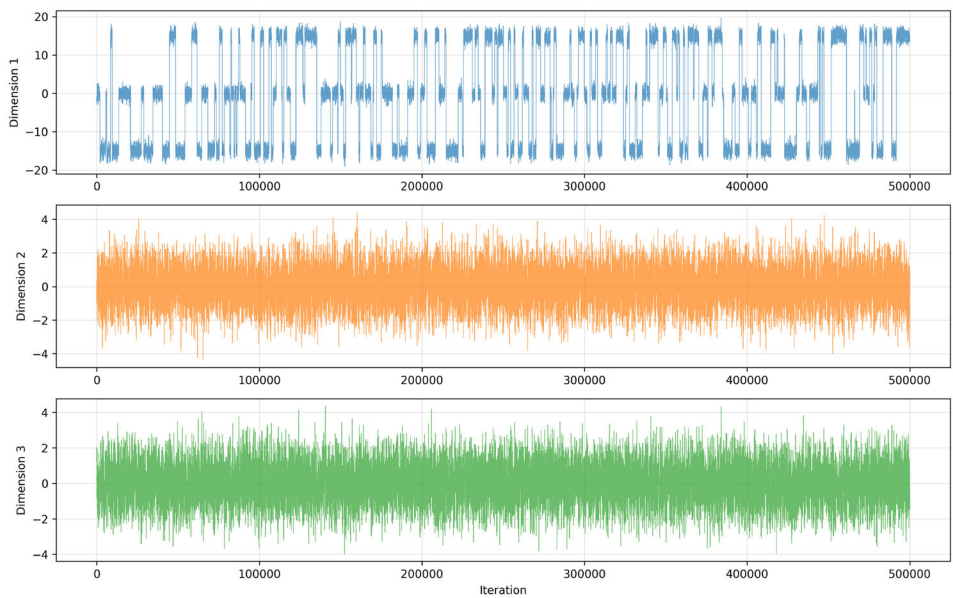
**Figure A9.** ESJD vs. acceptance rate for the Even Rosenbrock target  $\pi_9$  (Equation 17) under RWM with a Gaussian proposal in dimensions  $d \in \{5, 10, 20, 30\}$  from top-left to bottom-right. Red dotted line indicates an acceptance rate of 0.234.



**Figure A10.** ESJD vs. acceptance rate for the Hybrid Rosenbrock target  $\pi_{10}$  (Equation 18) under RWM with a Gaussian proposal in dimensions  $d \in \{5, 9, 19, 29\}$  from top-left to bottom-right. Red dotted line indicates an acceptance rate of 0.234.



**Figure A11.** Traceplot of the first three dimensions of the cold chain from one of the parallel tempering simulations on the rough carpet target distribution  $\pi_4$  (Equation 12).



**Figure A12.** Traceplot of the first three dimensions of the cold chain from one of the parallel tempering simulations on the three-mixture target distribution  $\pi_6$  (Equation 14).