Sampling from Multiscale Densities with Delayed Rejection Generalized Hamiltonian Monte Carlo

Gilad Turok
Flatiron Institute

Chirag Modi
Flatiron Institute, New York University

Bob Carpenter
Flatiron Institute

Abstract

Hamiltonian Monte Carlo (HMC) is the mainstay of applied Bayesian inference for differentiable models. However, HMC still struggles to sample from hierarchical models that induce densities with multiscale geometry: a large step size is needed to efficiently explore low curvature regions while a small step size is needed to accurately explore high curvature regions. We introduce the delayed rejection generalized HMC (DR-G-HMC) sampler that overcomes this challenge by employing dynamic step size selection, inspired by differential equation solvers. In generalized HMC, each iteration does a single leapfrog step. DR-G-HMC sequentially makes proposals with geometrically decreasing step sizes upon rejection of earlier proposals. This simulates Hamiltonian dynamics that can adjust its step size along a (stochastic) Hamiltonian trajectory to deal with regions of high curvature. DR-G-HMC makes generalized HMC competitive by decreasing the number of rejections which otherwise cause inefficient backtracking and prevents directed movement. We present experiments to demonstrate that DR-G-HMC (1) correctly samples from multiscale densities, (2) makes generalized HMC methods competitive with the state of the art No-U-Turn sampler, and (3) is robust to tuning parameters.

1 INTRODUCTION

Despite the success of Hamiltonian Monte Carlo (HMC) (Duane et al. 1987; Neal 2011; Betancourt 2017) and its auto-tuned variants like the No-U-Turn Sampler (NUTS) (Hoffman and Gelman 2014), these algorithms struggle to sample from densities with multiscale geometry. Such densities are characterized by varying curvature in which scales can differ by orders of magnitude throughout the space. Multiscale geometry is challenging because we need small HMC step sizes in high-curvature regions for numerical stability, whereas we require large step sizes in flat regions for efficient sampling. This pathology commonly arises in hierarchical models (Betancourt and Girolami 2015; Pourzanjani and Petzold 2019) in which small changes to a top-level parameter may induce large changes to parameters lower in the hierarchy. Neal's funnel is a density that exemplifies this multiscale behavior (see Figure 1a). Neal's funnel contains a neck region, with high density and small volume, that opens to a mouth region, with low density and large volume. HMC methods with a fixed step size, like NUTS, will struggle to sample from such densities.

Modi, Barnett, and Carpenter (2023) proposed delayed rejection HMC (DR-HMC) to sample from multiscale densities. Instead of falling back to the original state upon rejection, delayed rejection methods make additional proposals in the *same* iteration using a *different* proposal kernel. If chosen appropriately, this new proposal kernel can lead to a better chance of acceptance. Motivated with time-step adaptive integrators for differential equations, DR-HMC makes proposals with increasingly smaller leapfrog step sizes, resulting in *dynamic*, per-iteration step size selection. With these proposal kernels, DR-HMC can simulate more accurate Hamiltonian dynamics in high curvature regions, generating proposed states that are more likely to be accepted.

Although DR-HMC can successfully sample from multiscale densities, it introduces a few inefficiencies. Particularly, DR-HMC uses *small* step sizes to move across the entirety of *long* HMC trajectories, even if the small step size is necessary only for a short segment of the full trajectory. This makes DR-HMC iterations expensive, unable to compete with state of the art sam-

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

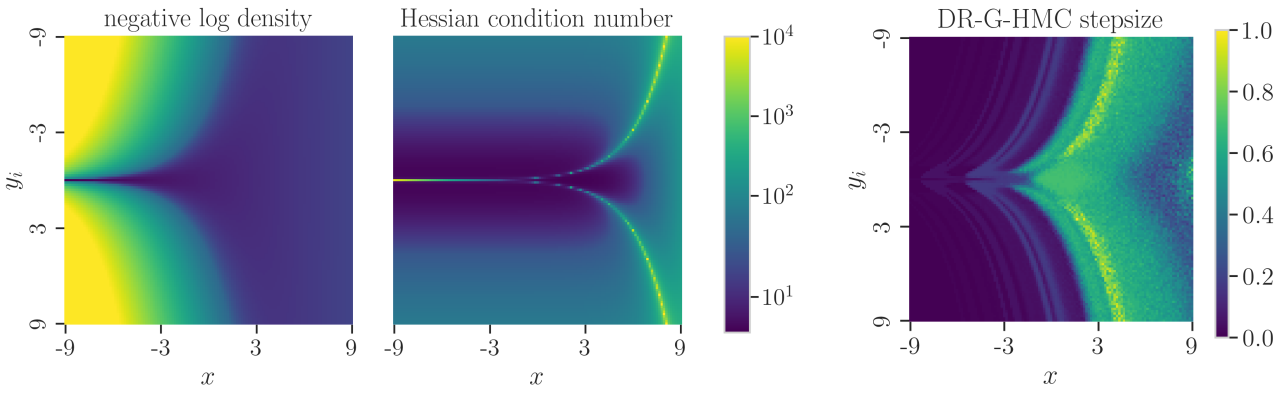


Figure 1: **(a) Neal's funnel exhibits multiscale geometry.** Neal's funnel is challenging to sample from because its (negative) log density and Hessian condition number vary by orders of magnitude throughout the space. **(b) DR-G-HMC handles multiscale geometry with dynamic step sizes.** To sample from Neal's funnel, DR-G-HMC uses large step sizes in low-curvature regions ($x \gg 0$) and small step sizes in high curvature regions ($x \ll 0$). See subsection B.1 for figure details.

plers like NUTS. To remedy this issue, we introduce the delayed rejection generalized HMC (DR-G-HMC) sampler.

Generalized HMC (G-HMC) (Horowitz 1991) partially resamples the momentum variable and (typically) simulates Hamiltonian dynamics for a single leapfrog step. Unlike HMC, G-HMC poses a Metropolis accept/reject step after every leapfrog step which avoids long and expensive trajectories that may ultimately be rejected. Hence a delayed proposal in DR-G-HMC is made with a smaller step size only for the single leapfrog step *where it is necessary*, unlike DR-HMC, which uses a smaller step size for the entire trajectory.

DR-G-HMC not only improves the computational efficiency of DR-HMC in robustly sampling multiscale distributions, it also overcomes longstanding inefficiencies in G-HMC. To maintain detailed balance in G-HMC, the momentum is flipped (negated) at the end of a sampling iteration. As a result, if a proposed state is rejected, a G-HMC chain reverses course. This results in less directed movement and increased autocorrelation for G-HMC samples, making G-HMC less efficient than HMC in practice. These rejections can be minimized by using an extremely small step size to more accurately simulate Hamiltonian dynamics, but this reduces the movement per-iteration and remains inefficient. DR-G-HMC provides an alternate way to minimize these rejections by making delayed proposals that increase the chances of acceptance. This minimizes the number of trajectory reversals while simultaneously being able to use large step sizes for the first proposals, making G-HMC competitive with state of the art samplers like NUTS.

Our primary contributions are as follows:

- We introduce the DR-G-HMC sampler that combines delayed rejection and generalized HMC while satisfying detailed balance.
- We demonstrate that DR-G-HMC robustly samples multiscale densities induced by hierarchical models, where it outperforms previous samplers like DR-HMC and NUTS.
- We show that DR-G-HMC solves the directed motion problem of generalized HMC, making it competitive with state of the art samplers like NUTS in sampling from densities without complex multiscale geometry.
- We present experiments to show that the performance of DR-G-HMC is insensitive to tuning different hyperparameters of the algorithm.

2 RELATED WORK

2.1 Generalized HMC

G-HMC, introduced by Horowitz 1991, is less efficient than HMC due to frequent reversals upon rejections in the Markov chain Monte Carlo chain. Neal (2020) proposed a scheme to cluster the acceptances together, allowing directed movement and hence higher efficiency. This method was later auto-tuned by Hoffman and Sountsov (2022) using many parallel chains. DR-G-HMC is an alternative approach to Neal's for fixing the directed motion problem of G-HMC, with the added benefits of multiscale sampling and insensitivity to tuning parameters.

2.2 Delayed Rejection

Delayed rejection methods have been studied for Metropolis-Hastings algorithms (Tierney and Mira 1999; Green and Mira 2001; Haario et al. 2006; Bédard, Douc, and Moulines 2014) but have only recently been adapted to HMC samplers. Modi, Barnett, and Carpenter (2023) apply delayed rejection to HMC in the DR-HMC sampler to robustly sample from multiscale densities. Although DR-HMC and DR-G-HMC share similarities, DR-G-HMC makes notable efficiency gains by using a smaller step size only where necessary. Sohl-Dickstein, Mudigonda, and DeWeese (2014) and Campos and Sanz-Serna (2015) present a different flavor of delayed rejection with the goal of increasing the autocorrelation length in G-HMC. These samplers *continue* the trajectory forward upon rejection with the same proposal kernel, which will fail if rejection is due to change in curvature (i.e., a diverging Hamiltonian). In contrast, we apply delayed rejection to G-HMC by *retrying* from the same starting point with a different proposal kernel. This key difference allows our method to sample from multiscale densities.

2.3 Step-size Selection

Several approaches have been proposed for step-size selection in Markov chain Monte Carlo samplers (theoretical: Atchadé 2006; Marshall and G. Roberts 2012; Gelman, Gilks, and G. O. Roberts 1997, adaptive: Hoffman and Gelman 2014, empirical: Coullon, South, and Nemeth 2023; Campbell et al. 2021). These approaches all struggle with multiscale densities because they fix a *single* step size. Other work selects *dynamic* step sizes that adapt to local curvature at every sampling iteration. Previous approaches include generalizing from Euclidean space to Riemmanian manifolds to explicitly deal with curvature (Girolami and Calderhead 2011; Betancourt and Girolami 2015) and applying implicit symplectic integrators that can adapt to the stiffness induced by multiscale densities (Pourzanjani and Petzold 2019; Brofos and Lederman 2021).[1] However both methods are computationally expensive, require implicit integrators that are hard to tune, and the former requires a Riemannian (i.e., positive definite) metric such as Fisher information. Cheaper dynamic step size selection is performed in Kleppe (2016) by bounding the error in the Hamiltonian and in Biron-Lattes et al. (2024) by sampling from a step-size distribution that maintains reversibility.

[1]Diagonal metrics are equivalent to a per-dimension step size; dense metrics account for curvature.

3 BACKGROUND

We seek to generate samples from a differentiable and potentially unnormalized probability density function $\pi(\theta)$ with parameters $\theta \in \mathbb{R}^D$. The target density π is often a Bayesian posterior.

3.1 Hamiltonian Monte Carlo

HMC generates samples by treating θ as position vector and introducing an auxiliary momentum vector $\rho \in \mathbb{R}^D$ for a fictitious particle with mass matrix M. The position and momentum define a Hamiltonian

$$H(\theta, \rho) = -\log \pi(\theta) + \frac{1}{2}\rho^T M^{-1} \rho$$

with corresponding Gibbs density

$$\begin{aligned} \tilde{\pi}(\theta, \rho) &\propto \exp(-H(\theta, \rho)) \\ &= \pi(\theta) \cdot \text{normal}(\rho \mid 0, M), \end{aligned} \tag{1}$$

for some symmetric, positive-definite mass matrix M. HMC generates samples (θ, ρ) in this augmented state space as a Metropolis-within-Gibbs sampler: a Gibbs step updates the momentum ρ and a Metropolis-Hastings step updates the position θ. From samples (θ, ρ) we can easily recover samples θ from the target density π by extracting the first D coordinates.

To generate the next state from the current state (θ, ρ), first Gibbs resample the momentum, $\rho' \sim \text{normal}(0, M)$. Then perform the Metropolis update as follows: generate a proposed state by simulating Hamiltonian dynamics, approximated with leapfrog integration, then flip (negate) the momentum. To do so, define the proposal map $F = PL_\epsilon^n$ as the composition of a momentum flip $P(\theta, \rho) = (\theta, -\rho)$ and n leapfrog steps L_ϵ^n with step size ϵ. The proposed state $(\theta^{\text{pr}}, \rho^{\text{pr}}) = F(\theta, \rho')$ is accepted with probability

$$\alpha = \min\left(1, \frac{\pi(\theta^{\text{pr}}) \cdot \text{normal}(\rho^{\text{pr}} \mid 0, M)}{\pi(\theta) \cdot \text{normal}(\rho' \mid 0, M)}\right),$$

otherwise we remain at the current state (θ, ρ').

3.2 Generalized Hamiltonian Monte Carlo

Instead of completely resampling the momentum as $\rho' \sim \text{normal}(0, M)$, G-HMC partially updates the momentum as $\rho' \sim \text{normal}(\rho\sqrt{1-\gamma}, \gamma M)$ for some damping parameter $\gamma \in (0, 1]$ (Horowitz 1991). The damping γ controls how much the momentum changes between iterations, while leaving the distribution over $\text{normal}(0, M)$ invariant. Next, G-HMC generates a proposed state $(\theta^{\text{pr}}, \rho^{\text{pr}})$ with proposal map F and accepts it with some probability α. Unlike HMC, most G-HMC algorithms use one leapfrog step per iteration

$(n = 1)$, with partial momentum refreshment promoting directed movement. For the rest of this work, we will fix $n = 1$ as it allows for immediate feedback when exploring a density. Lastly at the end of a G-HMC iteration the momentum is unconditionally negated. If the proposed state $(\theta^{\text{pr}}, \rho^{\text{pr}})$ is accepted, this undoes the negation from the proposal map F; if rejected, this negation makes the chain reverse direction. Thus G-HMC struggles to make directed motion because of inefficient reversals upon rejection. Standard HMC is unaffected by momentum negations because the momentum ρ is completely resampled at every iteration.

4 DELAYED REJECTION GENERALIZED HAMILTONIAN MONTE CARLO

DR-G-HMC applies delayed rejection to G-HMC to allow smaller step sizes to be used if a larger step size proposal is rejected during a single sampling iteration. The pseudocode is provided in Algorithm 1 and is briefly described here. More technical details, the proof of invariance over π, and the derivation of acceptance probability are presented in Appendix A.

Every iteration of DR-G-HMC begins with a Gibbs update by partially updating the momentum as $\rho' \sim \text{normal}(\rho\sqrt{1-\gamma}, \gamma M)$, same as G-HMC. Then, the sampler can make up to a maximum of K Metropolis-Hastings updates where each subsequent proposal is made only if the previous one is rejected.

For proposal k, we define a proposal map $F_k = PL_{\epsilon_k}^{n_k}$ with its own leapfrog step size ϵ_k and set the number of steps $n_k = 1$, following G-HMC. The acceptance probability α_k of the kth proposed state $(\theta^{\text{pr}}, \rho^{\text{pr}}) = F_k(\theta, \rho')$ is evaluated to account for rejecting the previous $k-1$ proposals and is discussed below. If accepted, $(\theta^{\text{pr}}, \rho^{\text{pr}})$ becomes the next state of the Markov chain. Otherwise we delay rejection and make another proposal. If a maximum of K proposals are made, we finally reject and set (θ, ρ') as the next state of the Markov chain. Lastly, the momentum is flipped regardless of whether any of the K proposals are accepted or rejected in order to maintain reversibility.

To maintain detailed balance, the acceptance probability α_k for accepting or rejecting the kth proposed state $(\theta^{\text{pr}}, \rho^{\text{pr}}) = F_k(\theta, \rho')$ is given as

$$\begin{aligned}
&\alpha_k(\theta, \rho', F_k(\theta, \rho')) \\
&\quad = \min\left(1, \frac{\pi(\theta^{\text{pr}}) \cdot \text{normal}(\rho^{\text{pr}} \mid 0, M)}{\pi(\theta) \cdot \text{normal}(\rho' \mid 0, M)}\right. \\
&\qquad \left. \times \prod_{i=1}^{k-1} \frac{1 - \alpha_i\big(\theta^{\text{pr}}, \rho^{\text{pr}}, F_i(\theta^{\text{pr}}, \rho^{\text{pr}})\big)}{1 - \alpha_i\big(\theta, \rho', F_i(\theta, \rho')\big)}\right). \quad (2)
\end{aligned}$$

The full derivation of this acceptance probability is given in subsection A.3, but we give a brief intuition here. The additional factor in α_k over the standard HMC acceptance probability is the product over k terms. The denominator in this product is the probability of rejecting the previous $k-1$ proposals originating from the current state $(\theta^{\text{rej}}, \rho^{\text{rej}}) = F_i(\theta, \rho')$. These terms have already been computed in previous proposals and are stored in a stack data structure, so they do not require additional computation. To maintain detailed balance, we need to balance it against the similar probability of rejecting the first $k-1$ proposals originating from the proposed state $(\theta^{\text{gh}}, \rho^{\text{gh}}) = F_i(\theta^{\text{pr}}, \rho^{\text{pr}})$. These are the terms in the numerator and require additional computation for every additional proposal. We call these "ghost states" and write $(\theta^{\text{gh}}, \rho^{\text{gh}})$, because they are the i-th proposal that would have been made and rejected in a hypothetical chain that had *started* from proposed state $(\theta^{\text{pr}}, \rho^{\text{pr}})$ and gone to the current state (θ, ρ').

The number of ghost states to be evaluated grows exponentially as $\mathcal{O}(2^k)$ for the kth proposal. However these delayed proposals to higher orders are (a) typically made only in the difficult regions of the phase space, and (b) proportional to the number of leapfrog steps required by to traverse the region of high curvature. Furthermore, these delayed proposals minimize the number of rejections which otherwise cause G-HMC trajectory to reverse due to the flipped momentum. As a result, the extra computational cost of delayed proposals is a favorable tradeoff and keeps DR-G-HMC computationally competitive with auto-tuned samplers like NUTS, even when step size reductions are not necessary.

If leapfrog step sizes $\epsilon_1 \ldots \epsilon_K$ are monotonically decreasing, then each subsequent proposal simulates more accurate Hamiltonian dynamics with a higher acceptance probability. This choice is *especially* well suited to multiscale densities because it allows for dynamic, per-iteration step size selection: large step sizes are used in the wide, mouth-like regions while smaller step sizes are used in narrow, neck-like regions. This can be seen in Figure 1b. While the step size for each proposal can be chosen independently, in this work we choose a geometric progression for step size by reducing them with a factor r at each stage: $\epsilon_k = \epsilon / r^{k-1}$, thereby reducing the number of tuning parameters.

Finally, unlike DR-HMC which retries subsequent proposals after rejecting a state n_k leapfrog steps away from the current state, DR-G-HMC retries proposals after *every* leapfrog step. This increases DR-G-HMC efficiency by (1) selecting step sizes suited to the local curvature at *every* point in a trajectory and (2) avoiding throwing away long trajectories upon rejection.

Algorithm 1 Delayed rejection generalized Hamiltonian Monte Carlo sampler

Globals: target density π, max proposals K, damping γ, initial step size ϵ, reduction factor r, mass matrix M

function SAMPLE(θ, ρ)
  $\rho' \sim \text{normal}(\rho\sqrt{1-\gamma}, \gamma M)$
  for k in $1 \dots K$ **do**
    $\theta^{\text{pr}}, \rho^{\text{pr}} \leftarrow$ PROPOSE(θ, ρ', k)
    $\alpha \leftarrow$ ACCEPT($\theta, \rho', \theta^{\text{pr}}, \rho^{\text{pr}}, k$)
    $u \sim \text{uniform}(0, 1)$
    if $u < \alpha$ **then**
      return FLIP($\theta^{\text{pr}}, \rho^{\text{pr}}$)
  return FLIP(θ, ρ')

function PROPOSE(θ, ρ, k)
  $\epsilon_k \leftarrow \epsilon / r^{k-1}$
  $\theta, \rho \leftarrow$ LEAPFROG(θ, ρ, ϵ_k)
  return FLIP(θ, ρ)

function FLIP(θ, ρ)
  return $\theta, -\rho$

function LEAPFROG(θ, ρ, ϵ)
  $\rho' \leftarrow \rho + \frac{\epsilon}{2} \nabla \log \pi(\theta)$
  $\theta' \leftarrow \theta + \epsilon M^{-1} \rho'$
  $\rho'' \leftarrow \rho' + \frac{\epsilon}{2} \nabla \log \pi(\theta')$
  return θ', ρ''

function ACCEPT($\theta, \rho', \theta^{\text{pr}}, \rho^{\text{pr}}, k$)
  $\alpha \leftarrow \dfrac{\pi(\theta^{\text{pr}}) \cdot \text{normal}(\rho^{\text{pr}} \mid 0, M)}{\pi(\theta) \cdot \text{normal}(\rho' \mid 0, M)}$
  for i in $1 \dots k-1$ **do**
    $\theta^{\text{gh}}, \rho^{\text{gh}} \leftarrow$ PROPOSE($\theta^{\text{pr}}, \rho^{\text{pr}}, i$)
    $\theta^{\text{rej}}, \rho^{\text{rej}} \leftarrow$ PROPOSE(θ, ρ', i) ▷ cached
    $\alpha \leftarrow \alpha \cdot \dfrac{1 - \text{ACCEPT}(\theta^{\text{pr}}, \rho^{\text{pr}}, \theta^{\text{gh}}, \rho^{\text{gh}}, i)}{1 - \text{ACCEPT}(\theta, \rho', \theta^{\text{rej}}, \rho^{\text{rej}}, i)}$
  return $\min(1, \alpha)$

5 EXPERIMENTS

In this section we compare the performance of DR-G-HMC, DR-HMC, and NUTS samplers. Experimental code can be found at https://github.com/gil2rok/drghmc.

5.1 Setup

Measuring Sampler Performance We evaluate a sampler's performance by measuring absolute standardized error for a fixed computational budget. Since computational cost is dominated by gradient evaluations, we run all samplers with a budget of $T = 10^6$ gradient evaluations. We do not use effective sample size (ESS) to measure relaxation because when a chain does not fully explore a challenging target density (e.g. in multiscale densities), the resulting bias is not captured by ESS. In contrast, the absolute standardized error between a sampler's draws and reference draws will detect this bias. For synthetic targets like Neal's funnel, we can generate reference samples independently from the target density. For other problems, we generates reference samples by running NUTS with target acceptance rate of 0.95, generating 10^6 samples, and thinning until roughly independent (as measured by ESS).

Given draws $\theta^{(1)}, \dots, \theta^{(N)} \in \mathbb{R}^D$, define a distribution

$$\text{U}(\theta \mid \theta_1, \dots, \theta_N) = \textstyle\sum_{n=1}^{N} \frac{1}{N} \text{I}(\theta = \theta_n).$$

For a function $f : \mathbb{R}^D \to \mathbb{R}$, expectations are given by

$$\mathbb{E}_{\text{U}(\theta \mid \theta_1, \dots, \theta_N)}[f(\theta)] = \tfrac{1}{N} \textstyle\sum_{n=1}^{N} f(\theta_n).$$

Given reference draws $\theta^{\text{ref}(1)}, \dots, \theta^{\text{ref}(N)}$ and sample draws $\theta^{\text{test}(1)}, \dots, \theta^{\text{test}(M)}$ to test generated with a budget of t gradient evaluations, let $p^{\text{ref}}(\theta) = \text{uniform}(\theta \mid \theta^{\text{ref}(1)}, \dots, \theta^{\text{ref}(N)})$ and p^{test} be defined similarly. For a given expectation function f, we define our reference answer as $\mathbb{E}_{p^{\text{ref}}}[f(\theta)]$ and our test answer as $\mathbb{E}_{p^{\text{test}}}[f(\theta)]$ and estimate the *error* of the test draws as $\mathbb{E}_{p^{\text{test}}}[f(\theta)] - \mathbb{E}_{p^{\text{ref}}}[f(\theta)]$.

To report a single statistic for every problem, we standardize the errors to the same scale by transforming to absolute Z score, and take the maximum across all dimensions to identify the worst performing parameter,

$$\mathcal{L}_{f,t}(\theta^{\text{test}}, \theta^{\text{ref}}) = \max_d \frac{\left|\mathbb{E}_{p^{\text{test}}}[f(\theta)] - \mathbb{E}_{p^{\text{ref}}}[f(\theta)]\right|}{\text{sd}_{p^{\text{test}}}[f(\theta)]}, \quad (3)$$

where sd is standard deviation. We compute the absolute standardized error $\mathcal{L}_{f,t}$ for functions $f(\theta) = \theta_d$ and $f(\theta) = \theta_d^2$ to measure the error in the mean and variance estimates of each parameter.

Choice of Parameters We generate parameter draws from NUTS, DR-HMC, and DR-G-HMC samplers, each with 100 chains. All chains are initialized from a randomly chosen sample from the reference samples, and the corresponding chains in all samplers are initialized from the same sample. Momentum is randomly initialized by sampling from $\text{normal}(0, M)$.

We run NUTS with the probabilistic programming language Stan (Carpenter et al. 2017) using the default target acceptance of 0.80, and a diagonal mass matrix

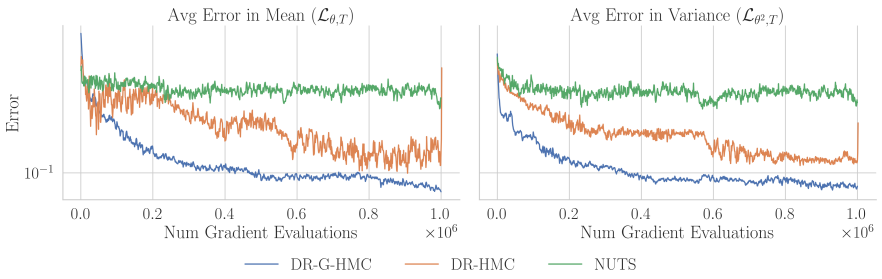


Figure 2: **Average error vs gradient evaluations for Neal's funnel.** Error in mean ($\mathcal{L}_{\theta,T}$) and variance ($\mathcal{L}_{\theta^2,T}$) averaged over 100 chains of sampler draws from $10D$ Neal's funnel. NUTS's error plateaus while delayed rejection methods do not. Our method DR-G-HMC achieves the lowest error.

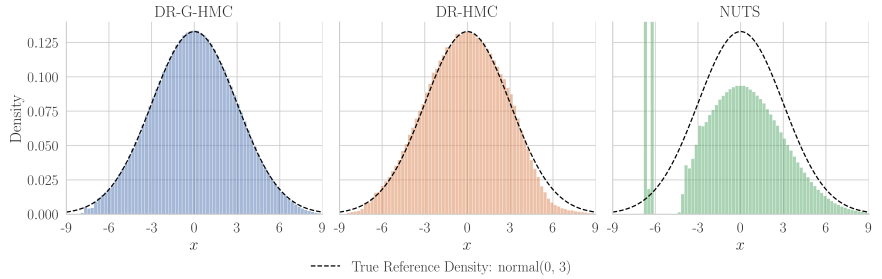


Figure 3: **Histogram of log scale parameter x in Neal's funnel.** Sampler draws are aggregated across all 100 chains of $10D$ Neal's funnel. Reference draws are sampled from the known density as $x \sim \text{normal}(0, 3)$. DR-G-HMC and DR-HMC can sample deep into the highly curved neck ($x \ll 0$) and DR-G-HMC can sample deep into the mouth ($x \gg 0$) with dynamic step size selection, while NUTS cannot.

M to be adapted for all problems except Neal's funnel, where we found the identity matrix to perform better. We implement DR-HMC and DR-G-HMC ourselves and access Stan models with BridgeStan (Roualdes et al. 2023). All experiments are run in parallel on 128 CPU cores with 256GB of memory.

Since DR-HMC and DR-G-HMC adapt step size dynamically as necessary, we run these samplers with an initial step size a factor c larger than the NUTS adapted step size, $\epsilon = c\epsilon_{\text{NUTS}}$, for $c = 2$. Both algorithms use the same mass-matrix as NUTS to ensure fair comparison between algorithms. We keep the maximum proposals of $K = 3$ and reduction factor of the step size, $\epsilon_k = \epsilon_1 / r^{k-1}$, to be $r = 4$.

Unlike NUTS, these algorithms do not adapt trajectory length automatically and hence this parameter needs to be chosen. DR-HMC uses the initial number of steps n as the 90th percentile of number of leapfrog steps from NUTS (following Wu, Stoehr, and Robert (2018)).[2] For the kth proposal, the number of steps are given by and $n_k = \tau / \epsilon_k$ such that the trajectory length $\tau = \epsilon_1 n$ is maintained constant. For DR-G-HMC, we keep the $n_k = 1$ for all proposals (including the first one) and damping $\gamma = 0.08$.

[2]Unlike Wu, Stoehr, and Robert (2018), we do not jitter step size.

5.2 Sampling From Multiscale Densities

We begin by evaluating the sampler performance on Neal's funnel, a synthetic multiscale density introduced in Neal (2003). A D-dimensional funnel defines the density with parameters $x \in \mathbb{R}, y \in \mathbb{R}^{D-1}$, as

$$\pi(x, y) = \text{normal}(x \mid 0, 3) \prod_{i=1}^{D-1} \text{normal}(y_i \mid 0, \exp(x/2)).$$

The funnel is extremely difficult to sample because its curvature, scale, and condition number vary dramatically throughout the density. This arises from the log scale parameter x that determines the spread of latent parameters $y_i \sim \text{normal}(0, \exp(x/2))$. Values of $x < 0$ lead to high curvature in y (the neck) while values of $x > 0$ lead to very low curvature in y (the mouth). Both regions are poorly conditioned and it is a challenge to sample *both* with a fixed step size.

In Figure 2 we examine the error across gradient evaluations for both the mean ($\mathcal{L}_{\theta,T}$) and variance ($\mathcal{L}_{\theta^2,T}$). NUTS cannot sample the funnel: it gets the wrong answer as its error plateaus quickly and does *not* decrease with hundreds of thousands of more iterations.

Target Posterior	Description	Dimension
eight schools	Centered parameterization, hierarchical model	9
normal100	Correlated dense covariance	100
irt 2pl	Item response theory with additive non-identifiability	144
stochastic volatility	High dimensional, correlated, and varying curvature	503

Table 1: **Summary of posterior densities.**

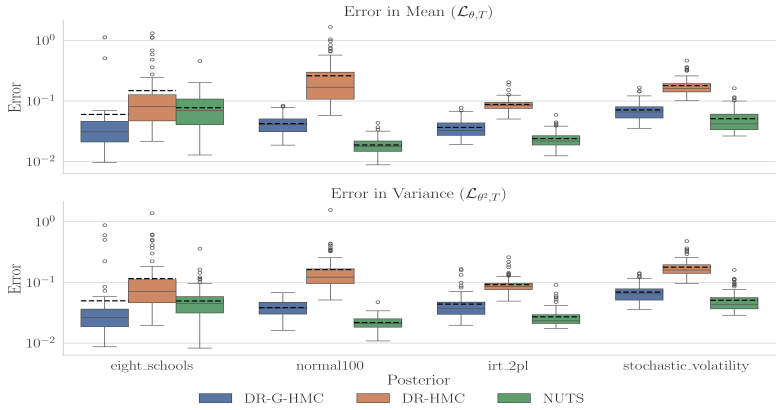


Figure 4: **DR-G-HMC overcomes the inefficiencies of generalized HMC.** Error in mean ($\mathcal{L}_{\theta,T}$) and variance ($\mathcal{L}_{\theta^2,T}$) is shown on the log scale for 100 chains. Visual elements represent the following: dashed black line is the mean, solid gray line is the median, colored box is the $(25, 75)$th percentile, whiskers are 1.5 times the inter-quartile range, and bubbles are outliers.

In contrast, delayed rejection methods reduce the error in the mean and variance *without* plateauing, with DR-G-HMC performing best. DR-G-HMC benefits from using a small step size only where necessary and not along entire HMC trajectories. In Figure 3 we perform further investigation by plotting the histogram of the (analytically available) log scale parameter $x \sim \text{normal}(0, 3)$ from our samplers. This confirms our diagnosis: the delayed rejection methods can sample deep into mouth ($x \gg 0$) and neck ($x \ll 0$) of the funnel, while NUTS is unable to sample past the $x = -5$ region with a fixed step size. DR-G-HMC and DR-HMC can robustly sample from Neal's funnel while NUTS will fail due to its multiscale geometry. Similar results are shown in Appendix B for Neal's funnel in 50, 100, and 250 dimensions.

5.3 Resolving the Inefficiencies of G-HMC

We examine a variety of real-world posterior densities summarized in Table 1 and detailed in Appendix C. We seek to demonstrate that in this setting DR-G-HMC alleviates the backtracking inefficiencies of vanilla G-HMC, making it competitive with NUTS.

We compare DR-G-HMC, DR-HMC, and NUTS samplers as they are of primary interest. However in Appendix D we benchmark a larger variety of samplers including those with and without delayed rejection, momentum refreshment, and multiple leapfrog steps.

We compute the error in mean ($\mathcal{L}_{\theta,T}$) and variance ($\mathcal{L}_{\theta^2,T}$) in Figure 4. DR-G-HMC achieves errors comparable to NUTS across a variety of posteriors while DR-HMC performs notably worse. DR-G-HMC especially outperforms NUTS on the eight schools posterior. Crucially, this demonstrates that delayed rejection removes the inefficiency due to reversing course that plagues G-HMC. This is especially impressive because NUTS extensively *auto-tunes* its parameters while DR-G-HMC does not. Furthermore, since DR-G-HMC outperforms DR-HMC on all posteriors by a significant margin, DR-G-HMC achieves substantial benefit from using a small step size only where necessary.

We emphasize that the main advantage of DR-G-HMC is that it can efficiently sample from multiscale densities (unlike NUTS) *and* from non-multiscale densities (unlike DR-HMC).

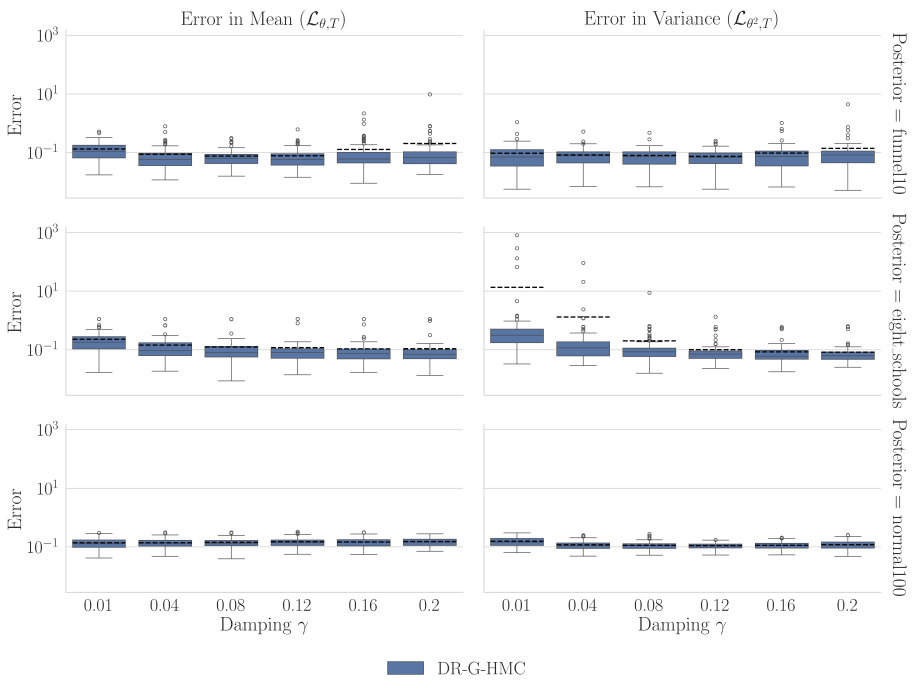


Figure 5: **DR-G-HMC is robust to the damping tuning parameter γ.** Error in mean ($\mathcal{L}_{\theta,T}$) and variance ($\mathcal{L}_{\theta^2,T}$) is shown for 100 chains of various posterior densities. Visual elements represent the following: dashed black line is the mean, solid gray line is the median, colored box is the $(25, 75)$th percentile, whiskers are 1.5 times the inter-quartile range, and bubbles are outliers.

5.4 Robust to Tuning Parameters

In this section we asses the sensitivity of DR-G-HMC to tuning parameters, especially the damping γ that determines directed motion for G-HMC samplers. HMC and its variants are notoriously sensitive to its trajectory length $\tau = n\epsilon$ that determines directed motion for HMC samplers. Before NUTS auto-tuned this parameter, HMC was incredibly difficult to use.

In Figure 5 we run DR-G-HMC on multiple posteriors for $T = 10^5$ gradient evaluations with damping values $\gamma = [0.01, 0.04, 0.08, 0.12, 0.16, 0.2]$. We measure error in mean ($\mathcal{L}_{\theta,T}$) and variance ($\mathcal{L}_{\theta^2,T}$) and find that DR-G-HMC errors are remarkably stable across a wide range of damping values and posteriors. This indicates DR-G-HMC is insensitive to the damping value γ and perhaps has an easier time making directed motion than HMC. Furthermore, in our box plot, we find the variance across DR-G-HMC chains is often larger than the variance between different damping values. This suggests chain initialization plays a larger role than the damping itself. Appendix E demonstrates robustness to additional DR-G-HMC tuning parameters. Appendix F compares wallclock runtime of DR-G-HMC and DR-HMC.

6 DISCUSSION

We have not addressed the open problem of automatically tuning the DR-G-HMC parameters. The damping factor γ is especially important because it determines how much momentum is conserved, analogous to trajectory length in HMC. In prior work, the MEADS algorithm (Hoffman and Sountsov 2022) auto-tuned the parameters for the non-reversible G-HMC slice sampling scheme of Neal (2003), including the damping factor γ. If similar work can be done, DR-G-HMC may see a performance increase similar to NUTS over HMC.

7 CONCLUSION

We introduced the DR-G-HMC sampler which applies delayed rejection to G-HMC to adapt to local curvature with dynamic step size selection. By combining the useful features of DR-HMC and G-HMC, our sampler *simultaneously* solves the directed motion problem of G-HMC and provides multiscale density sampling for hierarchical models. DR-G-HMC ultimately outperforms both its individual components, while being competitive with NUTS and robust to tuning parameters.

References

Atchadé, Yves F (2006). "An adaptive version for the Metropolis adjusted Langevin algorithm with a truncated drift". In: *Methodology and Computing in applied Probability* 8, pp. 235–254.

Bédard, Mylène, Randal Douc, and Eric Moulines (2014). "Scaling analysis of delayed rejection MCMC methods". In: *Methodology and Computing in Applied Probability* 16.4, pp. 811–838.

Betancourt, Michael (2017). "A conceptual introduction to Hamiltonian Monte Carlo". In: *arXiv preprint arXiv:1701.02434*.

Betancourt, Michael and Mark Girolami (2015). "Hamiltonian Monte Carlo for hierarchical models". In: *Current trends in Bayesian methodology with applications* 79.30, pp. 2–4.

Biron-Lattes, Miguel et al. (May 2024). "autoMALA: Locally adaptive Metropolis-adjusted Langevin algorithm". In: *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*. Ed. by Sanjoy Dasgupta, Stephan Mandt, and Yingzhen Li. Vol. 238. Proceedings of Machine Learning Research. PMLR, pp. 4600–4608. URL: https://proceedings.mlr.press/v238/biron-lattes24a.html.

Brofos, James and Roy R Lederman (2021). "Evaluating the implicit midpoint integrator for Riemannian Hamiltonian Monte Carlo". In: *International Conference on Machine Learning*. PMLR, pp. 1072–1081.

Campbell, Andrew et al. (2021). "A gradient based strategy for Hamiltonian Monte Carlo hyperparameter optimization". In: *International Conference on Machine Learning*. PMLR, pp. 1238–1248.

Campos, Cédric M and Jesús María Sanz-Serna (2015). "Extra chance generalized hybrid Monte Carlo". In: *Journal of Computational Physics* 281, pp. 365–374.

Carpenter, Bob et al. (2017). "Stan: A probabilistic programming language". In: *Journal of statistical software* 76.

Coullon, Jeremie, Leah South, and Christopher Nemeth (2023). "Efficient and generalizable tuning strategies for stochastic gradient MCMC". In: *Statistics and Computing* 33.3, p. 66.

Duane, Simon et al. (1987). "Hybrid monte carlo". In: *Physics letters B* 195.2, pp. 216–222.

Gelman, Andrew, John B Carlin, et al. (2021). "Bayesian Data Analysis Third edition". In: *Issue: April*.

Gelman, Andrew, Walter R Gilks, and Gareth O Roberts (1997). "Weak convergence and optimal scaling of random walk Metropolis algorithms". In: *The annals of applied probability* 7.1, pp. 110–120.

Gelman, Andrew and Jennifer Hill (2006). *Data analysis using regression and multilevel/hierarchical models*. Cambridge university press.

Girolami, Mark and Ben Calderhead (2011). "Riemann manifold Langevin and Hamiltonian Monte Carlo methods". In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 73.2, pp. 123–214.

Green, Peter J and Antonietta Mira (2001). "Delayed rejection in reversible jump Metropolis–Hastings". In: *Biometrika* 88.4, pp. 1035–1053.

Haario, Heikki et al. (2006). "DRAM: efficient adaptive MCMC". In: *Statistics and computing* 16, pp. 339–354.

Hastings, W Keith (1970). "Monte Carlo sampling methods using Markov chains and their applications". In: *Biometrika* 57.1, pp. 97–97.

Hoffman, Matthew D and Andrew Gelman (2014). "The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo." In: *J. Mach. Learn. Res.* 15.1, pp. 1593–1623.

Hoffman, Matthew D and Pavel Sountsov (Mar. 2022). "Tuning-Free Generalized Hamiltonian Monte Carlo". In: *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*. Ed. by Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera. Vol. 151. Proceedings of Machine Learning Research. PMLR, pp. 7799–7813. URL: https://proceedings.mlr.press/v151/hoffman22a.html.

Horowitz, Alan M (1991). "A generalized guided Monte Carlo algorithm". In: *Physics Letters B* 268.2, pp. 247–252.

Kim, Sangjoon, Neil Shephard, and Siddhartha Chib (1998). "Stochastic volatility: likelihood inference and comparison with ARCH models". In: *The review of economic studies* 65.3, pp. 361–393.

Kleppe, Tore Selland (2016). "Adaptive Step Size Selection for Hessian-Based Manifold Langevin Samplers". In: *Scandinavian Journal of Statistics* 43.3, pp. 788–805.

Marshall, Tristan and Gareth Roberts (2012). "An adaptive approach to Langevin MCMC". In: *Statistics and Computing* 22, pp. 1041–1057.

Metropolis, Nicholas et al. (1953). "Equation of state calculations by fast computing machines". In: *The journal of chemical physics* 21.6, pp. 1087–1092.

Modi, Chirag, Alex Barnett, and Bob Carpenter (Jan. 2023). "Delayed rejection Hamiltonian Monte Carlo for sampling multiscale distributions". In: *Bayesian Analysis* 1.1. ISSN: 1936-0975. DOI: 10.1214/23-ba1360. URL: http://dx.doi.org/10.1214/23-BA1360.

Neal, Radford M (2003). "Slice sampling". In: *The annals of statistics* 31.3, pp. 705–767.

Neal, Radford M (2011). "MCMC using Hamiltonian dynamics". In: *Handbook of Markov Chain Monte Carlo*. Ed. by Steve Brooks et al. Chapman and Hall/CRC.

– (2020). *Non-reversibly updating a uniform [0,1] value for Metropolis accept/reject decisions*. arXiv: 2001.11950 [stat.CO].

Pourzanjani, Arya A and Linda R Petzold (2019). "Implicit Hamiltonian Monte Carlo for sampling multiscale distributions". In: *arXiv preprint arXiv:1911.05754*.

Roualdes, Edward A. et al. (2023). "BridgeStan: Efficient in-memory access to the methods of a Stan model". In: *Journal of Open Source Software* 8.87, p. 5236. DOI: 10.21105/joss.05236. URL: https://doi.org/10.21105/joss.05236.

Rubin, Donald B (1981). "Estimation in parallel randomized experiments". In: *Journal of Educational Statistics* 6.4, pp. 377–401.

Sohl-Dickstein, Jascha, Mayur Mudigonda, and Michael DeWeese (2014). "Hamiltonian Monte Carlo without detailed balance". In: *International Conference on Machine Learning*. PMLR, pp. 719–726.

Tierney, Luke and Antonietta Mira (1999). "Some adaptive Monte Carlo methods for Bayesian inference". In: *Statistics in medicine* 18.17-18, pp. 2507–2515.

Wu, Changye, Julien Stoehr, and Christian P Robert (2018). "Faster Hamiltonian Monte Carlo by learning leapfrog scale". In: *arXiv preprint arXiv:1810.04449*.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes] The DR-G-HMC algorithm is detailed in Algorithm 1.
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes] In DR-G-HMC, the number of model + gradient evaluations for the kth proposal attempt is $\mathcal{O}(2^k)$ detailed in section 4, along with implications of caching.
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes] Annonymized source is submitted.

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes] Limited assumptions are used in the paper, but they can be found in Appendix A.
   (b) Complete proofs of all theoretical results. [Yes] subsection A.3 for derivation of acceptance probability α and subsection A.2 for proof of detailed balance.
   (c) Clear explanations of any assumptions. [Yes] See above.

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes] Included in the annonymized source code.
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes] Detailed in subsection 5.1.
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes] Detailed in subsection 5.1.
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes] Detailed in subsection 5.1.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes] For major software used, I cite them, primarily Carpenter et al. (2017); Roualdes et al. (2023).
   (b) The license information of the assets, if applicable. [Yes] We checked relevant licenses for software used.
   (c) New assets either in the supplemental material or as a URL, if applicable. [Yes] We include annonymized source code upon submission.
   (d) Information about consent from data providers/curators. [Not Applicable] Not using any public datasests.
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable] No sensitive content used.

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

(a) The full text of instructions given to participants and screenshots. [Not Applicable] Research does not involve humans.

(b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable] Research does not involve humans.

(c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable] Research does not involve humans.

A DR-G-HMC TECHNICAL DETAILS

First we review technical background on Markov chain Monte Carlo, Metropolis-Hastings, and HMC in subsection A.1. Then we prove that DR-G-HMC maintains an invariance over the target density in subsection A.2. Finally we derive the acceptance probability of DR-G-HMC that maintains detailed balance in subsection A.3. We use notation that differs from the remainder of the paper in this appendix.

A.1 Background

A.1.1 Markov Chain Monte Carlo

Markov chain Monte Carlo (MCMC) methods seek to generate samples $x \in S$ from a probability density function (pdf) π that is absolutely continuous (AC) over some state space S. MCMC methods do so by constructing a Markov chain with π as its stationary distribution using a transition kernel $k(x, y)$ that gives the pdf of transitioning from the current state x to the next state y.

Definition 1 (Transition kernel normalization)**.** *The transition kernel k is normalized as*

$$\int k(x, y)dy = 1 \qquad \forall x \in S. \tag{4}$$

Definition 2 (Invariance)**.** *A Markov chain with stationary distribution π must be invariant to the transition kernel k as*

$$\int \pi(x)k(x, y)dx = \pi(y) \qquad \forall y \in S. \tag{5}$$

Definition 3 (Detailed balance)**.** *An AC transition kernel k satisfies detailed balance if*

$$\pi(x)k(x, y) = \pi(y)k(y, x). \tag{6}$$

A non-AC transition kernel k maintains detailed balance if

$$\int_A \int_B \pi(x)k(x, y)dxdy = \int_A \int_B \pi(y)k(y, x)dxdy \qquad \text{for all subsets } A, B \subset S \tag{7}$$

since the two sides of Equation 6 may not always be defined.

Proposition 1. *If an AC transition kernel maintains detailed balance, it satisfies π-invariance by substituting Equation 6 into Equation 5 and applying Equation 4. (Non-AC transition kernels also maintain π-invariance but the more complex proof is omitted here.)*

A.1.2 Metropolis-Hastings

The Metropolis-Hastings algorithm (Metropolis et al. 1953; Hastings 1970) defines an AC proposal kernel $q(x, y)$ that gives the pdf of proposing a new state y from the current state x. With some probability $\alpha(x, y)$ the proposal made with $q(x, y)$ becomes the next state of the Markov chain. Otherwise, the Markov chain remains at the current state x. The Metropolis-Hastings transition kernel is

$$k(x, y) = q(x, y)\alpha(x, y) + \delta_x(y)r(x)$$

in which one can transition to state y by (1) accepting a proposal to move from x to y (2) by rejecting a proposal to leave y. To show Metropolis-Hastings is π-invariant, we must pick the acceptance probability α that maintains detailed balance (Definition 3) as

$$\pi(x)\left(q(x, y)\alpha(x, y) + \delta_x(y)r(x)\right) = \pi(y)\left(q(y, x)\alpha(y, x) + \delta_y(x)r(y)\right)$$

for the AC transition kernel q. Because the second term of the transition kernel, $\delta_x(y)r(x)$, is $x \leftrightarrow y$ symmetric, we need only consider the first term and simplify to

$$\pi(x)q(x, y)\alpha(x, y) = \pi(y)q(y, x)\alpha(y, x).$$

The most efficient acceptance probability is then $\alpha(x, y) = \min\left(1, \frac{\pi(y)q(y,x)}{\pi(x)q(x,y)}\right)$ because for every $x, y \in S$, either $\alpha(x, y)$ or $\alpha(y, x)$ is equal to one.

A.2 DR-G-HMC is π-Invariant

In this section we prove that DR-G-HMC updates leave the target density π-invariant over the state space $S = \mathbb{R}^{2D}$.

Definition 4 (Volume-Preserving). *A map $F : \mathbb{R}^{2D} \to \mathbb{R}^{2D}$ is volume preserving if*

$$\int_B dx = \int_{F(B)} dx \qquad \text{for all subsets } B \subset \mathbb{R}^{2D}$$

where $F(B) := \{F(x) : x \in B\}$ denotes the image of the set B.

Definition 5 (Involution). *A map $F : \mathbb{R}^{2D} \to \mathbb{R}^{2D}$ is an involution if $F^{-1} = F$ as maps or equivalently $F^2 = I$ for identity map I.*

Definition 6 (Shear). *Any map on $\mathbb{R}^{2D}$ of the form $(\theta, \rho) \to (\theta + G(\rho), \rho)$ or $(\theta, \rho) \to (\theta, \rho + G(\theta))$ for differentiable map $G : \mathbb{R}^D \to \mathbb{R}^D$ is called a shear.*

Proposition 2. *Any shear is volume-preserving.*

Proof. The Jacobian of some shear F with $D \times D$ blocks is $DF = \begin{bmatrix} I_D & J_G \\ 0 & I_D \end{bmatrix}$ or $DF = \begin{bmatrix} I_D & 0 \\ J_G & I_D \end{bmatrix}$. In both cases $\det(DF) = 1$ implying F is volume-preserving. □

Lemma 1. *Consider the momentum flip map P and $n \in \mathbb{N}$ compositions of the leapfrog map L_ϵ with $\epsilon > 0$ defined in Algorithm 1. Then the map $F : \mathbb{R}^{2D} \to \mathbb{R}^{2D}$ defined as $F = PL_\epsilon^n$ is a volume-preserving involution.*

Proof. P is trivially volume-preserving and L_ϵ is the composition of three shears, each of which is volume-preserving by Proposition 2. Thus the composition $F = PL_\epsilon^n$ is volume preserving. L_ϵ is time-reversible such that if $L_\epsilon(\theta^{(t)}, \rho^{(t)}) = (\theta^{(t+1)}, \rho^{(t+1)})$, then $L_\epsilon(\theta^{(t+1)}, -\rho^{(t+1)}) = (\theta^{(t)}, -\rho^{(t)})$ by inspection of the three steps of the leapfrog map. Since this holds for any n, we know $PL_\epsilon^n = (PL_\epsilon^n)^{-1}$ and thus F is an involution. □

Lemma 2. *The Metropolis-Hastings algorithm with a deterministic proposal kernel $q_F(x, y) = \delta(y - F(x))$ and acceptance probability α obeying*

$$\pi(x)\alpha(x, y) = \pi(y)\alpha(y, x) \qquad \forall x, y \in \mathbb{R}^{2D} \tag{8}$$

maintains an invariance over a density π if the map F is a volume-preserving involution.

Proof. For Metropolis-Hastings updates to remain π-invariant, we must maintain detailed balance in the weak sense of Equation 7 as

$$\int_A \int_B \pi(x)\alpha(x, y)q_F(x, y)dxdy = \int_A \int_B \pi(y)\alpha(x, y)q_F(y, x)dxdy$$

for all subsets $A, B \in \mathbb{R}^D$. We show this is true by transforming the left hand side to the right hand side in a series of steps that require volume-preservation $F = F^{-1}$ and involution $\det(DF) = 1$:

$$\begin{aligned}
&\int_A \int_B \pi(x)\alpha(x, y)q_F(x, y)dxdy \\
&\quad = \int_A \int_B \pi(y)\alpha(y, x)\delta(y - F(x))dxdy \qquad \text{by substitution of Equation 8} \\
&\quad = \int_{B \bigcap F^{-1}(A)} \pi(F(x))\alpha(F(x), x)dx \qquad \text{by shifting property of the delta distribution} \\
&\quad = \int_{F(B) \bigcap A} \pi(y)\alpha(y, F^{-1}(y))\cdot \mid \det(DF(F^{-1}(y)) \mid^{-1} dy \qquad \text{by substituting } y = F(x) \\
&\quad = \int_{F^{-1}(B) \bigcup A} \pi(y)\alpha(y, F(y))dy \qquad \text{by } F^{-1} = F \text{ and unity Jacobian factor} \\
&\quad = \int_A \int_B \pi(y)\alpha(y, x)\delta(x - F(y))dxdy \qquad \text{by shifting property of the delta distribution}
\end{aligned}$$

□

Theorem 1 (DR-G-HMC is invariant over the Gibbs density $\tilde{\pi}$)**.** *Let π be a continuous, differentiable pdf over $\mathbb{R}^D$ with an associated Gibbs density $\tilde{\pi}$ over $\mathbb{R}^{2D}$ in Equation 1. The Markov chain with DR-G-HMC updates, given by the composition of a Gibbs step and up to K Metropolis-Hastings step, maintains the Gibbs density $\tilde{\pi}$ as an invariant density.*

Proof. The Gibbs step is $\tilde{\pi}$-invariant because it preserves the conditional over the momentum ρ while leaving the position θ unaltered. The Metropolis-Hastings steps are $\tilde{\pi}$-invariant by Lemma 2 because they use a deterministic proposal kernel with a volume-preserving involution $F = PL_\epsilon^n$. □

Corollary 1. *If DR-G-HMC is invariant over the Gibbs density $\tilde{\pi}$, then it is invariant over the target density π.*

A.3 DR-G-HMC Acceptance Probability

DR-G-HMC produces samples (θ, ρ) from the Gibbs density $\tilde{\pi}$ (Equation 1) by generating states from a $\tilde{\pi}$-invariant Markov chain. In the Metropolis-Hastings updates of DR-G-HMC, we make upto K proposals and accept it with some probability that maintains detailed balance (Definition 3). In this section we derive this acceptance probability, displayed in Equation 2.

We will build towards the acceptance probability of arbitrarily many proposal attempts by first deriving the acceptance probability for $K = 1, 2, 3$. For each proposal attempt, we define the proposal kernel q, then the transition kernel k, and finally the acceptance probability α.

To simplify notation, we represent states from the $\tilde{\pi}$-invariant Markov chain as $x = (\theta, \rho)$. Furthermore, although our transition kernel k will be non-AC, we will write detailed balance statements as Equation 6 with the understanding that they will be interpreted as the weak sense in Equation 7.

A.3.1 One Proposal Acceptance Probability

Let the $\tilde{\pi}$-invariant Markov chain have current state x and proposed state y. We define a non-AC, deterministic proposal kernel with the proposal map $F_1 = PL_{\epsilon_1}^{n_1}$ as

$$q_1(x, y) = \delta(y - F_1(x)).$$

The proposal kernel q_1 puts all its probability mass on $y = F_1(x)$ and zero everywhere else. This defines the transition kernel

$$k(x, y) = q_1(x, y)\alpha_1(x, y) + \delta_x(y)r_1(x)$$

with $x \leftrightarrow y$ symmetry in the rejection term $\delta_x(y)r_1(x)$ and the factor $q_1(x, y)$ (by F in Lemma 1). With these symmetries, maintaining detailed balance reduces to satisfying

$$\tilde{\pi}(x)\alpha_1(x, y) = \tilde{\pi}(y)\alpha_1(y, x).$$

We choose the acceptance probability to maintain detailed balance with the maximum chance of acceptance as

$$\alpha_1(x, y) = \min\left(1, \frac{\tilde{\pi}(y)}{\tilde{\pi}(x)}\right).$$

A.3.2 Two Proposals Acceptance Probability

Let the $\tilde{\pi}$-invariant Markov chain have current state x, rejected state s_1, and proposed state y. We define a non-AC, deterministic proposal kernel with the proposal map $F_2 = PL_{\epsilon_2}^{n_2}$ as

$$q_2(x, s_1, y) = \delta(y - F_2(x)).$$

This defines a transition kernel that must account for all the ways to get to state y: (1) accepting the first proposal $q_1(x, y)$ with probability $\alpha_1(x, y)$; (2) accepting the second proposal $q_2(x, s_1, y)$ with some new probability

$\alpha_2(x, y)$; (3) rejecting the second proposal. Cases (2) and (3) must integrate over all possible rejected proposals s_1, giving us

$$k(x,y) = q_1(x,y)\alpha_1(x,y) + \int q_1(x,s_1)[1-\alpha_1(x,s_1)][q_2(x,s_1,y)\alpha_2(x,y) + r_2(x)\delta_x(y)]ds_1$$

with probability of rejecting the second proposal as r_2. Since the first term already satisfies detailed balance and $r_2(x)\delta_x(y)$ is $x \leftrightarrow y$ symmetric, maintaining detailed balance reduces to plugging in deterministic proposal kernels q_1, q_2 and satisfying

$$\int \tilde{\pi}(x)q_1(x,s_1)[1-\alpha_1(x,s)]q_2(x,s_1,y)\alpha_2(x,s_1,y)ds_1 =$$
$$\int \tilde{\pi}(y)q_1(y,s_1')[1-\alpha_1(y,s_1')]q_2(y,s_1',x)\alpha_2(y,s_1',x)ds_1'$$

for dummy variables s_1, s_1'. To pick the acceptance probability α_2 that satisfies detailed balance, we plug in deterministic proposal kernels q_1, q_2 as

$$\int \tilde{\pi}(x)\delta(s_1 - F_1(x))[1-\alpha_1(x,s_1)]\delta(y-F_2(x))\alpha_2(x,s_1,y)ds_1 =$$
$$= \int \tilde{\pi}(y)\delta(s_1' - F_1(y))[1-\alpha_1(y,s_1')]\delta(x-F_2(y))\alpha_2(y,s_1',x)ds_1'.$$

Unlike the derivation in Tierney and Mira (1999), we *evaluate* this integral as

$$\tilde{\pi}(x)[1-\alpha_1(x,F_1(x))]\delta(y-F_2(x))\alpha_2(x,F_1(x),y) =$$
$$\tilde{\pi}(y)[1-\alpha_1(y,F_1(y))]\delta(x-F_2(y))\alpha_2(y,F_1(y),x).$$

Since $F_2 = F_2^{-1}$, the delta distributions are equal if $y = F_2(x)$, allowing us to simplify detailed balance to maintaining

$$\tilde{\pi}(x)[1-\alpha_1(x,F_1(x))]\alpha_2(x,F_1(x),y) = \tilde{\pi}(y)[1-\alpha_1(y,F_1(y))]\alpha_2(y,F_1(y),x).$$

To maximize the acceptance rate under $y = F_2(x)$, we set

$$\alpha_2(x,F_1(x),y) = \min\left(1, \frac{\tilde{\pi}(y)}{\tilde{\pi}(x)}\frac{1-\alpha_1(y,F_1(y))}{1-\alpha(x,F_1(x))}\right).$$

With access to proposal maps F_1, F_2, we can rewrite the acceptance probability from current state x to proposed state $y = F_2(x)$ as

$$\alpha_2(x,F_2(x)) = \min\left(1, \frac{\tilde{\pi}(F_2(x))}{\tilde{\pi}(x)}\frac{1-\alpha_1(F_2(x),F_1(F_2(x)))}{1 \;-\; \alpha(x,F_1(x))}\right).$$

This formulation highlights the ghost states that emerge from $F_1(F_2(x))$.

A.3.3 Three Proposal Acceptance Probability

Let the $\tilde{\pi}$-invariant Markov chain have current state x, first rejected state s_1, second rejected state s_2, and proposed state y. We define a non-AC, deterministic proposal kernel with the proposal map $F_3 = PL_{\epsilon_3}^{n_3}$ as

$$q_3(x,s_1,s_2,y) = \delta(y - F_3(x))$$

This defines a transition kernel that must account for all four ways to get to state y: (1) accepting the first proposal q_1 with probability α_1; (2) accepting the second proposal q_2 with probability α_2; (3) accepting the third proposal q_3 with some new probability α_3; (4) rejecting all proposals and remaining at the current state y. Case (2) must marginalize over all second proposals s_1 while cases (3) and (4) must marginalize over all second and third proposals s_1, s_2, giving us

$$\begin{aligned} k(x,y) &= q_1(x,y)\alpha_1(x,y) \\ &\quad + \int q_1(x,s_1)[1-\alpha_1(x,s_1)]q_2(x,s_1,y)\alpha_2(x,s_1,y)ds \\ &\quad + \int q_1(x,s_1)q_2(x,s_1,s_2)[1-\alpha_1(x,s_1)][1-\alpha_2(x,s_1,s_2)] \\ &\qquad \times [q_3(x,s_1,s_2,y)\alpha_3(x,s_1,s_2,y) + r_3(x)\delta(y)]ds_1 ds_2 \end{aligned}$$

with probability of rejecting the third proposal as r_3. Since the first and second terms already satisfied detailed balance and $r_3(x)\delta_x(y)$ is $x \leftrightarrow y$ symmetric, maintaining detailed balance reduces to plugging in deterministic proposal kernels q_1, q_2, q_3 and satisfying

$$\begin{aligned}
&\int\int \tilde{\pi}(x)\delta(s_1 - F_1(x))[1-\alpha_1(x, F_1(x))]\delta(s_2 - F_2(x)) \\
&\qquad \times [1-\alpha_2(x, F_2(x))]\delta(y - F_3(x))\alpha_3(x, F_3(x))ds_1 ds_2 = \\
&\int\int \tilde{\pi}(y)\delta(s_1' - F_1(y))[1-\alpha_1(y, F_1(y)]\delta(s_2' - F_2(y))[1-\alpha_2(y, F_2(y))] \\
&\qquad \times \delta(x - F_3(y))\alpha_3(y, F_3(y))ds_1' ds_2'
\end{aligned}$$

for dummy variables s_1, s_2, s_1', s_2'. Following the same steps to derive α_2 and requiring $y = F_3(x)$, the highest rate of acceptance is

$$\alpha_3(x, F_3(x)) = \min\left(1, \frac{\tilde{\pi}(F_3(x))}{\tilde{\pi}(x)} \frac{[1-\alpha_1(F_3(x), F_1(F_3(x)))]}{1-\alpha_1(x, F_1(x))} \frac{[1-\alpha_1(F_3(x), F_2(F_3(x)))]}{1-\alpha_1(x, F_2(x))}\right).$$

This again reveals the dependencies on ghost points $F_1(F_3(x)), F_2(F_3(x))$.

A.3.4 kth Proposal Acceptance Probability

Following the same pattern, the general acceptance probability of transitioning from state x to the kth proposal y after rejecting $k-1$ previous proposals is

$$\alpha_k(x, F_k(x)) = \min\left(1, \frac{\tilde{\pi}(F_k(x))}{\tilde{\pi}(x)} \prod_{i=1}^{k-1} \frac{1-\alpha_1(F_k(x),\ F_i(F_k(x)))}{1 \quad - \quad \alpha_1(x, F_i(x))}\right).$$

To recover the original acceptance probability in Equation 2, we write out the current state x as (θ, ρ'), the proposed state y as $(\theta^{\text{pr}}, \rho^{\text{pr}}) = F_k(\theta, \rho')$, and rewrite the Gibbs density $\tilde{\pi} \propto \pi(\theta)\text{normal}(\rho \mid 0, M)$ as

$$\alpha_k(\theta, \rho', F_k(\theta, \rho')) = \min\left(1, \frac{\pi(\theta^{\text{pr}})\text{normal}(\rho^{\text{pr}} \mid 0, M)}{\pi(\theta)\text{normal}(\rho' \mid 0, M)} \cdot \prod_{i=1}^{k-1} \frac{1-\alpha_i\big(\theta^{\text{pr}}, \rho^{\text{pr}}, F_i(\theta^{\text{pr}}, \rho^{\text{pr}})\big)}{1-\alpha_i\big(\theta, \rho', F_i(\theta, \rho')\big)}\right).$$

B ADDITIONAL NEAL'S FUNNEL EXPERIMENTS

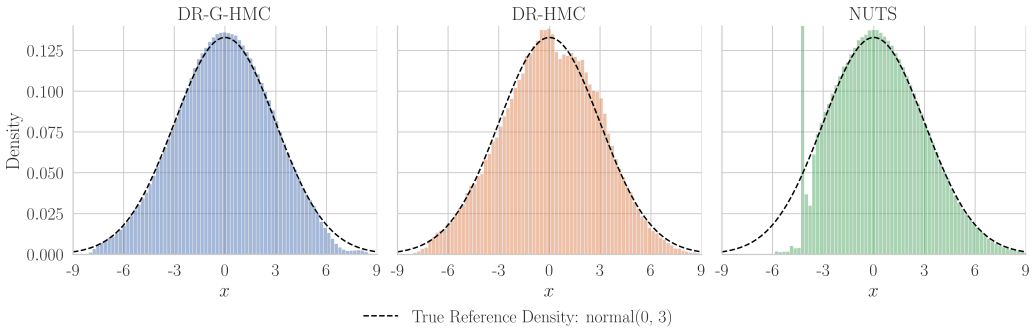
B.1 Details of Figure 1

Each (x, y_i) coordinate of Figure 1 represents a point in parameter space of the 10 dimensional Neal's funnel density: every (x, y_i) coordinate is transformed into a 10 dimensional parameter vector $\theta = \{x, y\}$ by repeating $y = \{y_i, \dots y_i\}$ nine times.

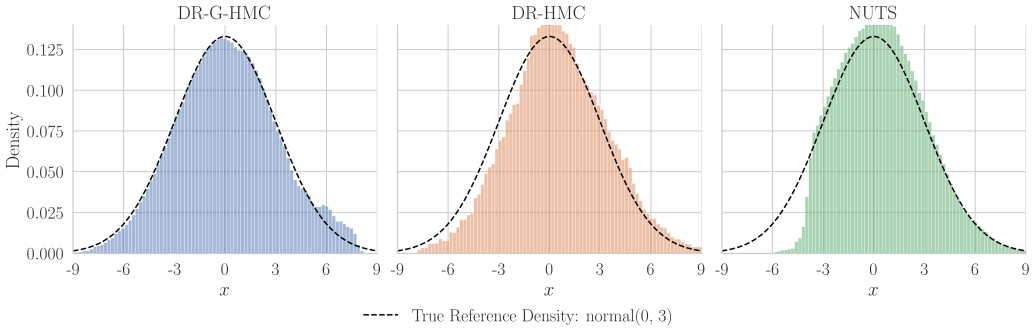
In Figure 1a we compute the negative log density and the condition number of its Hessian at each coordinate θ. In Figure 1b, we initialize DR-G-HMC from each coordinate θ and generate a single sample. This sample is generated by making sequential proposals attempts with decreasing step sizes. We record the mean step size that generates an *accepted* proposal from 100 such trials. To avoid proposal rejections, DR-G-HMC uses parameters $\epsilon = 2, r = 4, K = 10$ for many proposal attempts K that reach especially small step sizes ϵ_k; see Algorithm 1.

B.2 Neal's funnel in higher dimensions

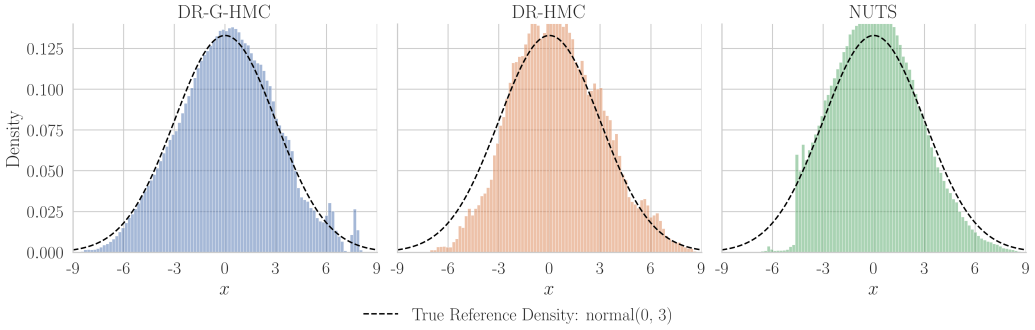
We sample from Neal's funnel in 50, 100, and 250 dimensions with 100 chains each run for $1,000,000$ gradient evaluations. In Figure 6 we plot the histogram of the log scale parameter x, with known density $x \sim \text{normal}(0, 3)$, aggregated across all chains. We observe similar behavior to the 10 dimensional setting: delayed rejection methods sample deep into the neck of the funnel ($x \ll 0$) while NUTS cannot, despite enormous amounts of compute.



(a) $50D$ **Neal's funnel.**



(b) $100D$ **Neal's funnel.**



(c) $250D$ **Neal's funnel.**

Figure 6: Even in higher dimensions, only delayed rejection methods like DR-G-HMC and DR-HMC can sample deep into the neck of the funnel ($x \ll 0$), while NUTS cannot.

C TARGET DENSITY DETAILS

C.1 Eight Schools

The eight schools data measures the impact of test-preparation interventions on student test scores across eight different schools (Rubin 1981; Gelman, Carlin, et al. 2021). The data contains the average difference in pre-test and post-test scores y_i and standard deviation σ_i for the ith school. The model parameterizes each school with an efficacy θ_i drawn from a hierarchical normal prior with unknown location μ and scale τ. The generative model consists of

$$\begin{aligned} \mu &\sim \text{normal}(0, 5) \qquad & \tau &\sim \text{cauchy}_+(0, 5) \\ \theta_i &\sim \text{normal}(\mu, \tau) \qquad & y_i &\sim \text{normal}(\theta_i, \sigma_i) \end{aligned}$$

with parameters $\theta = \{\mu, \tau, \theta_1 \dots \theta_8\}$. Eight schools is a classic problem in hierarchical Bayesian modeling . It is challenging because τ dictates the pooling of the treatment effects θ_i across schools, creating a multiscale, funnel-like geometry between them.

C.2 Normal100

We consider a posterior over a 100-dimensional normal distribution with mean zero and a dense covariance matrix Σ defined by $\Sigma_{ij} = \rho^{|i-j|}$ for some fixed $\rho \in (0, 1)$. This simulates a covariance matrix with high correlation between adjacent elements that slowly decreases between for distant dimensions i and j. The generative model is

$$\begin{aligned} \Sigma_{ij} &= \rho^{|i-j|} \quad i = 1 \dots N, j = 1 \dots N \\ y &\sim \text{normal}(0, \Sigma) \end{aligned}$$

for parameters $\theta = \{\rho, y_1 \dots y_N\}$.

C.3 Stochastic Volatility

Stochastic volatility models seek to model the volatility, or variance, on the return of a financial asset as a latent stochastic process in discrete time (Kim, Shephard, and Chib 1998). We are given mean corrected returns y_t at T equally spaced time steps and seek to sample the latent log volatility h_t, the mean log volatility μ, and persistence of the volatility term ϕ. The generative model is

$$\begin{aligned} \phi &\sim \text{uniform}(-1, 1) \qquad & \sigma &\sim \text{Cauchy}(0, 5) \qquad & \mu &\sim \text{Cauchy}(0, 10) \\ h_1 &\sim \text{normal}(\mu, \frac{\sigma^2}{1-\phi^2}) \qquad & h_t &\sim \text{normal}(\mu + \phi(h_{t-1} - \mu), \sigma^2) \quad t = 2, 3, \dots T \\ & & y_t &\sim \text{normal}(0, e^{h_t}) \quad t = 2, 3, \dots T \end{aligned}$$

with parameters $\theta = \{\mu, \sigma, \phi, h_1, \dots, h_T\}$. The hierarchical prior on the volatility parameters induces strong correlation.

C.4 Item Response Theory

Item response theory models how students answer questions on a test depending on student ability, question difficulty, and the discriminatinative power of the questions (Gelman and Hill 2006). For I students and J questions we are given the binary correctness y_{ij} of student i's answer on question j. The model uses the mean-centered ability of the student i as α_i, the difficulty of question j as β_j, and discrimination of question j as θ_j. These terms are combined into something like a logistic regression (but with a multiplicative discrimination parameter). The generative model is

$$\begin{aligned} \sigma_\theta &\sim \text{Cauchy}(0, 2) \qquad & \theta &\sim \text{normal}(0, \sigma_\theta) \\ \sigma_a &\sim \text{Cauchy}(0, 2) \qquad & a &\sim \text{lognormal}(0, \sigma_a) \\ \mu_b &\sim \text{normal}(0, 5) \qquad & \sigma_b &\sim \text{Cauchy}(0, 2) \qquad & b &\sim \text{normal}(\mu_b, \sigma_b) \\ y_i &\sim \text{Bernoulli-Logit}(a_i * (\theta - b_i)) \quad i = 1, \dots, N \end{aligned}$$

for parameters $\theta = \{\sigma_\theta, \theta, \sigma_a, a, \mu_b, \sigma_b, b\}$.

D ADDITIONAL SAMPLER EXPERIMENTS

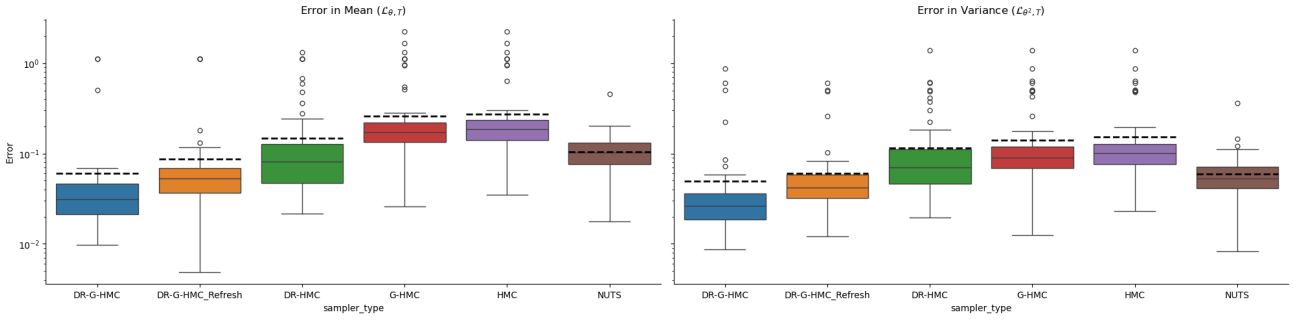
We provide an extensive comparison of samplers including HMC, G-HMC, DR-HMC, DR-G-HMC, NUTS, and DR-G-HMC with complete momentum refreshment (default DR-G-HMC has *partial* momentum refreshment). Note that DR-G-HMC with complete momentum refresh is equivalent to DR-HMC with one leapfrog step per proposal.

We compute error in mean and variance for these samplers on a variety of multiscale and non-multiscale densities. Recall that multiscale densities are commonly induced by hiearchical probablistic models.

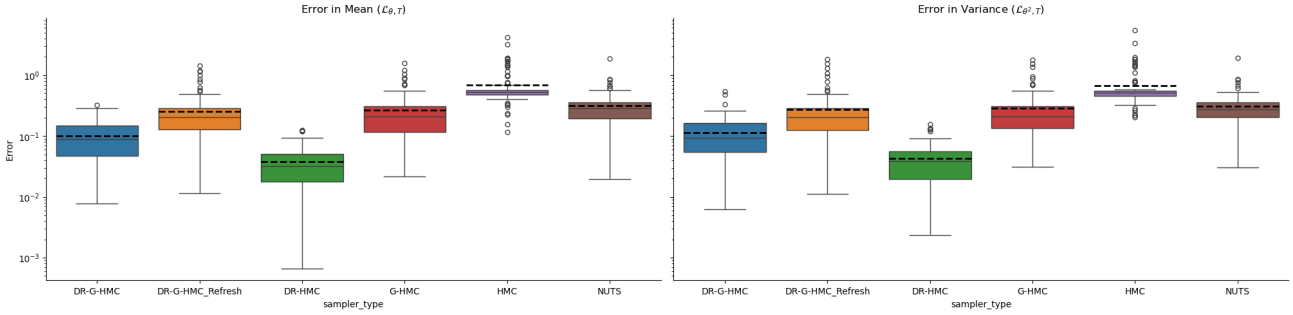
We examine multiscale densities (eight schools and banana) in Figure 7 and observe that all delayed rejection methods outperform G-HMC, HMC, and NUTS due to their adaptive step size selection.

We examine non-multiscale densities (irt-2pl and normal100) in Figure 8 and observe that DR-G-HMC, unlike DR-HMC, performs comparably to NUTS.

Across both sets of densities, we observe that *partial* momentum refresh (blue) outperforms *complete* momentum refresh (orange) in DR-G-HMC. This indeed suggests that continued motion, facilitated by partial momentum, is key to the performance of DR-G-HMC.

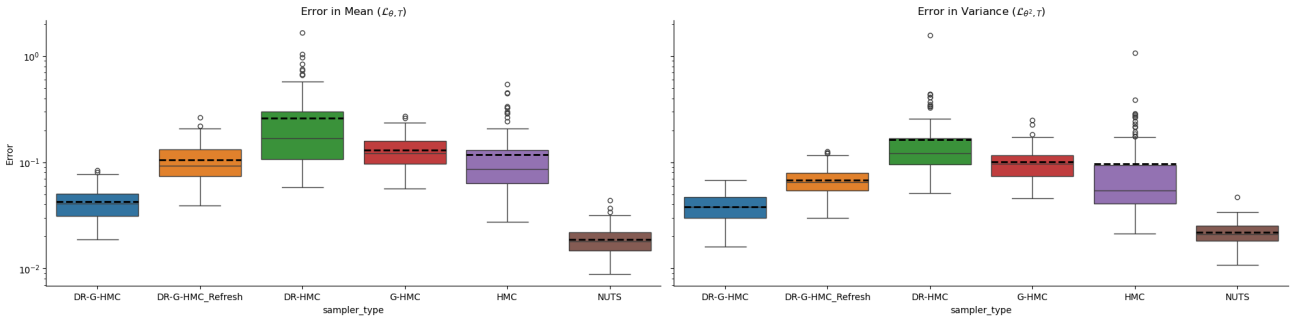


(a) Eight Schools

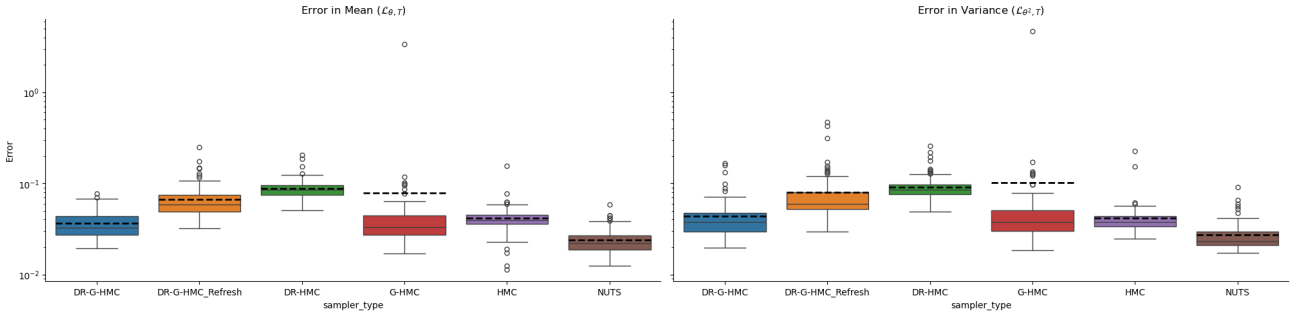


(b) 2 Dim Banana

Figure 7: **Multiscale densities.** Delayed rejection methods achieve low error on multiscale densities.



(a) Normal 100



(b) Item Response Theory

Figure 8: **Non-multiscale densities.** DR-G-HMC, unlike DR-HMC, achieves low error on non-multiscale densities.

E ADDITIONAL TUNING PARAMETER EXPERIMENTS

In this section we investigate how robust DR-G-HMC and DR-HMC are to the tuning of the following hyperparameters: max proposals K, step size factor c, and reduction factor r. See section 4 for details on these hyperparameters.

We run each sampler for $T = 10^5$ gradient evaluations on the funnel10, eight schools, and normal100 posteriors and compute the error in mean ($\mathcal{L}_{\theta,T}$) and variance ($\mathcal{L}_{\theta^2,T}$). We find that while both samplers are relatively robust, DR-G-HMC is more robust in multiple settings.

E.1 Maximum Proposals K

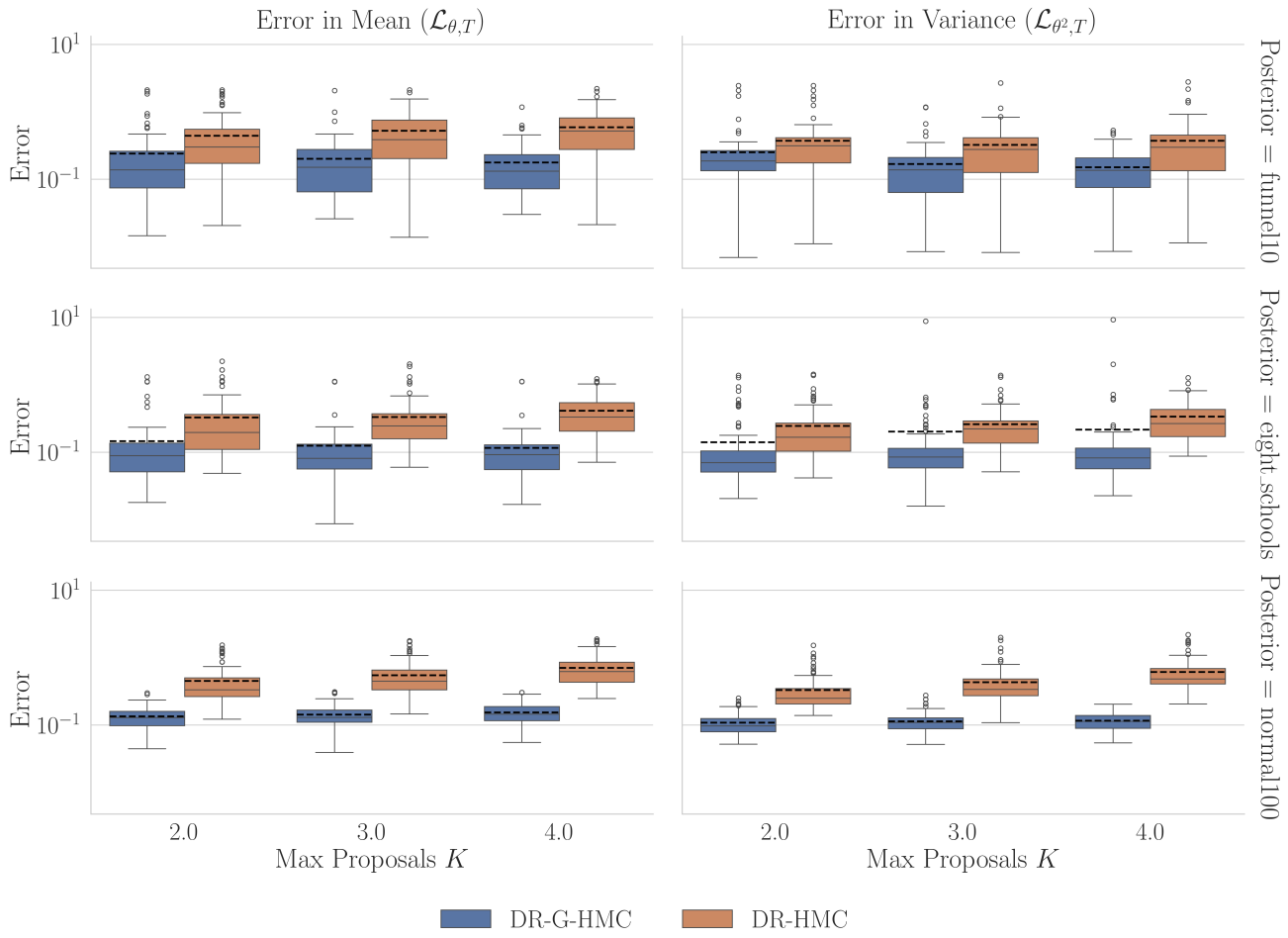


Figure 9: **DR-G-HMC is robust to the maximum number of proposals attempts K.** Error in mean ($\mathcal{L}_{\theta,T}$) and variance ($\mathcal{L}_{\theta^2,T}$) is shown for 100 chains of various posterior densities. Visual elements represent the following: dashed black line is the mean, solid gray line is the median, colored box is the $(25, 75)$th percentile, whiskers are 1.5 times the inter-quartile range, and bubbles are outliers.

We consider maximum proposal attempts $K = [2, 3, 4]$ in Figure 9. Both DR-G-HMC and DR-HMC errors are mostly constant across different K values. However, the average error across DR-HMC chains (dashed black line) increases with more proposal attempts, especially in the normal100 posterior. DR-G-HMC errors are stable throughout.

E.2 Reduction Factor r

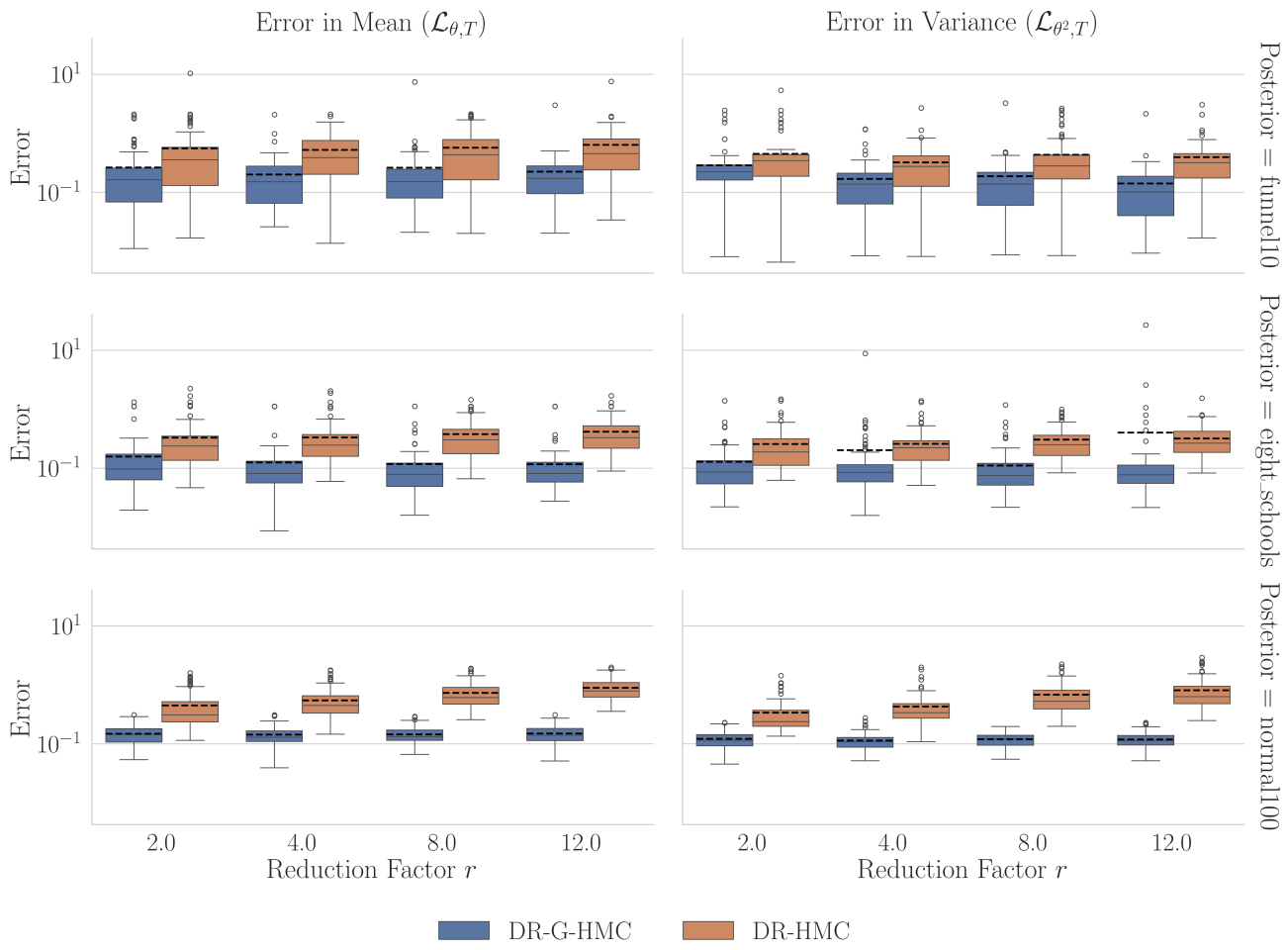


Figure 10: **DR-G-HMC is robust to tuning of the reduction factor r.** Error in mean ($\mathcal{L}_{\theta,T}$) and variance ($\mathcal{L}_{\theta^2,T}$) is shown for 100 chains of various posterior densities. Visual elements represent the following: dashed black line is the mean, solid gray line is the median, colored box is the $(25, 75)$th percentile, whiskers are 1.5 times the inter-quartile range, and bubbles are outliers.

We consider step size reduction factors $r = [2, 4, 8, 12]$ in Figure 10. Both DR-G-HMC and DR-HMC errors are relatively stable across different r values. However, the average error across DR-HMC chains (dashed black line) increases with more proposal attempts in the normal100 posterior. DR-G-HMC performance is constant everywhere except for variance of the eight schools posterior. For $r = 4, 12$, DR-G-HMC has a single outlier that dramatically increase its average error across chains (dashed black line).

E.3 Step Size Factor c

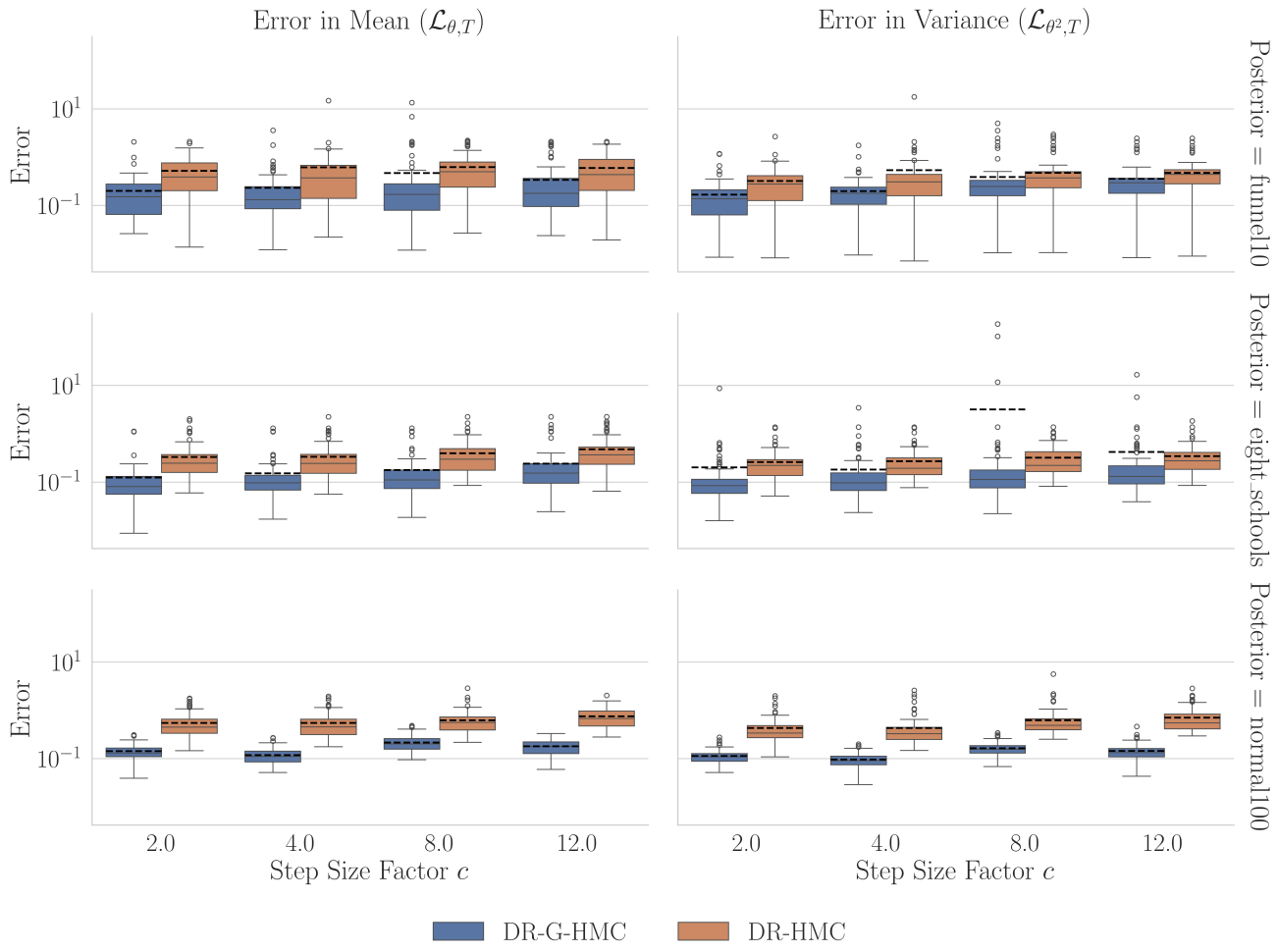


Figure 11: **DR-G-HMC is robust to tuning of the reduction factor r.** Error in mean ($\mathcal{L}_{\theta,T}$) and variance ($\mathcal{L}_{\theta^2,T}$) is shown for 100 chains of various posterior densities. Visual elements represent the following: dashed black line is the mean, solid gray line is the median, colored box is the $(25, 75)$th percentile, whiskers are 1.5 times the inter-quartile range, and bubbles are outliers.

We consider the step size factor $c = [2, 4, 8, 12]$ in Figure 11 that is used to compute the initial step size $\epsilon = c\epsilon_{\text{NUTS}}$ for NUTS's adapted step size ϵ_{NUTS}. The average error across DR-HMC chains (dashed black line) increases with a larger step size factor c in the normal100 posterior. DR-G-HMC performance is constant everywhere except for variance of the eight schools posterior. For $c = 8$, DR-G-HMC has a few outliers that dramatically increase its average error across chains (dashed black line).

F WALLCLOCK RUNTIME OF SAMPLERS

Profiling experiments reveal the runtime of our DR-G-HMC sampler is trivially longer than the runtime of DR-HMC. The overhead of DR-G-HMC arises from resampling the momentum at every iteration (not from computing the recursive acceptance probability).

For a fair comparison, we investigate n iterations of DR-G-HMC and a single iteration of DR-HMC with n leapfrog steps. In this setup, both samplers compute n gradients of the log density $\nabla \log p(x)$. Because the gradient is computed with automatic-differentiation, we get the log density $\log p(x)$ for free – this is later used to cheaply compute the acceptance probability.

	Neal's Funnel ($d = 250$)	**IRT-2PL** ($d = 144$)
DR-G-HMC	$0.0012(\pm 0.0003)$	$0.0019(\pm 6e-05)$
DR-HMC	$0.0009(\pm 0.0037)$	$0.0013(\pm 1.6e-4)$

Table 2: **Wallclock runtime of DR-G-HMC and DR-HMC samplers.** Mean and standard deviation of a single iteration are reported, averaged over 1000 trials and measured in seconds.

However, DR-G-HMC resamples the momentum ρ from a normal distribution n times while DR-HMC only does so once. This is the source of the minor runtime differences between DR-G-HMC and DR-HMC.

The exact slowdown of DR-G-HMC depends on the relative amount of time between resampling the momentum $\rho \sim \text{normal}(\cdot, \cdot)$ vs computing the gradient of the log density $\nabla \log p(x)$. This is impacted by the number of leapfrog steps n and how long it takes to compute the gradient of the log density $\nabla \log p(x)$. The latter involves differentiating through the prior and likelihood, which can be arbitrarily complex.

Table 2 shows runtime for DR-G-HMC is marginally slower than DR-HMC for Neal's Funnel and IRT-2PL target densities. However, as discussed in the main paper, DR-G-HMC outperforms DR-HMC when sampling from non-multiscale densities because of its algorithmic advantages. We thus expect that on most real-world densities DR-G-HMC will achieve a low error faster than DR-HMC.